УДК 004.056.4

ТЕСТИРОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧЕ КЛАССИФИКАЦИИ HTTP ЗАПРОСОВ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ TF-IDF

М. Т. Нгуен

Академия ФСО России

Поступила в редакцию 27.09.2019 г.

Аннотация. В настоящее время отмечается увеличение числа атак на информационные системы и их качество. Каждая атака может нарушать конфиденциальность, целостность и доступность информации. Большинство из них преследует финансовую выгоду, особенно веб-атаки, так как они являются самыми распространёнными по причине использования веб-приложения многими компаниями. Поэтому задача защиты личных данных является главной для всех организаций и компаний, решение которой требует использования систем обнаружения и предотвращения атак и межсетевого экрана. Эти средства используют следующий набор методов обнаружения атак: метод белого-чёрного списка, метод обнаружения атак по сигнатуре, метод обнаружения аномалий, и все они защищают веб-приложения на сетевом уровне. Так как современная сложная атака на веб-приложения чаще всего происходит на прикладном уровне, в виде HTTP/HTTPS запросов к сайту, у традиционных средств крайне ограничены возможности для обнаружения атак и широкого применения методов машинного обучения во многих областях информационной безопасности. В статье дается краткий обзор популярных атак на Веб-приложения, методов машинного обучения и их тестирование в задаче обнаружения атак на веб-приложения путём классификации HTTP запросов. Также приводятся выводы о эффективности применения методов машинного обучения к данной задаче. Целью исследования является повышение точности обнаружения атак на веб-приложения на основе применения методов машинного обучения и анализа атрибутов HTTP запросов в межсетевом экране для веб-приложения.

Ключевые слова: внедрение операторов SQL, XSS, отказ в обслуживании, CSRF, сигнатурный метод, метод обнаружения аномалий, метод машинного обучения.

ВВЕДЕНИЕ

В эпоху цифровых технологий большинство пользовательских приложений, таких как электронная почта, веб-сайты электронной коммерции, управление приложениями для «умного дома» и т. д., выполняются через Интернет. Параллельно с разработкой технологии прикладного программного обеспечения уязвимость самого программного обеспечения и информационной системы также появляется с большей частотой, чем раньше. Злоумышленники используют уязвимости нулевого дня (zero-day), чтобы атаковать информационные системы предприятий и организаций с злонамеренными целями. Атаки на веб-приложения открывают перед ними широкие возможности: доступ к внутренним ресурсам компании, личной информации, нарушение функционирования приложения или обход бизнес-логики – практически любая атака может принести финансовую выгоду для злоумышленника и убытки, как финансовые, так и репутационные, для владельца веб-приложения.

Кроме того, под угрозой находятся и пользователи веб-приложений, поскольку успешные атаки позволяют похищать учетные данные, выполнять действия на сайтах от лица пользователей, а также заражать рабочие станции вредоносным ПО.

Основными целями кибератак являются финансовые компании, такие как банки и компании электронной коммерции. В допол-

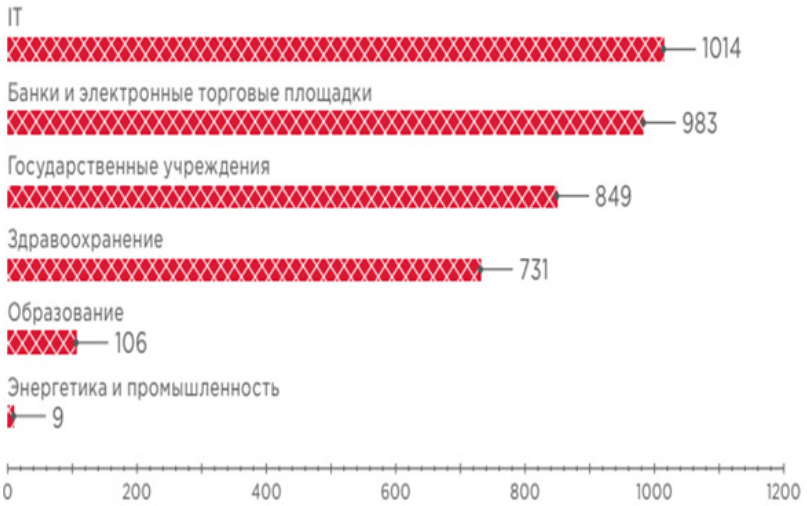

Рис. 1. Среднее число атак в день на веб-приложения одной компании

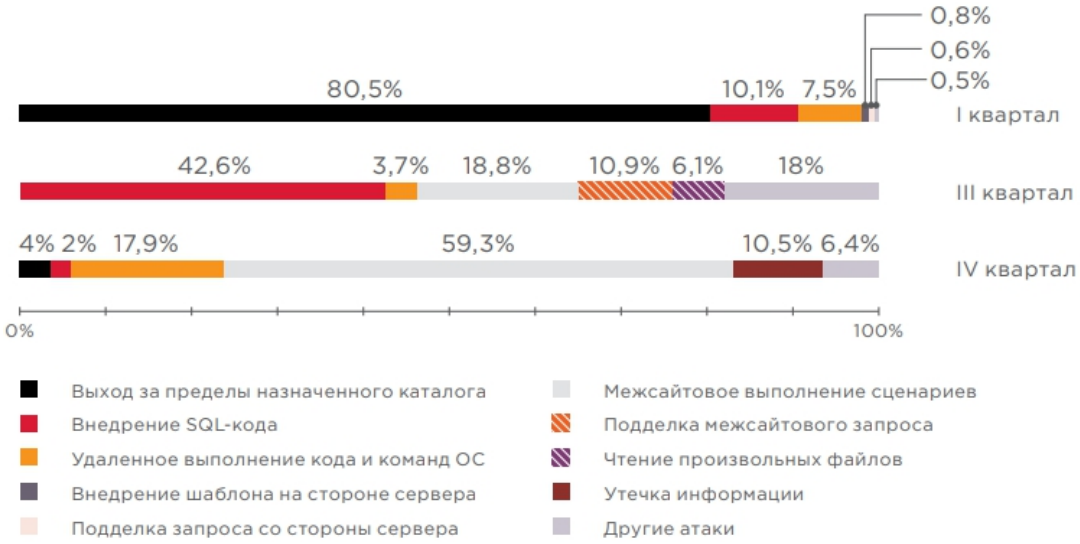

Рис 2. Топ-5 атак на веб-приложения банов и электронные торговые площадки 2017

нение к нарушению ежегодной деятельности повседневных организаций злоумышленники также имеют целью кражу информации об учетных записях пользователей для присвоения кредитных карт клиентов и банковских счетов. Поэтому защита юридических клиентов организаций становится более важной.

