**Инструкция по эксплуатации**

**ПО “BOTBLOCK”**

**Содержание**

[Описание и назначение ПО 3](#_heading=h.7pdtywqhs7do)

[Структура 3](#_heading=h.3i0sacogs75z)

[Установка и настройка 4](#_heading=h.3znysh7)

[1. Системные требования 4](#_heading=h.2et92p0)

[2. Установка и конфигурация Nginx 4](#_heading=h.tyjcwt)

[3. Создание службы запуска модуля 5](#_heading=h.3dy6vkm)

[4. Конфигурация модуля 5](#_heading=h.1t3h5sf)

[5. Логи блокировки запроса 7](#_heading=h.4d34og8)

[Функционал ПО 8](#_heading=h.2s8eyo1)

[1. Запуск модуля 8](#_heading=h.17dp8vu)

[2. Формат логов 8](#_heading=h.3rdcrjn)

[3. Алгоритмы блокировки и их настройка 8](#_heading=h.avm5vbbwh8hk)

[3.1 Блокировка IP-адреса с помощью Правил 8](#_heading=h.lnxbz9)

[3.2 Алгоритм формирования списка блокировки IP-адресов 8](#_heading=h.35nkun2)

[3.3 Настройка Правила 9](#_heading=h.1ksv4uv)

[3.4 Доступные зоны и примеры 12](#_heading=h.f7c1jd5ff4d)

[3.5 Алгоритм рейтлимитов 13](#_heading=h.44sinio)

[3.6 Настройка рейтлимитов 13](#_heading=h.2jxsxqh)

[4. Исключение IP-адреса из списка блокировки 16](#_heading=h.z337ya)

[5. Очистка списка блокировки IP-адресов 17](#_heading=h.3j2qqm3)

[Информация для контактов 17](#_heading=h.hf6fyok3mo3h)

# Описание и назначение ПО

ПО “BOTBLOCK” – программный продукт, который обеспечивает возможность формирования политик безопасности для создания репутационных списков IP-адресов на основе перечня доступных ограничений:

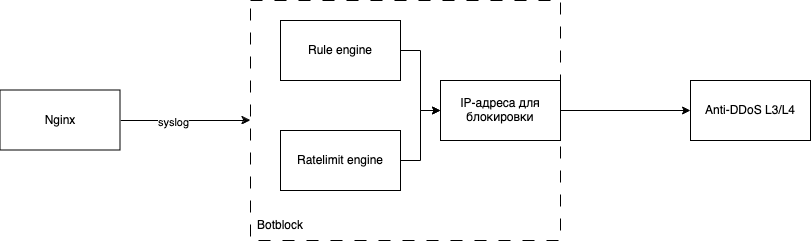
* ограничение по совпадению части анализируемого запроса с регулярным выражением;
* превышение разрешенного количества запросов в единицу времени.

ПО выполняет следующие функции:

* формирование правил фильтрации для анализируемых журналов запросов;
* формирование ограничения разрешённого количества запросов в минуту;
* сбор журналов запросов на анализ;
* применение правил и ограничений количества запросов на анализируемый журнал запросов;
* создание репутационного списка ip-адресов с временной меткой блокировки;
* очистка репутационного списка ручными средствами и автоматически по истечении времени блокировки.

# Структура

# 



# Установка и настройка

## 1. Системные требования

| **ОС** | Debian 11/12 |
| --- | --- |
| **Системный софт** | GCC 10 или выше |

## 2. Установка и конфигурация Nginx

Установить веб-сервер Nginx на отдельном сервере. В конфигурационный файл Nginx в блок http добавить следующие строки:

log\_format json\_combined escape=json

'{'

'"timestamp":"$time\_iso8601",'

'"host":"$host",'

'"scheme":"$scheme",'

'"port":"$server\_port",'

'"node":"$hostname",'

'"remote\_addr":"$remote\_addr",'

'"remote\_user":"$remote\_user",'

'"request":"$request\_uri",'

'"status": "$status",'

'"method": "$request\_method",'

'"body\_bytes\_sent":"$body\_bytes\_sent",'

'"http\_referrer":"$http\_referer",'

'"http\_user\_agent":"$http\_user\_agent",'

'"request\_time":"$request\_time",'

'"uaddr":"$upstream\_addr",'

'"uhost":"$proxy\_host",'

'"uct":"$upstream\_connect\_time",'

'"uht":"$upstream\_header\_time",'

'"urt":"$upstream\_response\_time",'

'"content\_type":"$sent\_http\_content\_type",'

'"request\_id":"$request\_id"'

'}';

access\_log syslog:**server=ip:514**,facility=local7,tag=nginx,severity=info json\_combined;

В директиве access\_log syslog в параметре server указать IP-адрес сервера, где будет установлен Модуль анализа логов, и порт 514.

