ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«F6 Preventive Proxy»

Описание функциональных характеристик

Содержание

ТЕРМИНЫ И СОКРАЩЕНИЯ

Термин	**Описание**
АС	Автоматизированная система АО «Будущее»
Бот	Программа, которая автоматически выполняет заранее настроенные повторяющиеся задачи. Боты могут использоваться как в «хороших» целях (например, виртуальные помощники), так и в «плохих» (спам-боты, вирусы, боты, производящие DoS- и DDoS-атаки)
Брутфорс	Метод взлома с помощью подбора данных для входа в систему. Чаще всего с помощью брутфорса злоумышленники угадывают пароли к учетным записям жертв
Заказчик	Лицо, которое использует на законных основаниях ПО на основании заключенного договора
Исполнитель	Работы Исполнителя на протяжении всего жизненного цикла могут исполняться: • АО «Будущее»; • Компанией-интегратором, по выбору Заказчика
ПО	Программное обеспечение «F6 Preventive Proxy»
Прокси-сервер	Сервер (комплекс программ) в компьютерных сетях, выполняющий роль посредника между пользователем и целевым сервером (при этом о посредничестве могут как знать, так и не знать обе стороны). Позволяет клиентам как выполнять косвенные запросы (принимая и передавая их через прокси-сервер) к другим сетевым службам, так и получать ответы
Разработчик	АО «Будущее»

Скрапинг	Метод получения данных с различных открытых веб-ресурсов, в основном автоматизированный, их анализ и извлечение нужной информации. Может быть использован в незаконных целях
Скриншот	Изображение, «снимок» экрана ПК или мобильного устройства, на котором запечатлено содержимое экрана устройства
СУБД	Система управления базами данных
ТС	Система взаимодействия Заказчика, позволяющая обмениваться сообщениями и создавать цепочки обращений, которая представляет из себя отдельный раздел «Службу Поддержки» в панели управления «F6 Fraud Protection». В случае недоступности указанных систем формат взаимодействия осуществляется через электронный почтовый ящик
Файл cookie	Файл с небольшим набором данных, которые веб-ресурс отправляет на компьютер пользователя для идентификации устройства
API (Application Programming Interface)	Программный интерфейс, то есть описание способов взаимодействия одной компьютерной программы с другими
DDoS-атака (Distributed Denial of Service)	Атака, целью которой является перегрузка сетевых ресурсов, делая их недоступными для их законных пользователей
Mobile SDK (далее – SDK)	Модуль программного обеспечения «F6 Fraud Protection» для встраивания в мобильные приложения
NAT (Network Address Translation)	Механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов

RSA	Криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших полупростых чисел
On-premise	Модель обслуживания, при которой данные и приложения размещаются и хранятся в инфраструктуре Пользователя
SaaS (Software as a Service)	Модель обслуживания, при которой программное обеспечение размещено в облачной инфраструктуре
SHA1	(Secure Hash Algorithm 1) — алгоритм криптографического хеширования
Web Snippet (далее – скрипт)	Модуль программного обеспечения «F6 Fraud Protection» для встраивания в WEB приложения

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Введение

Настоящий документ описывает функциональные характеристики программного обеспечения «F6 Preventive Proxy» (далее — ПО, F6 Preventive Proxy).

1.2 Назначение ПО

«F6 Preventive Proxy» — программное обеспечение для обнаружения интеллектуальных автоматизированных программ (ботов) и защиты от них, а также для снижения прямых убытков и издержек от мошеннической активности на веб-порталах и в мобильных приложениях (далее – АС Заказчика), которая происходит с использованием автоматизированных действий. В ПО используются машинное обучение и нейронные сети для анализа сессионных данных, экспертные скоринговые модели, анализ трафика, графовый анализ связанных ресурсов.

Preventive Proxy является подсистемой ПО «F6 Fraud Protection», которую можно использовать отдельно или с другими подсистемами продукта, комплексно защищая веб- или мобильное приложение и его пользователей от мошенничества.

