DOI 10.37882/2223–2966.2022.04.23

МЕТОД ОБНАРУЖЕНИЯ И АНАЛИЗА АНОМАЛЬНОГО HTTP-ТРАФИКА С ПОМОЩЬЮ ЯЗЫКОВЫХ МОДЕЛЕЙ И ВЕКТОРНОГО ПРЕДСТАВЛЕНИЯ HTTP-ЗАПРОСОВ

Ляшков Михаил Андреевич
Аспирант, Тамбовский государственный университет имени Г.Р. Державина
iwishcoolwork@gmail.com
Пчелинцев Сергей Юрьевич
Аспирант, Тамбовский государственный университет имени Г.Р. Державина
veselyrojer@mail.ru
Ковалева Ольга Александровна
Д.т.н., доцент, Тамбовский государственный университет имени Г.Р. Державина; Тамбовский государственный технический университет
solomina-oa@yandex.ru

THE METHOD FOR DETECTING AND ANALYZING ANOMALOUS HTTP TRAFFIC USING NATURAL LANGUAGE MODELS AND VECTOR REPRESENTATION OF HTTP REQUESTS

M. Liashkov
S. Pchelintsev
O. Kovaleva

Summary. The paper proposes to use modern unsupervised learning approaches to automatically construct a representation of HTTP requests, and then use it to classify anomalies in traffic. The solution is based on techniques used in natural language processing, such as Doc2Vec, which can potentially achieve a deep understanding of HTTP messages and therefore improve the performance of an intrusion detection system. An important property is the interpretability of such a model. To test how the solution would work in the real world, a RoBERTa language model adapted from natural language processing was trained on normal network traffic, and its ability to detect anomalous traffic that the model had not seen before was measured. The proposed method is evaluated on publicly available data from CSIC2010, CSE-CIC–IDS2018. According to the results obtained, training the model on exceptionally normal network traffic makes it possible to detect anomalous HTTP requests well, this approach also does not require expert markup, and vector representations provide interpretability, the system is able to indicate specific places in a particular HTTP request that it considers anomalous. In most cases, it is easy to remove normal network traffic and it is relatively difficult to remove a sufficient amount of malicious traffic, since systems are not under attack most of the time and either expert time or a configured external system is required to isolate malicious traffic from the entire flow. The paper provides an explanation of the results based on clusters that occur in the space of vectorized queries and a simple logistic regression classifier. A good separation after t-SNE indicates an easy separation of http requests on the specified datasets, and the vector representation of requests makes it possible to receive requests similar in semantics from history.

Keywords: anomaly detection, http traffic, language models, model training.

Аннотация. В работе предлагается использовать современные подходы обучения без учителя для автоматического конструирования представления HTTP-запросов, а затем использовать ее для классификации аномалий в трафике. Решение основано на методах, используемых в обработке естественного языка, таких как Doc2Vec, которые потенциально могут достичь глубокое понимание сообщений HTTP и, следовательно, повысить эффективность системы обнаружения вторжений. Немаловажным свойством является интерпретируемость такой модели. Чтобы проверить, как решение будет работать в реальных условиях, была обучена языковая модель RoBERTa, адаптированная из области обработки естественных языков, на нормальном сетевом трафике, после была измерена ее возможность детектировать аномальный трафик, который модель не видела до этого. Предлагаемый метод оценивается на публично доступных данных CSIC2010, CSE-CIC–IDS2018. Согласно полученным результатам, обучение модели на исключительно нормальном сетевом трафике позволяет хорошо детектировать аномальные HTTP-запросы, также такой подход не требует экспертной разметки и векторные представления дают интерпретируемость, т.е. система способна указать конкретные места в конкретном HTTP-запросе, которые она посчитала аномальными. В большинстве случаев легко снять обычный сетевой трафик и относительно сложно снять достаточное количество вредоносного трафика, так как системы основную часть времени находятся не под атакой и для выделения вредоносного трафика из всего потока требуется либо экспертное время, либо настроенная внешняя система. В работе приводится объяснение результатов на основе кластеров, возникающих в пространстве векторизованных запросов, и простого классификатора логистической регрессии. Хорошее разделение после t-SNE говорит о легком разделении http-запросов на указанных датасетах, а векторное представление запросов дает возможность получать похожие по семантике запросы из истории.

