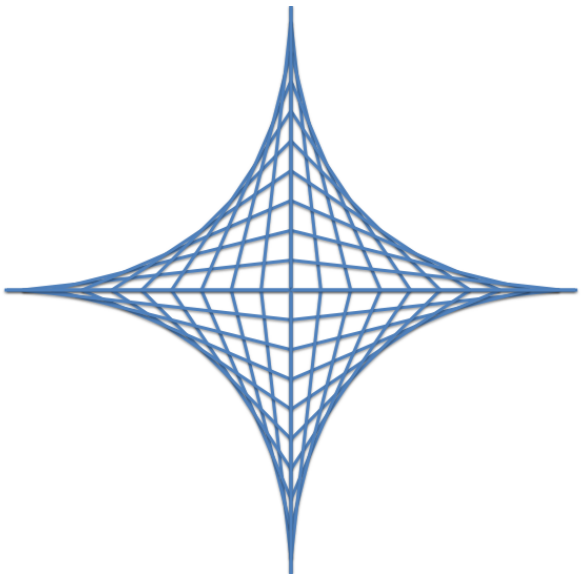
WebSpa
Single HTTP/S Request Authorisation Web Knocking
Administration Guide
_v0.8

Oliver Merki, Yiannis Pavlosoglou, Paweł Goleń, Joël Rouiller, Patryk Arciszewski

November 21, 2014



seleucus.net

1 Welcome

Thank you for taking an interest in the WebSpa. As the administration guide, this document aims to put you in control of setting up your web server to execute Operating System (O/S) commands in a covert manner. After reading this, you should have a clear understanding of how to setup WebSpa to monitor the logs of a web server, create new users and assign O/S commands to them.

This introductory section describes where this administration guide fits in the wider remit of WebSpa documentation. It gives a definition of what WebSpa is and provides an outline on how the sections of this document are structured.

1.1 This Guide & Other Supporting Documentation

The discrepant event discussed herein is web knocking. This document is one of three, with the purpose of describing how web-spa can be used. The three documents are:

- **WebSpa Administration Guide** This document describes how to setup and use the WebSpa server. It details how to create new users and add new action numbers with respective O/S commands assigned to them
- **WebSpa Specification Guide** This document describes the actual design detailing the use case, specification, requirements and actual attacks, which this tool has been engineered to withstand
- **WebSpa User Guide** This document describes how to use the WebSpa client for issuing commands through a URL request to a web server.

This administration guide aims to enable anyone who would be interested in using WebSpa to be able to setup the server side components of it. After configuring the server component of WebSpa, you'll be ready to use the corresponding client for issuing direct actions as O/S commands to it.

If this is your first time using WebSpa, please note that the server operations described in this document, will not work if a WebSpa client does not submit a web-knock to your web server in a timely manner.

Thus, knowing a client implementation goes hand-in-hand with a server instance for it, please also have a look at the WebSpa user guide document to see how the two can be used in tandem.

1.2 What is WebSpa?

WebSpa is a complete client/server tool that allows you to send premeditated commands to the system your web server is running on.

1.3 How To Read This User Guide

The scope of this administration guide is presented in eight consecutive sections, with a number of sub-sections. Each section stands as an independent entity and can be read alone. The sections are:

1. **Introduction** - An introductory section, describing the document structure as well as the other type of documentation available.
2. **Role & Responsibilities** - What an administrator of WebSpa is meant to actually do, where their responsibility starts and stops and how are they different from a WebSpa user.
3. **Obtaining WebSpa** - This section describes how to download the current version of WebSpa, the tool's respective system requirements and discusses why WebSpa does not have an installer.
4. **Running the WebSpa Server** - Takes the administrator through the process of running WebSpa in server mode for the first time, how to get help information from the tool and also discusses the files created.
5. **Service Operations** - Lists the service operations, in terms of checking the Web-Spa service status, starting the monitoring of the web server's access log file and stopping it.
6. **User Operations** - Describes how to list the current users of WebSpa, create new users and also how to toggle the active status for a user account.
7. **Action Operations** - Describes how to list the available actions for a user of WebSpa, create new actions and specify respective O/S commands to them.
8. **The WebSpa Server Configuration File** - Looks at the two java properties contained within the configuration file created and used when running WebSpa in *server mode*.
9. **The WebSpa Server Database Files** - Describes in detail the default database files created at first run, where the data added to the database is stored and what other files a WebSpa administrator might come across, relating the database.

As this document is a guide for administrators of WebSpa, no concluding section resides within this guide.

Sections 3, 4 & 5 describe the day to day operations that WebSpa administrator is due to encounter in terms of starting and stopping WebSpa, adding new users and creating new actions for them.

Section 7 discussing the setup of the configuration file is very important, as it specifies the location of the web server's log file that WebSpa will monitor. It also defines the regular expression for the line format of that log line.

Let's begin in describing web knocking through the administrator's eyes.

