Методы машинного обучения без учителя для поиска аномалий в веб-сессиях в реальном времени

А. Зяблицева

Аннотация **— Обнаружение аномалий в веб-сессиях— важная проблема, которая на данный момент активно развивается. Для обнаружения киберугроз, в основном, используется метод обнаружения аномалий на основе сигнатур. Такие методы эффективно обнаруживают известные атаки, но не могут обнаруживать неизвестные аномалии. Кроме того, существующие методы обнаружения аномалий, как правило, основаны на определении аномальных сессий на основе статических данных, имея размеченные тренировочные данные, а так же определяют сессию на основе IP-адреса. В данной работе рассмотрены основные способы сбора информации о поведении пользователя в сети, способы логирования и сбора информации о веб-сессиях, представлен и обоснован подход к поиску аномалий в веб-сессиях. Модель обнаружения аномалий в веб-сессиях использует алгоритма кластеризации без учителя, который не требует заранее размеченных данных, а для достижения наиболее эффективной кластеризации использует 12 полезных параметров сессий. Для представленной модели описана архитектура разработанного подхода. В выводах работы представлен результат тестирования и описаны дальнейшие направления развития проекта.**

Ключевые слова**—аномалии, кластеризация, веб-сессия, запрос, iDBSCAN.**

I. Введение

С ростом числа онлайн-платформ, интернет-сервисов и цифровых приложений увеличивается и вероятность веб-угроз, таких как фишинг, DDoS-атаки, вредоносные программы и многие другие.

Обнаружение аномального поведения пользователей в режиме реального времени становится все более важной задачей для обеспечения безопасности информационных систем и защиты конфиденциальности данных. Такие методы позволяют оперативно реагировать на потенциальные угрозы, идентифицировать атаки и вмешательства, минимизировать возможные потери.

Различные методы машинного обучения, такие как алгоритмы классификации, кластеризации, а также ансамблевые методы, используются для анализа данных о веб-сессиях и выявления аномалий в режиме реального времени. Автоматизированный мониторинг и постоянное обновление моделей позволяют быстро адаптироваться к изменяющимся условиям и новым видам угроз.

II. Веб-сессия

По мере развития веб-технологий кибератаки растут как по своему количеству, так и по серьезности своих намерений. Злоумышленники используют новые технологии, чтобы нанести больший ущерб обычным пользователям, использовав его во благо себе. Чтобы противостоять росту данной угрозы, необходимо повысить уровень кибербезопасности за счет развития технологий, направленных на своевременное обнаружение атак. Обычные пользователи часто являются жертвами недобросовестных пользователей, иногда по собственной невнимательности.

A.Логирование пользователя в сети

Анализ журналов веб-серверов и использование их данных для понимания поведения пользователей в сети открывают перед исследователями и разработчиками множество возможностей для улучшения веб-приложений и удовлетворения потребностей пользователей [1].

Файлы журнала могут находиться в трех разных местах [2]:

1) веб-серверы — в данных журналах, хранящихся на веб-серверах, регистрируются действия клиента, который обращается к веб-серверу через браузер;
2) веб-прокси-серверы — журналы логирования на веб-прокси-серверах обычно включают информацию о проксировании HTTP-запросов между клиентами и серверами. Они могут включать в себя записи о блокировках доступа к определенным сайтам, фильтрации контента, обнаружении вредоносных действий и другие события, связанные с работой прокси-сервера. Также в логах могут содержаться данные о пропускной способности, времени ответа сервера и другие метрики, которые помогают администраторам прокси-сервера следить за эффективностью его работы;
3) клиентские браузеры — журналы, хранящиеся в клиентских браузерах, могут храниться непосредственно в окне браузера клиента. Существуют специальные типы программного обеспечения, которые пользователь может загрузить в окно своего браузера. Несмотря на то, что файл журнала присутствует в окне браузера

Статья написана по результатам магистерской диссертации.
А. В. Зяблицева — МГУ имени М.В. Ломоносова, Москва, Россия (e-mail: azyablitseva666@gmail.com).

клиента, запись в файл журнала выполняется только веб-сервером.

В случае поиска аномалий в веб-сессиях наиболее интересным представляются логи с веб-серверов, поскольку они содержат хронологическую информацию о запросах, поступающих на сервер. Логи, в свою очередь, хранятся в 4 разных файлах [3]:

1) Access/Transfer log — журнал доступа/передачи данных — в нем хранится подробная информация о каждом запросе, отправленном из веб-браузеров пользователя на сервер. Например, с помощью хоста можно определить географическое местоположение, включая страну, штат и город, в котором находится пользователь, просматривающий данную страницу;
2) Error logs — журнал ошибок — содержит информацию об ошибках и неудачных запросах. Если страница, содержащая ссылку на файл, не существует или если пользователь не имеет права доступа к определенной странице или файлу, запрос может завершиться ошибкой;
3) Agent (Browser) log — журнал браузера — содержит информацию о браузерах и операционной системе, используемых разными пользователями для подключения к серверу;
4) Referrer log — журнал ссылок — содержит URL-адреса страниц на других сайтах, которые ссылаются на данную страницу.

B.Структура запроса

Несмотря на разные типы журналов логирования, все они обладают общей структурой[4]:

1) *IP-address/hostname* — IP-адрес клиента, отправляющего запрос, или, если работает обратный DNS и включен поиск по DNS, вводится имя хоста клиента;
2) *rfcname* — имя пользователя с удаленного сервера, при использовании IdentD [5] - протокола идентификации, описанном в RFC 1413, обеспечивающего способ идентификации пользователя для конкретного соединения TCP. Если значение отсутствует, вместо него ставится знак «-»;
3) *logname* — при использовании локальной аутентификации и регистрации, будет использоваться логин пользователя, аналогично, «-» если значение отсутствует;
4) *timestamp* — дата и время, когда был сделан запрос;
5) *page access method* — метод, используемый в HTTP-запросе, такой как GET, POST, PUT, DELETE и т.д.;
6) *path of requested file* — адрес, к которому клиент получил доступ;
7) *http version* — протокол, используемый в клиентском запросе, например, HTTP/1.1 или HTTP/2;
8) *server response code* — код состояния, возвращаемый сервером в ответ на запрос клиента;
9) *bytes received* — количество байт, отправляемых сервером в ответ на запрос клиента;
10) *refferrer url* — URL страницы, которая ведет на текущую страницу, помогает отследить источник трафика;
11) *user agent* — строка пользовательского агента предоставляет информацию о браузере, операционной системе и устройстве клиента;
12) *идентификатор сессии* — идентификаторы сессии для отслеживания сессий пользователя.