Ключевые слова: детекция аномалий, http-трафик, языковые модели, обучение модели.

Введение

В отчете организации OWASP Top Ten атаки путем инъекции являются угрозой номер один в современном Интернете. Но многие другие также очень распространены, такие как межсайтовый скриптинг, подбор паролей или контента, использование неправильных конфигураций сервера. Векторы атак, нацеленных на веб-серверы, естественным образом используют протокол HTTP в качестве транспортного механизма, поэтому крайне важно разработать решения, которые позволили бы не только обнаруживать аномалии в запросах протокола, но и помогать с потенциальным анализом инцидентов.

Когда дело доходит до обнаружения атак в сетевом трафике, есть две возможные стратегии: обнаружение шаблонов или обученные модели (контролируемое или неконтролируемое обнаружение аномалий). В контексте текстовых протоколов, таких как HTTP, сопоставление шаблонов основано на экспертных знаниях и наборах правил (шаблонов), совпадение которых указывает на атаку. Например, если документ содержит несколько символов, закодированных в процентах, это может означать, что кто-то пытается обойти входную очистку. На этом принципе построен известный инструмент Snort (как минимум, частично).

При обучении моделей наиболее важной задачей является построение признаков. За прошедшие годы было разработано множество методов векторизации текста или изучения признаков, включая простые, такие как набор слов, tf-idf или пакет n-грамм; и более продвинутые, такие как Doc2Vec, fasText, ELMo или BERT. Некоторые из них уже успешно используются для решения проблемы обнаружения аномалий в HTTP-трафике.

Торрано-Хименес и др. [1] предложили решение, сочетающее экспертные знания с построением признаков n-грамм. Они также используют несколько алгоритмов дерева решений в качестве классификаторов. В [2] авторы предлагают комбинацию tfidf и word2vec для генерации векторов, а затем применяют повышение градиента для их классификации. Вартуни и др. [3] предложили другую модель, основанную на n-граммах, которая также использует нейронную сеть автоэнкодера для дальнейшего уменьшения размерности данных. По аналогичному принципу авторы в работе [4] разработали решение на основе метода Doc2Vec.

В некоторых работах вместо того, чтобы сначала генерировать векторы, а затем использовать их в классификации, представлен подход глубокого обучения, который на выходе определяет тип метки. Другими словами, они полностью контролируются и нуждаются в метках для установки весов в скрытых слоях. К таким методам относятся, среди прочего, [5] (глубокая нейронная сеть на основе слоев LSTM и CNN) и [6] (сочетает промежуточные векторы, полученные из модели CNN, LSTM и MRN).

В этой работе представлен метод, который позволяет получить векторное пространство для классификации в режиме без учителя. Мы считаем, что этот подход больше подходит для реальных сценариев, поскольку он не требует какой-либо маркировки (векторы по-прежнему могут быть классифицированы любым алгоритмом обнаружения отклонений). Более того, было решено обучать модель только на обычном трафике, чтобы проверить, сможет ли используемый классификатор обнаруживать аномалии. Главная цель состояла в том, чтобы создать хорошо работающее пространство HTTP-запросов (точки хорошо разделены и не требуют больших усилий для классификации) и показать, как полученное пространство может быть использовано экспертом в качестве инструмента для анализа трафика.

Основная идея разработанного метода заключается в использовании модели RoBERTa [7] для получения векторных представлений HTTP-запросов. Демонстрируется, что, используя это представление, мы можем достичь современного уровня производительности в задаче контролируемого обнаружения аномальных HTTP-запросов и как векторные представления можно использовать для анализа HTTP-трафика и определения интерпретируемых шаблонов токенов, характерных для аномалий, обнаруженных в HTTP-трафике. Стоит отметить, что интерпретируемость результатов является важной и уникальной характеристикой предлагаемого метода. Наконец, показывается, что, используя предложенные векторные представления, мы можем группировать и визуализировать шаблоны в HTTP-трафике. Были выполнены эксперименты с использованием эталонных наборов данных CSIC2010, CSE-CIC–IDS2018.

