

# ALIEN Control Framework installation and configuration

The procedure described in this document has been tested on a Debian 6 Linux OS (both 32 and 64bit releases) with kernel version 2.6.32-5. However, it may also work for other Linux distributions.

Chapter 1 of the document describes the installation process of the ALIEN Control Framework ([Expedient](#), [TB Plugin](#), [TBAM](#) and [OFGW](#)) in a single machine (physical or virtual). As Expedient and AMsoil require two different versions of Python (2.6 and 2.7 respectively), this Chapter includes a section that explains how to achieve the coexistence of both Python 2.6 and Python 2.7 on the same machine.

Alternatively, the ALIEN Control Framework can be tested through a pre-configured VirtualBox Virtual Machine (see <https://wiki.man.poznan.pl/alien/index.php/T4.3#Software>) where Expedient, the TB Plugin, the TBAM and the OFGW are already installed. The VM is accessible with root permissions (the password is *openflow*) from the VirtualBox console or via SSH protocol.

Chapter 2 provides configuration and startup details for each single component. This is the only Chapter needed in case of using the Virtual Machine.

## Table of Contents

Table of Contents .....	1
1 Installation .....	2
1.1 Expedient and TBPlugin .....	2
1.2 Python 2.6 and 2.7 coexistence .....	3
1.3 TBAM with AMsoil framework .....	4
1.4 OFGW .....	4
2 Configuration and startup .....	7
2.1 TBAM with AMsoil framework .....	7
2.2 OFGW .....	8

# 1 Installation

## 1.1 Expedient and TBPlugin

The following operations require root access to the Linux Operating system:

- Install MySQL server:

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

- Create Expedient database:

```
$ mysql -p
$ mysql> CREATE DATABASE expedient;
$ mysql> GRANT ALL ON expedient.* TO user@127.0.0.1 IDENTIFIED BY
password;
```

- Clone the OCF repository under folder /opt:

```
$ cd /opt
$ git clone https://github.com/fp7-ofelia/ocf.git ofelia
```

- In the file  
/opt/ofelia/expedient/bin/versions/default/install/lib/pypelib change  
the line:

```
$ /usr/bin/wget https://github.com/fp7-
ofelia/pypelib/raw/deb/pypelib_latest_all.deb || error "Could not
download pypel$
```

into:

```
$ /usr/bin/wget --no-check-certificate https://github.com/fp7-
ofelia/pypelib/raw/deb/pypelib_latest_all.deb || error "Could not
download pypel
```

- Trigger OFVER installation by performing the following as a root user:

```
$ cd /opt/ofelia/expedient/bin
$ ./ofver install
```

- When installation starts, ofver will ask if it is an OFELIA project installation or not. Select Yes (Y) for OFELIA testbeds
- You will need to create the certificates for the Certification Authority (CA) first and for the component (i.e. Expedient) later. Do not use the same Common Name (CN) for both of them, and make sure that the CN you use in the component later certificate (you can use an IP) is the same you then set in the SITE\_DOMAIN field in the localsettings.py file
- Modify the localsettings.py or mySettings.py depending on the component (i.e. Expedient) being installed:

```
ROOT_USERNAME = "user"
ROOT_PASSWORD = "pass"
DATABASE_NAME = "expedient"
DATABASE_USER = "user"
DATABASE_PASSWORD = "password"
```

```

SITE_DOMAIN = "localhost:1234"
LDAP configuration (only needed in case of connection with
the OFELIA central LDAP server):
ENABLE_LDAP_BACKEND = True
LDAP_STORE_PROJECTS = True
ALLOW_LOCAL_REGISTRATION=False
AUTH_LDAP_BIND_PASSWORD = YOUR_LDAP_PASSWORD
AUTH_LDAP_BIND_DN = YOUR_LDAP_BIND_DN

```

- Run:  

```

$ ./opt/ofelia/expedient/src/python/expedient/clearinghouse/manage.py
create_default_root

```
- Clone the plugin folder:  

```

$ git clone https://github.com/fp7-ALIEN/OCF-TBPlugin.git TB-plugin

```
- Copy the folder TB-plugin/ALIEN\_plugin under  
/opt/ofelia/expedient/src/python/plugins/
- Copy the sa and ma cert files from the folder ALIEN\_plugin/certs to the folder  
deploy/trusted of TBAM
- Synchronize database:  

```

$ cd /opt/ofelia/expedient/src/python/expedient/clearinghouse
$ python manage.py syncdb

