Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2024. Т. 24, вып. 3. С. 442–451
Izvestiya of Saratov University. Mathematics. Mechanics. Informatics, 2024, vol. 24, iss. 3, pp. 442–451
https://mmi.sgu.ru https://doi.org/10.18500/1816-9791-2024-24-3-442-451, EDN: OJWHMC

Научная статья
УДК 004.032.2

Метод повышения качества обнаружения атак на веб-приложения с применением предобученных моделей естественного языка

О. А. Ковалева✉, **А. В. Самохвалов, М. А. Ляшков, С. Ю. Пчелинцев**

Тамбовский государственный университет имени Г. Р. Державина, Россия, 392036, г. Тамбов, ул. Интернациональная, д. 33

Ковалева Ольга Александровна, доктор технических наук, профессор кафедры математического моделирования и информационных технологий, solomina-oa@yandex.ru, https://orcid.org/0000-0003-0735-6205, AuthorID: 829259
Самохвалов Алексей Владимирович, кандидат педагогических наук, доцент, заведующий кафедрой математического моделирования и информационных технологий, samohvalov@gmail.com, https://orcid.org/0000-0002-3151-3250, AuthorID: 178141
Ляшков Михаил Андреевич, аспирант кафедры математического моделирования и информационных технологий, iwishcoolwork@gmail.com, https://orcid.org/0000-0002-7793-7024
Пчелинцев Сергей Юрьевич, аспирант кафедры математического моделирования и информационных технологий, veselyrojer@mail.ru, https://orcid.org/0000-0001-9195-8318

Аннотация. Исследуется использование методов глубокого обучения для повышения производительности защитных экранов веб-приложений (WAF). Описывается конкретный метод повышения качества функционирования защитных экранов и приводятся результаты его тестирования на публично доступных данных CSIC 2010. Большинство защитных экранов веб-приложений работают на основе правил, которые были составлены экспертами. При работе сетевые экраны проверяют HTTP-запросы, которыми обмениваются клиент и сервер для обнаружения атак и блокирования потенциальных угроз. Ручное составление правил требует времени экспертов, а распространяемые готовые наборы правил не учитывают специфику конкретных пользовательских приложений, поэтому допускают много ложноположительных срабатываний и пропускают много сетевых атак. В последние годы использование предварительно обученных языковых моделей привело к значительным улучшениям в разнообразном наборе задач обработки естественного языка, поскольку они способны выполнять перенос знаний. В статье описывается адаптация этих подходов на сферу информационной безопасности, т. е. использование предварительно обученной языковой модели в качестве средства извлечения признаков для сопоставления HTTP-запроса с вектором признаков. Эти векторы используются для обучения классификатора. Предложено решение, которое состоит из двух этапов. На первом этапе создается глубокая предобученная языковая модель на основе нормальных HTTP-запросов к веб-приложению. На втором этапе эта модель используется в качестве средства извлечения признаков и обучается с помощью одноклассового классификатора. Оба этапа совершаются для каждого приложения. Экспериментальные результаты показывают, что предлагаемый подход значительно превосходит подходы классического Mod-Security, основанного на правилах, настроенных с помощью OWASP CRS, и не требует участия эксперта по безопасности для определения правил срабатывания.

Ключевые слова: сетевые экраны, анализ HTTP-запросов, предварительно обученные языковые модели

Для цитирования: *Ковалева О. А., Самохвалов А. В., Ляшков М. А., Пчелинцев С. Ю.* Метод повышения качества обнаружения атак на веб-приложения с применением предобученных моделей естественного языка // Известия Саратовского университета. Новая серия. Серия: Математика.

Механика. Информатика. 2024. Т. 24, вып. 3. С. 442–451. https://doi.org/10.18500/1816-9791-2024-24-3-442-451, EDN: OJWHMC

Article

The quality improvement method for detecting attacks on web applications using pre-trained natural language models

O. A. Kovaleva✉, **A. V. Samokhvalov, M. A. Liashkov, S. Yu. Pchelintsev**

Derzhavin Tambov State University, 33 Internationalnaya St., Tambov 392036, Russia
Olga A. Kovaleva, solomina-oa@yandex.ru, https://orcid.org/0000-0003-0735-6205, AuthorID: 829259
Alexey V. Samokhvalov, samohvalov@gmail.com, https://orcid.org/0000-0002-3151-3250, AuthorID: 178141
Mikhail A. Liashkov, iwishcoolwork@gmail.com, https://orcid.org/0000-0002-7793-7024
Sergey Yu. Pchelintsev, veselyrojer@mail.ru, https://orcid.org/0000-0001-9195-8318

