INSTALLATION GUIDE

El Jefe 2.1

Document version: June 2014

Contents

1 Goal of this Document

This document enables you to realize the glory of a full installation of El Jefe 2.1 on various operating systems. You will need to be familiar with administering Linux in order to properly use this document. *Should you require any troubleshooting at anytime during the El Jefe installation please email us at eljefe@immunityinc.com.*

2 Introduction

The first step is to download the last version of El Jefe from Immunity's website: *https://eljefe.immunityinc.com/eljefe/*. The file comes in a tar/gzip format. You should also take the time now to verify the SHA-1 hash with the release email, which is likely archived in many places from being sent to our company mailing list, DailyDave. If you're not subscribed to DailyDave, then you should take the time to subscribe now.

Inside the downloaded package you will find the following components:

- Client: El Jefe client's source code
- Cuckoo: Cuckoo Sandbox v1.0 for automated malware analysis.
- Dependencies : All dependencies needed for the client's installer development.
- Docs: El Jefe documentation.
- Installer: All files needed to create an executable installer for the El Jefe client.
- Webapp: El Jefe server's source code.

Illustration 1: El Jefe root's folders.

3 Installation

3.1 El Jefe 2.1 Installation on Ubuntu 13.04 LTS

El Jefe comes with its own shellscript installer inside the webapp folder (webbapp/install_ubuntu.sh) for Ubuntu. The script was created and tested on Ubuntu 13.04 LTS (we highly recommend you to use that version). Run the script to install the El Jefe server. Root privileges are required. The script will install all the dependencies needed for El Jefe 2.1 and configure El Jefe 's database.

```
anibal@imm: ~/eljefe-2.1/webapp
anibal@imm:~/eljefe-2.1/webapp$ sudo ./install_ubuntu.sh
[sudo] password for anibal:
Installing El Jefe dependencies ...
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version.
python-chardet is already the newest version.
python-dev is already the newest version.
The following extra packages will be installed:
  libboost-dev libboost-filesystem1.49.0 libboost-program-options1.49.0
  libboost-system1.49.0 libboost-thread1.49.0 libboost1.49-dev
  libgoogle-perftools4 libpq5 libsnappy1 libtcmalloc-minimal4 libunwind8
  mongodb-clients mongodb-dev mongodb-server postgresql-9.1
  postgresql-client-9.1 postgresql-client-common postgresql-common
  python-bson-ext python-central python-egenix-mxdatetime
  python-egenix-mxtools python-setuptools python-sqlalchemy-ext python-support
Suggested packages:
  libboost-doc libboost1.49-doc libboost-chrono1.49-dev
  libboost-date-time1.49-dev libboost-filesystem1.49-dev
  libboost-graph-parallel1.49-dev libboost-graph1.49-dev
  libboost-iostreams1.49-dev libboost-locale1.49-dev libboost-math1.49-dev
  libboost-mpi1.49-dev libboost-program-options1.49-dev
  libboost-python1.49-dev libboost-random1.49-dev libboost-regex1.49-dev
  libboost-serialization1.49-dev libboost-signals1.49-dev
  libboost-system1.49-dev libboost-test1.49-dev libboost-thread1.49-dev
  libboost-timer1.49-dev libboost-wave1.49-dev doxygen default-jdk fop oidentd
  ident-server locales-all postgresql-doc-9.1 python-egenix-mxdatetime-dbg
```

Illustration 2: Running the installation script

A list of dependencies installed by the El Jefe installation script include the following:

- build-essential
- python-dev
- python-pip
- python-psycopg2

- postgresql
- mongodb
- django
- django-taggit
- django-bootstrap3
- requests
- numpy
- pymongo

