* **DDOS атака – что это?**

В общем случае DDOS (по-простому "дедос") - атака, во время которой злоумышленники атакуют ваш сервер (выделенный, облачный или виртуальный) большим количеством запросов, обработка которых отбирает драгоценные ресурсы у сервера, который в свою очередь не может обслужить обычных посетителей.

* **Какой ты, DDoS?**

Все DDoS атаки можно разделить на три обширные группы:

**Атаки, направленные на объём**

Данная категория атак направлена на насыщение полосы пропускания, соответственно, сила атаки измеряется в битах в секунду.

К этой категории разносятся различные виды флудов: UDP (На случайные порты хост-машины приходит бесчисленное количество пакетов, принуждая постоянно проверять, слушает ли данный порт какое-то приложение, и в случае ошибки возвращать пакет «ICMP Destination Unreachable». Естественно, такая активность поглощает ресурсы хост-машины, приводя к её недоступности.), ICMP (схож с UDP,  с максимальной частотой посылаются ping-запросы, принуждающие её давать эхо-ответы. Инструмент проверки доступности сетевого узла становится злобным пожирателем системных ресурсов.) и прочие потоки сфальсифицированных пакетов.  
  
**Атаки на уровне протоколов**

Эта категория направлена на ограничения оборудования или уязвимости различных протоколов. Такие атаки забивают ресурсы сервера либо промежуточного оборудования (фаерволы, балансировщики нагрузки и т.п.) паразитными пакетами, в результате чего системы оказываются неспособны обрабатывать полезные. Сила атаки измеряется в пакетах в секунду.

К этой категории относятся **SYN флуд** *(Машина, инициирующая соединение, отправляет хост-машине SYN пакет. Хост отвечает пакетом SYN-ACK, на что машина-инициатор должна ответить ACK пакетом. В случае SYN флуда ACK пакет не отправляется, в результате чего соединение некоторое время висит открытым и закрывается по тайм-ауту. Так как количество подключений, которые одновременно могут поддерживаться хост-машиной открытыми, ограничено, рано или поздно наступает насыщение, приводящее к отказу в обработке полезных пакетов.),*

**«пинг смерти»** *(Максимальный размер IP пакета — 65535 байт. Однако в процессе передачи по сети пакет дробится на части, соответствующие размеру окна. В результате манипуляций с получившимися субпакетами можно добиться, что при обратной «сборке» получится пакет, превышающий максимальный размер. Это может привести к переполнению выделенного буфера памяти и отказу в обслуживании для других пакетов.),* **атаки с фрагментированными пакетами и другие.**  
  
**Атаки на уровне приложений**

Как можно понять из названия, атаки направлены на уязвимости в приложениях и операционных системах (Apache, Windows, OpenBSD и т.п.). Они приводят к неработоспособности какого-либо приложения или ОС в целом. Как правило, состоящие из вполне невинных свиду запросов, такие атаки «ложат» веб-сервер. Интенсивность измеряется в запросах в секунду.  
  
Данный тип атак наиболее «убийственный». Они чрезвычайно узко направлены, благодаря чему могут создать весьма серьёзные проблемы атакуемому при малых затратах ресурсов атакующего. За последние 3-4 года данный тип атак становится преобладающим, и простой флуд HTTP GET запросов является одним из наиболее распространённых видов.

Среди таких атак: **Slowloris (***Атакующий сервер пытается открыть как можно больше HTTP соединений и держать их как можно дольше, понемногу отправляя частичные запросы. Лимит одновременных подключений на атакуемом сервере весьма быстро заканчивается, и он перестаёт принимать полезные запросы.***)**, **атаки нулевого дня** (К этому типу относят атаки, которые фиксируются впервые) и прочие.

* **Методы защиты**

1. **Отказаться от Windows Server**

Практика подсказывает, что сайт, который работает на винде (2003 или 2008 — неважно), в случае DDoS обречен. Причина неудачи кроется в виндовом сетевом стеке: когда соединений становится очень много, то сервер непременно начинает плохо отвечать.

