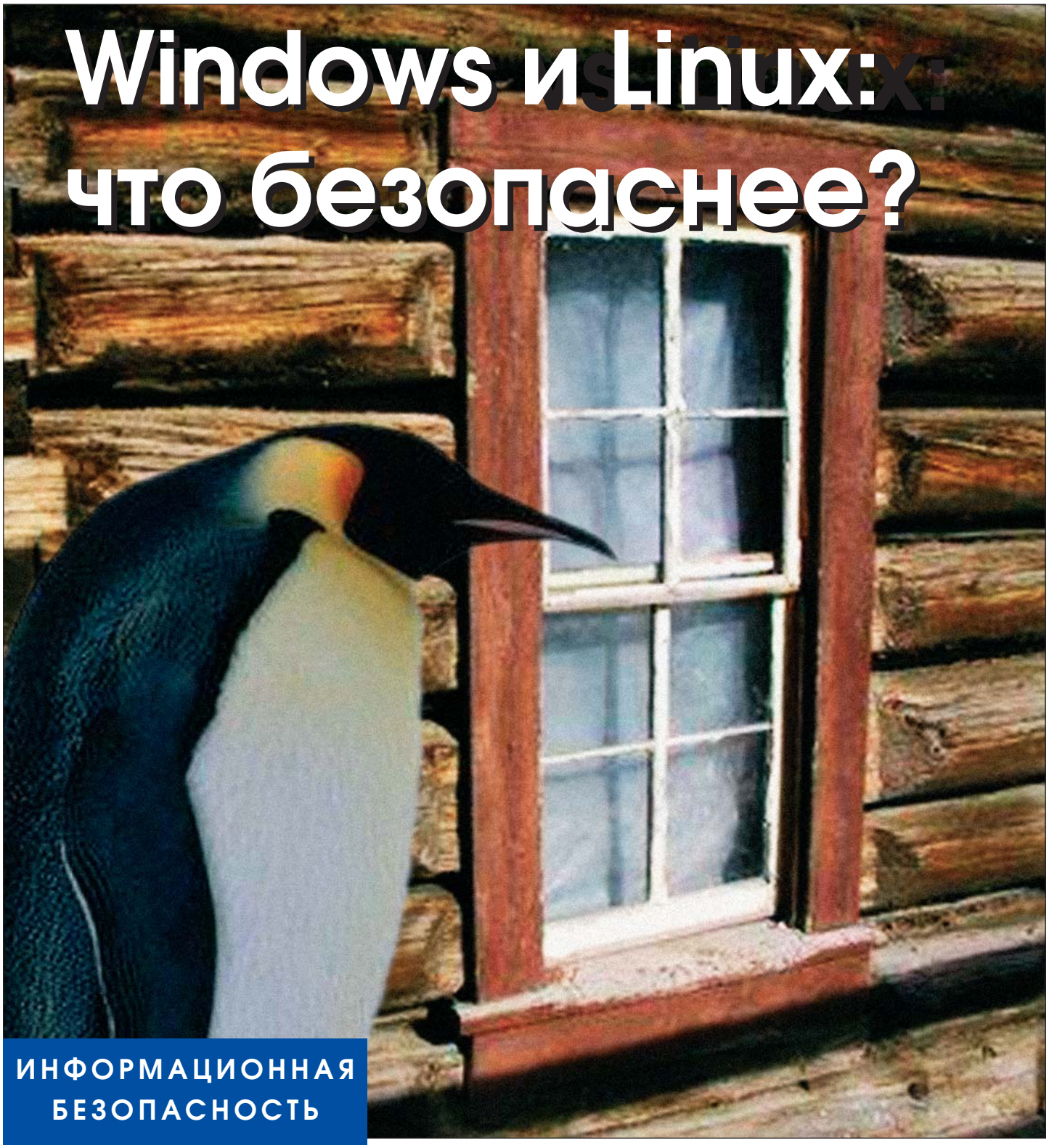
Jet Info

ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ № 3 (142)/2005

Windows и Linux: что безопаснее?



ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Windows и Linux: что безопаснее?

Николас Петрели

СОДЕРЖАНИЕ

Николас Петрели (Nicholas Petreley) работает в компьютерной индустрии почти двадцать лет. Он был внештатным автором, редактором, консультантом, преподавателем и программистом, работал исполнительным редактором в испытательном центре «InfoWorld» и главным редактором Интернет-издания «NC World Magazine».

В настоящее время Николас Петрели занимает должность аналитика по Linux в исследовательской компании Evans Data Corporation, работает в качестве обозревателя в журнале «ComputerWorld», а также пишет статьи и аналитические обзоры для Интернет-издания «LinuxWorld», основанного им в 1998 году.

Работы Николаса Петрели неоднократно отмечались различными наградами и призами.

От редакции

В последнее время во всем мире растет популярность различных версий операционной системы Linux, они все чаще используются как в частном бизнесе, так и государственными структурами. Такая же тенденция наблюдается и в России. Конкуренция операционных систем обостряет противостояние между Linux-сообществом и сторонниками Windows во главе с компанией Microsoft. Стороны приводят множество аргументов в пользу «своих» систем, и далеко не всегда очевидно, какие из них — маркетинговый ход, а какие — результат беспристрастного исследования. Так что же выбрать? Какие критерии применить?

Один из решающих факторов при выборе операционной системы — насколько хорошо она обеспечивает информационную безопасность. Именно вокруг сравнения безопасности Linux и Windows давно кипят нешуточные страсти.

Первым залпом в войне Windows и Linux стал доклад компании Forrester Research, в котором было показано, что Windows безопаснее, чем Linux. Для сравнения использовались собранные за период с 1 июня 2002 г. по 31 мая 2003 г. данные об уязвимостях в защите для всех платформ Windows и всех вариантов дистрибутивов Linux от Debian, MandrakeSoft, Red Hat и Novell. Как и следовало ожидать, сообщество Linux не осталось в долгу. Сразу после публикации доклада упомянутые в нем дистрибьюторы Linux выпустили совместное заявление, в котором утверждалось, что доклад Forrester Research вводит в заблуждение, особенно потому, что в нем «все уязвимости ошибочно рассматриваются как равнозначные, независимо от риска, который возникает вследствие этих уязвимостей».

Мы предлагаем вашему вниманию независимый обзор Николаса Петрели (Nicholas Petreley), который рассмотрел доводы обеих сторон. Отчет опубликован 22 октября 2004 г. в британском Интернет-издании The Register. Перевод на русский язык выполнен с разрешения автора.

И все же предлагаемый обзор — не окончательное решение данного вопроса.

Так, можно отметить, что Петрели не учел временной фактор, который мог бы показать, как по мере выхода новых версий изменяется ситуация с уязвимостями для той или иной платформы — улучшается или ухудшается, и как часто пользователи сталкиваются с критическими проблемами.

И если важно вести перечень уязвимостей и подсчитывать их количество, то не менее важно оценить, насколько быстро становятся доступными для пользователей программные коррекции уязвимостей, причем как критических, так и менее серьезных.

Кроме того, чтобы получить действительно беспристрастную картину, следовало бы сформировать стандартные «комплекты» Linux и Windows и проследить в течение определенного времени за уязвимостями этих комплектов, используя для анализа показатели, предложенные Петрели.

Но несмотря на отдельные недостатки, данный обзор — обоснованное мнение эксперта. В этом качестве он будет полезен всем, кого интересует относительная безопасность конкурирующих операционных систем — Windows и Linux.

О чем эта статья?

Много копий было сломано в спорах о том, действительно ли Linux — более безопасная операционная система, чем Windows. Мы сравнили Windows и Linux по значениям следующих параметров в 40 последних программных коррекциях/уязвимостях для Microsoft Windows Server 2003 и Red Hat Enterprise Linux AS v.3:

1. Серьезность уязвимостей в системе безопасности, которая оценивается по следующим показателям:
   - возможный ущерб (насколько велик вред от использования ОС с незакрытой уязвимостью?);
   - возможность использования (насколько легко использовать данную уязвимость?);
   - возможная доступность (какого рода доступ требуется для использования данной уязвимости?).
2. Количество уязвимостей, серьезность которых определяется как Критическая.

Результаты сравнения не оказались неожиданными. Даже по субъективным и заниженным стандартам, применяемым Microsoft, по меньшей мере 38% последних программных коррекций предназначены для ликвидации брешей, которые Microsoft относит к критическим. Только 10% программных коррекций и предупреждений в Red Hat относятся к брешам, имеющим критический уровень серьезности. Приведенные результаты получены при условиях, благоприятных для Microsoft и обоснованно жестких для Red Hat, так как они основаны на критериях Microsoft, а не на используемых нами более строгих показателях безопасности. Если применить наши собственные критерии, то количество критических брешей в Windows Server 2003 возрастет до 50%.

Результаты запроса к базе данных Computer Emergency Readiness Team (CERT) подтвердили наши выводы, сделав разницу еще более значительной. Расположив полученные данные по убыванию серьезности (от более критических к менее критическим), мы обнаружили, что уровень серьезности 39 из первых 40 записей в базе данных CERT для Windows превышает пороговое значение, установленное CERT для серьезных предупреждений. Лишь три из первых 40 записей оказались выше указанного порога в результатах запроса к этой базе данных для Red Hat. Запрос к базе данных CERT для Linux показал, что только шесть из первых 40 записей находятся выше этого порогового значения.

Следует также учесть тот факт, что в списки как для Red Hat, так и для Linux включаются бреши в программном обеспечении, которое функционирует в Windows, а это означает, что такие бреши относятся одновременно и к Linux, и к Windows. Ни одно из предупреждений, связанных с Windows, не относится к программному обеспечению, функционирующему в Linux.

Почему же тогда так много, на первый взгляд, убедительных уверений в том, что операционная система Linux в действительности менее безопасна, чем Windows? Обоснование вывода о меньшей безопасности Linux содержит вопиющие логические несоответствия. Нужно всего лишь чуть внимательнее рассмотреть этот вопрос, чтобы развенчать мифы и найти логические ошибки, лежащие в основе следующих часто повторяемых утверждений:

1. Windows подвергается такому количеству атак только потому, что имеет больше инсталляций, чем Linux. Следовательно, Linux была бы столь же уязвима, если бы имела столько же инсталляций.
2. Открытый код по природе своей значительно опаснее, поскольку злоумышленникам легче найти бреши в системе безопасности.
3. Для Linux имеется больше предупреждений об уязвимостях, чем для Windows, следовательно, Linux менее безопасна, чем Windows.
4. В случае операционной системы Linux проходит больше времени между обнаружением бреши и выпуском соответствующей программной коррекции, чем в случае Windows.

Ошибка утверждений 3 и 4 в том, что они игнорируют наиболее важные показатели, позволяющие оценить безопасность одной операционной системы по сравнению с другой. Как будет показано в разделе «Реальные показатели безопасности и серьезности», попытки характеризовать безопасность на основании одного показателя (например, по тому, сколько времени проходит между обнаружением бреши и выходом программной коррекции) не дают значащих результатов.

В заключение дан краткий обзор существенных различий в концепциях Windows и Linux, что позволяет понять, почему операционная система Windows более уязвима к атакам как на серверах, так и на настольных компьютерах, и почему Linux является более безопасной системой.

Развенчание мифов

Миф 1. Безопасность – вопрос количества: чем меньше инсталляций, тем безопаснее

Пожалуй, чаще всего повторяемый миф при сравнении безопасности Windows и Linux — утверждение, что с ОС Windows случается больше инцидентов, связанных с вирусами, сетевыми червями, троянскими программами и другими проблемами, только из-за того, что злоумышленники предпочитают вмешиваться в работу программного обеспечения, которое имеет наибольшую инсталляционную базу. Это рассуждение приводят в защиту ОС Windows и Windows-приложений. Windows преобладает на настольных компьютерах; именно поэтому на Windows и Windows-приложения направлено большинство атак, и именно поэтому не наблюдаются вирусы, сетевые черви и троянские кони для Linux. Нельзя отрицать, что до некоторой степени это верно, однако отнюдь не бесспорен вывод, который отсюда делается, а именно: Linux и Linux-приложения не являются более безопасными, чем Windows и Windows-приложения, просто Linux — слишком незначительная цель для того, чтобы тратить усилия на организацию атаки.

Это рассуждение легко опровергнуть, если учесть, что самым популярным программным обеспечением для web-серверов Интернета является Apache. Как следует из выполненного компанией Netcraft[1] обзора web-сайтов за сентябрь 2004 г., 68% web-сайтов используют web-сервер Apache и только 21% web-сайтов используют Microsoft IIS. Если проблемы с безопасностью возникают из-за того, что злоумышленники нацеливаются на самую обширную инсталляционную базу, то должно наблюдаться больше червей, вирусов и прочих вредоносных программ, нацеленных на Apache и те операционные системы, под управлением которых он функционирует, чем на Windows и IIS. Более того, должно регистрироваться больше атак против Apache, чем против IIS, так как из приведенного выше рассуждения следует, что проблема заключается в количествах, а не в уязвимостях.

Однако в реальности наблюдается обратная картина. В течение долгого времени IIS был главной целью для сетевых червей и других атак, и эти атаки были весьма успешными. Сетевой червь Code Red, который использовал переполнение буфера в сервисе IIS для получения контроля над web-серверами, заразил около 300 тысяч серверов, и его распространение прекратилось лишь потому, что это было предусмотрено в программном коде данного червя. Воздействие еще одного сетевого червя IISWorm оказалось ограниченным только благодаря тому, что эта программа была плохо написана, а вовсе не из-за успешной защиты IIS.