1. Анализ распространенных атак на веб-приложения

В этом пункте будут представлены девять самых популярных атак, совершенных в 2017 году [1]. Среди часто встречающихся атак на веб-приложения можно выделить следующие: «Межсайтовое выполнение сценариев» (39.1 % использования), «Внедрение операторов SQL» (24.9 % использования), «Выход за пределы назначенной директории» (6.6 % использования соответственно).

a. Внедрение операторов SQL

SQL-инъекция [2, 3] больше не является новой концепцией, но все же является одним из наиболее распространенных типов сетевых атак. SQL-инъекция – это метод, который использует уязвимости в запросах для получения данных небезопасных веб-сайтов в Интернете, что является очень популярным методом атаки. Его успех также относительно высок.

Инъекция SQL (для краткости называется SQLi) организована путем отправки вредоносных команд SQL на серверы баз данных с помощью запросов, разрешенных вашим сайтом, таких, как команды входа в систему. Любые входные данные сайта организаций

(теги ввода, строки запроса, файлы cookie и т. д.) могут быть использованы для отправки вредоносного кода.

Существуют распространенные типы ошибок SQL-инъекций:

• Неправильное обращение: Ошибки внедрения SQL такого типа обычно возникают из-за того, что программист или пользователь неясно определяет ввод данных или не выполняет этап проверки и фильтрации типа входных данных. Это может произойти, когда числовое поле используется в запросе SQL, но у программиста отсутствует проверка ввода для проверки типа данных, которые пользователь вводит как число.

• Ошибка конфигурации СУБД на сервере: иногда уязвимости могут существовать в программном обеспечении базы данных сервера, как в случае с функцией mysql_real_escape_string () серверов MySQL. Это позволит злоумышленнику выполнить успешную атаку SQL-инъекцией на основе необычных символов Юникода, даже когда ввод завершается.

• Изменение значения условия запроса: Этот тип ошибки позволяет злоумышленнику изменить значение условия в запросе, что искажает отображение приложения, содержащего эту ошибку.

• Время запаздывания: Этот тип ошибки внедрения SQL существует, когда время обработки одного или нескольких запросов SQL зависит от введенных логических данных или процесс обработки запросов механизма SQL занимает много времени. Злоумышленники могут использовать этот тип ошибки SQL-инъекции, чтобы определить точное время загрузки страницы, когда введенное значение верное.

Ошибки SQL-инъекций происходят из-за небезопасного программирования, поэтому лучшим решением является то, что программистам нужно быть осторожными при разработке веб-приложений. Для ошибок внедрения SQL специалист может использовать метод, применяя Prepare Statement для исправления, тогда входные данные от пользователя не будут выполняться в запросе. Для каждого конкретного языка и базы данных будут разные способы применения. Что касается атаки SQL-инъекцией в предложении ORDER BY, поскольку местоположение этого предложения не может использовать оператор Prepare, то для исправления атаки необходимо использовать действительный метод белого списка.

б. Выполнение команд ОС

Внедрение команд ОС (также называемое внедрением оболочки) – это уязвимость в веб-сайтах, которая позволяет злоумышленнику выполнять произвольные команды операционной системы (ОС) на сервере, на котором выполняются определенные службы. Злоумышленник может использовать эту уязвимость для использования, извлечения информации, передачи атак на другие системы внутри организации.

Многие случаи внедрения команд ОС являются слепыми уязвимостями. Это означает, что выходные данные не будут возвращены в ответе HTTP. Поэтому результат не будет отображаться на экране.

Задержки могут быть использованы для выявления слепых уязвимостей. Это вызовет задержку, позволяющую администратору подтвердить, была ли команда выполнена или нет, основываясь на времени, которое требуется приложению для ответа. Команда ping – эффективная команда для этого, так как она позволяет администратору системы указать пакет ICMP для отправки и время, необходимое для выполнения команды.

Уязвимость этого типа появляется в плагине WordPress DZS-VideoGallery (CVE: 2014 –9094), Gemitel 3.50 – Удаленное включение файлов/внедрение команд (CVE: 2004 – 1934) и т. д.

Самый эффективный способ предотвращения опасных команд – это прекратить использование команд. То есть никогда не вызывать команды ОС на уровне приложений. В некоторых случаях существуют разные способы выполнения необходимых функций с использованием API на более безопасной платформе. Если нельзя избежать использования команд ОС, необходимо выполнить строгую проверку подлинности ввода:

• проверка входных значений;

• необходимость приёма только данных в виде чисел;

• использование ввода только данных с буквенно-цифровых символов, без специальных символов, пробелов...

в. Выход за пределы назначенной директории

Целью атаки типа Path Traversal [4] является получение доступа к файлам и каталогам, которые расположены вне пределов, определенных конфигурацией (web root folder).

Злоумышленник часто использует в своих запросах последовательности типа «.../», чтобы попасть в корневой каталог. Для всех органов надлежащий контроль доступа к контенту сайта является ключевым фактором в работе защищенного сервера. Обратный путь в каталогах – это эксплойт HTTP, который позволяет злоумышленникам получать доступ к ограниченным каталогам, выполняя команды вне корневого каталога веб-сервера.