1.3 Функциональные возможности ПО

Функциональные возможности «F6 Preventive Proxy»:

- Проверяет легитимность Пользователя и его окружение;
- Выявляет и защищает веб- и мобильные приложения от такой вредоносной бот-активности, как скрапинг, брутфорс, кража аккаунтов, DDoS-атаки, несанкционированное использование API и др.;
- Анализирует действия пользователя, которые будут отсутствовать или отличаться от человеческих в случае бот-активности (прямолинейность траектории мышки, набор символов на клавиатуре, клики и т.д.);
- Выявляет использование средств автоматизации пользовательских действий;
- Проверяет IP-адреса и подсети, из которых приходят запросы к защищаемому ресурсу, на легитимность;
- Предоставляет дополнительные вердикты и скоринговые оценки в систему противодействия мошенничества Заказчика в целях снижения уровня ложноположительных выявлений мошенничества;
- Защищает параметры реальной пользовательской сессии от повторного использования ботами;

– Позволяет проводить анализ запросов с применением механизма гибкой настройки ограничителя запросов (за счет конфигурации параметров limiters и counters);

– Управляет входящими запросами при помощи капчи (антибот тестирование).

– Позволяет формировать пользовательские политики правил для выявления и блокировки ботов с использованием конструктора в пользовательском интерфейсе.

2 ТРЕБОВАНИЯ К СИСТЕМЕ

ПО функционирует в программно-аппаратных средах, отвечающих хотя бы одному из следующих требований:

- Среда поддерживается компилятором языка программирования Golang. Этому требованию соответствуют операционные системы на платформах Linux, Windows и macOS;
- Среда позволяет запустить систему контейнеризации Docker. ПО может функционировать в среде с ядром, поддерживающим контрольные группы и изоляцию пространств имён (namespaces); существуют сборки для Windows, MacOS (Intel and Apple chipset), популярных дистрибутивов Linux и ARM.

2.1 Технические требования

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

- CPU 6 ядер, 3 Ггц;
- 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.

3 ОБЩИЕ ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ ПО

На Рисунке 1 изображены общие принципы функционирования ПО.