Логи будут сохраняться в директории /home/debian/crpt\_access\_logs на сервере, указанном в access\_log syslog:server.

## 3. Создание службы запуска модуля

На сервере, на котором будет работать ПО “BOTBLOCK”, создать файл /etc/systemd/system/**botcheck.service** (служба для запуска Модуля анализа логов)и добавить в него следующие строки:

**[Unit]**

Description=Bot checker

After=network.target

**[Service]**

User=debian

Group=debian

ExecStart=<путь до файла crptab\_api> --conf /home/debian/crpt /config.ini

Restart=on-failure

WorkingDirectory=/home/debian

AmbientCapabilities=CAP\_NET\_BIND\_SERVICE

**[Install]**

WantedBy=multi-user.target

где:

User=debian – пользователь, от которого будет запущен сервис с правами на чтение и запись,

Group=debian – группа с правами на чтение и запись,

параметр /home/debian/crpt/config.ini – конфигурационный файл модуля.

## 4. Конфигурация модуля

В конфигурационном файле ПО “BOTBLOCK”

/home/debian/crpt/**config.ini** следует указать путь к файлам **rules.json, limits.json**:

**[DEFAULT]**

log\_level = ERROR

log\_folder = /home/debian/app\_logs

workers\_count = 4

**[LogSaver]**

path = /home/debian/crpt\_access\_logs

buffer\_size = 33000000

**[Rules]**

rules\_file = **rules.json**

default\_windows\_size = 1200

default\_temporary\_ban\_time = 30

temporary\_ban\_threshold = 2

permanent\_ban\_threshold = 5

default\_shift\_window = 1

**[Limits]**

limits\_file = **limits.json**

default\_ban\_time = 10

**[HttpServer]**

listen\_address = 0.0.0.0

listen\_port = 8080

temporary\_ban\_path = /temporary.txt

**[Syslog]**

listen\_address = 0.0.0.0

listen\_port = 514

где:

log\_level – уровень логирования; параметр может принимать значения: ERROR, WARNING, INFO; значение по умолчанию ERROR; в логах указывается тип сообщения, например, предупреждение: «LOG(warning) << “All ips has been unbaned”»; типы сообщений:

ERROR – сообщение об ошибке,

WARNING – предупреждение,

INFO – информационное сообщение;

log\_folder – файл логов приложения (с указанием абсолютного пути);

workers\_count – количество потоков > 1;

path – абсолютный путь до папки с логами;

buffer\_size – размер буфера с логами, при достижении которого логи сохраняются на диск; параметр задается в байтах; значение по умолчанию 67 108 864 байт = 64 × 1024 × 1024 байт;

rules\_file – файл с описанием Правил (с указанием абсолютного пути);

default\_windows\_size – время окна учета срабатываний Правил в секундах;

default\_temporary\_ban\_time – время временной блокировки по умолчанию в секундах;

temporary\_ban\_threshold – количество срабатываний для временной блокировки по умолчанию;

permanent\_ban\_threshold – количество срабатываний для постоянной блокировки по умолчанию; постоянной считается блокировка на 30 суток;

limits\_file – файл с описанием лимитов (с указанием абсолютного пути);

default\_ban\_time – время блокировки в секундах в случае нарушения одного из указанных лимитов;

default\_shift\_window – выполнять ли сдвиг окна после очередного срабатывания правил (1 – true, 0 – false);

listen\_address – адрес, на котором будет выполняться прием данных Syslog;

listen\_port – порт, на котором будет выполняться прием данных Syslog;

listen\_address – адрес, на котором будет выполняться прием HTTP-запросов;

listen\_port – порт, на котором будет выполняться прием HTTP-запросов;

temporary\_ban\_path – URI, на котором будет осуществляться прием запросов на текущий список заблокированных адресов;

## 5. Логи блокировки запроса

Логи модуля, которые помогут определить причину блокировки запроса, можно найти в папке /home/debian/app\_logs.