2 Role & Responsibilities

If you are installing WebSpa for the first time, it is important to understand this section. It describes what an administrator of WebSpa is meant to actually do, where their responsibility starts and stops and how are they different from a WebSpa user.

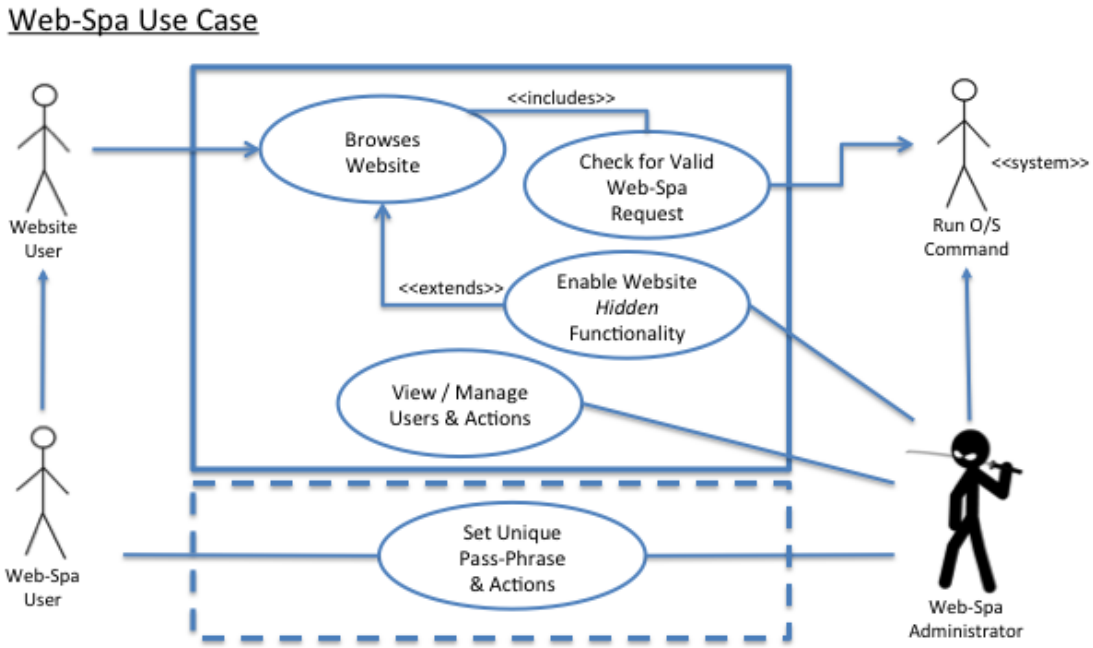


Figure 1: The WebSpa Use Case as an Extension of a Website Use Case

Overall, we can see that WebSpa is an encapsulating system that is expected to operate and function in the presence of a legitimate publicly available website. Certain types of specially crafted requests issued to the site will have the ability to execute premeditated O/S commands.

In order for that to take place, it is expected that an administrator of WebSpa is responsible for managing the respective users who have access to issue O/S commands.

2.1 Definition & Prerequisites

For a O/S user of the host on which the web server is running on, to be named a WebSpa administrator, all the following must be true:

1. They can start/stop WebSpa in '-server' mode
2. They have access to all the users pass-phrases
3. They have access to all of the user's O/S commands
4. They -at one time- had a covert channel of communication with each user

The administrator shares a secret pass-phrase with each user. They have been able to exchange this pass-phrase securely, using a different medium of communication with the user. Having this 'out-of-band' communications which occurs outside of the normal HTTP/HTTPS established communication channel is the key characteristic that any WebSpa administrator must have.

2.2 Administrator-User Out-of-band Communication

For WebSpa to operate correctly, it relies on the assumption that the event that the administrator and user have met using an 'out-out-band' channel of communication and have exchanged the user's unique pass-phrase.

This is a big assumption.

From the 3 factors of authentication that an administrator could setup a user of WebSpa with, she selects to use a pass-phrase. This is something that both the administrator and the user know (*"something you know"*) and is a secret that only they share.

Thus, the event of an administrator and a user meeting once and exchanging *"something they know"* leads to the response of a user being able to execute a number of predefined O/S commands on a web server by issuing web knock requests.

This leads to the outcome of a WebSpa user being able to establish a covert channel of communication using the web application layer, from that point forward in time. From that moment onwards, administrator and user do not need to interact again.

3 Obtaining WebSpa

This section provides important installation information to help you quickly get started using WebSpa on Windows, Mac OS and Linux operating systems.

3.1 System Requirements

You need a computer with the following:

- An Operating System (O/S) of your choice with Java Runtime Environment (JRE) 6 later installed
- 128 megabytes (MB), or greater, of RAM
- 10 MB of free hard-disk space

3.2 The Download

You can obtain the latest release of WebSpa from:

- https://sourceforge.net/projects/webspa

At the time or writing, the current version is webspa-0.8.zip (∼5MB) The download contains the following files:

- INSTALL
- LICENSE
- README
- docs
- src
- web-spa-08.jar
- web-spa.sh

Both client and server are bundled into a single executable jar. With the exception of a small shell script, some notes and documentation, all you need is the jar file to run WebSpa.