Обзор релевантных исследований

Анализ HTTP-запроса можно представить как задачу обработки естественного языка. Очень многое зависит от выбора языковой модели, которая позволила бы получить векторное пространство. В этой работе было решено использовать модель RoBERTa [8] (подробно описанную в следующем разделе), так как это современное решение для множества последующих задач обработки естественных языков. Модель не только решает проблему отсутствия предопределенного словарного запаса (OOV), но также, благодаря механизму самоконтроля, лучше кодирует токены с учетом их контекста (например, представьте, сколько вещей может быть представлено токеном «/» в HTTP-запросе).

Наиболее близкие к нашему исследованию работы, использующие набор данных CSIC2010, который также исследуется в работах [9, 10], и фокусирующиеся на представлении обучения без учителя. К сожалению, большинство из них по-разному определяют свои эксперименты, поскольку используют аномальный трафик в процессе генерации векторов.

Работа [4] является схожей по базовой идее, в основе которой метод векторизации Doc2Vec. HTTP-трафик обрабатывается моделью Doc2Vec в векторной форме, которая затем позволяет определить является ли трафик аномальным или нормальным. Модель Doc2Vec обучается на всем наборе CSIC2010 (как нормальный, так и аномальный трафик — комбинированные обучающие и тестовые наборы). HTTP-запросы модифицируются для представления первой строки этого запроса и группируются по 10 в «документы». Таким образом, каждый документ представляет либо 10 нормальных, либо 10 аномальных запросов. Классификация выполняется ансамблем классификаторов, обученных на данных, что составляет 70% от общего объема данных, и проверенных на тестовых данных, которые представляют оставшиеся 30%.

В работе [3] HTTP-запросы показаны в виде биграмм в словаре из 80 символов ASCII. Это дало 2572 функции, представляющие HTTP-запрос. Модель автоэнкодера использовалась для изучения представления классификатора. Алгоритм Isolation Forest использовался для определения связи данного HTTP-запроса в полученном векторном пространстве с нормальным или аномальным трафиком.

Авторы в работе [11] рассматривают полностью обучение с учителем. Нейронная архитектура LSTM–CNN используется для классификации HTTP-трафика. Сначала рекуррентная сеть LSTM обрабатывает HTTP-запрос на основе блочных признаков, затем выбранные состояния сети LSTM подаются в сверточную сеть, которая после обработки векторов передает их на выход в виде сети MLP. Далее происходит распределение HTTP-запроса в один из двух классов. Авторы метода сообщают об очень хороших результатах не только для коллекции CSIC2010, но и для коллекций CICIDS2017 и ISCX 2012, содержащих разные типы атак.

В исследовании [6] есть описание системы обнаружения веб-атак, основанной на ансамбле классификаторов и методов представления векторов из области естественной обработки языка. Эта система сначала токенизирует текст на основе подготовленного вручную словаря, содержащего маркеры, характерные для сетевого трафика. Результирующие текстовые представления векторизуются параллельно с использованием нейронных моделей на основе рекуррентных и сверточных сетей. Затем для них выполняется всесторонняя проверка, которая возвращает вектор оценки, определяющий выходную уверенность векторов в отношении их взаимного различия. Полученный вектор и векторы из нейронных моделей поступают в ансамбль классификаторов, который оценивает, является ли HTTP-запрос обычным трафиком или атакой. Метод апробирован на коллекции CSIC2010 и собственных коллекциях. К сожалению, нет информации о том, как именно производилось обучение модели.

Было решено воспроизвести работу [12] из-за схожести идеи. Авторы статьи разработали сверточный автоэнкодер, который учится реконструировать HTTP-сообщение, преобразованное в символьно-двоичное изображение. CAE (convolutional autoencoder) состоит из последовательных сверточных слоев, из которых основную часть составляют слои, основанные на архитектуре Inception-ResNet-v2. CAE включает в себя часть декодирования, выполненную в виде инвертированного кодера. Входными данными сети являются символьно-бинарные изображения, полученные преобразованием текста в матрицу вхождений символов из словаря (68 значений) в заданной позиции на протяжении всего HTTP-сообщения. Сеть учится воспроизводить такое представление только на обычных HTTP-сообщениях без аномалий и с критерием обучения бинарная кросс-энтропия. Предполагается, что обнаружение аномалий основано на отображении различий в реконструированном представлении в предположении, что сообщения HTTP, содержащие аномалии, будут иметь различное значение BCV (binary cross varentropy).