# Функционал ПО

## 1. Запуск модуля

Включить службу botcheck.service с помощью команды:

sudo systemctl enable botcheck.service

Запустить службу botcheck.service с помощью команды:

sudo systemctl start botcheck.service

Затем проверить запуск службы с помощью команды:

sudo systemctl status botcheck.service

## 2. Формат логов

Пример записи лога в лог-файле в формате JSON:

{"timestamp":"2024-01-24T14:52:18+03:00","host":"example.com","scheme":"http","port":"80","node":"vm-waf1","remote\_addr":"77.51.202.146","remote\_user":"","request":"/test.php?s=1%22%29%20AND%20%28SELECND%20%28%22kcp","status": "403","method": "GET","body\_bytes\_sent":"3548","http\_referrer":"http://example.com/test.php","http\_user\_agent":"sqlmap/1.7.7#stable (https://sqlmap.org)","request\_time":"0.000","uaddr":"","uhost":"","uct":"","uht":"","urt":"","content\_type":"text/html","request\_id":"7c013fc21b46a28a703ff7410483d704"}

где example.com – защищаемый ресурс.

## 3. Алгоритмы блокировки и их настройка

### 3.1 Блокировка IP-адреса с помощью Правил

Обнаружение вредоносных запросов осуществляется путем сигнатурного анализа трафика с помощью механизма задания Правил.

На основе заданных Правил формируется список IP-адресов, подлежащих блокировке. Список доступен при обращении к эндпоинту /temporary.txt.

### 3.2 Алгоритм формирования списка блокировки IP-адресов

Алгоритм формирования списка блокировки IP-адресов опирается на переменные конфигурационного файла config.ini.

Список IP-адресов, подлежащих блокировке, хранит IP-адреса в течение времени, заданного переменной default\_temporary\_ban\_time. По истечению данного времени адрес удаляется из списка.

Анализ IP-адреса производится в течении временного окна, задаваемого переменной default\_windows\_size. Временное окно конфигурируется для каждого Правила отдельно. По истечении временного окна счетчик срабатываний для IP-адреса обнуляется. Временное окно начинает отсчитываться с момента первого срабатывания Правила.

Задаются пороги срабатывания определенного числа Правил для любой записи, принадлежащей IP-адресу. В случае срабатывания паттерна количество раз, заданное в переменной temporary\_ban\_threshold, происходит добавление IP-адреса источника запроса в файл, задаваемый через переменную temporary\_ban\_path, на время, заданное переменной.

В случае срабатывания паттерна количество раз, заданное в переменной permanent\_ban\_threshold происходит добавление IP-адреса источника запроса в файл, задаваемый через переменную temporary\_ban\_path конфигурационного файла config.ini на 30 суток.

### 3.3 Настройка Правила

Каждое Правило представляет из себя JSON-объект файла rules.json, состоящий из следующих полей:

{

"zone": "host",

"pattern": "127.0.0.1"

"temporary\_ban": 2,

"temporary\_ban\_time": 200,

"permanent\_ban": 6,

"window\_size": 500,

"shift\_window": True

}

где:

zone – название зоны из записи лога, например: host, request, status, method (доступные зоны и примеры приведены ниже).

pattern – регулярное выражение, по вхождению которого определяется срабатывание Правила,

temporary\_ban – количество срабатываний для временной блокировки, если не указано, то используется значение temporary\_ban\_threshold из конфигурационного файла config.ini,

temporary\_ban\_time – время временной блокировки в секундах, если не указано, то используется значение default\_temporary\_ban\_time из конфигурационного файла config.ini,

permanent\_ban – количество срабатываний для постоянной блокировки, если не указано, то используется значение permanent\_ban\_threshold из конфигурационного файла config.ini,

window\_size – размер учетного окна в секундах, если не указано, то используется значение default\_windows\_size из конфигурационного файла config.ini.

shift\_window – если истина, то учетное окно будет сдвигаться после каждого сработавшего правила. Если параметр не указан, то используется значение default\_shift\_window из конфигурационного файла config.ini.

*Пример 1.* Если настроено следующее правило:

{

"zone": "request",

"pattern": "not\_allowed"

"temporary\_ban": 2,

"temporary\_ban\_time": 200,

"permanent\_ban": 6,

"window\_size": 500,

"shift\_window": false

}

тогда:

при переходе по ссылке, которая содержит в себе текст "not\_allowed" два раза с интервалом менее 500 секунд, IP-адрес будет заблокирован на 200 секунд;

при переходе по ссылке, которая содержит в себе текст "not\_allowed" шесть раз с интервалом менее 500 секунд между первым и последним запросом, IP-адрес будет заблокирован на 30 суток.

