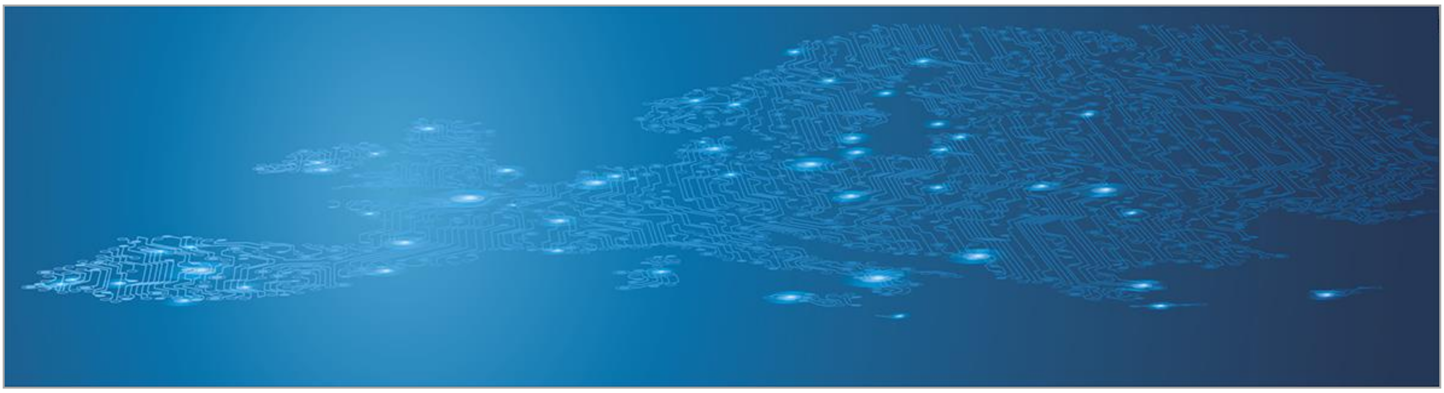


Proactive Incident Detection

Toolset, Document for students

September 2014

European Union Agency for Network and Information Security www.enisa.europa.eu

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Acknowledgements

Contributors to this report

We would like to thank all our ENISA colleagues who contributed with their input to this report and supervised its completion, especially Lauri Palkmets, Cosmin Ciobanu, Andreas Sfakianakis, Romain Bourgue, and Yonas Leguesse. We would also like to thank the team of Don Stikvoort and Michael Potter from S-CURE, The Netherlands, Mirosław Maj and Tomasz Chlebowski from ComCERT, Poland, and Mirko Wollenberg from PRESECURE Consulting, Germany, who produced the second version of this documents as consultants.

Agreements or Acknowledgements

ENISA wants to thank all institutions and persons who contributed to this document. A special 'Thank You' goes to the following contributors: Anna Felkner, Tomasz Grudzicki, Przemysław Jaroszewski, Piotr Kijewski, Mirosław Maj, Marcin Mielniczek, Elżbieta Nowicka, Cezary Rzewuski, Krzysztof Silicki, Rafał Tarłowski from NASK/CERT Polska, who produced the first version of this document as consultants and the countless people who reviewed this document.

Contact

For contacting the authors please use CERT-Relations@enisa.europa.eu

For media enquires about this paper, please use press@enisa.europa.eu.

Legal notice

Notice must be taken that this publication represents the views and interpretations of the authors and editors, unless stated otherwise. This publication should not be construed to be a legal action of ENISA or the ENISA bodies unless adopted pursuant to the Regulation (EU) No 526/2013. This publication does not necessarily represent state-of the-art and ENISA may update it from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources including external websites referenced in this publication.

This publication is intended for information purposes only. It must be accessible free of charge. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

Copyright Notice

Table of Contents

1 What Will You Learn

In this exercise you will learn how to set up and work with AbuseHelper. AbuseHelper is an Open Source Software designed to help CERT/CSIRT organization with consolidating and using information feeds.

- You will learn how to install and configure all parts of the AbuseHelper toolset
- You will learn how to take the application set into production
- You will learn how to identify useful information and how to handle it

2 Exercise Task

The instructor will give an introduction to the topic of information consolidation and background details to the development and technology used in AbuseHelper. Mandatory for this exercise is a working internet connection as the feeds will be pulled in from online sources. All necessary parts for running the application set are available on the Virtual Image (/usr/share/trainee/14_PID/).

You will find helpful information in the documents in the References folder.

