смертельная схватка — атака на DNS: Дэн Каминский против Криса Касперски белые начинают и проигрывают

крис касперски ака мыщъх, no-email

Дэн Каминский, сообщивший о дыре в DNS, вызвал бурный интерес общественности, подогреваемый замалчиванием деталей атаки, но хакеры реконструировали цепь событий, создав боевые exploit'ы, блокируемые только самыми свежими заплатками, проанализировав которые, мыщъх обнаружил намного более серьезные дыры чем Каминский…

введение или хронология событий

Обсуждения не безопасности протокола DNS начались практически одновременного с введением последнего в "промышленную" эксплуатацию. Только за последний пяток лет можно насчитать несколько десятков серьезных публикаций, исследовательских отчетов, бюллетеней безопасности. Но все эти разговоры велись в сугубо технических кругах и до широкой общественности не доходили. Массовые атаки на DNS сервера, зафиксированные еще в 90х годах, переворота не вызывали. DNS протокол во всех его реализациях продолжил степенное эволюционное развитие, а от атак приходилось отбиваться путем воздвижения дополнительных систем обнаружения вторжений.

Готовясь к очередной хакерской конференции Black Hat USA 2008 Дэн Каминский (Dan Kaminsky, dan@doxpara.com) – сотрудник компании IOActive, подготовил доклад, демонстрирующий новые атаки на DNS сервера, которые сам же и описал в статье "Kaminsky: Rootkit Causing Widespread Infection", опубликованной изданием www.eff.org еще в середине декабря 2005 года, но оставшейся незамеченной.

В 2007 году вышло множество отчетов экспертов по безопасности, касающихся DNS, и производители нехотя начали затыкать дыры, относящиеся еще к так называемой "Joe Stewart Birthday Attack", описанной в ноябре 2002 года и до сих пор не утратившей актуальности, хотя с тех времен очень многое изменилось.

Рисунок 1 Дэн Каминский собственной персоной

Каминский не был первым, кто обнаружил дыру в DNS, он даже не был тем, кто что-то вообще _обнаружил_ (читай: нашел нечто новое). Обычный PR. Внимание прессы. Ответная (вынужденная) реакция разработчиков DNS-серверов и операционных систем. Текст презентации отсутствует, Каминский отказывается разглашать технические детали, нагнетая мрачную атмосферу ужаса и паники.

Компании, специализирующиеся на компьютерной безопасности, лихорадочно анализировали заплатки, выпущенные производителями, пытаясь определить что именно было исправлено и куда направлен вектор атаки. Как и следовало ожидать, Каминский не откопал ничего принципиально нового, хотя и извлек из тьмы кромешной ранее использованные, но малоизвестные молодому хакерскому поколению атаки на DNS, устраняемые путем затягивая гаек и болтов, то есть усиления процесса латания DNS, начавшегося задолго до Каминского.

Между тем, на одном из блогов появилось достаточно полное описание предполагаемого сценария атаки, которое по одной версии было выболтано Каминским по пьяни, по другой — независимо реконструировано хакером Халваром Флейком (Halvar Flake) на основе анонса презентации Каминского. В настоящий момент убраны и анонс, и текст Халвара Флейка, на месте которого красуется "джентльменское" извинение, однако, как говориться, что в Сеть попало, то пропало и описание атаки немедленно расползлось по десяткам сайтов, форумов, блогов, живых журналах и прочих средств массовой информации, где их не найдет только ленивый.

Каминский еще не успел зачитать свою презентацию, как (недели за две до нее) на metasploit'е появилась пара свеженьких exploit'ов, один из которых предназначен для атак на DNS-сервера, другой — на рабочие станции. Анализ обоих (выполненных мыщъхем по долгу служебной необходимости) не выявил ничего принципиально нового. Такие трюки мыщъх использовал и сам, не для реальных атак, конечно, а для pen-testing'а, но что это меняет?!

Самое интересное — реальные дыры (о которых Каминский не имел ни малейшего представления) остались не заткнутыми и мыщъх сходу предложил два сценария внедрения в уже пропатченные системы, впрочем, не придав им большого значения, поскольку, Endeavor Security Inc (где сейчас работает мыщъх), в основном занимается разработкой и лицензированием сигнатур для различных систем обнаружения вторжения, многие из которых

не имеют даже пороговых датчиков, а в рамках "чистых" сигнатур (без привлечения специальных модулей) описать атаку на DNS практически невозможно в силу природы самой атаки. Практически. Но мыщъх все-таки описал, после чего связался с разработчиками осей и DNS на предмет: "так когда же будут готовы _нормальные_ патчи?!" (Корпоративный почтовый адрес — очень удобное средство для рассылок подобного рода).