3.3 Documentation

Within the *docs* folder of the WebSpa download, you can find three files in PDF format. These are:

- docs\00-web-spa-administration-guide.pdf
- docs\00-web-spa-specification-guide.pdf
- docs\00-web-spa-user-guide.pdf

These three guides constitute the supporting documentation for this tool.

3.4 No Installer?

We have intentionally made WebSpa lightweight. As a result, no further configuration or install is required. This is part of the design, in order to keep all that is required to run this tool in a single directory.

4 Running the WebSpa Server

As this tool is a command-line tool, in this section we take you through the steps required to run WebSpa in *server mode*.

4.1 Specifying the Server Parameter

To run WebSpa in *server mode* at the command prompt type:

```
user@bath-spa:~# java -jar webspa-0.8.jar -server
```

This will produce a welcome screen and a respective holding prompt for the tool.

```
Web-Spa - Single HTTP/S Request Authorisation
version 0.8 (web-spa@seleucus.net)

This is a holding prompt, type "exit" or "x" to quit

- type "service start" to start the WebSpa server
- type "help" or "?" for more options
```

As seen in the printout above, what you have is a holding prompt; in essence, the java *main* method is running in a *while-true* loop. Type *exit* or *bye* and WebSpa will exit.

WebSpa is designed as a log listener and does not accept any direct connections or implement any additional services. It scans the access log file of a web server for a specific pattern.

4.2 Getting Help

In order to obtain help, type:

```
web-spa-server>help

Usage: [COMMAND] [OPTION]

COMMANDS/OPTIONS:

  action
    add       - assign O/S commands to users
    show      - show the actions available for a WebSpa user
  config
    show      - show the configuration parameters loaded from file
  help
    [command] - show help for that command
    shortcuts - show various command shortcuts
  pass-phrase
    show      - show the pass-phrase for a user
    modify    - change the pass-phrase of a user
  service
    start     - start WebSpa
    status    - check if WebSpa is running
    stop      - stop WebSpa
  user
    activate  - activate / de-activate user accounts
    add       - add new users to the database
    show      - show the current WebSpa users

  exit        - stop the service and exit WebSpa

EXAMPLES:

  web-spa-server> user add
  web-spa-server> config show
  web-spa-server> action create

web-spa-server>
```

For each of the commands selected you have the ability to get further help by typing help and command name. Example:

```
web-spa-server>help action
```

The above would display the relevant help for the *action* command.

4.3 New Files Created

4.3.1 Options that Do Not Create Files

If we start with only the file *web-spa-0.8.jar* in the folder */opt/web-spa-0.7/*, no other file or folder will get created, unless the **-server** command line option is specified. Ergo:

```
root@bt:/opt/webspa# java -jar web-spa-0.8.jar -version 0.8
root@bt:/opt/webspa# java -jar web-spa-0.8.jar

Web-Spa - Single HTTP/S Request Authorisation
version 0.8 (web-spa@seleucus.net)

Usage: java -jar web-spa.jar [-option]

-client         : Run the client, generate valid requests
-help           : Print this usage message
-server         : Run the server
-version        : 0.8

Examples:

java -jar web-spa.jar -client
java -jar web-spa.jar -server

root@bt:/opt/webspa#
```

Running WebSpa with the *-version* command line option, or no options specified does not yield the creation of any additional files or folders within the current directory of execution.

```
root@bt:/opt/webspa# ls -l
total 3624
-rw-r--r-- 1 root root 3707875 2013-10-13 17:35 web-spa-0.7.jar
root@bt:/opt/webspa#
```

What is more, running WebSpa with the *-client* command line option does not yield the creation of any additional files or folders within the current directory of execution.

4.3.2 Options that Do Create Files

Running WebSpa in *server mode* creates a number of files. We see what these files are by running WebSpa with the *-server* option and then exiting WebSpa by typing *exit*.

```
root@bt:/opt/webspa# java -jar web-spa-0.8.jar -server

Web-Spa - Single HTTP/S Request Authorisation
version 0.8 (web-spa@seleucus.net)

This is a holding prompt, type "exit" or "x" to quit

- type "service start" to start the web-spa server
- type "help" or "?" for more options

web-spa-server>exit

Goodbye!

root@bt:/opt/webspa#
```

The files created (assuming that you had nothing else within */opt/web-spa-0.8/* other than the jar file) are as follows:

```
root@bt:/opt/webspa# ls -ltr
total 3636
-rw-r--r-- 1 root root 3707875 2013-10-13 17:35 web-spa-0.8.jar
-rw-r--r-- 1 root root     938 2013-10-13 17:59 web-spa-config.properties
-rw-r--r-- 1 root root    1571 2013-10-13 17:59 web-spa-db.script
-rw-r--r-- 1 root root     441 2013-10-13 17:59 web-spa-db.properties
root@bt:/opt/webspa#
```

Above, we see that a total of three files that have been created. Two of them, having the prefix **web-spa-db** are database files, while one, having the prefix **web-spa-config** is the file containing all configuration properties.