При воспроизведении этого решения пришлось преодолеть некоторые проблемы. В случае слоев основы в кодировщике и декодере нами скорректирован слой свертки из-за несоответствия размера и исправить слой свертки в модуле Reduction-B. Также в некоторых местах был настроен размер окна отступов, из-за отсутствия описания этих значений.

Предлагаемое решение

Предлагаемое решение состоит из описания методов токенизации, выбранной языковой модели, классификации, визуализации.

Кодирование пар байтов (BPE) — это метод сжатия, при котором наиболее распространенные пары байтов в данных заменяются байтом, не встречающимся в этих данных. BPE был перенесен в область обработки естественного языка как метод токенизации текста [13]. Основное внимание уделяется группированию наиболее часто встречающихся строк символов в учебном

корпусе. Метод начинается на уровне, где один символ представляет собой одиночный токен, сначала группируя пары символов, затем триплеты и так далее, пока не будет создан словарь, содержащий заданное для модели количество токенов. Методом расширения BPE является кодирование пар байтов на уровне байтов (BBPE) [14], которое основано на словаре байтов, а не символов. Это позволяет словарю оставаться небольшим, но при этом распознавать множество различных форм.

Метод основан на векторизации с использованием модели RoBERTa [8], которая, в свою очередь, основана на модели BERT [2]. BERT — это языковая модель, построенная на архитектуре Transformer. Она использует механизм внимания и позволяет последовательно обрабатывать данные, учитывая информацию о позиции токена в последовательности без использования рекурсии. Ключевым элементом BERT является включение двустороннего контекста для каждого токена обрабатываемой последовательности. Стандартная архитектура Transformer различает части кодера и декодера. Однако в случае BERT нас в основном интересует вывод кодировщика. В отличие от стандартных методов обработки последовательности, архитектура Transformer обрабатывает все токены одновременно, а не только в выбранном направлении. На вход метода подается последовательность токенов, которые сначала встраиваются в векторное пространство, а затем обрабатываются нейронной моделью. Процесс обучения модели BERT основан на двух методах: модели маскированного языка (MLM) и прогнозировании следующего предложения (NSP).

Для задач, описанных в нашей статье, токенизатор основан на BBPE и обучается на всем наборе данных, содержащем как нормальный, так и аномальный трафик. Это позволяет нам создать достаточно точный токенизатор и избежать ошибки смещения метода обнаружения аномалий за счет обнаружения большого количества коротких токенов после токенизации, что не обязательно будет означать аномальный трафик в реальности. Используемая нами реализация основана на решении, предоставленном HuggingFace Transformers [14]. Выход нашего токенизатора — это вход модели RoBERTa.

В этой работе были использованы следующие три алгоритма, они дают хорошее представление о том, как выглядит пространство встраивания: 1) логистическая регрессия, 2) случайный лес, 3) машина опорных векторов с линейным ядром. В наших экспериментах мы использовали реализации Scikit-learn [15] с большинством параметров, оставленными по умолчанию. При этом увеличили максимальное количество итераций в алгоритме случайного леса до 500.

При визуализации полученных вложений на двумерной плоскости был использован метод t-SNE [16] для уменьшения размерности. Считается, что этот метод сохраняет глобальную структуру лучше, чем классическое многомерное масштабирование [17], потому что он сохраняет сходство между точками, определенными как нормализованные гауссовы. Поэтому он использует евклидово расстояние в исходном пространстве. Сходства в низкоразмерном пространстве моделируются нормализованным распределением Стьюдента-t. Метод t-SNE минимизирует расхождение Кульбака-Лейблера между сходствами в обоих пространствах в отношении расположения точек в низкоразмерном пространстве.

Эксперименты

Каждый используемый нами набор данных был разделен на две части: часть обучения (обычные запросы, используемые только для обучения представлению) и часть вывода (обычный и аномальный трафик, закодированный с помощью модели и используемый для классификации).

Набор данных HTTP CSIC2010 [2] является набором данных, наиболее часто используемым в подобных задачах, поскольку он содержит готовые к использованию текстовые файлы с HTTP-запросами. Набор данных включает такие атаки, как внедрение SQL, переполнение буфера, сбор информации, раскрытие файлов, внедрение CRLF, XSS и несколько других атак. Авторы генерировали трафик, ориентируясь на одно веб-приложение электронной коммерции, что делает его идеальным для использования в нашем подходе. Из-за необходимости обнаруживать атаки с внедрением CRLF, мы решили кодировать символы CR (возврат каретки) и LF (перевод строки) как литеральные строки «\r» и «\n».

