№10(83) октябрь 2009 Системный администратор

Системный администратор

ежемесячный журнал www.samag.ru

№10(83) октябрь 2009

Основные грани Ruby: от главных конструкций до приложения

Новые возможности служб AD DS в Windows Server 2008 R2

Linux Live CD: создаем загружаемый диск

Контролируем трафик с помощью BWMeter

Пробуем OSSEC-HIDS – хостовую систему обнаружения атак

До часа «X» – меньше 100 дней

```
~$ perl
print "who can parse perl?"
only perl can parse perl~$
~$
```

SOA-технологии. Взгляд шире!

За что конкуренты любили Sun Microsystems?

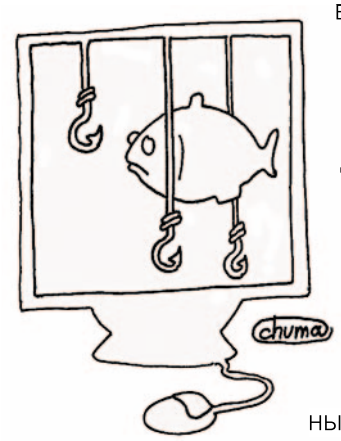
ISSN 1813-5579

Linux center www.linuxcenter.ru

Проблема выбора

Рассказывают, что в американской подземке когда-то никак не могли справиться с пассажирами, которые решили свести счеты с жизнью, бросившись под поезд. Времена на дворе стояли тяжелые, неудивительно, что число самоубийц росло от месяца к месяцу. Тогда на помощь призвали психологов. Они обратили внимание на таблички «Выхода нет».

Представили, как бредет человек в мрачной кишке коридора, загнанный своими проблемами в тупик. Поднимает глаза, видит яркое табло, так созвучное его нерадостным мыслям, «Выхода нет». После этого решиться шагнуть вниз с платформы – легче легкого.



Психологи нашли решение. В подземке появились новые таблички «Выход с другой стороны». А старые просто сняли. И это подействовало! Самоубийства пошли на убыль.

Так и в работе. Хотя, надеюсь, итог мучений для вас не столь драматичен. Бьешься над проблемой, не знаешь, с какой стороны к ней подступиться, какое решение окажется верным. Пока кто-то или что-то не подскажет тот самый «выход с другой стороны». Случается, на это уходят дни, недели…

Так вот, когда вам крайне, совершенно необходимо найти верное решение – откройте «Системный администратор». Просто найдите нужную статью. А если не увидите – сообщите на форуме, позвоните в редакцию. И в следующем номере вы прочтете именно то, что вам нужно!

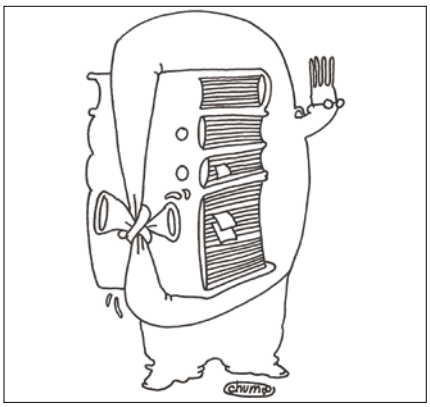
Всем, кто напишет статью, способную помочь хотя бы десятку наших читателей, мы подарим годовую подписку на наш журнал.

Руководство к действию – таков слоган «Системного администратора». РУКОВОДСТВО – ключевое слово в этой фразе.

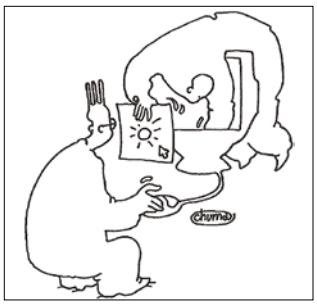
Не нужно рыться в технической литературе, блуждать по Интернету, допытываться у знакомых. Не теряйте время! Добиться результата простым и быстрым способом можно – обратитесь к «Системному администратору»!

Наша задача – показать простой путь к решению проблемы. Ваше право – выбрать его.

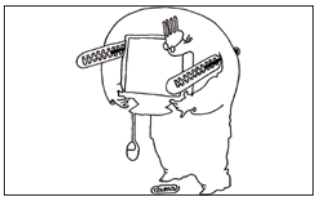
Галина Положевец,
главный редактор



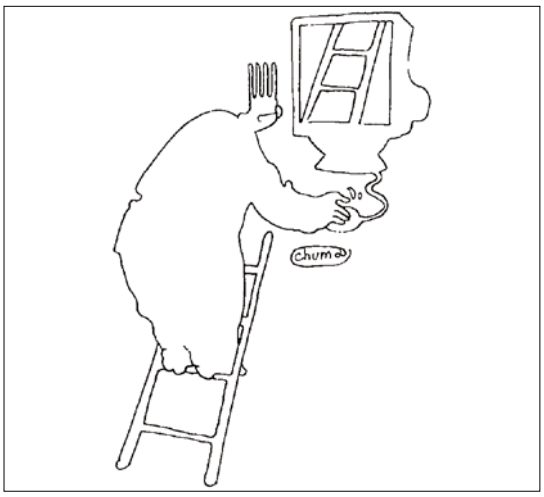
18



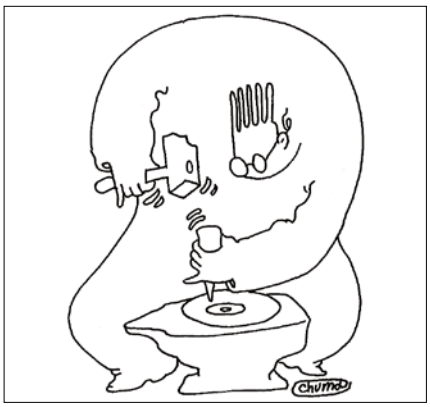
24

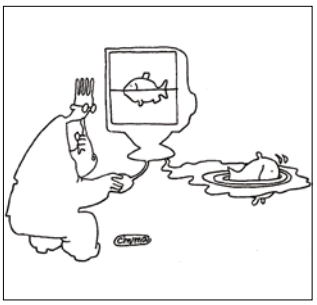


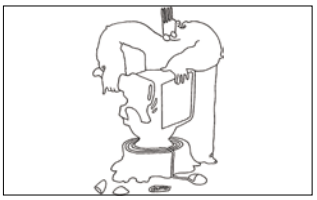
36

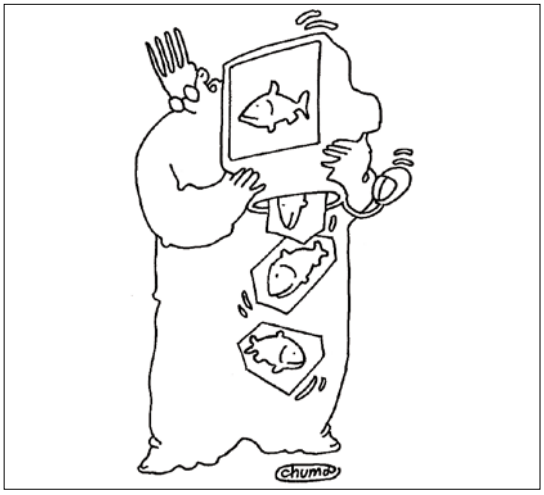


42









компании PingWin Software, автор журнала Вячеслав Калошин.

Агунда Алборова

Безопасность

Изучаем «1С»

Веб

Программирование

Ретроспектива

Семь лет с нами

Участвуй сам и расскажи друзьям! Как получить Адмиский приз?

Редакция «СА» продолжает разыгрывать Адмиский приз, и приглашает к участию новых игроков. Вам необходимо зарегистрироваться на сайте www.samag.ru и активировать код, который можно получить, купив номера журнала (№№7-12, 2009). Чем больше у вас заветных кодов, тем выше шансы стать победителем розыгрыша. Дополнительные коды смогут получить самые активные участники форума на сайте www.samag.ru.

Успехов и удачи!

Адмиский Приз

Розыгрыш будет проходить в три этапа:

I – участвуют коды из журналов №7, 8, 9, полученные с июля по сентябрь 2009 г.

II – участвуют коды из журналов №10, 11, 12, полученные с октября по декабрь 2009 г.

III – участвуют коды из всех шести номеров журнала за 2-е полугодие 2009 г.

Адмиский Приз

Юбилейная конференция Cisco Expo

Главное ИТ-событие года в России

12-14 октября в московском Центре международной торговли состоится десятая конференция по информационным технологиям Cisco Expo

Это мероприятие, превратившееся за последние годы в крупнейший ИТ-форум на территории стран СНГ, на сей раз пройдет под лозунгом «Знание – сила». Предстоящая конференция знаменательна еще и тем, что она состоится в рамках празднования 25-летия компании Cisco.

Московские форумы Cisco Expo де-факто приобрели статус ключевого события в индустрии информационно-коммуникационных технологий на территории СНГ. Прошлогодняя конференция установила новый рекорд посещаемости, собрав 2168 ИТ-специалистов, аналитиков и журналистов (более чем трехкратный рост по сравнению с Cisco Expo-2000) из 122 городов различных стран СНГ, а также из США, Ближнего Востока и Европы. Рекордной оказалась и отраслевая поддержка мероприятия – статусы спонсоров и партнеров получила 41 компания.

Организаторы нынешней конференции подготовили насыщенную программу, которая позволит ее участникам не только узнать о последних разработках Cisco и повысить свою квалификацию в сфере сетевых технологий, но и пообщаться с коллегами, а также расширить партнерскую сеть. По традиции, форум откроется приветственным словом от руководства ООО «Сиско Системс». После этого участников ожидают два пленарных доклада топ-менеджеров Cisco и выступления представителей компаний-спонсоров – NVision Group, EMC (Золотые партнеры), VMware (Серебряный партнер) и «АМТ-ГРУП» (Бронзовый партнер). Дальнейшая работа московской Cisco Expo-2009 будет проходить в рамках десяти технологических потоков. В них примут участие более 50 специалистов Cisco и представители компаний-партнеров, которые сделают около 100 докладов о решениях и технологиях Cisco и их практическом применении, а также проведут живые демонстрации продуктов и решений. По информационной насыщенности предварительная программа нынешней конференции уже превзошла показатели прошлогоднего мероприятия (в 2008 году было 7 потоков и порядка 80 докладов). Потоки Cisco Expo-2009 будут посвящены:

- сетевой инфраструктуре (использованию технологий маршрутизации и коммутации в корпоративных сетях);
- решениям для операторов связи;
- системам видеонаблюдения;
- решениям в области центров обработки данных;
- информационной безопасности;
- унифицированным коммуникациям;
- центрам обработки вызовов;
- мобильным и беспроводным решениям;
- оптическим сетям и системам;
- разработке и интеграции сторонних продуктов и решений (этот поток в программу Cisco Expo включен впервые, на нем сотрудники Cisco расскажут о совместных проектах с российскими компаниями-производителями).