И вот тут началось… Разработчики быстро въехали в ситуацию и попросили мыщъх'а отложить публикацию до тех пор, пока лекарство не будет готово. В смысле английскую публикацию. За русскую никакого базара не было, так что… читатели "Хакера" имеют возможность получить эксклюзивную информацию из первых рук, в смысле, лап первооткрывателя.

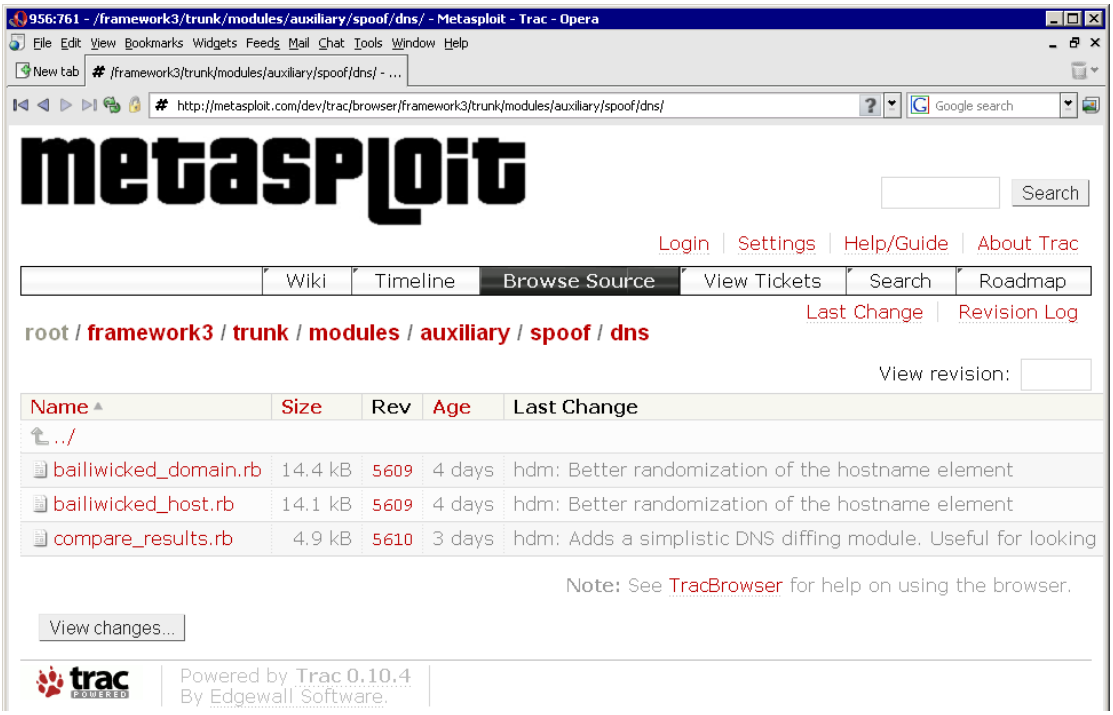

Рисунок 2 свежие exploit'ы на http://metasploit.com/dev/trac/browser/framework3/trunk/modules/auxiliary/spoof/dns/

>>> врезка подслушанный треп на форуме

В BIND еще в 1997 году была обнаружена обозначенная уязвимость. В OpenBSD ее пофиксили, а товарищи из ISC "исправили" ее по своему, в результате и появилась данная ошибка безопасности, о которой сегодня идет речь. Товарищи из OpenBSD несколько раз предупреждали ISC о том, что их реализация не является наилучшей, но как нетрудно предположить, ISC данные рекомендации игнорировал. В результате, в OpenBSD просто портировали свой же патч из предыдущей версии BIND в BIND 9. Так что в новости слово "опций" скорее всего следует читать как "фич" ;)

восход солнца над Сан-Франциско или как это работает

Короче. По существу. DNS протокол может работать как поверх TCP, так и поверх UDP, причем в 99% случаев используется именно UDP как более быстрый, менее ресурсоемкий, но в тоже время и менее защищенный. Чтобы послать подложный пакет, который будет воспринят жертвой как правильный, достаточно угадать (подобрать) идентификатор последовательности (TXID) и номер порта-отправителя (SP#). На этом заканчивается первая фаза атаки и начинается вторая.

В простейшем случае злоумышленник может отправить подложный DNS-ответ с подложным IP-адресом некоторого узла, на который ломится жертва, и куда она втихую будет перенаправлена. Хорошая идея — навязать ложный IP сервера обновлений и заменить жертву на хакерский узел, где лежат "хакнутые" заплатки, начиненные червями или прочей заразой.