Веб-серверы обеспечивают два основных уровня механизма безопасности:

• список контроля доступа (ACL);

• корневая директория.

Список контроля доступа используется во время аутентификации. Это список, который администратор сервера использует для определения того, какие пользователи или группы пользователей могут получать доступы, такие, как изменение или выполнение определенных файлов на сервере, а также другие разрешения.

Корневая директория – это специальная директория в файловой системе сервера, в которой доступ пользователей ограничен. Пользователи не могут получить доступ к чему-либо в этой директории. Например, корневая директория IIS по умолчанию в системе Windows – C: **\Inetpub\wwwroot**, и с встроенными правилами пользователи не могут получить доступ к C: **\Windows**, но могут получить доступ к C: **\Inetpub\wwwroot\news** и любому файлу в этой директории (при условии, что пользователь действителен в ACL).

Во-первых, администратор системы устанавливает веб-сервер с последней версией программного обеспечения. Во-вторых, администратору надо установить фильтр для любого пользовательского ввода.

г. Межсайтовое выполнение сценариев

Межсайтовое выполнение сценариев (Cross-site Scripting или XSS) [5] является распространенной уязвимостью в веб-приложениях. Чтобы воспользоваться уязвимостью XSS, злоумышленник вводит вредоносный код через скрипты, чтобы выполнить их на компьютерах пользователей. Как правило, атаки XSS используются для обхода контроля доступа и олицетворения пользователей. Сам программный код обычно пишется на HTML/JavaScript, но может быть также перенесен в VBScript, ActiveX, Java, Flash, или на любую другую поддерживаемую браузерами технологию.

XSS-уязвимость позволяет внедрить в генерируемую и затем передаваемую пользователю страницу формата HTML некий произвольный код, порой весьма вредоносный. Когда злоумышленник добивается, чтобы браузер пользователя выполнил его программный код, этот код будет запущен в безопасной среде сервера веб-сайта. С этим уровнем привилегий программный код вполне способен прочитать, изменить или передать важные данные, доступные браузеру. Сложность этой атаки состоит в том, что алгоритм фильтрации входящих данных не должен создавать необоснованных ограничений легальным пользователям, но в то же время должен делать невозможной XSS атаку со стороны злоумышленника.

д. Отказ в обслуживании

«Отказ в обслуживании» (DDoS) [6] – один из популярных типов атак на Веб-приложения. Количество атак данного типа в третьем квартале 2017 г. увеличилось на 8 % по сравнению со вторым кварталом 2017 г.

DDoS-атака – это тип атаки, при котором злоумышленник делает систему непригодной для использования или существенно замедляет работу системы для обычных пользователей, перегружая ресурсы системы. Хотя атака DDoS не может получить доступ к фактическим данным системы, но она может нарушить работу служб, предоставляемых системой. При атаке на систему будут использоваться самые слабые уязвимости системы.

DDoS-атаки могут использоваться для сокрытия других сетевых атак. Когда веб-сайт органа находится под атакой, внутренняя и внешняя группа специалистов по информационной безопасности обычно фокусируется на закрытии портов этих сайтов, очистке трафика и возобновлении его работы. Эти большие усилия предоставляют огромную возможность для других опасных атак таких как SQL-инъекция.

е. Подключение локальных файлов

В языке программировании PHP имеет команды include, require, include_once, require_one, которые позволяют текущему файлу вызывать другой файл, которые являются средой рождения LFI уязвимости. LFI-уязвимости (Local File Inclusion) позволяют злоумышленникам подключать внутренние файлы на сервере, например, файлы: passwd, php.ini, access_log, ... (знать конфиденциальную информацию) в зависимости от уровня безопасности сервера. Причиной этой ошибки является то, что при использовании вышеуказанных команд программист снова вызывает файл для открытия через переменную.

Эти переменные либо еще не инициализированы, либо определяются пользователем. Ошибки LFI часто связаны с ошибками загрузки. Злоумышленник загружает файл, содержащий код php на сервере, не обязательно тип файла .php. Затем злоумышленник использует LFI-уязвимость, чтобы прочитать содержимое загруженного файла. Когда сервер читает эти файлы, при обнаружении php-кода эти коды выполняются и, таким образом, реализуются намерения злоумышленника.

Для защиты веб-приложений, написанных на языке программирования PHP, программисту надо ограничить использование переменных в функциях (**include** и **require**), если они используются, то эти переменные полностью и правильно объявлены.

ё. Внедрение внешних сущностей XML

Как известно, PHP является распространенным интерпретируемым языком общего назначения с открытым исходным кодом. Уязвимость была обнаружена во встроенных классах PHP SoapClient и SoapServer. В PHP разрешены внешние сущности при обработке SOAP wsdl-документов, что позволяет атакующему читать произвольные файлы.

ж. Загрузка произвольных файлов

Уровень опасности этой уязвимости очень высок. В системе “eXtreme File Hosting” уязвимость позволяет удаленному пользователю выполнить произвольный PHP сценарий на целевой системе. Уязвимость существует из-за ошибки при обработке загружаемых файлов, содержащих несколько расширений.

Данная уязвимость может привести к выполнению произвольного кода и/или отказу в обслуживании.

з. Подделка межсайтовых запросов

CSRF (Cross-Site Request Forgery) – это атака с использованием аутентифицированных учетных данных пользователя на другой вебсайт. Веб-приложения работают, получая от пользователя команды HTTP, а затем исполняют их. Злоумышленники используют метод CSRF, чтобы обмануть браузер пользователя для отправки команд http веб-приложениям. Если сеанс пользователя не истек, команды злоумышленника будут выполняться с правами аутентификации пользователя.