Особое внимание на форуме будет уделено технологическим инновациям Cisco. Так, те, кто посетит поток, посвященный информационной безопасности, узнают о новейшей технологии фильтрации сетевого трафика, которая основана на репутационных характеристиках отправителя. Ведущие потока, посвященного центрам обработки данных (ЦОД), представят коммутаторы Cisco Nexus и унифицированную среду вычислений (Unified Computing), которые позволяют строить ЦОД нового поколения на основе технологий виртуализации, консолидации операций «ввода-вывода» и так называемых «облачных вычислений».

Кроме того, посетители конференции смогут принять участие в специально организуемых сессиях неформального общения со специалистами Cisco, а также в «круглом столе» по теме защиты персональных данных.

По сложившейся традиции, в рамках Cisco Expo будет организовано тестирование в соответствии с программой сертификации специалистов Cisco Career Certification. Это позволит любому желающему проверить свой уровень знаний в области вычислительных сетей и современных телекоммуникационных технологий (на прошлогодней конференции такое тестирование прошли более 100 человек).

Наконец, следует упомянуть, что во время работы Cisco Expo-2009 будет проходить выставка технологий и готовых к развертыванию инновационных решений. В ней примут участие компания Cisco и партнеры конференции.

Более детально ознакомиться с программой предстоящего мероприятия и зарегистрироваться для участия в нем можно на странице www.ciscoexpo.ru.

CodePlex Foundation — связующее звено между Open Source-разработчиками и коммерческими компаниями

Визитка

ДМИТРИЙ ШУРУПОВ,
ведущий рубрики

В Linux 2.6.31 улучшили скорость работы десктопов

9 сентября вышла очередная версия Linux-ядра – 2.6.31, одним из главных достоинств которой стали усовершенствования в работе на настольных ПК, позволившие повысить отзывчивость (интерактивность) работы X-сервера вдвое при высоких нагрузках на оперативную память или swap. Кроме того, появилась поддержка новой версии протокола USB (3.0), Kernel Mode Setting (KMS) для видеокарт ATI Radeon, новые инструменты тестирования и отладки (Gcov, Kmemchek. Kmemleak).

Интересно, что незадолго до этого Кон Коливас (Con Kolivas), который покинул группу разработчиков Linux-ядра около 2 лет назад, вернулся в мир Linux, объявив о выпуске нового планировщика задач – BFS (Brain Fuck Scheduler). Коливас хорошо известен в Linux-сообществе как человек, который был очень недоволен тенденциями в развитии ядра. По его мнению, явно прослеживается акцент в сторону индустриального применения Linux, в то время как миллионы обычных десктоп-пользователей из-за этого лишаются той производительности, которую могли бы получить. Именно поэтому Коливас в свое время выпускал популярные патчи к ядру, известные как ветвь -ck. Именно поэтому он вернулся для того, чтобы подарить миру новый планировщик BFS, предназначенный для использования на десктопах.

Как поясняет автор Brain Fuck Scheduler, этот планировщик получил свое название, поскольку он является невероятно простым и производительным, несмотря на эту простоту, таким образом, BFS призван помочь всем нам полностью переосмыслить то, каким должен быть шедулер операционной системы. Более подробная информация о BFS и о том, зачем он нужен, доступна в документе http://ck.kolivas.org/patches/bfs/bfs-faq.txt. EOF

Microsoft создала Open Source-организацию CodePlex Foundation

Корпорация Microsoft объявила о создании организации CodePlex Foundation, цель которой – стать связующим звеном между Open Source-разработчиками и коммерческими компаниями.

Среди задач новой организации CodePlex Foundation – обмен кодом и налаживание понимания между Open Source-сообществами и софтверными компаниями, повышение активности участия компаний в Open Source-проектах, пополнение ряда существующих Open Source-организаций новым форумом, в котором будут принимать активное участие и компании, и представители сообщества. Несмотря на схожесть названия новой организации с хостингом проектов Microsoft CodePlex, заявляется, что это независимые инициативы, объединенные общей миссией.

Единственным спонсором CodePlex Foundation на данный момент является Microsoft, однако в организации заинтересованы в поиске новых партнеров среди компаний – разработчиков программного обеспечения. Более подробные сведения об организации доступны на http://www.codeplex.org.

Впрочем, появление CodePlex Foundation вскоре встретило критику от Энди Апдегроува (Andy Updegrove), основателя ConsortiumInfo.org, известного защитника Open Source и стандартов. По его словам, в созданной Microsoft организации присутствуют серьезные проблемы, из-за которых она не сможет стать реальной силой в индустрии программного обеспечения с открытым кодом. Энди рассматривает CodePlex Foundation как организацию с жесткой структурой, где фактически вся власть отведена одной корпорации, а путь для других потенциальных участников затруднен. Отсутствие «убедительной инфраструктуры, которая позволит сообществу чувствовать себя в безопасности», по понятным причинам не будет способствовать эффективному взаимодействию организации с представителями Open Source-лагеря – именно в этом и заключается основной ее недостаток. Апдегроув привел и некоторые рекомендации, которые должны улучшить позиции CodePlex Foundation, – так, например, в совете директоров должно быть не менее 11 представителей, причем все – от разных компаний. EOF

Поступили в продажу новые SCM-устройства Panda GateDefender Performa серии 9000

Panda GateDefender Performa серии 9000 предоставляет самую мощную, надежную и легкую в использовании защиту периметра корпоративной сети, позволяя в рамках всех самых распространенных протоколов проверять входящий и исходящий трафик на все виды контентных угроз, а также выявлять зомбированные компьютеры (боты), расположенные внутри корпоративной сети.

В октябре в России начинаются продажи трех моделей устройств из новой серии: 9100, 9500 Light и 9500 Large, которые в зависимости от своей производительности рассчитаны на средние и крупные компании (до 500, 1000 и 2500 пользователей на одно устройство соответственно). Каждая модель объединяет три защитных модуля: антивирусный модуль, включающий в себя контент-фильтр; модуль антиспам; и модуль веб-фильтрации для ограничения доступа сотрудников к нежелательным сайтам и блокировки приложений P2P и IM. Модули продаются как в комплексе, так и отдельно.

Более того, устройства являются масштабируемыми: они поддерживают режим балансировки нагрузки, что позволяет создавать кластер параллельно подключенных устройств для защиты сети, состоящей из большего количества пользователей.

Новые устройства Panda GateDefender Performa работают на платформах компании Sun Microsystems, что гарантирует высокую производительность фильтрации (до 700 МБит/сек для HTTP, и до 520 сообщений/сек – для SMTP) и высокую надежность устройств. Подробнее о продукте: www.viruslab.ru/products/gdperforma. EOF

Визитка

ЭДУАРД ДОЛГАЛЕВ, выпускник МГУ и MBA Высшей школы международного бизнеса АНХ, к.т.н, руководитель направления по продажам ПО WebSphere IBM в России и СНГ

SOA. Взгляд шире!

Это не просто набор стандартов и технологий

Многие относятся к SOA-технологиям настороженно, если не сказать отрицательно. Но SOA – не миф. Это новый взгляд на работу и место ИТ в бизнесе

Что скрывается за SOA?

SOA (Services Orientated Architecture) – сервисно-ориентированная архитектура построения информационных систем. Этот подход основан на выделении сервисов (функций) и последующем их повторном использовании. Выделять общие части программного кода, скрывать логику выполнения функции за названием, а затем ее многократно использовать – это первое, что придумали программисты и продолжают активно практиковать до сих пор. Такой подход существенно сокращает затраты на техническую поддержку, модификацию информационных систем.

По моим наблюдениям с наибольшим недоверием к SOA относятся специалисты, которые имеют какой-либо опыт программирования. Для них выделять сервисы, а затем их повторно использовать – прописные истины. Но почему тогда так много внимания уделяется SOA сегодня? Почему это обсуждается, как некий революционный подход?

Попробуем разобраться.

Действительно, SOA основано на выделении общих частей информационных систем, оформлении их в сервис и повторном использовании сервисов. Очевидно, что повторное использование существенно экономит средства компании.

Другое важное завоевание SOA состоит в следующем. SOA позволяет бизнес-пользователям информационных систем получить доступ непосредственно к программному коду, написанному программистом.

Если раньше функции, написанные программистом, были спрятаны в упаковку черного ящика откомпилированного программного кода, то сегодня они приобрели для бизнес-пользователей вполне конкретные очертания, с ними стало возможно работать. Сегодня бизнес-пользователь сам может определять, когда и как будет вызвана функция (сервис), написанная программистом. При этом на экране бизнес-пользователь видит не программный код функции, а симпатичную картинку, которую он перемещает по экрану, вбивая данные в удобные экранные формы графического редактора.

Необходимо отметить, что цена за такое удобство – дополнительный программный код, который будет автоматически создан графическим редактором. Как следствие, информационные системы работают медленнее, чем, если эти изменения были бы проведены программистом непосредственно в коде. Раньше дополнительные издержки были недопустимы и ограничивались производительностью вычислительных машин. Сегодня, на фоне значительного роста мощности аппаратных ресурсов, такие издержки стали более чем незначительны.

Итак, бизнес дотянулся до программного кода. Бизнес-пользователь получил инструмент для работы с программным кодом. Теперь бизнес-пользователь может самостоятельно определять и изменять согласованную работу информационных систем (сервисов). Это, по моему мнению, и есть главное завоевание SOA.

Однако важно понимать, что речь не идет обо всем программном коде! Бизнес-пользователю интересны только те функции (сервисы), которые обеспечивают выполнение конкретных бизнес-задач. Например, «открытие счета», «пополнение счета», «отправление SMS-сообщения» и т.д. Бизнесу неинтересен программный код, который скрывается за этими сервисами. Бизнесу важно, что при вызове сервиса в информационных системах компании будут выполнены корректные операции, соответствующие данной бизнес-операции.