These files are only required when running WebSpa with the *-server* option. None of these files are required when running WebSpa with the *-client* option.

5 Service Operations

In this section we describe how to query what the current state of the WebSpa service is and also describe how to start and stop the service within the tool.

In order for WebSpa to be looking for web knocks in new lines of your web server's access log file, it must be in *started* status.

If WebSpa is in a *stopped* status, no monitoring of the web server's log file will be taking place in real time.

Thus, despite not being a service that listens on any port or spawns up a daemon, WebSpa has two states whereby the jar file is actively monitoring (or not) the respective log file.

5.1 WebSpa Service Status

In order to get the current status of the service, type:

```
web-spa-server>service status
[2013-12-21 21-50-37] WebSpa is Stopped.
web-spa-server>
```

You straight away know as an administrator of the tool that WebSpa is not monitoring the log file of your web server.

5.2 Starting the WebSpa Service

In order to start the WebSpa service, type:

```
web-spa-server>service start
[2013-12-21 21-51-16] Attempting to start WebSpa...
[2013-12-21 21-51-16] Found access log file: /var/log/apache2/access.log
[2013-12-21 21-51-16] Creating tail listener...
[2013-12-21 21-51-16] WebSpa server started!
[2013-12-21 21-51-16] Please make sure your web server is also up
web-spa-server>
```

An error message will be displayed in case the *service start* command encounters an error:

```
web-spa-server>service start
[2013-12-21 21-52-41] Attempting to start WebSpa...
[2013-12-21 21-52-41] Access log file NOT foun.../log/apache2/access.log
[2013-12-21 21-52-41] WebSpa Server Not Started

web-spa-server>
```

5.3 Stoping the WebSpa Service

Similarly to the *service start* command, type the following to stop the service:

```
web-spa-server>service stop
[2013-12-21 21-54-36] WebSpa Server Stopped
web-spa-server>
```

This will stop the service, thus stopping the monitoring of the log file for potential web knock requests.

6 User Operations

As an administrator of WebSpa, you have at your disposal a number of operations involving the user population, or user base of WebSpa.

6.1 Showing the WebSpa Users

The simplest of the user commands, is the one listing all the existing users of WebSpa. This is a *read-only* operation from the database, querying who the users are.

```
web-spa-server>user show

Users:
------------------------------------------------------------
ID  Active  Full Name               Last Modified
------------------------------------------------------------
11  false   Oliver                  2013-09-15 21:39:25.573
------------------------------------------------------------
```

This command returns a total of 4 columns, illustrating the user's unique ID, if they are an active user of WebSpa, their full name and the last time their record was updated or modified.

A user's unique ID comes in handy when a WebSpa administrator has to select a user for the purpose assigning an action number or or activating their account.

A user's activation status being set to *false* means that despite receiving a valid web knock for that user, WebSpa will not process that request because their account is not set to the active status of *true*.

6.2 Adding a new WebSpa User

To add another WebSpa user, type the command shown below and follow the instructions on the screen:

```
web-spa-server>user add
=[Required] Enter the New User's Full Name: Yiannis
=[Required] Enter the New User's Pass-Phrase:
=[Required] Re-enter the above value:
-[Optional] Please enter the New User's Email Address: yiannis@email.com
-[Optional] Please enter the New User's Phone Number: 555-123-123
```

You will be asked to type in the user's full name and select a unique pass-phrase. Two optional fields will also appear; specifying the user's e-mail address and their phone number.

You can verify that you have added a new user, by triggering the *user show* command.

```
web-spa-server>user show

Users:
------------------------------------------------------------
ID  Active  Full Name               Last Modified
------------------------------------------------------------
11  false   Oliver                  2013-09-15 21:39:25.573
12  false   Yiannis                 2013-09-17 19:42:28.517
------------------------------------------------------------
```

6.3 Activating & De-activating Users

In order for WebSpa to receive web knocks for a particular user, their Active status must be set to *true*. Note that all new users of WebSpa are created with a default status of *Active == false*.

This means that a command submitted by the new WebSpa user would not be executed, even if the web-knock was technically valid and within the right UTC minute.

In order to toggle the status of a user, type:

```
web-spa-server>user activate

Users:
------------------------------------------------------------
ID  Active  Full Name               Last Modified
------------------------------------------------------------
11  false   Oliver                  2013-09-15 21:39:25.573
12  false   Yiannis                 2013-09-17 19:42:28.517
------------------------------------------------------------

-[Optional] Select a User ID: 11
User with ID: 11 is in-active
-[Optional] Toggle user activation [Y/n]? Y
User with ID: 11 is active

web-spa-server>
```

This sets the user with ID 11 to have an active status. Now if WebSpa receives a web knock for this user it will process it and if it is valid, it will attempt to execute the respective O/S command.

