ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«Preventive Proxy»

Версия 1.0

РУКОВОДСТВО АДМИНИСТРАТОРА

СОДЕРЖАНИЕ

Аннотация

Настоящий документ содержит руководство администратора программного обеспечения «Preventive Proxy» версии 1.0 (далее – ПО).

Назначение ПО

Preventive Proxy - инновационное решение для выявления и противодействия вредоносной бот-активности в режиме реального времени при использовании веб-ресурсов и мобильных приложений.

Принцип выявления вредоносной или бот-активности основан на работе модулей, отвечающих за анализ сетевых запросов, поведения пользователя и окружения, в котором работает мобильное или веб-приложение.

Модуль Preventive Proxy - подсистема Fraud Hunting Platform, которую можно использовать отдельно или с другими подсистемами продукта, комплексно защищая сайт/приложение и его пользователей от мошенничества.

Внедрение ПО

Для внедрения ПО в инфраструктуру приложения понадобится:

- схема сети и архитектуры защищаемого приложения;
- необходимые доступы к подсистемам;
- информация о дата-центрах (локации);
- контактные данные ответственных сотрудников.

На основе полученной информации специалисты Group-IB смогут предоставить рекомендации по внедрению и настройке ПО, а также скоординировать внедрение и запуск ПО. Эти данные можно передать специалистам Group-IB в любом удобном формате.

Обновление ПО

Модуль Preventive Proxy непрерывно совершенствуется, добавляются новые технологические возможности. Новые версии ПО передаются Заказчику в оговоренные сроки.

При получении экземпляра с новой версией ПО администратор инфраструктуры Заказчика должен установить и настроить новую версию самостоятельно. У ПО отсутствует возможность принудительного обновления не из инфраструктуры Заказчика.

Пример установки ПО

Обработка входящего трафика в приложении работает следующим образом:

1. Весь трафик от фронтенда приходит на веб-сервер (например, nginx).

2. Веб-сервер расшифровывает и отправляет трафик на бэкэнд - сервера, скрипты, базы данных.

Классическая схема внедрения Preventive Proxy предполагает его установку на точку обработки трафика (на веб-сервер) в инфраструктуре клиента. Со стороны клиента сотрудник с правами администратора сможет настраивать, перезапускать и при необходимости отключать Preventive Proxy.

Такая схема установки Preventive Proxy называется установкой "в петлю" работает следующим образом:

1. Весь трафик от фронтенда идет на веб-сервер (чаще всего - nginx).
2. Веб-сервер отправляет трафик в Preventive Proxy, который проверяет и размечает трафик.
3. От Preventive Proxy размеченный трафик идет обратно на веб-сервер.
4. Веб-сервер блокирует бот-запросы (при работе в режиме блокировки), а легитимные запросы пользователей идут на бэкэнд мобильного или веб-приложения - сервера, скрипты, базы данных.

В некоторых случаях перед основным веб-сервером стоит еще один веб-сервер, выполняющий роль балансировщика (как правило, тоже nginx). В таком случае возможны варианты:

1. Preventive Proxy устанавливается "в петлю" на балансировщике.
2. Preventive Proxy устанавливается между балансировщиком и веб-сервером.
3. Preventive Proxy устанавливается "в петлю" на обоих веб-серверах.

Обработка трафика

Обрабатывать входящий трафик можно через:

- проксирование запросов через Preventive Proxy;
- разметку запросов с помощью auth-request в nginx.

Существует два основных вида запросов: на динамический и на статический контент. Обрабатывать запросы на статический контент через Preventive Proxy не эффективно. Такие запросы можно обрабатывать напрямую через веб-сервер, а динамические запросы, которые загружают серверную часть приложения, фильтровать через Preventive Proxy.

После установки модуля Preventive Proxy необходимо выбрать схему обработки запросов и настроить выбранный прокси-модуль (Preventive Proxy или nginx). Специалисты Group-IB могут предоставить примерный файл конфигурации, который будет адаптирован к инфраструктуре клиента.