2.1 Task 1 Setting up AbuseHelper

Fill this table:

No.	Required	Answer
	XMPP user names	
	XMPP user password	
	Name of lobby room	
	Output: *botnet status /var/lib/ah2/production*	
	Have you received alert mails?	

1. Ejabber Daemon[1]
   The Ejabber Daemon is the communication channel for AbuseHelper. It is essential for the exchange of information between bots and users. The daemon has been installed from the Ubuntu repository but some basic configuration changes have to be applied.

   ```
   sudo /etc/init.d/ejabberd start # Start the Jabber service
   ```

[1] Ejabberd Installation and Operation Guide

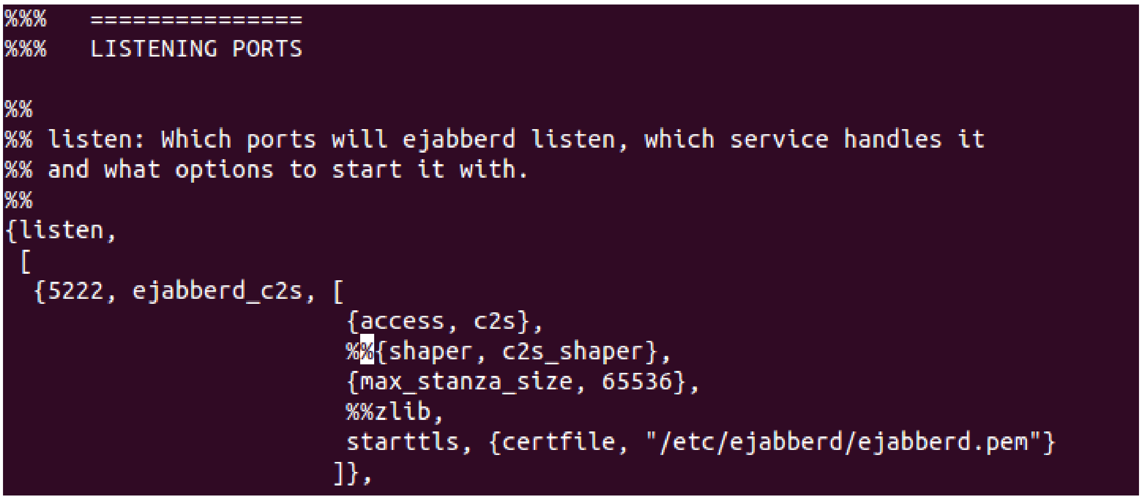
`sudo ejabberdctl register abusehel localhost exercise` *# register a user for the bots (username host password)*

`sudo ejabberdctl register trainee localhost exercise` *# register a user (username host password)*

`sudo vi /etc/ejabberd/ejabberd.cfg` *# open the ejabberd configuration file and edit the following lines*

`max_user_sessions 100` *# maximum sessions for a single user*

`s2s_default_policy deny` *# deny server to server communication*
`%%   {shaper, c2s_shaper},` *# search for and comment out the default shaper configuration*



```
%%%   ===============
%%%   LISTENING PORTS

%%
%% listen: Which ports will ejabberd listen, which service handles it
%% and what options to start it with.
%%
{listen,
 [
  {5222, ejabberd_c2s, [
                        {access, c2s},
                        %%{shaper, c2s_shaper},
                        {max_stanza_size, 65536},
                        %%zlib,
                        starttls, {certfile, "/etc/ejabberd/ejabberd.pem"}
                       ]},
```