*Конец примера.*

*Пример 2.* Если настроено следующее правило:

{

"zone": "request",

"pattern": "not\_allowed"

}

и config.ini с заданными директивами:

[Rules]

rules\_file = rules.json

workers\_count = 4

default\_windows\_size = 1200

default\_temporary\_ban\_time = 600

temporary\_ban\_threshold = 3

permanent\_ban\_threshold = 5

default\_shift\_window = 1

тогда:

при переходе по ссылке, которая содержит в себе текст "not\_allowed" три раза с интервалом менее 1200 секунд, IP-адрес будет заблокирован на 600 секунд.

при переходе по ссылке, которая содержит в себе текст "not\_allowed" пять раз с интервалом менее 1200 секунд между первым и последним запросом, IP-адрес будет заблокирован на 30 суток.

*Конец примера.*

*Пример 3.* Если настроено два правила, первое:

{

"zone": "request",

"pattern": "not\_allowed1"

"temporary\_ban": 2,

"temporary\_ban\_time": 200,

"permanent\_ban": 6,

"window\_size": 500,

"shift\_window": true

},

и второе правило:

{

"zone": "request",

"pattern": "not\_allowed2"

"temporary\_ban": 2,

"temporary\_ban\_time": 150,

"permanent\_ban": 6,

"window\_size": 400,

"shift\_window": true

},

тогда в случае 1:

1. `http://site.com/abc/not\_allowed1/def`, секунда 1

2. `http://site.com/abc/not\_allowed2/def`, секунда 20

3. `http://site.com/abc/not\_allowed2/def`, секунда 380 – блокировка на 150 секунд

4. `http://site.com/abc/not\_allowed1/def`, секунда 460 – блокировка на 200 секунд

5. `......................`, секунда 661 – разблокировка

при блокировке уже заблокированного адреса, время блокировки обновляется только, если новая блокировка больше, чем осталось секунд по текущей блокировке;

в случае 2:

1. `http://site.com/abc/not\_allowed1/def`, секунда 1

2. `http://site.com/abc/not\_allowed2/def`, секунда 20

3. `http://site.com/abc/not\_allowed1/def`, секунда 390 – блокировка на 200 секунд

4. `http://site.com/abc/not\_allowed2/def`, секунда 410 – блокировка на 150 секунд

5. `....................`, секунда 591 – разблокировка

в данном случае вторая блокировка на 150 секунд не перекрывает предыдущую на 200 секунд.

*Конец примера.*

### 3.4 Доступные зоны и примеры

request – URI запроса, например:

/test.php?s=1%29%20AND%20%28SELECT%20%28CASE%20WHEN%20

status – HTTP-статус строкой, например: 403

method – метод запроса, например: GET

http\_referrer – страница, с которой был выполнен запрос, например:

<http://example.com/test.php>

http\_user\_agent – User Agent, например:

sqlmap/1.7.7#stable (https://sqlmap.org)

request\_time – время выполнения запроса строкой, например: 0.021

content\_type – тип контента ответа, например: text/html

### 3.5 Алгоритм рейтлимитов

Рейтлимиты позволяют задать конечное число допустимых обращений IP-адреса, по истечению которых IP-адрес попадает в список заблокированных.

Рейтлимиты могут быть установлены на отдельные локейшены. Локейшены возможно задать с помощью регулярных выражений в переменной loc.

Рейтлимиты позволяют задать разрешенное количество кратковременного превышения лимита в переменной allowed\_burst.

Количество обращений для IP-адреса лимитируется в рамках временного окна, задаваемого параметром ban\_time. Если в переменной время не указано, используется общее время блокировки IP-адреса, задаваемое переменной default\_ban\_time в конфигурационном файле config.ini. По истечению временного окна счетчик для IP-адреса обнуляется.

### 3.6 Настройка рейтлимитов

Рейтлимиты представляют собой список JSON-объектов файла limits.json, состоящих из следующих полей:

[

{

"loc": "/location1\\.php",

"allowed\_burst": 10,

"requests\_per\_minute": 120,

"ban\_time": 200

},

{

"loc": "/location1\\.html",

"allowed\_burst": 5,

"requests\_per\_minute": 120,

"ban\_time": 300

},

{

"loc": "\\S+",

"requests\_per\_minute": 60

}

]

где:

loc – адрес, который с помощью регулярного выражения задает локейшен. Если ссылка, по которой перешел клиент, подходит под несколько локейшенов, то будет использован первый попавшийся лимит;

allowed\_burst – разрешенное количество кратковременного превышения лимита, при этом кратковременный лимит запросов allowed\_burst + 1. Значение по умолчанию: 0. Если requests\_per\_minute > 59, то настоятельно рекомендуется указать allowed\_burst > 1;

requests\_per\_minute – количество запросов в минуту. В работе используется интервал между двумя запросами. Например, если значение равно 30, то интервал должен быть не менее 2 секунд;

ban\_time – время блокировки в секундах после превышения заданного лимита. Если не указано, то будет использовано значение default\_ban\_time из конфигурационного файла config.ini.