Технические требования модуля Preventive Proxy

Для усредненной нагрузки в 10 тыс. запросов/сек (до 200 Мб/сек) необходимо выделить два физических сервера. Минимальные ресурсы сервера:

- CPU 6 ядер, 3Ггц;

- RAM 16 ГБ;
- SSD 100 ГБ.

Чтобы минимизировать время обработки запросов к приложению, Preventive Proxy можно настроить для проверки запросов только на динамический контент, а запросы на статический контент перенаправить через прокси-сервер в инфраструктуре защищаемого приложения.

При установке в инфраструктуре Заказчика, сервера для Preventive Proxy рекомендуется разместить в дата-центре, где находится инфраструктура защищаемого приложения, или в одной подсети с инфраструктурой защищаемого приложения. Это позволит обращаться к Preventive Proxy без преобразования сетевых адресов (NAT) и сократить задержки между запросом и ответом.

Проксирование запросов через Preventive Proxy

Для работы модуля Preventive Proxy нужны два файла конфигурации - статический и динамический.

В статическом файле конфигурации указываются параметры запуска и работы Preventive Proxy. Этот файл конфигурации формируется специалистами Group-IB с учетом особенностей инфраструктуры заказчика и предпочтений по обработке трафика. Статический файл конфигурации обычно хранится в одной локации с файлом анти-бот модуля Preventive Proxy и инициируется командой:

```
/proxy -cfg config.yaml
```

В динамическом файле конфигурации указываются параметры загрузки и обработки трафика. На основе этих параметров Preventive Proxy анализирует запросы пользователей мобильного или веб-приложения, обрабатывает значения токенов и принимает решения о наличии бот-активности. Динамический файл конфигурации хранится на сервере и формируется из настроек модуля Preventive Proxy в панели администратора Fraud Hunting Platform, путь к нему и интервал обновления указываются в статическом файле конфигурации.

Пример статического файла конфигурации

Статический файл конфигурации может выглядеть следующим образом:

```
version: v4.2

customer-hash: TestCustomer

mode: proxy

config-api:
http://127.0.0.1:3232/settings?key=SampleKey&customer_hash=TestCusto
mer
config-update: 2s
```

```
check-api:
http://127.0.0.1:3232/checkgssc?customer_hash=TestCustomer

listen:
  addr: 127.0.0.1:3233
  tls:  # default tls is disabled
    cert: /path/to/certfile
    key: /path/to/keyfile

proxy:
  addr: 127.0.0.1:3232
  host: example.com  # replace request 'host' header
  scheme: https
  ignore-ssl: true

whitepage-html: /path/to/html  # with {{INJECT}} macros
js-url: /js/sb.js
js-autoinject: true

log: https://sbbe.group-ib.ru/api/fl/proxy-logs  #
stdout|stderr|path/to/file
log-auth-token: G2u634h9hing62220Thoosh111ee9aiz
log-version: v2

alert-rules:
  41:
    reason: bad bots
    desc: "bad bot activity"

  398:
    reason: unwanted activity
    desc: "unwanted activity"
```

Параметры статического файла конфигурации

Для настройки проксирования трафика через Preventive Proxy необходимо указать следующие параметры:

Параметр	**Описание**
version	Версия модуля Preventive Proxy. `version: 4.2`

customer-hash	Идентификатор заказчика. Предоставляется специалистами Group-IB по запросу. `customer-hash: TestCustomer`
mode	Режим работы прокси-сервера. Возможные режимы: ● proxy - передача и проверка входящего трафика в Preventive Proxy, блокировка подозрительных запросов (режим по умолчанию); ● auth - только проверка входящего трафика. `mode: proxy`
config-api	Путь до второго (динамического) файла конфигурации, который обновляется через панель администратора Fraud Hunting Proxy. Необходимо указывать query-параметры customer_hash и key для идентификации заказчика. `config-api: http://127.0.0.1:3232/settings?key=SampleKey&customer_hash=TestCustomer`
config-update	Интервал обновления динамического файла конфигурации. Можно указать в милисекундах (ms), секундах (s) или минутах (m). `config-update: 2s`
check-api	Адрес для проверки корректности работы модуля Preventive Proxy с Fraud Hunting Platform. Необходимо указывать query-параметр customer_hash для идентификации заказчика. `check-api: http://127.0.0.1:3232/checkgssc?customer_hash=TestCustomer`

listen	Адрес и порт (addr), на котором модуль Preventive Proxy слушает входящие запросы.
	Протокол шифрования трафика tls по умолчанию не задан (отключен). Если выбрать протокол tls, понадобится указать путь до файла сертификата (cert) и ключа (key) шифрования.
	