6.4 The Pass-Phrase Command

The *pass-phrase* command allows the WebSpa administrator to either display, or modify the pass-phrase of a WebSpa user.

To **reveal** the pass-phrase of a WebSpa user, type the command shown below and follow the instructions on the screen:

```
webspa-server> pass-phrase show

Users:
------------------------------------------------------------
ID  Active  Full Name               Last Modified
------------------------------------------------------------
11  true    oliver                  2014-05-14 07:32:40.514
------------------------------------------------------------

-[Optional] Select a User ID: 11
-[Optional] Show pass-phrase [y/N]: y

ID: 11
Full Name: oliver
Pass-Phrase: top-secret
Last Modified: 2014-05-14 07:32:40.514
```

To **modify** the pass-phrase of a WebSpa user, type the command shown below and follow the instructions on the screen. The *pass-phrase modify* command can be accessed directly by using the *passwd* shortcut:

```
webspa-server> pass-phrase modify

Users:
------------------------------------------------------------
ID  Active  Full Name               Last Modified
------------------------------------------------------------
11  true    oliver                  2014-05-14 10:41:17.586
------------------------------------------------------------

-[Optional] Select a User ID: 11
-[Optional] Change pass-phrase [y/N]: y
=[Required] Enter the User's New Pass-Phrase:
=[Required] Re-enter the above value:

ID: 11
Full Name: oliver

Pass-Phrase Updated Successfully
```

7 Action Operations

In version 0.8 of WebSpa, the administrator has the possibility to either add additional actions, or list the ones that have previously been added. It is not possible to delete actions in the current release.

7.1 Listing Actions

In order to list the actions that a certain WebSpa user is allowed to execute, run the *action show* command and select the respective user by entering its ID:

```
web-spa-server>action show

Users:
------------------------------------------------------------
ID  Active  Full Name              Last Modified
------------------------------------------------------------
11  true    Oliver                 2013-09-19 18:35:52.154
12  false   Yiannis                2013-09-19 18:36:11.877
------------------------------------------------------------

-[Optional] Select a User ID: 11

Actions for user with ID: 11
------------------------------------------------------------
 #  O/S Command                    Last Executed
------------------------------------------------------------
 0  service ssh start              has never been executed
------------------------------------------------------------
```

Once a valid user ID has been selected, this command returns a total of 3 columns, illustrating the action number (a number from 0 to 9 inclusive), the O/S command corresponding to that action number and the last time that this O/S command was received via a web knock and executed.

7.2 Adding Actions

In the example above, user *Oliver* does currently only has one action assigned. In order to add an additional action, type *action add* and follow the action wizard:

```
web-spa-server>action add

Users:
------------------------------------------------------------
ID  Active  Full Name               Last Modified
------------------------------------------------------------
11  true    Oliver                  2013-09-19 18:35:52.154
12  false   Yiannis                 2013-09-19 18:36:11.877
------------------------------------------------------------

-[Optional] Select a User ID: 11
The existing actions for this user are:

Actions for user with ID: 11
------------------------------------------------------------
 #  O/S Command                     Last Executed
------------------------------------------------------------
 0  service ssh start               has never been executed
------------------------------------------------------------

=[Required] Enter the new O/S Command: service ssh stop
=[Required] Select an action number for this O/S Command [0,9]: 1
```

Note that you will not be able to assign another O/S command to an existing action number. In the above example the action number *0* is already in use and can thus cannot be re-used. You can use any other single digit number that does not already have an O/S command assigned to it.

You can proceed to list the actions for that user as a re-affirmation of having added an O/S command for the action number 1.

8 The WebSpa Server Configuration File

When running WebSpa in *server mode*, there is one configuration file, named **web-spa-config.properties** which is expected to be found in the directory in which the jar file is executed in.

If the file **web-spa-config.properties** is not found in the current directory, WebSpa will create a default file. This file will have two properties within it, as well as a number of comment lines, explaining what each of these two properties do.

```
root@bt:/opt/webspa# more web-spa-config.properties
```

We can see that this file is structured like a standard java properties file. You can *more*, *vi*, or otherwise edit this file. Every time WebSpa is run in *server mode* it will attempt to read the contents of this file.

8.1 Access Log File Property

This is the file that WebSpa *tails* in order to see if web knock has been received. The default property value is set to point to the Apache access log file found in */var/log*.

```
#
# The default location within the file system
# where the access log file of the web server
# resides.
#
# Some examples of default log files can be found below.
# Windows XP
# access-log-file-location
# =C:/Program Files/Apache Software Foundation/
# Apache2.2/logs/access.log
#
access-log-file-location=/var/log/apache2/access.log
```

Thus, the property **access-log-file-location** is a very important property in that it tells WebSpa where in the file system to find the web-server's access log file.

Specifying a non-existent access log file, will cause WebSpa not to work in server mode, as it would not be able to tail a file for valid web knock requests.