C.Обзор известных кибератак

Как было сказано ранее, кибератаки растут как по своему масштабу, так и по сложности. Чтобы понять, какие следы остаются после осуществленных атак, нужно рассмотреть известные на текущий день с целью выявления, как вредоносные сессии отличаются от сессий обычных пользователей:

1) DDoS атака (Distributed Denial of Service)[6] - может использоваться для загрузки веб-сайта или онлайн-сервиса большим объемом запросов, перегружая сервер и заставляя его работать медленно или не отвечать на запросы. В случае DDoS-атак несколько ботов кооперируются для одновременной атаки на одну систему, что усиливает воздействие атаки. Эти атаки направлены на перегрузку ресурсов веб-приложения, чтобы оно перестало отвечать на запросы пользователей. Злоумышленники могут использовать различные методы, такие как флуд запросы, переполнение буфера, атаки на протоколы (например, SYN Flood), чтобы создать проблемы доступности для законных пользователей;

2) Атаки перебора паролей (Brute Force) [7] - это метод, при котором злоумышленники пытаются автоматически или вручную перебирать различные комбинации паролей, пока не угадают правильный пароль, чтобы получить доступ к защищенной информации, учетным записям или системе. Это один из самых распространенных методов атаки на системы информационной безопасности. В логах доступа (*access.log*) при атаке перебора паролей можно обнаружить следующие признаки: увеличенное количество неудачных попыток входа — если злоумышленники осуществляют атаку перебора паролей, в логах будет фиксироваться большое количество неудачных попыток входа для одной или нескольких учетных записей; повторяющиеся запросы с различными учетными записями — атаки перебора паролей могут включать последовательный перебор различных учетных записей с разными паролями, поэтому в access.log можно заметить повторяющиеся запросы на аутентификацию с разными учетными записями; необычно высокая активность на страницах входа в систему: злоумышленники, осуществляющие атаку перебора паролей, могут активно обращаться к страницам входа в систему для отправки большого количества запросов на проверку пароля. Эта повышенная активность может так же отразиться в *access.log*;
3) атаки с использованием учетных данных[8] — могут использоваться для автоматизации процесса тестирования украденных комбинаций имени

пользователя и пароля на различных веб-сайтах и онлайн-сервисах с целью получения несанкционированного доступа к учетным записям пользователей. При таких атаках в *access.log* можно заметить подозрительные успешные входы: если в *access.log* обнаруживаются успешные аутентификации с необычных IP-адресов или устройств, это может свидетельствовать о том, что учетные данные были скомпрометированы и используются злоумышленником; аномальная активность: если в логах регистрируется большое количество успешных входов в систему с одного аккаунта или активность в необычное время суток, это также может быть признаком несанкционированного доступа к учетным данным;

4) атаки по извлечению данных — могут использоваться для извлечения больших объемов данных с веб-сайтов без разрешения. Такое извлечение может использоваться для различных целей, таких как сбор адресов электронной почты для спам-кампаний или извлечение конфиденциальной информации для кражи личных данных. Чрезмерное количество запросов к определенному ресурсу может свидетельствовать о данной атаке;
5) мошенничество с кликами — повышенное количество запросов веб-страниц за одну сессию, может использоваться для искусственного завышения количества кликов по онлайн-рекламе, что приносит доход злоумышленникам и приводит к финансовым потерям рекламодателям;
6) утечка информации о сессии (Session Hijacking)[9]: злоумышленники могут перехватить и использовать активные сессии пользователей для получения доступа к защищенным данным или выполнения нежелательных действий. В access.log можно обнаружить странные IP-адреса или необычные маркеры сессии, указывающие на несанкционированный доступ. Необычные маркеры сессии – это данные или идентификаторы, используемые для идентификации активных сессий пользователей на веб-сайте или веб-приложении. В обычном случае, каждая сессия пользователя имеет уникальный маркер, который позволяет серверу отслеживать активность конкретного пользователя. В случае таких атак злоумышленники могут пытаться изменить или подменить маркеры сессии пользователей, чтобы получить доступ к чужим учетным записям или конфиденциальным данным;
7) межсайтовый скриптинг (XSS)[10] — может использоваться злоумышленниками для внедрения вредоносных скриптов на веб-страницы, что позволяет им выполнять различные действия с браузером пользователя, такие как кража сессионных cookie или перенаправление на сайты с вредоносным содержимым. В access.log можно обнаружить неожиданные пользовательские агенты (user agents): злоумышленник может пытаться скрыть свою активность, притворившись обычным обычным агентом. Так же если в логах обнаруживаются подозрительные URL-адреса или параметры запросов, которые могут указывать на внедрение вредоносных скриптов, это может быть признаком атаки XSS;
8) межсайтовая подделка запроса (CSRF)[11] — предполагает выполнение несанкционированных действий от имени аутентифицированного пользователя через манипуляции с запросами к веб-приложению, которые отправляются без его ведома. Такая атака может характеризоваться множественными запросами с автоматическими действиями, в *access.log* можно обнаружить необычно большое количество однотипных запросов от разных пользователей или с одного источника. Это может быть вызвано автоматизацией атакующего сценария, при котором злоумышленник отправляет множество запросов для выполнения вредоносных действий на стороне пользователя. Так же CSRF-атаки могут быть направлены на выполнение вредоносных действий от имени авторизованных пользователей сразу по нескольким запросам от разных пользователей или устройств. К тому же в *access.log* можно заметить подозрительную синхронизированность таких действий, что также может быть признаком атаки.
9)

Это лишь некоторые из множества методов атак, нацеленных на веб-приложения. Важно обеспечить надлежащую защиту информационных систем путем регулярной проверки на уязвимости, обновления системы безопасности и обучения сотрудников, чтобы предупредить подобные инциденты.

