**Диплом**

Тема: Выявление аномальных клиентов web-сервера по последовательности их запросов

Задача решается в контексте выявления источников DDoS-атак на web-серверы.

По отдельным запросам затруднительно установить, сделан он легитимным

клиентом или атакующим. Однако для легитимных клиентов характерны

определенные последовательности запросов, по отклонению от которых можно

отличить аномальных клиентов. Для клиентов разных видов, которые можно выделить

по характеристикам запросом, могут быть типичны разные последовательности.

Исходные данные: выгрузка записей журнала web-сервера с описанием её состава.

Задание:

1. Изучить исходные данные и предметную область. Выбрать признаки запросов,

которые можно использовать при решении задачи и обосновать этот выбор.

1. Предложить математическую модель для описания последовательности запросов,

в которой запросы легитимных и аномальных клиентов будут отличаться.

1. Написать программу, которая реализует выбранную модель,

настраивает (обучает) модель по исходным данным

и может быть использована, чтобы выявлять аномальных клиентов

по потоку записей журнала web-сервера.

1. Оценить качество полученной модели. Предложить и обосновать метрики качества.

1. Написать отчет о проделанной работе в соответствии с требованиями НИУ «МЭИ».

**Часть 1**

**Что такое DDoS – атака**

DDoS-атака (Distributed Denial of Service «распределенный отказ от обслуживания») — это форма кибератаки на веб-системы с целью вывести их из строя или затруднить доступ к ним для обычных пользователей. Атакующие обычно используют распределенную сеть множества устройств, одновременно отправляющих запросы на сервер жертвы, пока он не будет перегружен. Такие кибератаки наиболее распространены, потому что могут довести до отказа любую систему без должной защиты, не оставляя юридически значимых улик.

Метод можно сравнить с ситуацией, когда множество людей толпится в дверях магазина и не дает никому пройти. В результате магазин теряет прибыль и вынужден предпринимать какие-то действия, чтобы их разогнать.

Существует также более простая версия подобных атак — DoS-атаки (Denial of Service). Они отличаются тем, что осуществляются из одной подсети. Современные инструменты защиты могут легко их распознать и заблокировать, поэтому таким атакам подвергаются в основном простые системы и личные компьютеры.

**Цели DDoS – атак**

Иногда DDoS-атаки ставят задачей захватить управление серверами, но современные архитектуры редко такое позволяют. Чаще всего целью является попытка помешать продукту работать по одной из следующих причин:

* Экономическая — хакеры выполняют заказ конкурента компании. Эта услуга пользуется большим спросом в даркнете. Атакам могут подвергаться интернет-магазины, банки и другие интересующие организации.
* Политическая — протест против политики какого-либо государства или информационная война. Целью могут быть правительственные организации, сайты министерств, СМИ.
* Вымогательство или шантаж — злоумышленник ставит владельцу сайта условия, после выполнения которых он остановит атаки.
* Личная неприязнь — киберпреступники пытаются создать проблемы сайту или организации, которая чем-то им не нравится. Например, атакам часто подвергаются игровые серверы различных киберспортивных команд.
* Развлечение — начинающие хакеры просто пробуют свои силы. Целью атаки может быть что угодно.

**Типы DDoS-атак**

Существует множество видов DDoS-атак, нацеленных на разные уязвимые места жертвы, но все они направлены на снижение пропускной способности сервера или исчерпание ресурсы системы. Среди них:

1. **Переполнение канала (flood)** — отправка множества запросов к компьютерной системе, чтобы переполнить ее каналы связи. Основные разновидности:
   * HTTP-флуд и ping-флуд — злоумышленник шлет запросы на пакеты больших размеров и подменяет свой IP-адрес, чтобы самому не стать жертвой своей же атаки.
   * Smurf-атака — атакующий рассылает ICMP-пакеты (протокол для передачи сообщений об ошибках) с помощью широковещательного IP-адреса, а затем меняет свой адрес на адрес жертвы. Компьютеры сети начинают отвечать на эти запросы, переполняя трафиком компьютер жертвы.
2. **Атака йо-йо** — особый вид атак на облачные приложения с автоматическим масштабированием. Злоумышленник атакует систему до того момента, когда она вынуждена будет расшириться. Затем трафик прекращается, оставляя службы перегруженными. Когда масштабирование пойдет в обратном направлении, трафик возобновится.
3. **Ошибки программирования** — выявление уязвимостей архитектуры системы жертвы и написание программ, которые смогут их эксплуатировать. Среди них:
   * Обработка исключений — атакующий ищет ошибки в коде системы, которые она не умеет обрабатывать, и провоцирует их. Это может вызвать критическую ошибку, которая приведет к отказу системы.
   * Переполнение буфера — злоумышленник эксплуатирует уязвимость системы, входные данные которой недостаточно проверяются. Он отправляет ей пакеты большего объема, чем она готова обрабатывать, в результате чего переполняется буфер и ядро дает сбой.
4. **Атака ботнета Mirai** — злоумышленник использует множество зараженных устройств интернета вещей, таких как часы с поддержкой Wi-Fi, стиральные машины, камеры слежения и прочих. Подобные устройства слабо защищены, поэтому их легко заразить и заставить отправлять запросы жертве. При достаточном количестве подконтрольных устройств атакующий может навредить даже самым мощным серверам.
5. **Маршрутизация и атаки DNS** — злоумышленник атакует [DNS-сервер](https://cloud.yandex.ru/docs/glossary/dns) жертвы. Бывают двух видов:
   * DoS-атака на уязвимости DNS-сервера — атакующий подменяет IP-адрес DNS-сервера атакуемого, после чего тот либо не может получить нужную HTML-страницу, либо попадает на узел к преступнику, где его данные могут захватить.
   * DDoS-атака на DNS-сервер — злоумышленник при помощи компьютеров-зомби (зараженных вредоносным кодом, позволяющим использовать их ресурсы) перенасыщает полосу пропускания DNS-сервера жертвы.