Abstract. This paper explores the use of deep learning techniques to improve the performance of web application firewalls (WAFs), describes a specific method for improving the performance of web application firewalls, and presents the results of its testing on publicly available CSIC 2010 data. Most web application firewalls work on the basis of rules that have been compiled by experts. When running, firewalls inspect HTTP requests exchanged between client and server to detect attacks and block potential threats. Manual drafting of rules requires experts' time, and distributed ready-made rule sets do not take into account the specifics of particular user applications, therefore they allow many false positives and miss many network attacks. In recent years, the use of pretrained language models has led to significant improvements in a diverse set of natural language processing tasks as they are able to perform knowledge transfer. The article describes the adaptation of these approaches to the field of information security, i.e. the use of a pretrained language model as a feature extractor to match an HTTP request with a feature vector. These vectors are then used to train the classifier. We offer a solution that consists of two stages. In the first step, we create a deep pre-trained language model based on normal HTTP requests to the web application. In the second step, we use this model as a feature extractor and train a one-class classifier. Both steps are performed for each application. The experimental results show that the proposed approach significantly outperforms the classical Mod-Security approaches based on rules configured using OWASP CRS and does not require the involvement of a security expert to define trigger rules.
Keywords: firewalls, HTTP request analysis, pre-trained language models
For citation: Kovaleva O. A., Samokhvalov A. V., Liashkov M. A., Pchelintsev S. Yu. The quality improvement method for detecting attacks on web applications using pre-trained natural language models. *Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2024, vol. 24, iss. 3, pp. 442–451 (in Russian). https://doi.org/10.18500/1816-9791-2024-24-3-442-451, EDN: OJWHMC

Введение

Стандартной практикой безопасности стало развертывание брандмауэра веб-приложений (WAF) [1] для выявления атак, использующих уязвимости веб-приложений. WAF — это программа, которая перехватывает и проверяет весь трафик между веб-сервером и его клиентами, пытаясь найти атаки внутри содержимого HTTP-пакета. Реализация WAF с открытым исходным кодом, ставшая стандартом де-факто, — это ModSecurity[1]. Действия, предпринимаемые этим WAF, управляются правилами, которые определяют с помощью регулярных выражений содержимое HTTP-пакетов. ModSecurity поставляется с набором

[1]Trustwave. URL: https://www.trustwave.com/en-us/ (дата обращения: 18.02.2022).

правил по умолчанию, известным как основной набор правил OWASP[2] (OWASP CRS), для обработки наиболее распространенных уязвимостей, включенных в первую десятку уязвимостей рейтинга OWASP[3].

Рассмотрим примеры таких правил. Правила являются директивами в файле настроек, описывающими, что делать с данными, полученными на основании других директив. Важной директивой для создания правил является директива SecRule, синтаксис которой представлен ниже.

SecRule ПЕРЕМЕННЫЕ ОПЕРАТОР [ДЕЙСТВИЯ]. Ниже приведен пример правила:
SecRule ARGS|REQUEST_HEADERS "@rx <script"id:101,msg: 'XSS Attack', severity:ERROR, deny,status:404

В данном случае:

- ARGS и REQUEST_HEADERS являются переменными (параметры и заголовки запросов соответственно);
- *@rx* является оператором, используемым для поиска шаблона в значениях переменных (в данном случае шаблоном является строка <script);
- *id, msg, severity, deny* и *status* являются действиями, выполняемыми при обнаружении соответствия строк шаблону.

Это правило используется для отражения XSS-атак с помощью проверки наличия строки <script в параметрах запроса и заголовке и для генерации сообщения 'XSS Attack'. В данном правиле используется действие id:101; с помощью него любой запрос, предназначенный для проведения атаки, будет отклонен со статусом 404.

Однако этот подход, основанный на правилах, имеет некоторые недостатки. Правила статичны и негибки по своей природе, поэтому OWASP CRS обычно дает довольно высокий уровень ложных срабатываний, который в некоторых случаях может приближаться к 40%[4], что потенциально приводит к отказу в обслуживании приложения. В систематическом обзоре, представленном в [2], анализируется доступная научная литература, посвященная обнаружению веб-атак с использованием методов машинного обучения.

В [3–5] были представлены решения, в которых подход ModSecurity к обнаружению веб-атак на основе правил дополняется моделями на основе машинного обучения, чтобы смягчить недостатки подхода на основе правил. Вместе с тем проблема высокого процента ложноположительных срабатываний (FPR) все еще остается нерешенной и требует дальнейшего исследования.