D.Выбор значимых параметров сессии

На основе приведенных параметров запросов, которые логируются веб-сервером, а так же последствий известных кибератак, для построения новой более эффективной модели поиска аномалий в веб-сессиях были выбраны следующие параметры сессий:

1) количество кликов – числовой атрибут, рассчитываемый как количество запросов, отправленных пользователем за один сеанс. Показатель количества кликов полезен для обнаружения присутствия веб-сканеров, поскольку более высокий показатель количества кликов может быть достигнут только с помощью автоматизированного скрипта (например, веб-сканера), то это показатель будет выше при аномальной сессии. Обычно этот показатель мал для обычного посетителя;
2) количество байт, запрошенных у сервера – числовой показатель, который отражает объем данных в байтах, запрошенный у сервера за один сеанс. Обычно веб-сканеры, такие как поисковые роботы, запрашивают значительно больше данных за одну сессию, чем обычные посетители. Это связано с тем, что поисковые роботы сканируют большое количество страниц для индексации и анализа содержимого, в то время как обычный пользователь может выполнять ограниченное количество запросов;
3) процентное соотношение запросов изображения –

числовой показатель, рассчитываемый как отношение количества запросов на HTML-страницы к количеству запросов на файлы изображений за одну сессию. При сканировании и индексации веб-контента поисковые веб-сканеры имеют определенные приоритеты и задачи. Они чаще всего обращают внимание на HTML-страницы, поскольку именно в них содержится основная информация и текстовое содержимое, которое затем анализируется для формирования поисковых результатов. Изображения, хотя и важная часть веб-контента, часто имеют меньший приоритет для поисковых роботов;

4) процент запросов к файлам PDF/PS – это числовой показатель, который показывает процент запросов к файлам в форматах PDF или PS, отправленных за одну сессию. Поисковые веб-сканнеры часто имеют более высокий процент запросов в формате PDF или PS, по сравнению с обычными пользователями, так как они стремятся проанализировать все доступные файлы для индексации. В то же время обычные пользователи чаще выбирают просмотр других типов контента;
5) процент ответов с ошибками – числовой показатель, рассчитываемый как процент ошибочных HTTP-запросов, отправленных за один сеанс. Это важный параметр, который позволяет оценить эффективность работы сервера и качество работы веб-приложения. Ошибки 4xx обозначают клиентские ошибки, которые возникают, когда сервер не может выполнить запрос из-за некорректных или недействительных данных, например, когда запрашиваемая страница не найдена или доступ к ней запрещен. Поисковые роботы, такие как поисковые веб-сканеры, склонны иметь более высокий процент ошибочных запросов 4xx по сравнению с обычными пользователями. Это обусловлено несколькими факторами. Во-первых, поисковые роботы сканируют большое количество веб-страниц за короткий промежуток времени, пытаясь обновить свой поисковый индекс. Из-за этого они могут случайно попадать на устаревшие или удаленные страницы, что ведет к ошибкам 4xx. Во-вторых, поисковые роботы могут переходить по ссылкам на страницы, которые были изменены или перемещены, но эти изменения еще не отразились в индексе;
6) процент специальных запросов - числовой атрибут, рассчитываемый как процент запросов типа не GET или POST, отправленных за одну сессию. Тип запросов HEAD, например, позволяет получить только заголовок ответа без передачи фактического содержимого страницы. Веб-сканеры и другие инструменты, выполняющие сканирование или анализ веб-ресурсов, часто используют метод HEAD для эффективного сбора информации о сервере и страницах, так как это оптимизирует процесс сканирования, экономит ресурсы, а так же обеспечивают безопасность и конфиденциальность. В отличие от веб-сканеров, обычные пользователи, использующие браузеры, чаще всего отправляют запросы типа GET, чтобы получить полное содержимое страницы и взаимодействовать с ней. Это позволяет просматривать изображения, видео, текстовые данные и другой контент в полном объеме, обеспечивая более полноценный опыт использования веб-ресурса;
7) процент запросов с неназначенными ссылками – числовой показатель, рассчитываемый как процент пустых или неназначенных полей для ссылок, заданных пользователем за одну сессию. Веб-сканеры могут использовать неназначенные ссылки для изучения структуры сайта и поиска новых страниц или участков, которые могут быть скрыты или недоступны обычным пользователям;
8) стандартное отклонение глубины запрашиваемой страницы – числовой атрибут, рассчитываемый как стандартное отклонение глубины страницы для всех запросов, отправленных за одну сессию. Во время обычного просмотра веб-страниц люди исследуют информацию, переходя по различным тематически связанным ссылкам, которые постепенно становятся более конкретными. Возникают трудности, если пользователь теряет ориентацию. В отличие от людей, сканеры не испытывают сложностей с навигацией на сайте и не ограничены структурой ссылок. После первоначального сканирования сканеры могут точно определить местоположение искомой информации, что позволяет им в последующие посещения отправлять запросы только к определенным файлам или ограничивать свой обход конкретными областями сайта, поэтому процент среднего отклонения глубины запроса у злоумышленника будет меньше чем у обычного пользователя;
9) процент последовательных HTTP–запросов - числовой атрибут, рассчитываемый как процент последовательных запросов к страницам, принадлежащим к одному и тому же веб-каталогу, и генерируемый в течение одного пользовательского сеанса. По тем же причинам, указанным в пункте 8, процент последовательных запросов у обычных пользователей будет выше.
10) процент запрос файла «Robots.txt»[12] – числовое соотношение, которое отражает процент запросов файлов robot.txt к общему числу запросов. Веб-администраторы, используя протокол исключения роботов, используют файл специального формата под названием robots.txt, чтобы указать роботам-посетителям, какие части их сайтов роботу посещать не следует. Большинство пользователей не знают о его существовании. Файл robots.txt позволяет управлять сканированием. С его помощью можно уменьшить количество запросов, которые Google отправляет серверу, или запретить сканировать разделы сайта, в которых содержится неважная или повторяющаяся информация;
11) доля посещений одной и той же страницы — числовое соотношение предварительно посчитанной наиболее популярной страницы в течение сессии к общему количеству запросов зла сессию. Однако такая ситуация будет оказывать

влияние только в случае большого количества посещений, если посещений невелико, то вероятно, что пользователь посетит одну и ту же страницу. Потому что некоторые атаки всегда посещают одну и ту же страницу, например атака с использованием грубой силы для взлома пароля.