As per the commented lines seen above, there is nothing preventing you from pointing WebSpa to any other file on the O/S. This file could be the access log file of a web-server other than Apache, such as IIS, etc.

8.2 Logging Regex Request Property

The second and final property found within the WebSpa configuration file is a regular expression aiming to extract two key bits of information from each line in the access log file.

The **logging-regex-for-each-request** property is the regular expression indicating where within each line of the access log file the IP address and part of the URL are located.

```
#
# This is log regex for 'grepping' the IP Address and the
# actual non-existent URL.
#
# Note: this regex must have exactly two groups (i.e. two
# sets of brackets) the first one being the IP Address and
# the second being the HTTP GET request
#
# The following example access.log line:
# 192.168.1.65 - - [01/May/2011:00:55:40 +0100]
# "GET /YouCannotBeHereRightNow! HTTP/1.0" 404 225
# would yield back:
#  192.168.1.65
#  YouCannotBeHereRightNow!
#
logging-regex-for-each-request=
^([\\d.]+) \\S+ \\S+ \\[[\\w:/]+\\s[+\\-]\\d{4}\\
] \"GET /(\\S*) HTTP\\/[1-2]\\.[0-1]\"
\\d{3} \\d+ \"[^\"]+\" \"[^\"]+\"
```

The extraction of the two parameters from each line in the log file takes place with the specified ().

Note that the regex specified must have exactly two groups (i.e. two sets of brackets) the first one being the IP Address and the second being the HTTP GET request.

If less than or more than two sets of brackets are specified in this property of the WebSpa configuration file, WebSpa will break at runtime. Horribly so.

In the standard Apache format, an example is given of an access log line being the following line.

```
192.168.1.65 - - [01/May/2011:00:55:40 +0100]
    "GET /YouCannotBeHereRightNow! HTTP/1.0" 404 225
```

Wanting to extract the IP Address *192.168.1.65* from this line we use the regex pattern:

```
^([\\d.]+)
```

Wanting to extract the GET request *YouCannotBeHereRightNow!* from this line we use the regex pattern:

```
\"GET /(\\S*)
```

Finally, notice little rules within the default regex, e.g.

```
HTTP\\/[1-2]\\.[0-1]\"
```

By default the regex pattern for WebSpa only allows for HTTP/1.0, HTTP/1.1, HTTP/2.0 and HTTP/2.1 protocol numbers to be passed as valid.

It is to the discretion of the WebSpa administrator to customise this configuration property according to the setup of the respective web server for which WebSpa is monitoring the logs.

The property **logging-regex-for-each-request** is the second of the two properties found in the WebSpa configuration file. Together with the property **access-log-file-location** they form the two properties which WebSpa relies on to operate correctly, when launched in *server mode*.

9 The WebSpa Server Database Files

When running WebSpa in *server mode*, there are a number of database files, prefixed with **web-spa-db** which are expected to be found in the directory in which the jar file is executed in.

If the two files **web-spa-db.script** and **web-spa-db.properties** are not found in the current directory, WebSpa will create each of these files from a default template.

Creating database files from the default template held within the jar file, implies that no users, actions, or respective web knock logs will be held within the database. You are in essence starting with a new, clean instance of the database.

9.1 Default Database Files Created at First Server Run

If you run WebSpa in *server* mode and simply exit the program, without adding any users or actions to it, the following two files are created.

```
root@bt:/opt/webspa# ls -ltr web-spa-db.*
-rw-r--r-- 1 root root 1571 2013-10-13 17:59 web-spa-db.script
-rw-r--r-- 1 root root  441 2013-10-13 17:59 web-spa-db.properties
root@bt:/opt/webspa#
```

These two files are part of the standard format for HSQL embedded databases and carry the definition of the four tables used by WebSpa.

9.2 Database Files Created when Adding Users

The WebSpa *server* operations rely on the standard HSQL embedded database operating in file mode. As such, no extra files are created unless they are required. This means that only once a user gets added to the database of WebSpa, do particular files holding the corresponding data get created.

Within WebSpa, adding a user is done by issuing the *user add* command, seen below.

```
web-spa-server>user add
=[Required] Enter the New User's Full Name: Yiannis
=[Required] Enter the New User's Pass-Phrase:
=[Required] Re-enter the above value:
-[Optional] Please enter the New User's Email Address: yiannis@mail.com
-[Optional] Please enter the New User's Phone Number: 555 123 445 222
web-spa-server>exit
```

We now see that two additional files have been created within the directory where we are running WebSpa.

```
root@bt:/opt/webspa# ls -ltr
total 4668
-rw-r--r-- 1 root root 3707875 2013-10-13 17:35 web-spa-0.8.jar
-rw-r--r-- 1 root root     938 2013-10-13 17:59 web-spa-config.properties
-rw-r--r-- 1 root root    1638 2013-10-13 21:31 web-spa-db.script
-rw-r--r-- 1 root root 1048576 2013-10-13 21:31 web-spa-db.data
-rw-r--r-- 1 root root     441 2013-10-13 21:31 web-spa-db.properties
-rw-r--r-- 1 root root    4725 2013-10-13 21:31 web-spa-db.backup
root@bt:/opt/webspa#
```

These are the files ending in **backup** and the **data**. We will examine how to add a user to WebSpa in the following sections.

