# SPECIAL ОПРОС

**АЛЕКСЕЙ ПЕТРОВ**

В IT 20 лет. Эксперт в области защиты данных, эксперт по компьютерным преступлениям, эксперт по сетевым коммуникациям и телефонии. Сертификаты от *Novell/3com/Bay/Siemens/Cisco/ISACA*. Консультант по вопросам ИТ-безопасности в Secproof Oy (*www.secproof.com*). Свободный консультант *Arhont.com*, *iPRO.lv*.

**ВЛАДИМИР СЕЛЕЗНЕВ**

Технический директор, интернет-провайдер «Синхролайн» (*www.sl.ru*).

**АРТУР ЕНАЛИЕВ**

Окончил МФТИ в 1999 году. На данный момент работает техническим директором ООО «Бест Хостинг».

**АНТОН КАРПОВ**

Специалист в области информационной безопасности. В «Х» пишет с переменной периодичностью вот уже несколько лет. Круг профессиональных интересов: сетевые атаки, безопасность UNIX-систем, безопасность беспроводных сетей..

**КРИС КАСПЕРСКИ**

Известен еще как мыщъх. Компьютеры грызет еще с тех времен, когда Правец-8Д считался крутой машиной, а дисковод с монитором были верхом мечтаний. Освоил кучу языков и операционных систем, из которых реально использует W2K, а любит FreeBSD 4.5. Живет в норе, окруженной по периметру компьютерами и стеллажами с литературой.

## КОГДА И ГДЕ BSD НЕЗАМЕНИМА?

**АЛЕКСАНДР АНТИПОВ:** BSD отличается высокой надежностью и стабильностью работы, поэтому лучше всего использовать ее в многозадачных и критических системах — массовых почтовых системах, для организации публичных сервисов, хостинга и т.п.

**АРТУР ЕНАЛИЕВ:** Первое — это серверные системы высокой надежности, которые настраиваются один раз и работают долго и счастливо. Второе — касается лицензии BSD. Она позволяет на базе открытого ПО сделать «свой» продукт с закрытым кодом.

**АНТОН КАРПОВ:** В мире операционных систем общего назначения, коими являются BSD, Linux и Windows, действовало, действует и будет действовать главное правило: самая лучшая (для какой-либо задачи) ОС — та, которую лучше знаешь и которую лучше умеешь «готовить». Так что про незаменимость речи не идет: на Win можно построить маршрутизатор с фильтрацией пакетов и балансировкой нагрузки, а на BSD — контроллер домена Windows. Другое дело, что все операционки, конечно, разрабатываются с прицелом на конкретный, узко очерченный круг задач, которые они могут выполнить хорошо (в идеале — лучше всех).

FreeBSD разрабатывается с целью быть лучшей на x86-серверах общего назначения (почта, веб, базы данных и тому подобное). Усилия разработчиков направлены, в первую очередь, на оптимизацию для работы на мультипроцессорных (SMP) системах: планировщик задач (шедулер), библиотека нитей (threads), поддержка SMP в ядре — все пишется в угоду тому, чтобы FreeBSD была самой быстрой и производительной на многопроцессорных кластерах. Впрочем, и на однопроцессорных машинах тоже.

NetBSD главной своей целью имеет абсолютную портабельность — нет ни одной операционной системы, поддерживающей такое разнообразие аппаратных платформ. Достигается это как можно меньшим количеством платформозависимого кода в системе, использованием абстракций. Каждый новый релиз NetBSD выпускается для всех платформ (на данный момент их 57). Так что если задаться целью найти современную ОС, чтобы вдохнуть вторую жизнь в откопанный невесть где старенький компьютер экзотической архитектуры, то выбор будет однозначный — NetBSD.

OpenBSD не имеет аналогов в opensource-мире по двум параметрам: безопасность и сетевые возможности. Ребята из OpenBSD были первыми, кто включил в базовую систему такие механизмы проактивной безопасности, как защиту от переполнений буфера в собираемых их компилятором (модифицированным gcc) программах, защиту от выполнения кода в стеке (механизм W^X), рандомизацию адресов выделяемой памяти (модификации в malloc(3), mmap(2)). Они также внедрили механизмы privilege separation и privilege revocation во все сетевые сервисы и утилиты. Средство удаленного администрирования UNIX-машин, OpenSSH, являющееся стандартом де-факто в UNIX-мире, — также дело рук парней из OpenBSD. Причем все эти механизмы входят и включены в систему by default, в отличие от порочной практики, принятой в Linux-мире, согласно которой хорошую систему надо собирать, руководствуясь принципом «с миру по патчу».

