Крис Касперски

ЭЛЕКТРОННАЯ ПОЧТА И БЕЗОПАСНОСТЬ DNS

kk@sendmail.ru

Крис Касперски
kk@sendmail.ru

Электронная почта и безопасность DNS

С развитием коммуникационных средств появилась возможность создания "виртуальных офисов", неотъемлемую часть которых составляет электронная почта. Корпоративная переписка представляет несомненный интерес для конкурентов и возникает естественный вопрос: "а как обстоят дела с безопасностью?".

Настоящая статья рассказывает о механизмах некоторых атак, использующихся злоумышленниками для реализации угроз перехвата почты, нарушения работоспособности рабочей станции получателя и проникновения в почтовый ящик получателя. Возможность фальсификации адреса отправителя ввиду своей тривиальности здесь не рассматривается.

Перехват почты

Алиса пытается отправить Бобу письмо. Его стремится похитить злоумышленник, называющий себя Евой. Техника, используемая Евой для перехвата, зависит от взаимного расположения жертв и атакующего. Ниже будут рассмотрены следующие комбинации:

а) Ева находится в одном сегменте локальной Ethernet-сети либо с Алисой, либо с Бобом, либо с их почтовым сервером;

б) Алиса и Боб находятся в одной локальной сети, доступной Еве через Интернет;

в) Алиса, Боб и Ева находятся в различных подсетях, подключенных к Интернет.

Перехват, основанный на широковещательной природе Ethernet (Ева находится в одном сегменте локальной Ethernet-сети либо с Алисой, либо с Бобом, либо с их почтовым сервером). Уязвимость локальных Ethernet-сетей объясняется тем, что они функционирует по принципу **множественного доступа с обнаружением несущей** (сокращенно CSMA/CD), все устройства взаимодействуют в одной среде и передачу одного устройства принимают все остальные. Образно такую схему можно уподобить телеконференции, даже если обратится к некому конкретному лицу, это сообщение получит каждый подписчик конференции. Разумеется, передача секретных данных по такому каналу невозможна, если, конечно, предварительно не зашифровать их.

Любой клиент Ethernet-сети может получить доступ ко всей информации, передаваемой по данному сегменту. Пассивный характер атаки не позволяет выявить ни факт перехвата конфиденциальной информации, ни (тем более) установить на каком узле расположен атакующий.

Единственная проблема, стоящая перед Евой – как проникнуть в локальную сеть, да еще и оказаться в одном сегменте с жертвой? Чаще всего атакующий находится вне локальной сети, в лучшем случае имея к ней доступ через Инетрнет.

Теоретически она может попробовать прорваться в здание компании, навешав лапшу на оттопыренные уши охранника, незаметно присесть за работающий терминал и,.. впрочем, при грамотном охраннике у нее вряд ли у нее что-то получится.

Перехват почты, основанный на уязвимости почтового сервера (Алиса и Боб находятся в одной локальной сети, доступной Еве через Интернет). Для атаки почтового сервера Ева может использовать ошибки его реализации, например, у некоторых из них в файле "/etc/aliases" присутствует строка приблизительного следующего содержания: "decode: |/usr/bin/uudecode", автоматически запускающая программу "uudecode", предназначенную для распаковки UUE-сообщений.

Многие расшифровщики помещают раскодированный текст в файл, указанный в UUE-заголовке, допуская его перезапись без запроса подтверждения от пользователя. Если почтовый сервер обладает наивысшими привилегиями (а так часто и бывает), программа uudecode унаследует их при запуске, предоставляя злоумышленнику возможность перезаписи ***любого*** файла в системе.

Например, Ева может внести в файл *"/.forward"* строку вида "\root, root@somehost.org, *eva@hotmail.ru*", организовав дублирование почты администратора на свой собственный адрес.

Точно такую операцию можно проделать и со всеми остальными пользователями системы. достаточно лишь изменить файлы ".forward", находящиеся в их домашних каталогах. Поскольку, пользователи могут самостоятельно редактировать свои конфигурационные файлы, администратор не может знать кто внес эти изменения легальный пользователь или злоумышленник.

Другая уязвимость заключается в возможности выполнить код на удаленном сервере, используя неявную поддержку конвейера в полях "MAIL FROM" и "RCPT TO" у некоторых реализаций почтового сервера..

