OX Abuse Shield

Release Notes for Release 2.0.0

2019-01-16

Copyright notice

1 General Information

Open-Xchange is pleased to announce the release of the new major version of OX Abuse Shield v2.0. Open-Xchange decided to re-name the product from Dovecot Anti-Abuse Shield to OX Abuse Shield. This name change corresponds to the ability to use OX Abuse Shield in different environments.

OX Abuse Shield provides abuse-prevention for Web Applications (including Webmail), POP, and IMAP. It is integrated with both OX App Suite and Dovecot Pro to prevent login and authentication abuse as well as protecting against brute-force attacks.

The goal of OX Abuse Shield is to detect brute forcing of passwords across many servers, services and instances, as well as enforce policy for authentication and authorization. In order to support the real world, brute force detection policy can be tailored to deal with ”bulk, but legitimate” users of your service, as well as botnet-wide slow-scans of passwords.

The new OX Abuse Shield v2.0 provides the following main improvements and new features:

- Fix extra : at the end of custom log lines when kv table is empty
- Return additional info about blacklist in allow and getDBStats RESTAPI functions
- New Lua functions to remove blacklist entries
- Add configuration setting ”setNumWebHookConnsPerThread”
- Add configuration setting ”setWebHookTimeoutSecs”
- Add support for querying replication status in showStringStatsDB()
- Add sibling received success/fail stats to siblings() command
- New custom stats framework
- New stats for all commands, including custom commands
- GeoIP2 support (MMDB-style DBs)
- New twResetField() function for statsDBs
- Configurable accuracy for HLL and CountMin types
- DB Synchronization for newly started wforce instances
- Support for replication over TCP
- Customizable log facility via a command line option
- New trackalert daemon
- Logstash Configuration and Elasticsearch Templates
- Kibana Reports and Dashboards
- Report API

The new wforce-policy 2.0.0 provides the following main improvements and new features:

- Lua reset function nows unblacklists IPs/Logins
- Return allow/report/reset functions in wforce policy module table
- IP Graylist - Only tarpit never reject IPs that are in the graylist
- New 24 hour policies that count tarpits in addition to existing policies that count rejects

- Support Elasticsearch 6
- Config option ”logPolicyCode” to enable/disable policy code logging
- Trackalert Policy
- Custom postfunc for Allow Command
- GeoIP2 Support (MMDB)
- Redis cache of previously used devices, locations, protocols.

What’s New in General

Open-Xchange now provides more detailed overviews, data sheet and product guide, relating to new product major release. These can be found at https://www.open-xchange.com/portfolio/whats-new/

A more detailed overview of the main functions and technical descriptions of OX Abuse Shield can be found at the Whitepaper under: http://software.open-xchange.com/products/weakforced/doc/OX_Whitepaper_OX_Abuse_Shield_2_0_0.pdf

General Information – Please Note

OX Abuse Shield v2.0 introduces an upgrade path from v1.4.4 to v2.0. Please read http://oxpedia.org/wiki/index.php?title=AppSuite:OX_Abuse_Shield#Debian_Upgrade_Instructions_with_OX_Abuse_Shield_v2.0 carefully to familiarize with the details, implications and our proposed upgrade path.

In accordance with the supported platform policy, OX Abuse Shield v2.0 supports Debian 9 (Stretch) and CentOS 7. Debian 8 (Jessie) is no longer supported with OX Abuse Shield v2.0.