Что касается сетевых возможностей, не побоюсь предположить, что «негласная» цель проекта OpenBSD — сделать, с точки зрения функционала, аналог популярных сетевых устройств от Cisco. В первую очередь речь здесь, конечно, идет о межсетевых экранах Cisco Pix. Но аналог открытый и свободный. OpenBSD «из коробки» умеет фильтровать и приоритезировать трафик с помощью мощнейшего пакетного фильтра pf, не имеющего аналогов. Настройка IPSec сводится к двум-трем телодвижениям, благодаря утилите ipsecctl(8), также не имеющей аналогов. То есть настройка IPSec стала не сложнее настройки правил файрвола. Есть поддержка агрегации сетевых интерфейсов (trunk) и прозрачного резервирования (failover) для построения отказоустойчивых кластеров, в том числе и для IPSec-соединений! В базовую систему входят демоны bgpd(8) и ospfd(8), для построения на базе OpenBSD динамического BGP или OSPF-маршрутизатора. Конечно, идеальных систем не бывает — разработчики OpenBSD не успевают следить за новым железом, да и производительность некоторых подсистем (файловая система, система нитей, поддержка SMP) оставляет желать лучшего. Однако, с точки зрения безопасности и по сетевым возможностям, OpenBSD — безусловно, система номер один, и не только в opensource-мире.

**КРИС КАСПЕРСКИ:** Рынок предлагает довольно большой выбор, а на цвет и вкус все фломастеры разные. BSD (особенно NetBSD) перенесена на множество платформ, от суперкомпьютеров до «контроллеров лифта» и прочих встраиваемых устройств. Все операционные системы семейства BSD бесплатны, поставляются в открытых исходных текстах (с правом модификации и доработки), хорошо масштабируются, выдерживают большую нагрузку даже на скромном железе, позволяя собрать приличный сервер, обслуживающий миллионы пользователей одновременно, оплатив только оборудование и, естественно, работу администратора. BSD неприхотлива, но для обращения с ней нужен хороший специалист, поскольку BSD ориентированна именно на специалистов.

**АЛЕКСЕЙ ПЕТРОВ:** BSD незаменима там, где максимально эффективно реализуются ее плюсы для решения конкретной задачи. Каждая операцион-

**АЛЕКСАНДР АНТИПОВ**

Руководитель проекта, автор/соавтор/корректор многочисленных статей ведущего отечественного портала по информационной безопасности *SecurityLab.ru*.

ная система имеет некие тактико-технические характеристики, обусловленные реализацией и внутренней организацией, а также поддержкой какого-то «железа». Но мало выбрать решение. Как правило, его надо реализовать и после этого еще и поддерживать. Для этого нужен знающий «механик»-администратор, который знает внутреннее устройство и может диагностировать неполадки и разобраться в их причине.

BSD будет незаменима там, где для «данной задачи» у нее будет больше баллов в сравнении с другими ОС. Скажем, в сравнении с Linux 2.2, BSD TCP/IP stack будет гораздо быстрее и на большой загрузке выдаст возможный максимум. TCP stack/netfilter BSD vs Linux 2.4 — уже в зависимости от ситуации и задачи придется выбирать либо BSD, либо Linux. BSD хуже справляется с некоторыми задачами (MySQL/SMP/threads/NUMA/SMP). Чего нельзя сказать о производительности некоторых сетевых апликаций. MySQL/SMP/Oracle/Java быстрее и лучше будут работать на Linux'e, nat/fw/ftpd/dns/apache — быстрее на BSD. Но во многих случаях с правильным тюнингом kernel'a эти тезисы очень спорны, и разрыв в производительности на одном и том же железе не так велик.