9.3 Total List of Database Files

The WebSpa *server* operations rely on the standard HSQL embedded database setup.

As a result, the data for the WebSpa database is constituted out of up to 6 files. These files are as follows:

- **web-spa-db.script** Created by default if such a file is not detected. The statements that make up the database are saved in this file. These are mostly CREATE statements and ALTER TABLE statements.
- **web-spa-db.properties** Created by default if such a file is not detected. This file carries the WebSpa database default settings (properties). These properties are used every time WebSpa attempts to connect to the database.
- **web-spa-db.data** This file carries the actual data (in terms of users and actions) of the WebSpa database.
- **web-spa-db.log** This file is only found in the file system if WebSpa is actively running in *server mode* or WebSpa was not shut down properly. When you restart WebSpa, this file will be processed and an automatic checkpoint will be performed.
- **web-spa-db.lck** This is a lock file showing that the database is in use. This file is used for controlling access to the database, gets created when running WebSpa in *server mode* and gets deleted after a clean exit.
- **web-spa-db.backup** This is the backup file, which can be as big as the data file.

From the above enumerated files, the files ending in **log** and **lck** files should not reside within the file system, after a clean *exit* of WebSpa.

The files ending in **script** and **properties** get created by default, if they are not found when starting WebSpa in *server mode*.

Finally, the database data (users, actions, pass-phrases and O/S commands) reside within the data file and are backed up in the backup file.

10 Abbreviations

1FA	1 Factor Authentication
AES	Advanced Encryption Standard
CLI	Command Line Interface
DDoS	Distributed Denial of Service
DoS	Denial of Service
GUI	Graphical User Interface
HTTP/S	Hypertext Transfer Protocol / Secure
KiSS	Keep It Simple, Stupid
HMAC	(Keyed-) Hash Message Authentication Code
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
JAR	Java ARchive
JRE	Java Runtime Environment
NCSC	National Cyber Security Centrum
NSA	National Security Agency
O/S	Operating System
OTP	One-Time Password
OTT	One-Time Token
RFC	Request for Comment
SHA	Secure Hash Algorithm
SPA	Single Packet Authorization
TCP	Transmission Control Protocol
TOTP	Time-Based One-Time Password
UCS	Universal Character Set
UML	Unified Modeling Language
URL	Uniform Resource Locator
UTC	Universal Time Coordinated (Unofficial)
UTF-8	UCS Transformation Format

11 Web-Spa Server Help Information

11.1 Jar File Help

Running the web-spa jar with no command line parameters yields:

```
Web-Spa - Single HTTP/S Request Authorisation
version 0.8 (web-spa@seleucus.net)

Usage: java -jar web-spa.jar [-option]

-client         : Run the client, generate valid requests
-help           : Print this usage message
-server         : Run the server
-version        : 0.8

Examples:

java -jar web-spa.jar -client
java -jar web-spa.jar -server
```

To run web-spa in *server mode* at the command prompt type:

```
user@bath-spa:~# java -jar web-spa-0.8.jar -server
```

This will produce a welcome screen and a respective holding prompt for the tool.

11.2 Web-Spa Server Help

Typing *'help'* when web-spa is running in *server mode* yields:

```
web-spa-server>help

Usage: [COMMAND] [OPTION]

COMMANDS/OPTIONS:

  action
    add       - assign O/S commands to users
    show      - show the actions available for a web-spa user
  config
    show      - show the configuration parameters loaded from file
  help
    [command] - show help for that command
    shortcuts - show various command shortcuts
  service
    start     - start web-spa
    status    - check if web-spa is running
    stop      - stop web-spa
  user
    activate  - activate / de-activate user accounts
    add       - add new users to the database
    show      - show the current web-spa users

  exit        - stop the service and exit web-spa

EXAMPLES:

  web-spa-server> user add
  web-spa-server> config show
  web-spa-server> action create

web-spa-server>
```