Порой даже сами разработчики не подозревают об этом, поскольку такая возможность не всегда очевидна. Например, команда "open" языка Perl может не только открывать файл, но и ***запускать*** его, если в имени присутствует символ "|", обозначающий вызов конвейера.

Попытки самостоятельной интерпретации передаваемых функции параметров встречаются достаточно часто. Особенно они характерны для UNIX-систем, в которых конвейер является популярным средством межпроцессорного взаимодействия. Однако разработчики, использующие в свой программе библиотеки сторонних поставщиков, не всегда проверяют как поведет себя та или иная функция, обнаружив символ конвейера, особенно если об этом ничего не написано в документации. Похожая проблема возникает и при разработке совместных проектов – один разработчик не может знать всех тонкостей функционирования модулей другого разработчика, а в результате такой нестыковки полученная программа может неявным образом поддерживать конвейер!

Точно так многие почтовые серверы оказываются уязвимы перед тривиальным срывом стека. Например, в феврале 2000 года, в SMTP сервере MMDF версии 2.44a-B4, работающего под управлением SCO-UNIX, обнаружилась ошибка переполнением буфера в полях "MAIL FROM" и "RPPT TO", позволяющая злоумышленнику выполнить любой код под привилегией root.

А несколькими месяцами ранее – в ноябре 1999 года, ошибка переполнения была обнаружена в POP3\SMTP сервере ZetaMail версии 2.1. Пароль, передаваемый командой "PASS", помещался в буфер фиксированного размера, и слишком длинная строка вылезала за его границы.

В том же ноябре 1999 появилось сообщение о наличие аналогичной уязвимости в версиях 3.2x SMTP-шлюза Interscan VirusWall, работающего под управлением Windows NT. На этот раз переполнение буфера происходило во время приветствия сервера командой "HELO" (если это приветствие оказывалось через чур многословным).

Примерно в то же самое время обнаружилась уязвимость IMAIL POP3-сервера. Версии 5.07, 5.05 и 5.06 не контролировали длину имени пользователя, передаваемую командой "USER". Такую же участь разделил Avirt Mail Server (версии 3.3a, 3.5). Сообщение о возможности переполнении буфера командой "PASS" появилось в начале ноября все того же 1999 года.

Универсальной защиты от ошибок реализации нет, все что можно посоветовать администратору – оперативно устанавливать свежие "заплатки" и обновления.

На FireWall надеяться бессмысленно – в данной ситуации он не спасет, поскольку допускает прием сообщений почтовым сервером, приходящих глобальной сети. Если же это запретить, прием писем от нелокальных респондентов станет невозможным, что в подавляющем большинстве случаев неприемлемо.

Перехват, основанный на уязвимости DNS серверов (Алиса, Боб и Ева находятся в различных подсетях, подключенных к Интернет). Не все получатели находятся в рамках одной локальной сети, и в некоторых случаях приходится связываться с ними через Интернет. Для установки соединения с удаленным узлом необходимо знать его IP адрес, но в e-mail-адресах обычно используются не IP-адреса, а доменные имена почтовых серверов абонентов. Для того, чтобы узнать адрес узла по его доменному имени обычно обращаются к службе DNS – распределенной базе данных, чаще всего функционирующей на базе протокола UDP.

Такой выбор объясняется тем, что протокол UDP работает без установки соединения, и, поэтому, имеет лучшую производительность по сравнению с TCP. Расплатой за скорость становится отсутствие контроля успешности доставки пакетов, невозможность определения подлинности отправителя, неспособность противостоять сбоям и выявлять дублирующиеся пакеты.

Узел, получивший UDP пакет, не может достоверно установить его истинного отправителя, следовательно, у злоумышленника существует возможность послать сообщение от чужого имении.

Для перехвата корреспонденции Еве достаточно послать серверу Алисы фальсифицированное послание, замаскированное под ответ настоящего DNS-сервера. Если тот не распознает обмана, он установит соединение не с почтовым ящиком Боба, а с узлом, указанным Евой, и передаст ему сообщение.

Если Алиса пренебрегла шифровкой и послала сообщение в открытом виде, Ева без труда сможет ознакомиться с его содержимым (но даже если письмо зашифровано, Ева смогла досадить Алисе по крайней мере тем, что Боб не получил ее послания и никогда теперь его не получит).