Before finishing the installation you must insert the username and password needed for the first login to El Jefe's WebUI. You will want to use a strong password here as some of the information may be considered sensitive and El Jefe also has the ability to download executables from the remote machines.


```
Creating tables ...
Creating table auth_permission
Creating table auth_group_permissions
Creating table auth_group
Creating table auth_user_groups
Creating table auth_user_user_permissions
Creating table auth_user
Creating table django_content_type
Creating table django_session
Creating table django_site
Creating table django_admin_log
Creating table django_comments
Creating table django_comment_flags
Creating table home_xmlusers
Creating table home_stations
Creating table home_binaries
Creating table home_privileges
Creating table home_events_privileges
Creating table home_events
Creating table home_unique_executions
Creating table home_binary_requests
Creating table taggit_tag
Creating table taggit_taggeditem
Creating table alerts_action
Creating table alerts_executionfilter_actions
Creating table alerts_executionfilter
Creating table alerts_testfilter_actions
Creating table alerts_testfilter

You just installed Django's auth system, which means you don't have any superusers defined.
Would you like to create one now? (yes/no):
```

Illustration 3: Installation script will request to create an account

```
You just installed Django's auth system, which means you don't have any superusers defined.
Would you like to create one now? (yes/no): yes
Username (leave blank to use 'root'): anibal
Email address: admin@immunityinc.com
Password:
Password (again):
Superuser created successfully.
Installing custom SQL ...
Installing indexes ...
Installed 0 object(s) from 0 fixture(s)
All done.
anibal@imm:~/eljefe-2.1/webapp$ █
```

Illustration 4: Installation Complete

3.2 El Jefe 2.1 Installation on Red Hat Enterprise 7

Aside from the Ubuntu Installer, El Jefe comes with a second installer shellscript for Red Hat Enterprise 7 in the webapp folder *webbapp/install_rhel.sh*. Run the script to install the El Jefe server. Root privileges are required.

Illustration 5: Installing El Jefe using RHEL shellscript

The script will require a username and password for the El Jefe's WebUI login.

Illustration 6: Installation Complete

3.3 Cuckoo Installation

Cuckoo is an open source sandbox that automates malware analysis. Version 2.1 of El Jefe includes Cuckoo to help analysts by providing further in-depth information about potentially harmful processes. ***The installation of Cuckoo is optional***, but we think you will find it quite useful once completed. Keep in mind that Cuckoo **cannot be installed from WITHIN a virtual machine**, unlike El Jefe itself.

The reason Cuckoo is so useful is that El Jefe will download a suspicious binary from any event upon the analyst's request. Then El Jefe will task Cuckoo with running that binary in a virtual machine. Cuckoo will then produce a complete report with the results and these results are then made available through the El Jefe WebUI.

The Cuckoo Sandbox is already inside El Jefe version 2.1 package release so there is no need for extra downloads (Check out the cuckoo folder for more information).

Cuckoo supports VirtualBox, KVM and VMWare for virtualization. We are using VMWare Workstation 9.0 in this guide as our virtualization software, but it should be rather trivial to install it in others. You can follow the guidelines in the following URL for assistance with other virtualization software (*http://docs.cuckoosandbox.org/en/latest/installation/guest/*).

4 Configuring the Virtual Machine

It is time to setup the virtual machine where we are going to run and analyze our binaries. First, we will need to create a new virtual machine. The preferred operating system by Cuckoo is Windows XP, but other Windows versions are also supported. We are using Windows 7 x64 in this guide.