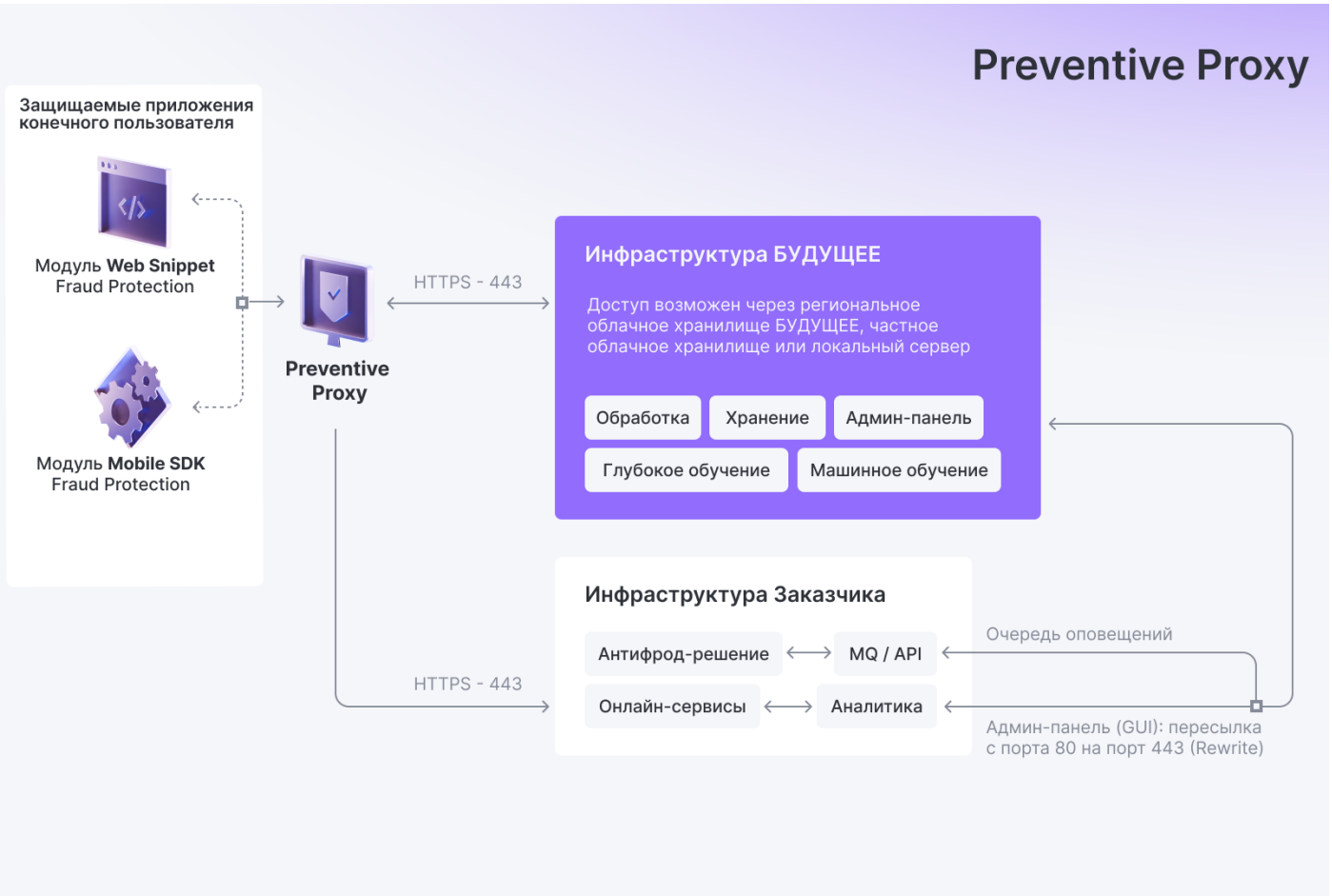

Рисунок 1. Общие принципы функционирования ПО.

Для работы Preventive Proxy понадобятся:

– клиентский модуль ПО «F6 Fraud Protection» (Web Snippet или Mobile SDK), который получает и передает в серверную часть поведенческие характеристики пользователя и окружения, в котором работает приложение;

– серверная часть ПО «F6 Fraud Protection» (Processing Hub). В ответ на данные, полученные из клиентского модуля, Processing Hub формирует и передает новый серверный файл cookie с вердиктом о наличии или отсутствии признаков бот-активности. При запросе из приложения клиентский модуль дополнительно формирует и передает клиентский файл cookie на базе серверного.

На основе данных, полученных из клиентского и серверного модулей Fraud Protection, Preventive Proxy проверяет наличие, корректность и уникальность файлов cookie на запросах с устройства пользователя, и на их основе принимает решение о наличии или отсутствии бот-активности в текущей пользовательской сессии.

4 РЕАЛИЗАЦИЯ ПО

4.1 Структура ПО

Архитектура ПО представляет собой сервис-ориентированную архитектуру, основанную на использовании распределенных, слабо связанных, заменяемых компонентов, оснащенных стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам.

Унификация программных интерфейсов осуществляется как минимум на следующих уровнях (но не ограничиваясь ими):

- браузера пользователя;
- мобильных приложений Заказчика;
- АС Заказчика;
- ПО.

4.2 Состав ПО

В состав ПО входят следующие подсистемы:

- *Подсистема получения данных с устройства пользователя*. Подсистема предназначена для получения параметров работы пользователей в рамках их сессии работы в защищаемом приложении Заказчика, содержащих первичные данные о мошеннической активности на стороне пользователей, дополнительные идентификационные данные пользовательских устройств и другие параметры;
- *Подсистема обработки данных*. Подсистема предназначена для обработки данных, полученных от подсистемы получения данных с устройств пользователей защищаемого приложения Заказчика – проверки данных на валидность и целостность;
- *Подсистема управления*. Подсистема, предназначенная для выполнения настроек и администрирования ПО;
- *Подсистема аналитики*. Подсистема предназначена для работы Аналитиков АС с выявленными событиями мошеннической активности, получения отчетов и статистики, настройки правил выявления мошеннической активности;
- *Подсистема информационного обмена*. Подсистема предназначена для экспорта/импорта данных между АС Заказчика и ПО как в режиме реального времени, так и в диалоговом (запросном) режиме и передачи в АС Заказчика вердиктов и скоринга выявленных фактов мошеннической активности;

– *Подсистема защиты информации.* Подсистема представляет собой программно-технический комплекс, предназначенный для защиты технических средств, программного обеспечения и данных от несанкционированного доступа к данным АС. Выполняет функции по идентификации и аутентификации сторон, производящих обмен информацией, функции по разграничению прав доступа к информационным ресурсам АС.