При оценке степени возможной угрозы необходимо учитывать, что для подделки ответа DNS злоумышленник должен подобрать верное значение порта-отправителя и уникального идентификатора, присутствующего в ответе DNS.

Илья Медведовский [1] сообщает, что "*начальное значение "порта отправителя" в UDP-пакете ≥ 1023 и увеличивается с каждым переданным DNS-запросом*", а "*значение идентификатора (ID) DNS-запроса устанавливается следующим образом. В случае передачи DNS-запроса с хоста оно зависит от конкретного сетевого приложения, вырабатывающего DNS-запрос. Эксперименты показали, что если запрос посылается из оболочки командного интерпретатора (SHELL) операционных систем Linux и Windows 95 (например, ftp nic.funet.fi), то это значение всегда равняется единице. Если же DNS-запрос передается из Netscape Navigator или его посылает непосредственно DNS-сервер, то с каждым новым запросом сам браузер или сервер увеличивает значение идентификатора на единицу*".

Во втором издании книги [2], выпущенной спустя два года, этот фрагмент не претерпел никаких изменений, а, ситуация, тем временем, несколько изменилась – в современных реализациях разработчики проявили тенденцию к генерации непредсказуемых идентификаторов, предотвращающих возможность атаки. А порт отправителя вовсе не всегда монотонно увеличивается от запроса к запросу, после освобождения сокета тот же порт может использоваться повторно.

Впрочем, к радости Евы, в сети до сих пор остаются пользователи, пользующиеся устаревшим программным обеспечением, поэтому, у нее есть вполне реальная возможность отгадать значения порта отправителя и идентификатора, но достаточно ли этого для успешности атаки? Нет!

Ева должна не только правдоподобно фальсифицировать пакет, но и доставить его жертве раньше, чем та получит ответ от настоящего DNS-сервера. Проще всего этого добиться на какое-то время блокировав DNS-сервер Алисы, к примеру, направив на него шквал DNS-запросов, обработка которых должна занять все свободное время сервера или же попытаться "завесить" его каким-нибудь другим способом (а такие способы в арсенале злоумышленников наличествуют в изобилии – не стоит даже на них и останавливаться).

Таким образом, Ева должна знать не только адреса Алисы и Боба, но и адрес DNS-сервера отправителя. А вот это уже представляет определенную проблему! В простейшем случае, когда Алиса использует DNS-сервер, предоставленный провайдером, выяснить его адрес можно без труда: достаточно спросить об этом у провайдера, которого, в свою очередь, можно определить по IP-адресу Алисы.

Сложнее атаковать корпоративного пользователя, использующего собственный DNS-сервер, для надежности закрытый извне FireWall-ом. Создать надежную защиту, проникнуть сквозь которую Ева сможет разве что при помощи динамита, теоретически вполне возможно. Первой головной болью атакующего станет определение адреса DNS-сервера (а точнее, DNS-серверов, поскольку их может быть и больше одного). Сканирование IP-адресов на предмет открытого 53 порта предотвратимо тем же FireWall-ом, блокирующим прием пакетов с внешних IP-адресов.

Впрочем, с этой проблемой атакующий может справится ("может" – следует понимать двояко – т.е. может справиться, а может и не справиться) – любой FireWall, не отягощенный искусственным интеллектом, действует по однажды заложенной в него схеме, а тем временем появляются новые все более и более изощренные алгоритмы сканирования, которые ему не так-то просто распознать!

Однако если администратор сети внедрит множество фальшивых DNS-серверов, "слушающих" 53-порт, но ничего сверх этого не делающих, "ослепленная" Ева будет вынуждена либо уложить всех их до единого (при этом вовсе не будучи уверена, что среди них присутствует и настоящий DNS), либо атаковать жертву каким-нибудь другим способом.

Даже если, Ева каким-то образом обойдет FireWall, она столкнется с проблемой неизвестности точного времени отправки письма. Ведь не может же она бесконечно долго забрасывать Алису пакетами, ибо такое поведение слишком быстро обратит на себя внимание и демаскирует атакующего. Если Алиса отвечает Бобу в абсолютно непредсказуемое время и у Евы нет никаких, даже косвенных зацепок, позволяющих хотя бы приблизительно определить время пересылки письма, ей придется изменить свою тактику и атаковать сам DNS-сервер.

