DNSSHIM [1]

DNSSHIM is an open-source software that implements the Domain Name Name System (DNS) protocol for the Internet. Its main feature is to work as a Hidden Master nameserver, that is, provide information only to authoritative slave servers. Furthermore he has the ability to manage, store and automatically resign zones using the security extension DNSSEC.

Features

- DNSSEC
- Zone Transfer via AXFR and IXFR
- TSIG Support
- Zone Notification with DNS NOTIFY
- Automatic Configuration of Slave Nameservers (BIND only)

Compatibility

DNSSHIM follows the standards defined by RFCs for communication with the slave servers, which includes zone transfer via AXFR or IXFR, using TSIG keys. That makes DNSSHIM compatible to work with most major nameservers software on the market.

The feature of automatic configuration of slave nameservers is currently available only for servers running BIND.

1 Installation

This document contains information for the compilation and installation of DNSSHIM. All instructions are valid for all platforms and operating systems that have a Java Virtual Machine (JVM) installed and configured correctly.

1.1 Requirements

1.1.1 Basic Requirements

The following requirements are necessary to use:

- JRE 6 (or greater)

[1] Version: *Rev* : 1355

1.1.2 Automatic Configuration of Slaves (OPTIONAL)

In order for the automatic configuration of slave nameservers to work, the following tools are required:

Server Hidden Master

1. SSH client
2. Rsync
3. Bourne Shell

Server Slave

1. SSH server
2. Rsync
3. Bind (with DNSSEC bis support)

1.2 Download

All DNSSHIM files, including source, binaries and docs, can be downloaded at `http://registro.br/dnsshim/` .

1.3 Configuring

DNSSHIM configuration files, by default, are located under the user's home directory. To change it, you can set an enviroment variable named `DNSSHIM_HOME`. In order to provide a more scalable architecture all files related to zones are hashed and stored under a three level directory structure. This behaviour exists in both main directories under `DNSHIM_HOME`: `/xfrd` and `/signer`.

1.3.1 Signer

All signer configuration files and private keys are located under the `signer` directory. Each zone managed by DNSSHIM has a private key located under:

`signer/FIRST_LEVEL/SECOND_LEVEL/THIRD_LEVEL`

The key name format is:

YYYYMMDDHHmmSS`-ZSK-257-`*KEYTAG*`.private`

All key files are extremely important, keep them safe.
The file `signer.properties` has two entries described below:

allowed_hosts_regex: An regular expression that defines which IP address can access signer. The default value is *.+* (anyone)

server_port: The port signer server will listen on.

1.3.2 XFRD Server

All XFRD configuration files are located under the `xfrd` directory. This directory holds public keys, zone informations like resource records and individual properties, main server configurations and slave informations. The file xfrd.properties is the main configuration file for XFRD. All entries are described bellow:

allowed_hosts_regex An regular expression that defines which IP address is allowed to access XFRD server. The default value is *.+* (anyone).

dns_server_port The TCP and UDP port that listen DNS queries. Default value is 53.

scheduler_high_priority (in hours) All signatures that expire in less than *scheduler_high_priority* will be resigned immediately. Default is 120 hours (5 days).

scheduler_low_priority (in hours) All signatures that expire between *scheduler_high_priority* and *scheduler_low_priority* will be rescheduled for resigning in some time within that interval. Default is 240 hours (10 days).

signer_cert_file The file that contains a TLS certificate. By default the communication between xfrd and signer uses an embedded TLS certificate.

signer_cert_password Only necessary if the option above (*signer_cert_file*) is defined.

signer_host Host where signer server is running. Default value is *localhost*.

signer_port The TCP port that the signer server is listen. Default is 9797.

slave_sync_enabled (true or false) If true the server will execute SlaveSync.sh script in order automatically configure slave nameservers.

slave_sync_period (in seconds) The period that the server rewrites zone configuration files used for configuration of the slaves. This option is only necessary if *slave_sync_enabled* option is true.

slave_path The directory where zone configuration files are located in the slaves.

slave_sync_path The path where the script SlaveSync.sh is located. Default value is `./`.

slave_user User used by SlaveSync.sh script to connect to slaves. Default is empty. Change it if you want enable automatic provisioning.

tsig_fudge The default fudge value for outgoing packets.

ui_cert_file The TLS certificate file used in communications between xfrd server and client. By default the communication use an embedded TLS certificate.

ui_cert_password Only necessary if the option above (*ui_cert_file*) is defined.

ui_port The TCP port where the server will receive client's requests.

