# SHIVA

## (Spam Honeypot with Intelligent Virtual Analyzer)

**The Honeynet Project**

Version 0.1

Author - Sumit Sharma

Co-author - Rahul Binjve

# 1. Introduction

## 1.1 About

**S**pam **H**oneypot with **I**ntelligent **V**irtual **A**nalyzer, is an open but controlled relay spampot (i.e. a spam honeypot) written in Python2.7 built on top of Lamson framework. SHIVA is an open-source honeypot, and is released under GNU General Public Licence version 3.

SHIVA provides capabilities of collecting and analyzing all spam thrown at it. Analysis of data captured can be used to get information of phishing attacks, scamming campaigns, malware campaigns, spam botnets, etc. SHIVA is written in Python and uses MySQL for storing information.

## 1.2 General Architecture

SHIVA is divided into two parts - Receiver and Analyzer. In short, the receiver part acts as an open relay SMTP server, collects all the spam thrown at it and dumps them into a local directory. The analyzer, then, picks up the spam and proceeds with analyzing and extracting the information.

### 1.2.1 Receiver

Receiver part is quite simple to understand. Receiver is a lamson project that starts a SMTP listener on specified host address and port. Receiver is configured to dump the incoming spam into a local directory. This local directory is specified in configuration file, under "global" section as "queuepath" variable.

By default, Python's smtpd.py doesn't provide support for authentication. Therefore, some new code was written and much was borrowed from secure_smtpd.py by Benjamin E. Coe. This code added support for AUTH command in smtpd.py. Enabling/ disabling and credentials for SMTP authentication can be managed from configuration file. Options for authentication can be found in "smtpauth" section. Figure 1 provides the general code flow of receiver.

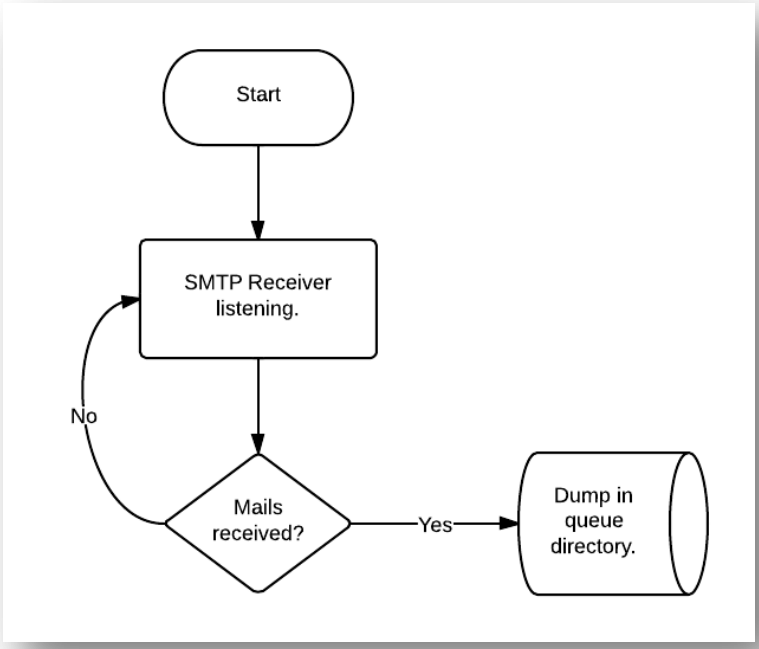


Figure 1. Code flow of receiver.

### 1.2.1 Analyzer

Analyzer is where things start to get interesting. Analyzer, again in a lamson project. When analyzer is started it starts QueueReceiver on the queue directory. This starts picking up the spam (that were dumped by receiver) and will pass it to SHIVA's custom modules. Normal code flow of Lamson is shown on Figure 2.

