QIWI Host Based Firewall (QHBF)

Инструкция по установке и настройке

Оглавление

[Технические требования: 3](#_Toc134782433)

[Установка серверной части 3](#_Toc134782434)

[Настройка серверной части 6](#_Toc134782435)

[Установка клиентской части 7](#_Toc134782436)

# Предварительные действия

Установочные файлы для серверной и клиентской частей комплекса доступны по ссылке: <https://owncloud.qiwi.com/index.php/s/aXPBi2jFcnK8P6W>

Пароль для скачивания: K2cTH7oDtc

На сервере установки должны быть предварительно установлены и настроены:

* ОС Debian 9 (stretch)
* Uwsgi 2.0.14+20161117-3+deb9u5
* Python3.5
* Nginx 1.10.3-1+deb9u7
* Redis server 3.2.12
* RabbitMQ 3.8.19
* Elasticsearch 8.5.3
* сетевой доступ к сервисам, которые должны быть доступны в QHBF

# Установка серверной части

1. Создайте пользователя www-data
2. Распакуйте архив в /var/
3. Дайте пользователю www-data права владельца на каталог /var/www/
4. Создайте файлы конфигурации:
	1. **/etc/uwsgi/apps-enabled/idm.ini**

|  |
| --- |
| [uwsgi]plugins = python3chdir = /var/www/idmmodule = idm.wsgi:applicationmaster = trueprocesses = 8listen = 100listen-queue-alarm = truelog-master-stream = truedaemonize = /var/log/uwsgi/uwsgi.logpythonpath = /var/www/site-packagesenv = LOG\_DIR=/var/log/uwsgienv = PATH=$PATH:${ORACLE\_HOME} |

* 1. **/etc/uwsgi/apps-enabled/hbf-agent.ini**

|  |
| --- |
| [uwsgi]plugins = python3chdir = /var/www/idmmodule = idm.wsgi:applicationmaster = trueprocesses = 16listen = 100listen-queue-alarm = truelog-master-stream = falsedaemonize = /var/log/uwsgi/uwsgi.logpythonpath = /var/www/site-packagesenv = LOG\_DIR=/var/log/uwsgienv = PATH=$PATH:${ORACLE\_HOME} |

* 1. **/etc/nginx/sites-enabled/idm-qiwi-com**

|  |
| --- |
| server {    proxy\_set\_header X-Real-IP $remote\_addr;    proxy\_set\_header X-Forwarded-For $remote\_addr;    proxy\_set\_header Host $host;    proxy\_pass\_request\_headers on;    client\_body\_buffer\_size 16M;    client\_max\_body\_size 16M;     listen 80;    listen [::]:80;    server\_name hbf.your.domain;     access\_log /var/log/nginx/access.log;    error\_log /var/log/nginx/error.log;     include /var/www/idm/conf/nginx.conf;     location ~ \.(DS\_Store|svn|git|htpasswd|htaccess|bash\_history) {        return 404;    }} server {     proxy\_set\_header X-Real-IP $remote\_addr;    proxy\_set\_header X-Forwarded-For $remote\_addr;    proxy\_set\_header Host $host;    proxy\_pass\_request\_headers on;    client\_body\_buffer\_size 16M;    client\_max\_body\_size 16M;     listen 443;    listen [::]:443;    server\_name hbf.your.domain;    access\_log /var/log/nginx/access.log;    error\_log /var/log/nginx/error.log;     include /var/www/idm/conf/nginx.conf;     location ~ \.(DS\_Store|svn|git|htpasswd|htaccess|bash\_history) {        return 404;    }} |

1. Создайте сервис **/etc/systemd/system/celery.service**

|  |
| --- |
| [Unit]Description=celeryAfter=syslog.target network.target[Service]Type=forkingEnvironmentFile=-/etc/default/%pUser=www-dataExecStartPre=-/bin/mkdir -p /var/run/celeryExecStartPre=-/bin/mkdir -p /var/log/celeryExecStartPre=-/bin/chown www-data:adm /var/run/celeryExecStartPre=-/bin/chown www-data:adm /var/log/celeryExecStart=/usr/local/bin/celery multi start worker1 hbf\_deploy -A idm -c 16 -c:hbf\_deploy 1 -B --pidfile=/var/run/celery/%%n.log --logfile=/var/log/celery/%%n.log --loglevel="INFO" -Q:hbf\_deploy hbf\_deployExecStop=/bin/sh -c "celery multi stopwait worker1 hbf\_deploy"Restart=alwaysRestartSec=10secSyslogFacility=local5SyslogIdentifier=celeryWorkingDirectory=/var/www/idmEnvironment=PYTHONPATH=/var/www/site-packagesPermissionsStartOnly=trueGroup=www-data[Install]WantedBy=multi-user.target |