Основной целью нашего исследования является повышение качества функционирования защитных экранов. Для достижения поставленной цели необходимо получить решение следующих задач:

- повышение процента выявленных атак (TPR);
- обеспечение низкого процента ложноположительных срабатываний (FPR).

TPR и FPR показывают соотношение запросов, соответственно правильно и неправильно классифицированных как атаки. Частота ложноположительных срабатываний FPR рассчитывается как $FP/(FP+TN)$, где FP — количество ложноположительных результатов, а TN — количество истинно отрицательных результатов ($FP+TN$ — общее количество отрицательных результатов). Это вероятность ложной тревоги: будет дан положительный результат, когда истинное значение окажется отрицательным.

Процент корректно классифицированных атак (TPR, также называемый чувствительностью) рассчитывается как $TP/(TP+FN)$, где TP — количество истинно положительных результатов, а FN — количество ложноотрицательных результатов ($TP+FN$ — общее количество положительных результатов). TPR — это вероятность того, что фактический положительный результат будет положительным.

[2]OWASP. Owasp modsecurity core rule set project. URL: https://coreruleset.org/ (дата обращения: 18.02.2022).
[3]OWASP. Owasp top ten project. URL: https://owasp.org/www-project-top-ten/ (дата обращения: 18.02.2022).
[4]Handling false positives with the owasp modsecurity core rule set. URL: https://www.netnea.com/cms/nginx-tutorial-8-handling-false-positives-modsecurity-core-rule-set/ (дата обращения: 18.02.2022).

В работе предложен подход, использующий методы глубокого обучения для повышения производительности ModSecurity. Он состоит из двухэтапной структуры обучения: сначала создается экстрактор признаков, используя методы глубокого обучения; затем обучается с учителем модель. HTTP-запросы рассматриваются как необработанный текст, а для его предварительной обработки обучается модель глубокого языка с архитектурой, предложенной в [6].

1. Обзор релевантных исследований

Методы обработки естественных языков (NLP) были значительно улучшены благодаря достижениям глубокого обучения [6–9]. Эти методы основаны на двухэтапном подходе. Во-первых, модель изучает глубокое контекстуальное представление слов из необработанного текста под самоконтролем (разновидность обучения с учителем, где в качестве меток используются прогнозы самой модели). Этот этап также называется предварительным обучением. Затем эту предварительно обученную языковую модель можно использовать в последующих задачах NLP.

Традиционные техники NLP представляют слова как атомарные единицы, а текст преобразуется в числовой вектор с использованием однократного кодирования. При таком подходе есть две основные проблемы. Во-первых, нет понятия сходства между словами, так как они представлены в словаре индексами [10]. Кроме того, размер вектора равен размеру словаря, вследствие чего у методов машинного обучения возникают проблемы, связанные с многомерными пространствами признаков, такими как проклятие размерности. С развитием методов машинного обучения стало возможным обучать более сложные модели.

Согласно исследованиям [11–14] одной из самых успешных концепций является использование векторных представлений слов, также известное как встраивание слов. В этом подходе слова представляются в непрерывном векторном пространстве гораздо меньшей размерности, чем предыдущий линейный вариант. Кроме того, было показано, что слова с семантическим сходством имеют тенденцию быть рядом в векторном пространстве [11]. В последнее десятилетие встраивание слов зарекомендовало себя как основной элемент многих NLP-систем. Однако, поскольку методы встраивания слов статичны, упускается важный элемент для полного захвата местного контекста, т. е. семантическое и синтаксическое значение слов. С помощью этих методов на самом деле учатся фиксировать общий (наиболее распространенный) контекст слов в своих представлениях, но они не помогают справиться с полисемией. Замена статических вложений глубокими контекстуальными представлениями привела к значительным улучшениям в разнообразном наборе задач NLP. Идея проста: слову присваивается представление, являющееся функцией всей входной последовательности (всей текстовой последовательности). Успех глубоких контекстуализированных представлений слов предполагает, что, несмотря на то что NLP-системы обучаются только с целью моделирования языка, они изучают легко переносимые и не зависящие от задачи свойства языка [15].

Авторами предлагается использовать глубокое контекстуализированное представление HTTP-запросов для извлечения векторов признаков, которые затем будут использоваться для обучения классификатора обнаружению атак на веб-приложения. На первом этапе предлагается создать с нуля глубокую предварительно обученную языковую модель, используя набор HTTP-запросов от веб-приложения, которое необходимо защитить. На втором этапе применяется стратегия, основанная на функциях, для преобразования каждого HTTP-запроса в вектор функций. Иными словами, как только будет получена предварительно обученная модель, каждый HTTP-запрос преобразовывается в числовое представление, используя веса последнего слоя сети, также известного как извлечение признаков. Используя эти представления в качестве входных данных, строится модель бинарной классификации на принадлежность к каждому классу.