Figure 1: ejabberd.cfg shaper configuration

{mod_muc,   [

%%{host, "conference.@HOST@"},

{access, muc},

{access_create, muc},

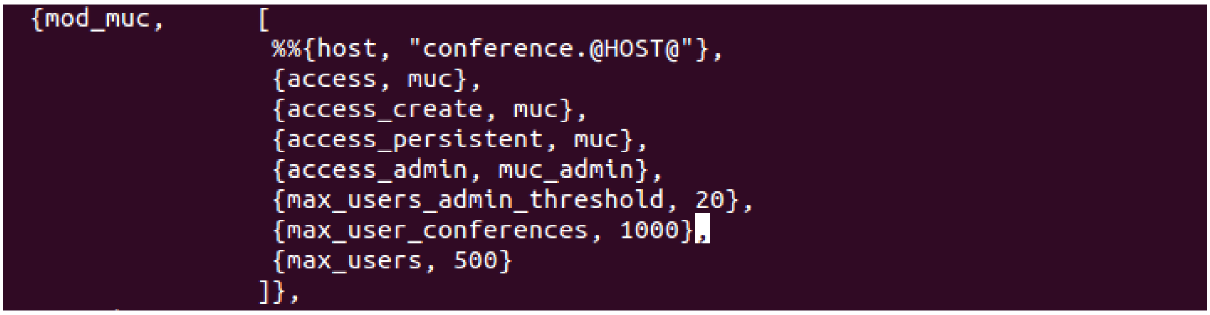
{access_persistent, muc},

{access_admin, muc_admin},

`{max_users_admin_threshold, 20},` *# add this entry*

`{max_user_conferences, 1000},` *# add this entry*

{max_users, 500}



```
  {mod_muc,      [
                  %%{host, "conference.@HOST@"},
                  {access, muc},
                  {access_create, muc},
                  {access_persistent, muc},
                  {access_admin, muc_admin},
                  {max_users_admin_threshold, 20},
                  {max_user_conferences, 1000},
                  {max_users, 500}
                 ]},
```

Figure 2: ejabberd.cfg shaper configuration

`sudo /etc/init.d/ejabberd restart` # Restart ejabberd server

2. AbuseHelper

   The next step will be to install the AbuseHelper applications and create the basic configuration.

   `sudo useradd -m abusehel` *# add a system user for AbuseHelper*

   `sudo mkdir -p /var/lib/ah2` *# create the working directory*

   `sudo chown root:abusehel /var/lib/ah2` *# ownership of the working directory*

   `sudo chmod 0750 /var/lib/ah2` *# directory access rights set to read, write*

   `cd /usr/share/trainee/14_PID/adds/abusehelper/` *# change your current directory (trainee for the students)*

   `sudo python setup.py install` *# run the AbuseHelper setup script*

   `cd /usr/local/lib/python2.7/dist-packages/abusehelper` *# change directory*

   `sudo python contrib/confgen/confgen.py /var/lib/ah2/production` *# start the configuration script*

   Enter the following information:

   XMPP username: `abusehel@localhost` # as defined during user registration

   XMPP password: `exercise` # you will be asked to enter this twice

   XMPP lobby channel: `abusehelper` # this is the initial channel to connect to when starting the Jabber client

   Configure mailer? `Yes` # let AbuseHelper send alert mails

   SMTP host: `localhost` # use the local MTA for delivery

   SMTP port: `25` # use the standard SMTP port

   SMTP auth user: `no auth` # no authentication necessary

   Mail sender: `abusehelper@localhost` # mail sender address

   `sudo chown -R root:abusehel /var/lib/ah2/production` *# access rights have to be corrected after the configuration script*

   `sudo chmod 0750 /var/lib/ah2/production` *# see above*

`sudo chmod g+w /var/lib/ah2/production/archive` *# see above*

`sudo chown abusehel /var/lib/ah2/production/log` *# this directory has been added and must be owned by the abusehel system user for logging*

`sudo chown abusehel /var/lib/ah2/production/state` *# see above*

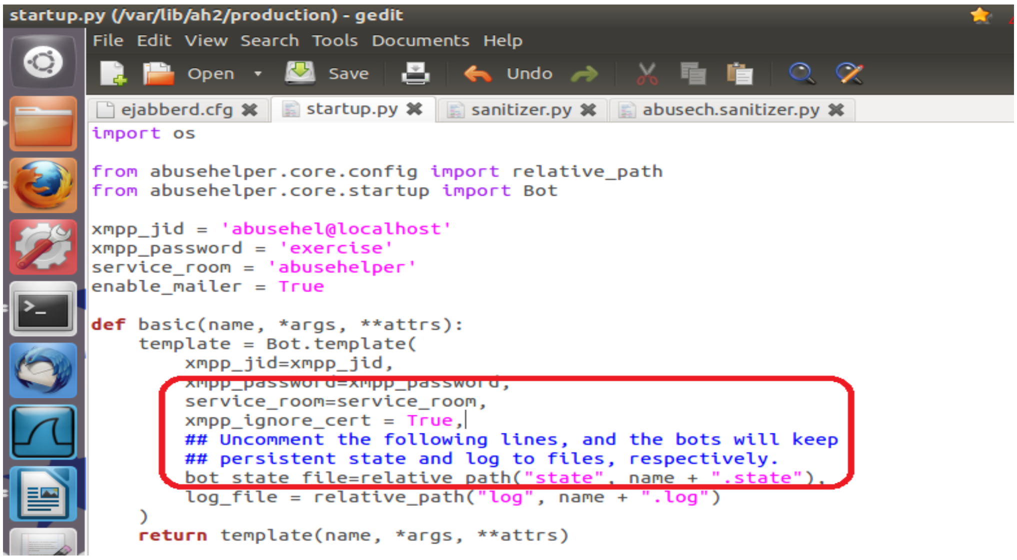
`sudo vi /var/lib/ah2/production/startup.py` *# open the startup script and check the entries made by means of the confgen script*