Чтобы понять значимость этого, необходимо оторваться от технологий и посмотреть на ИТ с точки зрения бизнеса. Он сегодня все чаще сталкивается с ситуацией, когда скорость вывода новых услуг или продуктов определяет успешность компании на рынке и победу над конкурентом.

При этом гибкость и маневренность бизнеса во многом определяется гибкостью и готовностью к изменениям информационных систем и ИТ в целом. Неготовность ИТ к быстрым изменениям становится сдерживающим фактором при завоевании компанией новых рынков.

Таким образом, с одной стороны, технологии готовы предоставить бизнесу новые рычаги для управления, а с дру-

гой – бизнес сегодня крайне заинтересован в увеличении гибкости и маневренности на высоко-конкурентных рынках, ему необходимы новые идеи и инструменты для управления. Все это объясняет рост интереса к SOA.

Единая виртуальная информационная система компании

SOA позволяет сопоставить бизнес-задачи с функциями программного кода информационных систем. Для того чтобы заработало SOA, в компании разворачивается программное обеспечение, которое формирует дополнительный слой над всеми информационными системами компании. В этом слое с помощью сервисов (бизнес-задач), определяемых бизнес-пользователями, формируется единая виртуальная информационная система. В этом слое формируются новые услуги и продукты компании.

Дополнительный слой позволяет отделить жизненный цикл новых услуг от жизненного цикла самих сервисов и информационных систем. С появлением SOA компания перестает быть зависимой от поставщиков информационных систем. Логика каждой новой услуги теперь не является частью информационной системы, она становится достоянием компании. Отныне поставщикам информационных систем раздаются задания не на реализацию новых услуг, а на предоставление сервисов.

Влияние SOA на организационную структуру

Как было отмечено выше, не все функции, написанные программистом, нужны бизнесу на уровне сервисов. Необходимо выделить сервисы, которые будут востребованы. Поскольку дополнительный слой формирует единую виртуальную информационную систему, то, очевидно, для этого слоя помимо определения сервисов необходимо подготовить и свою модель данных, понятную бизнесу и реализованную на языке бизнеса. Причем разработка модели данных часто не уступает в сложности задаче выделения сервисов. Таким образом, мы определили еще одну важную составляющую SOA – это модель данных.

Итак, две составляющие: сервисы и модель данных, от которых будет зависеть успех использования SOA в компании. Мы подошли к важной роли SOA – роли архитектора, на которого возлагается задача отделить «зерна от плевел»: правильно определить модель и форматы данных, принять решение, какие функции необходимо поднимать до уровня бизнеса и т.д. Устойчивость системы, ее жизнеспособность теперь во многом зависит от архитектора. Зафиксированные архитектором сервисы и данные становятся стандартом для вашей компании. При выборе новой информационной системы эти стандарты превращаются в требования к взаимодействию новой системы с информационной системой компании.

Роль архитектора возрастает и в глазах бизнеса. От архитектора теперь зависит, насколько удобными для бизнеса будут новые рычаги управления. Понимая это, во многих компаниях даже создаются специальные комитеты, где принимаются решения об изменении модели данных и сервисов.

Теперь становится понятно, что использование SOA не только привносит новые технологии, но и может влиять на изменения в организационной структуре компании.

Смещение акцентов с систем на сервисы

Помимо организационных изменений SOA смещает акцент с монолитных информационных систем в сторону сервисов. Внимание бизнеса переключается с названий и поставщиков информационных систем на качество и доступность нового сервиса. Теперь компания может работать не с лучшими поставщиками информационных систем, а с лучшими поставщиками сервисов. Это могут быть небольшие компании, но лучшие в предоставлении конкретного специализированного сервиса. Бизнес получает прекрасный инструментарий, ИТ и бизнес получают большую маневренность в работе с поставщиками.

Происходит важное изменение и в мировоззрении сотрудников: центры компетенции и ответственности начинают смещаться от крупных монолитных систем в сторону сервисов, качества предоставления сервисов и т.д.

Некоторые компании идут дальше, бизнес ранжирует сервисы по степени важности, определяет и контролирует соглашения об уровне качества предоставления сервисов (SLA). Самые передовые компании даже начинают считать доходность сервисов.

Теперь бизнес-пользователь может самостоятельно определять и изменять согласованную работу информационных систем (сервисов). Это и есть главное завоевание SOA

Таким образом, новые технологические решения, пришедшие вместе с SOA, обеспечивают большую гибкость бизнеса, сокращают время вывода новых услуг, позволяют больше зарабатывать и отвоевывать новые рынки. При этом меняются взаимоотношения бизнеса и ИТ, меняются правила работы с поставщиками систем, меняется даже организационная структура компании. Все перечисленное, на мой взгляд, и является причиной такого пристального внимания к SOA. И часто, обсуждая SOA, подразумевают все или часть вышеперечисленных изменений.

Требуйте SOA!

Нужно ли SOA вшей компании? Предположим, разработка, модификация систем осуществляется внешней командой. Вы ставите задачи, определяете правила приемки работы. В результате в заранее определенные сроки получаете результат. У вас отлаженный процесс работы, предсказуемые сроки и понятный процесс вывода новых услуг. Зачем вам использовать SOA?

Если в вашей компании не обсуждают SOA-стратегии, это не означает, что SOA бесполезно и вам ни к чему в этом разбираться. На самом деле, SOA в вашей компании уже есть. У вас уже есть сервисы, которые повторно используются, есть модель данных для согласованного взаимодействия серви-

Трудности есть, но они преодолимы

В 2006 году, при построении SOA в банке Ренессанс Капитал (Ренессанс Кредит), где в то время я занимал должность главного архитектора ИТ, мы сталкивались со следующими трудностями. Во-первых, сложно было убедить бизнес в необходимости трансформации в соответствии с подходом SOA. Но пока бизнес не поймет преимуществ SOA и не станет деятельно сотрудничать с ИТ, построение SOA невозможно. Стратегическую цель перехода к SOA определил тогдашний Chief Operating Officer Вольфганг Хайнрих. Однако основную роль пропагандиста SOA-подхода к построению ИТ-архитектуры в банке взял на себя Ярослав Медокс, бывший директором департамента развития ИТ. Благодаря их деятельности стал возможен постепенный переход к SOA-архитектуре в банке. Вторая проблема – кадры. Для построения SOA требуются квалифицированные сотрудники. Сначала мы привлекли специалистов компании «Неофлекс». Постепенно экспертиза появлялась и у сотрудников банка. В-третьих, закономерны технические сложности. Для нас они состояли в том, продуктовая линейка IBM WebSphere ESB V6 и IBM WebSphere Process Server V6 была еще довольно «сырой». Однако нам удалось впервые в России построить кластер WebSphere ESB и WebSphere Process Server, в так называемой, золотой топологии.

Алексей Букавнев,
IBM Certified SOA Solution Designer

сов, реализован механизм поддержки жизненного цикла сервисов и т.д.

Где же спрятано SOA? Искать его необходимо в новых черных коробках или адаптированных под вашу компанию модулях информационных систем, которые приносят ваши поставщики. Там уже есть сервисы (функции) и они уже хорошо стандартизованы и активно переиспользуются.

С каждой новой доработкой стоимость модификации системы становится дешевле за счет повторного использования сервисов. С каждой новой доработкой ваш поставщик все меньше тратит время на разработку новых сервисов, все больше акцент смещается в сторону лишь настройки согласованной работы уже существующих сервисов.

У поставщика вашей информационной системы снижаются издержки, одновременно растет маржа выполняемых работ. При заключении нового договора на модификацию системы SOA работает и помогает зарабатывать. Но не вам, а поставщику! Для вас мир SOA закрыт черной коробкой, которую вы получаете после выполнения очередного договора.

Теперь посмотрим, что происходит в вашем ИТ. Получить решение, пусть и в черной коробке, в фиксированные сроки с ожидаемой функциональностью – это отличный результат. Этот подход позволяет решить текущую задачу бизнеса. Это вполне приемлемо в краткосрочной перспективе. Но в долгосрочной перспективе с каждой новой черной коробкой вы все больше попадаете в зависимость от поставщика. Информационные технологии изменчивы. Меняются лидеры рынка информационных систем, появляются новые классы специализированных систем. Если ваша компания стремится стать лидером рынка или остаться работать на высококонкурентном рынке, ваши бизнес-лидеры должны постоянно что-то изобретать и предлагать своим клиентам. А это неминуемо требует внедрения новых специализированных решений и систем.

Рано или поздно, наступит момент, когда от каких-то черных коробок придется отказаться. Но какова будет цена такой замены? Что, если пара черных коробок, требующих замены, упакована в один металлический контейнер (монолитной информационной системы), который вы так долго финансировали? Из-за пары коробок в утиль пойдет все, включая те коробки, которые устраивают и обеспечивают работу части вашего бизнеса, не требующего модификации сегодня.

Что останется у вас после отправки контейнера в утиль? Бизнес-процессы? Сервисы? Модель данных? – Ничего! Ваше время, потраченное на обсуждение, проектирование бизнес- или технологических процессов, уникальные идеи и находки, также отправится в утиль, вслед за контейнером. Помимо внедрения новой функциональности теперь вам потребуется восстановить прежнюю, которая была уже однажды адаптирована для вас прежним поставщиком системы. Стоя перед блестящим новым контейнером с новыми коробками вам придется начать работу с чистого листа!

Разве это плохо? Ведь это вызов для ИТ-специалистов, это новая творческая работа для ИТ-специалистов. Это новые ИТ-бюджеты, новый штат сотрудников.

Это плохо для бизнеса! Решая краткосрочные задачи, ваш бизнес лишил себя возможности повторно использовать потраченное уже однажды время, уникальные идеи и находки, однажды уже реализованные. Стоя перед новым контейнером, ваш бизнес очутился на пороге революции, размеры и сроки которой зависят от размера контейнера. Сможет ли ваш бизнес пережить эту революцию? Хватит ли средств на замену? Как скажется на бизнесе потеря части функциональности или временная их недоступность?

Всего этого можно было избежать, если бы из каждой коробки к информационной системе компании вел интерфейсный шнур – это и есть сервис. Но готов ли пойти на это поставщик монолитной системы? Передать сервис клиенту, означает, что теперь клиент будет повторно использовать сервис, а новые услуги могут уже создаваться без привлечения поставщика системы. Для поставщика это означает существенную потерю доходов. Тем не менее, в нынешней экономической ситуации дробление монолитных систем на сервисы происходит. Это происходит с поставщиками, которые еще несколько лет назад не хотели даже слышать об этом. Требуйте SOA! Требуйте от поставщиков реализации сервисов, а не услуг!