В релевантной работе [16] Крюгеля и Вигны предлагается подход к обнаружению аномалий, при котором моделируются определенные характеристики параметров URL, такие как длина

параметра и порядок ввода, для создания вероятностной грамматики каждого параметра. В предложенном авторами статьи [16] методе используется весь запрос, а не только параметры URL, фиксируя нормальное поведение путем моделирования возникновения определенного набора токенов — специально сформированных идентификаторов, выделенных из HTTP-запроса, содержащих его характерные признаки. Это позволяет фиксировать поведение отправляемых данных при обычной работе приложения и использовать атаки, присутствующие в теле и заголовке запросов (не только в URL).

Некоторые авторы предлагают методы обнаружения аномалий, которые работают над упрощением значений параметров приложения. В [17] числа и буквенно-цифровые последовательности абстрагируются, представляя каждую категорию одним символом. В [18] Торрано-Хименес с соавт. представили метод обнаружения аномалий, в котором вместо самих токенов используется упрощение, учитывающее только частоты трех наборов элементов: символов, чисел и специального символа. В представленном в данной статье подходе весь запрос анализируется без дальнейшего упрощения.

В работе [19] используется метод встраивания слов для представления URL-адресов, который состоит из трех основных шагов:

– для отделения аномалий от нормальных выборок применяется ансамблевая модель кластеризации;
– авторы используют word2vec для получения семантических представлений аномалий;
– мультикластерный подход группирует аномалии в определенные типы.

Подход word2vec, основанный на методологии статических вложений слов (word embeddings), исходит из гипотезы локальности — «слова, которые встречаются в одинаковых окружениях, имеют близкие значения». Близость в данном случае понимается широко, как то, что рядом могут стоять только сочетающиеся слова. Отметим основные недостатки такого подхода [10]: отсутствие информации о контексте, в котором используется слово; в модели не учитывается то, что слово может иметь различное значение в зависимости от контекста использования; не очень хорошо обрабатываются неизвестные и редкие слова.

В связи с этим в разработанной модели, которая представлена в данной статье, статические вложения (word2vec) были заменены глубокими контекстуальными представлениями, которые использовались для получения семантических представлений обычных данных и применялись в качестве входных данных для построения одноклассовой модели.

В [20] Ю с соавт. предлагают метод, использующий двунаправленную долговременную кратковременную память (Bi-LSTM) с механизмом внимания для моделирования HTTP-трафика. Это подход обучения с учителем, поскольку сеть Bi-LSTM обучают предсказывать, является ли запрос аномальным или нет. В [21] Цинь с соавт. предлагают модель, которая изучает семантику вредоносных сегментов в полезной нагрузке, используя рекуррентную нейронную сеть (RNN) с механизмом внимания. Полезная нагрузка преобразуется в последовательность скрытых состояний с помощью RNN, а затем используется механизм внимания для взвешивания скрытых состояний в качестве вектора признаков для дальнейшего обнаружения. Таким образом, они также могут использовать скрытое состояние сети в качестве признаков для второго классификатора. Авторы работы [21] изучают веса RNN, модели извлечения признаков, используя нормальные и ненормальные экземпляры.

В отличие от [21], был предложен метод обучения, который состоит в построении предварительно обученной модели с самоконтролем, используя только обычные данные. Основным недостатком метода, предложенного в статье [21], является высокий процент ложноположительных срабатываний (FPR). Снижение процента ложноположительных срабатываний при использовании указанных выше методов возможно, но при этом существенно снижается и процент детектирования нежелательного контента (TPR).

В работе [22] рассматривается модель, использующая многоуровневый автокодировщик (SAE) и сеть глубокого доверия в качестве методов обучения признаков, в которых на этапе обучения выбираются только обычные данные. Впоследствии в качестве классификаторов

применяются OCSVM, Isolation Forest и Elliptic Envelope. В этой работе функции HTTP извлекаются с использованием n-грамм, а затем применяются модели глубокого обучения для уменьшения размерности, генерируемой векторами n-грамм.

Разработанный метод позволяет напрямую работать с HTTP-запросом и избегать построения n-грамм, требующих больших объемов данных для правильного сбора статистики каждой модели.

Таким образом, основными задачами нашего исследования являются повышение процента выявленных атак (TPR), а также обеспечение низкого процента ложноположительных срабатываний (FPR).

