три истории из жизни пен-тестера:

проникновение внутрь охраняемого периметра корпоративных сетей

крис касперски, no-email

если вы считаете, что ваша корпоративная сеть надежна защищена — вы заблуждаетесь. нет такой сети, которую нельзя было бы взломать, не прилагая особых усилий, не располагая специальным оборудованием и не прибегая к явному криминалу (физическому вторжению с огнестрельным оружием). имея за своими плечами многолетний опыт хакерских атак (естественно, осуществляемых с ведома и согласия "жертвы"), автор этой статьи хочет рассказать о наиболее типичных ошибках администраторов, существенно упрощающих процедуру взлома

введение

С технической точки зрения, между хакерами и пен-тестерами нет никакой разницы — и те и другие используют схожие приемы и методики, отработанные не одним поколением взломщиков и подробно описанные в доступной литературе. Пен-тестеры находятся даже в более жестких условиях, поскольку, им приходится играть в открытую, когда администратор предупрежден и вооружен. К тому же, пен-тестер не может позволить себе "угробить" систему и ему приходится буквально танцевать на острие, поскольку малейшая ошибка (даже непредумышленная) зачастую имеет необратимые фатальные последствия.

Ирония в том, что подавляющее большинство администраторы не знают как в _действительности_ действуют хакеры. Знания, почерпнутые из популярных руководств в стиле "как построить защищенную систему", сильно теоритизированы и оторваны от действительности. Не безопасность компьютерных систем — это аксиома, утверждать обратное — означает демонстрировать свою некомпетентность. Хотите конкретных примеров? Пожалуйста!

Рисунок 1 фото автора

история первая — хакерское сафари

Глубокая ночь. Все нормальные люди (и администраторы!) спят крепким сном, но в мире андеграунда самый пик активности. Хакеры гоняются за свежими дырами, о которых еще не знают ни производители программного обеспечения, ни специалисты по безопасности, ни администраторы… А раз не знают, то, следовательно, и не защищаются!

Кто-то ведет целенаправленный поиск дыр, погружаясь в миллионы строк дизассемблерного кода, реконструируя исходный алгоритм программы и пытаясь найти условия при которых она поведет себя совсем не так, как задумалось ее создателями (а, как известно, компьютерная программа делает то, что ей приказали сделать, а не что _хотели_ приказать). Целенаправленный поиск — требует высокой квалификации, полной самоотдачи и огромного вовлечения в исследовательский процесс. С коммерческой точки зрения он нерентабелен. Автор за последние два десятка лет "перемолотил" миллиарды ассемблерных строк, убив на это с полсотни тысяч часов (!) и обнаружив чуть больше дюжины критических дыр в популярных операционных системах (некоторые из которых, кстати говоря, впервые появившись еще в NT 3.x благополучно дожили до Server 2008, оставшись никем незамеченными и в критических ситуациях, когда нужно проникнуть в особо защищенную систему они здорово выручают, правда, в целом мероприятие по самостоятельному поиску дыр можно считать коммерчески проваленным, но это и не важно, поскольку, помимо денег есть такие понятия как "live for" и простой спортивный интерес).