CSRF очень редко появляется среди CVE (распространённых уязвимостей и опасностей) – менее 0.1 % в 2008 году, но на самом деле это «спящий гигант». CSRF остаётся важным вопросом безопасности. Хотя существует огромное количество атак на веб-приложения, известно множество способов обнаружения атак для защиты Веб-приложения.

Исследование атак на веб-приложения за 2017 и 2018 года показано, что злоумышленники активно атакуют веб-приложения, преследуя при этом разные цели: прямую кражу денежных средств, получение финансовой выгоды путем вымогательства, проникновение во внутреннюю инфраструктуру, политические цели, шпионаж и т. д. Любое веб-приложение, даже не являющееся непосредственной целью киберпреступников, может подвергнуться атаке. Поэтому проблема защиты веб-приложения организаций от атак станет ещё более актуальной.

В пункте 2 будут рассмотрены главные способы обнаружения атак на сегодняшний день.

2. Анализ результатов предшествующих работ

Во многих системах обнаружения атак (СОА) и межсетевых экранах для веб-приложения чаще используются следующие методы: сигнатурные методы [7], методы обнаружения аномалий [8], методы с использованием машинного обучения [9, 10].

а. Сигнатурные методы

Как и другие программы сканирования вирусов на основе сигнатур, большинство СОА пытаются обнаружить атаки на базы данных по признакам атаки. Когда злоумышленник пытается использовать известную уязвимость, СОА пытается поместить ее в свою базу данных. Например, Snort, бесплатный продукт на основе сигнатур, разработанный как для Unix, так и для Windows.

Поскольку это программное обеспечение с открытым исходным кодом, Snort способен разрабатывать базу данных сигнатур быстрее, чем любой другой механизм базы данных. Подпись Snort используется во всем продуктах информационной безопасности, от коммерческого межсетевого экрана до промежуточного программного обеспечения, такого как Hogwash.

Сигнатурные методы основаны на специальных структурах из изучающих атак. Системный администратор определит поля в атаке и на основании этого напишет правила принудительного применения в системе для реагирования на подобную атаку.

б. Методы обнаружения аномалий

Метод обнаружения аномалий – выявление аномалий, связанных с установлением базовой основы нормальной работы системы или поведения в системе, а затем оповещение администратору о возникновении отклонений. Трафик в сети меняется незначительно в нормальном состоянии работы системы.

Однако некоторые сети имеют необычные структуры, особенно военные или разведывательные сети, и, с другой стороны, действия, которые происходят на сервере, могут быть неуправляемыми. Следует отметить, что системный администратор хочет разделить события СОА на основе ненормальных событий (в отличие от известного описания движения) и ненормальных протоколов событий (отклоняться от стандартов сетевого протокола).

Некоторые эффективные модели обнаружения поведенческого активности включают в себя:

- статистическую модель;
- модель, основанную на теории информации;
- модель кластера;
- модель классификации.

в. Подходы с использованием методов машинного обучения

Система обнаружения атак выявляет несанкционированный доступ к компьютерным системам и их эксплуатацию, отслеживая ненормальную активность пользователя, основываясь на установлении правил или использовании команды онлайн-прогнозирования.

Однако эти меры оказались неэффективными, дорогостоящими, ненадежными и неспособными обновить себя, чтобы обнаружить новые атаки. Другим подходом, который преодолевает вышеуказанные ограничения и все больше демонстрирует превосходство, является применение методов машинного обучения с использованием множества различных методов. В этом пункте будут рассмотрены некоторые главные методы машинного обучения для сравнения работы этих методов в задаче классификации.

Байесовская сеть

Одним из наиболее часто используемых методов машинного обучения для обнаружения вторжений является Байесовская сеть. Байесовская сеть [11, 12] – это модель, которая кодирует вероятностные отношения между рассматриваемыми событиями (переменными) и предоставляет некоторый механизм для вычисления условных вероятностей их наступления.

Частный случай этой модели – наивный байесовский классификатор (Байесовский метод) со строгими предположениями о независимости входных переменных. Алгоритм наивной Байесовской классификации – это группа простых классификаций вероятно-

стей, основанных на теореме Байеса, предполагающей независимость между атрибутами. Даже если эти атрибуты связаны друг с другом, то этот метод считает, что атрибуты не зависят друг от друга.

Исследуя наивный Байесовский классификатор можно отметить следующие:

• наивные байесовские классификаторы часто используются в задачах классификации текста;

• алгоритм имеет быстрое время обучения и тестирования. Это связано с предположением независимости между атрибутами, если класс известен;

• наивные байесовские классификаторы дают лучшие результаты, чем логистическую регрессию при меньшем количестве обучающих данных, если предположение о независимости выполнено (основано на характере данных);

• алгоритм может работать с векторами признаков, которые являются непрерывными частями (с использованием гауссовского наивного Байесовского алгоритма), а остальные в дискретной форме (с использованием многочлена или Бернулли);

• при использовании мультиномиального наивного Байесовского сглаживания часто используется сглаживание Лапласа, чтобы избежать того факта, что компонент в данных теста не появляется в данных обучения.

Нейронная сеть

Искусственная нейронная сеть [13, 14] – это модель обработки информации, которая смоделирована на поведении нервной системы организма, включая большое количество нейронов, установленных для обработки информации. Искусственная нейронная сеть подобна человеческому мозгу, обученному на опыте (посредством обучения), способному хранить опыт знаний (знания) и использовать эти знания при прогнозировании неизвестных данных (невидимые данные).

В работе [15] проводится сравнительный анализ возможностей искусственной нейронной сети и метода дерева решений для решения задач выявления компьютерных атак. Исследователи приходят к выводу, что искусственная нейронная сеть эффективна для обобщения и малопригодна для обнаружения новых атак, в то время как деревья решений эффективны для решения обеих задач.