Download and Installation

For further details about OX Abuse Shield installation, mandatory and optional packages, policies, please refer to the documentation provided: http://oxpedia.org/wiki/index.php?title=AppSuite:OX_Abuse_Shield

Additional Information about new Functionalities

Fix Extra : at end of Custom Log Lines

The Lua infoLog, errorLog etc. functions would previously, when called as ’errorLog(”foo”, {})’ log:

```
foo :
```

Now the same call will log only:

```
foo
```

Return additional info about blacklist in allow and getDBStats REST API functions

The allow command will now return additional information when an IP/Login is blacklisted. The ’r_attrs’ object will contain four new fields:

- expiration - A string showing the date/time when the blacklist will expire
- reason - A string stating why the blacklist was created
- key - What was blacllisted, i.e. either ip, login or iplogin

- blacklisted - This will be set to 1

The getDBStats command will return additional information about blacklisted objects:

- bl_expire - A string showing the date/time when the blacklist will expire
- lb_reason - A string stating why the blacklist was created

New Lua functions to remove blacklist entries

The following new Lua functions are available:

- unblacklistNetmask
- unblacklistIP
- unblacklistLogin
- unblacklistIPLogin

See the wforce.conf manpage for more details.

New Configuration Setting setNumWebHookConnsPerThread

The webhook support has been completely refactored in order to achieve much higher performance with fewer resources. Previously a very high number of webhook threads was required to achieve good performance, whereas now a much smaller number of threads can achieve the same performance.

The previous per-webhook configuration key ”num_conns” is no longer supported. Instead the global configuration setting ”setNumWebHookConnsPerThread” is used.

For example:

```
setNumWebHookConnsPerThread(10)
```

The default is 10 connections per webhook thread.

Add configuration setting ”setWebHookTimeoutSecs”

The function setWebHookTimeoutSecs() is used to control the time for webhook requests, e.g.:

```
setWebHookTimeoutSecs(2)
```

Support for querying replication status in showStringStatsDB()

The showStringStatsDB() command now shows whether each StatsDB is configured for replication or not.

For example:

```
showStringStatsDB() DB Name Repl? Win Size/No Max Size Cur Size Field Name Type MyDB1 yes
1/15 524288 0 countLogins int diffPasswords hll MyDB2 no 600/6 5000 2093 diffIPs hll
```

Add Sibling received success/fail stats to siblings() command</para>
The siblings() command now shows success and failure stats about received messages as well as sent messages.
For example:

```
siblings() Address Send Successes Send Failures Rcv Successes Rcv Failures Note 127.0.0.1:4001
0 0 17 0 127.0.0.1:4002 0 0 0 0 Self
```

New Custom Stats Framework

Two new functions, ”addCustomStat” and ”incCustomStat” can be used to keep track of custom statistics. A new custom counter is created with ”addCustomStat”, e.g.

```
addCustomStat("custom_stat1")
```

Custom statistics are counters which track statistics over a 5 minute period. Every 5 minutes, the current values of all the custom stats counters are logged to the wforce log file, before the counters are reset.

Stats can be incremented with the ”incCustomStat” command:

```
incCustomStat("custom_stat1")
```

New stats for all commands

Previously there were no statistics logged for all the REST API commands; only allow and report commands. Now all REST API commands are tracked and statistics are reported, including for custom endpoints created from Lua.

For example:
```
command stats last 300 secs: addBLEntry=42 allow=393827 allow_allowed=299221
allow_blacklisted=3224 allow_denied=9884 allow_tarpitted=8373 delBLEntry=3 getBL=3949 getDBStats=322
ping=83764 report=38473 reset=0 stats=0 syncDBs=0 syncDone=0 custom stats last 300 secs:
customFunc1=3401
```

GeoIP2 support (MMDB-style DBs)

Maxmind are in the process of deprecating the GeoIP Legacy DB support, therefore this release supports GeoIP2 format databases, i.e. the MMDB format.

This release therefore deprecates the GeoIP legacy functions, which will be removed in a later release. The following functions are deprecated:

- initGeoIPDB()
- initGeoIPCityDB()
- initGeoIPISPDB()
- lookupCountry()
- lookupISP()
- lookupCity()

Due to differences in the way that the GeoIP2 API works, GeoIP Databases must be opened by specifying the filename of the database to be used. For example:

```
newGeoIP2DB("CityDB","/usr/share/GeoIP/GeoLite2-City.mmdb")
```

To retrieve a GeoIP DB to conduct queries against, use the following command:

```
local citydb = getGeoIP2DB("CityDB")
```

Once a database has been assigned to a local variable, it can be queried, for example:

```
my_country = countrydb:lookupCountry(newCA("8.8.8.8")) my_country = countrydb:lookupCountry(lt.remot
local my_isp = ispdb:lookupISP(newCA("128.243.16.21")) local gip_record = citydb:lookupCity(lt.remot
local my_city = gip_record.city local my_latitude = gip_record.latitude
```

For full details see ”man wforce.conf”.