**ВЛАДИМИР СЕЛЕЗНЕВ:** BSD очень хороша для загруженных, «больших» хостинг-серверов, которые должны выдерживать нагрузку с большой посещаемостью, в сравнении с Linux. В BSD есть несколько очень удобных инструментов для хостинг-платформ, которых раньше не было в Linux, в частности, это простейшая (но надежная) система изоляции системных и прикладных приложений от основной машины (Jail — виртуальные серверы). В Linux бесплатные технологии виртуализации серверов появились не так давно. Очень давно используется система установки и обновления приложений (ports), — наверное, лучшая в своем роде.

## ОТКРЫТЫЕ ИСХОДНИКИ: ПРОГРАММИРОВАНИЕ ИЛИ РЕЛИГИЯ?

**АЛЕКСАНДР АНТИПОВ:** И то, и другое. Долгое время открытые исходники были прежде всего религией, причем ортодоксальной. Такая религия привела к тому, что популярность этих ОС стала ничтожно мала по сравнению с Windows. Однако в последнее время, благодаря гигантам типа IBM и SUN и многомиллиардным вливаниям, у *Nix появился шанс отхватить долю рынка у Microsoft и Co. Правда, этому сильно мешают фанатично настроенные сторонники открытого кода.

**АРТУР ЕНАЛИЕВ:** С одной стороны, это искусство программирования, когда свой код не стыдно сделать открытым (показать другим). С другой стороны, образ мышления, можно сказать, даже религия, когда программист не просто решает поставленную задачу, а делает вклад в развитие огромного сообщества разработчиков ПО с открытым кодом, придавая, таким образом, своему занятию более глобальный смысл. Возможно, это и побуждает программистов делать открытый код более качественным и выверенным.

**АНТОН КАРПОВ:** Для кого как. Очевидно, что для харизматичного чудаковатого Столлмана открытые исходники — это уже давно религия, и сам он — апостол FSF ;). Для большинства же программистов, работающих над opensource-проектами, открытые исходники — это прежде всего возможность совместно заниматься общим делом. Принципиальное расхождение существует лишь в вопросе лицензирования открытого ПО. Как известно, самые популярные лицензии — это GPL (более открытая) и BSDL (более свободная). А вот вопрос использования лицензии, действительно, может легко перейти в религиозное русло :). Впрочем, для профессионального программиста вопрос качества ПО, конечно, гораздо важнее религиозных споров.

**КРИС КАСПЕРСКИ:** Это и программирование, и религия, причем довольно агрессивная. Открытый код позиционируется как универсальное решение всех проблем, сущий рай или, можно сказать, даже коммунизм. Закрытый код отметается сразу, даже если он работает лучше, быстрее, стабильнее. Для многих использование открытого софта является своеобразной формой протеста против Microsoft, и в этом есть свое рациональное зерно. Microsoft безраздельно властвует на рынке, навязывая нам свои уродские API, с не менее уродскими библиотеками, только потому, что большинство программистов даже не догадываются, что в этом мире кроме Windows есть что-то еще.

**АЛЕКСЕЙ ПЕТРОВ:** Для операционных систем открытый код гораздо более ценный критерий. Открытость кода позволяет правильнее и эффективнее писать апликации, понимая, что и как происходит внутри ОС. Всегда можно взять и проанализировать, что и как работает. Почти всегда можно взять базу кода и доработать его под себя, получив максимальный эффект. Не тратится время на изобретение колеса и велосипеда — значит, есть возможность двигаться дальше. Хороших алгоритмов и реализаций не так-то много, и патентование сильно тормозит и усложняет процесс развития в целом. Если кто-то запатентовал колесо — всем остальным на кубиках далеко не уехать, а многие текущие идеи и алгоритмы базируются на десятках лет опыта и вытекают из других — патентовать такие вещи в корне неправильно. Продажа «черных коробок», которые берут что-то на входе и выдают неизвестно что в результате — это часть бизнеса, защита идеи. Хорошо строить что-то из сложных «черных коробок» с сотней входов и выходов, без понятия логики — гораздо сложнее, часто конструкция может быть неустойчивой по абсолютно непонятным причинам. И я не против бизнеса и не ратую за то, чтобы ломать экономику, но часто выходит, что бизнес с радостью «за так» берет базу из BSD/GPL, но ничего туда не вкладывает!

