# Dovecot Anti-Abuse Shield
## **Release Notes for Release** 1.4.0

2017-10-05

## Copyright notice

# Contents

# 1 General Information

Dovecot Anti-Abuse Shield is included with Dovecot Pro and works with both Dovecot Pro and OX App Suite as a component to protect against login/authentication abuse.
Anti-Abuse Shield runs on a cluster of servers, and integrates with OX App Suite and Dovecot to detect abuse, brute force attacks and also to enforce common authentication/authorization policies across the platform.

Please note: Dovecot Anti-Abuse Shield is only available for customers who have a valid Dovecot Pro license and Dovecot Pro v2.2.27.1. onwards.

New Main Features include:

- The base DAAS package ships with a simple policy to handle brute-forcing protection. This is now supplemented with a separate package providing a comprehensive anti-abuse policy, with easy configuration, containing over 16 different policies including restrictions based on GeoIP, and user/IP whitelisting.
- Support for GeoIP based on City and ISP databases
- Support for reloading GeoIP databases without restarting DAAS
- Blacklisting based on CIDR ranges not just individual IP addresses

A Whitepaper can be found at: http://software.open-xchange.com/products/appsuite/doc/Whitepaper_Dovecot_Anti_Abuse_Shield.pdf.

For details of how to install and configure Dovecot Anti-Abuse Shield please refer to the instructions found at: http://oxpedia.org/wiki/index.php?title=AppSuite:Dovecot_Antiabuse_Shield.

### Additional GeoIP DB Support

The new config commands are described in full in `man wforce.conf`, the following is an extract:

- `initGeoIPCityDB()` Initializes the city-level IPv4 and IPv6 GeoIP databases. If either of these databases is not installed, this command will fail and wforce will not start. Ensure these databases have the right names if you're using the free/lite DBs - you may need to create symbolic links e.g. GeoIPCityv6.dat -> GeoLiteCityv6.dat. For example:

  ```
  initGeoIPCityDB().
  ```

- `initGeoIPISPDB()` Initializes the ISP-level IPv4 and IPv6 GeoIP databases. If either of these databases is not installed, this command will fail and wforce will not start. For example:

  ```
  initGeoIPISPDB()
  ```

### Support for reloading GeoIP DBs

The following new console command is available:

- `reloadGeoIPDBs()` Reload all GeoIP DBs that have been initialized. For example:

  ```
  > reloadGeoIPDBs()
  reloadGeoIPDBs() successful
  ```

### Netmask support in Blacklists

The following new Lua commands are available:

- `newNetmask(\<IP[/mask]\> )` Create and return an object representing a Netmask. For example:

  ```
  my_nm = newNetmask("8.0.0.0/8")
  ```

- `blacklistNetmask(\<Netmask\> , \<expiry\> , \<reason string\> )` Blacklist the specified netmask for expiry seconds, with the specified reason. Netmask address must be a Netmask object, e.g. created with `newNetmask()`. For example:

  ```
  blacklistIP(newNetmask("12.32.0.0/16"), 300, "Attempted password brute forcing")
  ```

There is also a new `netmask` parameter to the HTTP REST API `addBLentry` and `delBLEntry` commands. The netmask parameter is mutually exclusive with the existing ip parameter. For example:

```
curl -H "Content-Type:  application/json" -X POST --data '{"netmask":"2001:503:ba3e::2:30/64",
"expire_secs":100, "reason":"foo" }' http://localhost:8084/?command=addBLEntry -u user:password.
```

### Configurable Timeout for Redis Connections

Previously wforce would hang on startup or during operation if configured to use Redis and the Redis server was unavailable. Now wforce will timeout if the Redis server cannot be reached, and the timeout is configurable in Lua:

- `blacklistPersistConnectTimeout(<timeout secs> )` Set the connect timeout for connecting to the persistent redis DB. If the timeout is exceeded during connection at startup then wforce will exit, otherwise during normal operation if the timeout is exceeded, an error will be logged. For example:

  ```
  blacklistPersistConnectTimeout(2)
  ```

### New Lua Logging Functions

There is a new Lua logging function, which can be useful for diagnosing issues:

- `debugLog(\<log string\> , \<key-value map\> )` Log at `LOG_DEBUG` level the specified string, adding `"key=value"` strings to the log for all the kvs specified in the key-value map, but only if wforce was started with the (undocumented) `-v` flag (for verbose).