А где деньги?

Общаясь со специалистами из различных индустрий, для ответа на этот вопрос я всегда рекомендую обратиться к графику длинного хвоста. Предположим, мы смогли перенумеровать все социальные и другие разновидности групп своих клиентов. Предположим, мы аналогично пересчитали всевозможные услуги, которые ваша компания оказывает и могла бы оказывать. Строим график, где по вертикальной оси откладываем количество людей в группах, по горизонтальной оси наносим пронумерованные услуги по мере убывания их популярности у ваших клиентов. В итоге получится гипербола (см. рис. 1). А теперь задайте себе вопрос: где сегодня зарабатывает ваша компания? Ваши услуги заканчиваются недалеко от начала координат, а площадь под гиперболой от начала координат до самой непопулярной вашей услуги – и есть зарабатываемые вами сегодня деньги.

Площадь под кривой, которая прижимается к горизонтальной оси – это доходы, которые остаются без вашего внимания. И нетрудно понять, что они соизмеримы с вашими текущими доходами.

Аналогичную кривую можно нарисовать для соответствия бизнес-процессов и количества людей, которым необходима их автоматизация. Готова ли ваша компания запустить проект по выводу услуг для сообщества социальной среды, состоящей из десяти человек? Готова ли ваша компания начать проект по автоматизации бизнес-процесса для трех специалистов маркетинга, когда запускается тестовая маркетинговая компания на две недели?

При традиционном подходе это окажется нерентабельным: расходы на проектную команду, проектирование, тестирование... И главное, сроки. Если новый бизнес-процесс не заработает в течение нескольких дней, это потеряет актуальность.

Инновации IBM в SOA

Сегодня IBM предлагает SOA-решения, которые позволяют компаниям охватить бизнес из длинного хвоста: управления бизнес-правилами – WebSphere ILOG BRMS, система корреляции различных событий в нескольких системах – WebSphere Business Events, система построения ситуационных систем на основе виджитов – WebSphere sMash.

Продукты, составляющие основу линейки WebSphere для построения SOA, постоянно совершенствуются. Особое внимание уделяется сокращению времени разработки и модификации. Например, в WebSphere BPM появилась технология Business Spaces, основанная на виджитах. Теперь бизнес-пользователь в браузере с помощью мышки сам может собирать удобный для него интерфейс, без привлечения программиста. В WebSphere BPM бизнес-пользователь получает возможность самостоятельно спроектировать не только экранные формы, которые потом будут автоматически созданы, или бизнес-процессы, но и возможность самостоятельно отлаживать и прогонять бизнес-процессы.

Помимо программного обеспечения IBM предлагает для построения SOA фреймворки для различных индустрий с готовой моделью данных, бизнес-процессами и даже прописанными сервисами. Для отдельных индустрий IBM предлагает платформы, которые существенно расширяют даже модель бизнеса. Например, для «Телеком» – это платформа SDP (Service Delivery Platform), которая позволяет любому желающему создавать программы, где используются сервисы «Телеком-оператора». Тем самым с помощью программистов-энтузиастов Телеком-оператор охватывает интересы небольших групп клиентов.

Можно ли прикоснуться к SOA?

Да! Сегодня IBM предлагает аппаратное решение – семейство WebSphere DataPower SOA Appliances. Это совершенно новый класс решений. Устройства на аппаратном уровне оптимизированы для работы с сервисами, для их маршрутизации, трансформации и обработки.

Решения из семейства WebSphere DataPower SOA Appliances обеспечивают работу с цифровыми подписями, шифрованием трафика и отдельных частей передаваемой информации, реализует аутентификацию и авторизацию. Устройства обеспечивают проксирование сервисов, ведение SLA и мониторинг. Помимо многофункциональности хотелось бы выделить основные особенности семейства устройств:

- производительность;
- простота в использовании и обслуживании;
- быстрое внедрение;
- быстрый возврат инвестиций.

Новое семейство устройств быстро завоевало признание специалистов. WebSphere DataPower SOA Appliances обладатель ряда призов. Сегодня IBM предлагает уже пять аппаратных решений из этого класса.

К различным характеристикам SOA теперь добавляется вес и цвет!

***

Надеюсь, в этой статье мне удалось пролить свет на SOA и показать, что это не просто набор стандартов и технологий. В первую очередь, это новый подход к взаимодействию бизнеса и ИТ, новый взгляд на работу и место ИТ в бизнесе. За стандартами и технологиями, реализующими SOA, тянутся изменения в организационной структуре компании, смещаются акценты при построении ИТ, меняются правила работы с поставщиками информационных систем. Перед компаниями открываются возможности построения новых моделей бизнеса. EOF

Рисунок 1. График длинного хвоста

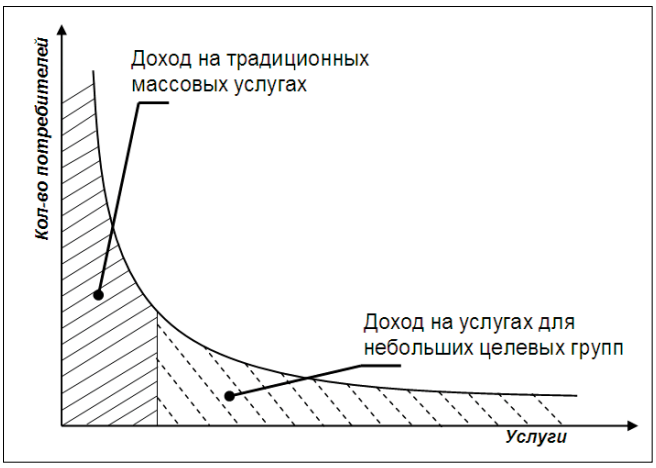


Рисунок 2. WebSphere DataPower SOA Appliances

Визитка

ВЛАДИММИР ЭНГЕЛЬС, выпускник ЮРТУ. Участвует в проектировании и реализации распределенных и SOA-систем. Ведущий системный архитектор SOA/BPM в московском представительстве компании Oracle

Это не панацея от бед!

О чем забывают при создании SOA-систем

Как подход к проектированию, SOA существует уже много лет. Однако в новых проектах специалисты сталкиваются с одними и теми же проблемами, что приводит к серьезным сложностям для бизнеса

А нужно ли вообще SOA бизнесу?

Бизнесу безразлично, какие технологии и продукты использует ИТ-подразделение, если это удовлетворяет потребности бизнеса. SOA-технологии не удовлетворяют напрямую ни одну из потребностей бизнеса. Поэтому, бесполезно рассматривать их в качестве «лекарства от всех болезней» бизнеса.

Более того, существует очень серьезный барьер «на входе в SOA», что проявляется в необходимости привлечения немалых финансовых, людских и временных ресурсов. А это, в свою очередь, перечеркивает возможность использовать SOA, как «инструмент экономии».

Но SOA может быть успешно использована для повышения эффективности бизнеса, если рассматривать эффективность более широко. Правда, универсальных средств быть не может, по причине того, что понятие эффективности зависит от целей компании, а у каждой компании они свои.

Однако, даже имея схожие цели, в текущей непростой ситуации разные компании могут выбрать разные стратегии достижения своих целей. Рассматривая потенциальные стратегии, разделим их условно на две группы: пассивные и активные.

Причем реализация активных стратегий практически всегда требует современных информационных технологий. Поэтому можно утвердительно ответить на поставленный выше вопрос: «Нужна ли сервисно-ориентированная архитектура бизнесу?». Если у компании есть силы, то новые SOA-проекты необходимо начинать именно сейчас, когда основная масса конкурентов это не приемлет. Такой подход позволит компании создать конкурентное преимущество и продвинуться вперед.

Для многих российских компаний сложилась уникальная ситуация. В России не так сильно, как в других странах, актуален «груз унаследованных систем». Использование инноваций – это единственный шанс стать технологическим лидером в свой отрасли.

Этап подготовки к проекту

Еще до инициализации проекта необходимо определиться с некоторыми вопросами.

Место SOA в стеке технологий

При реализации SOA-решений необходимо понимать, что основу этих решений составляет, так называемое, промежуточное программное обеспечение (промежуточное ПО). Само по себе оно не может являться целью бизнеса. Это необязательный компонент системы. В то же время это эффективный помощник для решения некоторых задач, для организации вычислительного процесса и для построения прикладных приложений и систем.

Но ничто не бывает бесплатно. За использование промежуточного ПО необходимо платить налог – либо финансами, либо производительностью, либо аппаратным обеспечением, либо чем-то другим. И это важно понимать, вступая на путь SOA.

Почему вокруг SOA так много названий, технологий и продуктов?

Еще одно важное отличие промежуточного ПО от других видов ПО – большой спектр решаемых задач. Отсюда появление большого количества компонентов и технологий относящихся к промежуточному ПО. И это число постоянно растет с развитием предметной области. Но далеко не все технологии и компоненты нужны в одном проекте. На практике для каждого решения желательно подбирать конкретный состав продуктов и технологий. Проблему выбора нередко серьезно затрудняет «маркетинговый шум».

Технологии ради технологий

Поскольку использование подхода SOA это всегда издержки, то каждый руководитель должен задать себе и членам команды вопрос: «Почему в данном проекте необходимо использовать именно SOA?» Если ответы будут неточными, общими, или вообще лежать в области «верю/не верю», то это первый признак того, что SOA-технология выбрана ради самой технологии. Такое использование смысла не имеет, хотя это еще часто встречается на практике. Для бизнеса важно лишь то, как ПО соотносится с бизнес-

целями. И если на данном этапе это не выяснить, то на следующих этапах уже будет очень сложно обосновать большие издержки.

SOA и моделирование бизнеса

Но чтобы понять, как соотносится будущее SOA-решение, к разработке которого приступает компания, с целями бизнеса, необходимо, как минимум, определиться с этими целями, а как максимум – смоделировать аспекты бизнеса, нужные для достижения поставленных целей, в том числе и создать модель процессов и сервисов деятельности компании.

И здесь SOA лучше других подходов вписывается в управление бизнес-процессами, предоставляя только необходимые сервисы и ничего лишнего.

А сколько это стоит?