CSE-CIC–IDS2018 [18] — это хорошо известный набор данных, разработанный Канадским институтом кибербезопасности. Несмотря на отличное качество, он не был предназначен для решения проблемы, обсуждаемой в этой статье. К счастью, он содержит несколько веб-атак, таких как «Brute Force — Web», «Brute Force — XSS» и «SQL Injection», которые мы извлекли из перехваченных пакетов (за 23 февраля 2018 г.). Этот набор данных отличается от других тем, что запросы обычного трафика направляются многим веб-приложениям. Для создания аномального трафика авторы использовали приложение DVWA 3, размещенное на одном компьютере. Это заставило нас внести небольшие изменения в запросы, чтобы избежать ложных корреляций:

1. Каждое поле «Хост» в запросах всегда указывало на один IP-адрес. Было решено случайным образом изменить это на любой адрес, который был в обычном наборе данных.

Таблица 1. Сравнение производительности предложенного метода векторизации (на основе RoBERTa) и CAE с использованием нескольких алгоритмов классификации. Все результаты получены из стратегии стратифицированной перекрестной проверки (k-fold) с k = 5.

Датасет	Метод	FPR90	FPR99	F1	MCC
CSIS2010	LR	0.017	0.075	0.951	0.916
	SVM	0.003	0.042	0.969	0.948
	RF	0.010	0.070	0.959	0.930
CSE-CIC–IDS 2018	LR	0.000	0.000	0.999	0.998
	SVM	0.000	0.000	0.999	0.998
	RF	0.000	0.000	0.999	0.998

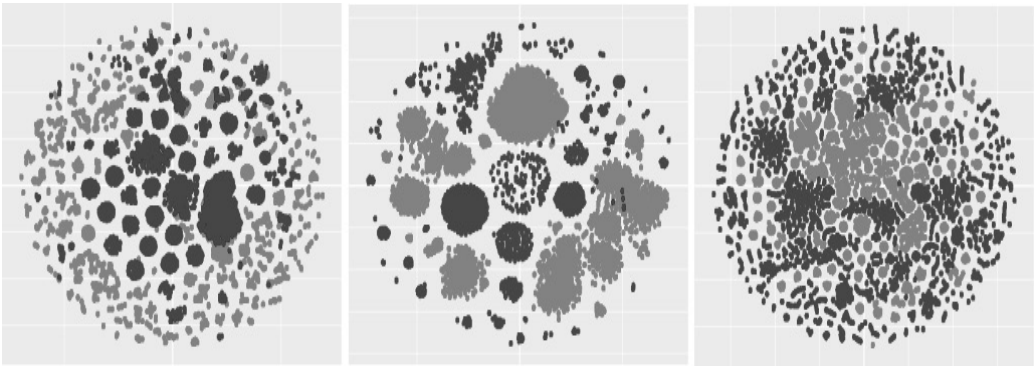

Рис. 1. Пространство векторизованных запросов уменьшено до 2D с помощью t-SNE. Серый цвет обозначает аномальный трафик.

2. Каждый URI в запросе всегда начинался с одной из следующих строк: «/DVWA/vulnerabilities/xss», «DVWA/dvwa», «/DVWA». Такие URI были удалены.
3. Каждый аномальный запрос содержал заголовок «Upgrade-Insecure-Requests», который также был удален.

Кроме того, обычный трафик не всегда содержал текстовые полезные нагрузки, поэтому мы удалили те, которые не соответствовали правильному «Content-Type» (например, «application/json»).

Для сравнения с другими подходами мы выбрали несколько работ, использующих набор данных CSIC2010, которые были описаны более подробно выше.

Вартуни и др. [3] провели эксперименты, аналогичные нашим, хотя они использовали только подмножество набора данных CSIC2010. Их решение не совсем обучение без учителя, поскольку они решили использовать алгоритм Isolation Forest для классификации. В статье сообщается о 84,12% оценки F1, что является минимумом того, что интересно для рассмотрения (т.к. методы классификации без учителя, как правило, работают хуже, чем с учителем).