Подсистемы получения данных, обработки данных, передачи данных, управления, аналитики, информационного обмена, защиты информации должны иметь возможность размещаться как в облачной инфраструктуре Исполнителя, так и в инфраструктуре Заказчика. Решение принимается на этапе внедрения.

4.3 Функции частей ПО

Система сбора контрольных данных о структуре защищаемого приложения осуществляет:

WebSnippet:

– Сбор данных о JavaScript-коде.
– Сбор данных об iframe.
– Сбор данных о формах.
– Сбор информации о том, как и почему вы или третьи лица используете файлы cookie.

Mobile SDK:

– Сбор признаков работы вредоносных приложений на мобильном устройстве пользователя (только в Android SDK).
– Сбор идентификационных данных мобильного устройства.
– Сбор признаков работы на эмуляторе мобильного устройства.
– Сбор данных о поведении пользователя.

Система сбора идентификационных данных пользователя в защищаемом приложении имеет следующие функции:

WebSnippet:

– Получение имени учетной записи пользователя на защищаемом веб-ресурсе из форм для ее ввода в целях идентификации пользователя на стороне АС Заказчика.

Mobile SDK:

– Модуль собирает параметры, для идентификации устройства пользователя. Параметры, которые однозначно идентифицируют мобильное устройство, передаются в серверную часть Fraud Protection в хэшированном виде, чтобы скрыть их исходные значения.

Система защиты обмена данными с АС имеет следующие функции:

– Шифрование идентификационных данных пользователя на публичном RSA-ключе Заказчика;
– Шифрование контрольных данных.

Система обмена данными с АС имеет следующие функции:

– Посылка зашифрованных контрольных данных в АС;
– Периодическая посылка сигнальных данных о работе пользовательского модуля в АС.

5 ВЗАИМОДЕЙСТВИЕ ПО С АВТОМАТИЗИРОВАННЫМИ СИСТЕМАМИ

5.1 Структура взаимодействия

Во взаимодействии участвуют следующие компоненты:

- Браузер или мобильное приложение на устройстве пользователя с загруженным пользовательским модулем в составе страницы защищаемого веб-ресурса или в составе мобильного приложения;
- АС Разработчика;
- АС Заказчика.

АС Разработчика состоит из следующих компонентов:

- Серверная инфраструктура. Принимает, обрабатывает и анализирует контрольные данные, полученные от пользовательского модуля;
- Сервер управления. Служит для взаимодействия Заказчика с АС Разработчика;
- АРМ администратора. Обеспечивает настройку и сопровождение АС.

АС Заказчика состоит из следующих компонентов:

- Совокупность веб-серверов и серверов приложений веб-ресурса;
- Модуль автоматизации. Использует API к АС Разработчика для автоматизации реагирования на выявленные подозрительные события. Необходимость разработки этого модуля и правила реагирования определяются Заказчиком;
- АРМ оператора. Служит для ознакомления с подозрительными событиями и управления настройками выявления таких событий.

5.2 Порядок взаимодействия

Для работы Preventive Proxy понадобятся:

– клиентский модуль ПО «F6 Fraud Protection» (Web Snippet или Mobile SDK), который получает и передает в серверную часть поведенческие характеристики пользователя и окружения, в котором работает приложение;

– серверная часть ПО «F6 Fraud Protection» (Processing Hub). В ответ на данные, полученные из клиентского модуля, Processing Hub формирует и передает новый серверный файл cookie с вердиктом о наличии или отсутствии признаков бот-активности. При запросе из приложения клиентский модуль дополнительно формирует и передает клиентский файл cookie на базе серверного.