2. Предлагаемое решение

Предлагаемое решение состоит из двух этапов. На первом этапе создается глубокая предобученная языковая модель, используя только обычные HTTP-запросы к веб-приложению. На втором этапе эта модель выступает в качестве средства извлечения признаков и обучается одноклассовый классификатор. Таким образом, каждое веб-приложение имеет свою собственную связку (как предварительно обученную языковую модель, так и одноклассовый классификатор).

В работе используются представления двунаправленного кодировщика с надежной оптимизацией из архитектуры Transformers (RoBERTa) [6]. На рис. 1 и 2 представлены компоненты предлагаемой архитектуры обучения.

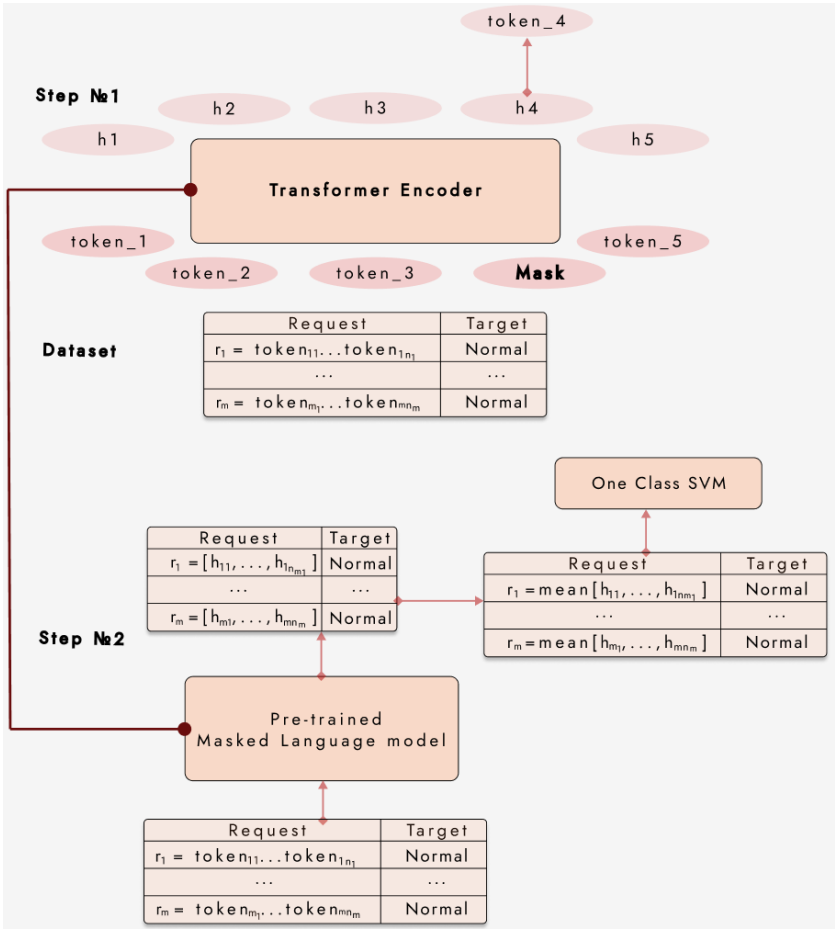

Рис. 1. Архитектура разработанной модели (цвет онлайн)

Fig. 1. The architecture of the developed model (color online)