В задачах классификации нейронные сети всегда дают хорошие результаты, когда количество входных параметров ограничено по сравнению с другими методами машинного обучения. В конкретной задаче классификации атак с двумя различными классами (атак и без атак) метод опорных векторов оказался доминирующим.

Метод k-ближайших соседей

Методом k-ближайших соседей (k-nearest neighbor, k-NN) является один из самых простых алгоритмов машинного обучении. При обучении этот алгоритм ничего не изучает из обучающих данных, каждый расчет выполняется, когда ему необходимо предсказать результат новых данных.

Для k-NN в задаче классификации метка новой точки данных выводится непосредственно из ближайших **k** точек данных в обучающем наборе. Метка тестовых данных может быть определена путем сравнения между ближайшими точками (ближайшими соседями), или она может быть выведена с помощью различного веса для каждой точки из множества ближайших точек.

Преимущества метода заключаются в следующем:

• простота прогноза результата новых наборов данных;

• отсутствие необходимости предполагать какое-либо распределение классов.

Недостатки метода k-NN:

• острая чувствительность k-NN к шуму, когда **k** мало;

• вычисление расстояния до каждой точки данных в обучающем наборе займет много времени, особенно для баз данных с большими измерениями и многими точками данных. С увеличением **k** сложность также возрастет;

• влияние хранения всех данных в памяти на производительность состояния.

Su Ming-Yang в своем исследовании [16] использовал смешанный подход: объединение генетического алгоритма и классификатор k-ближайших соседей для обнаружения атак типа «отказ в обслуживании». Этот под-

ход дает достаточно высокую точность обнаружения атак: 96.75 %.

Дерево принятия решений

Дерево принятия решения (ДПР) [17] является одним из самых популярных алгоритмов машинного обучения, доступных сегодня. Он используется в задачах классификации и регрессии.

Дерево принятия решения – это дерево, в котором каждый узел представляет характеристику (свойство), каждая ветвь представляет правило, а каждый лист представляет результат (конкретное значение или непрерывную ветвь).

По сравнению с другими методами анализа данных дерево решений имеет несколько преимуществ:

• простота объяснения работы метода дерева принятия решения;

• метод может обрабатывать как числовые данные, так и данные имена категорий;

• дерево принятия решения является моделью «белого ящика». Если в модели можно наблюдать данную ситуацию, то это можно объяснить с помощью булевой логики. Нейронные сети являются примером модели «черного ящика», потому что объяснение результатов слишком сложно для понимания;

• дерево принятия решения может быстро обрабатывать большие объемы данных. Персональные компьютеры могут использоваться для анализа больших объемов данных за достаточно короткое время, чтобы позволить стратегам принимать решения на основе анализа дерева решений;

Недостатки этого метода заключается в том, что:

• целевые атрибуты метода принимали только дискретные значения;

• результат классификации данных зависит от качества обучающей выборки;

• при решении задач классификации с большим числом классов этот метод не эффективен.

Метод опорных векторов

Метод опорных векторов является известным методом обучения с учителем, использующимся для задач классификации и регрессионного анализа.

Метод опорных векторов [18–20] относится к методам линейной классификации. В задачи классификации с двумя классами входит два множества точек, принадлежащих к двум разным классам, разделяющихся гиперплоскостью в этом пространстве. При этом гиперплоскость строится так, чтобы расстояния от нее до ближайших границ обоих классов (опорных точек) были максимальны, что обеспечивает наибольшую точность классификации.

После проверки некоторых наборов данных с применением метода опорных векторов следует отмечить его преимущества:

• метод опорных векторов применим для классификации текста, где размеры могут быть чрезвычайно большими;

• метод сводится к решению задачи квадратичного программирования в выпуклой области, которая всегда имеет единственное решение;

• возможность применять новое ядро, которое обеспечивает гибкость между линейными и нелинейными алгоритмами, что повышает производительность классификации.

Из недостатков можно отметить следующие:

• метод эффективен только для решения задач с двумя классами;

• чувствительность к шумам и стандартизации данных;

• отсутствие автоматического способа выбора функции ядра в случае линейной неразделимости классов.

МАТЕРИАЛЫ И МЕТОДЫ

Для проверки работы методов машинного обучения будет использован набор данных из нескольких источников данных средств защиты системы, таких как логовых файлов системы обнаружения и предотвращения вторжения, HTTP запросов (метод GET, POST) межсетевого экрана для веб-приложения и т. д. Академии криптографической техники г. Ханой и набора данных CSIC 2010 для классификации атак на HTTP запросах. Структура полного HTTP запроса (метод POST) показана на рис. 3.

```
POST http://localhost:8080/tienda1/publico/anadir.jsp HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)
Pragma: no-cache
Cache-control: no-cache
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=AE29AEEBDE479D5E1A18B4108C8E3CE0
Content-Type: application/x-www-form-urlencoded
Connection: close
Content-Length: 146
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
```

Рис. 3. Пример полного опасного HTTP запроса с методом POST

```
/top.php?stuff='uname >q36497765 #
/h21y8w52.nsf?<script>cross_site_scripting.nasl</script>
/ca000001.pl?action=showcart&hop=\"><script>alert('vulnerable')</script>&path=acatalog/
/scripts/edit_image.php?dn=1&userfile=/etc/passwd&userfile_name= ;id;
/javascript/mta.exe
/examples/jsp/colors/kernel/loadkernel.php?installpath=/etc/passwd\x00
/examples/jsp/cal/feedsplitter.php?format=../../../../../../../../../../etc/passwd\x00&debug=1
/phpwebfilemgr/index.php?f=../../../../../../../../../../etc/passwd
/cgi-bin/script/cat_for_gen.php?ad=1&ad_direct=../&m_for_racine=</option></select><?phpinfo();?>
/examples/jsp/cal/search.php?allwords=<br><script>foo</script>&cid=0&title=1&desc=1
/moodle/filter/tex/texed.php?formdata=foo&pathname=foo"+||+echo+db+4d+5a+50+00+02+00+00+00+04+00+0f+00+
/examples/jsp/colors/workarea/contentdesigner/ekformsiframe.aspx?id="><script>alert('nessus')</script>
/id;1627282494;fp;2;fpid;1/
/<img src="javascript:alert(cross_site_scripting.nasl);">.do
/bmeun223.exe?<meta http-equiv=set-cookie content="testhhwu=7044">
/htbin/windmail.exe
/<script>document.cookie="testrluj=1420;"</script>
/javascript/.passwd.jpg
/opensiteadmin/scripts/classes/databasemanager.php?path=http://192.168.202.118:8080/ghl9il?\x00
```