Часто, чтобы определить цену, необходимо проделать длинный путь. Вначале определить проблему, далее описать задачу, после этого выбрать под задачу необходимые технологии и продукты. И только, когда уже будет определена архитектура будущего решения и диаграмма его развертывания на аппаратном обеспечении, только тогда можно предметно говорить о цене. Чем меньше сделано на этом пути, тем менее обоснованными будут стоимостные оценки SOA-решения. Быстрее этот путь можно пройти только в одном случае, если задача и ее решение хорошо известны, а стоимостной анализ на практике проводился уже много раз.

Какова будет производительность?

Проблема заключается в том, что универсальных единиц измерения производительности вычислительных систем нет. Производительность – это комплексная характеристика всей системы, которая зависит от множества компонентов. Вот некоторые из них:

- конкретная совокупность задач на входе в систему и их соотношение (смесь задач);
- организация вычислительного процесса и используемые алгоритмы;
- используемые проектные решения, технологии и продукты;
- системное и промежуточное программное обеспечение;
- аппаратное обеспечение;
- а иногда и регламенты, относящиеся к выполняемой задаче.

Если две системы отличаются хоть одной компонентой, то их показатели производительности (например, количество обработанных задач на выходе системы в единицу времени) уже нельзя сравнивать.

На практике же сплошь и рядом встречается ситуация, когда на тестовых стендах, конфигурация которых существенно отличается от конфигурации будущей системы, снимаются «какие-то» характеристики производительности, причем делается это на смеси задач, абсолютно не относящихся к будущей системе. И тут же на основе этих снятых характеристик делаются далеко идущие выводы о производительности будущей системы.

Перечислим только некоторые проблемы такого подхода:

- тратятся ресурсы на бесцельное тестирование;
- на основе недостоверных данных может быть остановлен успешный проект;
- если проект не остановлен, то он существенно задерживается;
- формируются некорректные ожидания относительно характеристик промышленного SOA-решения.

Существует барьер «на входе в SOA» — необходимость привлечения немалых финансовых, людских и временных ресурсов

Как организовать разработку SOA-решений?

На этапе подготовки проекта существует еще и конфликт интересов с поставщиками SOA-решения, в случае, если оно заказывается на стороне. (Такого рода конфликт интересов, на самом деле, возникает и внутри организации, когда SOA-решение заказывается внутри компании. Но в этом случае он менее выражен, хотя и не менее болезненный.)

Данный конфликт интересов заключается в следующем. Заказчику, еще до начала проекта, желательно знать функциональные, нефункциональные и стоимостные характеристики будущей SOA-системы. В то же время, ввиду уникальности большинства SOA-решений, создаваемых под конкретные задачи и рассчитанные на работу в уникальном ИТ-окружении заказчика, поставщик не может предварительно предоставить такого рода информацию. А, проводя какие-либо исследования, оценки и другую предварительную проработку за свой счет, поставщик понесет существенные риски, почему и не идет на это.

Решение кроется в организации процесса разработки SOA-систем, которое упрощенно представлено в таблице ниже. На первом этапе разрабатывается концепция SOA-решения, которая проходит проверку, как правило, внутри виртуальной машины или на ноутбуке.

На втором этапе выбирается «наиболее характерный полигон», на основе которого реализуется пилотный проект. Пилот уже может использовать существующие системы, но ни в коем случае не «боевые», а «тестовые» экземпляры. Всегда возникает искушение «почти работающий пилот» запустить в эксплуатацию. Этого категорически нельзя делать, поскольку цели пилота противоречат целям промышленного проекта. Его задача – выяснить принципиальную жизнепригодность SOA-решения, в связи с чем, для уменьшения издержек на пилотное проектирование, могут быть сделаны существенные упрощения, например, в структуре системы.

И только когда проанализированы результаты пилота, и он признан удачным, можно с открытыми глазами начинать промышленный проект.

Распределение времени и средств по этапам складывается примерно следующее (см. таблицу 1). Как можно видеть, такой подход удобен всем:

- есть возможность концептуального понимания решения задачи без необходимости нести издержки;

- есть возможность провести предварительную, глубокую проработку уникального решения;
- стоимость предварительной проработки отлична от нуля, что позволяет поставщику привлечь высококвалифицированные ресурсы;
- в то же время, заказчик рискует лишь около 10% средств, а не всем бюджетом проекта, но получает за это полную картину относительно будущего SOA-решения;
- промышленное проектирование ведется с учетом накопленного опыта, полученного на этапе пилотного проектирования.

Этап проектирования

Теперь, когда принято решение о проектировании SOA решения, когда начинается закладываться фундамент будущей SOA-системы, важно избежать самых дорогостоящих ошибок – ошибок этапа проектирования.

Модель мастер-данных

Среди всей совокупности данных, используемых в компаниях, есть определенная специфичная категория (15-20% от всего объема), которая используется как «язык информационных систем», лежащий в основе самого бизнеса компании. Такие данные называют мастер-данными. Первая задача, стоящая перед архитектором (или аналитиком) при построении и/или развитии информационной системы, выделить (идентифицировать) мастер-данные и создать на их основе модель мастер-данных. Модель мастер-данных включает в себя следующие компоненты (но ими не ограничивается):

- определение типов данных (бизнес-объектов, справочников, вспомогательных данных и технологических данных);
- определение событий;
- определение исключительных ситуаций;
- определение интерфейсов.

В SOA-системах модель мастер-данных – это межсистемный язык, со своим, как правило, очень длительным, жизненным циклом и желательно с версионностью (смотрите, например, «Методы работы с моделью мастер-данных в SOA-проектах», http://soa.it-consultants.ru/?q=node/4). От того, насколько тщательно спроектирован этот язык, очень многое зависит, так как модель данных будет непосредственно использоваться во всех без исключения прикладных проектах.

SOA-системы: распределенные и многопоточные

SOA-системы уже не являются сосредоточенными программами, они по сути – распределенные, параллельные приложения, состоящие из множества независимых компонентов. Поэтому здесь нельзя применять традиционные подходы и алгоритмы для сосредоточенных систем. То, что тривиально решается в рамках сосредоточенных систем, например, передача параметра или перехват ошибок, в распределенных и многопоточных системах требует уже определенных усилий. Именно поэтому перед началом проектирования SOA-систем стоит освежить подходы проектирования, как распределенных и многопоточных, так и событийно-ориентированных систем.

Взаимодействие с внешним миром

В параллельных системах часто возникает еще одна проблемная область – организация взаимодействия с внешней средой. И поскольку логика такого взаимодействия может быть очень сложной, то имеет смысл выделить отдельный класс интерфейсных задач (сервисов), которые будут инкапсулировать такую логику. Хорошим примером такого подхода может служить набор из шести веб-сервисов продукта Oracle SOA Suite для организации интерфейсного взаимодействия с пользователем – Human Workflow Services.

Проблема взаимного исключения

Во всех параллельных и многопоточных системах, включая и SOA-системы, особенную остроту приобретает проблема организации совместного доступа к разделяемым ресурсам. А каждый сервис или произвольная их совокупность потенциально может стать таким разделяемым ресурсом. Для решения этой проблемы необходимо реализовывать механизмы синхронизации. Но о таких механизмах проектировщики часто забывают, полагаясь, что инфраструктура магическим образом прочитает мысли проектировщика и сама разрешит конфликт за ресурс.

В классическом решении проблемы используются семафоры, предложенные Дейкстрой еще в 1968 году. В SOA-решениях для конкретного ресурса (ресурсов) обычно используют менеджер транзакций и/или выделенный сервис, инкапсулирующий логику управления транзакцией, которые могут использовать широкий арсенал хорошо известных алгоритмов.

Интерфейс с пользователем

Еще одна область, о которой часто забывают проектировщики – удобство пользователей, доля которых в выполнении операций может доходить до 100% в SOA-системах с широким участием человека в выполняемых операциях. Именно принцип SOA позволяет клиентским приложениям абстрагироваться от того, как выполняется операция сервиса. Поэтому потенциально за каждым сервисом может находиться человек и удобство его работы нельзя игнорировать.

Этап разработки

Он является одним из самых технологически проработанных. В то же время до сих пор много SOA-проектов реализуется, так сказать, «на коленке». Хотя отсутствие стандартов, технологических и программных средств для автоматизации процессов разработки уже давно не является проблемой. Также с удивлением хочется отметить практически полное отсутствие интереса к средствам поддержки автоматического тестирования. Хотя такие возможности тоже широко представлены.

Таблица 1. Оптимальные этапы реализации SOA-решений

Этап	Продолжительность	Относительная стоимость для заказчика
Концептуальное проектирование (Proof-of-Concept)	2-6 недель	~ 0%
Пилотное проектирование (Pilot)	1-9 месяцев	~ 10%
Промышленное проектирование (Project/Program)	0,5-3 лет	~ 90%

Этап внедрения

На этапе внедрения SOA-системы подстерегают новые сложности. Так, поставляя гибкую, конфигурируемую SOA-систему, необходимо поставлять заказчику и регламенты ее изменения, а возможно и автоматизировать процессы, поддерживающие такие регламенты. То есть каждый заказчик, использующий гибкую SOA-систему должен, либо стать маленькой ИТ-компанией, вне зависимости от отрасли, в которой он работает, либо отдать конфигурирование системы на аутсорсинг.

Еще одна проблема внедрения, которая часто разрушает самые перспективные SOA-проекты – абсолютная непроработанность вопросов обучения конечных пользователей. А ведь именно от их мнения о системе зависит ее будущее.

Этап эксплуатации

Рассмотрим важные моменты при эксплуатации проекта.

Нужно ли гибкое конфигурирование и настройка в SOA-решении?

В первую очередь хочется выделить парадоксальную тягу ряда компаний к «возможностям менять характеристики на работающей системе». И это при тотальном игнорировании необходимости тестировать внесенные изменения и абсолютной невозможности организовать такой процесс тестирования внутри компании.

Самые известные средства данного класса решений – это поддержка бизнес-правил, изменение маршрутизации запросов средствами шины сервисов и формирование «на лету» цепочек утверждения документов. Все эти средства есть и поддерживаются, но они требуют более чем тщательного и продуманного подхода к использованию. Только представим себе, к каким последствиям может привести неправильное бизнес-правило о предоставлении скидки?! А ведь большинство таких ошибок не будут видны без специальных средств диагностики.

Поэтому для того чтобы пользоваться гибкой системой, в первую очередь гибким должен быть сам бизнес. В противном случае эта функция будет в лучшем случае не задействована, а в худшем – создавать только проблемы на этапе эксплуатации.