Да, известны сетевые черви для Apache, например, Slapper. (В действительности Slapper использует известную уязвимость в протоколе OpenSSL, а не в Apache). Но сетевые черви для Apache редко становятся сенсацией, поскольку их воздействие очень ограничено и они легко искореняются. На атакуемых сайтах уже были приняты меры, исключающие возможность использования ошибки в OpenSSL. Кроме того, благодаря модульной структуре Linux и UNIX было очень просто очистить и восстановить зараженные сайты. Для этого потребовалось всего несколько команд и не было необходимости в перезагрузке.

Возможно, именно поэтому в рейтинге Netcraft среди 50 web-сайтов, лидирующих по длительности непрерывной работы (промежуток времени между перезагрузками), оказалось 47, использующих Apache[2]. Ни на одном из этих 50 web-сайтов не использовались ни Windows, ни Microsoft IIS. Поэтому, если верно предположение, что злоумышленники атакуют наиболее распространенные программные платформы, то возникает вопрос: почему хакеры так успешно взламывают программное обеспечение и операционную систему, самые популярные среди настольных компьютеров, заражают 300 тысяч серверов IIS, но неспособны нанести подобный ущерб наиболее популярному web-серверу и его операционным системам?

Читатели, посетившие web-сайт Netcraft, не могут не заметить, что все 50 серверов из списка лучших по длительности непрерывной работы используют какую-либо разновидность BSD, в основном BSD/OS. Ни один из них не функционирует под управлением Windows и ни один — под управлением Linux. Самое продолжительное время непрерывной работы среди 50 лидеров составляет 1 768 дней подряд или почти 5 лет.

В результате складывается впечатление, что BSD по своей надежности превосходит все остальные операционные системы, однако информация

[1] Netcraft Web Survey for September 2004 — http://news.netcraft.com/archives/2004/09/index.html

[2] Netcraft Top 50 Servers With Longest Uptime (результаты могут отличаться от приведенных, так как информация обновляется ежедневно) — http://uptime.netcraft.com/up/today/top.avg.html

Netcraft, хотя и непреднамеренно, вводит в заблуждение. Netcraft определяет продолжительность непрерывной работы операционных систем по данным, которые регистрируются самими операционными системами. Linux, Solaris, HP-UX и некоторые версии FreeBSD регистрируют не более 497 дней непрерывной работы, после чего счетчики обнуляются, и отсчет начинается заново. Поэтому и кажется, что все web-сайты, размещенные на компьютерах под управлением Linux, Solaris, HP-UX и, в некоторых случаях, FreeBSD, перезагружаются каждые 497 дней, даже если они функционируют годами. При составлении обзора Netcraft ни для одной из этих операционных систем не будет зарегистрировано время непрерывной работы, превышающее 497 дней, даже если они годами функционируют без перезагрузки, так что названные системы никогда не окажутся среди 50 лидеров.

Это объясняет, почему Linux, Solaris и HP-UX не могут продемонстрировать такой же впечатляющий результат по количеству последовательных дней непрерывной работы, как BSD, даже если на самом деле эти операционные системы годами функционируют без перезагрузки. Но это не объясняет, почему среди 50 лидеров нет серверов под управлением ОС Windows. Windows не обнуляет счетчик времени непрерывной работы. Очевидно, просто не существует web-сайтов под управлением ОС Windows, которые функционировали бы без перезагрузки достаточно долго, чтобы попасть в список 50 лучших.

Принимая во внимание казус с 497-дневным циклом обнуления становится неясным, как сравнить продолжительность непрерывной работы для Linux и Windows, основываясь на опубликованных данных Netcraft. Два результата — не статистика, но и они кое о чем говорят, особенно если один из них относится к web-сайту компании Microsoft. На сентябрь 2004 года средняя продолжительность непрерывной работы web-серверов под управлением ОС Windows, поддерживающих собственный web-сайт компании Microsoft (www.microsoft.com), составила примерно 59 дней. Максимальная продолжительность непрерывной работы для Windows Server 2003 на том же сайте — 111 дней, минимальная — 5 дней. Сравним эти результаты с данными для сайта www.linux.com (типовой сайт под управлением ОС Linux), у которого и средняя, и максимальная продолжительность непрерывной работы составляет 348 дней. Точное совпадение средней и максимальной величины означает, что эти серверы либо обнулили счетчик времени 348 дней назад, проработав до этого 497 дней, либо 348 дней назад они были перезагружены или впервые включены.

Из всего вышесказанного следует вывод, что при оценке количества успешных атак на программное обеспечение решающий фактор — это качество, а не количество.

Миф 2. Открытый код опасен по определению

Впечатляющие данные о продолжительности непрерывной работы, полученные для Apache, позволяют усомниться еще в одном популярном мифе: открытый исходный код (в котором функциональная структура приложений является общедоступной) более опасен, чем патентованный исходный код (в котором функциональная структура является засекреченной), поскольку хакеры могут воспользоваться этим кодом для обнаружения и использования брешей.

Однако факты говорят о другом. Количество предназначенных для ОС Windows эффективных вирусов, программ-шпионов, сетевых червей, троянских и других вредоносных программ огромно, а количество компьютеров, постоянно заражаемых различными комбинациями вышеперечисленного, так велико, что вряд ли поддается исчислению. Вредоносное программное обеспечение «свирепствует» настолько, что для компрометации свежеустановленной, «непропатченной» версии Windows XP после подключения компьютера к Интернету требуется в среднем 16 минут — меньше, чем время, необходимое данной ОС для загрузки и инсталляции программных коррекций, которые позволят защитить этот компьютер[3].

Еще один пример. Web-сервер Apache имеет открытый код. Microsoft IIS — патентованный. В этом случае факты опровергают и миф о наибольшей популярности, и миф об опасности открытого кода. Web-сервер Apache — наиболее популярный web-сервер. Если бы оба эти мифа были правдой, следовало бы ожидать, что Apache и операционные системы, под управлением которых он функционирует, будут чаще подвергаться атакам, чем Microsoft Windows и IIS. На деле же все происходит с точностью до наоборот. Apache занимает почти монопольное положение в списке серверов с наибольшей продолжительностью непрерывной работы. Ни Microsoft Windows, ни

[3] Unpatched PC «Survival Time» Just 16 Minutes, Gregg Keizer, TechWeb News
http://www.internetweek.com/breakingNews/showArticle.jhtml?articleID=29106061

Microsoft IIS нет в списке 50 серверов с самым продолжительным временем непрерывной работы. Очевидно, тот факт, что злоумышленники имеют доступ к открытому коду Apache, не дает им никакого преимущества в организации более успешных атак против Apache, чем против IIS.

Миф 3. Выводы на основании единственной характеристики

Слабость других популярных мифов о большей безопасности Windows по сравнению с Linux в том, что они основаны на единственном показателе — каком-нибудь одном аспекте измерения безопасности. Это верно для любого источника информации, будь то данные компетентного исследования или малообоснованная информация на уровне слухов.

Широко известно утверждение: «Для Linux зарегистрировано больше предупреждений об уязвимостях, чем для Windows, поэтому можно говорить о меньшей безопасности Linux по сравнению с Windows». Не менее распространено и такое: «Время между обнаружением бреши и созданием соответствующей программной коррекции в случае Linux больше, чем в случае Windows, поэтому можно говорить о меньшей безопасности Linux по сравнению с Windows».

Второе утверждение совершенно непостижимо. Невозможно объяснить, почему считается, что по такому показателю, как среднее время от момента обнаружения бреши до выпуска соответствующей программной коррекции, Microsoft превосходит **какую-либо** из конкурирующих операционных систем, не говоря уже о Linux. Компании Microsoft потребовалось семь месяцев (!), чтобы устранить одну из самых серьезных уязвимостей в системе защиты (Microsoft Security Bulletin MS04-007 ASN.1 Vulnerability, компания eEye Digital Security сообщила об этой задержке в информационном бюллетене AD20040210). А ведь еще имеются бреши, о которых Microsoft открыто заявляет, что они **никогда** не будут ликвидированы. В бюллетене Microsoft Security Bulletin MS03-010 об уязвимости Denial Of Service (отказ в обслуживании) в Windows NT говорится, что эта уязвимость никогда не будет устранена. Позднее компания Microsoft заявила, что не будут устраняться уязвимости в Internet Explorer для всех версий операционных систем, выпущенных ранее Windows XP. В статистическом смысле случай с семью месяцами между обнаружением и устранением ошибки не окажет значительного влияния на среднее время реагирования при условии, что найдется достаточно много примеров чрезвычайно быстрого реагирования, которые скомпенсируют подобные аномалии, если это действительно аномалии. Но стоит включить в выборку только один случай «никогда» — и о статистическом среднем можно забыть навсегда.

Оставив в стороне эту неразрешимую загадку, рассмотрим, есть ли основания полагать, что Linux менее безопасна, чем Windows, потому что среднее время между обнаружением уязвимости и выпуском соответствующей программной коррекции в случае Linux больше, чем в случае Windows. Зададим себе вопрос: «Если прямо сейчас у меня случится сердечный приступ, в какое отделение скорой помощи я предпочту попасть? Туда, где среднее время между регистрацией пациента и оказанием ему помощи самое короткое? Или я предпочту отделение, где это среднее время дольше, но зато пациенты с самыми серьезными диагнозами получают помощь в первую очередь?»

Очевидно, что следует выбрать второе отделение, но вовсе не потому, что из приведенной выше информации ясно, что оно лучше первого. Этот вариант предпочтительнее, потому что он выбран по двум показателям, один из которых особенно важен для вас именно в данный момент. Можно с уверенностью утверждать: мало кто захочет попасть в больницу, о которой известно, что в ней есть шанс умереть от сердечного приступа, ожидая, пока врач закончит вправлять кому-то вывихнутый мизинец — каким бы впечатляющим ни было «среднее время реагирования на запрос о медицинской помощи». Трудность в том, что в приведенном примере недостаточно информации для принятия оптимального решения. Ничего не известно о том, как больница с наименьшим «временем реагирования» определяет, кому из больных следует уделить внимание в первую очередь. Кроме того, неплохо бы иметь информацию о показателе смертности среди больных, нуждающихся в неотложной помощи, о квалификации врачей и т. д.

Очевидно, что для правильного выбора есть только один способ — собрать об окрестных отделениях скорой помощи подробную информацию, т.е. узнать как можно больше важных показателей и найти разумный компромисс между этими показателями. Было бы непростительной безответственностью рекомендовать отделение скорой помощи для случая сердечного приступа, основываясь только на одном показателе, например, на времени реагирования, среднем для всех неотложных случаев, особенно если без труда можно

получить другую важную информацию, которая позволит сделать лучший выбор.

В той же степени неразумно и безответственно давать рекомендации или принимать серьезное экономическое решение на основании единственного показателя, такого как среднее время между обнаружением бреши и ее устранением для той или иной операционной системы, или такого показателя, как количество предупреждений об уязвимостях для того или иного продукта.

Любой отдельно взятый показатель может ввести в заблуждение, поскольку неясна его значимость. Рассмотрим утверждение, что известно больше предупреждений для программного обеспечения Linux, чем для Windows. Эта статистика бессмысленна, потому что не отвечает на самые важные вопросы. Сколько брешей из зафиксированных всеми предупреждениями об уязвимостях создают реальные угрозы? Насколько серьезны эти угрозы? Насколько вероятно, что их использование причинит серьезный ущерб компьютерам? Ответы на эти вопросы имеют большое значение. Что лучше, операционная система со 100 брешами, из-за которых возможен незначительный ущерб (а то и вовсе никакого) и которыми не сможет воспользоваться никто, кроме локальных пользователей с действующими учетными записями и физическим доступом к компьютеру? Или стоит предпочесть операционную систему с единственной критической брешью, из-за которой любой хакер из Интернета может уничтожить всю информацию на сервере? Ясно, что само по себе количество предупреждений не является значимым показателем, указывающим на большую безопасность одной операционной системы по сравнению с другой.

Архитектура Windows и архитектура Linux

Возможно, причиной мифа о том, что Windows намного чаще подвергается атакам, чем Linux, являются вирусы, троянские кони и сетевые черви, ориентированные на электронную почту и web-навигатор. Несомненно, на настольных компьютерах инсталляций Windows больше, чем инсталляций Linux. Весьма возможно, что программное обеспечение Windows **для настольных компьютеров** чаще подвергается атакам, потому что Windows преобладает на настольных компьютерах. Но остается без ответа один важный вопрос. Действительно ли атаки на Windows столь успешны исключительно из-за их многочисленности, или же этот факт объясняется наличием проектных недоработок и неудачных проектных решений Windows?

Значительная часть (если не большинство) вирусов, троянских коней, сетевых червей и других вредоносных программ, заражающих компьютеры с Windows, используют для этого уязвимости в Microsoft Outlook и Internet Explorer. Поэтому можно поставить вопрос по-другому, включив в рассмотрение тот же тип программного обеспечения для Linux (наиболее часто используемые web-навигаторы, программы электронной почты, текстовые процессоры и т. д.): имеет ли Linux столько же уязвимостей в системе безопасности, сколько их имеет Windows?

Архитектура Windows

Вирусы, троянские кони и другие вредоносные программы поражают настольные компьютеры с Windows по целому ряду причин, свойственных Windows и чуждых Linux:

1. Windows только недавно эволюционировала от однопользовательской модели к многопользовательской.
2. Windows по своей архитектуре является монолитной, а не модульной системой.
3. В Windows слишком широко используется RPC-механизм.
4. Windows фокусируется на знакомом графическом интерфейсе для настольных компьютеров.

Windows только недавно эволюционировала от однопользовательской модели к многопользовательской

Критики Linux любят повторять, что Linux — это «устаревшая» технология. Ирония же заключается в следующем: одна из самых больших проблем Windows в том, что именно этой операционной системе никак не удается избавиться от своей «устаревшей» однопользовательской архитектуры. В течение долгого времени происхождение от однопользовательской системы мешало Windows.