	`listen:`
	`  addr: 127.0.0.1:3233`
	`  tls:  # default tls is disabled`
	`  cert: /path/to/certfile`
	`  key: /path/to/keyfile`

proxy	Адрес бэкенд-сервера сайта, на который Preventive Proxy будет отправлять очищенный трафик. Не нужно указывать, если работает в режиме auth.
	В этом параметре описывается, куда (addr) и как (scheme, host, ignore-ssl) передавать трафик.
	
	`proxy:`
	`  addr: 127.0.0.1:3232`
	`  host: example.com  # replace request 'host' header`
	`  scheme: https`
	`  ignore-ssl: true`

whitepage-html	Проверочная страница, которая может быть кастомизирована. Необходимо указать путь до нее и разместить на ней инжект-макрос. Это необязательный параметр: если его не указать, будет выдаваться пустая страница со скриптом Web Snippet. `whitepage-html: /path/to/html  # with {{INJECT}} macros`
js-url	Путь до работающего модуля Web Snippet на Fraud Hunting Platform. `js-url: /js/sb.js`
js-autoinject	Параметр для автоматического инжектирования скрипта в html-документы, проходящие через Preventive Proxy. `js-autoinject: true`

log	Параметры логгирования.
	Возможные значения:
	● stdout - вывод в стандартный поток stdout (задан по умолчанию);
	● stderr - вывод в стандартный поток stderr;
	● path/to/file - вывод в файл (путь для хранения логов в файле);
	● http://user:password@my.logstach.com/path - вывод в http-плагин logstash (учетная запись и хост).
	
	`# stdout|stderr|path/to/file|http://user:password@my.logstach.com/path`
	`log: /dev/null`

log-auth-token	Токен для аутентификации в системе логгирования. `log-auth-token: SampleAuthToken`

log-version	Версия логов. `log-version: v2`

alert-rules	Алерты, при срабатывании которых Preventive Proxy будет блокировать запросы.
	
	Параметры:
	
	● reason - причина, по которой запрос блокируется. Это значение будет передано в заголовок X-Gib-Reason и логи Preventive Proxy. Значение по умолчанию - "unwanted activity";
	● desc - причина блокировки, которую увидит пользователь.
	
	`alert-rules:`
	`  41:`
	`  reason: bad bots`
	`  desc: "bad bot activity"`
	
	`  398:`
	`  reason: unwanted activity`
	`  desc: "unwanted activity"`

Пример динамического файла конфигурации

Динамический файл конфигурации может выглядеть следующим образом:

```
{
    "block": true,
    "gssc-lifetime": 2m,
    "gssc-repeat-count": 50,
    "fgssc-repeat-count": 2,
    "expired-repeat-count": 5,
    "expired-cache-ttl": 1h,
    "static": {
      "sec-fetch-dest": false,
      "default-url-extensions": true,
      "location": [
        "^/catalog.xml",
        "^/style/",
        "^/javascript/"
      ]
    },
    "whitelist": {
      "location": [
        "/private/mindbox/28/",
        "/private/yandex/"
```

```
      ],
      "ip": [
        "5.45.211.60/32",
        "62.105.143.45/32"
      ]
    },
    "strict": {
      "location": [],
      "ip": []
    },
    "gssc-key": SampleGSSCKey
}
```

Параметры динамического файла конфигурации

В динамическом файле конфигурации Preventive Proxy указываются следующие параметры:

Параметр	**Описание**
block	Определяет режим обработки трафика. Возможные значения: ● true - для режима блокировки; ● false - для режима маркировки, смешанного и зеркального режимов. `"block": true,`
gssc-lifetime	Срок жизни gssc-токена. Можно указать в милисекундах (ms), секундах (s) или минутах (m). `"gssc-lifetime": 2m,`
gssc-repeat-count	Максимальное число повторов gssc-токена за период его жизни (gssc-lifetime). `"gssc-repeat-count": 50,`
fgssc-repeat-count	Максимальное число использований одноразового fgssc-токена, в большинстве случаев должно быть равно 1. `"fgssc-repeat-count": 2,`
expired-repeat-count	Максимальное число использований просроченных токенов - используется для ослабления политики в мобильных приложениях, которые останавливают работу браузера при сворачивании. Для API адресов должно быть равно 0. `"expired-repeat-count": 5,`

expired-cache-ttl	Срок хранения информации об использованных токенах. Можно указать в милисекундах (ms), секундах (s), минутах (m) или часах (h). `"expired-cache-ttl": 1h,`

static	Правила определения и параметры обработки статического контента, который проксируется без проверки токенов.
	В этом параметре включается механизм определения статического контента на базе заголовка webkit и заголовка content-type ответа (sec-fetch-dest), указывается возможность использовать стандартный набор расширений файлов (default-url-extensions), и задается набор правил (регулярных выражений) по URL ресурсов для определения статического контента (location).
	
	`"static": {`
	`  "sec-fetch-dest": false,`
	`  "default-url-extensions": true,`
	`  "location": [`
	`  "^/catalog.xml",`
	`  "^/style/",`
	`  "^/javascript/"`
	`  ]`
	`},`

whitelist	Параметры обработки доверенных запросов.
	Набор правил (регулярных выражений) для определения доверенных запросов задается с помощью параметров location и ip.
	
	`"whitelist": {`
	`  "location": [`
	`  "/private/mindbox/28/",`
	`  "/private/yandex/"`
	`  ],`
	`  "ip": [`
	`  "5.45.211.60/32",`
	`  "62.105.143.45/32"`
	`  ]`
	`},`

strict	Фильтр запросов, для которых будет применяться режим блокировки, даже в режиме маркировки (при "block": false). Не указывается для режима блокировки (при "block": true).
	Набор правил (регулярных выражений) для определения блокируемых запросов задается с помощью параметров locations и ip.
	
	`"strict": {`
	`  "location": [],`
	`  "ip": []`
	`},`

gssc-key	Ключ для расшифровки токена gssc. Предоставляется специалистами Group-IB по запросу. `"gssc-key": SampleGSSCKey`

Проксирование запросов с использованием модуля auth_request в nginx

Через модуль auth_request в nginx можно настроить обработку трафика для режима маркировки. Для этого нужна версия nginx не ниже 1.5.4.

При реализации такой схемы обработки трафика можно настроить, для каких пользовательских запросов nginx запрашивает вердикт у Fraud Hunting Platform - например, можно отфильтровывать запросы на выдачу статического контента или запросы из доверенных учетных записей для оптимизации загрузки Preventive Proxy. В зависимости от полученного вердикта, nginx пропускает запрос на бэкэнд инфраструктуры клиента или помечает его как бот-активность.

Для пользовательских запросов можно настроить:

- таймауты - если проверка запроса занимает больше времени, чем обычно, он будет перенаправлен в обход модуля Preventive Proxy;
- фильтр передаваемой информации - например, для отсеивания запросов на передачу выбранных типов файлов (запросов на статический контент);
- фэйловер - механизм перенаправления запросов, например, когда заданное число запросов не дошло до Preventive Proxy, отправка запросов переключится на запасную ноду или пойдет в обход Preventive Proxy.

В работе модуля auth_request понадобится использовать заголовки:

- X-Real-IP - для добавления исходного (начального) адреса запроса;
- X-Original-Uri - для хранения адреса исходной точки запроса;
- X-Original-Host - для хранения адреса исходного сервера, обрабатывающего запрос.

Пример запроса и ответа через модуль auth_request:

```
curl -v https://sbproxy-auth.group-ib.com:10058/auth
...
X-Gib-Pass: false
X-Gib-Reason: valid X-Real-IP header required
…
```

Файл конфигурации формируется специалистами Group-IB с учетом особенностей инфраструктуры клиента и предпочтений по обработке трафика.