Такая возможность базируется на незащищенности взаимодействия DNS-серверов друг с другом. Для этой цели используется все тот же уязвимый протокол UDP, не позволяющий установить подлинность отправителя. В общих чертах работа типичного DNS-сервера выглядит так: получив запрос клиента, сервер просматривает содержимое своего локального кэша и в случае отрицательного результата перенаправляет запрос на DNS-сервер более высокого уровня. Для ускорения работы его ответ помещается в кэш и в дальнейшем необходимость прибегать к посторонней помощи отпадает.

Если Алиса ранее уже отправляла корреспонденцию Бобу, адрес узла получателя давным-давно находится в кэш DNS-сервера, и, на первый взгляд, у Евы нет никакой возможности подменить его своим. Но это предположение не совсем верно.

Ева может воздействовать на содержимое кэша штатными средствами: посылкой множеством разнородных запросов, атакующий рано или поздно вытеснит из кэша все старые записи, в том числе и адрес узла Боба (при этом важно отметить тот факт, что все запрошенные доменные имена должны существовать в природе, отсутствовать в кэше, и, наконец, размер кэша не должен быть слишком велик, иначе будет очень трудно, а то и невозможно, добиться его заполнения).

На втором этапе атаки Ева самостоятельно пошлет DNS-серверу запрос, содержащий имя узла получателя, и тут же обрушит на сервер шквал подложных ответов. Если хотя бы один из них будет принят за подлинный, в кэш попадет фальшивый адрес! Здесь, правда, присутствует одно допущение – молчаливо предполагается, что при каждом запросе DNS-запросе Алиса получает ответ от самого DNS-сервера. На практике же, по соображениям производительности, предыдущие запросы могут кэшироваться как узлом самой Алисы, так и одним из промежуточных хостов, соединяющих ее с DNS-сервером. Однако, старая информация не может оставаться в кэше бесконечно долго, т.к. это привело бы к очевидным проблемам, не позволяя отслеживать изменения адресов доменных имен (такое хоть и редко, но происходит), поэтому, хранящиеся в кэше данные периодически "освежаются". Определенными действиями Ева может ускорить развязку, однако, возможно, этого и не потребуется, т.к. в большинстве случаев, интервал достоверности информации, содержащийся в кэше невелик, и если Ева не шибко спешит, ей нет никакой нужны что-либо предпринимать.

Можно ли оградить себя от таких атак? С клиентской стороны однозначно – ***нет.*** Но вот администратор DNS-сервера теоретически может кое-что предпринять для повышения защищенности своих клиентов. Например, некоторые руководства по безопасности (не будем показывать пальцем) рекомендуют отказаться от использования UDP протокола и перейти на TCP. Протокол TCP выгодно отличается тем, что работает с установкой соединения и позволяет идентифицировать отправителей. Падение производительности компенсируется усиленной защищенностью. Однако...

Первой головной болью, с которой столкнется администратор при переходе на TCP, окажется гробовое молчание документации, ничего не говорящей о том, как такой трюк проделать (правда, есть куча прекрасной литературы по этой тематике). Например, в описании демона named, работающего под управлением Linux, возможность выбора TCP-протокола не упоминается вообще! Но даже если администратор сумеет связаться с разработчиками и ценой бессонных ночей научит свой сервер "разговаривать" на TCP, это не предотвратит угрозу атаки.

Во-первых, TCP-протокол вовсе не такой защищенный каким кажется. Некоторые программные реализации TCP/IP генерируют предсказуемые идентификаторы, позволяя злоумышленнику посылать пакеты от чужого имени. Именно это обстоятельство использовал небезызвестный Кевин Митник в своей атаке против Цутому Шимомуры (а сам Цутому помниться сказал "*Проблема не в Кевине, проблема в том, что большинство систем действительно плохо защищено; то, что делал Митник, остается осуществимым и сейчас*"). Впервые на эту уязвимость обратил внимание еще Моррис – старший, опубликовав в февральском номере журнала Bell Labs за 1985 году (за десятилетие до Митника!) подробный технический отчет, посвященный указанной проблеме[3], но даже сегодня, спустя пятнадцать лет после публикации, эта проблема остается актуальной.