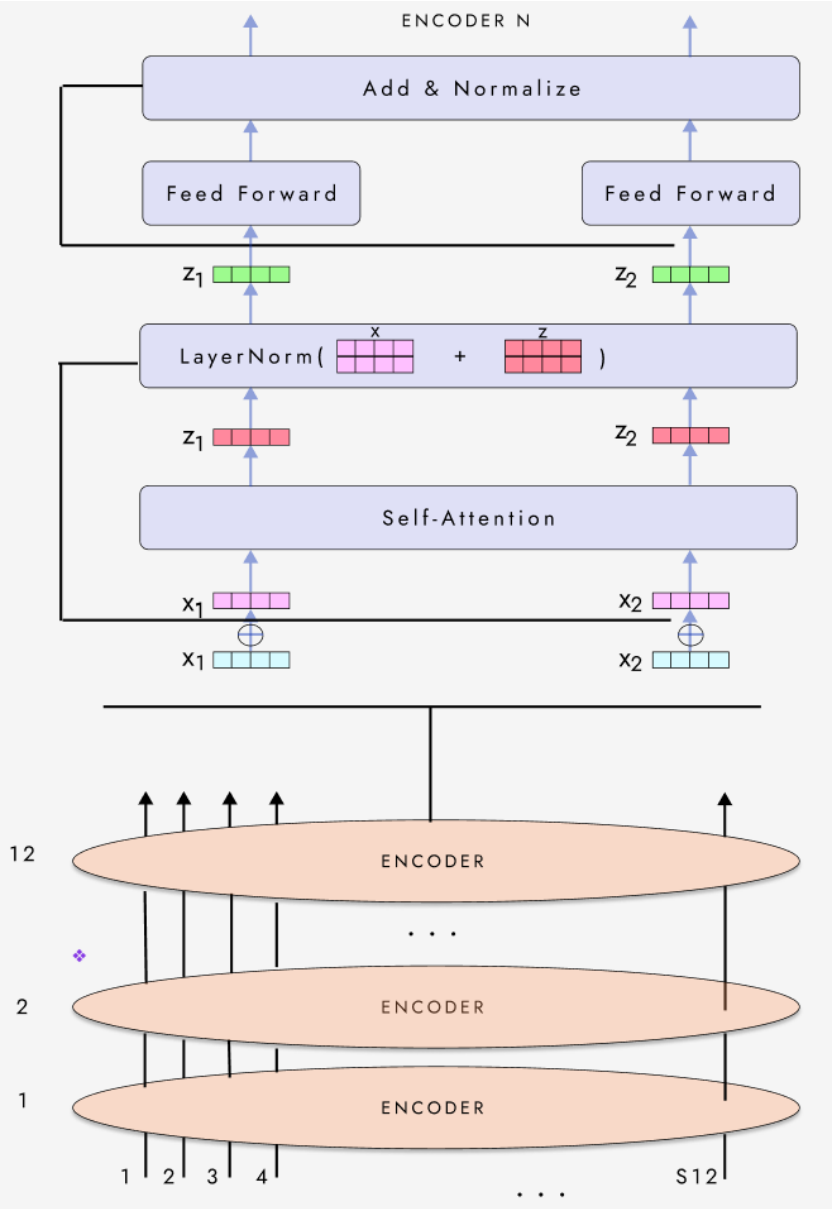

Рис. 2. Архитектура энкодера (цвет онлайн)

Fig. 2. Encoder architecture (color online)

На вход модели подается токенизированная версия HTTP-запроса, токенизация происходит BPE-токенайзером [23]. Архитектура сети, используемая для построения языковой модели, представляет собой многоуровневый двунаправленный Transformer Encoder [24]. Это основанная на внимании архитектура для моделирования последовательных данных, которая является альтернативой рекуррентным нейронным сетям (RNN) и способна фиксировать долгосрочные зависимости в последовательных данных. Этот токенизатор имеет словарь подслов из 50 000 единиц, который по-прежнему может кодировать любую входную строку без введения каких-либо неизвестных токенов.

Предлагаемая модель состоит из N уровней идентичных трансформеров, как показано в нижней части рис. 1. Каждый энкодер состоит из двух подслоев, первый слой содержит Multi-Head Attention механизм [24], а второй слой — нейронную сеть с прямой связью. Используются количество трансформеров $N = 12$, количество скрытых слоев H (они обозначаются h_1, .., h_n) и количество self-attention-heads $S = 12$. Обучение $BERT$-модели выполнялось с параметрами, предложенными в $BERT_{BASE}$ ($N = 12$, $H = 768$, $S = 12$).

Учитывая архитектуру модели, следующим шагом является определение стратегии обучения, т. е. цели обучения и механизмов обучения. В данном случае, чтобы изучить глубокое контекстуальное представление токенов, применяется подход к обучению с учителем. Случайным образом маскируются некоторые токены из ввода, а затем цель состоит в том, чтобы предсказать исходный замаскированный токен, основываясь только на его контексте. В [7] эту процедуру называют моделью маскированного языка (MLM). В отличие от шумоподавляющих автокодировщиков, эти модели предсказывают только замаскированные слова, а не восстанавливают весь ввод [7]. В попытке предсказать замаскированные токены модель должна иметь возможность извлекать некоторую информацию из языка, не только структурную информацию, но и некоторую семантическую информацию. Эта информация кодируется в весах слоев кодирования. Когда есть вектор признаков, упомянутый выше, то применяется хорошо известный классификатор OCSVM (One-Class Support Vector Machine), представленный в [25] с ядром радиальной базисной функции (RBF).