* **2. Расстаться с Apache**

Если у вас стоит Apache, то как минимум поставьте перед ним кеширующий прокси — nginx или lighttpd. Apache’у крайне тяжело отдавать файлы, и, что еще хуже, он на фундаментальном уровне (то есть неисправимо) уязвим для опаснейшей атаки Slowloris, позволяющей завалить сервер чуть ли не с мобильного телефона.

* **3. Использовать модуль testcookie**

Если на ваш сайт приходит DDoS, то максимально действенным способом дать отпор может стать модуль [testcookie-nginx](https://github.com/kyprizel/testcookie-nginx-module). Идея простая. Чаще всего боты, реализующие HTTP-флуд, довольно тупые и не имеют механизмов HTTP cookie и редиректа. Testcookie-nginx работает как быстрый фильтр между ботами и бэкендом во время DDoS-атаки, позволяющий отсеивать мусорные запросы. Что входит в эти проверки? Умеет ли клиент выполнять HTTP Redirect, поддерживает ли JavaScript, тот ли он браузер, за который себя выдает (поскольку JavaScript везде разный и если клиент говорит, что он, скажем, Firefox, то мы можем это проверить). Проверка реализована с помощью кукисов с использованием разных методов:

* **4. Код 444**

Типичный пример — поиск, который выполняет сложные запросы к базе. Естественно, этим могут воспользоваться злоумышленники, зарядив сразу несколько десятков тысяч запросов к поисковому движку. Что мы можем сделать? Временно отключить поиск. Пускай клиенты не смогут искать нужную информацию встроенными средствами, Nginx поддерживает нестандартный код 444, который позволяет просто закрыть соединение и ничего не отдавать в ответ

**5. Баним по геопризнаку**

 Вы можете жестко ограничить отдельные страны, от которых испытываете неудобство. Скажем, вряд ли у интернет-магазина фотоаппаратов из Ростова-на-Дону много пользователей в Египте. Это не очень хороший способ (прямо скажем — отвратительный), поскольку данные GeoIP неточны, а ростовчане иногда летают в Египет на отдых. Но если вам терять нечего, то следуйте инструкциям::

1. Подключите к nginx GeoIP-модуль ([wiki.nginx.org/HttpGeoipModule](http://wiki.nginx.org/HttpGeoipModule)).
2. Выведите информацию о геопривязке в access log.
3. Далее, модифицировав приведенный выше шелл-скрипт, проgrepайте accesslog nginx’а и добавьте отфутболенных по географическому признаку клиентов в бан.

* **6. Юзайте профайлер и отладчик**

Для наиболее распространенной платформы создания веб-сайтов — PHP + MySQL — узкое место можно искать с помощью следующих инструментов:

* + профайлер Xdebug покажет, на какие вызовы приложение тратит больше всего времени;
  + встроенный отладчик APD и отладочный вывод в лог ошибок помогут выяснить, какой именно код выполняет эти вызовы;
  + в большинстве случаев собака зарыта в сложности и тяжеловесности запросов к базе данных. Здесь поможет встроенная в движок базы данных SQL-директива explain.
* **7. Анализируйте ошибки**

Проанализируйте объем трафика, время ответа сервера, количество ошибок. Для этого смотрите логи. В nginx время ответа сервера фиксируется в логе двумя переменными: request\_time и upstream\_response\_time. Первая — это полное время выполнения запроса, включая задержки в сети между пользователем и сервером; вторая сообщает, сколько бэкенд (Apache, php\_fpm, uwsgi…) выполнял запрос. Значение upstream\_response\_time чрезвычайно важно для сайтов с большим количеством динамического контента и активным общением фронтенда с базой данных, им нельзя пренебрегать. В качестве формата лога можно использовать такой конфиг:

* **8. Отслеживайте количество запросов в секунду**

Также посмотрите на число запросов в секунду. В случае nginx вы можете примерно оценить эту величину следующей shell-командой (переменная ACCESS\_LOG содержит путь к журналу запросов nginx в combined-формате):

По сравнению с нормальным для этого времени дня уровнем количество запросов в секунду может как падать, так и расти. Растут они в случае, если пришел крупный ботнет, а падают, если пришедший ботнет обрушил сайт, сделав его полностью недоступным для легитимных пользователей

* **9. Лимитируем ресурсы (размеры буферов) в nginx**

Про что нужно помнить в первую очередь? Каждый ресурс имеет лимит. Прежде всего это касается оперативной памяти. Поэтому размеры заголовков и всех используемых буферов нужно ограничить адекватными значениями на клиента и на сервер целиком. Их обязательно нужно прописать в конфиге nginx.

