безопасность DNS — вопросы без ответов, ответы в пустоту

крис касперски, no-email

DNS-протокол, ставший неотъемлемой частью всемирной сети, крайне небезопасен по своей природе, что делает его весьма привлекательным для хакеров, вынуждающих производителей выпускать многочисленные исправления. последняя крупная реконструкция состоялась в августе 2008 года, однако, не успела она завершиться как хакеры предложили новые типы атак, пробивающих свежие заплатки и все вернулась на круги своя. представляет интерес рассмотреть эволюцию защитных средств — быть может, это подскажет нам, что ждет нас в будущем.

введение

Главным образом DNS протокол предназначен для разрешения доменных имен, то есть преобразования их в IP-адреса (и, при необходимости, наоборот). Определение адресов почтового сервера, обслуживающего данный домен, так же осуществляется не без помощи DNS. Например, мы посылаем письмо на адрес info@mamba.com — почтовый клиент вычленяет домен mamba.com и посылает DNS-серверу запрос — какой почтовый сервер "курирует" его. Это может быть и сам mamba.com и, например, mail.company.jp. Выходит, что на DNS держится и WEB, и почта, и многие другие сетевые службы.

Что произойдет, если в ответ на запрос, посланный DNS серверу, вернется хакерский пакет, утверждающий, что письма, адресованные @mamba.com, следует слать, скажем, на ppp666.small-ISP.ru? Жертва (а точнее используемый ей почтовый клиент), именно туда их и перешлет, в результате чего хакер получит доступ к конфиденциальной информации. Аналогичным образом работает перенаправление при заходе на сайты.

К огромной радости для хакеров, навязать жертве подложный DNS-ответ очень легко, а последствия атаки зачастую носят катастрофический характер, причем, в силу децентрализованной организации DNS-серверов, решить проблему "на месте", частным образом, невозможно. Установить надежный DNS-сервер может каждый — как домашний, так и корпоративный пользователь, однако, наш DNS осведомлен только о тех доменах, которые он "курирует" (доменах корпоративной или домашней сети). За все остальные отвечают "чужие" DNS-сервера, которым наш DNS вынужден посылать запросы, искренне веря в то, что ему возвратят честный ответ. А если нет? Допусти, DNS, которому адресован запрос, атакован, тогда, независимо от степени защищенности нашего DNS, письмо с конфиденциальной информацией, попадет в лапы хакера и мы никак не сможем этому противостоять.

Актуальность атак на DNS растет с каждым днем, а защищенность (даже с учетом последних обновлений) остается на уровне 90х годов прошлого века. Причем, подавляющее большинство администраторов даже и не пытаются защититься, а если и пытаются, то не знают как, поскольку в популярной литературе эта тема описана очень поверхностно. Настоящей статьей мы попытались заполнить этот пробел.

Figure 1: Root Server Locations and Areas of Redundant Connectivity

Рисунок 1 географическое расположение корневых DNS-серверов

fundamentals

В популярной форме DNS-протокол достаточно подробно описан на Википедии (http://en.wikipedia.org/wiki/Domain_Name_System), так что ограничимся поверхностным изложением наиболее фундаментальных основ.

DNS представляет собой протокол прикладного уровня, работающий поверх протоколов UDP или TCP, причем, UDP используется намного чаще, поскольку TCP (при высоком уровне нагрузки на сервер) не обеспечивает надлежащей производительности и обладает большей латентностью (т.е. возвращает ответ не так быстро, как это делает UDP).

Имеющиеся DNS-сервера можно (условно) разделить на две большие группы. Первые обслуживают только "закрепленные" за ними домены (домены локальной корпоративной сети, например) и в ответ на вопрос "какой IP адрес имеет www.microsoft.com" возвращают ошибку. Говоря техническим языком — они не принимают рекурсивных DNS-запросов. Что такое рекурсивный запрос? Если DNS-сервер не располагает информацией о заданном доменном имени, он посылает запрос вышестоящему DNS-серверу, который в свою очередь может вернуть либо готовый IP-адрес, либо адрес более "компетентного" DNS-сервера.

DNS-сервера, устанавливаемые Интернет-провайдерами, относятся ко второму типу, принимая рекурсивные запросы и возвращая ошибку только в том случае, когда узла с обозначенным именем либо вообще не существует, либо один из промежуточных DNS-серверов перегружен или еще не успел обновить информацию в своих таблицах.