We need to install Python and Python Imaging Library (PIL) (*http://www.pythonware.com/products/pil/*). If you are using Windows x64 you need to install a 32 bits Python version to support PIL.
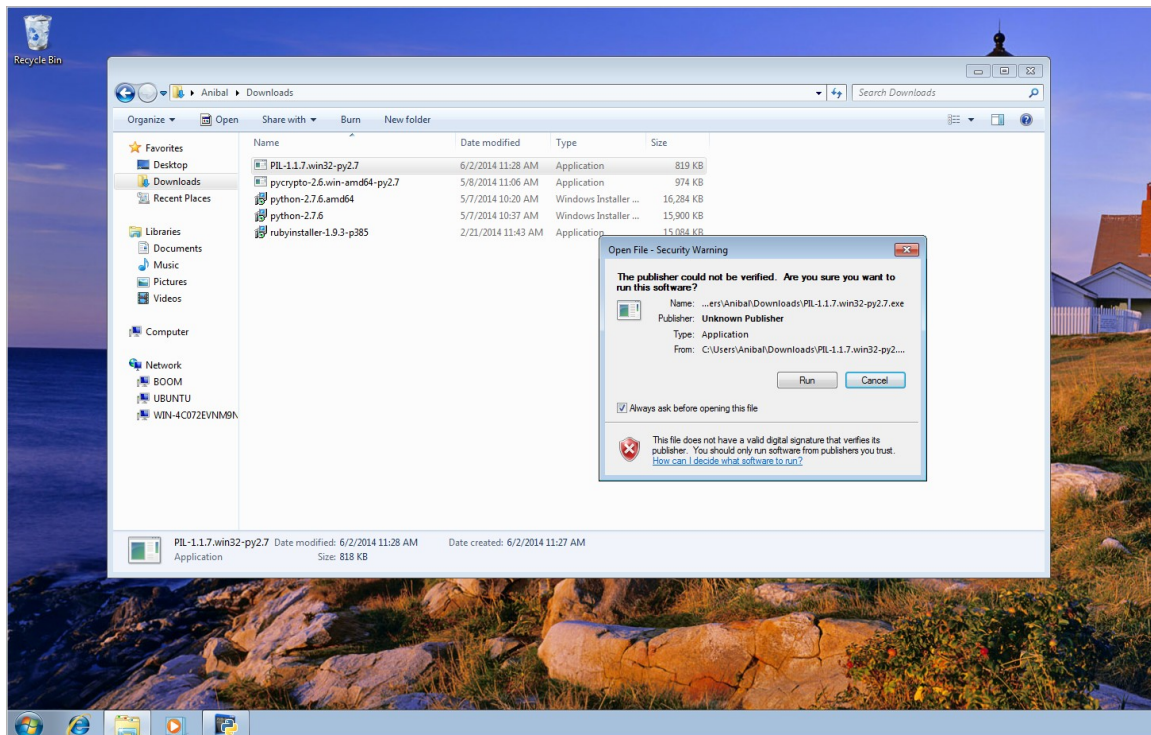

Illustration 7: Installing Python and PIL on the guest virtual machine

5 Network Setup

It is time to configure the virtual machine's networking. First, disable the Windows Firewall after that create a virtual host-only network for the guest and the host.

Illustration 8: Turning the Windows Firewall off on the guest virtual machine

To do this you will need to open the Virtual Network Editor and create a virtual host-only network for the guest and the host to communicate.
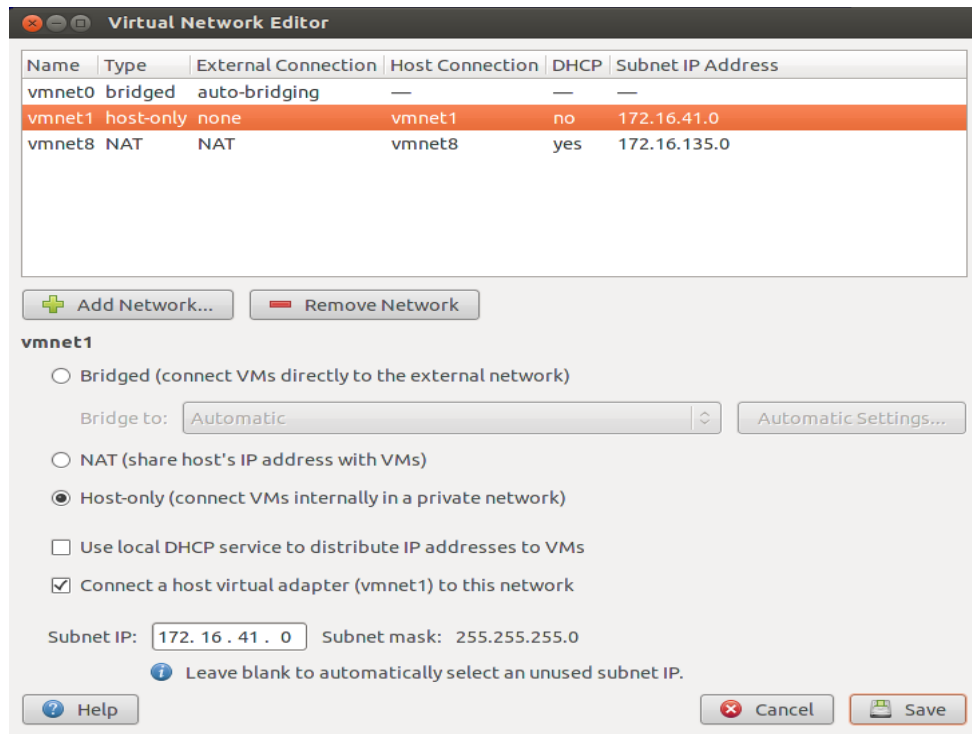
Illustration 9: Configuring Virtual Network Editor