12) частота доступа — представляет собой количество посещений в минуту для каждого пользователя. Скорость доступа обычного пользователя ниже, но скорость доступа к инструменту выше.
13) среднее отклонение хэша используемых IP-адресов. У обычного пользователя оно будет оставаться равным нулю, в то время как при совершении атак, которые используют чужие сессии, оно будет отлично от нуля.

III. Методы машинного обучения для поиска аномалий в режиме реального времени

Система обнаружения вторжений (IDS) - это средство безопасности, которое занимается мониторингом сетевого трафика и применяет другие инструменты, такие как антивирусное ПО, брандмауэры и средства контроля доступа, для выявления подозрительной активности и отправки сигналов об обнаруженных нарушениях. Основная цель IDS - обнаружить аномалии и сформировать отчеты. Некоторые системы могут принимать действия при обнаружении любой аномальной активности, например, блокировать трафик с подозрительных IP-адресов.

В зависимости от типа анализа IDS можно классифицировать как сигнатурные или аномальные [13, 14, 15]. Системы на основе сигнатур и аномалий работают похожим образом, но различаются в понимании определений «атака» и «аномалия». Атака определяется как последовательность действий, угрожающих безопасности системы, в то время как аномалия - это просто событие, вызывающее подозрения относительно безопасности. Каждый тип IDS имеет свои преимущества и недостатки.

Сигнатурные IDS хорошо обнаруживают известные атаки, но неспособны выявлять новые или минимально измененные варианты известных атак. В то время как аномальные методы обнаружения способны обнаруживать ранее неизвестные виды вторжений. Однако частота ложных срабатываний у аномальных методов чаще, чем у сигнатурных систем, из-за вероятных неточностей в аномалиях.

Цель работы заключается в определении аномальных веб-сессий в режиме реального времени, что делает применение сигнатур невозможным. Инновация — возможность определять неизвестные ранее атаки на основе только что полученных данных и их сравнении с уже обработанными данными.

Выделим 4 типа моделей машинного обучения [16]:

1) алгоритмы обучения с учителем могут использовать помеченные данные, где явно идентифицируются аномалии. Эти алгоритмы изучают закономерности и взаимосвязи на основе помеченных данных и впоследствии могут обнаруживать аномалии в новых сессиях. Таким образом, они подходят в том случае, когда доступны размеченные данные, как о нормальных, так и об аномальных сессиях. Это полезно для обнаружения известных типов аномалий, но может оказаться неэффективным при выявлении новых или развивающихся аномалий;
2) обучение без учителя — этот метод позволяет выявлять аномалии на основе базового списка аномалий или шаблонов в данных без заранее заданных меток. Алгоритмы кластеризации, например, k-средние, DBSCAN или модели гауссовой смеси, могут группировать похожие элементы и идентифицировать отклонения от нормальных кластеров как аномалии. Данный тип подходит, когда помеченных данных мало или они недоступны. Это полезно для выявления аномалий, которые отличаются от большинства сеансов;
3) частичное обучение - это метод, который объединяет помеченные и немаркированные данные для обучения модели. В контексте обнаружения аномалий в сессиях, он предполагает использование небольшого объема помеченных данных, представляющих обычные сессии, и большего объема немаркированных данных. Подходит, когда маркированных данных мало, но доступны некоторые примеры аномальных сессия. Это обеспечивает баланс между преимуществами контролируемого и неконтролируемого обучения;
4) глубокое обучение фокусируется на использовании искусственных нейронных сетей с несколькими слоями для извлечения высокоуровневых представлений из данных. Из основных недостатков можно выделить, что оно требует большого объема маркированных данных и значительных вычислительных ресурсов для обучения. Они могут быть подвержены переобучению, если не будут должным образом регламентированы или валидированы. Подходит, когда доступны большие объемы помеченных данных и существует необходимость фиксировать сложные закономерности и зависимости в данных сессии.

На основании изложенного выше, было решено использовать методы машинного обучения без учителя, так как в разрабатываемом программном продукте не предусмотрено наличие обучающих промаркированных тестовых данных.

Методы обучения без учителя направлены на выявление закономерностей в данных серии, не полагаясь на заранее помеченные аномалии. Эти методы могут выявлять аномалии на основе отклонений от присущей данным структуры. При рассмотрении техник МО без учителя, было решено использовать алгоритмы кластеризации.

Алгоритмы кластеризации группируют похожие сессии вместе на основе их характеристик и идентифицируют аномалии как сессии, которые не принадлежат ни к одному кластеру или являются выбросами [17].

Преимущества: гибкость в обнаружении различных типов аномалий, независимость от помеченных данных

для обучения.

Недостатки: трудность в определении подходящего порога для обнаружения аномалии, чувствительность к выбору метрики расстояния или алгоритма кластеризации.

Различные работы, существующие в данной области, предлагают совершенно различные алгоритмы реализации, в том числе [18, 19, 20, 21]. В некоторых аналогичных работах используется алгоритм кластеризации DBSCAN[22].

DBSCAN (пространственная кластеризация приложений с шумом на основе плотности) - это алгоритм кластеризации на основе плотности, который группирует объекты на основе их плотности и связности. Аномалии идентифицируются как экземпляры, которые не принадлежат ни к одному плотному кластеру или считаются точками шума.

Преимущества: может обнаруживать аномалии произвольных форм и размеров. Для этого не требуется заранее указывать количество кластеров. Он устойчив к выбросам и может обрабатывать наборы данных с различной плотностью.

Недостатки: производительность может зависеть от выбора метрики расстояния и параметров плотности, испытывает трудности с данными большой размерности из-за проклятия размерности.