3. Эксперименты

Производительность предлагаемого метода анализируется с точки зрения True Positive Rate (TPR) и False Positive Rate (FPR). В проведенном эксперименте TPR и FPR показывают соотношение запросов, соответственно правильно и неправильно классифицированных как атаки. Для оценки предлагаемого метода использовались те же наборы данных, что и в работах [4, 5]. Набор данных CSIC 2010 представляет собой набор обычных и аномальных HTTP-запросов для веб-приложения, которое предоставляет функции для совершения покупок в Интернете. Набор данных содержит 36 000 действительных запросов на обучение, еще 36 000 — на тестирование и 25 000 запросов аномального трафика. Результаты представлены в таблице.

Полученный TPR и FPR для каждого датасета, %
Table. TPR and FPR received for each dataset, %

Параметр	ModSecurity OWASP CRS v3-PL 1	ModSecurity OWASP CRS v3-PL 2	ModSecurity OWASP CRS v3-PL 3	Одноклассовый классификатор из работы [5]	Разработанная модель на базе RoBERTa + OCSVM
Процент корректно классифицированных атак TPR	26.6	29.5	52.6	39.6	47.1
Процент ложноположительных срабатываний FPR	0	0	13.9	5.3	7.5

Оценка выполнялась для каждого из наборов данных с использованием 70% действительных запросов для обучения и остальной части набора данных (30% действительных и 100% атак) для тестирования.

Результаты работы авторской модели представлены в таблице столбцом RoBERTa + OCSVM. Тестирование осуществлялось на базе датасета CISC 2010, содержащего HTTP/1.1 запросы для обнаружения уязвимостей веб-приложений на базе аномалий, а также примеры нормального HTTP-трафика для обнаружения ложных срабатываний систем защиты.

Оценка выполнялась с разными уровнями паранойи (PL). В PL 1 используются правила, которые почти никогда не вызывают ложноположительных срабатываний (в идеале никогда, но мы должны признать, что это может произойти в зависимости от локальной настройки). На уровне PL 2 добавляются дополнительные правила, которые обнаруживают больше атак, но есть вероятность, что дополнительные правила также вызовут ложную тревогу по вполне легитимному HTTP-запросу.

На уровне паранойи PL 3 добавляется больше правил для определенных специализированных видов атак, такие правила подходят для тестирования уязвимостей довольно серьезных

систем, например онлайн-банкинга[5]. Это приводит к еще большему количеству ложноположительных срабатываний, для минимизации которых требуется детальный анализ работы системы экспертом по безопасности и последующая точная настройка системы.

ModSecurity с уровнями паранойи 1 и 2 не дает ложноположительных срабатываний за счет крайне низкого TPR. При более строгом уровне паранойи (PL 3) FPR составляет 13.9%, а TPR — 52.6%. Метод с использованием одноклассового идентификатора, предложенный в работе [5], позволяет детектировать менее 40% атак. Разработанный авторами метод дает низкий процент ложноположительных срабатываний FPR (7.5%) при высоком проценте детектирования атак — TPR 47.1% на используемом наборе данных.

Заключение

Разработанный метод позволяет использовать языковое представление HTTP-запросов на основе глубокого преобразования для решения проблемы обнаружения атак веб-приложений.

Использовался датасет CSIC 2010 для предварительной подготовки модели глубокого языка для HTTP-запросов. Был предложен и проанализирован двухэтапный подход к обучению, состоящий в том, чтобы сначала преобразовать HTTP-запрос в непрерывное пространство с помощью кодировщика-преобразователя, а затем применить OCSVM для отделения обычного трафика от атак.

Представленные результаты экспериментов превосходят ModSecurity с уровнем паранойи 1 и 2, а также результаты, представленные в [4], позже улучшенные в [5], по проценту успешно детектированных атак (TPR). Также следует отметить снижение в 1.8 раза процента ложноположительных срабатываний (FPR) по сравнению с ModSecurity с уровнем паранойи 3 при сопоставимом TPR.

Список литературы / References

1. Hacker A. J. Importance of web application firewall technology for protecting web-based resources. *ICSA Labs an Independent Verizon Business*, 2008, pp. 7. Available at: https://img2.helpnetsecurity.com/dl/articles/ICSA_Whitepaper.pdf (accessed December 28, 2022).
2. Sureda Riera T., Bermejo Higuera J. R., Bermejo Higuera J., Martinez Herraiz J. J., Sicilia Montalvo J. A. Prevention and fighting against web attacks through anomaly detection technology. A systematic review. *Sustainability*, 2020, vol. 12, iss. 12, art. 4945. https://doi.org/10.3390/su12124945
3. Betarte G., Martinez R., Pardo A. Web application attacks detection using machine learning techniques. *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Orlando, 2018, pp. 1065–1072. https://doi.org/10.1109/ICMLA.2018.00174
4. Betarte G., Gimenez E., Martinez R., Pardo A. Improving web application rewalls through anomaly detection. *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Orlando, 2018, pp. 779–784. https://doi.org/10.1109/ICMLA.2018.00124
5. Martinez R. *Enhancing web application attack detection using machine learning*. Montevideo, UdelaR – Area Informatica del Pedeciba, 2019. 82 p.
6. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. RoBERTa: A robustly optimized BERT pretraining approach. *ICLR 2020 Conference Blind Submission*. Addis Ababa, 2020. Available at: https://openreview.net/forum?id=SyxS0T4tvS (accessed January 15, 2023).
7. Devlin J., Chang M. W., Lee K., Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT 2019*. Minneapolis, 2019, pp. 4171–4186.
8. Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019, vol. 1, iss. 8, pp. 9.
9. Peters M. E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. New Orleans, 2018, pp. 2227–2237. https://doi.org/10.18653/v1/N18-1202