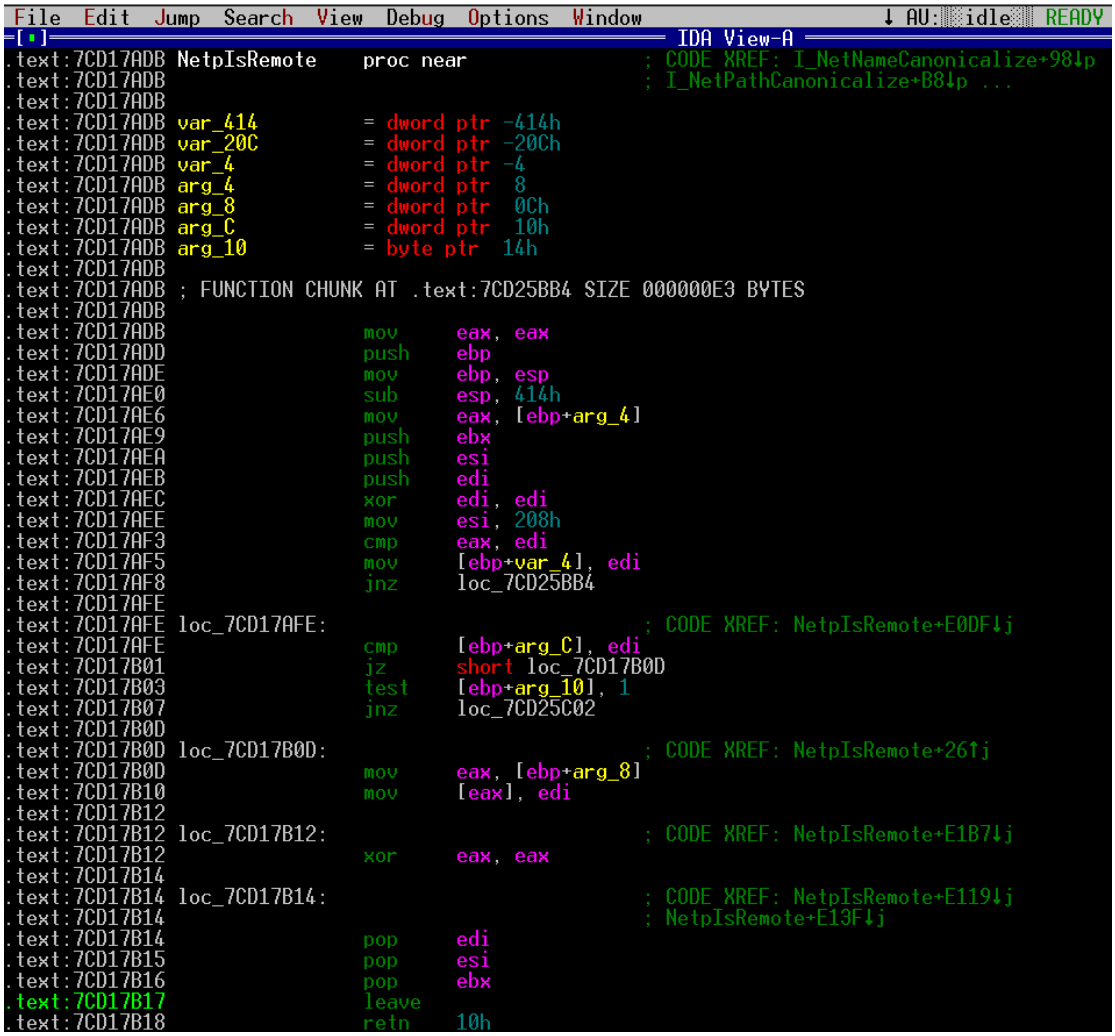

Рисунок 2 дизассемблер — основной хакерский инструмент для целенаправленного поиска дыр

Часто вместо дизассемблера используется "слепой" поиск. Хакер просто выбирает приложение и "скармливает" ему текстовые строки запредельной длины, в надежде, что хоть одна из них вызовет переполнение внутренних буферов. Тоже относится и к двоичным файлам/протоколам обмена данным. Присваивая различным полям заведомо некорректные значения, хакер наблюдает за реакцией программы, дожидаясь пока она не выдержит издевательств и "упадет", после чего останется только исследовать сброшенный дамп памяти и модифицировать уязвимые поля строго дозированным образом, чтобы вместо краха приложение передало бразды правления хакерскому коду. Слепой поиск намного более продуктивен и большинство дыр обнаруживаются именно так, однако, гарантий, что дыра действительно будет обнаружена — в этом случае нет никаких. С другой стороны, количество популярных приложений исчисляется сотнями и даже тысячами. Проанализировать такой объем кода в дизассемблере просто нереально, а вот слепой поиск позволяет _быстро_ обнаружить наиболее грубые дыры, благодаря чему он, собственно говоря, и популярен.