11.3 Server Action Help

Typing *'help action'* when web-spa is running in *server mode* yields:

```
web-spa-server>help action

The ACTION command allows you as a web-spa administrator to:

1. Add New Actions
2. Show the Current Actions

1. Adding New Action 'action add'

When adding new action, you MUST:

1.1 Specify the O/S command

This is used to map the action number to an actual Operating
System (O/S) command.

It is important that the user assigned to execute this command
would be authorised to do so under the permissions that the
web-spa jar file is run under.

Also a valid, active user must already exist in order for the
O/S command to run successfully. Please user the command
'user add' to first add a user before using the 'action add'
command.

1.2 Specify a Unique Action Number [0-9]

Each command must be mapped to a single digit number. This
number is a value that will be asked to be entered by a web-spa
user utilising the web-spa client.

Each O/S command is mapped to a unique number and each user can
only have 10 commands allocated to their pass-phrase.

If more than 10 O/S commands are required, a web-spa
administrator would have to create a secondary user account
with a separate unique pass-phrase.

2. Show the Current Actions 'action show'
```

```
This option displays the existing O/S commands for a user of
web-spa. It lists each command number, the respective actual
O/S command and the time it was last executed.

|Example Output:
|Actions for user with ID: 12
|------------------------------------------------------------
| #  O/S Command                      Last Executed
|------------------------------------------------------------
| 0  sh -c "service apache2 star...  has never been executed
| 1  sh -c "service apache2 stop...  has never been executed
| 2  sh -c "/etc/init.d/ssh star...  has never been executed
| 3  sh -c "/etc/init.d/ssh stop...  has never been executed
| 4  sh -c "service mysql start;...  has never been executed
| 5  sh -c "service mysql stop; ...  has never been executed
|------------------------------------------------------------

In this example you can see web-spa being setup to allow for
the start/stop of the apache web server, SSH and also the
MySQL database.

Please note that 'action show' only works on an individual
user basis: You can only list the actions of one user at a
time.


web-spa-server>
```

11.4 Server Configuration Help

Typing *'help config'* when web-spa is running in *server mode* yields:

```
web-spa-server>help config

The configuration of web-spa, when running in server mode can
be found in a file called:

'web-spa-config.properties'

This file contains two (2) very important properties:

1. access-log-file-location
2. logging-regex-for-each-request

1. The access-log-file-location property is the location within
the filesystem where the access log file of the web server is.

This has a default value:
'/var/log/apache2/access.log'

Which can be modified before starting web-spa in server mode.

2. The logging-regex-for-each-request property is the regular
expression indicating where within each line of the access log
file the IP address and part of the URL are located.

This has a default value:
'^([\\d.]+) \\S+ \\S+ \\[[\\w:/]+\\s[+\\-]\\d{4}\\]
\"GET /(\\S*) HTTP\\/[1-2]\\.[0-1]\" \\d{3} \\d+ \"[^\"]+\"
\"[^\"]+\"'

Which can also be modified before starting web-spa in server
mode.

Please note that the regex specified must have exactly two
groups (i.e. two sets of brackets) the first one being the IP
Address and the second being the HTTP GET request.

web-spa-server>
```

11.5 Server Service Help

Typing *'help service'* when web-spa is running in *server mode* yields:

```
web-spa-server>help service

The SERVICE command allows you to start, stop and query the
current status of the web-spa service. It does this in the
following way:

1. Start the Service 'service start'
2. Stop the Service 'service stop'
3. Query the Status 'service status'

1. Starting the Service 'service start'

This enables to 'tail' log listening to begin on the file of
your choice. The file is specified within the first property
 of the config setup. Please type 'show config' to understand
 which file this is.

2. Stopping the Service 'service stop'

This disables the log listening service and stops web-spa from
accepting any more commands in the form of web-knocks.

3. Querying the Service Status 'service status'

This will return a message of the service either being started
or stopped. An example output can be seen below:

|web-spa-server>service status
|
|Web-Spa is Stopped
|
|web-spa-server>

In this example the web-spa service is stopped and no further
O/S commands can be received as web-knocks.

web-spa-server>
```

11.6 Server User Help

Typing *'help user'* when web-spa is running in *server mode* yields:

```
web-spa-server>help user

The USER command allows you as a web-spa administrator to:

1. Add New Users
2. Show the Current Users
3. Activate / De-activate User Accounts

1. Adding New Users 'user add'

When adding new users, you MUST:

1.1 Specify the user's name

This is used for reference purposes and is not actively
entered in the generation of a new web-spa request.

1.2 Specify the user's unique pass-phrase

What identifies each user uniquely is the pass-phrase you
select. Also the security of the web-spa channel is defined
by the strength of the password you set.

Finally, there are two optional fields that you MAY populate,
but have the choice of leaving blank. These are the user's
e-mail address and phone number.

2. Show the Current Users 'user show'

This option displays the existing users of web-spa. It lists
their unique 'ID', if their account is 'Active', each user's
'Full Name' and the last modification date of their account.

|Example Output:
|------------------------------------------------------------
|ID  Active  Full Name              Last Modified
|------------------------------------------------------------
|11  true    Jennifer               2013-09-01 22:43:40.459
|12  true    Paul                   2013-09-01 22:43:51.020
|13  false   Sandra                 2013-09-01 22:44:25.884
```

```
|_____________________________________________________________

3. Activate / De-activate User Accounts 'user activate'

This option toggles the activation and de-activation of a
user account.

Selecting a user with an existing ID (e.g. 11) will give you
the option to toggle their activation status.

Thus, if a user is active you can de-activate their account
and vice-versa.
web-spa-server>
```