Наверху иерархии находятся так называемые "корневые" (root) DNS-сервера, к которым может (бесплатно!) обращаться любой желающий. Для этого достаточно установить у себя локальный DNS-сервер работающий (в общем случае) намного быстрее и надежнее, чем DNS провайдера, который к тому же часто становится объектом хакерских атак. Однако, не стоит думать, что обращение к корневым серверам безопасно. Сами корневые сервера действительно надежно защищены, но они хранят в своих базах отнюдь не IP-адреса конечных узлов, а адреса DNS-серверов, обслуживающих данные узлы и вот эти самые сервера защищены намного хуже, если вообще защищены и потому могут быть легко атакованы.

Клиентская часть DNS-протокола реализована в виде так называемого "резольвера" (resolver) так же называемого "стабом" (stub), посылающего DNS-серверу рекурсивные запросы и получающего ответы. Кстати говоря, всякий DNS-сервер включает в себя "резольвер", а потому, с небольшой натяжкой можно сказать, что стаб — это "усеченный" DNS-сервер, обслуживающий только данный компьютер со всеми установленными на нем приложениями.

Действительно, когда The Bat! или FireFox пытаются разрешить доменное имя, то они обращаются к локальному стабу, точно так, как сам стаб обращается к DNS-серверу. То есть, теоретически, без стаба можно и обойтись, формируя DNS-запросы самостоятельно, однако, практически все существующие приложения опираются на готовый стаб, поскольку, это проще. Расплатой за простоту становится небезопасность. Приложение, самостоятельно формирующее

DNS-запросы, может предпринять дополнительные действия по блокировке входящих подложных DNS-ответов, сформированных хакером. Приложение, опирающееся на стаб операционной системы, оказывается у нее в заложниках.

сценарии атак

DNS протокол, работающий на базе TCP, а так же безопасное расширение известное под названием DNSSEC, защищены _намного_ лучше тех реализаций, что работают поверх UDP, однако, мы будем рассматривать именно DNS over UDP, поскольку, массовая миграция на безопасные решения в рамках глобальной сети попросту невозможна, причем, если выбор между TCP и UDP решается установкой одной-двух галочек в свойствах сервера, то DNSSEC поддерживают далеко не все DNS-сервера. К чему это ведет?!

Допустим, я владею доменом nezumi.org.ru (я действительно владею им!) и у меня установлен свой собственный DNS-сервер, располагающий информацией о поддоменах типа www.nezumi.org.ru, ftp.nezumi.org.ru, user_name.nezumi.org.ru и т.д. (у меня действительно установлен такой сервер). Представим, что некий пользователь хочет выяснить какой IP-адрес соответствует имени ftp.nezumi.org.ru. Он отправляет запрос корневым серверам, которые возвращают ему адрес DNS, обслуживающего зону .ru. Сервер, обслуживающий зону .ru, знает адрес DNS'a, закрепленного за ".org.ru", который, в свою очередь, знает адрес моего DNS-сервера.

Вот теперь моему серверу направляется "безопасный" DNS-запрос в формате DNSSEC об адресе ftp.nezumi.org.ru, а в ответ… тишина. Ну не поддерживает мой сервер DNSSEC, ну что ты будешь делать!!! И не только он один не поддерживает! Пройдет еще немало времени, прежде чем DNSSEC станет стандартом де-факто, если вообще станет.

Рисунок 2 маленький домашний DNS сервер

Каким образом это можно использовать для атак?! Начнем с атаки на DNS-сервер. Классический сценарий (активно использовавшийся еще в 90х годах) построен приблизительно по следующей схеме. Сначала на атакуемый DNS сервер обрушивается шквал разнородных рекурсивных DNS-запросов с целью очистить DNS-кэш, вытеснив из него популярные доменные имена типа www.microsoft.com. Затем, когда информация о www.microsoft.com оказывается утрачена, атакующий посылает серверу рекурсивный DNS-запрос с просьбой разрешить доменное имя www.microsoft.com, которого уже нет в кэше и сервер вынужден обращаться к более компетентным товарищам. Вот тут-то жертву и ждет сюрприз. Вслед за запросом, хакер посылает подложный ответ от имени более компетентного DNS-сервера,

содержащий фиктивный IP-адрес www.microsoft.com, который будет сохранен в кэше жертвы и при всех последующих запросах, взломанный DNS сервер станет возвращать подложный IP-адрес, не имеющий ничего общего с www.microsoft.com и заманивающий жертву на хакерский узел со всеми вытекающими отсюда последствиями.

