УДК 621.565.94:004.2

# ПРИМЕНИМОСТЬ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ЗАДАЧ КЛАССИФИКАЦИИ АТАК НА ВЕБ-СИСТЕМЫ. ЧАСТЬ 3

**К.В. Смирнова[1], А.О. Смирнов[2], В.М.Плотников[3]**

[1,3]Одесская национальная академия пищевых технологий, Одесса, Украина
ORCID: [1]0000-0002-3818-8083, [2]0000-0002-9459-6292,
E-mail: [1] smirnova.kathrin@gmail.com, [2] smyrnov.aleksandr.dev@gmail.com,[3] vmplotnik@gmail.com



*Анотация: Рассмотрена возможность применения машинного обучения для задач классификации вредоносных запросов к веб-приложению. Рассматриваемый подход исключает использование детерминированных систем анализа (например, экспертных), и строится на применении каскада нейронных сетей или же персептронов по приближенной модели к реальному человеческому мозгу. Основной замысел работы состоит в том, чтобы дать возможность описывать сложные векторы атак, состоящие из множеств признаков, абстрактными терминами для составления обучающей выборки, контроля качества распознавания и классификации каждого из слоев (сетей), участвующих в работе, с возможностью корректировать не всю сеть, а только малый ее участок, в обучение которого закралась ошибка или неточность. Дизайн разработанной сети можно описать как каскадную масштабируемую нейронную сеть.*

*В разработанной системе обнаружения вторжений использована трехслойная нейронная сеть. Слои возможно наращивать независимо друг от друга каскадами.*

*Во второй части [2] рассматривался вопрос минимизации ложных срабатываний средствами нейронной сети и ее архитектуры. Несомненно, выносить и обучать отдельные нейроны или подсети сети для обнаружения попыток обойти рассматриваемую систему обнаружения вторжений - это верное решение. Однако, следует упомянуть и подход, который позволяет повысить точность и уменьшить ложные срабатывания - токенизацию.*

Abstract: *The possibility of applying machine learning for the classification of malicious requests to a web application is considered. This approach excludes the use of deterministic analysis systems (for example, expert), and is based on the application of a cascade of neural networks or perceptrons on an approximate model to the real human brain. The main intention of the work is to enable to describe complex attack vectors consisting of feature sets, abstract terms for compiling a training sample, controlling the quality of recognition and classifying each of the layers (networks) participating in the work, with the ability to adjust not the entire network, but only a small part of it, in the training of which the error or inaccuracy crept in. The design of the developed network can be described as a cascaded scalable neural network.*

*The developed system of intrusion detection uses a three-layer neural network. Layers can be built independently of each other by cascades.*

*In the second part [2], the issue of minimizing false positives by means of a neural network and its architecture was considered. Undoubtedly, taking out and training individual neurons or subnets of the network to detect attempts to circumvent the intrusion detection system under consideration is the right decision. However, one should also mention the approach, which allows to increase the accuracy and reduce false positives - tokenization.*

*Ключевые слова:* Нейронная сеть, машинное обучение, система обнаружения вторжений, защита веб-приложений, информационная безопасность, токенизация.

Keywords: Neural network, machine learning, intrusion detection system, protection of web applications, information security, tokenization.

## Введение

Одной из основных задач информационной безопасности на сегодняшний день является обеспечение безопасности веб систем. Практически все доступные сайты регулярно подвергаются различного рода атакам. Большинство представленных на рынке систем обнаружения вторжений направлены на распознавание нецелевых атак. Целевые же

атаки, проводимые злоумышленником, а не ботом, распознать, порой, крайне сложно по причине невозможности прогнозирования всех возможных векторов атак и используемого инструментария.

В работе рассматривается применимость машинного обучения для задач классификации атак не веб системы. В описанной в [1][2] системе обнаружения вторжений используется трехслойная нейронная сеть. Слои возможно наращивать независимо друг от друга каскадами. В первом слое за каждый класс распознавания атаки отвечает своя сеть и корректность проверяется именно на этой сети. Для обучения этого слоя подобраны такие классы вещей, которые могут быть классифицированы однозначно как да или нет, то есть линейно разделимы. Таким образом, получен слой не просто нейронов, а их микросетей, который наилучшим образом может определить есть какой-то класс данных в запросе или нет.