**ВЛАДИМИР СЕЛЕЗНЕВ:** Открытые исходники — это очень удобно. Если у тебя что-то не работает в приложении или что-то работает, но, на твой взгляд, неправильно, то у тебя есть прекрасная возможность заглянуть внутрь программы и посмотреть, что конкретно происходит в этот момент. И, в конце концов, докопаться до сути проблемы, найти «баг» или узнать, в чем ошибся ты сам. Это, конечно, крайний вариант, и к нему редко обращаются, но это не позволяет опустить руки и сказать: «раз что-то не работает, то поделать с этим ничего нельзя». Всегда можно решить проблему.

## ПОЧЕМУ В ОТКРЫТОМ LINUX БОЛЬШЕ ДЫР, ЧЕМ В ОТКРЫТОМ BSD?

**АЛЕКСАНДР АНТИПОВ:** Количество дыр — величина, зависящая от множества параметров: сложности кода, его длины, профессиональных навыков программистов и т.п. Главная причина в том, что BSD на протяжении множества лет разрабатывается строго определенной командой, а линукс прежде всего разрабатывается огромной армией фанатиков, профессиональный уровень многих из которых очень низок.

**АНТОН КАРПОВ:** Основных причин две. Первая — банальна: Linux просто популярнее BSD. Linux имеет поддержку больших компаний, что немаловажно для многих заказчиков. Вполне логично, что, чем более система распространена, тем более пристальное внимание ей уделяют эксперты по безопасности, да и просто взломщики. Конечно, это не означает, что если, скажем, OpenBSD войдет в каждый дом, то в ней обнаружат критические проблемы вроде RPC DCOM :). Однако факт остается фактом — чем популярнее и востребованнее система, тем чаще ее ковыряют эксперты по безопасности, и тем чаще в ней находятся проблемы безопасности, так как они — увы и ах! — есть везде.

Однако немаловажен и тот факт, что в Linux и *BSD-системах применяются разные модели разработки. FreeBSD разрабатывает узкий круг профессионалов — это люди, имеющие право внесения изменений в исходные коды системы (commit bit), каждый из которых ответственен за определенную подсистему (maintainer). В проекте также имеется офицер безопасности (security officer), следящий, в том числе, и за высоким уровнем качества программирования. Все важные патчи должны пройти через него, прежде чем будут добавлены в дерево исходных кодов. Что же касается OpenBSD, то здесь все очевидно — проект изначально имеет целью создание самой безопасной ОС на Земле, а, согласно лидеру проекта, Theo de Raadt, «безопасность определяется качеством» (под качеством понимается, конечно, и качество кода). И с этим трудно спорить. Разумеется, такой механизм более инертен, чем «базарная» модель разработки Linux-ядра, когда сотни разработчиков присылают патчи, за качеством кода которых порой никто не следит.

В результате мы имеем такую «вилку». Динамично развивающаяся ОС, подхватывающая поддержку всех новых технологий, но зато имеющая невысокое качество кода, что приводит к обнаружению все новых дыр. Или консервативная система, где на первое место ставится качество выпускаемого продукта, а не его функционал, что снижает вероятность наличия и, соответственно, обнаружения проблем безопасности.

**КРИС КАСПЕРСКИ:** Потому что «открытость» на безопасность практически никак не влияет. Аудит кода (особенно чужого) на предмет безопасности — весьма трудоемкое занятие. И наивно думать, что миллионы экспертов по всему миру не имеют никакого более интересного занятия, чем ковыряться в недрах Linux'а, который развивается весьма стремительно и объединяет как профессионалов, так и пионеров. Причем Linux — это фактически только ядро, разрабатываемое одной командой, с более или менее централизованной системой управления, а дистрибутивы клепают все, кто попало и из чего попало. Отсюда и дыры.

BSD развивается намного медленнее, причем базовый код, написанный еще черт знает когда, остается практически без изменений, что делает появление новых дыр достаточно маловероятным явлением (в OpenBSD за все восемь лет ее существования была обнаружена только одна серьезная дыра).