Возможности данной атаки весьма ограничены, поскольку хакер не может захватить более одного узла за раз, да и к тому же, у многих DNS-серверов кэш настолько емкий, что его не переполнишь, особенно, если политика вытеснения данных из кэша препятствует удалению популярных записей. В августе 2008 специалист по информационной безопасности Дэн Каминский (Dan Kaminsky) "воскресил" хорошо известный (и хорошо забытый!) альтернативный сценарий, позволяющий захватывать целые домены одним махом. Текст доклада (на английском, в формате mp3) выложен в отрытый доступ и с ним может ознакомится любой желающий: http://www.blackhat.com/html/webinars/kaminsky-DNS.html, если ему не наскучит слушать часовую речь.

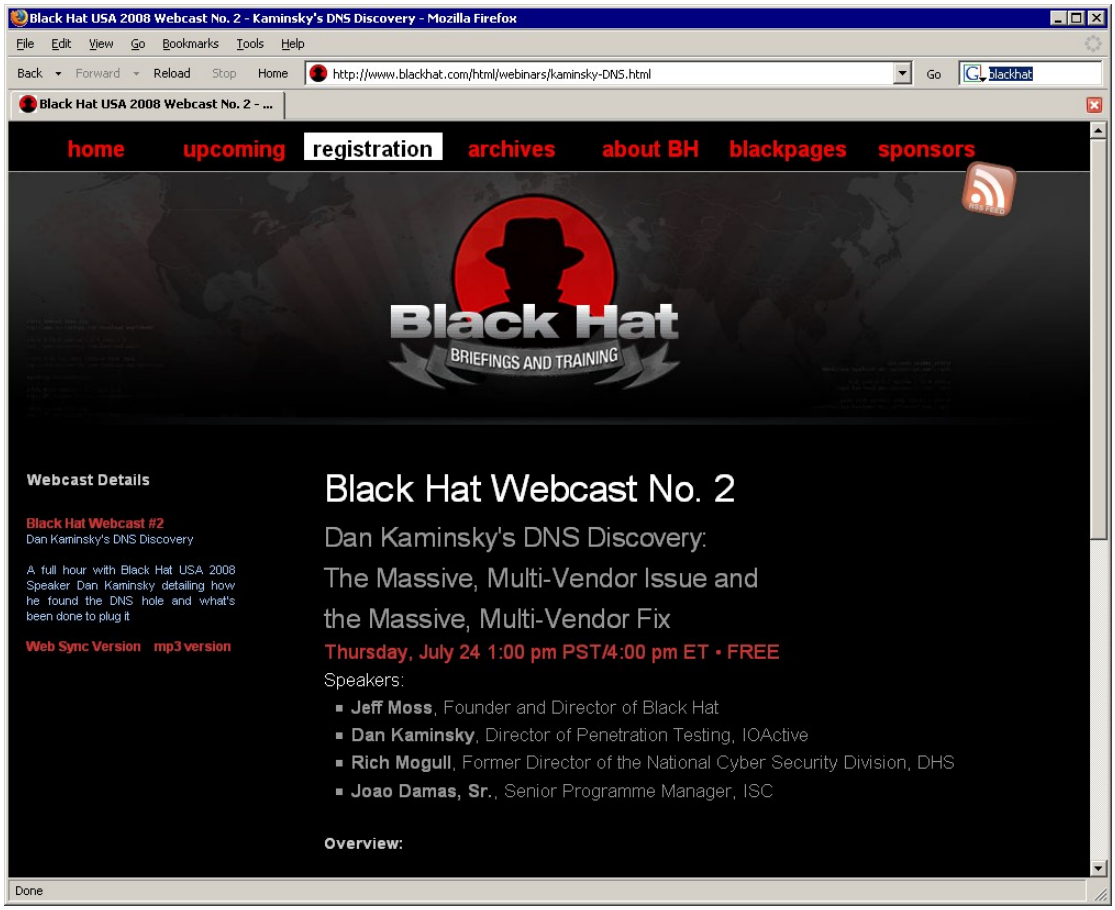

Рисунок 3 речь Дэна Каминского в mp3

Здесь же приводится квит-эссенция, кстати говоря, "реконструированная" задолго до самого доклада многими хакерами и специалистами по информационной безопасности. Все очень просто. Атакующий направляет серверу рекурсивный DNS-запрос с просьбой разрешить доменное имя I_just_love_you_william_I_wish_you_live_forever.microsoft.com, которого, естественно, в DNS кэше нет, поскольку такого имени нет вообще. Но ведь сервер об этом не знает!!! И потому посылает своим собратьям рекурсивный запрос: "а скажите-ка мне друзья…" и тут же получает подложный ответ от хакера, в котором не только указан IP-адрес узла I_just_love_you_william_I_wish_you_live_forever.microsoft.com (фиктивный, разумеется), но и адрес DNS-сервера якобы лучше других осведомленного о поддоменах microsoft.com. Атакуемый DNS запоминает эти данные и теперь _любые_ запросы на разрешение имен в зоне *.microsoft.com отправляются прямиком на хакерский сервер! Последствия атак данного типа носят воистину убийственный характер.

Что же касается стабов (то есть клиентских компьютеров), то к ним применим только первый сценарий атаки, однако, на практике его оказывается более чем достаточно. Например, хакер может навязать "посторонний" сервер обновлений, с которого система автоматически

скачает троянизированные заплатки, обеспечивая червям и вирусам все необходимые условия для размножения.

от одной дыре к другой