We are using the 172.16.41.0 network which makes the guest configuration as follows:

> IP Address: 172.16.41.159
> Netmask: 255.255.255.0
> Gateway: 172.16.41.1

Cuckoo supports DNS resolution so you can choose 172.16.41.1 as your DNS server. In our experience, we get better analysis results by using a public DNS server (for example the Google DNS: 8.8.8.8 and 8.8.4.4).

The host configuration (for the vmnet adapter) is as follows:

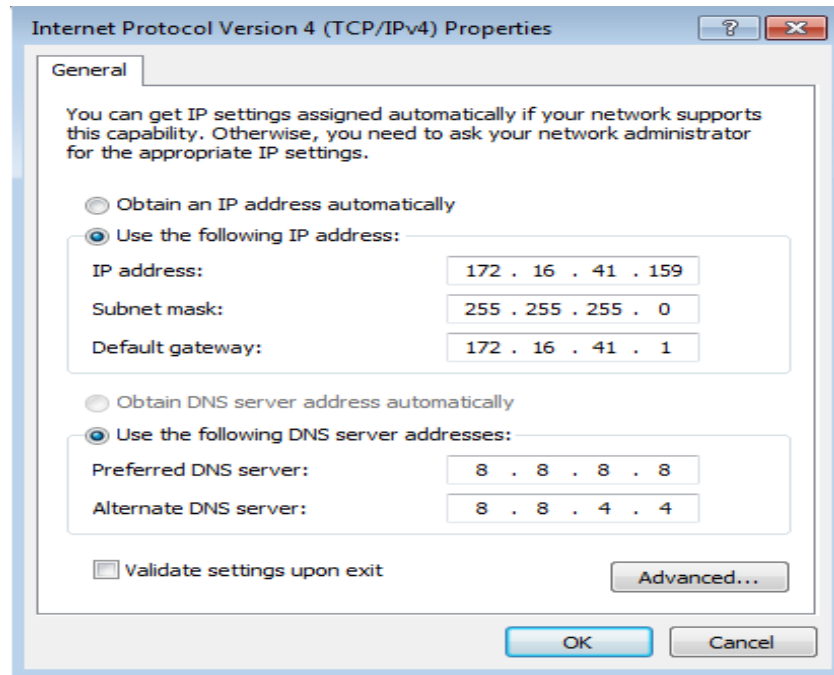> IP Address: 172.16.41.1
> Netmask: 255.255.255.0

Illustration 10: Setting-up the network on the guest virtual machine

5.1 Configuration of Networking

The next step is to configure packet forwarding on the host machine to give the guest machine Internet access. We can do this with the following commands (replacing eth0 with your outgoing interface and vboxnet0 with your virtual interface):

```
$ iptables -A FORWARD -o eth0 -i vmnet1 -s 172.16.41.0/24 -m conntrack --ctstate
NEW -j ACCEPT
$ iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
$ iptables -A POSTROUTING -t nat -j MASQUERADE
$ sysctl -w net.ipv4.ip_forward=1
```

This concludes the networking setup.