Лю и др. [11] достигли 99,12% TPR и 0,22% FPR (FPR99 = 0,22%). Их решение основано на парадигме обучения с учителем, поэтому мы, вероятно, получим немного худшие результаты. Однако стоит отметить, что их набор тестов меньше и более несбалансирован, чем наш — теоретически это могло привести к небольшому завышению результатов (особенно TPR).

Здесь также уместно упомянуть статью [4], авторы которой представляют наиболее близкий подход. Его авторы представили результаты, близкие к идеальному классификатору, но также сделали некоторые предположения, которые видятся сомнительными. Во-первых, языковая модель (Doc2Vec) обучается на всем наборе данных (для обучения и вывода используются одни и те же образцы). Во-вторых, они объединили десять разных образцов в один документ, что значительно упрощает решение задачи. Как правило, использование

```
GET http://localhost:8080/tienda1/publico/caracteristicas.jsp?id=%27%3B+DROP+TABLE+usuarios%3B+SELECT
+*+FROM+datos+WHERE+nombre+LIKE+%27%25 HTTP/1.1\r\n
Pragma: no-cache\r\n
Cache-control: no-cache\r\n
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5\r\n
Accept-Encoding: x-gzip, x-deflate, gzip, deflate\r\n
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5\r\n
Accept-Language: en\r\n
Host: localhost:8080\r\n
Connection: close\r\n
\r\n
\r\n
```

Рис. 2. Случайный аномальный запрос из CSIC2010, выделен аномальный трафик

Doc2Vec в качестве языковой модели — неплохая идея, но доказано, что RoBERTa лучше справляется с несколькими последующими задачами языковой обработки. Более того, Doc2Vec не решает проблему отсутствия словарного запаса. После создания векторов мы использовали их для обучения трех разных классификаторов.

Результаты для набора данных CSIC2010 показывают, что нам легко удалось превзойти наши базовые требования. Лучшим классификатором оказался SVM с F1-Score 96,9%. По сравнению с результатами, которые были приняты выше за идеальные, мы получили FPR99 = 4,2%, что намного хуже, чем в [11]. Полагаем, что это в основном связано с различным использованием данных.

Результаты для набора данных CSE-CIC–IDS2018 показывают, что мы получили идеальный классификатор. Это означает, что либо набор данных действительно просто классифицировать, либо мы допустили некоторые ошибки при обработке данных.

Полученные в результате применения нашего подхода данные приведены в таблице 1.

Результаты работы t-SNE представлены на рисунке 1.

Визуализации на рисунке 1 показывают, что классы хорошо разделены даже в более низком пространстве. Это показывает, что похожие HTTP-запросы на самом деле сгруппированы вместе, и дает возможность исследовать окрестности любой выборки. В табл. 2 показаны несколько n-х ближайших (по евклидову расстоянию) отсчетов к заданному в первой строке. Как видно, атаки SQL Injection на самом деле близки друг к другу. Интересно, что дальше по соседству с образцом находятся другие виды инъекций — команды операционной системы. Мы считаем, что этот подход может быть использован экспертом для выявления подобных попыток атаки и, следовательно, для помощи в анализе после инцидента. На рисунке 2 представлен случайный аномальный запрос.

Чтобы понять, какие особенности запросов доказывают их аномальность, мы сначала сгенерировали список токенов, а затем набор документов, которые были основаны на исходном, но из каждого из них было удалено по одному токену. Затем мы сгенерировали векторы для набора и классифицировали их с помощью LR. Когда образец без данного токена приближается к гиперплоскости, это означает, что токен был релевантен классу, определенному плоскостью. Мы собрали все расстояния для каждого документа в наборе, а затем нормализовали их, используя масштабирование минимум-максимум. Окончательная оценка представляет собой разницу полученного расстояния от среднего значения. Интенсивность цвета на 3 и 4 представляет эти баллы для каждого маркера (мы окрашивали только маркеры, положительно коррелирующие с аномалиями).

В таблице 2 представлены 24 основных функции для двух наборов данных (сумма по 50 различным документам).

Для CSIC2010 мы решили создать таблицу (см. таблицу 3) для окрестности выборки, упомянутой в таблице 2. Интересно, что токен «\» в целом имеет наивысший балл. Это связано с внедрением CRLF (токены с отрицательной корреляцией должны сбалансировать эту функцию). Остальная часть таблицы показывает токены, тесно связанные с SQL-инъекциями.