Insert this line after 'service_room=service_room,' in the 'def basic' section:

`xmpp_ignore_cert = True,` *# this deactivates checking ssl certificates*

Comment out the following line in the 'def configs' section:

```
#                    yield                    basic("roomgraph")
```



startup.py (/var/lib/ah2/production) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
ejabberd.cfg
startup.py
sanitizer.py
abusech.sanitizer.py

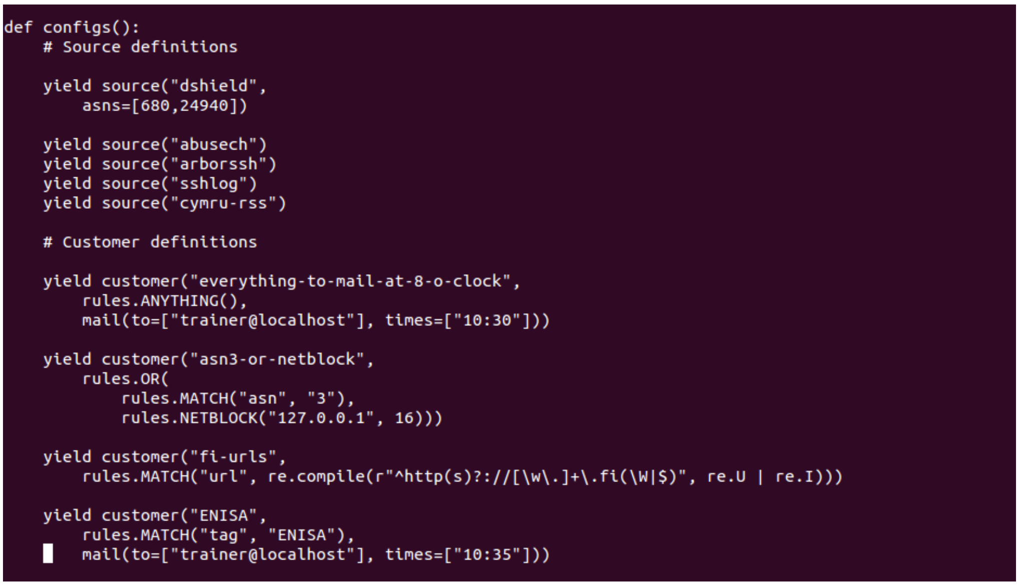
```
import os

from abusehelper.core.config import relative_path
from abusehelper.core.startup import Bot

xmpp_jid = 'abusehel@localhost'
xmpp_password = 'exercise'
service_room = 'abusehelper'
enable_mailer = True

def basic(name, *args, **attrs):
    template = Bot.template(
        xmpp_jid=xmpp_jid,
        xmpp_password=xmpp_password,
        service_room=service_room,
        xmpp_ignore_cert = True,
        ## Uncomment the following lines, and the bots will keep
        ## persistent state and log to files, respectively.
        bot_state_file=relative_path("state", name + ".state"),
        log_file = relative_path("log", name + ".log")
    )
    return template(name, *args, **attrs)
```

Figure 3: Startup.py

Configure the mail recipient in the runtime.py file:

`sudo vi /var/lib/ah2/production/runtime.py`

Change the recipient from someone@example.com to trainee@localhost



```
def configs():
    # Source definitions

    yield source("dshield",
        asns=[680,24940])

    yield source("abusech")
    yield source("arborssh")
    yield source("sshlog")
    yield source("cymru-rss")

    # Customer definitions

    yield customer("everything-to-mail-at-8-o-clock",
        rules.ANYTHING(),
        mail(to=["trainer@localhost"], times=["10:30"]))

    yield customer("asn3-or-netblock",
        rules.OR(
            rules.MATCH("asn", "3"),
            rules.NETBLOCK("127.0.0.1", 16)))

    yield customer("fi-urls",
        rules.MATCH("url", re.compile(r"^http(s)?://[\w\.]+\.fi(\W|$)", re.U | re.I)))

    yield customer("ENISA",
        rules.MATCH("tag", "ENISA"),
        mail(to=["trainer@localhost"], times=["10:35"]))
```