Система Windows изначально была разработана так, чтобы обеспечить и пользователям, и приложениям свободный доступ ко всей системе, а это значит, что кто угодно мог скомпрометировать критичную системную программу или файл. Кроме того, это означает, что вирусы, троянские и другие вредоносные программы могли скомпрометировать любую системную программу или файл, потому что Windows не изолировала пользователей и приложения от критичных областей операционной системы.

Операционная система Windows XP стала первой версией Windows, в которой проявились существенные результаты усилий по изолированию пользователей от системы, так что в Windows XP каждый пользователь имеет свои собственные личные файлы и ограниченные системные полномочия. Но из-за этого многие Windows-приложения, разработанные для предыдущих версий, перестали функционировать, поскольку раньше они могли получать доступ и модифицировать программы и файлы, доступ к которым теперь разрешен только администратору. Именно поэтому в Windows XP предусмотрен режим совместимости, то есть режим, который позволяет программам функционировать так, будто они работают в прежней незащищенной однопользовательской среде. Кроме того, именно по этой причине каждая новая версия Windows несет в себе угрозу отказа приложений, работавших в предыдущих версиях. Поскольку Microsoft вынуждена модифицировать Windows, чтобы она «вела себя» как многопользовательская система, новые ограничения выводят из строя приложения, которые функционировали при отсутствии этих ограничений.

Windows XP — это прогресс, но и Windows XP нельзя назвать настоящей многопользовательской системой. Например, Windows XP поддерживает функцию, которую Microsoft называет «Fast User Switching» (быстрое переключение пользователей), она же позволяет двум и более пользователям входить в систему Windows XP на одном компьютере в одно и то же время. Однако есть загвоздка. Это возможно тогда и **только тогда**, когда данный компьютер **не** принадлежит какому-либо домену сети Windows. Причина в том, что сеть Microsoft разработана в предположении, что пользователи входят в сеть **только** со своего собственного компьютера. Microsoft либо не может, либо не желает внести необходимые изменения в операционную систему и «конструкцию» сети, чтобы адаптировать этот сценарий к возможностям Windows XP.

Windows Server 2003 — это следующее приближение к настоящей многопользовательской системе, но даже в Windows Server 2003 не удалось ликвидировать все бреши в системе защиты, унаследованные от однопользовательской системы. Именно поэтому в Windows Server 2003 пришлось отключить использование по умолчанию многих функций web-навигатора (например, ActiveX, написание сценариев и др.). Если бы Microsoft перестроила эти функции для работы в безопасном и изолированном режиме в настоящей многопользовательской среде, они не создавали бы серьезных угроз, перед которыми по-прежнему беззащитна Windows.

Windows по своей архитектуре является монолитной, а не модульной системой

Монолитная система — это система, в которой большинство функций интегрировано в единый модуль. Противоположностью такой системе является система, в которой функции распределены по нескольким уровням, причем каждый уровень имеет ограниченный доступ к другим уровням.

Хотя часть недостатков Windows — «наследство» ее исходной однопользовательской архитектуры, другие ее недостатки — прямое следствие обдуманных проектных решений, таких как монолитная архитектура (интегрирование большинства функций в ядро операционной системы). Компания Microsoft фактически вытеснила web-навигатор Netscape, интегрировав Internet Explorer в свою операционную систему так тесно, что не использовать IE стало практически невозможно. Нравится это пользователю или нет, но Internet Explorer вызывается при использовании справочной системы Windows, Outlook и многих других приложений как Microsoft, так и независимых производителей. Конечно, коммерческие интересы Microsoft требуют, чтобы использование каких-либо иных продуктов, кроме Internet Explorer, было крайне затруднительным. Microsoft успешно превращает конкурирующие продукты в ненужные, интегрируя в свою операционную систему все больше и больше сервисов, предоставляемых такими продуктами. Но в результате этого подхода получается монстр из сложным образом взаимодействующих сервисов (то есть, по определению, монолитная система).

Взаимозависимости такого рода имеют два неприятных каскадных побочных эффекта. Во-первых, в монолитной системе каждая брешь в какой-либо одной части системы проявляется во всех сервисах и приложениях, зависящих от этой части системы. Интегрировав Internet Explorer в операционную систему, Microsoft создала систе-

му, в которой любая брешь в Internet Explorer подвергает настольный компьютер с Windows угрозам, реализация которых может нарушить не только работу web-навигатора, но и весьма далеких от него объектов. Следовательно, одна-единственная брешь в Internet Explorer проявляется в бесчисленном множестве других приложений, многие из которых используют Internet Explorer неявным для пользователя образом, что дает этому пользователю ложное ощущение безопасности.

Такая архитектурная модель оказывает намного более глубокое воздействие, чем обычно представляют. Так, в монолитной системе уязвимости в защите оказываются более критичными, чем следовало ожидать.

Проиллюстрировать это поможет простая аналогия. Представим себе идеально устроенную операционную систему, которая состоит из трех сфер: одна — в центре; вторая, большего размера, охватывает первую; а третья сфера охватывает две внутренние. Конечный пользователь «видит» только внешнюю сферу. Это уровень, где пользователь запускает приложения, например, текстовые процессоры. Текстовые процессоры используют необходимые функции, предоставляемые второй сферой, например, средства визуализации графических изображений или форматирования текста. Эта вторая сфера (ее специалисты обычно называют «пользовательскими процессами») не имеет прямого доступа к критичным частям системы. Чтобы выполнить свою работу, она должна запросить разрешение от самой внутренней сферы. Внутренняя сфера выполняет самые важные функции, поэтому имеет непосредственный доступ ко всем критичным частям системы. Она управляет дисками, памятью и всем остальным. Эта сфера называется «ядро» и является сердцем операционной системы.

В случае вышеописанной архитектуры брешь в программах графического отображения не может нанести глобального ущерба компьютеру, потому что функции визуализации не имеют прямого доступа к наиболее критичным частям системы. Даже если пользователь загрузит в текстовый процессор изображение с внедренным вирусом, этот вирус не сможет повредить ничего, кроме собственных файлов пользователя, поскольку функция графического отображения находится вне центральной сферы и не имеет доступа ни к одной из критичных частей системы.

Проблема Windows заключается в том, что в ней не соблюдаются разумные конструкторские принципы разделения функций по соответствующим уровням, описанным выше. Windows вкладывает слишком много функций в ядро, центральную сферу, то есть туда, где можно причинить самый большой ущерб. Например, если интегрировать функции графического отображения в центральную сферу (ядро), эти функции получат возможность повредить всю систему. Таким образом, как только обнаружится брешь в алгоритме графического отображения, чрезмерно интегрированная архитектура Windows облегчит использование этой бреши для получения полного контроля над системой или для разрушения всей системы.

Наконец, монолитная система нестабильна по своей природе. Когда в системе так много взаимосвязей, изменение одной из частей этой системы порождает множество угроз. Единственное изменение в системе может оказать (и обычно оказывает) каскадное воздействие на все сервисы и приложения, которые зависят от этой части системы. Именно поэтому пользователи Windows трепещут при мысли об установке программных коррекций и обновлений. Обновления, исправляющие одну часть Windows, часто нарушают работу других сервисов и приложений. В качестве иллюстрации можно привести следующий факт: для пакета обновлений Windows XP service pack 2 уже составлен постоянно растущий список случаев, когда его установка привела к выходу из строя сторонних приложений. Это естественное явление в монолитной системе — любое изменение в одной части механизма влияет на весь механизм и на все приложения, которые зависят от этого механизма.

В Windows слишком широко используется RPC-механизм

Аббревиатура RPC означает «удаленный вызов процедуры» (Remote Procedure Call). RPC — это то, что происходит, когда одна программа отправляет через сеть указание другой программе выполнить какое-либо действие. Например, одна программа может использовать RPC, чтобы дать другой программе указание рассчитать среднюю стоимость чая в Китае и вернуть результат. **Удаленным** вызовом процедуры этот механизм называется потому, что не имеет значения, функционирует ли «другая программа» на том же компьютере, на соседнем, или где-то в Интернете.

RPC-механизмы — это потенциальная угроза безопасности, поскольку их предназначение — позволить компьютерам, находящимся где-то в сети, давать данному компьютеру указания выполнить те или иные действия. Как только обнаруживается брешь в программе, разрешающей использование RPC-механизма, у любого, кто располага-

ет подключенным к сети компьютером, появляется возможность использовать эту брешь, чтобы заставить уязвимый компьютер выполнить какие-либо действия. К сожалению, пользователи Windows не могут заблокировать RPC-механизм, так как Windows использует его, даже если компьютер не подключен к сети. Многие сервисы Windows устроены именно так. В некоторых случаях можно блокировать RPC-порт на межсетевом экране, но Windows так широко использует RPC-механизмы в основных функциях, что подобная блокировка не всегда возможна. Удивительно, но некоторые из наиболее серьезных уязвимостей в Windows Server 2003 (см. Табл. 1) — следствие брешей в самих RPC-функциях Windows, а не в приложениях, которые их используют. Самый распространенный способ использовать уязвимость, связанную с RPC-механизмом — атаковать сервис, использующий RPC, а не сам RPC-механизм.

Важно отметить, что RPC-механизмы не всегда необходимы, отчего становится еще непонятнее, почему Microsoft так широко их использует. Предположим, требуется создать web-сайт, используя два сервера. Один сервер будет работать в качестве сервера базы данных, второй — в качестве web-сервера. В этом случае серверу базы данных необходимо использовать RPC, потому что web-сервер находится на отдельном компьютере и должен иметь возможность доступа к серверу базы данных через сетевое подключение. (Даже в этом случае следует сконфигурировать сервер базы данных так, чтобы он «слушал» только данный web-сервер, но не другие компьютеры). Если же и сервер базы данных, и web-сервер функционируют на одном компьютере, использование RPC-механизмов на сервере базы данных не только не обязательно, но и нежелательно. Web-сервер должен иметь прямой доступ к серверу базы данных, потому что они оба функционируют на одном компьютере. Ни технических, ни логических причин подключать к сети сервер базы данных нет, поскольку такое подключение создает лишнюю угрозу безопасности.

Вопрос о серверах баз данных поднят из-за того, что сетевой червь Slammer, один из самых опасных червей, когда-либо существовавших в Интернете, использовал на редкость неуместное применение RPC-подобных сетевых соединений, реализованное Microsoft. За короткое время Slammer заразил так много систем, что Интернет практически перестал функционировать.

Сетевой червь Slammer вызвал хаос, использовав две бреши в Microsoft SQL Server, который является сервером клиент-серверной базы данных SQL-типа. Одна брешь — это самая бесполезная функция Microsoft SQL Server, позволяющая одновременно запустить несколько копий сервера базы данных на одном компьютере. Почему эта функция бесполезна? Если вы незнакомы с работой серверов баз данных, представьте это себе следующим образом. В обычных условиях бессмысленно запускать несколько копий сервера базы данных на одном компьютере, потому что одна копия — это все, что нужно, даже если ее использует множество разных приложений. Потребность в одновременном запуске нескольких серверов баз данных на одном компьютере также вероятна, как потребность в одновременном запуске двух копий Windows XP на одном компьютере. Несколько копий сервера базы данных запускаются не по ошибке исключительно редко, да и то лишь в высокопроизводительных приложениях или для тестирования и разработки[4].

Простой способ обеспечить одновременную работу нескольких не мешающих друг другу копий SQL Server — создать RPC-механизм, который сортирует запросы на получение данных таким образом, что, например, приложение-факс

[4] Кажется, мы поняли, почему Microsoft решила установить по умолчанию именно этот режим функционирования SQL Server. Многие сторонние приложения используют процессор SQL Server по умолчанию. Если бы на компьютере могла функционировать только одна копия SQL Server, то Microsoft пришлось бы разработать удобные средства, позволяющие программе инсталляции обнаружить установленный и функционирующий SQL Server, а затем обеспечить удобный способ инсталляции, интеграции и администрирования особых требований сторонних приложений в своей собственной базе данных и в таблицах, функционирующих на данном сервере. Это очень элегантное решение, минимизирующее используемые ресурсы, поскольку всегда требуется только одна копия SQL Server. Но такой подход потребовал бы большой дополнительной работы со стороны Microsoft или со стороны независимых разработчиков. Намного проще реализовать решение, позволяющее сторонним приложениям не заботиться о том, инсталлирован ли SQL Server. По схеме, реализованной Microsoft, любое стороннее приложение может просто инсталлировать свою собственную копию SQL Server, не беспокоясь о том, установлен ли SQL Server на данном компьютере, какая версия SQL Server инсталлирована, как сконфигурирован существующий SQL Server. Стремясь привлечь независимых разработчиков к использованию SQL Server, Microsoft выбрала принцип наименьшего действия и разработала систему, в которой любое приложение может инсталлировать свою собственную копию SQL Server, которая не взаимодействует с другими копиями SQL Server, функционирующими на том же компьютере. Из-за этого возникла необходимость в запуске несколько копий SQL Server с задействованным RPC-механизмом, который следовало бы использовать как можно реже. Этот наименее затратный подход имел чрезвычайно пагубные последствия. Если бы Microsoft разработала SQL Server так, чтобы он функционировал как единственная копия, не подключенная по умолчанию к сети, то сетевой червь Slammer не нашел бы достаточно компьютеров с работающим SQL Server, чтобы причинить сколько-нибудь значительный ущерб.

запрашивает одну копию SQL Server, а приложение-ежедневник — другую. Ситуация усложняется тем, что и средства разработки Microsoft поддерживают тот же принцип монолитности, который используется в продуктах Microsoft. Поэтому самые разные приложения — программы планирования времени, программное обеспечение для факсимильной связи, системы управления проектами — почти 200 приложений, многие из которых предназначены для настольных систем, используют неоправданно уязвимый процессор SQL Server. В результате сотни тысяч, если не миллионы, людей используют **настольные** приложения, зависящие от процессора SQL Server с его многочисленными разрешенными сетевыми сервисами, немалая часть которых уязвима к атакам из Интернета. Вряд ли можно придумать лучший способ устроить катастрофу.