Сложность реализации атак подобного рода в том, что рабочие станции кэшируют DNS запросы, более того, система не принимает DNS-ответов, которые не запрашивались. То есть, хакер должен дождаться момента, когда жертва пошлет DNS-запрос и сгенерировать подложный ответ прежде, чем это сделает настоящий DNS-сервер! На самом деле, обе проблемы имеют весьма элегантное решение. DNS-кэш обычно невелик и по умолчанию вмешает в себя около сотни отрезолвленных IP-адресов, а потому послав жертве HTML письмо с кучей картинок, лежащих на внешних серверах с разными доменными именами, хакер вытеснит из кэша все старые записи после чего последняя ссылка в письме, ведущая на сервер обновлений, гарантированно пошлет обозначенный запрос в сеть. Предшествующая ей ссылка на WEB-сервер, подконтрольный хакеру, подскажет точное время когда следует начинать генерацию подложных пакетов. Если хотя бы один из них будет воспринят как правильный, в DNS-кэш попадет "левый" адрес сервера с апдейтами, имеющий все шансы "дожить" до очередной сессии обновлений.

Атака на DNS-сервер сулит еще большие перспективы. Допустим, мы отправляем серверу запрос на разрешение доменного имени xxx.google.com, заведомо отсутствующее в его кэше, поскольку, такого имени вообще нет и атакуемый DNS-сервер обращается к вышестоящим серверам за помощью. Если хакер успеет возвратить подложный пакет впереди всех и этот пакет будет воспринят жертвой как правильный, атакуемый DNS-сервер запомнит хакерский IP, направляя ему все последующие доменные имена *.goodle.com для преобразования их в IP, считая его наиболее компетентным DNS-сервером, лучше других разбирающегося в домене *.goodle.com. То есть, хакер одним махом захватывает целый домен со всеми поддоменнами. Именно этот сценарий и предложил Каминский для атак.

На первый взгляд, ситуация близка к расстрелу, ведь захватывая домены один за другим, атакующий может манипулировать траекторией сетевого трафика по своему усмотрению, воруя конфиденциальную информацию, троянизируя исполняемые файлы и вытворяя кучу других фокусов. Однако, при ближайшем рассмотрении все не так просто. Сценарий Каминского известен со времен первой молодости Интернет и его эффективность слишком преувеличена. Чтобы захватить домен нужно послать подложный DNS-пакет, угадав TXID/SP#, что в общем случае требует посылки большого количества пакетов, легко засекаемых даже самой примитивной системой обнаружения вторжения. Во-вторых, информация о кредитках обычно передается через SSL (соединение заведомо устойчивое к перехвату), исполняемые файлы (особенно системные) снабжены цифровыми подписями, которые хрен подделаешь. В общем, даже в случае успешного исхода атаки, возможности хакера весьма ограничены, особенно если учесть существование дополнительных защитных комплексов, тех же антивирусов, например.

>>> врезка BIND и товарищи

Основная причина популярности BIND'а — инертность мышления. В роли кэширующего рекурсивного DNS-сервера BIND просто ужасен и уже на 1k рекурсивных запросах в секунду среднее время ответа возрастает сначала до сотен миллисекунд, постепенно увеличиваясь до десятков секунд! Реально под такой нагрузкой живет только PowerDNS, который к тому же поддерживает авторизацию (чего не умеет делать BIND). DJNDNS работает вполне устойчиво, однако, в силу выбора нестойкого алгоритма, рандомизации ломается чуть ли не в лет (даже с последними установленными заплатками, выпущенными после Каминского);

о чем молчат заплатки

Латать DNS-сервера и DNS-резолверы (входящие в состав всех операционных систем) начали задолго до "дыры" Каминского. Древние системы использовали инкрементный TXID (увеличивающийся на единицу с каждым DNS-запросом) и фиксированный порт источника, что делало DNS-атаки чрезвычайно простым занятием. Не нужно быть пророком, чтобы угадать заранее известную пару 16-битных чисел, варьирующуюся в очень узких пределах.

После первой волны атак разработчики, почесав у себя под хвостом, написали несложную функцию, генерирующую TXID на основе системных часов, значение которых удаленному атакующему неизвестно (ну, во всяком случае так принято считать), однако, это не сильно смутило хакеров (пионеров, пользующихся готовыми exploit'ами мы в расчет не берем). Во-первых, энтропия (то есть мера беспорядка) в этом случае намного ниже 16-бит. Младшие биты системных часов обычно представляют собой константу, поскольку, аппаратный таймер

не обеспечивает заданного временного разрешения. Старшие же биты так же обычно представляют собой константу и на достаточно коротком временном отрезке меняется только середина 16-битового поля TXID.