1. Выполните команду sudo systemctl daemon-reload

Пример конфигурации /var/www/idm/settings.py:

|  |
| --- |
| ALLOWED\_HOSTS = [“hbf.your.domain”]SECRET\_KEY = ‘your\_random\_secret\_key’REDIS\_HOST = 'your\_redis\_host'REDIS\_PORT = 6379HBF\_REDIS\_DB = 1REDIS\_PASSWORD = "your\_redis\_password"RABBIT\_USER = "your\_rabbit\_user"RABBIT\_PASSWORD = “your\_rabbit\_password”RABBIT\_HOST = 'your\_rabbit\_host\_address:5672'RABBIT\_VHOST = 'your\_rabbit\_vhost'ES\_HOST = "https://your\_elasticsearch\_host:9200"ES\_USER = "your\_es\_user"ES\_PASSWORD = "your\_es\_password"DATABASES = {    'default': {        'ENGINE': 'django.db.backends.postgresql',        'NAME': 'idm',        'USER': 'your\_postgres\_user',        'PASSWORD': 'your\_postgres\_password',        'HOST': 'your\_postgres\_host\_address',        'PORT': '5432',        'OPTIONS': {            'options': '-c search\_path=idm'        },        'CONN\_MAX\_AGE': 100    },    'inventory': {        'ENGINE': 'django.db.backends.postgresql',        'NAME': 'idm',        'USER': 'your\_postgres\_user',        'PASSWORD': 'your\_postgres\_password',        'HOST': 'your\_postgres\_host\_address',        'PORT': '5432',        'OPTIONS': {            'options': '-c search\_path=inventory'        },        'CONN\_MAX\_AGE': 100    },    'migration': {        'ENGINE': 'django.db.backends.postgresql',        'NAME': 'idm',        'USER': 'your\_postgres\_user',        'PASSWORD': 'your\_postgres\_password',        'HOST': 'your\_postgres\_host\_address',        'PORT': '5432',    }} |

1. В каталоге /var/www/idm выполните команды:

|  |
| --- |
| export PYTHONPATH=/var/www/site-packagespython3.5 manage.py migratepython3.5 manage.py createsuperuser/etc/init.d/uwsgi startservice nginx start |

Установка завершена, комплекс запущен.

# Настройка серверной части

Используя данные пользователя, созданного на последнем шаге установки, авторизуйтесь в административной панели по адресу [http://**hbf.your.domain**/\_admin](http://hbf.your.domain/_admin) и создайте новых пользователей.

Предоставьте новому пользователю права: перейдите по пути Начало › Пользователи и группы › Пользователи, выберите необходимого пользователя и в разделе “Права доступа” добавьте группы idm-hbf-view и idm-hbf-edit

В postgresql, в таблице idm.inventory.hosts добавьте записи, пример:

|  |
| --- |
| {'a\_records': None, 'cname\_records': None, 'environment': 'testing', 'extra': {'ad': {'distinguishedName': '', 'useraccountcontrol': ''},  'vmware': {'delete': '', 'power\_state': ''}}, 'id': 1, 'ip': ['192.168.1.2'], 'name': 'hbf-client-01', 'owners': None, 'products': None, 'role': 'hbf', 'service': 'client1', 'sources': None, 'tags': None, 'txt\_records': None} |

# Установка клиентской части

Описанные в этом разделе действия необходимо выполнить на каждом сервере, который подключается к QHBF API.

1. Распакуйте архив hbf-client.tar.gz:

|  |
| --- |
| tar zxvf hbf-client.tar.gz |

1. Скопируйте скрипт:

|  |
| --- |
| cp hbf-client/hbf-apply.sh /usr/local/bin/hbf-apply.sh |

1. Отредактируйте в скрипте строчки:

|  |
| --- |
| service=client1role=hbfenvironment=testing |

1. Измените значения service, role, environment в соответствии с параметрами сервера:

|  |
| --- |
| config\_url=http://79.171.174.26/api/hbf/iptables-rules?service=${service}&role=${role}&environment=${environment}&fqdn=${fqdn}&hash=${old\_hash} |

1. Поменяйте url - пропишите ip или DNS имя сервера, предоставляющего QHBF API.
2. Добавьте авторизационные данные от basic auth QHBF API
3. В файл /root/.netrc добавьте авторизационные данные доступа к QHBF API в формате:

|  |
| --- |
| machine qhbf.domain.comlogin hbfpassword password |

1. Настройте запуск скрипта по расписанию через cron: в файл /var/spool/cron/crontabs/root добавьте строчку:

|  |
| --- |
| \*/5 \* \* \* \* /bin/bash -c "flock -n /var/lock/hbf.lock -c \\"sleep \\$((RANDOM \\% 200)); /usr/local/bin/hbf-apply.sh > /dev/null 2>&1\\"" |

1. Создайте директорию для логов:

|  |
| --- |
| mkdir -p /var/log/hbf/ |

Установка завершена.