В результате Slammer смог атаковать огромное количество компьютеров, потому что на всех процессорах SQL Server использование этих функциональных возможностей разрешено по умолчанию. Хотя SQL Server еще не интегрирован в Windows, его повсеместное использование в различных приложениях — от программного обеспечения для факсимильной связи до программ планирования времени — фактически превращает его в часть более крупной монолитной системы, что открывает возможность такой организации атаки, которая характерна для монолитной системы. К сожалению, весьма вероятно, что SQL Server будет тесно интегрирован в File System WinFS, новую перспективную файловую систему Windows, первоначально предназначавшуюся для Longhorn, операционной системы нового поколения. Горячим сторонникам идеи интеграции SQL Server в операционную систему следует помнить об истории с сетевым червем Slammer.

Windows фокусируется на знакомом графическом интерфейсе для настольных компьютеров

Компания Microsoft считает знакомый интерфейс Windows главным аргументом[5] в пользу перехода на Windows Server 2003. Цитата с web-сайта компании Microsoft: **«Знакомый интерфейс Windows облегчает использование Windows Server 2003. Новые удобные мастера упрощают установку специальных ролей и выполнение обычных задач управления сервером...»**

Пропагандируя такое использование, Microsoft предлагает администраторам работать с ОС Windows Server 2003 на самом сервере, зарегистрировавшись с полномочиями администратора. В результате администратор Windows становится наиболее уязвимым к брешам в системе защиты, поскольку использование уязвимых программ, таких как Internet Explorer, создает угрозы безопасности сервера.

Архитектура Linux

По данным обзора Summer 2004 Evans Data Linux Developers Survey, 93% разработчиков Linux сталкивались не более чем с двумя случаями компрометации компьютера с ОС Linux. 87% встретился только один подобный инцидент, а в практике 78% не было ситуаций, когда компьютер под управлением Linux сколько-нибудь серьезно пострадал от действий злоумышленников. В тех редких случаях, когда им удалось добиться успеха, основной причиной были ненадлежащим образом сконфигурированные настройки безопасности.

Однако для данной статьи важнее тот приведенный в обзоре факт, что 92% респондентов никогда не сталкивались со случаями заражения ОС Linux вирусами, троянскими и другими вредоносными программами.

То обстоятельство, что вирусы, троянские и другие вредоносные программы редко могут (если вообще могут) заразить Linux-системы, частично можно объяснить следующими причинами:

1. Linux имеет долгую историю использования тщательно проработанной многопользовательской архитектуры.
2. По своей архитектуре Linux является в основном модульной системой.
3. Функционирование Linux не зависит от RPC-механизма, а сервисы обычно по умолчанию настроены не использовать RPC-механизм.
4. Серверы Linux идеально подходят для удаленного администрирования.

При дальнейшем чтении следует помнить о вариациях в используемых по умолчанию конфигурациях различных дистрибутивов ОС Linux, поэтому то, что верно для Red Hat Linux, может оказаться неправильным для Debian, и еще больше отличий может быть в SuSE. По большей части в том,

[5] Top 10 Benefits of Windows Server 2003 — http://www.microsoft.com/windowsserver2003/evaluation/whyupgrade/top10best.mspx (На русском языке: Десять веских оснований для перехода на Windows Server 2003 — http://www.microsoft.com/rus/windowsserver2003/whyupgrade/top10best.mspx)

что касается конфигураций по умолчанию, все основные дистрибутивы Linux следуют одним и тем же разумным правилам.

Linux имеет долгую историю использования тщательно проработанной многопользовательской архитектуры

Linux никогда не была однопользовательской операционной системой. Поэтому в ней с самого начала заложен принцип изолирования пользователей от приложений, файлов и каталогов, воздействующих на операционную систему в целом. Каждому пользователю предоставляется пользовательский каталог, в котором хранятся все его файлы данных и файлы конфигурации, принадлежащие этому пользователю. Когда пользователь запускает какое-либо приложение (например, текстовый процессор), оно запускается с ограниченными полномочиями данного пользователя. Запущенное пользователем приложение имеет право на запись только в собственный каталог этого пользователя. Оно не может ничего записать в системный файл или даже в каталог другого пользователя, если только администратор явным образом не предоставит данному пользователю такое право.

Еще важнее, что Linux предоставляет почти все функциональные возможности (например, визуализацию изображений в формате JPEG) в виде модульных библиотек. Поэтому, когда текстовый процессор отображает JPEG-изображения, соответствующие функции запускаются с теми же ограниченными полномочиями, что и сам текстовый процессор. Если в программах визуализации JPEG-изображений имеется брешь, злоумышленник сможет использовать ее только для получения таких же полномочий, как у данного пользователя, что значительно ограничивает масштабы возможного ущерба. В этом преимущество модульных систем, и эти системы ближе к идеалу операционной системы, описанному выше как конструкция из трех сфер (см. раздел «Windows по своей архитектуре является монолитной, а не модульной системой»).

Ограничения по умолчанию — свойство модульной архитектуры Linux; едва ли возможно отправить по электронной почте пользователю Linux такое сообщение, которое заразит вирусом весь компьютер. Не имеет значения, насколько неудачно сконструирован почтовый клиент или как именно он прореагирует на вирус — полномочия, установленные для клиента, позволят ему заразить или повредить только файлы своего пользователя. Web-навигаторы, работающие в ОС Linux, не поддерживают такие небезопасные по своей природе объекты, как элементы управления ActiveX, но даже если бы они поддерживались, вредоносный элемент ActiveX смог бы запуститься только с полномочиями того пользователя, который запустил web-навигатор. И в этом случае самый большой вред, который он смог бы принести — это заразить или удалить собственные файлы пользователя.

Даже сервисы, например, web-серверы, обычно запускаются как пользователи с ограниченными полномочиями. Так, Debian GNU/Linux запускает web-сервер Apache как пользователя «www-data», принадлежащего к группе с тем же именем «www-data». Если злоумышленник на компьютере с Debian получит полный контроль над web-сервером Apache, он сможет воздействовать только на файлы, принадлежащие пользователю «www-data», то есть на web-страницы. В свою очередь, MySQL, сервер базы данных SQL-типа, часто используемый вместе с Apache, запускается с полномочиями пользователя «mysql». Даже если Apache и MySQL вместе обслуживают web-страницы, злоумышленник, получивший контроль над Apache, не будет иметь полномочий, позволяющих использовать уязвимость в Apache для получения контроля над сервером базы данных, потому что сервер базы данных «принадлежит» другому пользователю.

Кроме того, пользователи, ассоциированные с такими сервисами, как Apache, MySQL и т.д., часто устанавливаются с учетными записями, не имеющими доступа к командной строке. Поэтому, если злоумышленник сможет получить права учетной записи пользователя MySQL, он не сможет воспользоваться этой уязвимостью для того, чтобы дать произвольные команды на сервер Linux, поскольку данная учетная запись не может вызывать команды.

Напротив, ОС Windows изначально сконструирована таким образом, что всем пользователям и приложениям предоставляется административный доступ ко всем файлам на компьютере. Только постепенно Windows была переработана так, чтобы изолировать пользователей и их действия от остальной системы. Операционная система Windows Server 2003 близка к достижению этой цели, но метод, избранный Microsoft для создания барьера между пользователем и системой, все тот же: широкое использование постоянно меняющихся корректировок существующей «конструкции» вместо ее фундаментальной переработки на основе новой концепции, в которой во главу угла поставлены реализация многопользовательских возможностей и безопасность.

По своей архитектуре Linux является модульной, а не монолитной системой

Linux — это операционная система, сконструированная, в основном, по модульному принципу, от ядра (центрального «мозга» Linux) до приложений. В Linux практически нет нерасторжимых связей между какими-либо компонентами. Не существует единственного процессора web-навигатора, используемого справочными системами или программами электронной почты. В самом деле, нетрудно сконфигурировать большинство программ электронной почты так, чтобы использовать встроенный web-навигатор для отображения HTML-сообщений либо запускать любой нужный web-навигатор для просмотра HTML-документов или перехода по ссылкам, приведенным в тексте сообщения. Следовательно, брешь в одном процессоре web-навигатора необязательно представляет опасность для каких-либо других приложений на данном компьютере, так как почти никакие другие приложения, кроме самого web-навигатора, не зависят от единственного процессора web-навигатора.

Не все в Linux является модульным. Две наиболее популярные графические среды, KDE и GNOME, в каком-то смысле монолитны по своей архитектуре. По крайней мере, монолитны настолько, что в принципе обновление одной части GNOME или KDE может нарушить работу других частей GNOME или KDE. Но и GNOME, и KDE не до такой степени монолитны, чтобы требовалось использование приложений, разработанных специально для GNOME или KDE. Приложения GNOME или любые другие приложения можно запускать под KDE, а KDE или любые другие приложения — под GNOME.

Ядро Linux поддерживает модульные драйверы, но в значительной мере является монолитным ядром, потому что сервисы в этом ядре взаимозависимы. Все отрицательные последствия монолитности минимизируются тем, что ядро Linux, насколько это возможно, разработано как наименьшая часть системы. Linux почти фанатично придерживается следующего принципа: «Если задача может быть выполнена вне ядра, она должна быть выполнена вне ядра». Это означает, что в Linux почти каждая полезная функция («полезная» означает «воспринимаемая конечным пользователем») не имеет доступа к уязвимым частям системы Linux.

Напротив, ошибки в драйверах графических адаптеров являются частой причиной «синего экрана смерти» в Windows. Это происходит из-за того, что Windows интегрирует графику в ядро, где подобная ошибка может вызвать отказ системы. Не считая нескольких известных исключений (например, коммерческий драйвер графики NVidia), Linux заставляет все графические драйверы функционировать вне ядра. Ошибка в графическом драйвере может вызвать сбой в графическом приложении, но не может вызвать отказ всей системы. В случае такой ошибки достаточно просто перезапустить графическое приложение. Никакой перезагрузки компьютера не требуется.

Linux не зависит от RPC-механизма

Как уже говорилось в разделе о Windows, аббревиатурой RPC обозначается удаленный вызов процедуры (Remote Procedure Call). RPC-механизм позволяет одной программе дать указание второй программе выполнить какое-либо действие, даже если «вторая программа» работает на другом компьютере. Первая программа при помощи RPC отправляет другой программе указание выполнить определенные расчеты и вернуть результат. Напомним, что **удаленным** вызовом процедуры этот механизм называется потому, что не имеет значения, функционирует ли «другая программа» на том же компьютере, на другом компьютере, стоящем в соседней комнате, или где-то в Интернете.

В большинстве дистрибутивов Linux программы инсталлируются так, что по умолчанию доступ в сеть отключен. Например, MySQL, сервер базы данных SQL-типа, обычно инсталлируется так, что он не ожидает инструкций из сети. Если создать web-сайт, установив Apache и MySQL на одном и том же компьютере, то для взаимодействия Apache и MySQL не требуется, чтобы MySQL прослушивал сеть. Напротив, SQL Server прослушивает сеть независимо от того, необходимо ли это. Если требуется, чтобы MySQL слушал сеть, эту функцию следует включить вручную, а затем в явном виде определить пользователей и компьютеры, которым разрешен доступ к MySQL.

Даже если Linux-приложения по умолчанию используют сеть, они чаще всего сконфигурированы так, что могут отвечать только локальному компьютеру и игнорируют любые запросы других компьютеров в сети.

В отличие от Windows Server 2003, компьютер с операционной системой Linux останется полностью работоспособным, даже если на нем заблокировать практически все RPC-сервисы, связанные с использованием сети.

Серверы Linux идеально подходят для удаленного администрирования

Сервер Linux можно, а часто и нужно, инсталлировать без подключенного монитора и администрировать удаленно. Очень часто такой тип инсталляции идеален для серверов, поскольку при удаленном администрировании сервер не подвергается таким угрозам, как при локальном администрировании.

Например, можно войти в систему на своем настольном компьютере в качестве обычного пользователя с ограниченными полномочиями и администрировать сервер под управлением ОС Linux через административный web-интерфейс. Даже самая критическая уязвимость в защите web-навигатора сможет воздействовать только на локальную учетную запись пользователя на его настольном компьютере, не затронув сервер.

Возможно, это одно из самых главных отличий Linux от Windows, потому что этот фактор сводит на нет многие критические уязвимости, общие для Linux и Windows, например, уязвимости web-навигаторов Mozilla и Internet Explorer.

Реальные показатели безопасности и серьезности

Чтобы надлежащим образом оценить риски, связанные с выбором операционной системы для какой-либо определенной задачи, необходимо рассмотреть множество показателей. Иногда они действуют кумулятивно, в других случаях — компенсируют друг друга.

Существуют три очень важных фактора риска, которые существенно зависят друг от друга. Комбинация этих трех факторов решающим образом влияет на общую серьезность той или иной бреши в системе защиты. Такими элементами показателя общей серьезности являются: **возможный ущерб, возможность использования и возможная доступность**.

Элементы показателя общей серьезности

Возможный ущерб любой обнаруженной уязвимости показывает, какой вред может быть нанесен в результате ее использования злоумышленником. Уязвимость, позволяющая раскрыть все пароли администраторов, имеет высокий показатель возможного ущерба. Этот показатель для бреши, в результате использования которой начинает мерцать экран, обычно намного ниже, он возрастает только в том случае, если это конкретное повреждение трудно исправить.