Figure 4: Runtime.py

3. Start AbuseHelper
   Now we are ready to start the AbuseHelper application along with the basic bots.

   `sudo su - abusehel -s /bin/bash` *# change to the abusehel system user*

   `botnet start /var/lib/ah2/production` *# start the bots defined in the startup.py script*

   `botnet status /var/lib/ah2/production` *# ask for the status, at least one instance should be running*

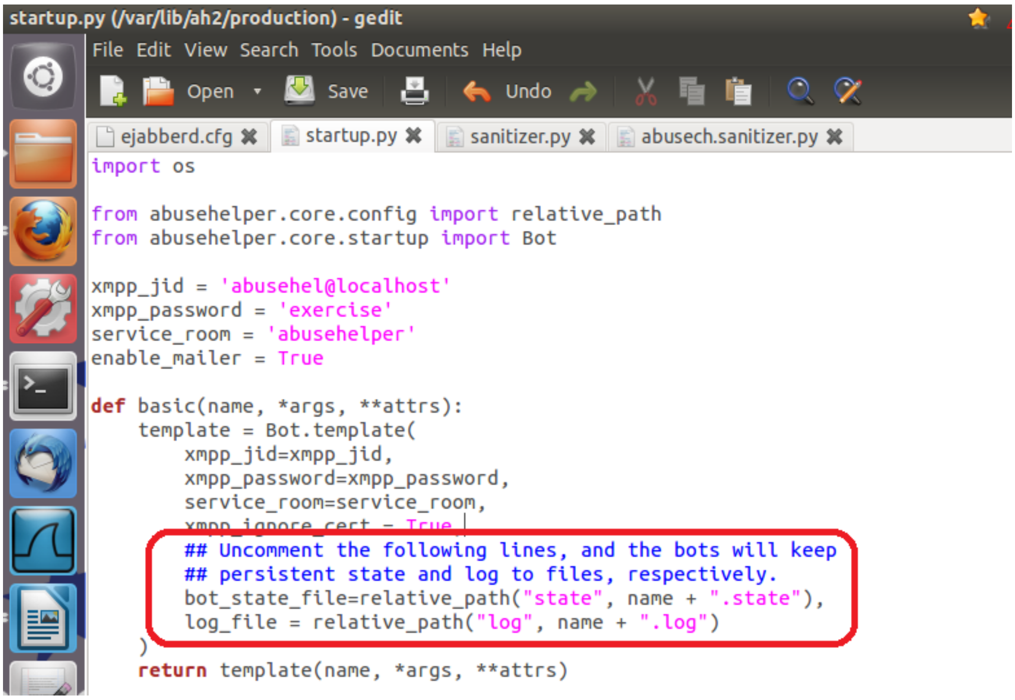
   `botnet stop /var/lib/ah2/production` *# stop the AbuseHelper bots*

   Logs can be found in these directories:

   /var/lib/ah2/production/log/

   /var/log/ejabberd/

   To enable logging functionality for every bot (logs can be found from /var/lib/ah2/production/log) uncomment the lines outlined below in the picture.


startup.py (/var/lib/ah2/production) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
ejabberd.cfg
startup.py
sanitizer.py
abusech.sanitizer.py

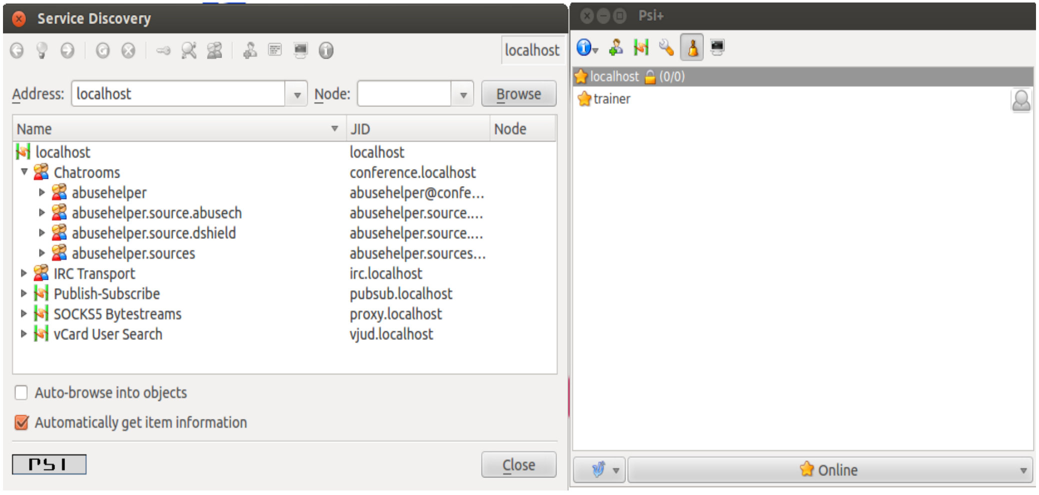
```
import os

from abusehelper.core.config import relative_path
from abusehelper.core.startup import Bot

xmpp_jid = 'abusehel@localhost'
xmpp_password = 'exercise'
service_room = 'abusehelper'
enable_mailer = True

def basic(name, *args, **attrs):
    template = Bot.template(
        xmpp_jid=xmpp_jid,
        xmpp_password=xmpp_password,
        service_room=service_room,
        xmpp_ignore_cert = True
        ## Uncomment the following lines, and the bots will keep
        ## persistent state and log to files, respectively.
        bot_state_file=relative_path("state", name + ".state"),
        log_file = relative_path("log", name + ".log")
    )
    return template(name, *args, **attrs)
```