DBSCAN применяется к статической базе данных. Несмотря на то, что он хорошо справляется со своими задачами, для разрабатываемого программного продукта крайне важно иметь возможность поэтапного обновления кластеризации [23]. При использовании оригинального алгоритма при добавлении нового объекта (сессии) или модификации текущего объекта пришлось бы заново запускать алгоритм для всей имеющейся базы данных. Поскольку DBSCAN основан на анализе плотности, добавление нового объекта влияет на текущую кластеризацию только в окрестности этого объекта. Поэтому для более эффективной работы продукты был выбран инкрементная версия алгоритма DBSCAN для добавления новых элементов в уже существующую базу данных кластеризации.

Преимущества iDBSCAN [24]:

1) адаптивность — iDBSCAN может адаптироваться к изменяющемуся распределению данных и обеспечивать гибкость, приспосабливаясь к новым точкам или изменениям в кластере на лету;
2) масштабируемость — благодаря постепенному обновлению модели кластеризации iDBSCAN может эффективно обрабатывать большие динамические наборы данных, обеспечивая масштабируемость;
3) снижение вычислительных затрат — iDBSCAN снижает вычислительную сложность, связанную с повторным запуском всего процесса кластеризации при минимальных изменениях набора данных.
4)

Изучив работы [25, 26] было принято решение использовать именно данную версию iDBSCAN с добавочным разбиением данных на части. Данная работа подтверждает, что добавочное разделение объектов на части помогает уменьшить пространство поиска окрестности объекта. Когда появляется новый объект, окрестность определяется только через сканирование объектов ближайшего раздела, а не всего набора данных. Алгоритм секционирования предлагает деление данных на заранее определенное количество секций и назначение новому объекту части с ближайшим центром тяжести. Этот подход выбран из-за его способности учитывать динамический характер набора данных, позволяя старым объектам изменять положение после добавления новых. Также алгоритм гарантирует стабильность благодаря механизму скорости обучения, который позволяет объектам последовательно перемещаться к центрам тяжести с минимальными изменениями.

IV. Архитектура

Разрабатываемый программный продукт изначально предполагает две части: основную, где будет происходить обработка логов и их кластеризация, и часть тестирования, которая будет имитировать отправку запросов режиме реального времени.

Так как изначальной задачей является поиск аномалий в режиме реального времени, крайне важно на этапе тестирования обеспечить возможность имитации real-time отправки. Модуль тестирования построчно считывает новые события из *access.log* и отправляет сообщения в Kafka. Для универсальности сообщения он передает просто в виде строки. Модуль может быть адаптирован для автоматического считывания новых строк из логов, если нет возможности настроить запить запросов сразу в Kafka.

Основной сервис заключает в себе всю основную логику. Сервис состоит из трех микросервисов, каждый из которых которые отвечает за определенные события. Наиболее простой модуль data-model. Как видно из названия, это служебный модуль, хранящий информацию обо всех используемых и хранимых сущностях.

Модуль targeting-handler настроен на считывание событий из Kafka. При получении нового сообщения он обрабатывает его и преобразует к заранее определенному виду (apache). Журналы доступа к веб-серверу Apache обычно имеют стандартный формат, который включает различную информацию об HTTP-запросах к серверу. Ниже приведен общий пример формата журналов доступа Apache, но формат логов может быть настроен и расширен в соответствии с требованиями безопасности, требованиями отчетности и другими потребностями администраторов серверов:

SessionId IP-address rfcnmae logname [Datestamp] "METHOD URL protocol" status_code bytes_recieved "Referer" "User-Agent" .

После приведения сообщений к указанному формату по REST они отправляются в основной модуль wsad-api.

Задача данного микро-сервиса делится два части: сбор запросов в сессию и кластеризация.

Для того чтобы объединить запросы в сессию, они группируются по sessionId. В файлы конфигурации вынесен параметр тайм-аута — время, по истечении которого сессия помечается как завершенная. В то же время работает планировщик, который так же по заданному крону совершает обход добавленных сессий в поисках завершенных среди них. Найденные необработанные завершенные сессии он сразу же отправляет для кластеризации.

iDBSCAN работает без перерыва в течение работы всей программы. Благодаря выбранному алгоритму, при получении новых сессий не приходится пересчитывать все элементы заново. Изменяются только те, на которые влияет новый объект. При обнаружении сессии похожей на аномальную, он помечает ее как шум и удаляет из списка посещаемых сессий. Кроме того можно активировать автоматическое очищение таблицы по планировщику, чтобы не использовать устаревшие данные и не тратить большое количество ресурсов на их хранение.

V. Эксперименты

Для тестирования разработанной системы используются несколько наборов данных, полученных из логов определенной компании за разные периоды времени. Логи предварительно были размечены для анализов результатов тестирования. Файлы легирования хранят в себе такие аномалии, как запрос одной и той же страницы, попытка аутентификации с неправильным паролем и/или сертификатом, запрос по несуществующим хостам, сессии со слишком большой частотой запросов и еще некоторыми видами атак. Поиск открытых наборов данных оказался неэффективным способом, так как наборы *access.log* представлены отдельно — без оценки экспертов с объединенными запросами в сессию и признаком аномальности.

Тестирование произведено в несколько этапов, чтобы оценить, как изменяется эффективность работы программы при увеличении количества входных запросов. Набор данных *access.log* был разбит на несколько разных наборов разной длины.

Таблица 1 - наборы данных для тестирования

datasets	**Number of Requests**
Datasets1	1450
Datasets2	3424
Datasets3	6542
Datasets4	19430

При использовании алгоритма iDBSCAN на вход необходимо подать два параметра на вход - радиус окрестности *Eps* и пороговое значение плотности окрестностей *Minpts*. Руководствуясь существующими указаниями, пороговое значение плотности установлено равным количество параметров в кластеризируемых сессиях - 12. Радиус окружения выбрался в соответствии с результатами эксперимента.

Результаты анализа приведены в таблице 2. WSAD без ошибок отрабатывает на не очень больших данных, при увеличении подаваемых запросов появляются ложноположительные срабатывания, хотя они и не критичны.