Возможность использования показывает, насколько просто или сложно использовать данную уязвимость, потребуется ли высокая квалификация в программировании или эту уязвимость сможет использовать любой человек с самыми элементарными знаниями в указанной области.

Возможная доступность показывает, какой уровень доступа необходим для использования данной уязвимости. Если любой начинающий хакер (тот, кого обычно называют «script kiddies») из Интернета может использовать брешь на сервере, защищенном межсетевым экраном, то эта брешь имеет очень высокую возможную доступность. Если же использовать брешь сможет только сотрудник компании, имеющий действующие регистрационное имя и пароль, причем исключительно с компьютера, находящегося в здании компании, то возможная доступность этой бреши значительно меньше.

Показатель общей серьезности и взаимосвязь между тремя факторами риска

Один или несколько из вышеперечисленных факторов риска могут оказать решающее воздействие на общую серьезность ошибки. Рассмотрим следующую ситуацию. Руководитель информационной службы компании, занимающейся электронной коммерцией через web-сайт, узнает от аналитика по безопасности, что обнаружена брешь в операционной системе, под управлением которой функционируют серверы компании. Злоумышленник может использовать эту брешь, чтобы удалить всю информацию с дисков на всех серверах, используемых компанией.

Возможный ущерб — последствия от этой бреши катастрофичны.

Хуже того, аналитик добавляет, что с технической точки зрения эту брешь использовать тривиально просто. **Возможность использования** имеет критический уровень.

Пора нажимать аварийную кнопку, не так ли? Но предположим, что аналитик добавляет еще несколько слов. Использовать эту брешь может только тот, у кого есть ключ от серверной комнаты, потому что эта данная уязвимость требует физического доступа к компьютерам. Этот единственный *ключевой* показатель, простите за каламбур, радикально изменяет общую серьезность угрозы, порожденной данной конкретной брешью. Крайне низкая **возможная доступность** переводит стрелку на шкале серьезности с «тревога!» на «под контролем».

Наоборот, другая уязвимость может быть доступна любому начинающему хакеру в Интернете, но по-прежнему будет иметь незначительную серьезность, если **возможный ущерб** от этой бреши является несущественным.

Возможно, теперь более понятно, почему обманчива, если не совершенно безответственна, методика оценки безопасности по одному-единственному показателю, например, по количеству предупреждений об уязвимостях. В самом крайнем случае следует рассмотреть также указанные три фактора риска. Какая операционная система заслуживает больше доверия — с сотней брешей пренебрежимо малой серьезности или же с десятком брешей чрезвычайно высокой серьезности (использование бреши ведет к катастрофе)? Если при оценке не учитывать общую серьезность брешей, то подсчет их количества в лучшем случае не имеет никакого значения, в худшем — вводит в заблуждение.

Исключение из правила

Показатель общей серьезности имеет три упомянутых «главных» компонента. Выше же было показано, как низкий **возможный ущерб** или низкая **возможная доступность** могут практически свести на нет другие факторы риска, какими бы высокими они ни были. **Возможность использования** — исключение из этого правила. Брешь, для использования которой требуется высококвалифицированный специалист, значительно меньше компенсирует высокий показатель **возможного ущерба** или **возможной доступности**.

Объясняется это просто. Если для использования бреши необходимо попасть в комнату с компьютерами, то дело не только в том, что это трудно, но и в том, что любая попытка проникнуть в эту комнату увеличивает для злоумышленника риск *быть пойманным*. И именно поэтому брешь, которую может использовать только сотрудник компании, менее серьезна, чем брешь, которую может использовать любой начинающий хакер из Интернета. В первом случае риск быть пойманным намного выше, чем во втором.

С другой стороны, анонимные злоумышленники-программисты весьма заурядной квалификации могут в течение нескольких недель или месяцев разрабатывать программу, позволяющую использовать какую-либо брешь в защите, практически не рискуя при этом быть пойманными. Единственная важная задача, стоящая перед таким злоумышленником, — активировать вредоносную программу так, чтобы невозможно было отследить ее автора.

Уже поверхностное знакомство с современным состоянием вредоносного программного обеспечения показывает самоочевидность этого исключения. Мало кто воспользуется базукой, чтобы проложить себе путь в машинный зал и «взломать» находящиеся там серверы. Но существует бесчисленное множество троянских программ, сетевых червей и вирусов, которые попрежнему заражают множество компьютеров, и одна из причин в том, что программисты, талантливые и не очень, считают признаком профессионализма умение преодолеть технические трудности написания вредоносного кода или переработки вредоносного кода, написанного другими. Очевидно, что технические трудности необязательно компенсируют высокую опасность бреши, вызванную другими причинами.

Применение показателя общей серьезности

Только оценив общую серьезность конкретной бреши, можно перейти к осмыслению таких показателей, как «сколько предупреждений об уязвимости существует для Windows по сравнению с Linux» или «сколько времени проходит между обнаружением и исправлением ошибки в случае Windows и в случае Linux».

Предположим, для одной операционной системы зарегистрировано намного больше предупреждений об уязвимостях, чем для другой. Этот показатель имеет смысл только в одном случае — если для данной системы имеется также больше предупреждений об уязвимостях **с** *высоким уровнем общей серьезности*. Одно дело, если довольно часто случаются разные мелкие неприятности, не представляющие практически никакой опасности, и совсем другое — когда регулярно обнаруживаются пусть и немногочисленные, но такие бреши, которые ставят под удар всю компанию.

Предположим, что для некоторой операционной системы зафиксировано более короткое

время между обнаружением бреши и выходом соответствующей программной коррекции. И в этом случае показатель имеет смысл только тогда, *когда это время относится к брешам с высокой общей серьезностью*. Одно дело — несколько месяцев ждать коррекции для бреши, использование которой может нанести незначительный ущерб небольшому количеству компьютеров или даже совсем не причинить никакого вреда. И совсем другое дело — ожидать месяцами коррекции для бреши, использование которой может поставить под удар всю компанию.

Способы оценки показателей

Возможная доступность

Данный показатель учитывает меры, которые необходимо принять для получения доступа к компьютеру с целью использовать уязвимости в системе защиты. Обычно такие меры попадают в одну из перечисленных ниже категорий. На практике реальный порядок этих категорий может меняться, но приведенный список может оказаться удобным ориентиром. Кроме того, следует заметить, что существует несколько экзотических сложных случаев, которые в приведенном списке не рассматриваются. Например, закрытая программной коррекцией брешь в Windows Server 2003 сама по себе была не очень доступна, но она позволяла злоумышленнику сделать систему незащищенной перед серьезными угрозами. То есть это было одно из звеньев в цепочке уязвимостей. Категории в списке перечислены в порядке возрастания их серьезности.

1. Необходим физический доступ к компьютеру, но не требуется наличия действующей учетной записи пользователя.
2. Необходим физический доступ к компьютеру и требуется действующая учетная запись пользователя.
3. Необходима действующая учетная запись пользователя, но не нужен физический доступ к атакуемому компьютеру. Достаточно доступа по локальной сети (из корпоративной сети компании).
4. Необходима действующая учетная запись пользователя, но не нужен физический доступ к атакуемому компьютеру. Атакуемый компьютер доступен через Интернет с удаленного компьютера.
5. Можно использовать брешь удаленно, через Интернет, не имея действующей учетной записи пользователя на атакуемом компьютере, но невозможно достичь бреши напрямую. Существует еще один барьер, например, маршрутизатор или межсетевой экран. Для этой категории трудно найти надлежащее место при перечислении по уровню серьезности, поскольку правильно сконфигурированный межсетевой экран может обеспечить стопроцентную защиту, но не всегда. Плохо сконфигурированный межсетевой экран может вообще не обеспечивать никакой защиты.
6. Можно использовать брешь удаленно, через Интернет, не имея действующей учетной записи пользователя на атакуемом компьютере, но невозможно достичь бреши напрямую. Существует еще один, более трудный для преодоления барьер. Этим барьером может быть другая программа (например, брешь существует в Microsoft SQL Server, но для ее использования необходимо внедрить элемент управления ActiveX или Javascript в web-страницу, доступную через Microsoft Internet Information Server). В некоторых случаях для получения непрямого доступа необходимо вовлечь в этот процесс пользователя. Например, придется разослать по электронной почте сообщения, которые направят пользователей на web-страницу, содержащую вредоносный элемент управления или код. Широко используется способ, когда пользователю предлагается открыть файл, вложенный в электронное письмо. Серьезность этой категории меняется в зависимости от того, насколько искусно это вовлечение замаскировано под невинное действие.
7. Использовать брешь можно удаленно, через Интернет, не имея действующей учетной записи пользователя на атакуемом компьютере, но невозможно достичь бреши напрямую. Тем не менее, брешь используется косвенно, но автоматически. Например, брешь в операционной системе Windows используется немедленно и автоматически, как только пользователь открывает электронное письмо с помощью программы Outlook.
8. Использовать брешь можно удаленно, через Интернет, просто отправив через сеть информацию на атакуемый компьютер. Например, уязвимость типа «отказ в обслуживании» (DoS) можно использовать просто отправив специальные сетевые пакеты на атакуемый web-сайт, что сделает этот web-сайт недоступным для других пользователей Интернета.

Возможность использования

Этот показатель учитывает технические трудности, которые необходимо преодолеть, чтобы использовать брешь в защите. Обычно эти трудности попадают в одну из перечисленных ниже категорий. Категории перечислены в порядке возрастания их серьезности (на практике реальный порядок этих категорий может меняться, но приведенный список может оказаться удобным ориентиром).

1. Брешь существует, но еще не обнаружена. Для использования этой бреши необходимы либо исключительная компетентность, либо счастливый случай.
2. Для использования бреши необходимы высокая квалификация в программировании и глубокое знание операционной системы, но о существовании этой бреши известно недостаточно широко, из-за чего маловероятно, что ее используют многие нарушители.
3. О существовании бреши известно, и для ее использования необходимы высокая квалификация в программировании и глубокое понимание того, как функционируют атакуемые программное обеспечение и операционная система.
4. Для использования бреши необходима высокая квалификация в программировании, но уже создан вирус, троянская программа или сетевой червь, которые могут служить основой для атаки. Программисту требуется только модифицировать этот код, чтобы использовать новую брешь или сделать данный вирус более опасным.
5. Для создания кода, использующего брешь, необходима высокая квалификация в программировании, но подходящий код уже существует, и достаточно средней квалификации в программировании, чтобы усовершенствовать или модифицировать этот код так, что он будет использовать существующую брешь или «будущие» бреши.
6. Для использования бреши достаточно средней или начальной квалификации в программировании либо же элементарного знания компьютера.
7. Не имеет значения, насколько трудно использовать брешь, поскольку вся работа по созданию средств для использования этой бреши уже проделана, а вредоносный код сделан общедоступным для его применения новичками.
8. Кто угодно может использовать брешь, введя простой текст в командной строке или указав URL в web-навигаторе.

Возможный ущерб

Оценить этот показатель труднее всего. Требуется определить, по крайней мере, два различных набора категорий. Во-первых, надо учесть, какой вред причинит использование бреши приложениям или компьютерам. Во-вторых, необходимо оценить возможный ущерб с точки зрения последствий для всей компании. Например, в ситуации, когда брешь позволяет нарушителю прочитать неопубликованные web-страницы. Вред от этого незначителен, если на компьютере не хранится конфиденциальная информация. Но если неопубликованная web-страница содержит какую-либо конфиденциальную информацию, например, номера кредитных карт, то общий возможный ущерб может быть очень большим, при том что возможный технический ущерб минимален. Ниже приводятся (в порядке возрастания серьезности) наиболее важные факторы, которые следует учитывать при оценке возможного технического ущерба в результате использования какой-либо конкретной бреши.

1. Брешь влияет только на производительность другого компьютера, но не настолько, чтобы он перестал отвечать на запросы.
2. Брешь влияет только на собственные программы или файлы нарушителя, но не затрагивает файлы или программы других пользователей.
3. Брешь делает незащищенной информацию в файлах других пользователей, но не учетную запись администратора или системные файлы.
4. Брешь позволяет нарушителю просматривать, изменять или удалять пользовательские файлы. Она не дает возможности просматривать, изменять или удалять файлы администратора или системные файлы.
5. Брешь позволяет нарушителю просматривать критичную информацию либо путем исследования сетевого трафика, либо путем доступа с правами «только чтение» к файлам администратора или к системным файлам.
6. Брешь позволяет нарушителю получить некоторые, но не все, полномочия административного уровня, возможно, в ограниченном окружении.
7. Брешь позволяет нарушителю вызвать отказ системы или каким-то иным путем заставить ее не отвечать на обычные запросы. Это — типичная атака «отказ в обслуживании». Однако нарушитель фактически не может получить контроль над компьютером, за исключением того, что компьютер перестанет отвечать на запросы.

8. Брешь позволяет нарушителю изменить или удалить все привилегированные файлы и информацию. Нарушитель может получить полный контроль над атакуемой системой и фактически причинить такой же ущерб, какой способен нанести полностью авторизованный системный администратор.

Уровень общей серьезности

С учетом описанных выше трех факторов общая серьезность угроз может изменяться от Минимальная до Катастрофическая. Рассмотреть все перестановки не представляется возможным, но несколько примеров могут оказаться полезными. Эти примеры основываются на категориях возможного ущерба в сочетании с различными категориями возможной доступности и возможности использования.