Figure 5: Startup.py state and log configuration

4. Start Jabber clients

Communication with AbuseHelper and gathering information from the bots is mainly done by means of Jabber clients. There are several Jabber clients installed on the VM, you should at least try the following ones:

- Psi+
  Graphical client, you will have to trust the certificate presented by the Jabber service manually.



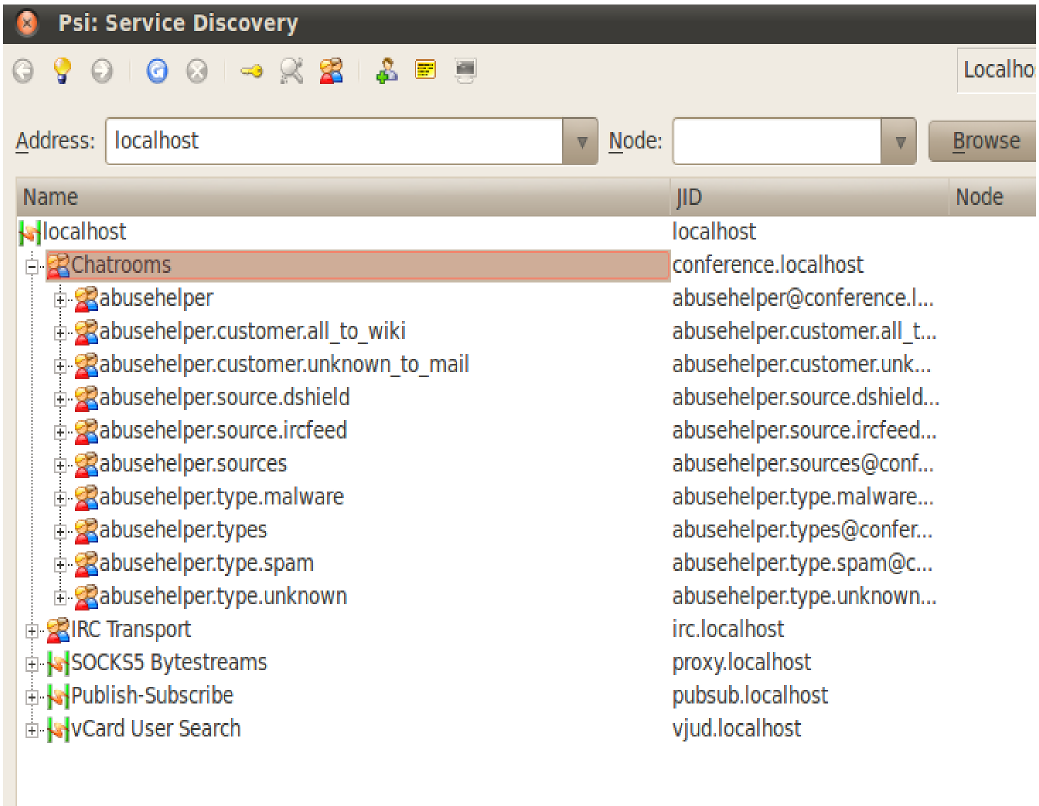
Address: localhost Node:

Name	JID	Node
localhost	localhost	
Chatrooms	conference.localhost	
abusehelper	abusehelper@confe...	
abusehelper.source.abusech	abusehelper.source....	
abusehelper.source.dshield	abusehelper.source....	
abusehelper.sources	abusehelper.sources...	
IRC Transport	irc.localhost	
Publish-Subscribe	pubsub.localhost	
SOCKS5 Bytestreams	proxy.localhost	
vCard User Search	vjud.localhost	