Пример файла конфигурации

Файл конфигурации для модуля auth_request в nginx может выглядеть следующим образом:

```
server {
  ...
  location = /test {
    ...
    auth_request /auth;
    auth_request_set $reason $upstream_http_x_gib_reason;
    proxy_set_header X-Gib-Reason $reason;
    ...
    }
  location = /auth {
    internal;
    proxy_connect_timeout 3s;
    proxy_send_timeout 15s;
    proxy_read_timeout 30s;
    proxy_pass https://<FHP Backend URL:port>/auth;
    proxy_pass_request_body off;
    proxy_set_header Content-Length "";
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Original-Uri $request_uri;
    proxy_set_header X-Original-Host $server_name;
    }
  ...
}
```

Настройки модуля auth_request

Для тестовой локации location = /test указываются параметры для проверки аутентификации пользовательских запросов:

Параметр	**Описание**
auth_request	Директива для передачи внутренней локации, куда будет перенаправлен запрос пользователя для последующей аутентификации. `auth_request /auth;`

auth_request_set	Директива для передачи информации о вердикте: ● $reason - в эту переменную передается вердикт о наличии бот-активности в пользовательском запросе. Используется для хранения и последующей передачи вердикта в заголовок X-Gib-Reason; ● $upstream_http_x_gib_reason - из этой переменной передается вердикт о наличии бот-активности в пользовательском запросе. `auth_request_set $reason $upstream_http_x_gib_reason;`
proxy_set_header	Директива для передачи вердикта о наличии бот-активности в пользовательском запросе в заголовок X-Gib-Reason. Вердикт передается из переменной $reason. `proxy_set_header X-Gib-Reason $reason;`

Для `location = /auth` указываются параметры для передачи пользовательских запросов:

Параметр	**Описание**
internal	Определение внутренней адресации. `internal;`
proxy_connect_timeout	Директива для таймаута подключения к прокси-серверу. Если время подключения превышает заданное значение, запросы будут пересылаться в обход прокси. `proxy_connect_timeout 3s;`
proxy_send_timeout	Директива для таймаута отправки сообщений на прокси-сервер. Если время на отправку сообщения превышает заданное значение, запросы будут пересылаться в обход прокси. `proxy_send_timeout 15s;`
proxy_read_timeout	Директива для таймаута прочтения сообщений прокси-сервером. Если время на прочтение сообщения превышает заданное значение, запросы будут пересылаться в обход прокси. `proxy_read_timeout 30s;`

proxy_pass	Директива для передачи подзапросов аутентификации службе аутентификации Fraud Hunting Platform. `proxy_pass https://<FHP Backend URL:port>/auth;`
proxy_pass_request_body	Директива для добавления тела исходного запроса при передаче подзапроса на аутентификацию. Возможные значения: ● on - тело исходного запроса будет включено в подзапрос на аутентификацию; ● off - тело исходного запроса не будет включено в подзапрос на аутентификацию. `proxy_pass_request_body off;`

proxy_set_header	Директива для передачи информации о пользовательском запросе в заголовки.
	Задаются значения заголовков:
	● Content-Length - длина передаваемого тела запроса. Если тело запроса не передается на аутентификацию, задается нулевая длина;
	● X-Real-IP - адрес конечной точки запроса (клиента) из переменной $remote_addr;
	● X-Original-Uri - URI исходной точки запроса целиком (с аргументами) из переменной $request_uri;
	● X-Original-Host - адрес исходного сервера, принявшего запрос, из переменной $server_name.
	
	`proxy_set_header Content-Length "";`
	`proxy_set_header X-Real-IP $remote_addr;`
	`proxy_set_header X-Original-Uri $request_uri;`
	`proxy_set_header X-Original-Host $server_name;`

Режимы обработки трафика

Preventive Proxy может работать в режимах блокировки, маркировки, смешанном или зеркальном режиме обработки запросов с последующей обработкой вердиктов на стороне веб-сервера/балансировщика или серверной части приложения.