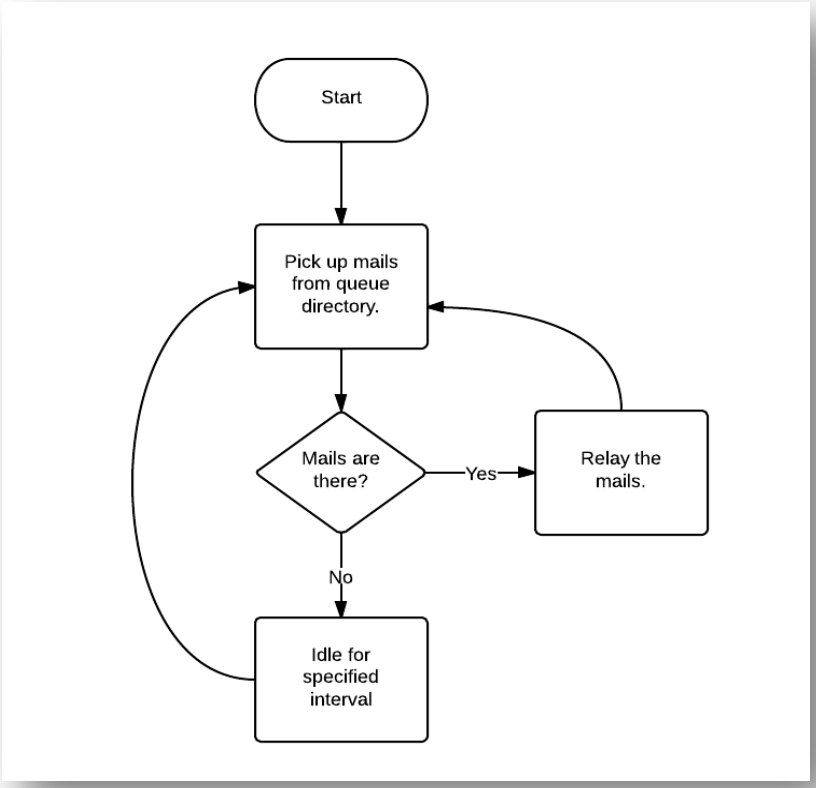


Figure 2. Normal flow of lamson.

## 1.3 Use Cases

SHIVA , other than being used as traditional open but controlled relay honeypot, can be used for other purposes too. Listed below are some of the possible usage scenarios:

- **Collecting Spam**: SHIVA can be used as an SMTP server that'll dump all the spam that it receives, into a local directory. If we start only the receiver part, then SHIVA will act as an SMTP server, and all mails will be dumped in queue directory.
- **Analyzing Stored Spam**: SHIVA can also be used to analyze the spam that might've been collected from various sources. All we need to so is to transfer all the spam into the queue folder and start the analyzer part.
- **Collecting Spam Attachments**: SHIVA can also be used as a tool to only collect the attachments from spam. To just collect attachments, you can set "localdb" option in configuration file as "False". This will not store any information in database but will continue to dump the attachments in the local directory.
- **Hpfeeds Sharing**: SHIVA can be easily configured for sharing data to hpfeeds. There's a dedicated section in configuration file that deals with all the things needed to set up hpfeeds sharing. It can be primarily used for hpfeeds sharing only, by disabling local database storage.

## 1.4 Other Spam Honeypots

Open relay honeypots include Jackpot, written in Java by Jack Cleaver; smtpot.py, written in Python by Karl A. Krueger; and spamhole, written in C. The Bubblegum Proxypot is an open source honeypot (or "proxypot").

## 1.5 Authors


- **Sumit Sharma** <sumit.iips@gmail.com>
- **Rahul Binjve** <rahulbinjve@gmail.com>

# 2. Setting up SHIVA

Installing SHIVA is pretty straightforward and simple. It is as easy as running a single bash script, given you've required permissions and prerequisites installed.

## 2.1 Prerequisites

We expect following packages to be installed in system before you start the installation procedure.

- Python2.7
- exim4-daemon-light
- g++
- python-virtualenv
- python-dev
- libmysqlclient-dev
- mysql-client (optional, only if you want to save spam data in databases.)
- mysql-server (optional, only if you want to set up MySQL database in same machine.)
- phpmyadmin (optional, only if you want to monitor your databases, the GUI way.)

## 2.2 Installation

### 2.2.1 Preparing Your System

If you've a freshly Ubuntu system, you can install all the required packages using this simple command (assuming Python is pre-installed).

```
$ sudo apt-get install python-dev exim4-daemon-light g++ python-virtualenv libmysqlclient-dev
```

### 2.2.2 Getting SHIVA (the Github way)

Clone the SHIVA's Github repository in the folder where you want to install.

```
$  git clone 10.91.1.70/RahulB/shiva.git shiva-installer
```

You can confirm that repository has been cloned properly by checking the content of *"shiva-installer"* folder.

```
$ cd shiva-installer

$ ls
```

### 2.2.3 Installing

- The installer can be run as:

```
$ ./install.sh
```

- After dependencies have been checked, you'll be asked to confirm if you want to setup local databases or not.

```
Do you wish to setup local databases for storing spam?

[Y]es/ [N]o...
```

If the user inputs "Yes", appropriate instructions will be printed on console to set up databases.

Installation will proceed further, then. If everything goes well, everything will be installed in *"shiva"* folder.

### 2.2.4 Setting up database

If you want to store spam data, you'll need to setup two databases. This section deals with setting up MySQL databases.

**Note:** The following steps assume that you're setting up MySQL server on the same machine. However, this practice is discouraged.