Во-вторых, популярные операционные системы содержат одну малоизвестную тонкость (о том, что не все из них умеют формировать TCP-запросы к DNS приходится вообще молчать): посылая DNS-запрос, они не знают на каком именно протоколе работает сервер и предполагают, что по умолчанию выбран UDP (во всяком случае, так себя ведут LINUX и Windows). Поэтому, в указанных случаях се-

анс обмена начинается с посылки UDP-пакта. При нормальном ходе вещей сервер может дать клиенту указание перейти на TCP-протокол, но если злоумышленник сумеет послать жертве подложный ответ раньше, чем это успеет сделать настоящий сервер, клиент окажется введенным в заблуждение со всеми вытекающими отсюда последствиями.

Тем не менее, несмотря на указанную уязвимость DNS, реализовать такую атаку на практике чрезвычайно затруднительно - слишком много препятствий придется преодолеть взломщику прежде, чем он достигнет цели. Но невысокая вероятность успешной атаки – еще не защищенность! В критических случаях для обеспечения собственной безопасности при соединении с узлом можно использовать его IP-адрес, а не доменное имя.

Если Алиса пошлет сообщение по адресу *bob@123.456.789.1* никакие манипуляции с DNS сервером не позволят Еве перехватить такое письмо, поскольку, в этом случае обращений к DNS не происходит! С неудобствами же запоминания ряда бессмысленных цифр вполне можно смириться, особенно если вспомнить, что популярные почтовые клиенты поддерживают адресные книги и автоматически подставляют адрес получателя по его имени.

Однако, такой подход не лишен недостатков. Во-первых, отправитель скорее всего не знает IP адреса узла получателя и вынужден обращаться к DNS, чтобы это узнать. А DNS-сервер, как было показано выше, мог быть атакован Евой, навязавший ему ложный IP адрес доменного имени почтового сервера получателя.

Во-вторых, с одним доменным именем может быть связано множество IP-адресов. Это позволяет равномерно распределить нагрузку между несколькими серверами и значительно увеличивает надежность системы: если один из серверов откажет, обработку запросов продолжат все остальные. В принципе отправитель может выбрать какой-то один, понравившейся ему IP-адрес, но это приведет к существенному снижению надежности связи, что в ряде случаев неприемлемо.

Таким образом, лучшее что можно предпринять – защититься FireWall-ом, при необходимости перевести DNS-сервер на TCP протокол, запретить DNS-клиенту принимать UDP пакеты, установить максимальный размер DNS-кэша, предотвращая возможность его переполнения, не использовать реализаций протокола TCP/IP, генерирующих предсказуемые идентификаторы, и, при возможности, попросить выполнить эти требования администратора DNS-сервера более высокого уровня.

Перехват, основанный на уязвимость транзитных узлов. В идеальном случае сервер отправителя устанавливает прямое соединение с почтовым ящиком получателя и опускает в него конверт. Но такая схема не всегда соответствует реальной действительности. Огромное количество ежесекундно обрабатываемой корреспонденции заставило разбить процесс доставки сообщений на множество этапов и создать специальные узкоспециализированные узлы, занимающиеся исключительно транспортировкой почты от одного сервера к другому.

Такая схема позволила оптимизировать сетевой трафик, но породила целый ряд проблем: письма стали теряться, "застревать", искажаться двойной перекодировкой и т.д. В довершение ко всему многие промежуточные узлы слабо защищены и представляют легкую мишень для взлома.

Для связи друг с другом промежуточные узлы часто используют доменные имена, поэтому у злоумышленника существует возможность перехватить поток корреспонденции направленным штурмом ложных DNS-ответов на один из промежуточных узлов.

Такая атака требует: а) знания маршрута письма; б) знания адреса DNS-сервера, которому промежуточный узел направляет запрос.

Выяснить маршрут письма можно несколькими способами. Если сервер Алисы допускает отправку корреспонденции без аутентификации клиента (такие сервера, хотя и редко, но все же встречаются), Ева сможет соединиться с ним и послать письмо сама себе. Все промежуточные узлы оставят свои адреса в его заголовке. Не факт, что к Бобу корреспонденция ходит тем же самым путем, но адреса первых одного-двух узлов обычно идентичны независимо от конечного получателя.

Если же эта операция не удастся (а она, скорее всего, и не удастся), Ева может вступить в переписку с Алисой или любым другим пользователем, отправляющим письма с ее сервера. Еве достаточно получить в свои руки хотя бы один конверт, чтобы восстановить траекторию путешествия письма.