Таблица 2 - результаты тестирования

datasets	**The number of session is actually abnormal**	**The number of true positives**	**The number of false positives**	**The number of false negative**
Datasets1	14	14	0	0
Datasets2	29	29	0	0
Datasets3	47	49	2	0
Datasets4	78	83	7	1

VI. Развитие и предстоящие работы

На данный момент реализована система, которая в режиме реального времени позволяет обнаруживать различные атаки на веб-сессии. На данный момент анализ производится на основе статических параметров сессии, которые высчитываются по ее окончании. Однако, кажется целесообразным добавить еще и возможность сравнивать сессии, принадлежащие одному пользователю, чтобы помимо обнаружения просто аномалий среди всех прочих, можно было бы учитывать особенности конкретного пользователя и его паттерн поведения. К тому же в основу статических методов можно также добавить анализ строки запроса, чтобы можно было на основе подозрительных строк запроса повышать коэффициент аномальности у конкретных сессий.

VII. Заключение

В результате проведенного исследования была предложена новая модель, которая позволяет более точно находить аномалии в веб-сессиях в режиме реального времени.

Модель получает данные о новых запросах из журнала логов, хранящегося на веб-серверах, так как данные журналы предоставляют всю информацию о свершенных пользователем действиях в последовательном формате. Кроме того, использование именно *access.log* объясняется предоставлением наиболее полной информации.

Проанализировав основные шаблоны поведения пользователей в сети и наиболее известные на данный момент кибератаки, было предложено 12 различных полезных параметров сессий, которые используются

при кластеризации. Полезность параметров объяснена в данной статье. Для реализации поиска аномалий в веб-сессиях был выбран подход машинного обучения без учителя, а именно, алгоритм кластеризации, так как он группирует похожие сессии на основе их характеристик и идентифицирует аномалии как сессии, которые не принадлежат ни к одному кластеру или являются выбросами. Выбор iDBSCAN можно обосновать его ключевыми свойствами: оперативные обновления кластеризации, частое изменение данных.

После разработки модели был реализован программный продукт, который позволяет в режиме реального времени находить аномалии в веб-сессиях [27]. В частности, было реализовано вспомогательное приложение, которое позволяет воспроизводить имитацию возникновения запросов в режиме реального времени [28]. Программный продукт требует наличия jdk-21 и gradle 7.2 для запуска, в остальном может быть запущен на любой платформе. Исходный код приложения выложен на GitHub под лицензией MIT.

После разработки продукта было произведено тестирование на реальных данных, которое помогло оценить эффективность разработанной модели.

Технологии не стоят на месте, кибер-атаки становятся все масштабнее и все изощреннее. Злоумышленники находят способы обходить защиту для получения личной выгоды и пользы, подвергая риску множество обычных пользователей. На данный момент существует не так много эффективных способов для обнаружения ранее неизвестных атак в режиме реального времени. Развитие в данном направлении может сохранить личную и конфиденциальную информацию, денежные активы и не только.

Эффективная система обнаружения аномалий в веб-сессиях представляет собой неотъемлемую часть комплексной стратегии кибербезопасности. Реализация механизмов мониторинга и анализа действий пользователей в режиме реального времени помогает предотвращать атаки, связанные с утечкой информации, фишингом, CSRF и XSS, обеспечивая высокий уровень защиты данных и конфиденциальности.

Благодарности

Автор благодарна сотрудникам ДКБ Сбербанк за полезные замечания при рецензировании магистерской диссертации.

Тема магистерской диссертации продолжает серию работ об использовании искусственного интеллекта в кибербезопасности [29].

Редакция журнала INJOIT традиционно отмечает, что все публикации в журнале, связанные с цифровой повесткой, начинались с работ В.П. Куприяновского и его соавторов [30-32]