*Пример 1*. Настроен рейтлимит:

{

"loc": "/script\\.php",

"allowed\_burst": 3,

"requests\_per\_minute": 60,

"ban\_time": 200

}

Директива "allowed\_burst": 3 означает, что разрешено 4 запроса в секунду.

Тогда:

разрешено делать запросы на `/script.php` только один раз в секунду;

при этом разрешено сделать сразу 4 запроса без блокировки, но 5ый запрос можно будет сделать только через 5 секунд после 4го.

Случай 1:

1. `http://site.com/script.php`, секунда 1

2. `http://site.com/script.php`, секунда 1

3. `http://site.com/script.php`, секунда 1

4. `http://site.com/script.php`, секунда 1

5. `http://site.com/script.php`, секунда 2 – блокировка на 200 секунд

Случай 2:

1. `http://site.com/script.php`, секунда 1

2. `http://site.com/script.php`, секунда 1

3. `http://site.com/script.php`, секунда 1

4. `http://site.com/script.php`, секунда 2

5. `http://site.com/script.php`, секунда 2

6. `http://site.com/script.php`, секунда 5 – нет блокировки

*Конец примера.*

*Пример 2*. Настроен рейтлимит:

{

"loc": "/script\\.php",

"allowed\_burst": 10,

"requests\_per\_minute": 120,

"ban\_time": 200

}

тогда:

разрешено делать запросы на `/script.php` только два раза в секунду;

при этом разрешено сделать сразу 11 запросов без блокировки, но 12ый запрос можно будет сделать только через 5 секунд после 11го.

*Конец примера.*

*Пример 3*. Настроено два рейтлимита. Рейтлимит первый:

{

"loc": "/script.\\.php",

"allowed\_burst": 10,

"requests\_per\_minute": 120,

"ban\_time": 200

},

и второй рейтлимит:

{

"loc": "/script1\\.php",

"allowed\_burst": 10,

"requests\_per\_minute": 100,

"ban\_time": 700

}

тогда:

при переходе по ссылке `/script1.php` всегда будет работать первый рейтлимит, а второй никогда не сработает.

*Конец примера.*

*Пример 4*. Рейтлимит:

{

"loc": "/script\\.php",

"requests\_per\_minute": 30,

"ban\_time": 200

}

тогда:

разрешено делать запросы на `/script.php` только один раз в две секунды;

при первом нарушении этого условия IP-адрес будет заблокирован на 200 секунд.

*Конец примера.*

## 4. Исключение IP-адреса из списка блокировки

Удаление IP-адресов из списка блокировки осуществляется через эндпоинт /unban.

С помощью необязательного параметра interval можно задать интервал в секундах. Из списка блокировки будут удалены все IP-адреса, блокировка которых истекает раньше указанного интервала времени.

Если параметр interval не указан, то он считается равным количеству секунд в 45 сутках.

Если указан параметр ip, то будет выполнена разблокировка указанного адреса. Если адреса не существует в списке блокировок, то будет выполнена очистка его состояния. Если состояния нет, то ничего не будет выполнено. Если указан параметр ip, то параметр interval игнорируется. Корректность указанного адреса не проверяется, если строка не является корректным адресом, то его не будет в таблице состояний и никакой ошибки не возникнет.

*Примеры.*

1. Пример разблокировки всех заблокированных:

/unban

1. Пример разблокировки IP-адресов, у которые оставшееся время блокировки менее 300 секунд:

/unban?interval=300

1. Пример разблокировки IP-адресов, у которые оставшееся время блокировки менее суток:

/unban?interval=86400

1. Пример разблокировки IPv4-адреса:

/unban?ip=1.2.3.4

1. Пример разблокировки IPv6-адреса:

/unban?ip=2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d

## 5. Очистка списка блокировки IP-адресов

Полная очистка состояния ПО “BOTBLOCK” осуществляется через эндпоинт /clear\_all.

# Информация для контактов

Для контактов с командой разработчиков просьба обращаться по следующим контактам:

E-mail: support@weblock.ru