Следующие слои не обучены на распознавание самих атак, они обучены тому, что совокупность атак рождает определенные угрозы. Это позволяет более тонко распознать попытки атакующего обойти систему защиты, а также классифицировать цель атаки, а не только ее факт. Простое наращивание слоев позволяет максимально минимизировать процент ложных срабатываний.

На данный момент времени среди предложенных на рынке систем предотвращения вторжений практически отсутствует адаптивность к неизвестным атакам. И выявление атаки происходит уже по факту ее совершения, а не на этапе ее возможного предотвращения.

### Цели и задачи

Цель работы - рассмотреть возможность применения машинного обучения для задач классификации вредоносных запросов к веб-приложениям. Рассматриваемый подход исключает использование детерминированных систем анализа (например, экспертных), и строится на применении каскада нейронных сетей или же персептронов по приближенной модели к реальному человеческому мозгу.

Задачами работы является повышение точности выявления атак на веб систему и уменьшение процента ложных срабатываний в рассматриваемой системе обнаружения вторжений.

### Материалы и методы

Для достижения цели целесообразно применить токенизацию. Токенизация предполагает разбиение текста на токены - в самом простом случае это просто слова. Применяя для разбиения простые регулярные выражения, можно потерять смысловую нагрузку.

### Теоретическая часть

Рассмотрим механизмы работы WAF (Web Application Firewalls. Этапы обработки входящего трафика в большинстве WAF одинаковы. Условно можно выделить следующие этапы:

- парсинг HTTP-пакета, который пришел от клиента;
- выбор правил в зависимости от типа входящего параметра;
- нормализация данных до вида, пригодного для анализа;
- применение правила детектирования;
- вынесение решения о вредоносности пакета (WAF или обрывает соединение, либо пропускает пакет дальше, на уровень приложения).

Все этапы, кроме правил детектирования, достаточно хорошо изучены и в большинстве фаерволлов одинаковы. Если проанализировать виды логик обнаружения атак в наиболее популярных WAF, то лидировать будут:

- регулярные выражения;
- окенайзеры, лексические аналзаторы;
- репутация;
- выявление аномалий;
- score builder.

Большинство же фаерволлов используют механизмы регулярных выражений для поиска атак. Обусловлено это и исторически (изначально WAF использовал регулярные выражения), и простотой подхода. Регулярные выражения выполняют поиск вредоносного паттерна в HTTP-параметре. Например, выражение (?i)(<script[^>]*>.*?) ищет XSS-инъекцию в теле запроса. Первая часть ((?i)) делает последующую часть выражения нечувствительной к регистру, вторая часть ищет открывающийся тег <script с произвольными параметрами внутри тега и произвольный текст после символа >. Однако, при таком подходе стоит вопрос обхода WAF за счет уязвимостей в правилах. Выделяют следующие типы обхода WAF, вытекающие из подобных уязвимостей:

- модификаторы, числовые квантификаторы и позиционные указатели;
- ошибки логики;
- особенности парсеров и опечатки;
- уязвимые регулярные выражения;
- использование новых техник эксплуатаций уязвимостей.

Помимо недостатка в виде возможности обхода WAF, использование регулярных выражений для поиска атак может приводить к большому количеству ложных срабатываний системы, что снижает ее эффективность.

Токенизация при лексическом анализе - это преобразование лексических единиц из последовательностей в определенные токены, которым проще обучить нейронную сеть. Такой подход важен, когда представление каких-то

символов или сочетаний символов необходимо принимать как единую смысловую единицу, и в другой последовательности, или же по отдельности, они могут нести иную смысловую нагрузку.

Если разложить любой запрос к web приложению на составляющие практичные для анализа, можно выделить три основные группы последовательностей: ключевые слова, ключевые символы (частота повторений), а также символьные и символьно-буквенные последовательности, которые несут определенную смысловую нагрузку только в совокупности, но не по отдельности.