Подделать DNS-ответ достаточно просто, особенно, если сервер работает на UDP-протоколе. Для успешной атаки достаточно угадать (или подобрать) всего два 16-битных поля. Идентификатор транзакции (TXID) и порт источника (SP#). В ранних версиях DNS-серверов порт источника был фиксирован, а TXID представлял собой вполне предсказуемое число, увеличивающееся на единицу с каждый посланным DNS-запросом и уменьшающееся с каждым DNS-ответом. Хакеру было достаточно послать всего десяток подложных пакетов, чтобы хотя бы один из них воспринимался жертвой как подлинный со всеми вытекающими отсюда последствиями.

После серии успешных атак, разработчики DNS использовали несложную временную функцию, базирующуюся на системном таймере, и "планомерно" увеличивающую TXID. Но хакеры даже и не думали расстраиваться. Это _никак_ не усложняет атаку на DNS-сервера. Злоумышленник просто отправляет последовательность DNS-запросов и смотрит на заголовки ответов, определяя текущий TXID плюс приблизительный инкремент. На несильно загруженных серверах (где атакующий едва ли не единственный вопрошающий) мы получаем очень гладкую кривую арифметической прогрессии, позволяющую рассчитывать последующие TXID на базе предыдущих.

Со стабами это уже не работает, поскольку они не принимают запросов, а только посылают ответы, однако… зная приблизительное время работы компьютера после последней (пере)загрузки операционной системы мы можем рассчитать ожидаемый TXID, особенно если форсируем перезагрузку, например, DoS атакой. В любом случае, реальная энтропия намного ниже 16-бит, поскольку, низкое таймерное разрешение приводит к тому, что младшие биты представляют собой константу. Старшие биты так же чаще всего равны нулю (счетчик еще не "намотал" нужное количество тиков). Короче, изменяется только середина. А вот ее-то как раз и легко угадать.

ОК, разработчики использовали генератор псевдослучайных чисел, выбирая TXID наугад, что было должно положить конец атакам, но… не положило, особенно в тех случаях, когда использовался стандартный библиотечный rand() или другая очень слабая псевдослучайная функция, позволяющая предсказать последующий член последовательности на базе предыдущих, значения которых атакующий может получить, направляя серверу вполне легальные DNS-запросы. Со стабами ситуация несколько сложнее, однако, учитывая, что rand() в большинстве случаев инициализируется значением системного таймера — реальная энтропия все равно оказывается намного ниже 16 бит.