Библиография

1. Grace L. K., Maheswari V., Nagamalai D. Analysis of web logs and web user in web mining //arXiv preprint arXiv:1101.5668. – 2011.
2. Yixin Wu, Yuqiang Sun, Cheng Huang, Peng Jia, Luping Liu, "Session-Based Webshell Detection Using Machine Learning in Web Logs", Security and Communication Networks, vol. 2019, Article ID 3093809, 11 pages, 2019. https://doi.org/10.1155/2019/3093809
3. Sisodia D. S., Verma S. Web usage pattern analysis through web logs: A review //2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE). – IEEE, 2012. – С. 49-53.
4. Cooley, R., Mobasher, B., and Srivastava, J, “Web mining: information and pattern discovery on the World Wide Web”, International Conference on Tools with Artificial Intelligence, Newport Beach, IEEE, 1997, pp. 558-567
5. Johns M. S. Identification protocol. – 1993. – №. rfc1413.
6. Bawany N. Z., Shamsi J. A., Salah K. DDoS attack detection and mitigation using SDN: methods, practices, and solutions //Arabian Journal for Science and Engineering. – 2017. – Т. 42. – С. 425-441.
7. Park J. et al. Network log-based SSH brute-force attack detection model //Computers, Materials & Continua. – 2021. – Т. 68. – №. 1.
8. Chen Q. et al. Privwatcher: Non-bypassable monitoring and protection of process credentials from memory corruption attacks //Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. – 2017. – С. 167-178.
9. Gill R., Smith J., Clark A. Experiences in passively detecting session hijacking attacks in IEEE 802.11 networks //ACSW Frontiers 2006: Proceedings of the 4th Australasian Symposium on Grid Computing and e-Research (AusGrid 2006) and the 4th Australasian Information Security Workshop (Network Security)(AISW-NetSec 2006)[CRPIT, Volume 54]. – Australian Computer Society, 2006. – С. 221-230.
10. Sarmah U., Bhattacharyya D. K., Kalita J. K. A survey of detection methods for XSS attacks //Journal of Network and Computer Applications. – 2018. – Т. 118. – С. 113-143.
11. Calzavara S. et al. Mitch: A machine learning approach to the black-box detection of CSRF vulnerabilities //2019 IEEE European Symposium on Security and Privacy (EuroS&P). – IEEE, 2019. – С. 528-543.
12. Geiger I. I., Stuart R. Robots. txt: An Ethnographic Investigation of Automated Software Agents in User-Generated Content Platforms : дис. – UC Berkeley, 2015.
13. Martin Roesch et al. Snort: Lightweight intrusion detection for networks. In Lisa, volume 99, pages 229-238, 1999.
14. Magnus Almgren, Herve Debar, and Marc Dacier. A lightweight tool for detecting web server attacks. In NDSS, 2000.
15. Christopher Kruegel and Giovanni Vigna. Anomaly detection of web- based attacks. In Proceedings of the 10th ACM conference on Computer and communications security, pages 251-261, 2003.
16. Sarker I. H. Machine learning: Algorithms, real-world applications and research directions //SN computer science. – 2021. – Т. 2. – №. 3. – С. 160.
17. Swarndeep Saket J., Pandya S. An overview of partitioning algorithms in clustering techniques //International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). – 2016. – Т. 5. – №. 6. – С. 1943-1946.
18. Truong Son Pham, Tuan Hao Hoang, and Vu Van Canh. Machine learning techniques for web intrusion detection— a comparison. In 2016 Eighth International Conference on Knowledge and Systems Engineer- ing (KSE), pages 291-297. IEEE, 2016.
19. Pengfei Liu, Weiping Wang, Xi Luo, Haodong Wang, and Chushu Liu. Nsdroid: efficient multi-classification of android malware using neighborhood signature in local function call graphs.International Journal of Information Security, pages 1-13, 2020.
20. Weiping Wang, Jianjian Wei, Shigeng Zhang, and Xi Luo. Lscdroid: Malware detection based on local sensitive api invocation sequences. IEEE Transactions on Reliability, 69(1):174-187, 2019.
21. Jingxi Liang, Wen Zhao, and Wei Ye. Anomaly-based web attack detection: a deep learning approach. In Proceedings of the 2017 VI International Conferen
22. Sun Y. et al. Wsad: An unsupervised web session anomaly detection method //2020 16th International Conference on Mobility, Sensing and Networking (MSN). – IEEE, 2020. – С. 735-739.
23. Saha J., Mukherjee J. IPD: An Incremental Prototype based DBSCAN for large-scale data with cluster representatives //arXiv preprint arXiv:2202.07870. – 2022. —DBSCAN
24. Azhir E. et al. An efficient automated incremental density-based algorithm for clustering and classification //Future Generation Computer Systems. – 2021. – Т. 114. – С. 665-678.—DBSCAN
25. Bakr A. M., Ghanem N. M., Ismail M. A. Efficient incremental density-based algorithm for clustering large datasets //Alexandria engineering journal. – 2015. – Т. 54. – №. 4. – С. 1147-1154
26. S. Young, I. Arel, A fast and stable incremental clustering algorithm, in: Seventh International Conference on Information Technology: New Generations (ITNG), April 2010, pp. 204–209.

27. Репозиторий с исходным кодом программного продукта «J-wsad». [Электронный ресурс]. URL: https://github.com/jbrigett/j-wsad.git (дата обращения 10.05.2024).
28. Репозиторий с исходным кодом программного продукта «J-wsad». [Электронный ресурс]. URL: https://github.com/jbrigett/j-wsad-utilities.git (дата обращения 10.05.2024).
29. Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "Artificial intelligence and cybersecurity." International Journal of Open Information Technologies 10.9 (2022): 135-147. (in Russian)
30. Искусственный интеллект как стратегический инструмент экономического развития страны и совершенствования ее государственного управления. Часть 2. Перспективы применения искусственного интеллекта в России для государственного управления / И. А. Соколов, В. И. Дрожжинов, А. Н. Райков [и др.] // International Journal of Open Information Technologies. – 2017. – Т. 5, № 9. – С. 76-101. – EDN ZEQDMT.
31. Куприяновский, В. П. Демистификация цифровой экономики / В. П. Куприяновский, Д. Е. Намиот, С. А. Синягов // International Journal of Open Information Technologies. – 2016. – Т. 4, № 11. – С. 59-63. – EDN WXQLIJ.
32. Розничная торговля в цифровой экономике / В. П. Куприяновский, С. А. Синягов, Д. Е. Намиот [и др.] // International Journal of Open Information Technologies. – 2016. – Т. 4, № 7. – С. 1-12. – EDN WCMIWN.

Unsupervised machine learning methods for finding anomalies in real-time web sessions

Alexandra Zyablitseva

***Abstract*—Detecting anomalies in web sessions is an important issue that is currently actively developing. To detect cyber threats, the signature-based anomaly detection method is generally used. Such methods effectively detect known attacks, but cannot detect unknown anomalies. In addition, existing anomaly detection methods are usually based on identifying abnormal sessions based on static data, having training data with tags, and also determine the session based on the IP address. In this paper, the main methods of collecting information about user actions on the network, forms of logging and collecting information about web sessions are considered, an approach to finding anomalies in web sessions is presented and justified. The anomaly detection model in web sessions uses an unsupervised clustering algorithm that does not require pre-marinated data, and the most effective clustering uses 13 of the most useful session parameters. The architecture of the developed approach is described for the presented model. At the end of the work, the result of the tethering is presented and further directions for the development of the project are described.**