**АЛЕКСЕЙ ПЕТРОВ:** BSD писался и пишется инженерами — код перепроверяется и буквально вылизывается по крохам. В BSD хороший version control и менеджмент кода. Linux пишется разнородной группой энтузиастов-программистов, код часто просто не успевают проверять, темпы разработки ядра просто скоростные (особенно это относится к ядру 2.6.x).

## ЧТО ВАЖНО ПРИ ОПТИМИЗАЦИИ СИСТЕМЫ?

**АЛЕКСАНДР АНТИПОВ:** Оптимизация состоит из двух этапов: оптимизация под аппаратную часть и оптимизация под программную среду. В первом случае большой эффект дает правильная компиляция ядра — включение необходимых параметров оптимизации при компилировании, компиляция под конкретно используемый тип процессора. Затем оптимизация сетевых настроек под тип используемой сетевой карты и особенности сетевой среды. В случае программной среды для каждого типа (почтовый сервер, web-сервер, база данных) можно писать большие статьи, но принцип остается неизменным — тюнинг параметров ядра, сетевого окружения и дисковой подсистемы.

**АРТУР ЕНАЛИЕВ:** Главное — не навредить. Другими словами, оптимизируя какой-либо параметр системы, важно следить за тем, чтобы другие параметры той же системы «не портились».

**КРИС КАСПЕРСКИ:** Важно знать, что ты делаешь. Вслепую много не наоптимизируешь. Прежде всего необходимо отбросить концепцию «бутылочного горлышка», то есть самого узкого места, тормозя-

щего все остальные. Если система настроена неправильно — тормозить будет все, хотя ярко выраженных «бутылочных горлышек» может и не быть, что сделает профилировщик бесполезной игрушкой и останется только эксперимент. Кстати говоря, влияние тех или иных параметров на производительность очень часто обнаруживается чисто случайно. Об этом не говорится в документации, и даже сами разработчики пожимают плечами, и только опыт...

**АЛЕКСЕЙ ПЕТРОВ:** Знание и понимание того, как и что работает. Изначально система «настроена» под некое среднее или завышено среднее, и оптимизация заключается в трех шагах. Первый — сбор и анализ статистики (скажем, каких процессов и операций производится больше, распределение и использование памяти, какие сетевые операции — продолжительность и особенности tcp/udp-сессий). Второй — выбор решения, в идеале — просчет решений и выбор более эффективного и оптимального (просто добавить память или поменять настройки ее распределения, конфиги squid/mysql/apache, плюс пересборка ядра). Третий — непосредственно реализация, оптимизация-тюнинг и подгонка решения (изменение стандартных значений tcp/ip-стека, сокеты, буферизация, timeout'ы...). Плюс первые два шага по новой.

**ВЛАДИМИР СЕЛЕЗНЕВ:** Оптимизировать можно сами приложения, которые ты используешь. Например, настроить MySQL, чтобы он использовал необходимое количество памяти или определенную схему работы с клиентами (Child vs Tread), в Apache можно отключать неиспользуемые модули, что уменьшает количество памяти, выделяемой для каждого клиента. Также можно перекомпилировать приложения под твои требования из исходных текстов, включив только то, что нужно, и исключив ненужные функции. Также желательно, чтобы приложение поддерживало работу на нескольких процессорах, если он у тебя не один. Есть замечательный документ «Getting Maximum Performance from MySQL», его положения часто можно перенести и на другие приложения. Эффект может достигать 30-50 процентов.

Второй вариант — когда ты оптимизируешь саму систему, и все приложения на ней начинают работать быстрее. В операционных системах на основе открытых исходных текстов можно перекомпилировать ядро системы. Что позволит максимально использовать аппаратные возможности машины. Здесь нужно четко указать, какое «железо» используется, и на основании этого будет собрано новое ядро, которое будет занимать меньше памяти и работать немного быстрее.

В общем, при оптимизации важно понимать, что для работы твоего приложения является «узким местом»: например, работа с диском, памятью, с процессором или с соединениями по сети. В каждом случае надо находить параметры, которые за это отвечают, и устранять проблему.