…кто-то глушит кофе, а кто-то чай. Зеленый. Или черный. Неважно что, главное, чтобы продержаться на ногах и не дать закрыться слипающимся глазам. Ага! Блуждание по хакерским блогам и форумам неожиданно выводят на страничку одного японского специалиста по безопасности, обнаружившего дыру в популярной программе Abode PageMaker, приводящую к возможности захвата управления машиной при открытии специальным образом сконструированного документа. Автор не знает японский (хоть и честно пытается его изучить), однако, в данном случае глубоких знаний и не требуется. Английские слова и дизассемблерный код понятны и без пояснительных иероглифов. Есть даже ссылка на демонстрационный (proof-of-concept) exploit, который, правда, написан с ошибками и не работает. Такое впечатление, что его создатель так спешил обнародовать информацию о дыре, что вообще не тестировал свое детище… И ведь есть ему куда спешить. Буквально через считанные часы о дыре узнают

десятки человек, начнется обмен ссылками и образуется что-то вроде цепной реакции. А через несколько дней дыра будет описана на крупных порталах, посвященных безопасности, производители изготовят "противоядие", администраторы начнут скачивать и устанавливать заплатки, но… все это будет не сейчас, все это будет еще более потом. А пока…

А пока автор, заправившись очередной порцией крепкого чая без сахара, скачивает PageMaker из сети, чтобы разобраться с дырой и создать реально работающий exploit, на что уходит порядка шести часов, главным образом потраченных на скачку различных версий PageMaker'a и "универсализацию" exploit'a, поскольку точная версия PageMaker'a, установленная у жертвы, автору неизвестна.

Кстати о жертве. Искать клиентов, когда информация о дыре уже просочилась в Сеть — бессмысленно. Ни за что не успеть, особенно, учитывая, что нам нужен не просто клиент, а клиент, использующий PageMaker – далеко не самую популярную программу, применяющуюся в издательском/полиграфическом бизнесе. И ведь список используемого ПО у клиента не потребуешь, но и действовать вслепую — тоже не вариант.

Открою маленький профессиональный секрет. Поиск клиентов начинается задолго до поиска дыр, после чего переходит в вялотекущую стадию подписания контрактов с кучей многочисленных уточнений, изменений и дополнений. Главное — тянуть время, не забывая о комплексе разведывательных мероприятий, в число которых входит и определение перечня используемого программного обеспечения. Как это делается? Проще простого! Если на очередном витке уточнения договора мы послали клиенту список изменений в формате PageMaker и он не "послал" нас обратно, значит, такая программа у него установлена и, соответственно, наоборот. Кстати говоря, если клиент не обременен интуицией, можно _форсировать_ установку необходимого нам программного обеспечения его же собственными руками. Действительно, мы подготовили документ в формате PageMaker'a и если клиент не может его открыть, то… гм, чуть-чуть социальной инженерии ("PageMaker это же такая удобная и замечательная программа!!!", "…как это так вы не можете открыть документ?! ну так установите PageMaker и откройте! уж не хотите ли вы сказать, что нам теперь нужно перебивать его в Word'e? а вы вообще знаете сколько дыр за последнее время обнаружено в Word'e?!").

Но мне везет. В базе моих клиентов, стоящий в очереди на пен-тестинг, находится организация, использующая PageMaker, после чего мне остается только дождаться утра (после бессонной ночи), связаться с боссом и перевести договоры из вялотекущей стадии в плоскость активных действий. ОК, документ в формате PageMaker'a (начиненный shell-кодом) отослан, договор подписан. Взлом завершен. Shell-код создал новую учетную запись с моим именем и отправил уведомление администратору, пораженному тому как быстро его подломали и что операция пен-тестинга завершилась еще не успев начаться. Во всяком случае так думает администратор, не догадывающийся о скрытой работе, проведенной задолго до начала "ударной" фазы операции.

В чем ошибка администратора? Формально, администратор тут ни при чем. Проблема ведь не в Page Maker'e. В остальных программах дыр даже больше и какой бы набор приложений ни использовала фирма — если она имеет выход в Сеть, если она обменивается документами с внешним миром — взлом _неизбежен_ хоть какие защитные меры ни предпринимай.