Переход на криптостойкие псевдослучайные функции обещал настоящий прорв в области безопасности, однако, 16 бит — это все-таки очень небольшая величина и атакующему в среднем достаточно (в среднем) послать 2^16/2 = 32.768 подложных пакетов для успешной атаки. Конечно, если их посылать одному узлу, то такая подозрительная активность легко выявляются любой системой обнаружения вторжений (это раз) и хакер просто не успеет опередить настоящий DNS-сервер, после ответа которого жертва прекращает прием хакерских пакетов.

Однако, возросшая скорость телекоммуникационных каналов уже к началу 2000х годов позволила передавать заданное кол-во пакетов за доли секунды, что сопоставимо (или даже меньше) времени ответа от настоящего DNS-сервера. К тому же, если хакер (или сетевой червь) не имеет конкретной цели, а просто рассылает пакеты на случайные узлы, то разослав всего по одному пакету на 32.768 узлов, он с высокой степенью вероятности взламывает хотя бы одну машину. А что такое 32.768 узлов для современного Интернета?!

Рисунок 4 Дэн Каминский на BlackHat'e

Для предотвращения атак Дэн Каминский предложил рандомизовать не только TXID, но и порт источника, что дает нам энтропию близкую к 32-битам и метод слепого перебора уже отдыхает. Но это в теории. На самом деле, часть портов зарезервирована под другие нужды и реально DNS-сервер может использовать лишь ограниченный диапазон SP#, что дает нам энтропию от 20 до 24 (28) бит. Конечно, 20 (и особенно 28) намного больше 16ти, но…

Хакеры с ходу изобрели атаку именуемую "port exhausting" и опирающуюся на ограниченное количество портов источника. Достаточно обрушить на сервер шквал рекурсивных запросов, поступающих быстрее, чем возвращаются ответы от вышестоящих серверов, как через короткое время практически все порты окажется занятыми и количество оставшихся обратится в нуль, следовательно, предсказать очередной назначаемый порт хакер сможет без особого труда.

Выходит, что мы снова скатываемся к 16ти битам TXID'a? Нет, не выходит. Уже не выходит. Объемы жестких дисков и оперативной памяти за последние годы существенно возросли и даже очень качественные криптографически стойкие пседослуайные функции оказываются вполне предсказуемыми, поскольку каждый последующий член не абсолютно случаен, а генерируется на базе ему предшествующих. Существует возможность создать заранее предвычисленные таблицы, позволяющие по нескольким предыдущим номерам TXID угадывать последующий. Для Server 2003 и Server 2008 (со всеми установленными заплатками) эти таблицы занимают… 800 Гигабайт плюс еще 100 Гбайт расходуются на индексы, плюс еще 1 Гбайт на индексы верхнего уровня, хранимые в оперативной памяти. Результат — фантастически точные предсказания TXID, который замышлялся как абсолютно случайный и непредсказуемый. А что такое 900 Гбайт по современным понятиям? Собрать такой компьютер по силам даже домашнему пользователю, не говоря уже о заказных коммерческих взломах.