На основе данных, полученных из клиентского и серверного модулей «F6 Fraud Protection», Preventive Proxy проверяет наличие, корректность и уникальность файлов cookie на запросах с устройства пользователя, и на их основе принимает решение о наличии или отсутствии бот-активности в текущей пользовательской сессии.

5.3 Данные, передаваемые пользовательскими модулями

Пользовательские модули передают следующие контрольные данные с устройства пользователя:

WebSnippet:

1. Данные о пользователе:
   - результат применения алгоритма SHA1 к имени учетной записи пользователя;
   - результат применения алгоритма RSA с публичным ключом Заказчика к имени учетной записи клиента;
   - характеристики движения курсором.
2. Данные о странице защищаемого веб-ресурса:
   - javascript-код, загружаемый на страницы веб-ресурса;
   - структуру и атрибуты веб-форм, размещенных на страницах веб-ресурса;
   - атрибуты следующих HTML-элементов: iframe, object, embed, applet.
3. Данные о браузере, через который производится доступ на веб-ресурс:
   - User-Agent, куда входят:
     - Браузер и его версия;
     - Операционная система и ее версия;
     - Разрядность операционной системы;
     - Название и модель устройства клиента.
   - Accept-Encoding;
   - Accept-Language;
   - разрешение экрана;
   - глубина цвета;
   - доступность ActiveX;
   - часовой пояс;
   - шрифты браузера;
   - плагины браузера;
   - поддерживаемые языки;
   - canvas-отпечаток.

Mobile SDK:

1. Данные о пользователе:
   – результат применения алгоритма SHA1 к имени учетной записи пользователя;
   – результат применения алгоритма RSA с публичным ключом Заказчика к имени учетной записи клиента;
   – характеристики нажатий и движения по экрану.
2. Данные об устройстве:
   – Название сотового оператора.
   – Серийный номер сим-карты.
   – Версия ПО.
   – Аппаратный идентификатор устройства.
   – Бренд мобильного устройства.
3. Данные сети, в которой находится устройство:
   – IP адрес устройства.
   – Идентификаторы точек доступа.
   – Сетевые сертификаты устройства.

По согласованию с Заказчиком, перечень собираемых данных может различаться в зависимости от конфигурации пользовательских модулей и особенностей защищаемых приложений.

6 ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

6.1 Обеспечение конфиденциальности пользовательских данных

В АС Разработчика не передается пользовательская информация кроме обезличенного имени учетной записи пользователя или иного обезличенного идентификатора. Имя учетной записи пользователя передается в виде:

- результата хеш-функции от имени учетной записи;
- результата шифрования имени учетной записи с использованием публичного RSA-ключа Заказчика.

Обе операции производятся непосредственно на устройстве пользователя.

Получаемая Заказчиком информация о подозрительном событии содержит зашифрованное имя учетной записи. Используя соответствующий приватный RSA-ключ, только Заказчик может получить исходное имя пользователя.

Таким образом, имя пользователя недоступно третьим лицам, в том числе Разработчику.

6.2 Защита передаваемых данных

Весь обмен информации между пользовательским модулем, АС Разработчика и АС Заказчика производится по протоколу HTTPS.

Передаваемые данные из пользовательского модуля в АС Разработчика дополнительно кодируются в целях защиты от вредоносного программного обеспечения, функционирующего на устройстве пользователя.

6.3 Безопасность периметра АС Заказчика

Обмен между АС Заказчика и АС Разработчика всегда инициируется только со стороны Заказчика на следующие домены:

- https://fp-api.facct.ru;
- https://fp-back.facct.ru;
- https://ru.id.facct.ru.

Для защиты периметра Заказчика может быть применен любой тип фильтрации, ограничивающий обмен между АС Заказчика и указанными сайтами АС Разработчика. Необходимо отметить, что любому из вышеуказанных доменных имен соответствует

несколько IP-адресов, которые используются для обеспечения отказоустойчивости и распределения нагрузки.

6.4 Обеспечение доступности

Недоступность АС Разработчика никак не отражается на доступности и работоспособности защищаемого приложения как на стороне пользователя, так и на стороне Заказчика.

Тем не менее, АС Разработчика обеспечивает отказоустойчивость своей инфраструктуры.