☐ Auto-browse into objects
☑ Automatically get item information

Figure 6: Psi+ initial configuration

This screenshot shows the service discovery feature:

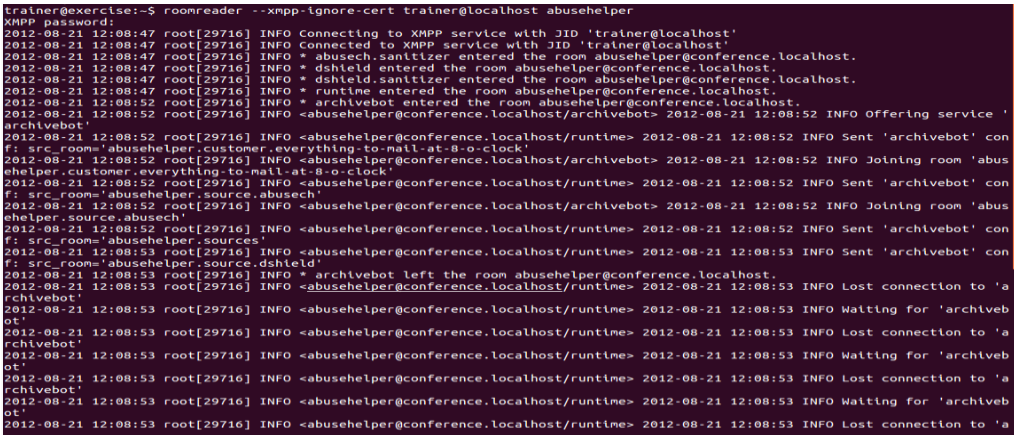


Address: localhost

Name	JID	Node
localhost	localhost	
Chatrooms	conference.localhost	
abusehelper	abusehelper@conference.l...	
abusehelper.customer.all_to_wiki	abusehelper.customer.all_t...	
abusehelper.customer.unknown_to_mail	abusehelper.customer.unk...	
abusehelper.source.dshield	abusehelper.source.dshield...	
abusehelper.source.ircfeed	abusehelper.source.ircfeed...	
abusehelper.sources	abusehelper.sources@conf...	
abusehelper.type.malware	abusehelper.type.malware...	
abusehelper.types	abusehelper.types@confer...	
abusehelper.type.spam	abusehelper.type.spam@c...	
abusehelper.type.unknown	abusehelper.type.unknown...	
IRC Transport	irc.localhost	
SOCKS5 Bytestreams	proxy.localhost	
Publish-Subscribe	pubsub.localhost	
vCard User Search	vjud.localhost	

Figure 7: Psi+ service discovery

- Roomreader
  command line client, comes with AbuseHelper
  `roomreader --xmpp-ignore-cert trainee@localhost abusehelper`



```
trainer@exercise:~$ roomreader --xmpp-ignore-cert trainer@localhost abusehelper
XMPP password:
2012-08-21 12:08:47 root[29716] INFO Connecting to XMPP service with JID 'trainer@localhost'
2012-08-21 12:08:47 root[29716] INFO Connected to XMPP service with JID 'trainer@localhost'
2012-08-21 12:08:47 root[29716] INFO * abusech.sanitizer entered the room abusehelper@conference.localhost.
2012-08-21 12:08:47 root[29716] INFO * dshield entered the room abusehelper@conference.localhost.
2012-08-21 12:08:47 root[29716] INFO * dshield.sanitizer entered the room abusehelper@conference.localhost.
2012-08-21 12:08:47 root[29716] INFO * runtime entered the room abusehelper@conference.localhost.
2012-08-21 12:08:52 root[29716] INFO * archivebot entered the room abusehelper@conference.localhost.
2012-08-21 12:08:52 root[29716] INFO <abusehelper@conference.localhost/archivebot> 2012-08-21 12:08:52 INFO Offering service '
archivebot'
2012-08-21 12:08:52 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:52 INFO Sent 'archivebot' con
f: src_room='abusehelper.customer.everything-to-mail-at-8-o-clock'
2012-08-21 12:08:52 root[29716] INFO <abusehelper@conference.localhost/archivebot> 2012-08-21 12:08:52 INFO Joining room 'abus
ehelper.customer.everything-to-mail-at-8-o-clock'
2012-08-21 12:08:52 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:52 INFO Sent 'archivebot' con
f: src_room='abusehelper.source.abusech'
2012-08-21 12:08:52 root[29716] INFO <abusehelper@conference.localhost/archivebot> 2012-08-21 12:08:52 INFO Joining room 'abus
ehelper.source.abusech'
2012-08-21 12:08:52 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:52 INFO Sent 'archivebot' con
f: src_room='abusehelper.sources'
2012-08-21 12:08:53 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:53 INFO Sent 'archivebot' con
f: src_room='abusehelper.source.dshield'
2012-08-21 12:08:53 root[29716] INFO * archivebot left the room abusehelper@conference.localhost.
2012-08-21 12:08:53 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:53 INFO Lost connection to 'a
rchivebot'
2012-08-21 12:08:53 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:53 INFO Waiting for 'archiveb
ot'
2012-08-21 12:08:53 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:53 INFO Lost connection to 'a
rchivebot'
2012-08-21 12:08:53 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:53 INFO Waiting for 'archiveb
ot'
2012-08-21 12:08:53 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:53 INFO Lost connection to 'a
rchivebot'
2012-08-21 12:08:53 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:53 INFO Waiting for 'archiveb
ot'
2012-08-21 12:08:53 root[29716] INFO <abusehelper@conference.localhost/runtime> 2012-08-21 12:08:53 INFO Lost connection to 'a
```