2 Client

In order to interact with DNSSHIM, it is necessary to have a client that supports its protocol. A python client library, named *pydnssshim*,that fully supports all commands, is distributed separately and is available at the DNSSHIM website. You are free to use it or implement your own client. Please visit DNSSHIM website for more information about clients.

3 Deployment

Starting Signer

```
$ java -jar -Dlog4j.configuration=log4j-signer.properties dnsshim-signer.jar [2]
```

Starting XFRD Server

```
$ java -jar -Dlog4j.configuration=log4j-xfrd.properties dnsshim-xfrd.jar
```

Important note: in order to start using DNSSHIM effectively, you must create a user and login to the server. These operations, as well as many others, should be performed by a client application. For more information see the section 2.

3.1 Troubleshooting

Could not determine local IP address. Using loopback In the */etc/hosts* file, check the IP address for the hostname of the machine.

3.2 Automatic Configuration of Slave Nameservers

DNSSHIM has a feature which allows for automatic configuration of slave nameservers, so that once a new zone is created in DNSSHIM, the slave will be authoritative for that zone within a period of time with no further intervention. It works by automatically synchronizing a file with all the configuration for the zones a slave should be serving and by sending a signal to the slave to reload BIND periodically.
In order to setup for automatic configuration of slave nameservers, there are several step that should be followed:

[2]For convenience, DNSSHIM comes with two log4j pre-configured files. If you prefer you can use your own.

1. Make sure that BIND is properly installed on the hosts that are supposed to work as slave nameservers. Also make sure all the machines, DNSSHIM and the slaves, have the tool *rsync* installed and running.

   On the DNSSHIM host

2. Open the xfrd.properties file.

3. Set the variable "slavesync_path" to the path where the SlaveSync.sh script is.

4. Make sure the SlaveSync.sh script is executable.

5. Set the variable "slave_path" to the directory where BIND is running on the slave machine.

6. Set the variable "slave_sync_enable" to "true".

7. Optionally, set the variable "slave_sync_period" to the interval (in seconds) between two slave synchronizations.

8. Start the Signer and the DNSSHIM server (3).

9. Create a TSIG key for the servers which will be synchronized using the command `new-tsig-key` in the DNSSHIM client (2).

10. Create a new slave group using the command `new-slave-group` in the DNSSHIM client (2).

11. Add the desired slave(s) to the newly created slave group using the command `add-slave` in the DNSSHIM client (2).

12. Assign the slave group to the desired zone(s) using the command `assign-slave-group` in the DNSSHIM client (2).

13. As the user who will be running DNSSHIM, create an SSH key with an empty password, which will be used to access the slave servers

    On each of the Slave hosts:

14. Create a new user named "*dnsshim*"

15. Put the public key of the SSH key generated on the DNSSHIM host into the file " /.ssh/authorized_keys"

16. Make sure that the user "*dnsshim*" has write permissions on the directory where BIND is running, because it will be saving the zone files in it.

17. Create a file named "named.zones" in the same directory of the named.conf (use "touch" command)

18. Edit the named.conf file

19. Add the following directive to the named.conf file. This tells BIND to include the file where the configuration for all DNSSHIM zones are.

    ```
    include ‘‘named.zones’’;
    ```

20. Also add the following directives to the named.conf file:

    ```
    key “tsig-key” { algorithm hmac-md5; secret “tsig-secret”; };
    server dnsshim_ip { keys “tsig-key”; };
    controls {
                        inet * allow { localhost; } keys { “tsig-key”; };
    };
    ```

    Where *tsig-key* is the name of the TSIG key assigned to the slave, *tsig-secret* is the secret of the key, *dnsshim_ip* is the IP address of the host running DNSSHIM.

    The first two lines tell BIND that DNSSHIM is using the specified TSIG key.

    The `controls` ensures that rndc running locally is allowed to control the BIND server, mainly by restarting it.

21. Put the script dnsshim_slave.sh, which comes with the DNSSHIM distribution, in the same directory where BIND is running and make sure it is executable. This script is used to create the directory structure for the zone and to remove unnecessary zone files from zones that have been already removed

22. Restart the BIND on the slave hosts. Make sure that BIND is running as the user “*dnsshim*”

4 Compilation

If you want build from source the following steps will guide you.

4.1 Requirements

The following requirements are necessary to compile:

- JDK 6 (or higher)
- Ant 1.7 (or higher)
- Apache Commons Codec 1.3 (or higher)
- Apache log4j 1.2.15 (or higher)

4.2 Build

Unzip source, in the directory DNSSHIM execute:

```
$ ant dist
```

After the execution of this target a directory named *dist* will be create containing two JARs (signer and XFRD).