```
- Restart Apache  

```

service apache2 restart

```

## 1.2 Python 2.6 and 2.7 coexistence

- Packages to be installed and removed:  

```

apt-get install build-essential libsqlite3-dev zlib1g-dev
libncurses5-dev libgdbm-dev libbz2-dev libreadline5-dev libssl-dev
libdb-dev libffi-dev libxml2-dev libxslt-dev libxmlsec1 xmlsec1
libxmlsec1-openssl libxmlsec1-dev
apt-get autoremove pip

```
- Get and install a stable version of Python2.7:  

```

$ wget http://www.python.org/ftp/python/2.7.3/Python-2.7.3.tgz --no-
check-certificate
$ tar -xzf Python-2.7.3.tgz && cd Python-2.7.3
$ ./configure --prefix=/usr --enable-shared
$ make
$ sudo make install
$ sudo update-alternatives --install /usr/bin/python python
/usr/bin/python2.6 20

```

```
$ sudo update-alternatives --install /usr/bin/python python
/usr/bin/python2.7 10
& sudo update-alternatives --set python /usr/bin/python2.7
```

- Install pip:

```
$ wget https://bootstrap.pypa.io/get-pip.py
& python2.7 get-pip.py
```

## 1.3 TBAM with AMsoil framework

- Install AMsoil dependencies:

```
$ sudo pip install sqlalchemy Flask-XML-RPC flup pyopenssl python-
dateutil pytz lxml six blinker M2Crypto Flask Flask-XML-RPC Jinja2
M2Crypto MarkupSafe SQLAlchemy Werkzeug blinker cffi cryptography
flup itsdangerous lxml pika pyOpenSSL pycparser python-dateutil pytz
six wsgiref web.py
```

- Restore default Python version for Expedient compatibility:

```
$ sudo update-alternatives --set python /usr/bin/python2.6
```

- Clone the TBAM with the AMsoil framework

```
$ git clone https://github.com/fp7-alien/OCF-TBAM
```

## 1.4 OFGW

- LDAP client configuration for connecting to the central LDAP:
  1. DNS bind between LDAP url and its IP address (add the line LDAP\_IP\_address LDAP\_url to/etc/hosts);
  2. Install package libpam-ldapd (e.g, apt-get install libpam-ldapd) and follow the procedure on the
  3. UI to configure the package. When required:
    - a. Insert the LDAP url (ldap://ldap.ibbt.fp7-ofelia.eu)
    - b. Set the correct DN (dc=fp7-ofelia,dc=eu)
    - c. Select not to root management
    - d. Authorize the modification of nsswitch.conf: check that passwd,group,netgroup are selected.
    - e. Add auth required pam\_access.so in /etc/pam.d/common-auth
    - f. Add session required pam\_mkhomedir.so skel=/etc/skel umask=0022 in /etc/pam.d/common-session
    - g. Create file /etc/security/access.conf (e.g., touch /etc/security/access.conf) and add this three line:

```
+ : ALL : LOCAL
```

```
+ : @proj_<project_UUID>_<project_name> : ALL
- : EXCEPT root login:ALL EXCEPT LOCAL
```

4. After the configuration, the root user is able to login through a SSH connection. Other users can authenticate only if both TBAM Agent and TBAM RM are up and running and the users are associated to the current experiment.
- Install TBAM Agent's requirements:

- Open vSwitch version higher 2.0.0:

```
$ sudo apt-get install build-essential fakeroot
$ sudo update-alternatives --set python /usr/bin/python2.7
$      wget      http://openvswitch.org/releases/openvswitch-
2.1.2.tar.gz
$ tar -zxvf openvswitch-2.1.2.tar.gz && cd openvswitch-2.1.2
$ fakeroot debian/rules binary
$ cd ../ && sudo dpkg -i openvswitch-common openvswitch-
datapath-dkms openvswitch-datapath-source openvswitch-pki
openvswitch-switch
$ sudo ovs-vsctl add-br switch -- set Bridge switch fail-
mode=secure
$ sudo ovs-ofctl add-flow switch "priority=0, actions=drop"
```

- If any of the last two commands does not work properly run the next command and re-execute the previous commands:

```
$ sudo rmmod bridge
```

- The port configuration considers two ports: one connected to OFELIA (OFELIAPort) and the other to an ALIEN device (ALIENPort). The port configuration requires these three commands:

```
ovs-vsctl add-port switch ALIENPort -- set Interface
ALIENPort ofport_request=2
ovs-ofctl mod-port switch 2 up
ovs-vsctl add-port switch OFELIAPort -- set Interface
OFELIAPort ofport_request=3
ovs-ofctl mod-port switch 3 up
```