Адрес DNS-сервера можно выяснить тривиальным сканированием. Большинство крупных транзитных узлов имеют свои собственные DNS-сервера, расположенные с ними в одной и той же подсети. Конечно, администратор атакуемого узла может поставить FireWall, препятствующий определению адреса его DNS-сервера, однако, не все промежуточные узлы снабжены такой защитой.

Для обеспечения конфиденциальности переписки Алисе надлежит либо полностью отказаться от транзитной пересылки, либо доверять ее только надежным узлам. И то, и другое можно реализовать клиентскими средствами, не обращаясь за помощью к администратору. Существуют специальные программы для, так называемой, ***директивной рассылки***. Они не нуждаются в серверах исходящей почты и при отправке сообщения устанавливают соединение непосредственно с почтовым ящиком получателя. Если ни одной такой программки под рукой нет, на худой конец сойдет и штатный почтовый клиент. Достаточно в поле адреса сервера исходящей почты прописать адрес почтового ящика получателя сообщения.

Для переписки с одним лишь Бобом предложенная выше схема вполне приемлема, но если Алиса переписывается с огромным количеством респондентов, возникают определенные сложности. Например, ящики некоторых получателей на момент отправки сообщения могут не работать (или работать нестабильно) и попытки отправки корреспонденции придется бесконечно повторять вновь и вновь. Как бы возложить эту задачу на чужие плечи и при этом не ухудшить безопасность (вариант нанять секретаря здесь не рассматривается)?

Изучая спецификацию SMTP протокола, ведающего пересылкой корреспонденции, можно обнаружить давно забытый рудимент, оставшихся (и практически исчезнувший на сегодняшний день) с тех незапамятных времен, когда никакой автоматической маршрутизации еще не существовало, и траекторию путешествия пакетов отправителю приходилась указывать самостоятельно.

В UNIX-системах для этой цели использовался лес "бангов" – восклицательных знаков, перечисляющих все промежуточные узлы один за другим. Разработчики SMTP протокола трансформировали банги в удобочитаемые запятые, отче-

го, полный e-mail адрес получателя стал выглядеть приблизительно так: "<@one, @two:Bob@mail.ru>", где "one" и "two" адреса промежуточных узлов, по цепочке пересылающих почту друг другу. Если ни один промежуточный узел не указан, сервер-отправитель самостоятельно определяет маршрут передачи сообщения. Если прямое соединение между узлами one и two невозможно, письмо либо возвратится назад к отправителю, либо узел one передаст его через посредника, которого выберет самостоятельно. Т. е. при условии соблюдения стандартов гарантируется, что письмо посетит все перечисленные узлы в указанном порядке, но не факт, что оно пройдет ***только*** через перечисленные узлы.

Правильный подбор узлов предполагает, что каждый узел надежно защищен от посягательств злоумышленника, и обеспечивает непосредственную связь с своим "соседом".

Если Алиса успешно справится с подбором транзитных серверов, ей останется решить одну маленькую проблему, а именно: популярные почтовые клиенты откажутся принять такой адрес, считая его "неправильным". Выход из ситуации состоит в использовании специализированных клиентов, которых нетрудно найти в сети (во избежание обвинений в рекламе никакие конкретные ссылки не приводятся).

Нарушение работоспособности рабочей станции получателя

Угроза раскрытия конфиденциальности переписки - не единственная проблема. Ева может послать Алисе особое зловредное сообщение, которое в лучшем случае испортит Алисе настроение, а в худшем похитит с ее компьютера секретную информацию, или даже отформатирует жесткий диск.

В начале октября 2000 года некий злоумышленник, имени которого история не сохранила, послал автору этой стати прелюбопытное письмо, которое по замыслу Евы предназначалось для блокирования работы почтового клиента. В теле сообщения, переданного в HTML формате, присутствовал следующий тривиальный Java-код: "while (1) { alert ("Hello, Kris\nI love your!"); }".

При попытке просмотра письма на экране появлялся модальный диалог до закрытия которого весь интерфейс почтового приложения блокировался. А поскольку диалог выдавался в бесконечном цикле, закрыть его никакой возможности не было (т.е. закрыть-то возможность была, но он тут же появлялся вновь). После перезагрузки (или снятия процесса по Alt-Ctrl-Del) и повторного запуска почтового клиента все повторялось вновь. Суть заключалась в том, что для удаления письма было необходимо перейти в папку "Входящие", отчего в области предварительного просмотра отображалось последнее полученное сообщение, а вместе с ним запускался зловредный скрипт.