1. Если анонимный злоумышленник из Интернета сумеет снизить производительность компьютеров компании, то оценка последствий может варьироваться от «мелкая неприятность» до «разрушительный удар по финансам» в зависимости от того, насколько критичной является производительность системы для деятельности компании.
2. Атака на собственную учетную запись является бессмысленной, но такая саморазрушительная деятельность может создать для ИТ-подразделения лишнюю работу по восстановлению.
3. Потенциальная серьезность бреши, позволяющей просматривать только файлы работающего в том же здании сослуживца, минимальна, даже если эту брешь тривиально просто использовать. Уровень серьезности возрастает, когда файлы этого сослуживца содержат конфиденциальную информацию, и уменьшается, если велика вероятность поимки нарушителя. С другой стороны, если просматривать конфиденциальные файлы в сети компании может любой злоумышленник из Интернета (высокая возможная доступность), общая серьезность бреши резко возрастает.
4. Если брешь позволяет нарушителю изменять или удалять файлы работающего в том же здании сослуживца, серьезность этой бреши тем меньше, чем регулярнее выполняется в компании резервное копирование и чем больше вероятность поимки нарушителя. Если нарушитель может изменять файлы учетной записи пользователя на удаленном компьютере, то серьезность меняется в зависимости от важности этой учетной записи и предоставляемого ею сервиса. Например, серьезность может варьироваться от незначительного искажения web-страниц до их полного удаления.

Дополнительные соображения

Дисбаланс приложений

В дебатах о превосходстве одной операционной системы над другой часто упускается из виду, что уязвимости в системе защиты почти всегда коренятся в приложениях. Данное обстоятельство создает определенные трудности при сравнении Windows и Linux, потому что эти две системы находятся в неравном положении с точки зрения переносимости и доступности приложений.

С одной стороны, большинство популярных приложений для Microsoft Windows составляют приложения Microsoft, и они запускаются только под Windows. Когда обнаруживается брешь в Microsoft Exchange, можно обоснованно полагать, что эта проблема затронет только клиентов Windows. Microsoft Exchange не функционирует ни в Linux, ни в Solaris, ни где-либо еще, кроме Windows.

С другой стороны, web-сервер Apache чаще всего ассоциируется с Linux, UNIX или другими UNIX-системами, но Apache функционирует и под управлением Windows. Поэтому, сравнивая общую безопасность Windows и Linux, можно ли считать, что брешь в Apache — это дефект только Linux? Или она негативно сказывается и на Linux, и на Windows?

Еще больше усложняет проблему то, что известно несколько случаев, когда брешь в Apache почти (или даже совсем) не опасна для Linux, но является серьезной уязвимостью в Windows. Обратная ситуация встречается крайне редко, если вообще встречается. Следует ли понижать оценку общей безопасности Windows потому, что эта система более подвержена неблагоприятному воздействию, чем Linux, при использовании программного обеспечения, которое обычно ассоциируется с Linux?

Вопрос о том, учтен ли какой-либо из этих факторов при сравнении общей безопасности Windows и Linux, ни в коем случае нельзя оставлять без внимания.

Настройка и администрирование

Наконец, различие в подходе к настройке и администрированию серверов в Linux и в Windows, как уже отмечалось, возможно, является самым важным различием между этими двумя операционными системами.

Windows предлагает использовать знакомый интерфейс, что означает администрирование ОС Windows Server 2003 на самом сервере. Linux не полагается на локальное использование графического интерфейса и не поощряет этого отчасти потому, что функционирование графической среды на сервере — это необоснованная трата ресурсов, а отчасти потому, что от этого возрастают угрозы безопасности сервера. Так, любой сервер, предлагающий использовать графический интерфейс на компьютере сервера, предлагает также выполнять на сервере похожие операции, например, использовать web-навигатор. В результате сервер подвергается угрозам, порожденным уязвимостями в защите web-навигатора. Любой сервер, побуждающий пользователя к удаленному администрированию, защищен от подобных угроз. Если администрирование сервера Linux осуществляется удаленно, через учетную запись пользователя на рабочей станции, то брешь в web-навигаторе создает угрозу только этой удаленной рабочей станции, а не серверу. Именно поэтому брешь в защите web-навигатора потенциально опаснее для Windows Server 2003, чем для Red Hat Enterprise Server AS.

Сравнение 40 последних программных коррекций системы защиты

В последующих разделах приведена информация о 40 самых последних программных коррекциях уязвимостей в защите ОС Windows Server 2003 (как утверждается, самой безопасной версии Windows) и Linux Red Hat Enterprise AS v.3 (как утверждается, конкурента Windows Server 2003). Данные по программным коррекциям и уязвимостям для Windows Server 2003 взяты непосредственно с web-сайта компании Microsoft, а для Red Hat Enterprise AS v.3 — с web-сайта Red Hat.

В операционной системе Windows Server 2003 были самые серьезные бреши в системе защиты. По собственной классификации Microsoft, 38% исправленных уязвимостей оценены как Критические. Если применить показатели, описанные в предыдущих разделах, это значение возрастет до 40—50%. Многие бреши, оцененные как Критические в Windows XP или других версиях, получили более низкую оценку в Windows Server 2003 только потому, что теперь по умолчанию для Internet Explorer и Outlook устанавливаются жестко ограничивающие настройки — настолько ограничивающие, что эти программы практически невозможно использовать, не изменив хотя бы некоторые из установленных по умолчанию настроек.

Совершенно иная картина получается с последними 40 уязвимостями Red Hat. Только четыре из них оценены по нашим показателям как Критические (Red Hat не указывает уровень серьезности своих предупреждений). Это означает, что 10% из 40 самых последних обновлений имеют серьезность Критическая. Фактически же эта оценка работает в поддержку Microsoft, поскольку для двух брешей можно легко доказать, что их серьезность ниже критической, что уменьшит процент критических брешей до 5%.

Программные коррекции и уязвимости Microsoft Windows Server 2003

В Табл. 1 содержится информация об уязвимостях из 40 последних программных коррекций системы защиты, выпущенных компанией Microsoft[6].

Microsoft обозначила 15 из этих 40 уязвимостей как Критические. Это означает, что, согласно субъективному анализу самой Microsoft, 38% из последних обнаруженных и исправленных проблем имеют серьезность Критическая, то есть наивысшую из возможных.

Однако в применяемой Microsoft методике оценки серьезности брешей есть два спорных момента.

1. Microsoft часто присваивает бреши серьезность **Критическая** для всех операционных систем Windows, *кроме* Windows Server 2003, когда той же бреши присваивается более низ-

[6] Microsoft Security Bulletin, Current Downloads (результаты могут отличаться от приведенных, так как информация регулярно обновляется) — http://www.microsoft.com/technet/security/CurrentDL.aspx

кий статус — **Важная**. Причина этого различия в том, что установленные в Windows Server 2003 настройки по умолчанию отличаются от настроек по умолчанию в других версиях Windows. Microsoft следующим образом описывает различные настройки[7]:

«Уровень безопасности, установленный для зоны Интернет — Высокий. Такая настройка запрещает загрузку скриптов, элементов управления ActiveX, Microsoft Java Virtual Machine (MSJVM), HTML-контента и файлов. Отключено автоматическое обнаружение сайтов интрасети. Такая настройка передает все web-сайты интрасети и все сетевые пути UNC (Universal Naming Convention), не перечисленные в явном виде в зоне локальной интрасети, в зону Интернет.

Заблокированы установка по требованию (Install On Demand) и использование дополнительных компонентов web-навигатора, разработанных не компанией Microsoft. Такая настройка не позволяет web-страницам автоматически устанавливать дополнительные компоненты, а также не позволяет функционировать компонентам, разработанным не компанией Microsoft.

Заблокирован мультимедиа-контент. Такая настройка не позволяет запускать музыкальные клипы, анимационные клипы и видеоклипы».

Хотя некоторые из этих настроек по умолчанию (например, блокировка мультимедиаконтента) абсолютно логичны для сервера, почти невозможно представить, чтобы кто-либо из использующих Windows Server 2003 не изменил настройки, описанные в первом абзаце. Эти настройки делают Internet Explorer почти бесполезным для администратора сервера, желающего использовать web-навигатор для выполнения административных задач, загрузки обновлений и т. д. Понижать уровень серьезности, предполагая, что пользователи Windows Server 2003 оставят такие настройки по умолчанию без изменений — значит, в лучшем случае, заблуждаться. Если бы пользователям Windows Server 2003 предлагалось администрировать сервер удаленно, это могло бы уменьшить данную угрозу. Но Microsoft рекламирует знакомый локальный интерфейс Windows как главное преимущество Windows Server 2003.

2. В приведенный ниже список (табл. 1) включены бреши, уровень серьезности которых ограничен в соответствии с полномочиями пользователя. Эти случаи отмечены в таблице: в столбце «Полномочия» указано «Пользователь». Но поскольку Windows Server 2003 — это сервер, то очевидно, что большинство пользователей, непосредственно работающих на компьютере под управлением Windows Server 2003, будут администраторами. Даже если предположить, что все станут использовать оптимальные приемы работы на настольном компьютере, очевидно, что администраторы Windows Server 2003 входят в систему с полномочиями администратора. Поэтому в тех случаях, когда серьезность брешей «ограничивается» полномочиями пользователя, большую часть времени уровень серьезности фактически не уменьшается, так как пользователь будет иметь полномочия администратора. В качестве примера можно привести брешь, описанную в Microsoft Security Bulletin MS04-015. По указанной выше причине эта брешь заслуживает оценки Критическая, а не Важная. Парадоксально, но подобные бреши в Linux заслуживают понижения оценки, потому что Linux не предлагает администраторам работать в графической среде непосредственно на сервере.

Приняв во внимание все обстоятельства, следует оценить как Критические еще по меньшей мере пять уязвимостей. Это означает, что по показателям, описанным в предыдущих разделах, 50% перечисленных брешей оцениваются как Критические. Если уязвимость должна иметь оценку Критическая с учетом того, что администратор, скорее всего, изменит те настройки по умолчанию, благодаря которым Microsoft понизила уровень серьезности, то этот факт отмечается в скобках. Но при общем сравнении эти уязвимости не рассматривались как Критические. Комментарий в скобках показывает, что Microsoft преднамеренно недооценивает серьезность данной бреши на основании необоснованного допущения — настройки по умолчанию, установленные в Windows Server 2003, существенно меняют ситуацию.

[7] Default Settings Different on Windows Server 2003 Эти настройки перечисляются на нескольких страницах в разделе «Frequently Asked Questions, What is Internet Explorer Enhanced Security Configuration?» Вот одна из таких страниц: http://www.microsoft.com/technet/security/bulletin/ms03-032.mspx

Табл. 1. Программные коррекции и уязвимости Microsoft Windows Server 2003

Дата	Windows Server 2003	Описание	Способ	Путь
Сентябрь 14, 2004	Microsoft Security Bulletin MS04-028	Переполнение буфера при обработке изображений в формате JPEG (GDI+) делает возможным запуск программного кода	Специально сформированное изображение в формате JPEG	Десятки приложений
Июль 30, 2004	Microsoft Security Bulletin MS04-025	Междоменная уязвимость в методе навигации	Вредоносный web-сайт	IE
Июль 30, 2004	Microsoft Security Bulletin MS04-025	Вредоносный BMP-файл	Вредоносный web-сайт	IE
Июль 30, 2004	Microsoft Security Bulletin MS04-025	Вредоносный GIF-файл	Вредоносный web-сайт	IE
Июль 13, 2004	Microsoft Security Bulletin MS04-024	Уязвимость в оболочке Windows делает возможным удаленный запуск программного кода	HTML Email, посещение вредоносного web-сайта	IE
Июль 13, 2004	Microsoft Security Bulletin MS04-023	Уязвимость в HTML showHelp делает возможным запуск программного кода	HTML Email, посещение вредоносного web-сайта	IE, Help and Support Center
Июль 13, 2004	Microsoft Security Bulletin MS04-023	Уязвимость в HTML-справке делает возможным запуск программного кода	HTML Email, посещение вредоносного web-сайта	IE, Help and Support Center
Июль 13, 2004	Microsoft Security Bulletin MS04-018	Накопительное обновление безопасности для Outlook Express	Специально сформированный заголовок электронного письма	Outlook Express 6
Июнь 8, 2004	Microsoft Security Bulletin MS04-017	Уязвимость в Crystal Reports Web Viewer делает возможным раскрытие информации и атаку типа «отказ в обслуживании»	Специально сформированный HTTP-запрос	Visual Studio .Net, IIS
Июнь 8, 2004	Microsoft Security Bulletin MS04-016	Уязвимость в DirectPlay делает возможной атаку типа «отказ в обслуживании»	Отправка вредоносного пакета на сервер	IDirectPlay4
Май 11, 2004	Microsoft Security Bulletin MS04-015	Уязвимость в центре справки и поддержки делает возможным удаленный запуск программного кода	HTML Email, посещение вредоносного web-сайта	IE, Help and Support Center
Апрель 13, 2004	Microsoft Security Bulletin MS04-014	Уязвимость в Microsoft Jet Database Engine делает возможным запуск программного кода	Специально сформированный запрос в Jet (SQL) Engine	Jet Engine (SQL Server), IIS
Апрель 13, 2004	Microsoft Security Bulletin MS04-013	Накопительное обновление безопасности для Outlook Express	HTML Email, посещение вредоносного web-сайта	MHTML Handling of Outlook Express
Апрель 13, 2004	Microsoft Security Bulletin MS04-012	Уязвимость в стандартной библиотеке RPC	RPC	RPC
Апрель 13, 2004	Microsoft Security Bulletin MS04-012	Уязвимость в сервисе RPCSS	Специально сформированное сообщение	RPCSS