6 Agent Installation

We need to install the Cuckoo agent on the guest by copying *agent.py (*located in *cuckoo/agent* folder) to the Windows Startup folder.
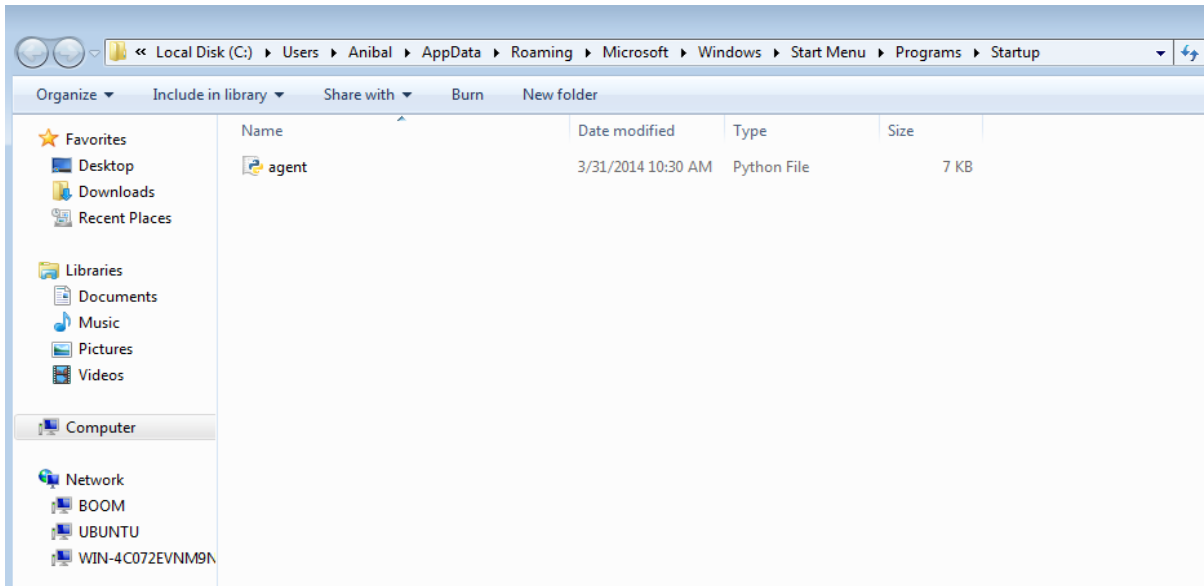

Illustration 11: Coping the agent on the startup folder

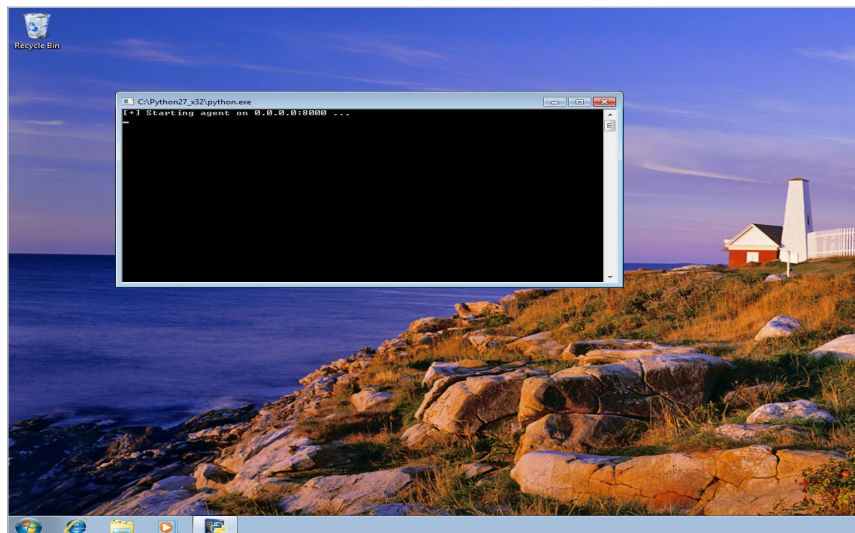The next step is reboot the virtual machine and the agent will be started!