New twResetField() function for statsDBs

The twReset() function can be used to reset all the fields for a given key, but previously there was no way to reset an individual field. Now the function "twResetField()" can be used to achieve this, e.g.:

```
statsdb:twResetField(lt.login, "countLogins")
```

Configurable accuracy for HLL and CountMin types

The HLL and CountMin types of StatsDB entries are probablistic data structures, which trade accuracy for memory usage. Previously the accuracy (and thus memory usage) was hardcoded, however now their accuracy can be tuned. Increasing accuracy however means a (potentially very large) increase in memory usage, so extreme care must be taken before modifying these parameters.

The function setHLLBits() can be used to change the accuracy of the HLL type. The value supplied can be between 4 and 30, with the default value being 6.

The function setCountMinBits() can be used to set the accuracy of the CountMin type.

See the wforce.conf manpage for full details.

DB Synchronization for newly started wforce instances

Normally when a wforce instance starts, it has a "fresh" set of Stats DBs, and therefore can take a reasonable period of time (an hour or more depending on the policy) before it starts giving the same answers as other wforce servers in a cluster which have been running for some time. This issue is now addressed with the ability to tell a wforce server to find another server which has been running for longer than a configurable period of time, from which it can download the entire set of Stats DBs. While a server is in the process of downloading the Stats DBs from another server, it is in a "warmup" state; this fact is reflected in a new return value from the "ping" REST API endpoint.

In order to enable this feature, the "addSyncHosts()" function must be used, once for each host that will be contacted on startup, for example:

```
-- Add 10.2.3.1:8084 as a sync host, -- and use the password "super" -- Send the DB dump
to 10.2.1.1:4001 -- and let me know on 10.2.1.1:8084 when the dump is finished addSyncHost("10.2.3.1
"super", "10.2.1.1:4001", "10.2.1.1:8084")
```

The default time that the sync hosts must have been "up" for is 3600 seconds, however that can be configured using "setMinSyncHostUptime()", e.g.:

```
setMinSyncHostUptime(1800)
```

The replication of data between the sync host and the wforce instance that is starting up is always performed over TCP.

See wforce.conf for full details.

Support for replication over TCP

The addSiblings() and setSiblings() functions now take an extra (optional) parameter that specifies whether the replication should use UDP or TCP. The default is UDP. If the protocol is specified, the port must also be specified.

For example:

```
setSiblings({"127.0.1.2", "127.0.1.3:4004", "127.0.2.23:4004:tcp"}) addSibling("192.168.1.23")
```

```
addSibling("192.168.1.23:4001:udp") addSibling("192.168.1.23:4003:tcp")
```

Customizable log facility via a command line option

The new ”-f” or ”–facility” command line option can be used to set the syslog facility used for wforce logging.

For example:
```
wforce -f "local0"
```

New trackalert daemon

A new daemon ”trackalert” is part of the product. This daemon shares a lot of functionality with wforce, particularly in terms of Lua support. However the REST API for trackalert is much simpler, consisting only of ”report” and ”stats” endpoints.
The trackalert daemon is designed to process login reports sent to it by wforce, use those reports to determine whether the login is suspicious. It is also designed to run Lua functions on a periodic basis using a configurable scheduler, in order to run tasks such as finding suspicious IPs or compromised accounts.

The trackalert daemon works best with the Lua policy delivered in the separate wforce-policy package. That policy implements suspicious login alerts using historical report data stored in Elasticsearch, as well as periodic searches of Elasticsearch to find suspicious IPs and compromised accounts.