* **client\_header\_buffer\_size\_**\_ Задает размер буфера для чтения заголовка запроса клиента. Если строка запроса или поле заголовка запроса не помещаются полностью в этот буфер, то выделяются буферы большего размера, задаваемые директивой large\_client\_header\_buffers.
* **large\_client\_header\_buffers** Задает максимальное число и размер буферов для чтения большого заголовка запроса клиента.
* **client\_body\_buffer\_size** Задает размер буфера для чтения тела запроса клиента. Если тело запроса больше заданного буфера, то все тело запроса или только его часть записывается во временный файл.
* **client\_max\_body\_size** Задает максимально допустимый размер тела запроса клиента, указываемый в поле «Content-Length» заголовка запроса. Если размер больше заданного, то клиенту возвращается ошибка 413 (Request Entity Too Large).
* **10. Настраиваем тайм-ауты в nginx**

Ресурсом является и время. Поэтому следующим важным шагом должна стать установка всех тайм-аутов, которые опять же очень важно аккуратно прописать в настройках nginx.

* **reset\_timedout\_connection on;** Помогает бороться с сокетами, зависшими в фазе FIN-WAIT.
* **client\_header\_timeout** Задает тайм-аут при чтении заголовка запроса клиента.
* **client\_body\_timeout** Задает тайм-аут при чтении тела запроса клиента.
* **keepalive\_timeout** Задает тайм-аут, в течение которого keep-alive соединение с клиентом не будет закрыто со стороны сервера. Многие боятся задавать здесь крупные значения, но мы не уверены, что этот страх оправдан. Опционально можно выставить значение тайм-аута в HTTP-заголовке Keep-Alive, но Internet Explorer знаменит тем, что игнорирует это значение
* **send\_timeout** Задает тайм-аут при передаче ответа клиенту. Если по истечении этого времени клиент ничего не примет, соединение будет закрыто.
* **Знайте, с кем связаться**

Вы должны чётко знать, к кому обратиться в случае, если Вы уже под атакой либо чувствуете её приближение. Это может быть внутренний отдел безопасности, удалённый сотрудник, инженер дата-центра, сервис сопровождения по вопросам безопасности и т.п.

* **Зачем это нужно?**
* недобросовестная конкуренция
* мести обиженного клиента неугодному сервису
* интернет-вандализм
* идеологический хактивизм

**Хактиви́зм** — использование [компьютеров](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и [компьютерных сетей](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) для продвижения политических идей, [свободы слова](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%B0_%D1%81%D0%BB%D0%BE%D0%B2%D0%B0), защиты [прав человека](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B0%D0%B2%D0%B0_%D1%87%D0%B5%D0%BB%D0%BE%D0%B2%D0%B5%D0%BA%D0%B0) и обеспечения [свободы информации](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%B0_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8)

**Доступно?**

* Приложения для проведения DDoS атак
* Аренда ресурсов ботнета: 5$ за час, от 40$ за сутки

Безмерно доступно. На сегодняшний день существует немало свободно доступных в интернете приложений для проведения DDoS атак. Некоторые из них используют механизмы атак, которым сложно противодействовать, другие позволяют объединить всех пользователей в добровольный ботнет, что даёт возможность пользоваться чужими ресурсами для проведения атак и раздавать взамен свои. При этом таким на первый взгляд аматорским атакам бывает сложно противостоять даже хорошо подготовленным коммерческим системам.  
  
Другой способ — аренда ресурсов ботнета. Интернет полон ресурсов, предоставляющих подобные услуги по весьма символическим ценам: от $5 за час, от $40 за сутки. За такие, можно сказать, смешные деньги запросто можно «заказать» своих конкурентов по электронной коммерции и принести им намного более существенные убытки.