Рисунок 3 обитель пен-тестера

>>> врезка Open Source vs. Proprietary Software

Атаковать приложения, распространяемые вместе с исходными текстами, намного сложнее, чем закрытое программное обеспечение и вот почему. Существует конечное и относительно небольшое количество версий закрытых приложений, что делает их предсказуемым с точки зрения хакера, знающего по каким адресам какие данные/команды лежат, что именно содержится в регистрах, etc. Словом, у взломщика есть полная картина и продуманный план действий.

Открытое программное обеспечение может быть перекомпилировано с произвольными настройками произвольным компилятором под любой тип процессора/опциями оптимизации. В результате чего мы получаем практически бесконечное множество вариаций двоичного кода, лишая хакера всякой информации о том, что и где у нас расположено. Как следствие — атака _чрезвычайно_ затрудняется или даже становится невозможной. Естественно, если мы действительно перекомпилируем приложение, а не используем готовую бинарную сборку.

Рисунок 4 открытые операционные системы намного надежнее, чем Windows, но и они могут быть взломаны

история вторая — мертвый сезон или "жучки" в email

Не то, чтобы часто, но все же выпадают такие периоды времени, когда неделями не обнаруживается ни одной свежей критической дыры, пригодной для атаки на систему, а кушать – как гласит известная пословица — хочется даже по ночам. Но, чтобы кушать, пен-тестеру нужно кого-то атаковать и хотя для автора пен-тестинг является всего лишь побочным источником дохода, когда этот источник пересыхает, становится дискомфортно и сухо как в Сахаре.

Но у пен-тестеров всегда есть тузы в рукавах. Вот тут один придур…, ой, извиняюсь, клиент, хочет чтобы испытали его брандмауэр и проникли внутрь корпоративной сети. Дает IP-адрес публичного web-сервера. Доменное имя, слабо сказать, да?! Ну да ладно, мы не гордые, браузер автоматически подставит его и сам. Вот только толку с этого… Web-сервер находится вне охраняемого периметра корпоративной сети и никак не связан с последней. А даже если бы и был связан — это бы не сильно помогло, потому как брандмауэр… ну, короче, эта такая штука… В общем, сошлись с администратором на том, что ему достаточно получить топологию локальной сети, чтобы признать факт атаки состоявшимся. Он уже опробовал несколько хакерских программ, предназначенных для сканирования сети через брандмауэр, убедился в их полной неэффективности, но на всякий случай решил посмотреть как сканируют сети "профессионалы".

Это и стало той фатальной ошибкой, которая позволила автору пополнить свой счет без особых напряжений мозговых извилин. Почему-то (всегда хотел знать почему!) некоторые администраторы страшно бояться слова "сканирование" и берегут список внутренних IP-адресов как зеницу ока, не зная что его можно получить кучей различных методов.

Хорошо, заказ получен, договор подписан. Идем, значит, на этот публичный сайт и находим на нем публичный email. Отправляем письмо с ненапряженным текстом со своего собственного почтового сервера. Обыкновенное такое письмо в HTML-формате со ссылкой на картинку. Обыкновенную такую картинку, расположенную опять-таки на подконтрольном автору Web-сервере. И… сюрприз!!! Корпоративная сеть, естественно, имеет доступ в Интернет (а иначе как бы она могла получать е-майлы?!) и Web-доступ через Proxy-сервер тоже имеется. Почтовый клиент вполне стандартный — Outlook Express, HTML-формат, естественно, не запрещен, загрузка картинок не отключена и Proxy конечно же не анонимный, а потому он честно выдает реальный IP-адрес клиента, который, кстати говоря отличается от IP-адреса, прописанного в заголовке ответного письма, что наводит на определенные размышления. Компания достаточно крупная, она использует свой собственный почтовый сервер, конфигурация которого предписывает включать IP-адрес отправителя в заголовок. При нормальном ходе вещей эти адреса должны совпадать, а их несовпадение с высокой степенью

вероятности указывает на то, что узел-отправителя имеет несколько интерфейсов и, соответственно, несколько IP-адресов, причем, эти адреса принадлежат различным подсетям. Возникает предположение о беспроводной сети, подключенной к основной локальной сети посредством транслятора сетевых адресов (он же NAT), ну а техника "разоблачения" трансляторов хорошо известна хакерам, благо, что почтовый сервер организации "заботливо" прописывает адреса локальных портов в заголовке письма и после достаточно непродолжительной переписки с секретарем (по публичному адресу) автор не только заполучил еще ряд электронных адресов, не указанных на Web-сайте компании, но и даже определил тип сетевого транслятора (кстати, аппаратный), а по нему и тип используемого беспроводного оборудования.