«Сильное взаимодействие» или «Слабое взаимодействие»

Еще одно неудобство, возникающее на этапе эксплуатации, вызвано слишком широким увлечением сильным взаимодействием систем, причина которого в проектировании SOA-решения как сосредоточенной системы. Использование сильного взаимодействия приводит к образованию избыточного количества жестких связей, что сводит на нет все усилия по достижению общей гибкости решения, за счет применения SOA. Второе следствие слишком широкого использования сильного взаимодействия приводит к практически полному выпадению из поля зрения проектировщика такого понятие как событие. Мало кто проектирует события в SOA-системах..., мало кто их анализирует... В то же время случаются ситуации, когда события невозможно игнорировать, например, асинхронные исключительные ситуации.

Но, вместо того, чтобы еще на этапе проектирования в систему заложить средства контроля и обработки таких событий и просто обработать их в процессе исполнения, разработчики предпочитают построить в своих системах хитроумные конструкции для компенсации нештатной ситуации или «вычисляющие» события по ряду следов, оставленных в системе.

24*7 или 8*5?

Практически каждое техническое задание, где есть требование высокой готовности, требуется готовность на уровне 24*7. Причем одни хотят высокую готовность, выражаемую как 99,9, другие как 99,99, а некоторые и как 99,9999. Получается своего рода негласное соревнование. А ведь конечная цена SOA-решения в зависимости от числа этих девяток увеличивается по экспоненциальному закону.

При анализе же задачи часто убеждаешься, что заказчику достаточно и скромной величины 8*5. В крайнем случае, с учетом шестидневной рабочей недели и при работе во всех часовых поясах России это будет 16*6. Конечно же, режим 24*7 нужен для некоторых компаний, но уж точно не для всей SOA-системы. И здесь очень важно еще на этапе проектирования провести группировку участков системы, к которым предъявляются различные требования по уровню поддержки высокой готовности.

Этап вывода из эксплуатации

Про этот этап забывают практически в 100% проектов. А ведь просто введение понятия определенного срока эксплуатации для разрабатываемой системы (или ее версии) может снять много неопределенностей еще на этапе проектирования. Ввиду отсутствия явного срока эксплуатации, у проектировщика нет возможности провести оптимизацию с учетом этого параметра. И, как мне кажется, отсутствием срока эксплуатации прикрывают то, что никем не проводился анализ деятельности бизнеса на перспективу, а, следовательно, и анализ потребности в информационных технологиях и их характеристиках. EOF

Таблица 2. Продукты Oracle для всех этапов проекта

Задача	Продукт Oracle
Моделирование бизнеса (бизнес-цели, бизнес-процессы и пр.). Увязка бизнес-целей и сервисной модели	Oracle Business Analysis Suite
Платформа для SOA-приложений	Oracle SOA Suite
Платформа для приложений пользовательского интерфейса (UI)	Oracle WebCenter
Инфраструктура для SOA-решений	Oracle VM Oracle JRockit JVM Oracle Weblogic Server Oracle Tuxedo
Высокопроизводительные SOA-решения	Oracle Grid Oracle JRockit JVM Oracle Coherence Oracle Event Server Oracle MessageQ
Среда разработки	Oracle JDeveloper

SOA: плюсы и минусы

Готовы ли компании к SOA-архитектуре?

Аналитическое агентство CNews Analytics /CNA/ исследовало настроения участников рынка. Оказалось, что интерес к SOA остается высоким. Многие компании корректируют свои SOA-стратегии, чтобы подготовить ИТ-инфраструктуру и бизнес к быстрому выходу из кризиса

Что нужно учитывать?

SOA – это долгосрочный трансформационный проект. Могут пройти годы – перед тем как бизнес почувствует реальные результаты (например, возможность быстрой адаптации ИТ под новые процессы).

CIO, разработчики и бизнес-пользователи видят SOA (и его эффективность) по-разному.

Преимущества SOA:

- способность быстро адаптировать ИТ к меняющимся задачам бизнеса;
- стимул для анализа и оптимизации бизнес-процессов компании;
- возможность обеспечить интеграцию при консолидации компаний (= систем);
- возможность снизить TCO систем.

Особенности SOA:

- продолжительные сложные проекты;
- переосмысление ИТ-стратегии предприятия в целом.

Однако

- проект SOA должен решать задачи не ИТ-департамента, а бизнеса в целом;
- часто результат SOA – красивая инфраструктура, но не реальная ценность для бизнеса.

Сложности на пути SOA

Организационные – заинтересованность руководства, мотивация сотрудников.

Экономические – большие начальные инвестиции.

Ресурсные – дефицит кадров, неготовность инфраструктуры.

Технические – трудоемкость проектов, большой объем подготовительных работ.

Психологические – консервативность, излишние амбиции на старте, быстрое разочарование.

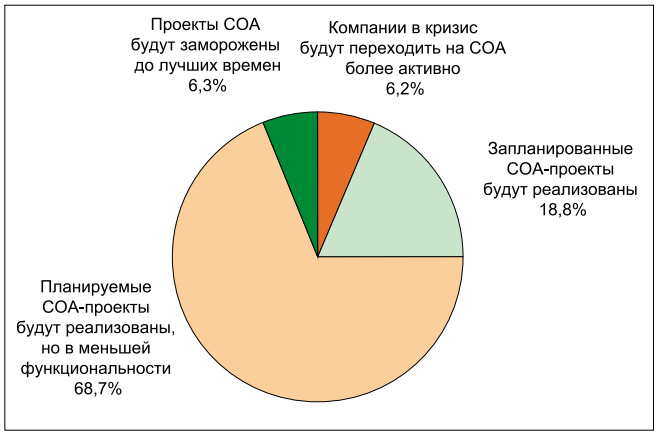
Стоимость перехода

SOA-проект требует значительных инвестиций сразу по нескольким направлениям:

Создание сервисов: создание повторно используемого компонента примерно в два с половиной раза дороже простого. Это значит, что сервис создается при условии дальнейшего использования (минимум трижды).

Рисунок 3. Готов ли российский рынок к массовым SOA-проектам? (Источник: CNews Analytics, 2009)

Рисунок 4. Будут ли компании выделять средства на SOA в кризис? (Источник: CNews Analytics, 2009)



Софт: ESB (самая затратная часть), репозиторий, системы тестирования, управления и мониторинга.

Железо: возможно потребуется дополнительно оборудование (не самая большая статья расходов).

Обучение ИТ-персонала работе с SOA.

Какова отдача?

IBM SOA Infrastructure Adoption Study провела опрос 900 ИТ-директоров и топ-менеджеров компаний из Северной Америки, Европы и Азии:

- 50% игроков, реализующих SOA-проекты, констатируют возврат инвестиций;
- более 60% из тех, кто ставил задачу сокращения издержек, отмечают достижение этой цели.

Infosys: После перехода на COA с каждым годом возрастает количество повторных использований сервисов (+10-15% в первый год, +20-25% во второй год, +30-40% на третий год и т.д.) – что напрямую сказывается на сокращении затрат.

Что можно измерить переходя на SOA?

- повышение эффективности (применительно к бизнес-процессам – BPM);
- сокращение административных издержек;
- повышение прозрачности существующих бизнес-процессов;
- сокращение бумажного документооборота;
- ускорение внедрения процессов;
- сокращение циклов проектов;
- сокращение совокупной стоимости развития и поддержки приложений.

Старт SOA-проекта

Ваша компания готова к старту SOA-проекта, если у нее есть:

Инфраструктура – достаточная пропускная способность сети, доступность серверов.

Архитектура – приложения позволяют выставлять данные в качестве сервисов.

Информация – готовность данных к представлению в виде сервисов, адекватная структура хранилища.

Человеческие ресурсы – уровень квалификации, знакомство с технологиями, адаптивность, готовность к работе в новых парадигмах.

Управление – заинтересованность топ-менеджмента, инновационный настрой, инициативность CIO.

Шесть ошибок в SOA -проектах

Беспорядочный подход к сервисам. Начиная проект, компании часто бездумно выделяют веб-сервисы; среди них оказываются избыточные или ненужные, а процедура становится неоправданно дорогой.

Игнорирование бизнес-аналитиков. Участие бизнес-аналитиков с самого начала проекта важно для его успешного завершения, чтобы фокус не оказался исключительно на технических задачах и выделении веб-сервисов. В первую очередь SOA должна быть бизнес-задачей.

Внимание к продуктам в ущерб идеологии. Компании часто фокусируются на инструментах, интеграционных платформах и софте, хотя рассматривать их имеет смысл только после составления детального плана миграции на новую архитектуру. Без особой необходимости не стоит сразу закупать много новых ИТ-продуктов.

Приоритет большим и сложным проектам. Лучше начать с небольших и минимально рисковых проектов (менее очевиден результат, но зато хорошая площадка для обучения, накопления опыта и необходимой базы знаний). У сложных и крупных инициатив риски очень высоки, они чаще проваливаются.

Непонимание ключевых ролей. Бизнес-роли должны быть заложены еще на стадии планирования. Часто у проектной команды и ее руководителя нет четкого понимания – кто является ключевым «владельцем» данных. Проект, в котором нет изначального понимания, какие подразделения владеют какими сервисами, обречен.

Надежда на быстрое распространение. Многие ожидают, что SOA-инициативы будут быстро распространяться от одного бизнес-подразделения к другому, и считают неудачей, если этого не происходит. Однако именно постепенный медленный переход на сервисы всего предприятия является залогом более эффективной инфраструктуры. EOF

Стереотипы поведения уничтожают преимущества

Разработка композитных приложений в сервис-ориентированной архитектуре требует изменений не только в технологиях, но в первую очередь в процессах развития и управления корпоративной информационной системой. Сформированные годами стереотипы поведения как у ИТ-специалистов, так и у бизнес-заказчиков способны свести на нет все преимущества SOA. Вопреки распространенному мнению, основные затраты времени приходятся не непосредственно на разработку ИТ-решений, а на выработку и согласование требований, соблюдение процедур и правил внесения изменений. Изменение подобной практики займет не один день. Пожалуй, наиболее эффективным подходом внедрения новых методов управления является организация центра компетенции по интеграции (Integration competence center), дополняющего традиционные процессный и проектный подходы к управлению ИТ.