Во-вторых, посылая запросы DNS-серверу и получая от него пакеты, хакер "вытягивает" оттуда TXID, которые в подавляющем большинстве случаев ложатся на тривиальную прямую арифметической прогрессии с минимальным разбросом, которая тем меньше, чем ниже загруженность сервера и канала, связывающего его с атакующим. Хакеры вкурили в ситуацию и атаки вспыхнули с новой силой. Разработчики плюнули и сделали то, что им полагалась сделать с самого начала — полную рандомизацию TXID, что, кстати говоря, намного проще сказать, чем запрограммировать. Ведь TXID должны представлять собой уникальные идентификаторы и не повторяться дважды в течении короткого интервала времени. Алгоритмы, основанные на системных часах, обеспечивают такое распределение в силу "стрелы времени", а вот функции типа rand() требуют специальной доработки "напильником". Реализация усложняется, повышая вероятность косяков, но… чего не сделаешь ради безопасности?!

Порт источника долгое время рандомизировать не хотели, оставляя его как последний бастион, но… 16-битное поле TXID даже при 100% рандомизации (не достижимой на практике) не слишком-то хорошая защита и для успешной атаки хакеру достаточно в среднем послать 2^16/2 == 32.768 пакетов. Учитывая пропускные способности современных сетей да еще и наличие распределенных ботнетов, атака займет считанные минуты! Система обнаружения вторжений, конечно, среагирует, но противостоять атаке не сможет, т.к. не известно с какого узла ботнета придет следующий подложный пакет (а вот если хакер не меняет своего IP, то его легко заблокировать).

До Каминского большинство систем использовало простой инкрементный алгоритм (номер порта источника увеличивается на единицу до тех пор, пока не встретится первый свободный порт), но это касается исключительно рабочих станицей с DNS-резолверами. У DNS-серверов порт-источника обычно фиксирован и по умолчанию равен 53. Хотя разработчики BIND 9 заблаговременно предоставили опциональную возможность его рандомизации и когда владельцы других систем лихорадочно качают патчи, пользователи BIND 9 просто меняют одну строчку в конфиге, продолжая пить пиво, а потом меняют ее обратно, поскольку, рандомный порт на сервере это не есть хорошо — возникают проблемы с брандмауэрами, трансляторами сетевых адресов и куча прочих конфликтов.

Рисунок 3 блог Каминского на http://www.doxpara.com/ с DNS-чекером, предназначенным для "честной" (как пишет Каминский) проверки DNS на уязвимость, но (как показывает TCPDUMP) ведущий нечестную игру и вообще, Дэн проявляет большую активность, фиксируемую сенсорами распределенной сети компании Endeavor Security, что наводит на определенные размышления

мыщъхиный армагедон

Заплатки, выпущенные для предотвращения (читай: затруднения) атак на DNS-сервера и рабочие станции, прежде всего рандомизовали TXID вместе с портом-источника, а так же изменили политику кэширования DNS-ответов (в клинических случаях просто отключили DNS-кэш, наплевав на производительность). Казалось бы: ну вот теперь мы, наконец, защищены! Да как бы не так!!! Мыщъх, располагает по меньшей мере двумя сценариями эффективных атак, которые сейчас и опишет и которые "пробивают" полностью пропатченные: NT (вплоть до Вислы), Free/OpenBSD, BIND 9, DJNDNS, а так же частично PowerDNS.

задыхаясь от жажды

Хорошо, порт-источника рандомизирован и выбирается наугад из широкого пула (точные цифры варьируются от одной системы к другой и в грубом приближении составляет 16.384, т.е. 12-бит, хотя тот же PowerDNS использует всего 1024 пота, что при большом количестве запросов ведет к бальному DoS'у). Допустим, что сервер (или рабочая станция) использует криптостойкую функцию рандомизации, угадать следующее значение которой невозможно. Означает ли это, что атакующему необходимо перебирать 16 бит TXID вместе с 12 битами порта источника?