- Install conntrack:

```
apt-get install conntrack
```

- Install iptables (version => 1.4.14):

```
apt-get autoremove iptables
apt-get install dh-autoreconf pkg-config
git clone git://git.netfilter.org/iptables.git
git checkout -b 1.4.20 remotes/origin/stable-1.4.20
./autogen.sh && ./configure && make
sudo make install
cd /sbin
ln -s /usr/local/sbin/iptables iptables
sysctl net.ipv4.ip_forward=1
```

- Install Management Plane dependencies:  
`$ sudo pip install -r PYTHON_REQUIREMENTS`
- Clone the repository:  
`$ git clone https://github.com/fp7-alien/OCF-TBAM`

## 2 Configuration and startup

### 2.1 TBAM with AMsoil framework

- Configure, if not already done, the ALIEN and OFELIA ports as mentioned in port configuration of “Install TBAM Agent’s requirements” section. If you are using the Virtual Machine that integrates the ALIEN Control Framework, ALIEN and OFELIA ports are configured as OvS virtual ports (respectively vi0 and vi1) for testing purposes. Before configuring the ALIEN and OFELIA ports, run the following commands to remove the virtual ports:

```
$ sudo ovs-vsctl del-port switch vi0
$ sudo ovs-vsctl del-port switch vi1
```

- Start TBAM:  

```
$ cd OCF-TBAM && python2.7 AMsoil/src/main.py
```
- Start TBAM worker (see <https://github.com/fp7-alien/OCF-TBAM#tbam-resource-manager>)  

```
$ cd OCF-TBAM && python2.7 AMsoil/src/main.py --worker
```
- Now should be possible to add an aggregate manager with the type “ALIEN Resource Aggregate” from the Administration tab of the Expedient (available at [https://VM\\_ip\\_address](https://VM_ip_address)). Configure the different parameters and as server address use <https://localhost:8001/> as shown in the next figure.

The screenshot shows a form for adding a new ALIEN aggregate manager. The form has the following fields and controls:

- Name:** A text input field containing "ALIEN AM Create-Net".
- Description:** A text area containing "ALIEN Aggregate Manager, Create-Net, Povo (IT)".
- Geographic Location:** A text input field containing "Trento". Above this field is a warning icon and the text: "AMs within the same island must share the exact same location."
- Sync resources?:** A checkbox that is checked.
- Server URL:** A text input field containing "https://localhost:8001/".
- Buttons:** At the bottom, there are three buttons: "Update", "Delete", and "Cancel".

- Create a new project and a new slice and add the new ALIEN aggregate: an ALIEN topology is displayed.
- The southbound interface of the TBAM Resource Manager allows the communication with the OFGW through the TBAM Agent. However, for testing purposes, this interface is

disabled by default and TBAM will use fake fixed data. The connection to the TBAM Agent can be enabled/disabled by changing the `CONN_WITH_AGENT` parameter to `True/False`. It is possible to activate the connection with the OFGW, in particular with the TBAM-Agent that need to be started as described in the next section OFGW.

```
$ cd OCF-TBAM/AMsoil/admin && python2.7 config_client.py --set  
CONN_WITH_AGENT=1
```

Restart TBAM (see Start TBAM) to update the new configuration.

- Additional note: the connection to the TBAM Agent uses other three parameters OFGW IP address (`OFGW_ADDRESS`) and port (`OFGW_PORT`) and certificates for the SecureXMLRPC (`CLIENT_KEY_PATH` and `CLIENT_CERT_PATH`) (see [the detailed installation process in the repository](#)). The certificates for the SecureXMLRPC between OFGW and TBAM are configured and installed for testing purposes.
- It is possible to see the log with the following command:  

```
$ tail -f OCF-TBAM/AMsoil/log/amsoil.log
```

## 2.2 OFGW

The OFGW is the only entry point for the OpenFlow devices. In particular, it behaves as a proxy for the control plane. For this reason, all the switches must be configured to point the **TCP port 6633** of the OFGW machine for the OpenFlow control messages.

- Start TBAM-Agent:  

```
$ cd OCF-OFGW/TBAM-Agent && python2.7 src/main.py
```
- Configure the ALIEN network topology:  

```
$ cp neighbours.yaml.example neighbours.yaml  
$ nano neighbours.yaml
```

[Additional configuration of the Management Plane are detailed in the repository.](#)
- Start the Management plane's daemon:  

```
$ cd OCF-OFGW/ofgw_mngt/ && python2.7 daemon_ofgw_mngt.py start
```
- Access the Expedient GUI ([https://VM\\_ip\\_address](https://VM_ip_address)) the topology written to the aforementioned file should be visible.