По уровню модели OSI DDoS-атаки делятся на три вида:

* Атака на прикладном уровне (Layer 7) — хакер отправляет HTTP-запросы на генерацию веб-страниц. Например, HTTP-флуд, Slowloris (попытка открыть множество подключений и не прекращать их).
* Атака на транспортном уровне (Layer 4) — злоумышленник пытается переполнить каналы доставки данных. Например, Smurf-атака, SYN-флуд (поток TCP или SYN-пакетов) и другие.
* Атака на сетевом уровне (Layer 3) — атакуются сетевые протоколы и оборудование. Например, IP-флуд и ICMP-флуд, ставящие целью перегрузить пропускную способность целевой сети.

**Методы защиты от DDoS**

Если атака уже происходит, а у сервера нет должной защиты, то быстро что-то предпринять, вероятно, не удастся. Нужно иметь запасной сервер, возможность перенастроить DNS, ограничить скорость или уметь выявлять аномальный трафик, чем и будем заниматься далее. Также злоумышленники зачастую заранее проверяют защиту жертвы непродолжительными атаками. Если вовремя это распознать, можно заранее предпринять какие-то действия.

Надежные методы защиты носят в основном проактивный характер:

* Разработайте план действий на случай атаки. Например, приобретите резервные серверы.
* Выявите уязвимости системы раньше, чем это сделает атакующий. Можно использовать программы, имитирующие DDoS-атаку, например LOIC, HULK, DDOSIM Layer 7.
* Соблюдайте интернет-гигиену, поддерживайте ПО в актуальном состоянии.
* Пользуйтесь средствами защиты. Прежде всего, это [SSL](https://cloud.yandex.ru/docs/glossary/ssl-certificate), но для продвинутой защиты также можно использовать файрвол веб-приложений и другие инструменты.
* Не держите в открытом доступе ваши реальные IP-адреса и периодически меняйте их.
* Попробуйте облачные технологии. Провайдеры имеют более высокие мощности и сами заботятся о безопасности клиентов.

DDoS-атаки — одна из главных угроз для интернет-бизнеса, а убытки от них на мировом рынке исчисляются миллионами долларов, поэтому лишней не будет любая защита.

## Тенденции и будущее DDoS-атак

В начале распространения интернета мощными DDoS-атаками считались нападения всего с нескольких десятков компьютеров и с объемом трафика, исчисляемым мегабитами. Сегодня сеть ботов может насчитывать сотни тысяч устройств, а атаки с трафиком до 400 Гбит/с можно заказать по цене смартфона среднего сегмента.

Можно с уверенностью сказать, что эти цифры будут только расти в будущем. Кроме того, развитие интернета вещей, повсеместное внедрение машинного обучения и появление 5G-сетей также означает, что атаки будут ужесточаться. Однако принципиально новых видов DDoS-атак ждать не стоит: они не появлялись уже несколько лет и пока нет причин полагать, что это изменится.

По словам экспертов по защите от кибератак, пока затраты на проведение DDoS-атак покрываются потенциальной выгодой, их количество и мощность будет расти. Однако развиваются и технологии защиты, поэтому, чтобы пополнить ряды успешных киберпреступников, придется значительно сильнее стараться.

1. Проверка того, скольким запросам соответствует определенный статус

Результат выполнения кода:

Статус "301" - 100994 (50.305%)

Статус "404" - 1265 (0.63%)

Статус "499" - 6223 (3.1%)

Статус "400" - 362 (0.18%)

Статус "200" - 51314 (25.559%)

Статус "304" - 592 (0.295%)

Статус "405" - 38995 (19.423%)

Статус "503" - 344 (0.171%)

Статус "504" - 522 (0.26%)

Статус "408" - 151 (0.075%)

Статус "206" - 2 (0.001%)

Вывод:

\*\*200 OK — успешный запрос. Если клиентом были запрошены какие-либо данные, то они находятся в заголовке и/или теле сообщения. Появился в HTTP/1.0.