Конечно же нет! Порты — расходные материалы и они используется не только DNS'ом, но и многими другими службами, реализованными поверх UDP, что позволяет атакующему вполне легальными средствами захватить большое количество портов, просто посылая пакеты соответствующим службам и ожидая от них ответа. Вот как раз таки, при генерации ответа и происходит "заем" порта, освобождаемого только после завершения отправки исходящего пакета. Даже если на сервере нет никаких других UDP служб, атакующий просто обрушивает на DNS шквал легальных запросов, опустошая пул доступных портов, что в конечном счете может привести к отказу в обслуживании. Раньше такой проблемы просто не возникало, поскольку один и тот же порт использовался для всех DNS-ответов, а теперь же, с рандомизацией, DNS может брать только свободные порты, количество которых тает буквально на глазах и псевдослучайная функция назначения порта превращается во вполне предсказуемую.

От 12ти обозначенных бит не остается и следа! А потому рандомизацией портов можно смело пренебречь, сфокусировавшись на предсказании TXID.

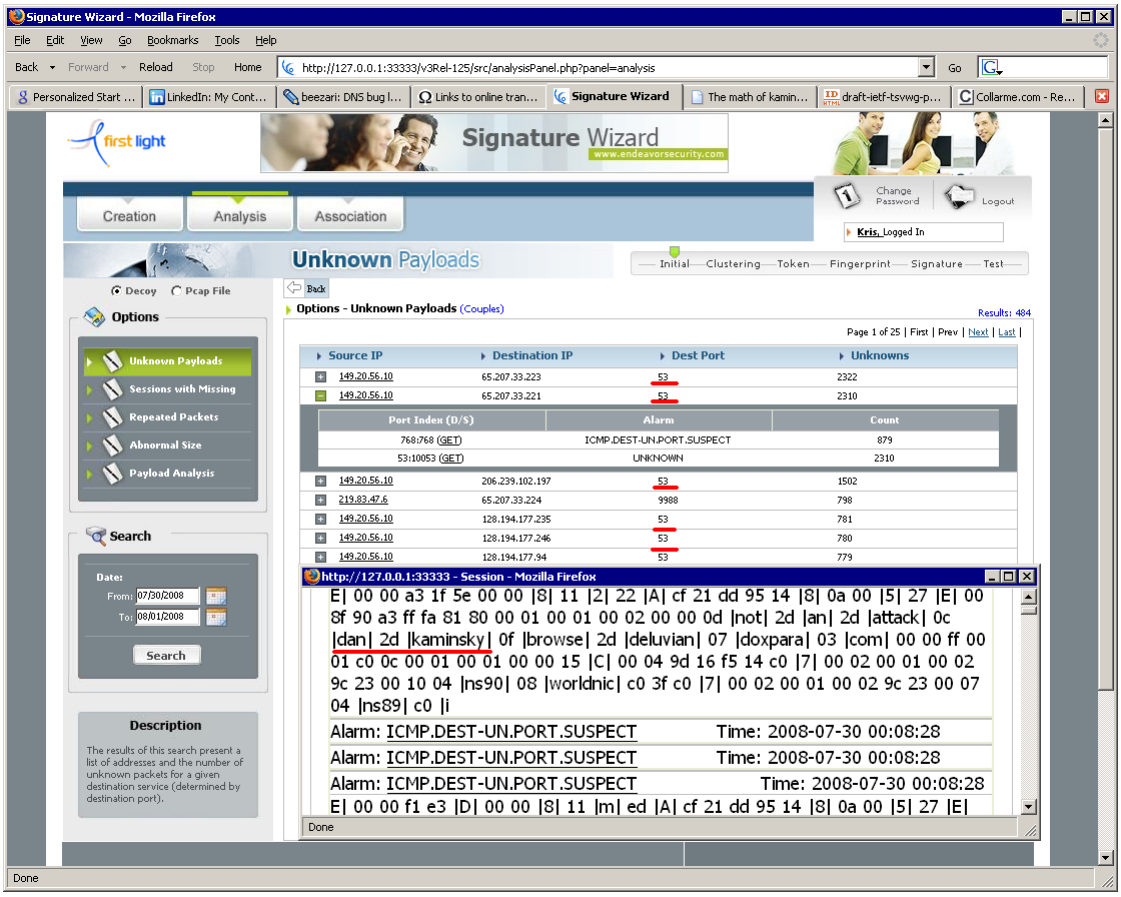

Рисунок 4 мыщъх сидит в своей норе и поглощая одни сосиска за другой, лениво шарит лапами по клаве, наблюдая как Каминский атакует один DNS сервер за другим (а еще девочку из себя строит. ага! и даже не стесняется! ну откуда же ему знать, что в сети везде понатыканы датчики, сенсоры и что сервера, которые он "атакует" это honey-pot'ы)

гадание на кофейной гуще с хронометром в руках

В идеале, для выбора TXID следует использовать абсолютно криптостойкую функцию, генерирующую настоящий белый шум, однако, без привлечения специального оборудования осуществить подобную затею невозможно, не говоря уже о том, что к выбору TXID предъявляются достаточно жесткие требования — они не должны повторяться на коротком временном участке, иначе возникнет путаница чей это пакет и кому он адресован.