Before setting up databases, run the following command to install the dependencies:

```
$ sudo apt-get install mysql-server mysql-client
```

After the above packages have been installed, follow these steps to setup databases:

- Edit the configuration file located at *"shiva/shiva.conf"* and provide the necessary parameters required under the *"database"* section.
- Run the python script that'll creates necessary databases and tables.

```
$ python shiva/dbcreate.py
```

- If script exits without any error, databases have been created.

### 2.2.5 Setting up exim

For relay, exim is required.

Since, by default exim starts listening on port 25, we need to configure it to listen on port 2500. You can simply configure exim by running the bash script provided. Follow the steps below:

```
$ cd shiva

$ sudo sh setup_exim.sh
```

This will configure exim and restart it.

**Note**: You should have "*exim4-daemon-light*" package installed.

# 3. Configuration

Configuring SHIVA is a piece of cake, credits to ConfigParser, a Python library. Everything that needs to be configured, is in "shiva/shiva.conf" file. The file is divided into various sections, namely:

- global
- receiver
- analyzer
- database
- hpfeeds

Below description has been provided about each option.

## 3.1 Global

Configuration(s) that applies for both, analyzer and receiver.

- queuepath - Path where all the spam will be dumped and retrieved for analyzing. This will be updated by installer script itself.

**Note**: Even though the 'queuepath' can be changed, currently it is recommended to use default settings only.

## 3.2 Receiver

SHIVA receiver's configuration.

- listenhost - The host where to start SMTP receiver on, usually your public IP. "localhost", by default.
- listenport - The port to listen on for incoming SMTP connections. 25, by default.
- sensorname - Name of the sensor. "shiva", by default.

### 3.2.1 SMTP AUTH

If you want SMTP AUTH in SHIVA, edit these options.

- authenabled - Boolean value to enable/disable SMTP AUTH. "False", by default.
- smtpuser - Username for SMTP AUTH.
- smtppasswd : Password for SMTP AUTH.

## 3.3 Analyzer

All the options for SHIVA analyzer.

- relay : enable/disable relaying of spam.
- globalcounter : Number of total spam to be relayed, in a specific duration.
- individualcounter : Number of times an indivisual spam is to be relayed, in a specific duration.
- relayport : 2500 The port on which MTA is listening.
- relayhost : The host on which MTA is listening.
- undeliverable_path : Path where distorted spam will be dumped. Will be updated by installer script.
- schedulertime : Duration (in minutes) to be passed to shivascheduler module.
- rawspampath : Path where raw spam samples will be dumped. Will be updated by installer script.
- attachpath : Path to dump attachments . Will be updated by installer script.
- inlinepath : Path to dump inline attachments. Will be updated by installer script.

## 3.4 Database

Database related configurations.

- localdb - Boolean value to enable/disable database storage.
- host - MySQL host to connect. "localhost", by default.
- user : MySQL username. "root", by default.
- password : MySQL password. "password", by default.

## 3.5 Hpfeeds

Options for enabling Hpfeeds sharing.

- enabled - Boolean value to enable/disable hpfeeds.
- host - Hpfeeds host.
- port - Hpfeeds port.
- ident - Ident for hpfeeds.
- secret - Secret key for hpfeeds.

# 4. Running SHIVA

For running SHIVA, open two terminals. One for receiver and another one for analyzer. To start SHIVA, we need to activate the respective virtual environments and starting the lamson instances.

## 4.1 Shiva Receiver

Shiva Receiver is the first half of SHIVA that starts a SMTP server on host and port that you've specified in config file. To start the receiver, follow these steps. (**Note:** Requires root rights.)

```
$ sudo su

# cd shiva/ShivaReceiver/
```

- Activate virtual environment.

```
# source bin/activate
```

- Now, the prompt should be similar to this

```
(ShivaReceiver) #
```

- Now, we'll start lamson that'll start SMTP server.

```
(ShivaReceiver) # cd Receiver/

(ShivaReceiver) # lamson start
```

To check if lamson started correctly, we can either check the log files or by using *'netstat'* Linux command.

- Checking logs

```
(ShivaReceiver) $ cd logs/

(ShivaReceiver) $ head lamson.log
```

This will output something like "SMTP Receiver successfully started."

Or, the 'netstat' command.

```
(ShivaReceiver) $ sudo netstat -natp | grep 25
```

This command should show *"python"* listening on port 25.