\*\*206 Partial Content — сервер удачно выполнил частичный GET-запрос, возвратив только часть сообщения. В заголовке Content-Range сервер указывает байтовые диапазоны содержимого. Особое внимание при работе с подобными ответами следует уделить кэшированию. Появился в HTTP/1.1.

\*\*301 Moved Permanently — запрошенный документ был окончательно перенесен на новый URI, указанный в поле Location заголовка. Некоторые клиенты некорректно ведут себя при обработке данного кода. Появился в HTTP/1.0.

\*\*304 Not Modified — сервер возвращает такой код, если клиент запросил документ методом GET, использовал заголовок If-Modified-Since или If-None-Match и документ не изменился с указанного момента. При этом сообщение сервера не должно содержать тела. Появился в HTTP/1.0.

\*\*400 Bad Request — сервер обнаружил в запросе клиента синтаксическую ошибку. Появился в HTTP/1.0.

\*\*404 Not Found[20] — самая распространённая ошибка при пользовании Интернетом, основная причина — ошибка в написании адреса Web-страницы. Сервер понял запрос, но не нашёл соответствующего ресурса по указанному URL. Если серверу известно, что по этому адресу был документ, то ему желательно использовать код 410. Ответ 404 может использоваться вместо 403, если требуется тщательно скрыть от посторонних глаз определённые ресурсы. Появился в HTTP/1.0.

\*\*405 Method Not Allowed — указанный клиентом метод нельзя применить к текущему ресурсу. В ответе сервер должен указать доступные методы в заголовке Allow, разделив их запятой. Эту ошибку сервер должен возвращать, если метод ему известен, но он не применим именно к указанному в запросе ресурсу, если же указанный метод не применим на всём сервере, то клиенту нужно вернуть код 501 (Not Implemented). Появился в HTTP/1.1.

\*\*408 Request Timeout — время ожидания сервером передачи от клиента истекло. Клиент может повторить аналогичный предыдущему запрос в любое время. Например, такая ситуация может возникнуть при загрузке на сервер объёмного файла методом POST или PUT. В какой-то момент передачи источник данных перестал отвечать, например, из-за повреждения компакт-диска или потери связи с другим компьютером в локальной сети. Пока клиент ничего не передаёт, ожидая от него ответа, соединение с сервером держится. Через некоторое время сервер может закрыть соединение со своей стороны, чтобы дать возможность другим клиентам сделать запрос. Этот ответ не возвращается, когда клиент принудительно остановил передачу по команде пользователя или соединение прервалось по каким-то иным причинам, так как ответ уже послать невозможно. Появился в HTTP/1.1.

\*\*499 Client Closed Request — нестандартный код, предложенный и используемый nginx для случаев, когда клиент закрыл соединение, пока nginx обрабатывал запрос.

\*\*503 Service Unavailable — сервер временно не имеет возможности обрабатывать запросы по техническим причинам (обслуживание, перегрузка и прочее). В поле Retry-After заголовка сервер может указать время, через которое клиенту рекомендуется повторить запрос. Хотя во время перегрузки очевидным кажется сразу разрывать соединение, эффективней может оказаться установка большого значения поля Retry-After для уменьшения частоты избыточных запросов. Появился в HTTP/1.0.

\*\*504 Gateway Timeout — сервер в роли шлюза или прокси-сервера не дождался ответа от вышестоящего сервера для завершения текущего запроса. Появился в HTTP/1.1.

Вывод: 50% запросов была редиректами, а 25.5% успешными

1. Проверка того, какие запросы являются нагруженными и насколько нагруженными, а также выведем только основные случаи

162 байтов - в 100939 со статусом 301

117 байтов - в 459 со статусом 404

251 байтов - в 661 со статусом 404

0 байтов - в 6223 со статусом 499

650 байтов - в 154 со статусом 400

343011 байтов - в 432 со статусом 200

40551 байтов - в 644 со статусом 200

4514 байтов - в 105 со статусом 200

7315 байтов - в 105 со статусом 200

2723 байтов - в 101 со статусом 200

34621 байтов - в 128 со статусом 200

7985 байтов - в 165 со статусом 200

14150 байтов - в 131 со статусом 200

8721 байтов - в 141 со статусом 200

16014 байтов - в 898 со статусом 200

568005 байтов - в 112 со статусом 200

16006 байтов - в 913 со статусом 200

339523 байтов - в 3773 со статусом 200

50921 байтов - в 33951 со статусом 200

32390 байтов - в 501 со статусом 200

0 байтов - в 592 со статусом 304

552 байтов - в 18681 со статусом 405

150 байтов - в 20312 со статусом 405

190 байтов - в 154 со статусом 503

592 байтов - в 190 со статусом 503

562 байтов - в 303 со статусом 504

160 байтов - в 219 со статусом 504

0 байтов - в 151 со статусом 408