Забавно, что в заголовке электронного письма, полученного от администратора, один и тот же IP-адрес упоминался дважды — прямое следствие отправления писем непосредственного с почтового сервера. В принципе мелочь, но из таких мелочей постепенно складывалась общая топология сети.

Ссылка на картинку, вставленная в письма, позволила среди всего прочего распознать наличие различных защитных комплексов, в частности, антивирусов и спам-фильтров. Спам-фильтры, обрабатывая входящую корреспонденцию, неизбежно загружали картинку с помощью GET-запроса, который существенно отличается от GET-запроса Proxy-серверов и почтовых клиентов. Используемое антивирусное обеспечение имело непримечательный GET, но обладало уникальным алгоритмом загрузки картинки. Для "разгрузки" трафика загружалась не вся картинка, а только небольшая часть ее на основе анализа которой делался вывод о ее (не)безвредности. Количество считываемых байт от начала файла сыграло роль своеобразного "отпечатка пальца" по которому был идентифицирован не только производитель антивируса, но даже и диапазон версий последнего.

Получив (и отправив) в общей сложности несколько сотен писем сотрудникам компании (естественно, не от своего имени, а под видом потенциального клиента), автор "декодировал" топологию корпоративной сети (в которой оказалось две беспроводных сети, три системы обнаружения вторжений, кэширующий HTTP-сервер, Proxy-сервер, анти-спам фильтр, брандмауэр уровня приложений и антивирусное обеспечение, установленное как на почтовом сервере, так и на клиентских узлах).

А теперь финал. Ряд используемого программного обеспечения имел известные, но не заткнутые администратором дыры (которому, наверное, было лень скачивать обновления), что позволило запустить в сеть "шпиона" — исполняемый код, захвативший управление целевым узлом и выполняющий сканирование непосредственно изнутри локальной сети, что завершило построение топологии.

Сказать, что администратор был ошеломлен, означает ничего не сказать. Да еще св. Кондратий едва не хватил! Он-то считал, что надежно защитил вверенную ему сеть от всевозможных злоумышленников, а тут ему не только выдали полную топологию, но еще и засадили зловредную программу и ведь все это время администратор не спал, а протоколировал все события, весь трафик и втыкал в километровые логи на предмет поиска подозрительной активности, но ничего подозрительного в логах не было.

Вердикт: если действительно хочется скрыть интимные подробности организации локальной сети, то следует сконфигурировать все программное обеспечение так, чтобы оно не вставляло IP-адреса в заголовки пакетов. Но даже в этом случае у злоумышленника остается возможность идентификации установленного программного обеспечения по характеру загрузки WEB-элементов и содержимому служебных полей GET-запросов. "Стереть" же все "отпечатки пальцев" практически невозможно. Во всяком случае, квалификации рядового администратора для этого явно недостаточно.

Рисунок 5 разбитое стекло хрупкой иллюзии безопасности

история третья — от обновлений не всегда бывает польза

И вот… настоящее испытание. Клиент, заказавший пен-тестинг, использует в качестве внешних узлов сервера на базе различных версий Linux и BSD, настроенных на максимальный уровень безопасности. Узлы, имеющие доступ во внешний мир, выделены в отдельную подсеть, огороженную брандмауэрами и прочими защитными комплексами, построенными на базе OpenBSD — самой защищенной операционной системе.

Даже если отправить письмо на публичный адрес, и захватить полный контроль над узлом получателя это не даст абсолютно ничего, поскольку получатель находится внутри охранного периметра, а вся внутренняя переписка завязана на локальную сеть _физически_ отключенную от Интернет и внутренней сети, общающейся с внешним миром.