Рис. 4. Структура файла опасных HTTP запросов

Набор данных включает в себя такие атаки, как внедрение SQL, переполнение буфера, сбор информации, раскрытие файлов, внедрение CRLF, межсайтовое выполнение сценариев, подделка параметров.

Процесс тестирования состоит из двух фаз: фазы обучения и фазы обнаружения атак.

Фаза обучения состоит из трёх модулей.

• Модуль извлечения: по запросам, полученным от клиента, автор будем фильтровать части, необходимые для обработки запросов, включая URI, пути и параметры запросов, полезную нагрузку. При анализе полного HTTP запроса автор фокусируется на данных в красной рамке (рис. 3). После процесса извлечения данные законных и опасных запросов будут сохранены в соответствующих файлах (good_request.txt и bad_request.txt). Структура этих файлов представлена на рис. 4.

• Модуль векторного пространства используется для преобразования строковых данных в векторы, метод реализуется с помощью технологии tf-idf. Применим технологию tf-idf в нашей задаче, для каждой строки данных запроса автор найдет слова в составе запроса. Для вычисленя важности каждого слова t в запросе d в совокупности запросов D используются формулы:

$$tfidf(t,d) = tf(t,d) * idf(t,D) \qquad (1)$$

в формуле (1) вычисляются значения tf, idf как:

$$tf(t,d) = \frac{count(t,d)}{\sum_{v \in d} count(v,d)}, \qquad (2)$$

где $count(t,d)$: количество слова t в запросе d и $count(v,d)$: количество остальных слов в запросе d.

$$idf(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}, \qquad (3)$$

где $|D|$: количество всех рассмотренных запросов и $|\{d \in D : t \in d\}|$: количество тех запросов, содержащих слово t.

После процесса вычисления tf-idf будут преобразованы строковые данные запросов в векторы. Вектор формируется из значений tf-idf всех слов, содержащихся в этом запросе.

• Модуль обработки данных: автор использовал 6 главных методов машинного обучения для проверки их работы. После процесса обучения по каждому методу на заданном наборе данных все пороги будут сохранены в базе данных.

При реализации методов машинного обучения в межсетевом экране для веб-приложения фаза обнаружения состоит из трёх модулей, но имеет отличия в модуле обработки данных от соответствующего модуля фазы обучения. После классификации запроса межсетевой экран для веб-приложения не только сохраняет необходимые новые пороги в базе данных, но и решает блокировать или выполнять эти классифицированные запросы на сервере. Процесс работы межсетевого экрана для веб-приложения на фазе обнаружения представлен на рис. 5.

После исследования многих работ о методах машинного обучения в области информационной безопасности автор отметит, что эти методы имеют эффективные алгоритмы, широко распространены в настоящее время и применяются не только во многих системах обнаружения атак, но и в системах обнаружения вторжений.

Далее проверим рассмотренные методы машинного обучения, используя предложенный вышеуказанный процесс тестирования. Выбранный набор данных извлекается из 20000 опасных запросов и 50000 нормальных запросов, будем использовать перекрестную проверку (cross-validation) для оценки результатов.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

В процессе тестирования работы методов машинного обучения с данными из HTTP запросов все запросы принадлежат к двумя классам: либо C_1 (0 – не атака), либо C_2 (1 – атака). После работы модуля векторного пространства все данные, которые извлекаются из законных запросов, будут маркированы «0», а данные из опасных запросов будут «1». Эти векторы являются входами модуля классификации методов машинного обучения.

В данной работе проведен сравнительный анализ шести методов обнаружения компьютерных атак на Веб-приложения с целью выбора наиболее эффективного. Результат проверки набора данных с 80 % данных для обучения и 20 % данных для тестирования,