Доступ	Полномочия	Ущерб	Участие пользователя	Серьезность (оценка Microsoft)
Удаленный (через Интернет)	Администратор	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Критическая
Удаленный (через Интернет)	Пользователь	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Средняя (должно быть: Критическая)
Удаленный (через Интернет)	Пользователь	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Нет
Удаленный (через Интернет)	Пользователь	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Критическая
Удаленный (через Интернет)	Пользователь	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Важная (должно быть: Критическая)
Удаленный (через Интернет)	Пользователь	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Критическая
Удаленный (через Интернет)	Пользователь	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Критическая
Удаленный (через Интернет)	Пользователь	Отказ в обслуживании (отказ Outlook Express)	Нет	Средняя
Удаленный (через Интернет)	Сервис	Удаляет файлы, Привилегированный доступ к информации, отказ в обслуживании (DoS)	Нет	Средняя
Удаленный (через Интернет)	Сервис	Отказ в обслуживании (DoS) на многопользовательском игровом сервере (Multiplayer Game Server)	Нет	Средняя
Удаленный (через Интернет)	Пользователь	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Важная (должно быть: Критическая)
Удаленный (через Интернет)	Сервис	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Важная
Удаленный (через Интернет)	Администратор	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Да	Критическая
Удаленный (через Интернет)	Администратор	Полный контроль, Неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Критическая
Удаленный (через Интернет)	Сервис	DoS (сервис RPCSS перестает отвечать на запросы)	Нет	Важная

Табл. 1. Программные коррекции и уязвимости Microsoft Windows Server 2003 (продолжение)

Дата	Windows Server 2003	Описание	Способ	Путь
Апрель 13, 2004	Microsoft Security Bulletin MS04-012	RPC поверх HTTP	Специально сформированное сообщение	IIS/COM Internet Services
Апрель 13, 2004	Microsoft Security Bulletin MS04-012	Идентификатор объекта	Специально сформированное сообщение, требуется действующий регистрационный ИД	IIS/COM
Апрель 13, 2004	Microsoft Security Bulletin MS04-011	Уязвимость в LSASS	Специально сформированное сообщение	LSASS
Апрель 13, 2004	Microsoft Security Bulletin MS04-011	Уязвимость в PCT	Специально сформированное TCP-сообщение	PCT/SSL, приложения, использующие SSL (IIS)
Апрель 13, 2004	Microsoft Security Bulletin MS04-011	Уязвимость в HTML-справке делает возможным запуск программного кода	HTML Email, посещение вредоносного web-сайта	HTML Help
Апрель 13, 2004	Microsoft Security Bulletin MS04-011	Уязвимость в H.323/ICF	Специально сформированное сообщение	NetMeeting
Апрель 13, 2004	Microsoft Security Bulletin MS04-011	Negotiate SSP	Специально сформированное сообщение	IIS
Апрель 13, 2004	Microsoft Security Bulletin MS04-011	Уязвимость в SSL	Вредоносное сообщение	IIS/SSL
Апрель 13, 2004	Microsoft Security Bulletin MS04-011	Уязвимость ASN.1 «Double Free»	Специально сформированный запрос на аутентификацию	ASN.1, используется многими приложениями
Февраль 10, 2004	Microsoft Security Bulletin MS04-007	Уязвимость в библиотеке ASN.1 может допустить запуск кода	Специально сформированный запрос на аутентификацию	ASN.1, используется многими приложениями
Февраль 10, 2004	Microsoft Security Bulletin MS04-006	Уязвимость в службе WINS (Windows Internet Name Service) может допустить запуск кода	Специально сформированное сообщение, переполнение буфера	WINS
Февраль 2, 2004	Microsoft Security Bulletin MS04-004	Междоменная уязвимость	HTML Email, посещение вредоносного web-сайта	IE
Февраль 2, 2004	Microsoft Security Bulletin MS04-004	Уязвимость операции перетаскивания (Drag-and-Drop)	HTML Email, посещение вредоносного web-сайта	IE
Февраль 2, 2004	Microsoft Security Bulletin MS04-004	Неправильная URL-канонизация	HTML Email, посещение вредоносного web-сайта	IE

Доступ	Полномочия	Ущерб	Участие пользователя	Серьезность (оценка Microsoft)
Удаленный (через Интернет)	Пользователь, сервис	DoS (сервер перестает отвечать на запросы)	Нет	Низкая
Удаленный (через Интернет)	Сервис, администратор	DoS (требуется перезапуск IIS)	Нет	Низкая
Только локальный администратор	Нет	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Низкая
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Низкая
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Требуется	Критическая
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Важная
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Критическая
Удаленный (через Интернет)	Нет	Перезагрузка системы в результате DoS	Нет	Важная
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Критическая
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Критическая
Удаленный (через Интернет)	Администратор	Отказ в обслуживании (WINS перестает отвечать на запросы), возможен полный контроль	Нет	Важная
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Требуется	Средняя
Удаленный (через Интернет)	Пользователь	Загружает программы без уведомления	Требуется	Средняя
Удаленный (через Интернет)	Пользователь	Фальсификация web-сайта	Требуется	Важная

Табл. 1. Программные коррекции и уязвимости Microsoft Windows Server 2003 (окончание)

Дата	Windows Server 2003	Описание	Способ	Путь
Январь 13, 2004	Microsoft Security Bulletin MS04-003	Переполнение буфера в компонентах MDAC делает возможным запуск программного кода	Фальсификация локального сервера SQL Server	MDAC
Январь 13, 2004	Microsoft Security Bulletin MS04-001	Дефект фильтра H.323 сервера Internet Security and Acceleration Server 2000 делает возможным удаленный запуск программного кода	Специально сформированное сообщение, переполнение буфера	Microsoft Firewall Service, Microsoft Internet Security and Acceleration Server
Ноябрь 11, 2003	Microsoft Security Bulletin MS03-048	Междоменная уязвимость	HTML Email, посещение вредоносного web-сайта	IE
Ноябрь 11, 2003	Microsoft Security Bulletin MS03-048	Уязвимость в XML-объекте	HTML Email, посещение вредоносного web-сайта	IE
Ноябрь 11, 2003	Microsoft Security Bulletin MS03-048	Уязвимость операции перетаскивания (Drag-and-Drop)	HTML Email, посещение вредоносного web-сайта	IE
Октябрь 15, 2003	Microsoft Security Bulletin MS03-045	При переполнении буфера элементов управления «Список» и «Поле со списком» может возникнуть возможность запуска кода	Использование бреши в графическом элементе управления	Windows API
Октябрь 15, 2003	Microsoft Security Bulletin MS03-044	Переполнение буфера центра справки и поддержки Windows может поставить под угрозу безопасность системы	HTML Email, посещение вредоносного web-сайта	IE, Help and Support Center, Протокол HCP
Октябрь 15, 2003	Microsoft Security Bulletin MS03-043	Переполнение буфера службы Windows Messenger может допустить запуск кода	Специально сформированное сообщение	Служба Messenger Service, отключенная по умолчанию
Октябрь 15, 2003	Microsoft Security Bulletin MS03-041	Уязвимость в механизме проверки кода подлинности может допустить удаленный запуск кода	Вредоносный элемент управления ActiveX, используемый без разрешения при недостаточном объеме памяти	ActiveX Authentication
Сентябрь 10, 2003	Microsoft Security Bulletin MS03-039	Переполнение буфера RPCSS может допустить запуск кода	Специально сформированное сообщение	RPCSS

Доступ	Полномочия	Ущерб	Участие пользователя	Серьезность (оценка Microsoft)
Удаленный (через Интернет)	Сервис	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Важная
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Критическая
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Требуется	Средняя (должно быть: Критическая)
Удаленный (через Интернет)	Пользователь	Нарушитель может прочитать известные файлы в системе	Требуется	Низкая
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Требуется	Средняя (должно быть: Критическая)
Локальный пользователь с действующим ИД	Пользователь	Полный контроль, неограниченный	Нет	Низкая
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Требуется	Критическая
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Критическая
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Требуется	Критическая
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS	Нет	Критическая

Табл. 2. Программные коррекции и уязвимости Red Hat Enterprise Linux AS v.3

Дата	Red Hat Advanced Server	Описание	Метод	Путь
Сентябрь 7, 2004	RHSA-2004:400-15	Обновленный пакет gaim позволяет устранить проблемы безопасности	Отправить специально подготовленные данные в GAIM-клиент	GAIM (Instant Messenger)
Сентябрь 1, 2004	RHSA-2004:323-09	Обновленный пакет lha позволяет устранить уязвимость в системе защиты	Убедить пользователя использовать специально сформированную команду	Хорошо сформированный LHA-архив, убеждающий пользователя использовать команду
Сентябрь 1, 2004	RHSA-2004:349-10	Обновленные пакеты http позволяют устранить брешь mod_ssl в системе защиты	Прервать SSL-запрос в определенном состоянии	Apache 2.0.50 и более ранние версии
Сентябрь 1, 2004	RHSA-2004:436-07	Обновленный пакет rsync позволяет устранить проблему безопасности	Отправить специально сформированную команду rsync	rsync 2.6.2 и более ранние версии
Август 31, 2004	RHSA-2004:350-12	Обновленные пакеты krb5 позволяют устранить проблемы безопасности	Отправить специально сформированный запрос на аутентификацию	Kerberos authentication
Август 26, 2004	RHSA-2004:432-08	Обновленный пакет acrobat позволяет устранить проблемы безопасности	Специально сформированный закодированный файл (uuencoded)	Acrobat Reader
Август 20, 2004	RHSA-2004:414-19	Обновленные пакеты qt позволяют устранить проблемы безопасности	Специально сформированный файл с изображением	Qt (инструментарий, используемый KDE)
Август 5, 2004	RHSA-2004:378-08	Обновленные пакеты Ethereal позволяют устранить проблемы безопасности	Отправить вредоносные пакеты	программа контроля сети Ethereal
Август 4, 2004	RHSA-2004:373-13	Обновления GNOME VFS для уязвимости extfs	Убедить пользователя открыть специальный URI	GNOME-VFS
Август 4, 2004	RHSA-2004:402-08	Обновленные пакеты libpng позволяют устранить проблемы безопасности	Создать специально сформированный png-файл, убедить пользователя посетить web-сайт	libpng
Август 4, 2004	RHSA-2004:421-17	Обновленные пакеты mozilla позволяют устранить проблемы безопасности	Несколько способов, включая вредоносный java-скрипт	web-навигатор Mozilla
Август 3, 2004	RHSA-2004:413-07	Обновленные пакеты kernel позволяют устранить уязвимости в системе защиты	Доступ к большим объемам памяти	Kernel
Июль 29, 2004	RHSA-2004:308-06	Обновленный пакет ipsec-средств	Проверить сертификат X.509	ipsec-средства
Июль 29, 2004	RHSA-2004:409-05	Обновленные пакеты sox позволяют устранить переполнения буферов	Специально подготовленный WAV-файл	sox (Sound eXchange)

Доступ	Полномочия	Ущерб	Участие пользователя	Серьезность
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Нет	Важная (Gaim обычно не используется на сервере)
Загрузка или иной способ получения файла со сжатием lha	Пользователь	Полный контроль, неограниченный	Да	Низкая (lha — это редко используемый устаревший формат сжатия)
Удаленный (через Интернет)	Сервис	Расход ресурсов ЦП (возможно, DoS)	Нет	Важная
Удаленный (через Интернет)	Сервис	Позволяет читать/записывать файлы, не определенные в качестве доступных с помощью rsync	Нет	Важная (rsync не является общедоступным сервисом, и chroot сводит эту уязвимость на нет)
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Критическая
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Да	Важная (Acrobat обычно не используется на сервере)
Удаленный (через Интернет)	Пользователь	Отказ Qt, возможно выполнение кода	Да	Важная
Удаленный (через Интернет)	Администратор	Отказ Ethereal, возможно выполнение кода	Нет	Критическая
Нет	Пользователь	Позволяет выполнять действия в качестве пользователя	Да	Низкая
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Да	Важная (web-навигатор обычно не используется на сервере)
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Да	Важная (web-навигатор обычно не используется на сервере)
Локальный пользователь с действующим ИД	Нет	DoS (сервер перестает отвечать на запросы)	Да	Низкая
Удаленный (через Интернет)	Нет	Не прерывается обмен кодами при неудачной верификации	Нет	Важная
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Да	Важная

Табл. 2. Программные коррекции и уязвимости Red Hat Enterprise Linux AS v.3 (продолжение)

Дата	Red Hat Advanced Server	Описание	Метод	Путь
Июль 22, 2004	RHSA-2004:259-23	Обновленные пакеты samba позволяют устранить уязвимости	Специально подготовленная HTTP-аутентификация	Samba (сервисы Windows)
Июль 19, 2004	RHSA-2004:392-13	Обновленные пакеты php позволяют устранить проблемы безопасности	Неочевидная хэш-атака	PHP
Июль 6, 2004	RHSA-2004:342-10	Обновленные пакеты httpd позволяют устранить проблемы безопасности	Подделать удостоверяющий центр (CA) SSL, которому SSL доверяет, или использовать большой объем памяти	Apache with SSL
Июль 2, 2004	RHSA-2004:360-05	Обновленные пакеты kernel позволяют устранить проблемы безопасности	Подмонтировать файловую систему NFS с уязвимого компьютера	Kernel
Июнь 18, 2004	RHSA-2004:249-07	Обновленные пакеты libpng позволяют устранить проблемы безопасности	Создать специально сформированный png-файл, убедить пользователя посетить web-сайт	libpng
Июнь 17, 2004	RHSA-2004:255-10	Обновленные пакеты kernel позволяют устранить уязвимости в системе защиты	Запустить такие функции, как fsave и frstor	Kernel
Июнь 14, 2004	RHSA-2004:240-06	Обновленный пакет SquirrelMail позволяют устранить несколько уязвимостей	Пользователь электронной почты может запустить специально подготовленный URL	PHP, Squirrelmail
Июнь 9, 2004	RHSA-2004:233-07	Обновленный пакет CVS позволяет устранить проблемы безопасности	Отправить специально подготовленные инструкции в CVS	CVS
Июнь 9, 2004	RHSA-2004:234-06	Обновленные пакеты Ethereal позволяют устранить проблемы безопасности	Отправить вредоносные пакеты	программа контроля сети Ethereal
Июнь 9, 2004	RHSA-2004:236-14	Обновленные пакеты krb5	Использовать искаженные аутентификационные имена	Kerberos authentication
Июнь 9, 2004	RHSA-2004:242-06	Обновленный пакет squid позволяет устранить уязвимость в системе защиты	Отправить чрезмерно длинный пароль	Кэш и прокси Squid
Май 26, 2004	RHSA-2004:174-09	Обновленный пакет utempter позволяет устранить уязвимость	Если сервис utempter активен, позволяет написать приложение, которое использует брешь	utempter