При использовании такого подхода мы получаем систему распознавания вредоносных запросов очень высокой точности и качества. Однако, есть и другая проблема - проблема распознания недопустимого содержимого в ответах приложения.

Построив нейронную сеть методом, описанным в [1,2], мы получаем качественный и гибкий классификатор, который невозможно обмануть классическими методами. Однако, всегда существует опасность пропустить слабо коррелирующий признак атаки и выдать атакующему определенную часть информации.

Допустим, что параметр id уязвим к SQL injection. Чтобы среагировала система защиты и нейронная сеть нашла признаки атаки, атакующий должен начать саму атаку. Однако, определенные данные он может получить и просто нарушив форматирование запроса - передав в параметр символ ' или же ", таким образом сломав валидность строки запроса. Это, в свою очередь, вызовет определенную ошибку и, если подавление вывода не включено, вывод этой ошибки атакующему, что сообщит ему о наличии данной уязвимости, а также раскроет определенные данные.

Для решения данной проблемы необходимо контролировать не только то, что поступает на вход приложению, но и то, как приложение на это реагирует и что отдает пользователю.

**Практическая часть**

Чтобы избежать ложных срабатываний и исключить атаку на IDS посредством добавления комментариев или прочего контента схожего на раскрытие данных для провокации системы на блокировку легитимной страницы предлагается применить такой алгоритм действий.

Если сработал аномальный детектор, но ответ от нейронной сети, ответственной за вход, не показал ничего - запрос к приложению пропускается. После этого данные ответа проверяются вторым детектором, который ищет признаки исключений и вывода ошибок, а также сигнатуры запрещенного контента (например, файлы конфигурации). И если детектор что-то находит - блокирует ответ.

Помимо этого, ответ проверяется постоянно на наличие кодов ошибки от 400 до 500, и если находит - перекрывает ответ собственным, чтобы избежать утечку любой информации. Например, при ошибке подключения, если временно не доступен сервер с базой данных, в 500 ошибке может вернуться адрес базы, логин и пароль, которые были использованы для подключения, и сама ошибка, что, в свою очередь, никто не должен видеть, кроме разработчиков системы.

Если была выявлена уязвимость параметра и ее не смогла правильно распознать первая нейронная сеть, но ответ распознала вторая - это означает, что, возможно, мы имеем случай уязвимости, которую могут пытаться эксплуатировать в обход IDS. В этом случае используем атаки злоумышленника во благо. Так как мы имеем все признаки атаки, все аномальные значения, которые были указаны, мы можем на их основе выпустить виртуальный патч-правило, которое сразу будет блокировать запрос к определенному ресурсу и его функционалу определенного типа и содержимого, дав разработчику время на создание полноценного патча

Каким образом, с учетом вышеописанного, будет проектироваться сеть. Один из нейронов (сеть нейронов) обучим находить признаки ошибок, универсально. Это будут ключевые слова и какие-то явно коррелирующие с ошибками признаки. Вынесение этого в отдельный нейрон необходимо и для того, чтобы снизить риск false-positive срабатываний сети на признаках, которые коррелируют с типом ошибки, но не ее наличием как таковой.

Еще один нейрон обучим обобщенно для распознания тех типов данных, которые отдавать потенциально опасно. Это конфигурационные файлы, файлы с данными, которые содержат пароли или логины и так далее.

Второй слой обучается на классификацию и уточнение ошибок, раскрытие данных. Так же необходимо создать и обучить нейроны или небольшие сети для классификации косвенных признаков, по которым будет выявляться к какому типу можно отнести ошибку. Таким образом можно будет классифицировать какой тип атаки спровоцировал данную ошибку в случае, если первая сеть ничего не смогла выявить в запросе.

На основании этих данных можно построить набор тестов, которые необходимо будет произвести с параметрами, чтобы выпустить автоматический виртуальный патч.