В случае датасета CSE-CIS-IDS2018 было взято 50 случайных выборок, чтобы определить, почему набор данных так легко классифицировать. Как вы можете видеть, «Accept» и «Accept-Encoding» тесно связаны с подмножеством аномалий. Это означает, что эти две линии появляются почти в каждом образце практически в неизменном виде (это очень сильная особенность). Именно поэтому мы получили такие отличные результаты. У нас

Таблица 2. Топ 24 особенностей аномального трафика для датасета CSE-CIS-IDS2018 и CSIC2010

CSE-CIS-IDS2018				CSIC2010			
токен	оценка	токен	оценка	токен	оценка	токен	оценка
-	18.56	html	5.29	\	203.38	5	17.20
/	16.33	xhtml	5.24	r	174.68	3	17.10
Accept	14.32	;	5.09	+	68.24	LIKE	12.28
:	11.41	+	4.81	%	41.09	+%	11.69
xml	10.74	0	4.15	.	36.05	datos	11.58
,	9.55	text	3.19	=	32.34	+*+	11.28
Encoding	9.13	9	2.59	/	31.32	FROM	11.14
deflate	7.78	=	2.30	n	28.86	TABLE	11.09
.	7.50	8	2.13	1	23.78	WHERE	11.04
gzip	7.42	GET	2.11	B	22.58	DROP	10.57
application	7.19	login	1.94	27	17.75	&	10.54
q	6.85	php	1.43	0	17.30	SELECT	10.21

Таблица 3. Пример поиска N-го ближайшего сэмпла запроса к заданному (образец 0)

N	Строка
0	/vaciar.jsp? B2=Vaciar+carrito%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
1	/vaciar.jsp? B2=Vaciar+carrito%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
5	/vaciar.jsp? B2=Vaciar+carrito%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
10	/vaciar.jsp? B2=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
15	/entrar.jsp?errorMsg=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
25	/entrar.jsp?errorMsg=Credenciales+incorrectas%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25
50	/anadir.jsp?id=2&nombre=Jam%F3n+Ib%E9rico&precio=85&cantidad=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+ datos+WHERE+nombre+LIKE+%27%25&B1=A%F1adir+al+carrito
100	/anadir.jsp?id=3&nombre=Queso+Manchego&precio=39&cantidad=86&B1=%3C%21—%23exec+cmd%3D%22rm+-rf+%2F% 3Bcat+%2Fetc%2Fpasswd%22+—%3E
200	/anadir.jsp?id=2&nombre=Queso+Manchego&precio=85&cantidad=86%22%3E%3C%21—%23EXEC+cmd%3D%22dir+%22—%3E% 3C&B1=A%F1adir+al+carrito

была аналогичная проблема с CSIC2010. Оказалось, что каждый аномальный запрос заканчивался символом «\n», а не «\r\n».

Заключение

Был предложен метод анализа аномального HTTP-трафика, использующий векторные представления HTTP-запросов на основе модели RoBERTa — современного метода представления текста. Используя эти представления, возможно различать нормальные и аномальные HTTP-пакеты. Была показана эффективность представлений в парадигме обучения с учителем с использованием наборов данных CSIC2010 и CSE-CIC–IDS2018. Было продемонстрировано, что эти модели обобщаются на новые данные, поскольку они успешно обнаружили аномалии на отложенных данных.

Важной характеристикой предлагаемого метода является то, что векторные представления позволяют нам анализировать аномальные HTTP-запросы с точки зрения интерпретируемых подмножеств токенов/признаков. Было показано, как получить такие информативные шаблоны.

В дополнение к этому хочется отметить, что векторизованные HTTP-запросы имеют тенденцию группироваться в четкие, непересекающиеся кластеры похожих, нормальных или аномальных запросов. Данное исследование использовало только данные, собранных в формате pcap с помощью регистратора сетевого трафика. Мы не использовали доступ к HTTP-серверу и журналы ошибок. Журналы ошибок и логи доступа веб-серверов можно легко сопоставить с проанализированными данными и даже использовать для автоматической классификации атак, что формирует еще одно направление для дальнейшей работы.