***Keywords*—anomalies, clustering, web session, query, iDBSCAN.**

REFERENCES

1. Grace L. K., Maheswari V., Nagamalai D. Analysis of web logs and web user in web mining //arXiv preprint arXiv:1101.5668. – 2011.
2. Yixin Wu, Yuqiang Sun, Cheng Huang, Peng Jia, Luping Liu, "Session-Based Webshell Detection Using Machine Learning in Web Logs", Security and Communication Networks, vol. 2019, Article ID 3093809, 11 pages, 2019. https://doi.org/10.1155/2019/3093809
3. Sisodia D. S., Verma S. Web usage pattern analysis through web logs: A review //2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE). – IEEE, 2012. – C. 49-53.
4. Cooley, R., Mobasher, B., and Srivastava, J, "Web mining: information and pattern discovery on the World Wide Web", International Conference on Tools with Artificial Intelligence, Newport Beach, IEEE, 1997, pp. 558-567
5. Johns M. S. Identification protocol. – 1993. – №. rfc1413.
6. Bawany N. Z., Shamsi J. A., Salah K. DDoS attack detection and mitigation using SDN: methods, practices, and solutions //Arabian Journal for Science and Engineering. – 2017. – T. 42. – C. 425-441.
7. Park J. et al. Network log-based SSH brute-force attack detection model //Computers, Materials & Continua. – 2021. – T. 68. – №. 1.
8. Chen Q. et al. Privwatcher: Non-bypassable monitoring and protection of process credentials from memory corruption attacks //Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. – 2017. – C. 167-178.
9. Gill R., Smith J., Clark A. Experiences in passively detecting session hijacking attacks in IEEE 802.11 networks //ACSW Frontiers 2006: Proceedings of the 4th Australasian Symposium on Grid Computing and e-Research (AusGrid 2006) and the 4th Australasian Information Security Workshop (Network Security)(AISW-NetSec 2006)[CRPIT, Volume 54]. – Australian Computer Society, 2006. – C. 221-230.
10. Sarmah U., Bhattacharyya D. K., Kalita J. K. A survey of detection methods for XSS attacks //Journal of Network and Computer Applications. – 2018. – T. 118. – C. 113-143.
11. Calzavara S. et al. Mitch: A machine learning approach to the black-box detection of CSRF vulnerabilities //2019 IEEE European Symposium on Security and Privacy (EuroS&P). – IEEE, 2019. – C. 528-543.
12. Geiger I. I., Stuart R. Robots. txt: An Ethnographic Investigation of Automated Software Agents in User-Generated Content Platforms : дис. – UC Berkeley, 2015.
13. Martin Roesch et al. Snort: Lightweight intrusion detection for networks. In Lisa, volume 99, pages 229-238, 1999.
14. Magnus Almgren, Herve Debar, and Marc Dacier. A lightweight tool for detecting web server attacks. In NDSS, 2000.
15. Christopher Kruegel and Giovanni Vigna. Anomaly detection of web- based attacks. In Proceedings of the 10th ACM conference on Computer and communications security, pages 251-261, 2003.
16. Sarker I. H. Machine learning: Algorithms, real-world applications and research directions //SN computer science. – 2021. – T. 2. – №. 3. – C. 160.
17. Swarndeep Saket J., Pandya S. An overview of partitioning algorithms in clustering techniques //International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). – 2016. – T. 5. – №. 6. – C. 1943-1946.
18. Truong Son Pham, Tuan Hao Hoang, and Vu Van Canh. Machine learning techniques for web intrusion detection— a comparison. In 2016 Eighth International Conference on Knowledge and Systems Engineer- ing (KSE), pages 291-297. IEEE, 2016.
19. Pengfei Liu, Weiping Wang, Xi Luo, Haodong Wang, and Chushu Liu. Nsdroid: efficient multi-classification of android malware using neighborhood signature in local function call graphs.International Journal of Information Security, pages 1-13, 2020.
20. Weiping Wang, Jianjian Wei, Shigeng Zhang, and Xi Luo. Lscdroid: Malware detection based on local sensitive api invocation sequences. IEEE Transactions on Reliability, 69(1):174-187, 2019.
21. Jingxi Liang, Wen Zhao, and Wei Ye. Anomaly-based web attack detection: a deep learning approach. In Proceedings of the 2017 VI International Conferen
22. Sun Y. et al. Wsad: An unsupervised web session anomaly detection method //2020 16th International Conference on Mobility, Sensing and Networking (MSN). – IEEE, 2020. – C. 735-739.
23. Saha J., Mukherjee J. IPD: An Incremental Prototype based DBSCAN for large-scale data with cluster representatives //arXiv preprint arXiv:2202.07870. – 2022. —DBSCAN
24. Azhir E. et al. An efficient automated incremental density-based algorithm for clustering and classification //Future Generation Computer Systems. – 2021. – T. 114. – C. 665-678.—DBSCAN
25. Bakr A. M., Ghanem N. M., Ismail M. A. Efficient incremental density-based algorithm for clustering large datasets //Alexandria engineering journal. – 2015. – T. 54. – №. 4. – C. 1147-1154
26. S. Young, I. Arel, A fast and stable incremental clustering algorithm, in: Seventh International Conference on Information Technology: New Generations (ITNG), April 2010, pp. 204–209.
27. A repository with the only code of the "J-wsad" software product. [electronic resource]. URL: https://github.com/jbrigett/j-wsad.git (accessed 05/10/2024).
28. A repository with the only code of the "J-wsad" software product. [electronic resource]. URL: https://github.com/jbrigett/j-wsad-utilities.git (accessed 05/10/2024).
29. Namiot, Dmitry, Eugene Ilyushin, and Ivan Chizhov. "Artificial intelligence and cybersecurity." International Journal of Open Information Technologies 10.9 (2022): 135-147. (in Russian)
30. Iskusstvennyj intellekt kak strategicheskij instrument jekonomicheskogo razvitija strany i sovershenstvovanija ee gosudarstvennogo upravlenija. Chast' 2. Perspektivy primenenija iskusstvennogo intellekta v Rossii dlja gosudarstvennogo upravlenija / I. A. Sokolov, V. I. Drozhzhinov, A. N. Rajkov [i dr.] // International Journal of Open Information Technologies. – 2017. – T. 5, # 9. – S. 76-101. – EDN ZEQDMT.
31. Kuprijanovskij, V. P. Demistifikacija cifrovoj jekonomiki / V. P. Kuprijanovskij, D. E. Namiot, S. A. Sinjagov // International Journal of Open Information Technologies. – 2016. – T. 4, # 11. – S. 59-63. – EDN WXQLIJ.
32. Roznichnaja torgovlja v cifrovoj jekonomike / V. P. Kuprijanovskij, S. A. Sinjagov, D. E. Namiot [i dr.] // International Journal of Open Information Technologies. – 2016. – T. 4, # 7. – S. 1-12. – EDN WCMIWN.