Максим Смирнов,
руководитель департамента архитектуры систем поддержки бизнеса ОАО «ВымпелКом»

Поддержка SOA-решений сложнее привычных соединений

Трудности возникают по нескольким направлениям. Это трудности технологические (некоторые решения в стиле SOA построить с соблюдением заявленных атрибутов качества крайне сложно); ресурсные (пока SOA является только архитектурной парадигмой, используемой при построении решений, но не выделена в проект, достаточно приоритетный для банка, недостаточно ресурсов для процессов стандартизации, определения бизнес-объектов и т.д., объединяемых словом governance).

Некоторые трудности возникают, конечно, в связи с тем, что решения в стиле SOA на этапе накопления сервисов являются более сложными и затратными, чем при peer-to-peer-интеграции. Это трудности как на уровне менеджеров проектов, которым приходится разъяснять, почему в проект добавлен «архитектурный налог» в виде создания сервисов, так и на уровне, например, системных администраторов, для которых поддержка SOA-решений объективно сложнее привычных peer-to-peer-соединений.

Эдуард Петренко,
главный эксперт департамента ИТ, управление архитектуры и системных центров компетенции ЗАО «ЮниКредит Банк»

До часа «X» – меньше 100 дней

Что делать с персональными данными?

На вопросы «СА» отвечают сопредседатели комитета по информационной безопасности МОО «Союз ИТ-директоров России» Виктор Минин и Юрий Шойдин

– Бизнес-сообщество сегодня активно обсуждает Федеральный закон «О персональных данных». 1 января 2010 года информационные системы персональных данных (ИСПДн) должны быть приведены в соответствие с его требованиями. Времени для этого мало. Что делать?

– Необходимо разделять позицию законодателя в отношении операторов, которые обязаны обрабатывать персональные данные (ПДн) согласно требованиям ФЗ и в отношении субъектов персональных данных, права которых должны быть защищены.

Очень условно деятельность оператора ПДн можно разделить на две части – «нетехническая» и «техническая». «Нетехническая» (административная и документарная) сторона вопроса защиты оператором обрабатываемых персональных данных заключается в формировании документарной базы, являющейся основой защиты ПДн, и административных мероприятий, позволяющих судить о защите ПДн.

Если говорить о «технической» стороне, необходимо учитывать, что в информационной системе предприятия приложений, обрабатывающих ПДн, значительно больше, чем мы себе представляем.

Поэтому их выявление и выполнение технических мероприятий по защите – одна из первостепенных задач оператора.

К сожалению, вышедший ФЗ имеет целый ряд существенных нестыковок с ранее действующими законодательными актами.

– Что это за противоречия?

– Согласно закону данные, попадающие под его действие, можно трактовать как самостоятельный режим «ограниченного использования». Значит, необходимо «вынесение» ПДн из других существующих режимов. С точки зрения законодательства такой подход неверен, потому что «персональные данные» по классификации являются информацией, а не режимом.

В законе «Об информации, информационных технологиях и защите информации» указывается, что персональные данные не должны передаваться пользователю без согласия субъекта, а в законе «О персональных данных» отмечается ответственность оператора персональных данных за их распространение.

Еще одной острой юридической проблемой является лицензирование. Дело в том, что закон «О персональных данных» не предусматривает лицензирования деятельности оператора ПДн.

С другой стороны, есть закон «О лицензировании отдельных видов деятельности», согласно которому в определенных случаях, например когда оператор обрабатывает ПДн не только для собственных нужд, их защита подлежит лицензированию.

Стоит отметить, что всем операторам придется получать соответствующие лицензии ФСТЭК и/или ФСБ. По крайней мере, до тех пор, пока вопрос лицензирования не будет разъяснен регуляторами.

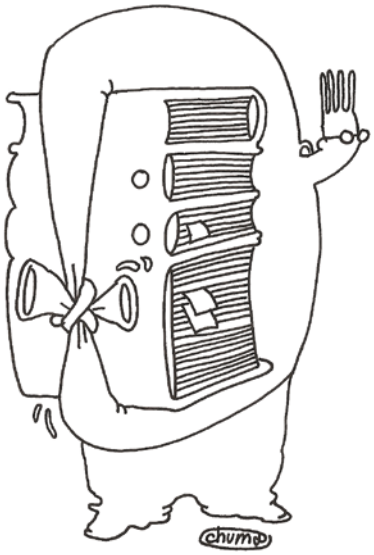
Есть и другие трудности. Одна из них – организационно-финансовая. Так, многие компании и организации не включили в свой бюджет на 2009 год затраты на создание систем по защите ПДн. Поэтому в текущем году они могут не успеть их создать. Особенно с учетом финансового кризиса.

Еще хуже обстоит ситуация с органами власти и бюджетными организациями. При разработке и утверждении закона «О персональных данных» Правительство РФ не рассматривало выделение бюджетным организациям средств на приведение своих инфор-

Досье

ВИКТОР МИНИН, советник председателя Совета МОО АЗИ, председатель Общественного консультативного совета по научно-технологическим вопросам информационной безопасности Комиссии по информационной безопасности при Координационном совете государств – участников СНГ по информатизации при РСС

ЮРИЙ ШОЙДИН, директор по ИТ ГК «Интарсия», член правления СПб Клуба ИТ-директоров, член МОО СоДИТ



Многие компании не включили в бюджет на 2009 г. затраты на создание систем по защите ПДн. Поэтому они могут не успеть их создать

мационных систем в соответствие с требованиями закона.

В результате существует два возможных варианта развития ситуации. Первый вариант – если ряд существенных поправок не будет урегулирован до конца 2009 года, то, возможно, придется сдвинуть крайний срок приведения информационных систем персональных данных в соответствие с требованиями ФЗ № 152 с 1 января 2010 года на более поздний срок.

Второй вариант – закон вступит в действие с 1 января 2010 года, а все коллизии будут решаться в судебном порядке, персонально.

– Кто и как будет осуществлять контроль над его выполнением?

– Контроль операторов персональных данных осуществляется тремя регуляторами. ФСБ России отвечает за лицензирование деятельности операторов ПДн, при использовании ими при защите ПДн криптографических средств защиты. ФСТЭК России контролирует требования при категорировании и в части технической защиты ИСПДн, а также лицензирование деятельности операторов ПДн при осуществлении ими технической защиты конфиденциальной информации. И, наконец, Россвязькомнадзор контролирует деятельность оператора в части обеспечения организации защиты ПДн, регистрацию операторов ПДн и ведение реестра операторов.

Будем надеяться, что контролирующие организации будут производить проверки не по очереди, а одновременно.

– С чего начать разработку системы защиты персональных данных?

– Всем операторам нужно, в первую очередь для себя, закрепить документально основные понятия обработки конкретных персональных данных субъектов.

Единый документ об обработке персональных данных в организации должен предвосхитить все вопросы, которые могут возникнуть у уполномоченного органа. Поэтому необходим предварительный анализ документации оператора, которая содержит персональные данные субъектов.

При проведении такого анализа и составлении текста «Положения» важно учитывать возможность типизации ПДн. К примеру, когда оператор обрабатывает персональные данные субъектов с единой целью, например, для предоставления услуги связи, в «Положении» должны быть четко сформулированы понятия цели, способов, перечня обрабатываемых персональных данных и прочее.

Если оператор обрабатывает ПДн работников, преследуя различные цели, к примеру, для начисления заработной платы, установления пропускного режима, учета в кадровом делопроизводстве, возникает потребность их типизировать относительно цели обработки, и это должно быть отражено в «Положении».

Такая система описания позволит оператору эффективно ориентироваться в документах, содержащих персональные данные, давать ответы на запросы субъектов и уполномоченного органа.

Еще одна проблема – сроки хранения документов и информации, содержащей персональные данные. Законодательство, пожалуй, впервые устанавливает для информации и документов максимальный, и к тому же условный, срок хранения – «по достижении целей обработки». Организации должны будут установить сроки хранения документов, содержащих ПДн, причем необходимо заранее продумать обоснование выбранных сроков хранения.

Что касается технических мероприятий, то они должны включать в себя учет лиц, допущенных к работе с персональными данными в информаци-

Кстати

По оценкам Россвязькомнадзора, операторами персональных данных являются около 3 млн российских компаний и организаций. Под надзор должны попасть более полутора миллионов из них. В прошлом году Россвязькомнадзор провел 76 проверок операторов ПДн. Результатом проверки стало 19 предписаний об устранении обнаруженных нарушений, пять административных дел переданы в суд. Сегодня лидерами по нарушениям являются банки и другие финансовые организации. Нарушения, отмеченные у них, в большей степени носят организационный характер. На втором месте – предприятия ЖКХ, а третье и четвертое делят операторы связи и страховые компании.

Документы, которые должны быть у оператора ПДн

- Письменное согласие субъекта персональных данных.
- Нормативный документ (перечень), аккумулирующий информацию о персональных данных, обрабатываемых оператором (в том числе их категория, объем и сроки хранения).
- Положение о персональных данных и их защите.
- Должностные инструкции сотрудников, имеющих отношение к обработке персональных данных.

Дополнительно нужно отметить особенности документа «согласие субъекта ПДн». Требования к форме письменного согласия установлены ФЗ (п.4 ст.9). Письменная форма применяется в случаях если:

- персональные данные включаются в общедоступные источники;
- оператором обрабатываются специальные (раса, национальность, политические взгляды, религиозные убеждения, состояние здоровья, интимной жизни) и биометрические персональные данные (характерные физиологические особенности).

Кроме того, письменное согласие субъекта требуется:

- при любой передаче (распространении) его персональных данных;
- при запрашивании любых персональных данных субъекта у третьего лица;
- при принятии оператором решения, порождающего юридические последствия в отношении субъекта персональных данных;
- при трансграничной передаче персональных данных на территории иностранных государств, не обеспечивающих адекватной защиты прав субъектов персональных данных.

онной системе. При этом лица, доступ которых к ПДн, обрабатываемым в информационной системе, необходим для выполнения служебных (трудовых) обязанностей, должны допускаться к соответствующим данным на основании списка, утвержденного оператором или уполномоченным лицом.

Сам по себе этот список для информационной системы мало что значит. Важно обеспечить разграничение доступа к приложениям в соответствии со списком и дать пользователям именно те права, которые им необходимы для выполнения должностных обязанностей. Сделать это без эффективной системы управления идентификацией и доступа будет весьма затруднительно.