PowerDNS и MS DNS используют достаточно качественные функции для генерации TXID и атака на них – тема отдельного разговора (вообще-то, PowerDNS и сам не знает, что использует, читая псевдослучайные числа из псевдоустойства /dev/urandom, которая может вообще выдавать что угодно, а DJBDNS использует системную дату). Остальные же системы используют довольно простые и вполне предсказуемые алгоритмы — зная предыдущие члены псевдослучайной последовательности, мы можем с высокой степенью вероятности вычислить следующие (вероятность тем выше, чем больше у нас членов).

В случае DNS-сервера никаких проблем у атакующего не возникает — он просто посылает ему легитимные запросы, получая ответы с TXID в заголовке. Зная алгоритм, используемый для рандомизации (а он известен, с точностью до системы, версию которой определить не так уж и сложно), хакеру остается всего лишь… гм, даже без всякой математики метод перебора очень даже рулит и тривиальный brute-force (осуществляемый на локальной машине, _без_ обращения к серверу) находит возможные варианты быстрее, чем хакер пьет кофе (чай, пиво, батон с колбасой).

С рабочими станциями ситуация обстоит чуть-чуть сложнее. Ведь им DNS-запросы не пошлешь. И DNS-ответов не словишь. Нам известен алгоритм, используемый для генерации, TXID, известно даже, что для его "затравки" используется системный таймер, но вот значение самого таймера… Прочитать его удаленно?! Да без проблем! При желании можно обойдись

даже без Java-скриптов. Ведь тот же самый таймер используется не только в UDP, но и в TCP (для генерации начального номера последовательности), а потому, если машина способна отправлять хоть какие пакеты во внешнюю сеть — значение таймера восстанавливается без труда, а по нему уже вычисляется "затравка".

Главным образом это относится к Free/OpenBSD, BIND 9 и DJNDNS, использующих некриптостойкие функции. PowerDNS и MS DNS стоят особняком, однако, поскольку они так же подвержены захвату UDP-портов, то 16-битный TXID пусть он хоть сто раз криптостойкий и ни разу не предсказуемый — плохое средство защиты от хакеров.