Впрочем, автор статьи ничуть не пострадал, поскольку из соображений безопасности держал Java-машину выключенной, но живо представляет себе, что за слово слетело бы с его губ, окажись Java-машина по небрежности включенной.

Через пару дней пришло другое сообщение, на этот раз открывающее в бесконечном цикле множество окон размером миллион на миллион пикселей, что в очень короткое время привело бы к зависанию Windows 9x, но установленная на компьютере автора операционная система Windows NT такую атаку благополучно пережила.

Подобные "атаки" вызывают ухмылку у профессионалов, но способны поставить в тупик новичков, не имеющих никакого представления о виртуальных Java-машинах и не знающих как выйти из такой ситуации. В результате – потерянное время, нервы, настроение и ворчание побеспокоенного администратора. Но как бы не было велико его желание "вырубить у всех юзеров Яву напрочь", такое решение не всегда допустимо, т.к. простор многих сайтов после этого окажется невозможен.

Подобные шутки противны, но не более того. Куда опаснее скрипты, похищающие секретные файлы с локального диска пользователя. Такие атаки становятся возможны благодаря многочисленным ошибкам в браузерах и виртуальных Java-машинах. Даже последняя на момент написания статьи, пятая версия браузера Internet Explorer, запущенная под управлением Windows 2000, остается небезопасной. Вызов windows.open() в сочетании с функцией location() позволяет выполнить Java-апплет в контексте локального документа, вследствие чего скрипт злоумышленника получает доступ к его содержимому и может передать этот файл в руки Евы.

Поддержка плавающих форм в Internet Explorer 5.01 (и в некоторых других версиях) реализована с ошибкой. Событие "*NavigateComplete2*", извещающие о завершении переселения документа на новое местоположение, открывает Еве доступ к этому документу, даже если он расположен на локальном диске клиента. Код, приведенный ниже, демонстрирует чтение файла `"C:\test.txt"` выводя его содержимое в диалоговом окне:

```
<IFRAME ID="Z"></IFRAME>
<SCRIPT for=Z event="NavigateComplete2(x)">
 alert(x.document.body.innerText);
</SCRIPT>
<SCRIPT>
 Z.navigate("file://c:/test.txt");
</SCRIPT>
```

Но этим поток ошибок не заканчивается. Ева способна встроить в HTML-письмо файл помощи Windows, записанный в формате chm. Эти файлы могут содержать команды вызова исполняемых файлов, причем последние не обязательно должны находится на локальном диске жертвы, и вполне могут располагаться на компьютере злоумышленника, ведь благодаря встроенной в Windows поддержке CIFS (*Common Internet File System*) различия между локальными и удаленными файлами сглаживаются!

Опасность такой атаки заключается в том, что от Алисы не требуется выполнения никаких дополнительных действий – достаточно просмотреть содержимое письма, и код Евы тут же получит управление. На фоне этой угрозы, нашумевший вирус "I LOVE YOUR", и все прочие насекомые, требующие для своего запуска открытия прикрепленного к письму вложения, выглядят детской игрушкой.

Недавно в Outlook Express 5.x была обнаружена серьезная ошибка, позволяющая слишком длинным полем заголовка письма, переполнить буфер и передать управление на код злоумышленника еще на стадии получения письма с сервера, т. е. задолго до того, как его успеют проверить все существующие антивирусы! Нетрудно вообразить себе как это могло бы быть использовано расторопными злоумышленниками (окажись они не такими ленивыми)! Впрочем, такая возможность у них еще есть – ведь не все пользователи заботятся об обновлении своих приложений и уязвимый Outlook Express 5.x до сих пор широко распространен.

В шутку можно сказать: письма, особенно присланные в HTML-формате, лучше вообще не читать, а уж тем более, полученные в пятницу, приходящуюся на тринадцатое число тринадцатого месяца.