## КТО ВЫЖИВЕТ: WIN, LINUX ИЛИ BSD?

**АЛЕКСАНДР АНТИПОВ:** Вопрос о выживании уже не стоит, все эти ОС на рынке более 10 лет. Этого достаточно, чтобы, как минимум, держаться на плаву. Вопрос в том, кто будет лидировать в ближайшее время, тоже не стоит — лидерство Windows очевидно и непоколебимо. Будут идти локальные войны за отдельные секторы рынка, позиции в которых Linux и FreeBSD традиционно сильны — массовые сервисы, хостинг, базы данных, распределенные вычисления и т.п.

**АРТУР ЕНАЛИЕВ:** Выживут все. Для всех этих систем на ближайшее время работы хватит.

**АНТОН КАРПОВ:** Как поклоннику BSD, конечно, хотелось бы видеть тотальный BSD World Domination. Но я не думаю, что в ближайшие годы из этих ОС кто-то должен обязательно умереть. И вот почему. Вокруг Linux, с одной стороны, уже давно нет того ажиотажа, что царил во времена 2.2 и 2.4 ядер. Не даром Линус Торвальдс был включен недавно CNN в десятку людей, больше не влияющих на информационную индустрию. Феерия по поводу «крутой и свободной» ОС прошла, и на первый план теперь выходят проблемы Linux — проблемы модели разработки, проблемы качества кода. Даже самые ярые поклонники этой ОС признают, что с разработкой ядра 2.6 ситуация близка к неразберихе: одни, вроде бы устоявшиеся, подсистемы выносят из ядра, другие ломают от релиза к релизу, про третьи между тем забывают (так, разработчики официально признали, что с подсистемой 802.11 в Linux — беда). Торвальдс время от времени делает заявления в духе «хватит патчей, все усилия направляем на стабильность».

С другой стороны, Linux имеет поддержку множества компаний, и многим пользователям наплевать, что там творится с ядром, пока такие крупные вендоры, как Red Hat или Suse выпускают свои релизы и продают техподдержку. Свободные BSD-системы также умирать не планируют, медленно, но верно прогрессируя. И пусть по некоторым показателям и функционалу они находятся там, где Linux был несколько лет назад, путь развития BSD кажется более продуманным и выверенным. BSD верит в эволюцию, а не в революцию. Пожалуй, главная проблема открытых BSD в том, что за ними не стоят крупные компании. Многие заказчики просто боятся доверять свой бизнес системе, не имеющей мощного коммерческого вендора, предлагающего не просто «коробку», а готовое решение.

Что же касается Windows, то проблемы безопасности в ней находили, находят и находить будут. Наличие в сетевой серверной ОС бреши вроде нашумевшей в 2004 году уязвимости в RPC DCOM, когда любой мог получить полный удаленный контроль над ОС, на которой даже не запущено ни одного сетевого сервиса, — уже достаточный повод для разработчиков, чтобы обильно посыпать себе голову пеплом, отправить операционку на свалку истории и забыть о ней, как о недоразумении. В Microsoft от-

лично понимают, что наличие компонентов IE в серверной системе — лишь дополнительная брешь в безопасности. Однако, чтобы исправить все накопившиеся годами ошибки, надо переписывать ОС «с нуля», а это почти нереальная задача.

**КРИС КАСПЕРСКИ:** Все три перечисленные системы существуют (или точнее даже сосуществуют) уже черт знает сколько лет, под них написано нефиговое количество софта, вложены нехилые деньги, обучены специалисты... Поэтому в обозримом будущем умирать никто не собирается. Но Windows, сосредоточенная в одних руках, имеет меньше шансов на выживание, чем Linux и BSD, которые никому не принадлежат, то есть принадлежат всем.

Лично я, увидев, что сделали с Windows 2000, долго плевался и сказал, что когда поддержка Windows 2000 будет прекращена, я лучше перейду на Debian или BSD, чем сяду за XP или, того хуже, Longhorn. Преимущество Linux/BSD в том, что они не навязывают своим пользователям никакой особенной идеологии. Это конструктор — что хочешь, то и собираешь. А вот попробуй запустить Windows без графического интерфейса, без IE и без кучи других не нужных мне вещей, причем так, чтобы работали нужные мне программы! В долговременной перспективе это означает лишь одно — BSD приобретает пользователей, а Windows их теряет.