  For example:`debugLog("This will only log if wforce is started with -v", { logging=1, foo=bar })`

### Wforce changes cwd to config directory

Wforce will now change working directory to the directory containing the `wforce.conf` file. This enables Lua modules to be used, with paths relative to the configuration directory (e.g. `/etc/wforce`).

### Webhooks support Basic Authentication

A new configuration key has been added to Webhooks, to support basic authentication:

- `basic-auth` Adds an Authorization header to Webhooks, for servers which expect Basic Authentication. The username and password are provided as `"user:pass"`. For example:

  ```
  config_key["basic-auth"] = "wforce:super"
  ```

### Support for Parsing device_id

The `device_id` parameter to the allow and report commands existed in 1.2.x, however it was not parsed by wforce. Now the parameter is parsed if the protocol parameter is one of `http`, `imap` or `mobileapi`. Note that if the protocol parameter is non-existent or does not match the above list, then `device_id` will not be parsed. The `device_id` is parsed into key value pairs of the `device_attrs` object, as follows:

- `LoginTuple.device_attrs` Additional array of attributes about the device, which is parsed from the `device_attrs` string. The protocol string is used to determine how to parse `device_id`,

so that MUST also be present. For all protocols, the following keys are set wherever possible: `os.family`, `os.major`, `os.minor`. For `http`, the following additional keys are set wherever possible: `device.family`, `device.model`, `device.brand`, `browser.family`, `browser.major`, `browser.minor`. For `imap`, the following additional keys are set wherever possible: `imapc.family`, `imapc.major`, `imapc.minor`. For `mobileapi`, the following additional keys are set: `app.name`, `app.brand`, `app.major`, `app.minor`, `device.family`. For example:

```
if (lt.device_attrs["os.family"] == "Mac OS X")
then
-- do something special for MacOS
end
```

- `LoginTuple.protocol` A string representing the protocol that was used to access mail, i.e. `http`, `imap`, `pop3`, `mobileapi` etc. `LoginTuple.protocol` MUST be set in order to parse `device_id` into `device_attrs`, however currently only `http`, `imap` and `mobileapi` are recognized protocols when parsing `device_id`. For example:

```
if (lt.protocol == "http")
then
-- do something
end
```

### New TLS Parameter

The new `tls` parameter is used to indicate whether the client session used TLS or not. It may not be completely reliable, e.g. if TLS offload is done on load balancers.

- `LoginTuple.tls` A boolean representing whether the login session used TLS or not. If the client is using TLS offload proxies then it may be set to `false`.

### Content-Length instead of Chunked Encoding for Webhooks

Webhooks previously used Chunked encoding, however that is not currently supported by Logstash, so Content-Length is used instead.

### New Console Commands

The following new console commands are available:

- `showPerfStats()` Returns information about performance statistics. Stats beginning with WTW refer to the time that worker threads waited in a queue before running. Stats beginning with WTR refer to the time that worker threads took to run. Each stat is in a bucket, where each bucket represents a time range in ms, e.g. 0-1. A server that is not overloaded will have most stats in the 0-1 buckets. For example:

```
> showPerfStats()
WTW_0_1=2939287
WTW_1_10=9722
WTW_10_100=4
WTW_100_1000=0
WTW_Slow=0
WTR_0_1=2939229
WTR_1_10=2837
WTR_10_100=131
WTR_100_1000=0
WTR_Slow=0
```

- `reloadGeoIPDBs()` Reload all GeoIP DBs that have been initialized. For example:

```
> reloadGeoIPDBs()
reloadGeoIPDBs() successful
```

# 2 Shipped Product and Version

Dovecot Anti-Abuse Shield 1.4.0-rev1

Find more information about product versions and releases at http://oxpedia.org/wiki/index.php?title=AppSuite:Versioning_and_Numbering and http://documentation.open-xchange.com/.