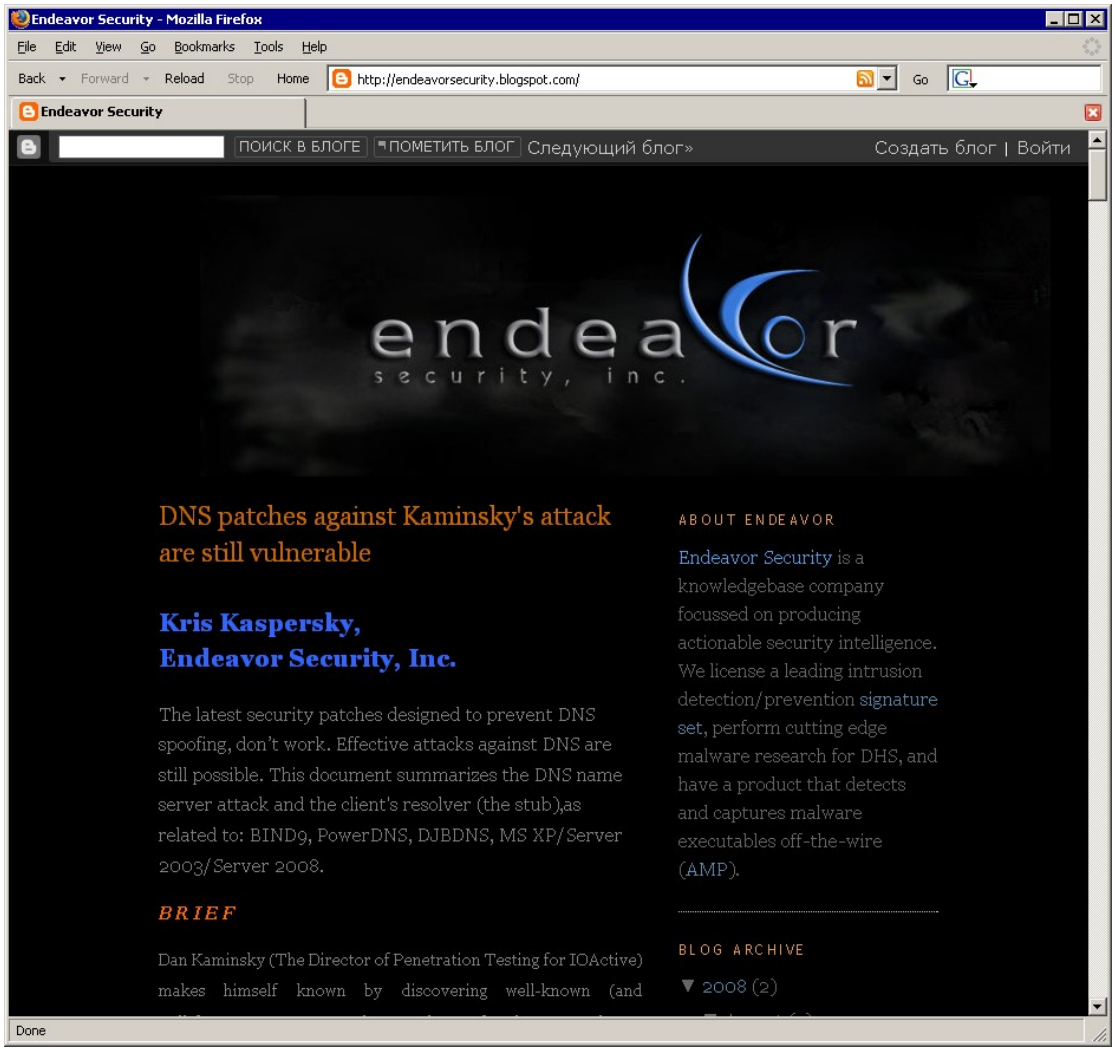

Рисунок 5 исследования автора на блоге компании Endeavor Security, Inc

заключение

DNS сервера и стабы непрерывно лаются с момента введения их в массовую эксплуатацию, однако, дыры не только не исчезают, но продолжают прибывать ударными темпами. Вот и очередная грандиозная реконструкция, спровоцированная Дэном, обернулась лишь вредом, в котором нет ни грамма пользы. По сути дела, Дэн взбудоражил хакеров, многие из которых до этого были совершенно не в курсе темы, а теперь активно пишут атакующие программы число (и качество) которых стремительно растет и если не случится чуда, то всемирную сеть ждут довольно мрачные деньки.

Впрочем, ничего ужасного не случится. Автором этой статьи были разработаны алгоритмы распознавания атак на DNS, а фирма Endeavor Security, Inc (крупнейший поставщик сигнатур для систем обнаружения/предотвращения вторжений) уже разослала всем ведущим производителям сетевого оборудования/программного обеспечения (CISCO, DLINK, Checkpoint и т.д.) необходимые обновления, блокирующие межсетевые атаки. Так что глобальных эпидемий не случится, но вот в рамках одной подсети (или совокупности нескольких сетей) — атаки уже фиксируются!

Без всяких претензии на рекламу — сигнатуры поставляемые компанией Endeavor Security, Inc работают и блокируют нарушителей, однако, приобретать их непосредственного у самой компании — расточительно (подписка стоит порядка $13.000 в год), однако, ничто не мешает воспользоваться более дешевой продукцией одного из многочисленных подписчиков Endeavor'a, список которых можно найти на сайте: www.EndeavorSecurity.com

Рисунок 6 набор сигнатур FirstLight, поставляемый компанией Endeavor Security, Inc