HONEYPY INSTALLATION

* Command to download HoneyPy is:

wget <https://github.com/foospidy/HoneyPy/archive/0.4.8.tar.gz>

* Unzip the file with the command tar -xzf 0.4.8.tar.gz, this will create the folder HoneyPy-0.4.8
* On Debian run the following commands as root (or using [sudo](https://en.wikipedia.org/wiki/Sudo)) too ensure we have both python and pip installed :

apt-get install -y python-requests python-twisted python-pip pip install twitter dnslib

RUNNING HONEYPY

* run HoneyPy with its “out-of-the-box” configuration:

python Honey.py

* At the HoneyPy console prompt type **start** and hit enter

[image]

HoneyPy HoneyPot is now successfully running !!

* Running HoneyPy in console mode may cause your terminal session to terminate. When terminal session is compromised HoneyPy also gets terminated. To prevent this we have two methods:

1. Use terminal utility screen:

To install screen on Debian run apt-get install screen.

1. Run HoneyPy in daemon mode using:

python Honey.py –d

* Running HoneyPy services on low ports (ports below 1024).

[port wali image]

This is telling us that we have HoneyPy configured to listen on low ports and should be run as a non-root user. But a process needs root user privilege to open and listen on low ports.

We need a way for HoneyPy to run services like telnet, ftp, ssh, etc. to receive connections on these low ports. This can be done using **iptables** to enable port forwarding on a low port to a high port that HoneyPy can use.

Iptables are complex to work on and can be learnt with the help of ipt-kit.

**How to use ipt-kit to configure iptables for HoneyPy :**

**Note**: modifying iptables does require root user privilege so you will need to run the ipt-kit commands as root.

1. HoneyPy can generate a script file for us. If you are still in the HonePy console type quit to exit.

Next run:

*python Honey.py -ipt*

This will generate a the file **/tmp/honeypy-ipt.sh**

1. Download ipt-kit from <https://github.com/foospidy/ipt-kit/archive/v1.1.tar.gz>.
2. Extract ipt-kit to a directory of your choosing, **tar -xzf v1.1.tar.gz.**
3. Change directory to **ipt-kit-v1.1** and copy honeypy-ipt.sh to the current directory **cp /tmp/honeypy-ipt.sh.**
4. Run the script **./honey-ipt.sh** (this needs to be run wirth root privilege, so either run this as the root user or use sudo)
5. run: **./ipt-survive-reboot.sh**

To ensure the persistence of updated iptables.

**Configuring HoneyPy**

Configuration for HoneyPy itself is very simple and straight forward. The configuration file is in <your HoneyPy directory>/etc/honeypy.cfg

You can actually run HoneyPy without changing this file, but by default, events are only logged to the default log file. The default log file is not the nicest file to read, which is why HoneyPy has more useful “loggers” that can be enabled.

The current set of loggers as of this blog post are:

* Twitter — Tweet events on Twitter
* HoneyDB — Post events to HoneyDB (future blog post on HoneyDB to come).
* Slack — Post events to a Slack channel.
* Log stash — Post events to log stash.
* Elastic search — Post events directly to Elasticsearch

To configure these loggers simply change the enabled configuration option to **Yes** and restart HoneyPy.

Configuring HoneyPy Services

HoneyPy’s services configuration file is located at **<your HoneyPy directory>/etc/services.cfg**, and this is the file that tells HoneyPy what services to run, and what ports to run them on.

In the **services.cfg** file you’ll see several entries that look like the example below. Each entry represents a service that HoneyPy will launch when started.

[screenshot]

* **[Echo]** - specifies the name of the service. The name can be anything you want and will be what is displayed in downstream loggers, e.g. Twitter, HoneyDB, Elasticsearch, etc. Note, the service name must not contain spaces.
* **Plugin -** tells HoneyPy which plugin to load and use for service emulation. All available plugins live in the plugins directory and they are what emulates a service, e.g. Telnet, FTP, SSH, etc.
* **low\_port -** if the service is going to listen on a low port specify that port number here. Otherwise, the value should be the same as the next line, port. Note this also specifies the protocol for the service, tcp or udp. The format to use is <protocol>:<port number>
* **port**- this is the port HoneyPy will actually listen on. Note, this also specifies the protocol for the service, tcp or udp. The format to use is <protocol>:<port number>
* **description** -is a short description of what this service does.
* **Enabled-**can have the value Yes or No, and this tells HoneyPy whether this service should be enabled or disabled at startup.