Доступ	Полномочия	Ущерб	Участие пользователя	Серьезность
Администратор	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Да	Низкая (требуется предварительная аутентификация пользователя с помощью inetd/hosts.allow)
Удаленный (через Интернет)	Сервис	Позволяет выполнять программный код в качестве пользователя Apache	Нет	Низкая (очень трудно использовать, зависит от структуры сайта)
Удаленный (через Интернет)	Сервис	Позволяет выполнять программный код в качестве пользователя Apache; возможно, DoS	Нет	Средняя (из-за возможности DoS-атаки)
Локальный пользователь с действующим ИД, должна функционировать NFS	Группа	Возможно, изменяет файл, принадлежащий другой группе	Нет	Низкая
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Да	Важная
Локальный пользователь, запускающий программы, которые вызывают отказ kernel	Нет	Отказ в обслуживании (сервер перестает отвечать на запросы)	Да	Низкая (нарушитель должен запустить программы на сервере)
Удаленный пользователь Интернета с действующим регистрационным ИД	Сервис	Модифицирует содержимое базы данных, функционирует как другие пользователи web-mail	Нет	Важная (требуется пользователь с действующей учетной записью)
Удаленный пользователь Интернета с действующим регистрационным ИД	Сервис	Выполняет программный код с полномочиями пользователя CVS	Нет	Важная (требуется пользователь с действующей учетной записью)
Удаленный (через Интернет)	Администратор	Полный контроль, неограниченный, DoS (сервер перестает отвечать на запросы)	Нет	Критическая
Удаленный (через Интернет)	Администратор	Неизвестен	Нет	Низкая (используемая по умолчанию конфигурация Kerberos на Red Hat не имеет этой уязвимости)
Локальный пользователь с действующим ИД	Сервис	Выполняет программный код с полномочиями пользователя Squid	Нет	Низкая (требуется действующая учетная запись пользователя; используемая по умолчанию конфигурация Squid не имеет этой уязвимости)
Локальный или удаленный пользователь с действующим ИД	Администратор	Позволяет перезаписывать привилегированные файлы с symlink	Нет	Низкая (требуется действующая учетная запись пользователя; utempter является скрытым сервисом, который очень трудно использовать)

Табл. 2. Программные коррекции и уязвимости Red Hat Enterprise Linux AS v.3 (окончание)

Дата	Red Hat Advanced Server	Описание	Метод	Путь
Май 26, 2004	RHSA-2004:219-07	Обновленные пакеты tcpdump позволяют устранить различные уязвимости	Специально подготовленные ISAKMP-пакеты	tcpdump
Май 21, 2004	RHSA-2004:064-11	Обновленные пакеты samba позволяют устранить уязвимость в системе защиты	Случайное изменение учетной записи samba	Samba (сервисы Windows)
Май 21, 2004	RHSA-2004:120-12	Обновленные пакеты OpenSSL позволяют устранить уязвимости	Отправить специально подготовленные SSL-пакеты	OpenSSL
Май 19, 2004	RHSA-2004:180-10	Обновленные пакеты libpng позволяют устранить аварию	Специально подготовленное png-изображение, убедить пользователя посетить web-сайт	libpng
Май 19, 2004	RHSA-2004:190-14	Обновленный пакет CVS позволяет устранить проблемы безопасности	Специально подготовленная CVS- команда	CVS
Май 19, 2004	RHSA-2004:192-06	Обновленный пакет rsync позволяет устранить проблемы безопасности	Отправить специально подготовленную rsync-команду	rsync
Май 17, 2004	RHSA-2004:222-11	Обновленные пакеты kdelibs решают проблемы безопасности URI	Специально подготовленный URI, убедить пользователя посетить web-сайт	KDE
Май 11, 2004	RHSA-2004:165-09	Обновленный пакет ipsec-средств позволяет устранить уязвимости в демоне ISAKMP	Специально подготовленный ISAKMP-заголовок чрезвычайно большого объема	ipsec-tools
Май 11, 2004	RHSA-2004:188-14	Обновленные пакеты kernel для Red Hat Enterprise Linux 3 Update 2	Самая серьезная из исправленных ошибок — возможная эскалация полномочий при монтировании томов Netware	Kernel
Апрель 22, 2004	RHSA-2004:183-03	Обновленные пакеты kernel позволяют устранить уязвимость в системе защиты	Написать программу, чтобы получить полномочия root (администратора)	Kernel
Апрель 17, 2004	RHSA-2004:153-09	Обновленные пакеты CVS позволяют устранить проблемы безопасности	Подделать пути доступа, чтобы перезаписать файлы	CVS
Апрель 14, 2004	RHSA-2004:133-12	Обновленный пакет Squid позволяет устранить уязвимость в системе защиты	Специально подготовленные URL для просмотра запрещенных web-сайтов	Кэш и прокси Squid
Апрель 14, 2004	RHSA-2004:160-05	Обновленные пакеты OpenOffice позволяют устранить уязвимость в системе защиты для neon	Специально подготовленные строки форматов, убедить пользователя посетить web-сайт	OpenOffice

Доступ	Полномочия	Ущерб	Участие пользователя	Серьезность
Удаленный (через Интернет)	Нет	Вызывает аварию tcpdump	Нет	Низкая (tcpdump — это всего лишь утилита, которую администраторы используют для проверки TCP-трафика)
Нет	Нет	Может изменить пароль пользователя на такой, который легче раскрыть	Да	Низкая (очень маловероятный случай с маловероятными последствиями)
Удаленный (через Интернет)	Нет	Может вызвать аварию OpenSSL, Отказ в обслуживании (OpenSSL перестает отвечать на запросы)	Нет	Важная (из-за возможности DoS-атаки)
Удаленный (через Интернет)	Нет	Вызывает аварию приложения, использующегося для вывода изображения	Да	Низкая (перезапускает приложение после его аварии)
Локальный или удаленный пользователь с действующим ИД	Сервис	Выполняет программный код с полномочиями пользователя CVS	Нет	Важная (требуется пользователь с действующей учетной записью)
Удаленный (через Интернет)	Сервис	Позволяет читать/записывать файлы, не определенные в качестве доступных с помощью rsync	Нет	Важная (rsync не является общедоступным сервисом, и chroot сводит эту уязвимость на нет)
Удаленный (через Интернет)	Пользователь	Полный контроль, неограниченный	Да	Важная
Удаленный (через Интернет)	Нет	Отказ в обслуживании (сервер перестает отвечать на запросы)	Нет	Критическая
Локальный или удаленный пользователь с действующим ИД	Нет	Нет	Нет	Низкая (скрывает исправления ошибок)
Локальный пользователь с действующим ИД	Администратор	Полный контроль, неограниченный	Нет	Важная (требуется пользователь с действующей учетной записью)
Локальный или удаленный пользователь с действующим ИД	Сервис	Перезаписывает файлы вне каталогов CVS	Нет	Важная (требуется пользователь с действующей учетной записью)
Локальный или удаленный пользователь с действующим ИД	Нет	Позволяет просматривать web-страницы, заблокированные с помощью Squid	Нет	Средняя (обычно используется для обмана Squid с целью получить доступ к запрещенным сайтам, таким как порносайты, но может использоваться и для доступа к блокированным страницам интрасети)
Удаленный (через Интернет)	Пользователь	Выполняет программный код	Да	Средняя (OpenOffice обычно не используется на сервере)

Программные коррекции и уязвимости Red Hat Enterprise Linux AS v.3

В Табл. 2 содержится информация об уязвимостях из 40 последних программных коррекций системы защиты, выпущенных компанией Red Hat.

Компания Red Hat не определяла уровень серьезности. Для оценки каждой уязвимости использованы показатели, описанные в данной статье, при этом учитывалось, что серверы Linux обычно администрируются с настольных систем, а не через графический интерфейс на самом сервере. Многие оценки сопровождаются краткими комментариями, которые помогут читателям понять эту оценку.

Из 40 уязвимостей только четыре оценены как Критические. Это означает, что 10% из 40 последних обновлений имеют серьезность Критическая.

Но если принять во внимание особенности программного обеспечения, к которому относятся две из четырех уязвимостей, можно утверждать, что серьезность этих двух уязвимостей не следует оценивать так высоко. Эти две уязвимости связаны с программой Ethereal. Ethereal — это одно из нескольких доступных средств контроля сетевых компонентов и прослушивания сети («sniffer»). Программа Ethereal запускается при необходимости, а не в качестве постоянного сервиса, поэтому вероятность того, что она будет работать в момент, когда кто-то пытается воспользоваться ее уязвимостью, крайне мала. Если по этой причине понизить серьезность указанных уязвимостей до Важная, то только 5% из 40 последних предупреждений следует считать Критическими.

Уязвимости в сервисах IPSEC и Kerberos более обосновано оценены как Критические, поскольку эти сервисы функционируют на постоянной основе.

Лишь немногие уязвимости позволяют злоумышленнику действовать на уровне администратора. Однако даже в этих редких случаях, как правило, имеются факторы, уменьшающие опасность. Например, уязвимость в Samba (июль 22, 2004, RHSA-2004:259-23) можно использовать только в том случае, если кто-либо сконфигурирует inetd (через файл hosts.allow) так, что известному пользователю и компьютеру разрешается доступ к этому сервису. Если система сконфигурирована правильно, то никто, кроме авторизованного известного пользователя, не может получить доступ к программе конфигурации Samba, чтобы использовать данную уязвимость. В противном случае серьезность этой уязвимости следовало бы оценить как Критическую. Для использования других брешей, позволяющих получить административный доступ, также необходимо быть известным пользователем с действующим идентификатором. Это уменьшает угрозу и снижает серьезность, поскольку значительно увеличивается вероятность поимки злоумышленника.

Результаты запросов к базе данных CERT по уязвимостям

Американская группа Computer Emergency Readiness Team (CERT) использует свой собственный набор показателей для оценки серьезности брешей в защите. Результат выражается числом в диапазоне от 0 до 180, причем значение 180 означает самую серьезную уязвимость. Шкала является нелинейной. Иначе говоря, уязвимость с оценкой 100 не является в два раза более серьезной, чем уязвимость с оценкой 50.

CERT считает любую уязвимость с оценкой 40 или выше достаточно серьезной, чтобы включить ее в специальное техническое предупреждение, выпускаемое CERT Advisory и US-CERT.

Мы сделали запросы к базе данных CERT по ключевым словам «Microsoft», «Red Hat» и «Linux». К сожалению, средства web-поиска на сайте CERT не позволяют обеспечить в полной мере желаемую детализацию и долговечность результатов. Особенно это верно для результатов поиска по «Red Hat» и «Linux». Результаты поиска по «Linux» включают в себя несколько уязвимостей Oracle, общих для Linux, UNIX и Windows. Результат по «Red Hat» с данными о самой серьезной уязвимости даже не содержит среди подробностей указания на Red Hat как на уязвимую систему. Результаты поиска по «Microsoft» представляются вполне точными, так как и в подробностях, и в самих записях указаны бреши именно в программном обеспечении Microsoft. Вследствие этого результаты несколько искажаются не в пользу Linux и Red Hat. Тем не менее, даже если принять эти результаты, проигнорировав искажение для Linux и Red Hat, все равно получается, что большинство записей в базе данных CERT относится к

Microsoft, и эти записи содержат информацию о самых серьезных брешах.

Запрос к базе данных CERT по слову «Microsoft» дал 250 результатов, причем две первые записи описывают бреши с показателем серьезности 94,5. 39 записей описывают бреши с серьезностью 40 или выше. Средняя оценка серьезности по 40 первым записям — 54,67. (Усреднение проведено по 40 записям, а не по 50 или больше, так как поиск для «Red Hat» дал только 46 записей).

Запрос к базе данных CERT по слову «Red Hat» дал 46 результатов. Первая запись описывает брешь с показателем серьезности 108,16. Только три записи (против 39 для Microsoft) содержат оценку серьезности 40 или выше. Среднее значение серьезности по первым 40 записям — 17,96.

Запрос к базе данных CERT по слову «Linux» дал 100 результатов. Первая запись описывает брешь с показателем серьезности 87,72. Только шесть записей содержат оценку серьезности 40 и выше. Среднее значение серьезности по первым 40 записям — 28,48.

Не стоило ожидать, что эти результаты совпадут с результатами нашего анализа по последним программным коррекциям. CERT использует другие критерии отбора, другой порядок дат, к тому же CERT не ограничивается только Windows Server 2003 и Red Hat Enterprise Linux AS v.3. Но результаты запросов к базе данных CERT отражают тот факт, что бреши в системе защиты Windows оказываются серьезными гораздо чаще, чем бреши в Linux, что соответствует нашим выводам.

Jet Info
ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ
Издается с 1995 года
Издатель: компания Джет Инфо Паблишер

Главный редактор: Дмитриев В.Ю. *(vlad@jet.msk.su)*
Технический редактор: Лапина И.К. *(lapina@jet.msk.su)*
Россия, 127015, Москва, Б. Новодмитровская, 14/1
тел. (095) 411 76 01
факс (095) 411 76 02
email: JetInfo@jet.msk.su http://www.jetinfo.ru

Подписной индекс по каталогу Роспечати **32555**