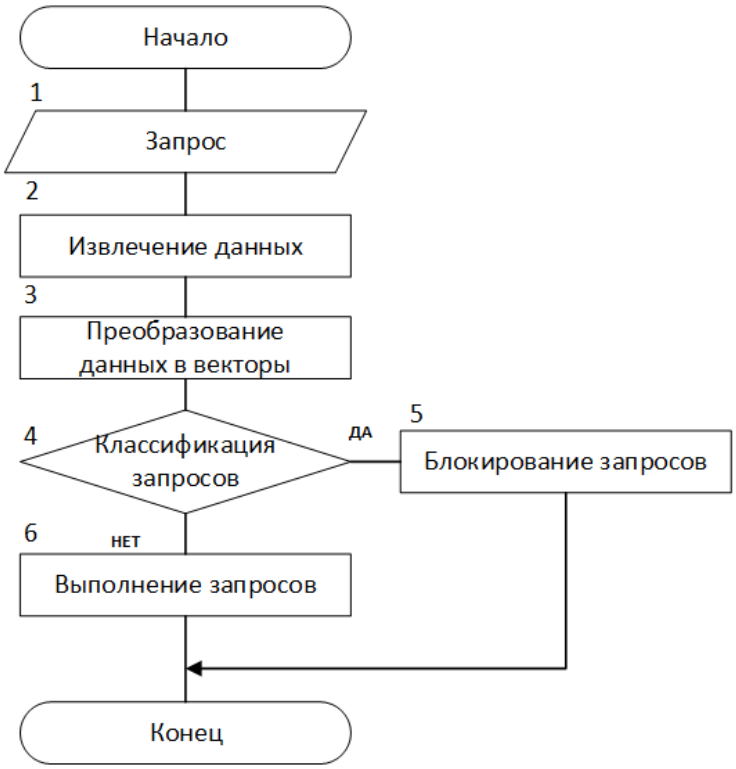

Рис. 5. Процесс работы межсетевого экрана для веб-приложения на фазе обнаружения

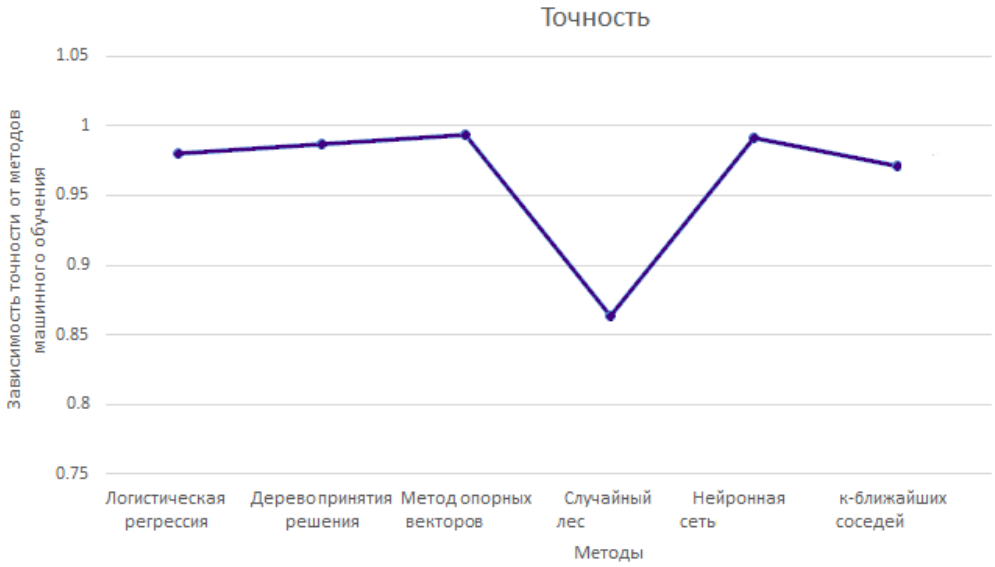

Рис 6. Точность классификации запросов методов машинного обучения для заданного набора данных

представлен на рис. 6 (с библиотекой машинного обучения scikit-learn).

Известно, что работа методов машинного обучения зависит от его параметров, поэтому в эксперименте были использованы различные параметры для каждого метода.

В некоторых методах будут использованы такие параметры, как:

• случайный лес (количество деревьев от 10 до 500);

• k-ближайших соседей (количество соседей от 2 до 100);

• дерево принятия решения (начальное число, используемое генератором случайных чисел равно нулю);

• логистическая регрессия (class_weight= 'balanced' – сбалансированный режим);

• метод опорных векторов: линейный метод опорных векторов и нелинейный метод опорных векторов с различными функциями ядра (полиномиальное однородное, полиномиальное неоднородное, радиальная базисная функция, радиальная базисная функция Гаусса, сигмоидная функция);

• нейронные сети используются перцептрон Розенблатта, многослойный перцептрон, рекуррентная нейронная сеть, и т. д., со сигмоидной функцией.

На рис. 6 показаны самые лучшие результаты классификации этих методов на заданном наборе данных (были реализованы с библиотекой scikit-learn на языке программирования Python v.2.7).

При проверке набора данных следует отметить, что:

• комбинация методов машинного обучения с технологией tf-idf даёт лучший результат точности классификации, чем применение классических вышеуказанных методов машинного обучения (на пункте 2).

• два метода: метод опорных векторов и нейронная сеть имеют высокую точность классификации для задачи двух классовой классификации;

• при увеличении количества рассматриваемых параметров два метода (метод опорных векторов и нейронная сеть) требуют высокой мощности вычисления;

• метод опорных векторов дает автору лучший результат классификации данных с двумя классами, чем применение нейронной сети;

• линейный метод опорных векторов и метод опорных векторов с Гауссовой функцией ядра имеют лучшие результаты, чем метод опорных векторов с остальными функциями ядра таких функций, как полиномиальное, сигмоид.

В этой работе автор предлагает новый процесс тестирования методов машинного обучения по использованию не только классических методов машинного обучения

выше, но и технологии tf-idf (оценка важности слова в HTTP запросе) для преобразования строковых данных в векторы формулами (1). Эти векторы состоятся из значений tf-idf каждого слова в запросе, и являются входом процесса классификации.

Так как технология tf-idf работает только со словами, автор рекомендует в будущих исследованиях использовать модуль анализа свойств параметров HTTP запросов и модуль оценки важности ключевых символов, характеризующих конкретные атаки.

ЗАКЛЮЧЕНИЕ

В эпоху цифровых технологий объем аналитических данных растет в геометрической прогрессии. Новое требование ставится за пределами точности задачи классификации, чтобы удовлетворить требованиям расширения системы с большими объемами данных и соответствия времени обнаружения и реакции под инцидентами системы.

Таким образом, использование соединения сигнатурных методов и методов машинного обучения делает системы обнаружения атак более интеллектуальными и автономными при обнаружении новых атак, поскольку статические методы могут быть обойдены атакующими.

Дальнейшая работа будет посвящена изучению систем обнаружения атак в облачной среде и повышению точности обнаружения атак, поскольку облачные вычисления являются основным сдвигом парадигмы компьютерных сетей.