Проникновение в почтовый ящик получателя

Еще одной целью атаки Евы может быть почтовый ящик получателя. Если Ева сумеет каким-то образом подобрать к нему пароль, она получит доступ ко всей, хранящейся в нем корреспонденции. Первый залог безопасности Боба – жутко длинный, бессмысленный, иррегулярный пароль, найти который лобовым перебором Ева не сможет даже за всю оставшуюся жизнь. Выбирая такой пароль, Боб должен учитывать, что многие почтовые службы поддерживают опцию "забыли пароль?". В общих чертах суть ее состоит в следующем: регистрируясь в системе, пользователь сообщает некоторую дополнительную информацию о себе, например, дату рожденья любимой крысы своей подруги. Если Боб нечаянно утеряет пароль, он сможет ввести эту информацию и получить новый. Из самых общих соображений следует: контрольная информация должна быть столько же непредсказуема, как и сам пароль, иначе это облегчит Еве проникновение в систему. Например, всевозможных дат в диапазоне между 1950 и 2000 годом существует немногим более десяти тысяч и для их перебора Еве не потребуется много времени, поэтому использование какой-либо даты в качестве контрольной информации неразумно!

Предположим, Боб правильно отнесся к выбору пароля и контрольной информации. Может ли он теперь чувствовать себя в безопасности? Если используемая Бобом почтовая служба не содержит ошибок реализации, позволяющих Еве получать доступ к содержимому чужих ящиков (вообще-то системы без ошибок на сегодняшний день большая редкость, если они вообще есть), то на первый взгляд ему ничто не угрожает.

Между тем, протокол POP3, использующийся для доставки почты с сервера на компьютер клиента, передает имя пользователя и пароль в открытом, незашифрованном виде. Еве достаточно перехватить сетевой трафик и извлечь соответствующие пакеты. Широковещательная среда локальных сетей Ethernet позволит осуществить эту операцию без труда, а межсегментную атаку Ева сможет реализовать "подмятием" DNS.

Вообще-то, в спецификации POP3 протокола можно обнаружить необязательную для реализации команду "APOP", шифрующую пароль перед его отправкой на сервер и не чувствительную к перехвату. Однако ее поддерживают не все почтовые службы и не все почтовые клиенты. Выяснить, как обстоят дела в конкретно взятом случае можно у администратора системы (например, популярный сервер mail.ru поддерживает такой метод аутентификации пользователей).

Выводы

Проанализировав сложившуюся ситуацию, приходится констатировать: гарантировать конфиденциальность электронной переписке на сегодняшний день невозможно. Угроза исходит не только от спецслужб, наподобие СОРМ, но и простых Васей Пупкиных, вооруженных одним лишь модемом, простеньким персональным компьютером и базовыми техническими знаниями. Все атаки, описанные выше, не представляют никакого секрета и хорошо известны как специалистам по безопасности, так и злоумышленникам. Отсутствие громких прецедентов, связанных с хищением почты, не дает повода надевать розовые очки. В отличие от украшения WEB-страничек своим graffiti, перехват корреспонденции происходит незаметно, а жертвам невыгодно разглашать факт свершившейся атаки, особенно если злоумышленникам удалось похитить действительно ценную информацию.

В идеальном случае следовало бы посоветовать отказаться от пересылки почты на удаленные узлы Internet, а если такая операция неизбежна – обязательно шифровать содержимое послания такими утилитами, наподобие PGP.

Ни в коем случае не стоит экономить на специалистах и принимать администратором человека без глубоких знаний устройства сети и должного опыта работы, но и не стоит переоценивать возможности даже самого талантливого "гуру". Администратор – не бог и устранить уязвимости базовых протоколов и систем защиты он не силах.

Строго говоря, любой протокол Internet, основанный на TCP/IP, принципиально уязвим для взлома, а переход на новые, защищенные протоколы, по некоторым причинам затягивается и происходит не так быстро, как хотелось бы.

В то же время, необходимо отметить относительную малочисленность описанных выше атак в связи с трудностями их реализации. Так что, не стоит впадать в панику несмотря на всю свою не безопасность Интернет живет, работает и постепенно улучшает свою защищенность.

Список литературы

1. И. Медведовский, П. Семьянов, В.Платонов "Атака через Internet" НПО Мир и Семья Санкт-Петербург 1997
2. И. Медведовский, П. Семьянов, Д. Леонов "Атака на Internet" ДМК Москва 1999
3. Bell Labs Computer Science Technical Report #117, February 25, 1985

Крис Касперски, независимый эксперт по компьютерной безопасности

С автором можно связаться по адресу **kk@sendmail.ru**