For the trackalert daemon to be effective, wforce must be configured to send reports to both trackalert and Elasticsearch using webhooks.

Logstash Configuration and Elasticsearch Templates

This release ships with sample logstash configuration and Elasticsearch mapping template to ensure that the report data is stored in a consistent form by Elasticsearch.
The minimum version of ELK (Elasticsearch, Logstash, Kibana) that is required is version 6.

Kibana Reports and Dashboards

This release ships with a sample set of reports and dashboards for Kibana (version 6+).

Report API

A REST API to handle querying and modification of the data stored in Elasticsearch.
Currently this ships as an informational feature for experimentation; a future release will ship this API as a package shipping a deployable and supported webapp.
The API is documented using OpenAPI (Swagger); consult the documentation on documentation.open-xchange.com.

IP Graylist

A new IP Graylist feature has been added to the wforce policy, which allows a list of IPs (CIDRs) to be defined that will never be rejected, only tarpitted. For example this can be used for customer IP ranges, to ensure that customers are never blocked.

See the ”IP Graylist” section of the wforce_policy.8 manpage for more details.

New 24-hour Policies

In conjunction with the IP Graylist, new 24-Hour Policies have been created, which count tarpits rather than rejects. The new policies are identical to the existing policies in all other respects.

The new policies are:

- maxTarpitsPerHourPerIP
- maxTarpitsPerHourPerUser
- maxTarpitsPerHourPerIPUser
- maxTarpitsPerDayPerIP
- maxTarpitsPerDayPerUser
- maxTarpitsPerDayPerIPUser

Support Elasticsearch 6

Elasticsearch 6 is the first in a series of Elasticsearch releases that deprecate various features, particularly the the concept of index types. The changes required to support Elastic 6 mean that the minimum supported version of Elasticsearch is 6.x.

Config option "logPolicyCode"

There is a new wforce config option "logPolicyCode". By default this is set to true, but if set to false, the policy_code field will not be logged.

Trackalert Policy

The new trackalert daemon that is part of the wforce 2.0.0 release requires a policy to be effective. This release provides an initial policy for the trackalert daemon, which performs the following functionality:

- Queries Elasticsearch when a new report is received, in order to determine if the login is suspicious. Uses the device type, location (country) and protocol to make this determination. Sends an alert about suspicious logins to either the administrator via a webhook, or the user via email.
- Schedules regular queries against Elasticsearch to determine potentially suspicious IP addresses and potentially compormised accounts. Sends the results of those queries via a webhook.

See trackalert_policy.8 manpage for more details.

Custom Postfunc for Allow Command

The wforce policy now allows a custom postfunc function for the allow command:

```
allow_postfunc = myAllowFunction
```

See wforce_policy.8 manpage for more details.

GeoIP2 Support

With the imminent deprecation and removal of the free GeoIP legacy databases, GeoIP2 support has been added to the wforce and trackalert policies.

All policies now use GeoIP2 commands, so these policies will break if GeoIP2 MMDB databases are not available.

Redis cache of previously used devices, locations, protocols

The information stored in Elasticsearch is considered canonical, however performing a full Elasticsearch query for every received report can be resource intensive and high latency, particularly when most logins are not suspicious.

Therefore the trackalert policy maintains a cache of known ”good” devices, locations, and protocols. Any login that matches these will immediately be considered non-suspicious. Logins that do not match these will trigger a full Elasticsearch query.

The redis cache can be used by the wforce policy, using the following config keys:

```
redis = { redis_enabled = true, expire_secs = 5184000, -- 60 days expiry for redis cache
entries redis_server = "127.0.0.1", redis_port = 6379,}
```

The ”suspiciousLogin” field is returned by wforce allow commands if a login does not match the data found in Redis. This does not mean that the login is necessariyl bad, but it may be used to trigger extra checks, e.g. second factor authentication.

2 Shipped Product and Version

OX Abuse Shield 2.0.0-rev8

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/.