Preventive Proxy добавляет к обрабатываемым запросам заголовки: X-Gib-Reason (результаты анализа - статус проверки) и X-Gib-Pass (рекомендации по блокировке или прохождению запроса). На основе значений этих заголовков модуль Preventive Proxy обрабатывает запросы.

Все запросы от веб-сервера/балансировщика отправляются на проверку модулем Preventive Proxy. По умолчанию, Preventive Proxy работает в режиме блокировки - все подозрительные запросы блокируются. Если нет необходимости блокировать запросы, специфика работы приложения не предполагает блокировку запросов или их надо

проверять на серверной стороне приложения, можно выбрать другой режим обработки запросов.

Настройки для выбранного режима обработки запросов указаны в файле конфигурации Preventive Proxy.

Режим блокировки

По умолчанию, модуль Preventive Proxy работает в режиме блокировки: все подозрительные запросы к бэкенду мобильного или веб-приложения, в которых найдены признаки бот-активности, будут заблокированы.

Если в запросе найдены признаки бот-активности, такой запрос блокируется или отправляется на дополнительную проверку (например, с помощью CAPTCHA).

Настройки режима блокировки:

```
block: true
```

Режим маркировки

В режиме маркировки запросы к бэкенду веб- или мобильного приложения пропускаются всегда, даже если в них есть признаки бот-активности.

После проверки анти-бот модулем Preventive Proxy, к каждому запросу добавляется информация, на основе которой можно настроить проверку запросов на серверной стороне приложения.

Настройки режима маркировки:

```
block: false
```

Смешанный режим

В смешанном режиме большинство запросов к бэкенду мобильного или веб-приложения пропускаются. Для отдельных запросов можно включить блокировку.

В смешанном режиме в файле конфигурации дополнительно описываются списки блокировки ("block: true") для:

- путей (locations) - блокируются запросы, которые обращаются к этим директориям;
- IP-адресов и подсетей - блокируются или отправляются на проверку запросы, которые пришли с этих адресов.

Пример настроек смешанного режима:

```
block: false

strict:
  location:
    - ^/strict/
  ip:
    - 8.8.8.8/32
```

Зеркальный режим

В зеркальном режиме обработки трафика запросы к бэкенду мобильного или веб-приложения пропускаются всегда. Запросы передаются в асинхронном режиме, без ожидания вердикта от Preventive Proxy.

Каждый запрос дублируется и копия заголовков проходит через Preventive Proxy. После проверки в Preventive Proxy, к запросу добавляется информация, на основе которой можно настроить проверку запросов на серверной стороне приложения и сбор статистики:

- заголовки X-Gib-Reason и X-Gib-Pass;
- файлы cookie gssc и fgssc.

Для обработки трафика в зеркальном режиме понадобится версия nginx не ниже 1.13.4.

Пример настроек зеркального режима:

```
location / {
    mirror /mirror;
    mirror_request_body off;
    ...
}
location = /mirror {
    internal;
    proxy_pass https://<group-ib_sb-proxy>:443;
    proxy_pass_request_body off;
    proxy_set_header Host <group-ib_sb-proxy>;
    proxy_set_header Content-Length "";
    proxy_set_header X-Original-URI $request_uri;
    proxy_set_header X-Real-IP $remote_addr;
}
```

Инструкция по запуску ПО

Для запуска ПО в ОС Linux выполните следующие действия:

1. Сохраните файл конфигурации config-ros.yaml и бинарный файл ПО sb-proxy в любую общую папку.
2. Откройте эту папку в терминале и введите команду:

   ```
   ./sb-proxy -cfg config-ros.yaml
   ```

   Если после выполнения данной команду вы увидели строку

   ```
   proxy mode started at :7000
   ```

   это значит, что ПО запущено в режиме проксирования и слушает входящие запросы на порту 7000. Входящие запросы проксируются на адрес

127.0.0.1:7001, его можно изменить, отредактировав файл конфигурации (поле addr) и перезапустив ПО.

3. Чтобы проверить работоспособность программы, используйте команду

   ```
   curl 127.0.0.1:7000
   ```

   Эта команда отправит входящий HTTP-запрос на порт 7000. Результат обработки запроса ПО отобразит в терминале (параметр stdout), в котором она запущена.