Illustration 12: MISSING CAPTION

Now it is time to make a snapshot. Before executing a malware binary the snapshot is reverted and then the binary is executed and analyzed.
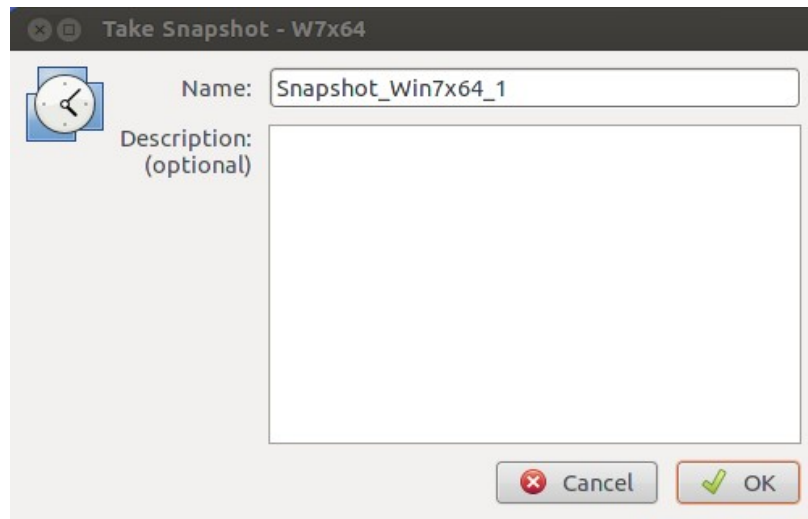

Illustration 13: Taking a snapshot of the guest virtual machine

Summary:
vm path = $HOME/vmware/W7x64/W7x64.vmx
vm snapshot = Snapshot_Win7x64_1
vm IP = 172.16.41.159

7 Final Configuration

If you opted to install Cuckoo, you will need to configure the files in Cuckoo which is running on our El Jefe server. First, modify the *cuckoo/conf/auxiliary.conf* and change the interface to the one that you will use to communicate with the guest virtual machine.

```
 1 [sniffer]
 2 # Enable or disable the use of an external sniffer (tcpdump) [yes/no].
 3 enabled = yes
 4
 5 # Specify the path to your local installation of tcpdump. Make sure this
 6 # path is correct.
 7 tcpdump = /usr/sbin/tcpdump
 8
 9 # Specify the network interface name on which tcpdump should monitor the
10 # traffic. Make sure the interface is active.
11 interface = vmnet1
12
13 # Specify a Berkeley packet filter to pass to tcpdump.
14 # bpf = not arp
```

Illustration 14: Configure the interface to be used

Next, modify *cuckoo/conf/cuckoo.conf* to set the result server IP address on the [resultserver] section (in our case it is the El Jefe Server (172.16.41.1)).

Illustration 15: Set up the result server IP

Finally, modify *cuckoo/conf/vmware.conf*. If you are planning to use just one virtual machine then you will just need to modify everything inside [cuckoo1]. Otherwise, you will have to create an additional section for each extra virtual machine used. Inside these sections you will have to set the virtual machine path, snapshot name and IP address of the machine (in our example it is 172.16.41.159).

Illustration 16: Setting-up the last configuration details

We are done setting Cuckoo for El Jefe!

![Immunity logo]

8 Optional Settings (recommended)

When Cuckoo is analyzing submitted files there are some extra modules and libraries that will improve the analysis which will provide the analyst with a better report. **These modules are optional but their installation is highly recommended.**

Run setcap to give tcpdump root privileges without needing to have Cuckoo run as root.

```
$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
```

9 Starting El Jefe

Now you are ready to start up El Jefe. We recommend running these commands inside a screen session.