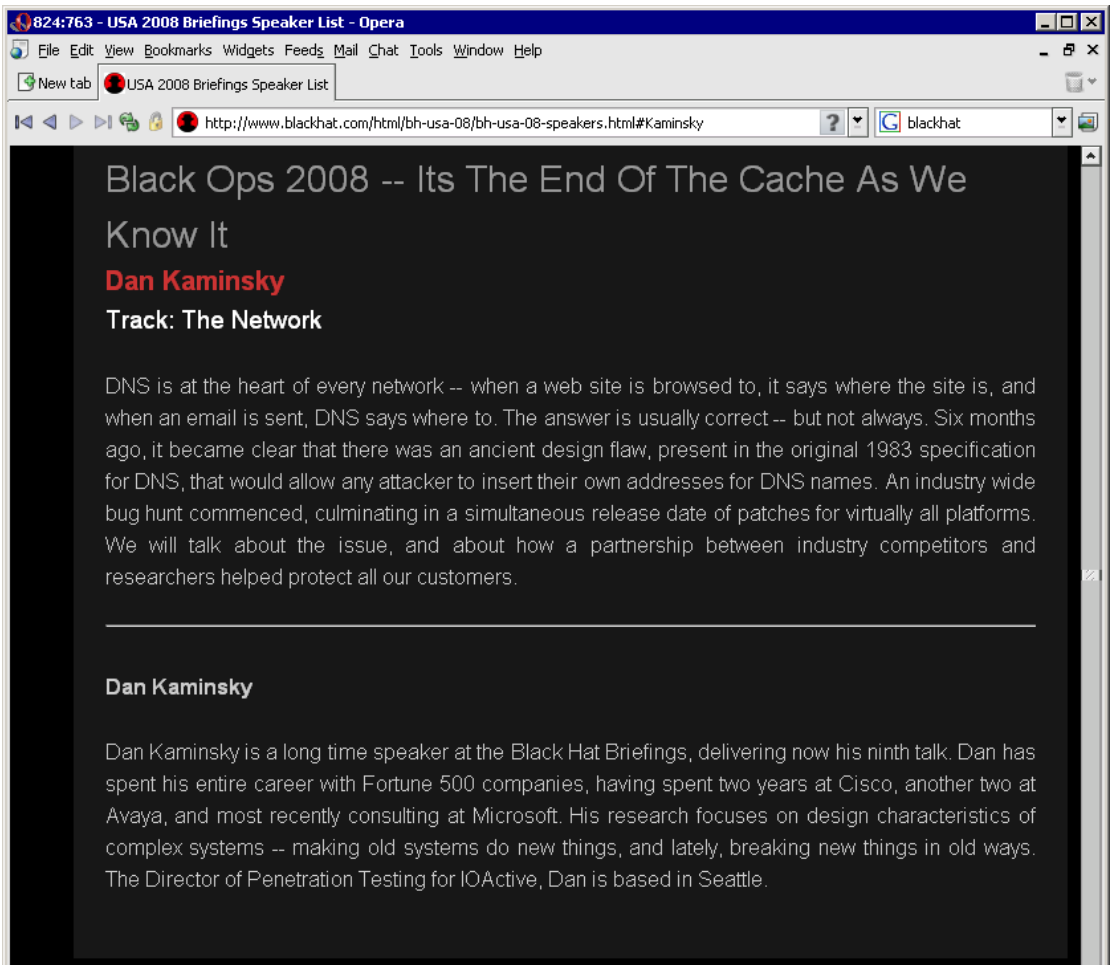

Рисунок 5 Каминский на BlackHat'е

как от этого защищаются

Установка свежих заплаток _однозначно_ не решает проблему и угроза атаки слишком велика, чтобы позволить себе ее игнорировать. Мелкие ISP и офисные сервера достаточно легко перевести на DNS over TCP, существующий еще черт знает когда и, в отличии от классического DNS over UDP, практически не подверженный атакам данного типа. А еще есть DNSSEC, название которого говорит само за себя.

Однако, все эти решения работают лишь на узлах с небольшой нагрузкой. В промышленных масштабах такие варианты даже не обсуждаются. И что остается?! Самое простое — использовать PowerDNS. Он действительно намного более надежен, да и работает побыстрее стандартного BIND 9.

Так же не помешает установить качественную IDS/IDP. Как она работает? Зависит от реализации. Например, садится на интерфейс и ловит все входящие/исходящие DNS пакеты и если обнаруживает достаточно большое количество входящих DNS ответов, которым не соответствовали DNS-запросы — сразу же поднимает тревогу. Слабость такого решения в том, что "левые" пакеты сыплются и без всякой атаки, а количество подложных DNS-ответов при

целевой атаке много ниже порога чувствительности сенсора. Именно так подавляющее большинство IDS/IPS и работает.

Более сложные системы защиты парсят трафик на сетевом уровне, "выдергивают" оттуда DNS-ответы и обращаясь к корневым root-серверам через TCP, определяют достоверность предоставленной информации. Решение, конечно, надежное, но… пропускная способность находится на уровне трубки от ниппеля.

Мыщъх совместно с Алиской (замечательной девушкой из Endeavor Security) разработал скоростной потоковый алгоритм, реализуемый на основе чистого сигнатурного анализа проходящего трафика. Руководящая идея заключается в том, что нормальный DNS-сервер не отвечает дважды в течении короткого времени, поскольку первый ответ будет скэширован и второго запроса просто не последует. А вот хакер, даже располагающий определенной информацией о TXID/SP# вынужден посылать намного больше одного DNS-ответа, содержащий тот же самый отрезовлеенный IP — явный симптом атаки. Подписки Endeavor'a получат обновленные сигнатуры на общих основаниях, остальные же — могут поставить Snort, написав сигнатуры самостоятельно.

>>> врезка установка "заплатки" на BIND 9

```
$vi /etc/bind/named.conf
options {
        pid-file "/var/run/bind/run/named.pid";
        directory "/etc/bind";
        auth-nxdomain no;
        allow-recursion { none; };
        dnssec-enable yes;
        /*
         * If there is a firewall between you and nameservers you want
         * to talk to, you might need to uncomment the query-source
         * directive below.  Previous versions of BIND always asked
         * questions using port 53, but BIND 8.1 uses an unprivileged
         * port by default.
         */
        // query-source address * port 53;
};
$ /etc/init.d/bind9 restart
```

Листинг 1 установка "заплатки" для BIND 9, осуществляемая добавлением пары слешей в файл /etc/bind/named.conf

заключение

Сейчас, когда пишутся эти строки, разработчики DNS-серверов и операционных систем совместно с компанией Endeavor Security, разрабатывают стратегический план ликвидации дыр, обнаруженных мыщъхем, и к моменту выхода журнала из печати, лекарство (в виде очередной порции свежих заплаток) уже появится в аптеках. Но не стоит отчаиваться (или обольщаться — в зависимости от вашей ориентации). Это не первая и далеко не последняя дыра в DNS. Так что время покажет.