В начале делается запрос с подставкой в него простых данных. Если ошибка сохраняется, то, скорее всего, это ложная тревога или же часть приложения функционирует неверно. Об этом, несомненно, стоит оповестить администратора. Но противодействовать этому уже за рамками задач IDS. Если же ошибка не воспроизводится без символов, но воспроизводится с ними - нужно выпустить правило, которое запретит ввод этих символов в любом запросе к данному ресурсу.

Может показаться, что выполняется двойная работа - система распознала вторым слоем ошибку, в патче нету необходимости. Однако, необходимость выпуска такого патча связана напрямую с тем, что найдя такую уязвимость атакующий может подставить в запрос данные так, что ошибки, вызванной в последствии, не будет, а вредоносный код выполнится на такие действия.

Обычно необходимо несколько запросов. Большая часть этих запросов будет отражена первой нейронной сетью, так как они будут содержать признаки атаки. Еще не малую часть закроет вторая нейронная сеть, которая видит результаты действий в выводе. Однако, не стоит забывать о том, что небольшое количество запросов, которые еще не являются атакой в строгом понимании этого слова, могут дать дополнительные данные в руки атакующего или произвести небольшие изменения. Именно от таких случаев нужно защитить пользователя выпуском патча.

**Заключение**

В работе систем обнаружения вторжений важным критерием является высокая точность распознавания вредоносного запроса. Несмотря на полученные в [2] качественный классификатор, всегда существует опасность пропустить слабо коррелирующий признак атаки и выдать атакующему определенную часть информации. Для решения данной проблемы необходимо контролировать не только то, что поступает на вход приложению, но и реакцию приложения на полученный запрос.

Предложенный алгоритм позволяет избежать ложных срабатываний системы и исключить атаку на нее за счет дополнительной проверки данных детектором, который ищет признаки исключений и вывода ошибок, а также сигнатуры запрещенного контента, а также проверки на наличие кодов ошибки от 400 до 500.

**Литература**

[1] Smirnova K., Smirnov A., Olshevska O. MACHINE LEARNING IMPLEMENTATION FOR THE CLASSIFICATION OF ATTACKS ON WEB SYSTEMS. PART 1 //Автоматизация технологических и бизнес-процессов. – 2017. – Т. 9. – №. 2. – С. 3-6.

[2] Smirnova K., Smirnov A., Plotnikov V. MACHINE LEARNING IMPLEMENTATION FOR THE CLASSIFICATION OF ATTACKS ON WEB SYSTEMS. PART 2 //Автоматизація технологічних та бізнес-процесів. – 2017. – Т. 9. – №. 3.

[3] Мельников В. Г., Трифанов А. В. Методы обхода межсетевых экранов для приложений //Интерэкспо Гео-Сибирь. – 2017. – Т. 9. – №. 2.

[4] Yeole A. S., Meshram B. B. Analysis of different technique for detection of SQL injection //Proceedings of the International Conference & Workshop on Emerging Trends in Technology. – ACM, 2011. – С. 963-966.

**References**

[1] K. Smirnova, A. Smirnov, and V. Plotnikov, “Machine Learning Implementation For The Classification Of Attacks On Web Systems. Part 2,” *Автоматизація технологічних і бізнес-процесів*, vol. 9, no. 3, 2017.

[2] K. Smirnova, A. Smirnov, and V. Plotnikov, “Machine Learning Implementation For The Classification Of Attacks On Web Systems. Part 2,” *Автоматизація технологічних і бізнес-процесів*, vol. 9, no. 3, 2017.

[3] S. Yeole and B. B. Meshram, “Analysis of different technique for detection of SQL injection,” *Proceedings of the International Conference & Workshop on Emerging Trends in Technology - ICWET 11*, 2011.

УДК 62-933.6:004

# AUTOMATIC CONTROL OF PARAMETERS OF A NON-STATIONARY OBJECT WITH CROSS LINKS

**Pavlov A.I.**

Odessa National Academy of Food Technologies, Odessa





*Abstract: Many objects automatic control unsteady. This is manifested in the change of their parameters. Therefore, periodically adjust the required parameters of the controller. This work is usually carried out rarely. For a long time, regulators are working with is not the optimal settings. The consequence of this is the low quality of many industrial control*