Среди ограничений текущей работы следует отметить следующие. Размер языковой модели относительно большой (размерность векторного представления около 3000), что влияет на время обучения и вывода (процесс обучения для набора данных CSIC2010 занял около 7 часов с использованием двух графических процессоров RTX 2080Ti). Необходимы дальнейшие исследования, чтобы уменьшить размер языковой модели, уменьшить размерность векторного представления, и чтобы упростить последующий анализ.

В этой работе мы использовали методы обучения с учителем для обнаружения аномалий HTTP. Этот подход требует, чтобы аннотированный набор данных был доступен для обучения модели (что может быть дорого размечать), и позволяет нам обнаруживать только те типы атак, которые известны в обучающих данных. Эти ограничения можно смягчить, используя неконтролируемые (без учителя) методы обнаружения аномалий. Хорошая разделимость на t-SNE показывает, что это может быть эффективно при использовании векторных представлений.

Наконец, явное ограничение этой и подобных работ связано с ограниченной доступностью современных репрезентативных наборов данных о сетевом трафике. Большая часть исследований основана только на одном наборе данных CSIC2010 и других самостоятельно подготовленных данных (собранных вручную, не опубликованных и не извлеченных из pcap). Качество таких наборов данных трудно определить, и оно может быть сомнительным, что влияет на качество/обобщение моделей трафика.

ЛИТЕРАТУРА

1. Carmen Torrano-Gimenez, Hai Thanh Nguyen, Gonzalo Alvarez, and Katrin Franke. Combining expert knowledge with automatic feature extraction for reliable web attack detection. Security and Communication Networks, 8(16):2750–2767, 2015.
2. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
3. Ali Moradi Vartouni, Saeed Sedighian Kashi, and Mohammad Teshnehlab. An anomaly detection method to detect web attacks using stacked auto-encoder. In 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), pages 131–134. IEEE, 2018.
4. Saikat Das, Mohammad Ashrafuzzaman, Frederick T Sheldon, and Sajjan Shiva. Network intrusion detection using natural language processing and ensemble machine learning. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pages 829–835. IEEE, 2020.
5. Carmen Torrano Giménez, Alejandro Pérez Villegas, and Gonzalo Álvarez Marañón. HTTP data set CSIC2010. Information Security Institute of CSIC (Spanish Research National Council), 2010.
6. Chaochao Luo, Zhiyuan Tan, Geyong Min, Jie Gan, Wei Shi, and Zhihong Tian. A novel web attack detection system for internet of things via ensemble classification. IEEE Transactions on Industrial Informatics, 2020.
7. Jieling Li, Hao Zhang, and Zhiqiang Wei. The weighted word2vec paragraph vectors for anomaly detection over http traffic. IEEE Access, 8:141787–141798, 2020.
8. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, DBLP: journals/corr/abs-1907–11692, 2019.

9. Ляшков М.А., Арзамасцев А.А. Применение сетей долгой краткосрочной памяти при обнаружении аномалий в HTTP-трафике // Материалы и методы инновационных исследований и разработок: межд. конф. (Оренбург, 20 октября 2018). Уфа. Изд-во АЭТЕРНА, 2018. С. 17–20.
10. Ляшков М.А., Арзамасцев А.А. Разработка методов автоматической настройки системы обнаружения вторжений // EurasiaScience: сборник статей XXVII международной научно-практической конференции (Москва, 15 февраля 2020). Москва. Изд-во «Научно-издательский центр Актуальность.РФ», 2020. С. 80–81.
11. Jiaxin Liu, Xucheng Song, Yingjie Zhou, Xi Peng, Yanru Zhang, Pei Liu, and Dapeng Wu. Deep anomaly detection in packet payload. arXiv preprint arXiv:1912.02549, 2019.
12. Seungyoung Park, Myungjin Kim, and Seokwoo Lee. Anomaly detection for HTTP using convolutional autoencoders. IEEE Access, 6:70884–70901, 2018.
13. Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909, DOI: 10.18653/v1/P16–1162, 2015.
14. Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural machine translation with byte-level subwords. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 9154–9160, 2020.
15. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.
16. Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008.
17. Ingwer Borg, Patrick J.F. Groenen, and Patrick Mair. Applied Multidimensional Scaling and Unfolding. Springer Publishing Company, Incorporated, 2nd edition, 2018. ISBN3319734709, 9783319734705.
18. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In ICISSp, pages 108–116, 2018.

Тамбовский государственный университет имени Г.Р. Державина