## 4.2 Shiva Analyzer

Shiva Analyzer is the part of SHIVA where all the analyzing work is done. Analyzer starts lamson and waits for spam to arrive in "queue" folder. Queue's path is defined in "shiva.conf" similar to this

```
[global]

queuepath : /path/to/shiva/queue/
```

- To start the receiver, follow these steps.

```
$ cd shiva/ShivaAnalyzer/
```

- Activate virtual environment.

```
$ source bin/activate
```

- Now, the prompt should be similar to this

```
(ShivaAnalyzer) $
```

- Now, we'll start lamson that'll start queue receiver.

```
(ShivaReceiver) $ cd Analyzer/

(ShivaReceiver) $ lamson start
```

To check if lamson started correctly, we can check the log files.

- Checking logs

```
(ShivaReceiver) $ cd logs/

(ShivaReceiver) $ head lamson.log
```

This will output something like "Queue receiver started on queue dir /path/to/shiva/queue/."

## 4.3 Checking Logs

Logs for receiver can be found at

```
$ cd /path/to/shiva/ShivaReceiver/Receiver/logs/

$ ls

lamson.err  lamson.log  lamson.out  clearlogs.sh
```

- lamson.err - All the errors and relay logs are stored in this file.
- lamson.log - All the logs generated by SMTP receiver.
- lamson.out - Anything supposed to be printed on console by program is written in this file.
- clearlogs.sh - Bash files that'll delete the log files.

Similarly for analyzer, logs are at

```
$ cd /path/to/shiva/ShivaReceiver/Receiver/logs/

$ ls

lamson.err  lamson.log  lamson.out  clearlogs.sh
```

- lamson.err - All the errors and relay logs are stored in this file.
- lamson.log - All the logs generated while analyzing spam.
- lamson.out - Anything supposed to be printed on console by program is written in this file.
- clearlogs.sh - Bash files that'll delete the log files.

# 5. General Problems and Precautions

Coming soon.

# 6. Frequently Asked Questions

- *"SHIVA doesn't work or keeps on breaking while analyzing spam"?*

SHIVA is in infant stage, therefore, we believe that this is a normal scenario. We're constantly working and trying to make it more stable and user-friendly. If you cannot make SHIVA run, please read this documentation carefully. If that doesn't help, refer the log files and google if you see any error.

Even then if you can't make SHIVA work, blame it on the complexity of setting up a honeypot. *wink*

- *Deactivating virtual environment?*

Python's virtual environments can be simply deactivated by

```
$ deactivate
```

- *There's unusual amount of storage used by SHIVA, why?*

SHIAV produces lots of logs. There are receiver logs, analyzing logs, hpfeeds logs, relay logs, and what not. If you see high storage usage, you can go to logs folder and run the '*clearlogs.sh*' script. Log files path information can be found here.

- *How to confirm if both receiver and analyzer are running?*

- Since, receiver and analyzer are two different parts, if everything's working fine, there must be two '*Lamson'* instances running on your system. This can be confirmed by using *'ps',* another great command.

```
$ ps -el | grep -i lamson
```

This command will show the Lamson processes that are running in system. If this shows 2 lamson instances running, you're good to go, most probably.

- *What are the recommended system specification?*

- These are the recommended system specifications:

    - Processor - Intel Dual Core or higher
    - OS - Debian-based. Tested on Ubuntu 12.04/13.04 and Mint Linux 15.
    - RAM - 512 Mb or more
    - Hard disk - 5 Gb or more.
    - Graphic card - Meh, kidding?

- *I found a bug. Now?*

- If you found a bug, report it to us. For contact info, check authors' info. If you've patch for it, that's awesome! If you haven't, go find them. There's lots of 'em.

- *Adding support for Virus Total/ Cuckoo sandbox API?*

Due to the simple design of SHIVA, a small module could be easily written, to send the attachments and URLs received in a spam to Virus Total or Cuckoo sandbox for further analysis.

- *My IP has been blocked by Google/ Yahoo!, Hotmail, etc. What do I do now?*

- You got your IP blocked? Great! That happens more often, than not. If your IP is blocked, there's nothing that we could do, actually. However, we'll suggest to set relay counters to 0 (i.e. stop relaying) for some days and wait for your IP to get unblocked (if it does). And when you restart relaying, keep the relay counters low.

- How do I contribute?

- The project is in infant stage, so we need contribution from community. Actually, loads of contribution. If you've a bug, report it to us, For contact info, check authors' info. If you've a patch or you've added some cool feature, clone our github repository, send us a pull request. We'll be more than happy to merge your patches.

Please note that while submitting patches, make sure your patch follows PEP 8 -- Style Guide for Python Code.

# 7. References

Following documents were referred while preparing this document.


- https://write-the-docs.readthedocs.org/en/2013/writing/oss.html
- http://www.honeynet.org/files/KYT-Glastopf-Final_v1.pdf
- http://docs.cuckoosandbox.org/en/latest/