Рисунок 6 кто кого поимел

>>> врезка ссылки по теме

- **все о Каминском:**
  - блог Каминского с DNS чекером, предназначенным на проверки серверов на уязвимость (блог на английском, чекер, понятное дело, на машинном): http://www.doxpara.com;
  - одно из многочисленных зеркал, содержащее копию утекшей информации, разъясняющий суть уязвимости, переоткрытой Дэном (на английском языке): http://beezari.livejournal.com/141796.html;
  - поисковый запрос для старика гугла, выдающий ссылки на остальные зеркала: http://www.google.com/search?hl=en&q=Poisoning+CXOPQ.VICTIM.COM+is+not+super+valuable+to+Mallory;
  - статья, в которой Каминский рассказывает о той же самой дыре в DNS, обнаруженной им еще в ноябре 2005, да и то не им (на английском языке): www.eff.org/deeplinks/2005/11/kaminsky-rootkit-causing-widespread-infection;
  - сводная информация по дыре в DNS на OpenNet (на русском языке): http://www.opennet.ru/opennews/art.shtml?num=16872;
- **exploit'ы:**
  - пара exploit'ов от metasploit'а, работают медленно (без предсказания TXID/SP#), но все-таки работают, особенно против непатченных систем (на языке Руби): metasploit.com/dev/trac/browser/framework3/trunk/modules/auxiliary/spoof/dns/;
  - блог на metasploit'е, поясняющий как работают exploit'ы (на английском языке): http://metasploit.com/blog/;
- **заплатки для разных систем**:

- фикс для BIND9, вращающегося под FreeBSD (на языке Си): http://security.freebsd.org/patches/SA-08:06/bind63.patch;
- обновленная версия DNS-сервера DJBDNS (на языке Си в архиве): http://cr.yp.to/djbdns/djbdns-1.05.tar.gz;
- заверения разработчика PowerDNS, что никакие дыры ему вообще не страшны: http://mailman.powerdns.com/pipermail/pdns-users/2008-July/005536.html;
- непредвзятое мнение комитета CERT по вопросам PowerDNS (на английском): http://www.kb.cert.org/vuls/id/CRDY-7FFQZ6;
- инструкция как пропадчить BIND 9 от атаки Каминского в редакторе "vi": www.howtoforge.com/how-to-patch-bind-to-avoid-cache-poisoning-debian-etch;
- последствия наложения заплаток (на английском языке): http://www.securitytracker.com/archives/cause/31.html;

- **бюллетени безопасности**:
  - BIND 9 DNS Cache Poisoning: www.securiteam.com/securitynews/5VP0L0UM0A.html;
  - Multiple DNS implementations vulnerable to cache poisoning http://www.kb.cert.org/vuls/id/800113;
  - BIND Vulnerabilities and Solutions: http://www.openbsd.org/advisories/res_random.txt http://www.undeadly.org/cgi?action=article&sid=20070725193920&pid=15;
  - наиболее реалистичная математическая атака времени, требующегося для успешного взлома пропатченного DNS-сервера (на английском языке): http://www.ops.ietf.org/lists/namedroppers/namedroppers.2008/msg01194.html;
- **теоретические осмысления проблемы безопасности DNS:**
  - FAQ по раномизации (на английском языке): http://www.isc.org/index.pl?/sw/bind/index.php;
  - что представляет собой рандомизация в натуре (на английском языке): http://tools.ietf.org/html/draft-vixie-dnsext-dns0x20-00; http://tools.ietf.org/html/draft-ietf-tsvwg-port-randomization-01;
- **атака Каминского, описанная задолго до Каминского**:
  - "Joe Stewart Birthday Attack", датируемая ноябрем 2002 (на английском языке): http://archive.cert.uni-stuttgart.de/bugtraq/2003/04/msg00311.html;
  - обстоятельное описание атаки на DNS, июнь 2007 (на английском языке): http://www.trusteer.com/bind9dns;
- **следом за один армагедоном придет другой**:
  - просто шутка мыщъхиных друзей из южной афики, где белые выжили всех негров и стали хакерствовать, огранизовав интернет-страну (на английском): http://www.sensepost.com/blog/2343.html;