First, run the web interface. Inside *webapp/* you must type :

```
$ python manage.py runserver
```

```
anibal@imm:~/eljefe-2.1/webapp$ python manage.py runserver
Validating models...

0 errors found
June 23, 2014 - 09:34:06
Django version 1.6.5, using settings 'webapp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Illustration 17: Running the WebUI

The next step is start the XML Server. You must set the IP address of El Jefe Server in webapp/xmlserver/settings.py

Immunity Inc. 1130 Washington Ave. 8th Floor Miami Beach, FL 33139
(p) 786-220-0600 (f) 786-513-8100 (web) www.immunityinc.com

```
#settings for xml server
LOGHOST = "192.168.1.127"
LOGPORT = 5555

#logging options
LOGGING_FILENAME="ElJefeXMLServer.log"
LOGGING_LEVEL="INFO"

#be sure to use your own certs
KEYFILE='certs/server.key'
CERTFILE='certs/server.pem'
CA_CERTFILE='certs/cacert.pem'

# Email alert settings
ENABLE_EMAIL_ALERTS = False
EMAIL_ACCOUNT = "youraccount@domainthatshouldreallybeset.com"
SMTP_SERVER = "127.0.0.1"
SMTP_PORT = 25
~
~
~
~
```

Illustration 18: Setting XML Server IP address

Now you can start the XML Server. In *webapp/xmlserver* you must type:

```
$ python ElJefeXMLServer.py
```

```
anibal@imm:~/eljefe-2.1/webapp/xmlserver$ python ElJefeXMLServer.py
[*] Starting server....
[*] Serving HTTPS on 192.168.1.127 port 5555
```

Illustration 19: Running the XML Server

Finally, run the Cuckoo sandbox server. Inside cuckoo/ run:

```
$ python cuckoo.py
```


Ilustration 20: Running Cuckoo

You should now be able to log into El Jefe using the username and password created during the installation by going to *http://ElJefeIP:8000/*.
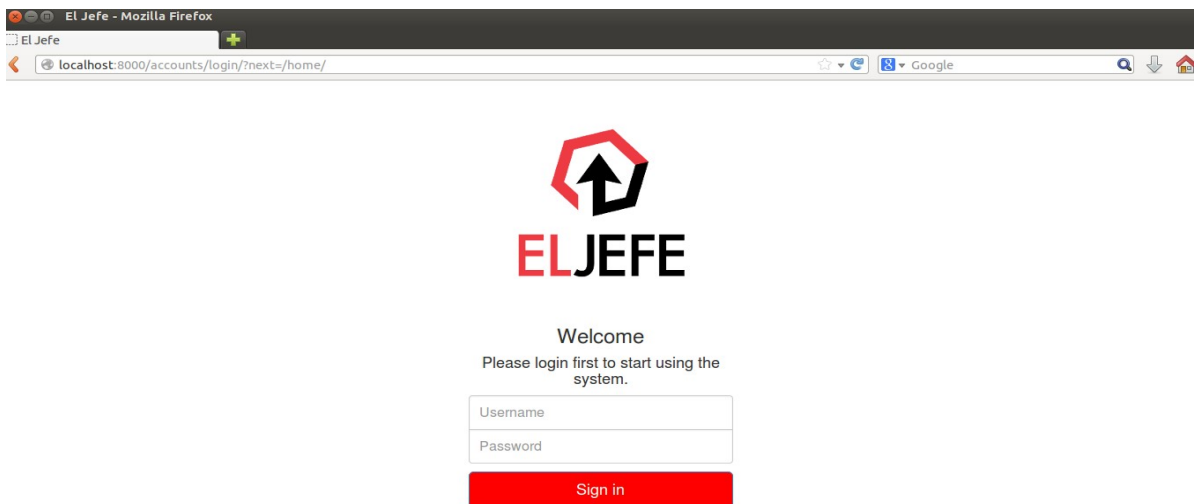

Illustration 21: Log-in into El Jefe UI

Now that you are logged in it is now time to catch some threats. Enjoy the ride!