[5] *Folini C.* Working with Paranoia Levels. URL: https://coreruleset.org/20211028/working-with-paranoia-levels/ (дата обращения: 15.01.2023).

10. Mikolov T., Chen K., Corrado G., Dean J. Efficient estimation of word representations in vector space. *Computer Science*, 2013. arXiv:1301.3781v3 [cs.CL]. https://doi.org/10.48550/arXiv.1301.3781
11. Bengio Y., Ducharme R., Vincent P., Janvin C. A neural probabilistic language model. *Journal of Machine Learning Research*, 2003, vol. 3, pp. 1137–1155.
12. Olah C. Deep learning, NLP, and representations. *GitHub blog, posted on 2014*, July, 7. Available at: https://colah.github.io/posts/2014-07-NLP-RNNs-Representations/ (accessed January 15, 2023).
13. Luong M. T., Socher R., Manning C. D. Better word representations with recursive neural networks for morphology. *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, 2013, pp. 104–113.
14. Zou W. Y., Socher R., Cer D., Manning C. D. Bilingual word embeddings for phrase-based machine translation. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1393–1398.
15. Ethayarajh K. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, 2019, pp. 55–65. https://doi.org/10.18653/v1/D19-1006
16. Kruegel C., Vigna G. Anomaly detection of web-based attacks. *Proceedings of CCS*, 2003, pp. 251–261. https://doi.org/10.1145/948109.948144
17. Corona I., Ariu D., Giacinto G. HMM-Web: A framework for the detection of attacks against web applications. *Proceedings of ICC*, 2009, pp. 1–6. https://doi.org/10.1109/ICC.2009.5199054
18. Torrano-Giménez C., Pérez-Villegas A., Marañón G. Á. An anomaly-based approach for intrusion detection in web traffic. *Journal of Information Assurance and Security*, 2010, vol. 5, pp. 446–454.
19. Yuan G., Li B., Yao Y., Zhang S. Deep learning enabled subspace spectral ensemble clustering approach for web anomaly detection. *2017 International Joint Conference on Neural Networks (IJCNN)*. Anchorage, AK, USA, 2017, pp. 3896–3903. https://doi.org/10.1109/IJCNN.2017.7966347
20. Yu Y., Yan H., Guan H., Zhou H. DeepHTTP: Anomalous HTTP Traffic Detection and Malicious Pattern Mining Based on Deep Learning. *IET Information Security*. Singapore, 2020, vol. 1299. https://doi.org/10.1007/978-981-33-4922-3_11
21. Qin Z. Q., Ma X. K., Wang Y. J. Attentional payload anomaly detector for web applications. *International Conference on Neural Information Processing*. Springer, 2018, pp. 588–599. https://doi.org/10.1007/978-3-030-04212-7_52
22. Vartouni A. M., Teshnehlab M., Kashi S. S. Leveraging deep neural networks for anomaly-based web application firewall. *IET Information Security*, 2019, iss. 13, pp. 352–361. https://doi.org/10.1049/iet-ifs.2018.5404
23. Sennrich R., Haddow B., Birch A. Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, 2015, pp. 1715–1725. https://doi.org/10.18653/v1/P16-1162
24. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention is all you need. *Advances in Neural Information Processing Systems 30*, 2017. https://doi.org/10.48550/arXiv.1706.03762
25. Scholkopf B., Platt J. C., Shawe-Taylor J., Smola A. J., Williamson R. C. Estimating the support of a high-dimensional distribution. *Neural Computation*, 2001, iss. 13, pp. 1443–1471. https://doi.org/10.1162/089976601750264965

Поступила в редакцию / Received 28.01.2023
Принята к публикации / Accepted 02.02.2023
Опубликована / Published 30.08.2024