Автору было предложено проникнуть внутрь сети, физически отключенной от всех узлов, контактирующих с внешним миром. Отключенной в прямом смысле этого слова. И даже для активации Windows использовалась специальная процедура, предусматривающая перенос ключей на сервер активации через сменные носители, что указывает на высочайший уровень безопасности, обусловленный высоким уровнем секретности обрабатываемых данных.

Как может быть то, чего вообще не может быть? Как достичь того, чего вообще нельзя достичь?! Как, черт возьми, проникнуть в локальную сеть, отключенную от внешнего мира?! Предложение нацепить маску, ворваться внутрь здания, перестрелять охрану, заполучив физический доступ к защищенным узлам — увы, не вариант. Это не шпионский боевик, это реальная жизнь! А автор отнюдь не герой фильма "Foolproof" и по спортивной подготовке значительно уступает мистеру Андерсону из "Матрицы".

Рассуждая логически — допустим, один из узлов внутренней сети имеет модем. Или беспроводной адаптер. Или… но, увы, даже если это и так… модем наверняка подключен к внутренней АТС, не имеющей выхода во внешний мир, а, если и имеющей, то автор по любому не специалист по АТС и совершенно не представляет как их можно взломать. Беспроводной адаптер? Что ж, вполне вероятно. Кстати сказать, стек протоколов для "Голубого зуба" до сих пор сырой как вода и дыр там… Без всяких паролей и других штучек посылкой специальным образом "снаряженных" пакетов можно вызывать переполнение буфера и получить наивысший уровень привилегий. Остальное — уже дело техники. Но прежде чем эта техника вступит в строй, необходимо найти узел с адаптером Голубого Зуба и установить с ним связь, что очень

непросто, поскольку физическое расположение здания компании автору было неизвестно, а заниматься реальным шпионажем в планы как-то не входило.

Ситуация казалась безнадежной, но… автор все же решил попробовать. Определив по IP-адресам внешних узлов название компании Интернет-провайдера, услугами которого пользовался заказчик атаки, автор зашел на его головной сайт, поблуждав по которому наткнулся на сервер обновлений для Linux и BSD-систем. Подобные сервера предоставляют достаточно многие провайдеры. Действительно, какой смысл всем клиентам качать обновления напрямую с далеких зарубежных серверов, когда провайдер, скачав их всего один раз, может перевести клиентов на локальный трафик (значительно более дешевый, да и скорость повыше будет).

Возникло сумасшедшее предположение — а что, если администратор этой засекреченной организации качает обновления не напрямую, а берет их непосредственно с сервера провайдера? Наскоро заключив договор на пен-тестинг с провайдером (на условиях очень выгодных для провайдера), автор легко взломал его, получив доступ к серверу обновлений. Дальше все пошло как по маслу. Администратор засекреченной организации действительно качал обновления не напрямую и утянул "подложное" обновление вместе с остальными. После чего, следуя служебной инструкции, записал его на DVD, вытащил сменный носитель из лотка привода и держа его в руках вошел в святую святых — внутрь особо охраняемой сети, физически отрезанной от всего внешнего мира.

Естественно, доступа к охраняемым данным автор так и не получил, т.к. никакие данные не выносились наружу (администратору это было делать строго запрещено), однако, возможность модификации данных (удаление, искажение файлов), естественно, была приобретена и автор, создав в определенном каталоге файл с определенным содержанием, отрапортовал об успешной атаке.

Теперь администратор качает все обновления напрямую, что, конечно, затрудняет атаку, но отнюдь не делает ее невозможной, особенно в случае использования операционных систем с открытым кодом. Задумывались ли вы сколько людей имеют доступ к коду? Сколько хакеров могут посылать заплатки, обновления, etc и насколько тщательно они проверяются перед вкладыванием на официальный сервер обновлений?!

заключение

Мы живем в небезопасном мире и ключевое слово здесь _живем_ в смысле существуем вопреки всем угрозам. Иначе и не может быть. Компьютеры неидеальны, защитные механизмы несовершенны, не взламываемые системы существуют только на бумаге, однако, при всей шаткости коммуникационных систем цивилизация не только не загибается, но даже еще растет, процветает и развивается.

Да, взломы случаются. Их можно затруднить, но полностью предотвратить угрозу атаки невозможно в принципе!