**ВЛАДИМИР СЕЛЕЗНЕВ:** Выживут все, но в разных сегментах. Linux завоевывает свою долю на рынке серверов за счет других Unix-подобных операционных систем, наверное, в основном, с закрытым исходным кодом. Плюс небольшой процент насажденных десктоп-клиентов, например, решение на уровне руководства компании «пересадить» всех сотрудников на Linux. BSD — это почти полностью серверная платформа, отчасти она конкурирует здесь с Linux. А Windows никто реально не сможет потеснить с рынка десктопов в обозримом будущем, как не удалось это сделать MacOS и OS/2, не удастся и Linux.

## ЧТО ПРОЩЕ СЛОМАТЬ: WIN, LINUX ИЛИ BSD?

**АЛЕКСАНДР АНТИПОВ:** Простота взлома определяется множеством параметров: слабой конфигурацией по умолчанию, наличием простых в эксплуатации незакрытых дыр, массовостью использования и т.п. Пару лет назад ответ на этот вопрос был бы очевиден, однако с выходом SP2 для Windows XP и будущим релизом Windows Vista явного лидера тут нет.

**АРТУР ЕНАЛИЕВ:** Покажите конфиги — скажу, что проще сломать.

**КРИС КАСПЕРСКИ:** Если все системы залатаны и сконфигурированы правильно, то один хрен их взломаешь, — нужно искать дыры, о которых никто не знает. И тут возникает противоречие: тексты Linux'а открыты и легко читаемы, но дыр там немного, а в OpenBSD, наверное, нет совсем). Тексты Windows закрыты (впрочем, их легко найти в осле), а процесс дизассемблирования отнимает намного больше сил и времени, чем анализ открытых кодов, но и дыр в Windows столько... Что, как говориться, чем больше их находишь, тем больше их остается.

**АЛЕКСЕЙ ПЕТРОВ:** «Проще» сломать то, что уже изначально слабее и изначально содержит в себе больше ошибок. Плюс еще один очень важный фактор — как это «что-то» настроено, насколько хорошо в этом вопросе разбирается администратор (оценивает риски, выбирает решение, настраивает, знает все нюансы, следит за работой системы, понимает механизмы и тонкости ее работы). Комбинация «дырявая ОС» + «хороший админ» — как правило, надежнее, чем «более секьюрная ОС» + «плохо разбирающийся/ленивый админ». Хотя идеал, естественно — «хороший админ» + «хорошая ОС».

BSD — хорошо выверяемый код, более долгая жизнь самого проекта и кода, больше количество пройденных проверок и неплохой уровень программирования (хотя порой бывают досадные логические ошибки в реализации, не связанные с безопасностью). Результат — неплохая безопасность в целом.

Windows — по количеству уязвимостей прочно держит первое место, ошибок там по-прежнему много на разных уровнях, но закрытость кода, его обилие, громадный объем, плюс запутанность скорее порождают нежелание без необходимости искать дыры. И вообще нужно следовать принципу: работает — лучше не трогать.

Linux'у хватает ошибок: отсутствие менеджмента и проверки кода, большая скорость разработки, огромный «штат» не очень хорошо организованной команды свободных приходящих-уходящих разработчиков. Громаднейшее количество «ползающих» по этому коду проверяющих — стабильное первое-второе место по количеству обнаруженных уязвимостей.

**ВЛАДИМИР СЕЛЕЗНЕВ:** Как показывает опыт, взломать можно все, но взлом происходит в 90% случаев на уровне приложения. Так что фактор ОС не так важен. Здесь важно, какие приложения ты используешь, как они поддерживаются, как часто и как быстро в случае проблем обновляются. И OpenSource-команда, и MicroSoft быстро реагируют и выпускают заплатки. Может быть, OpenSource иногда быстрее. Но, если команда разработчиков небольшая или приложение не популярное, заплатку можно ждать долго. С другой стороны, популярные приложения OpenSource чаще привлекают внимание хакеров.

Если ты силен в программировании, то систему с открытым кодом сломать проще. И в такой системе можно исправить ошибку в коде самому. OpenSource-программы написаны с меньшим количеством ошибок, работают надежнее, быстрее. Это мой выбор **C**