СПИСОК ЛИТЕРАТУРЫ

1. *Килюшева, Е.* Атаки на Веб-сайты в 2016 году: боты и простые уязвимости / Е. Килюшева, Е. Гнедин // Кибербезопасность 2016-2017: от итогов к прогнозам. – 2017. – С. 38–42.

2. Mishra, S. SQL Injection Detection Using Machine Learning: Master's Theses and Graduate Research. – USA, 2019. – 51p.

3. Literature survey on detection of web attacks using machine learning / A. Gupta [et al.] // International Journal of Scientific Research Engineering & Information Technology. – 2018. – Vol. 3. – P. 1845–1853.

4. *Xiong J., Zolotov V.* Fast path traversal in a relational database-based graph structure. – 12/20/2018. – US Patent App. 16/038,498.

5. Cross-site Scripting. – Режим доступа: https://www.styler.ru/styler/xss/ (Дата обращения: 25.10.2017).

6. *Jalan, R., Kamat, G., Szeto R. W.* Mitigating tcp syn ddos attacks using tcp reset. –3 28/2019. – US Patent App. 16/198,981.

7. *Babiker, M.* Web application attack detection and forensics: A survey / Babiker Mohammed, Karaarslan Enis, Hoscan Yasar // 6th International Symposium on Digital Forensic and Security (ISDFS). – IEEE. 2018. – P. 1–6.

8. An improved payload-based anomaly detector for web applications / Jin Xiaohui [et al.] // Journal of Network and Computer Applications. – 2018. – Vol. 106. – P. 111–116.

9. *Ross, K.* SQL Injection Detection Using Machine Learning Techniques and Multiple Data Sources: Master's Theses and Graduate Research. – USA, 2018. – 27p.

10. *Silva, N.* Network Intrusion Detection Systems Design: A Machine Learning Approach / N. Silva, D. G. Gomes // Anais do XXXVII Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos. – SBC. 2019. – P. 932–945.

11. *Veni R, H.* Identifying Malicious Web Links and Their Attack Types in Social Networks/ H. Veni R, H. Reddy A, C. Kesavulu // International Journal of Scientific Research in Computer Science, Engineering and Information Technology. – 2018.– P.1060–1066.

12. *Fouladi, R. F.* Frequency based DDoS attack detection approach using naive Bayes classification / R. F. Fouladi, C. E. Kayatas, E. Anarim // 2016 39th International Conference on Telecommunications and Signal Processing (TSP). – IEEE. 2016. – P. 104–107.

13. *Atienza, D.* Neural analysis of http traffic for web attack detection / D. Atienza, A. Herrero, E. Corchado // Computational Intelligence in Security for Information Systems Conference. – Springer. 2015. – P. 201–212.

14. *Goyal, B.* A Competent Approach for Type of Phishing Attack Detection Using Multi-Layer Neural Network / B. Goyal, M. Bansal // Interna-

tional Journal of Advanced Engineering Research and Science. – 2017. – Vol. 4, no. 1. – P. 210–215.

15. *Bouzida, Y.* Neural networks vs. decision trees for intrusion detection / Y. Bouzida, F. Cuppens // IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM). Vol. 28. – 2006. – P. 29–37.

16. *Su Ming-Yang.* Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers/ Ming-Yang Su // Expert Systems with Applications. – 2011. – Vol. 38. – №. 4. – P. 3492–3498.

17. A novel hierarchical intrusion detection system based on decision tree and rules-based models / A. Ahmim [et al.] // arXiv preprint arXiv:1812.09059. – 2018. – 6p.

18. Anomaly-based web application firewall using HTTP-specific features and one-class SVM / Epp Nico [et al.] // Workshop Regional de Seguran¸ca da Informa¸c~ao e de Sistemas Computacionais. – 2017. – 11p.

19. *Tian, Z.* A Distributed Deep Learning System for Web Attack Detection on Edge Devices / Z. Tian [et al.] // IEEE Transactions on Industrial Informatics. – 2019. – P.99–107.

20. *Ye Jin.* A DdoS attack detection method based on SVM in software defined network / Ye Jin [et al.] // Security and Communication Networks. – 2018. – Vol. 2018.– P. 1–8.

Нгуен Мань Тханг – сотрудник, Академия ФСО России. E-mail: chieumatxcova@hotmail.com

TESTING MACHINE LEARNING METHODS IN THE PROBLEM OF CLASSIFYING HTTP QUERIES USING TECHNOLOGY TF-IDF

M. T. Nguyen

Academy FSO Russia

Annotation. Nowadays, the number of attacks on the information system is rapidly increasing not only in the amount but also in quality. Each attack violates the properties of confidentiality, integrity, and accessibility of information, so most attacks pursue financial gain, especially a Web attack because almost companies use web applications for their business. The issue of protecting personal data from these attacks is becoming a major issue for all organizations and companies. Thus, the need to use an intrusion detection system, an intrusion prevention system and a firewall to protect these data is relevant. These systems use many attack detection methods, such as the white list and blacklist, signature-based detection method, anomaly detection method, but they protect web applications at the network level. Since the modern complex attack on web applications most often occurs at the application level, in the form of HTTP/HTTPS queries to the website, where these traditional systems have extremely limited capabilities to detect attacks and widespread benefit of machine learning methods in many areas of information security. This article gives a brief overview of some types of popular attacks on Web applications, main machine learning methods and their testing in the task of problem detection web application attacks by classifying HTTP requests on Web Application Firewall. Also, this article is given a conclusion about the working of machine learning methods to identify the most effective method from them. Our future research aims to increase the accuracy of attack detection on web applications by using machine learning methods and analyzing attributes of HTTP requests on the web application firewall.

Keywords: SQL injection, XSS, DDOS, CSRF, signature method, anomaly detection method, machine learning method.

Nguyen Manh Thang – Contributor, Academy FSO Russia. E-mail: chieumatxcova@hotmail.com