Figure 8: Roomreader

2.2 Task 2 Working with Abusehelper

1. Making yourself familiar with AbuseHelper
   First step will be to watch the different subrooms and identify the information flow. The table at the end of this section helps to structure and evaluate the learning process.
2. Carry on and include additional data feeds
   In this task the students should identify and describe the data feeds in the /usr/share/lib/python2.7/dist-packages/abusehelper/contrib section and document which to include. Afterwards they should configure the bots in the startup.py and runtime.py files (error messages will be logged to the bot files in the /var/lib/ah2/production/log/ folder).
3. Filter information feeds
   There are different ways to filter the incoming information to be more relevant to your organisations infrastructure.
   Start with dshield and open the runtime.py. You will find an entry regarding the ASN. Change the (Autonomous System Numbers) ASN to your organisation's network(s). Edit runtime.py (/var/lib/ah2/production/runtime.py) to filter ASN numbers. A list linking ASN to organisations can be found here. The functionality of this filter mechanism is implemented in the dshield bot itself.
   Sanitizers take the raw data provided by the bots, clean it according to the configuration and deliver it into the abusehelper.sources room. Examples for sanitizer scripts are available in /var/lib/ah2/production/custom/. These can be easily adapted for other bots. These fields/tags can be used in rules. Create a sanitizer script for one of the bots from contribution and modify it to add the "ENISA" tag to the output.
   You can write rules to filter output. First tweak the def _mail section in the runtime.py to use abusehelper.sources as data input. Then add a customer definition to send all data tagged with "ENISA" to trainee@localhost

Fill the table with the requested information:

No.	Question	Answer
1	Which feeds are standard?	
2	Which information do these deliver?	
3	Where are additional feed bots available?	
4	Integrate the Arbor SSH bot startup.py: yield basic('arborssh','abusehelper.contrib.arbor.ssh') runtime.py: yield source('arborssh')	

5	Integrate the sshlogbot startup.py: yield basic('sshlog', 'abusehelper.contrib.sshlogbot.sshlogbot', path='/var/log/auth.log') runtime.py: yield source('sshlog')	
6	Integrate the Team Cymru RSS feed startup.py: yield basic('cymru rss','abusehelper.contrib.rssbot.rssbot', feeds='http://www.team-cymru.org/News/secnews.rss') runtime.py: yield source('cymru-rss')	
7	Create sanitizer scripts for the included bots (copy from existing abusech.sanitizer.py)	
8	Modify one sanitizer to add tag=ENISA to the output and send corresponding report to localhost mailbox.	
9	Name three bots you find most useful for your work and give reasons for your decision	

3 Conclusion

Finishing this exercise you will have learned the following:

- Installation of Abusehelper
- Base configuration of AbuseHelper and default bots
- Evaluating and integrating bots from the contribution folder
- Filtering of input and output information with rules
- Accessing AbuseHelper information

The provided information should give you a starting point for the evaluation/implementation of AbuseHelper in your organization.

ENISA
European Union Agency for Network and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office
1 Vass. Sofias & Meg. Alexandrou
Marousi 151 24, Athens, Greece

PO Box 1309, 710 01 Heraklion, Greece
Tel: +30 28 14 40 9710
info@enisa.europa.eu
www.enisa.europa.eu