Кроме того, должны быть реализованы мероприятия, направленные на предотвращение несанкционированного доступа к персональным данным и/или передачи их лицам, которые не имеют права доступа к такой информации, а также обеспечивающие своевременное обнаружение фактов несанкционированного доступа к ПДн. Все запросы пользователей ИСПДн на получение персональных данных и факты предоставления ПДн по этим запросам должны регистрироваться автоматизированными средствами информационной системы в электронном журнале обращений, содержание которого должно периодически проверяться ответственными лицами оператора.

Сложность при категорировании, защите и сертификации ИС возникает в связи с тем, что современные автоматизированные ИС используют программные продукты, целиком охватывающие все предприятие. При повсеместном использовании ERP-систем, где этот модуль интегрирован в саму систему, защищать и сертифицировать придется всю автоматизированную ИС.

– Как создать систему защиты ПДн?

Во-первых, нужно организовать руководство вопросами организации защиты ПДн.

Во-вторых, необходимо создать два направления по обеспечению защиты обрабатываемых ПДн. Это техническое направление и направление «нетехнической» защиты, обязанности которого, в свою очередь, можно перераспределить по структурным подразделениям согласно целям обработки.

Например, защита персональных данных в кадровом делопроизводстве, в контрольно-пропускной системе, юридическом структурном подразделении (при учете договоров с субъектами ПДн). При этом должностной инструкцией должна быть установлена персональная ответственность работников этих направлений за надлежащую защиту ПДн.

Приступать к технической защите можно после проведения категорирования персональных данных и определения их объема. Эта операция позволит определить класс защиты ПДн и осуществить подбор необходимых средств, удовлетворяющих требованиям класса защиты. Необходимо также продумать создание системы технической защиты персональных данных. Произвести подготовку и провести аттестацию системы технической защиты ПДн.

– Как классифицировать ИСПДн? Каковы требования к аттестации информационных систем?

– На основании закона классификацию ИСПДн оператор персональных данных должен осуществить само-

Этапы проведения обработки персональных данных

- Обследование (аудит) бизнес-процессов компании на наличие в них ПДн.
- Инвентаризация ИС, обрабатывающих ПДн.
- Оценка законности обработки ПДн и наличие согласия субъектов на обработку.
- Контроль и корректировка договорных отношений с субъектами.
- Формирование перечня ПДн и проведение категорирования.
- Определение целей обработки ПДн.
- Определение сроков и условий прекращения обработки ПДн.
- Разграничение доступа пользователей к ПДн в ИСПДн.
- Формирование документов регламентирующих работу с ПДн.
- Формирование модели угроз, содержащей актуальные угрозы информационной безопасности персональным данным при их обработке в информационной системе.
- Классификация ИСПДн.
- При, необходимости определенной в №152-ФЗ, направить уведомление об обработке ПД в уполномоченный орган по защите прав субъектов персональных данных, Роскомнадзор (http://www.rsoc.ru/main/directions/874).
- Приведение системы защиты ПДн в соответствие требованиям регуляторов.
- При необходимости, определенной методическими документами ФСТЭК России и ФСБ России получить необходимые лицензии.
- Аттестация ИСПДн.
- Эксплуатация ИС – мониторинг, выявление и реагирование на инциденты ИБ.

стоятельно. При этом модель угроз создается не исходя из своего опыта и здравого смысла, а на основании методики и базовой модели, утвержденной ФСТЭК.

Что касается общих требований безопасности ИСПДн, то они определены в законе. Оператор при обработке ПДн обязан принимать необходимые организационные и технические меры, в том числе использовать шифровальные (криптографические) средства для защиты ПДн от неправомерного или случайного доступа к ним, уничтожения, изменения, блокирования, копирования, распространения и иных неправомерных действий.

– Какие могут быть последствия, если ничего этого не делать?

– Законодательство устанавливает уголовную ответственность за нарушение неприкосновенности частной жизни [1], за нарушение тайны переписки, телефонных переговоров, почтовых, телеграфных или иных сообщений [2].

Кроме того, гражданское законодательство предусматривает защиту нематериальных благ граждан, включающих, в частности, неприкосновенность частной жизни, личную и семейную тайну, деловую репутацию. Устанавливается компенсация морального вреда, возможность требования по суду возмещения убытков и опровержения сведений, порочащих честь, достоинство и деловую репутацию гражданина [3].

Использование несертифицированных информационных систем, баз и банков данных, несертифицированных средств защиты информации, если они подлежат обязательной сертификации, влечет их конфискацию [4].

А нарушение правил защиты информации и отказ в предоставлении гражданину информации может привести к административному приостановлению деятельности организации сроком до 90 суток [5].

Разглашение информации, доступ к которой ограничен ФЗ, и в частности персональных данных (за исключением случаев, если разглашение такой информации влечет уголовную ответственность), лицом, которое имело к ней доступ по служебным или профессиональным обязанностям, карается административным штрафом до десяти тысяч рублей [6]. EOF

1. Уголовный кодекс Российской Федерации, ст. 137.
2. Уголовный кодекс Российской Федерации, ст. 138.
3. Гражданский кодекс Российской Федерации, часть 1, ст. 150, 151, 152.
4. Кодекс Российской Федерации об административных правонарушениях, ст. 13.12.2.
5. Кодекс Российской Федерации об административных правонарушениях, ст. 5.39 и ст.13.12.
6. Кодекс Российской Федерации об административных правонарушениях, ст. 13.14.

Визитка

АЛЕКСАНДР ЕМЕЛЬЯНОВ, *инженер группы информационных технологий Владимирского филиала ООО «Татнефть-АЗС-Запад»*

Windows Server 2008 R2

Новые возможности служб AD DS

Чем порадуют системных администраторов функциональные нововведения службы каталогов в Windows Server 2008 R2

Компания Microsoft все годы своей деятельности придерживается основной концепции создания продуктов – удобство и простота управления вкупе с максимальной безопасностью. С этим, я думаю, согласятся как рядовые пользователи, так и инженеры, обслуживающие серверные операционные системы. Примером этому служит то, какие изменения претерпели ОС с момента выхода первых версий для рабочих групп. Поэтому выход нового релиза, даже для текущей версии продукта, практически всегда обещает преподнести очередной пусть и не большой, но приятный сюрприз.

Новшества службы каталогов «созыва» R2 нельзя назвать революционными, но многие из них определенно заслуживают внимания администраторов.

Приведем перечень новых функциональных возможностей Active Directory:

- новая оснастка Центр администрирования AD DS (Active Directory Administrative Center);
- возможность введения компьютера в домен, не имея физического подключения к контроллеру;
- анализатор оптимальных конфигураций;
- корзина AD DS;
- встроенные командлеты PowerShell;
- возможность создания учетных записей для сервисов.

Теперь обо всем по порядку.

Центр администрирования Active Directory

Основанная на Windows PowerShell новая оснастка для управления службой каталогов позволит администраторам выполнять повседневные действия, как создание учетных записей и управление уже существующими, создание и управление группами безопасности, а также организационными подразделениями. Вдобавок ко всему консоль управления снабжена гибкой системой поиска и создания шаблонов для выполнения рутинных повторяющихся действий, тем самым экономя время администратора.

Помимо управления доменом, в состав которого входит компьютер с установленным центром администрирования, из данной оснастки можно подключиться к другому домену при условии наличия одно- или двухсторонних доверительных отношений между доменами.

Новый центр администрирования устанавливается только на машины под управлением Windows Server 2008 R2. В скором времени инженеры Microsoft обещают дополнить возможность его запуска на Windows 7.

Автономный ввод компьютера в домен AD

Операция довольно проста, основная цель – введение в домен компьютера, не имеющего прямой связи с контроллером домена. Расскажу о некоторых особенностях ее проведения.

Для выполнения всех манипуляций требуется утилита djoin.exe. Ввести таким образом можно в домен под управлением контроллера Windows Server 2003 и выше, однако в качестве клиентов для присоединения могут выступить только машины с операционными системами Windows 7 или Windows Server 2008 R2. Кроме этого, в доменах Windows Server 2003/2008 должен присутствовать хотя бы один компьютер Windows 7 или Windows Server 2008 R2.

Именно на одной из этих ОС запускается djoin.exe для создания файла с данными для подключения клиента-компьютера. Синтаксис ее выглядит следующим образом:

```
djoin.exe /provision /domain имя_целевого_домена ↵
        /machine имя_компьютера ↵
        /dcname имя_контроллера_домена [/downlevel] ↵
        /savefile c:\blob.txt
```

Параметр /downlevel используется в доменах Windows Server 2003/2008.

Одновременно с созданием файла blob.txt в схеме службы каталогов появляется учетная запись компьютера. Далее этот файл каким-либо образом передается на подключаемую машину, где снова запускается утилита djoin.exe, которая имеет немного другой синтаксис:

```
djoin.exe /requestODJ /loadfile c:\blob.txt ↵
        /windowspath %SystemRoot% /localos
```

Это все. После перезагрузки рабочая станция будет при-

соединена к домену и при появлении связи с ним будет, возможно, работать в его контексте безопасности.

Анализатор оптимальных конфигураций

Поскольку поиск и устранение ошибок в рамках администрирования Active Directory зачастую превращается в задачу не из легких, разработчики Microsoft решили немного облегчить жизнь администраторам.

Название службы Best Practices Analyzer говорит само за себя. Предназначена она для анализа инфраструктуры AD при развертывании и управлении, подсказывая, где у вашей службы каталогов «больные места». Этот анализ основывается по большей части на опыте крупных компаний, а также специалистов по развертыванию Active Directory.

По сути своей анализатор оптимальных конфигураций – это небольшая система сообщений в консоли управления сервером, где человек, обслуживающий AD, может почерпнуть информацию о проблемах и недочетах инфраструктуры.

Корзина службы каталогов

Здесь будет указано лишь основное предназначение данного функционала службы каталогов, а также причины его возникновения. О том, как включать корзину AD (по умолчанию она отключена) и использовать ее на благо администратора, можно прочитать в [1].

Заглянем немного в прошлое. В предыдущих версиях Active Directory для восстановления объектов приходилось прибегать к использованию архивных копий, загружать контроллер в режиме восстановления службы каталогов либо прибегать к помощи утилит сторонних производителей (например, Quest Recovery Manager от Quest Software). И пусть стараниями всеми уважаемого Марка Руссиновича был изобретен инструмент ADRestore, это решило лишь часть проблем с возвращением к жизни удаленных учетных записей.

Речь идет о том, что при восстановлении аккаунт пользователя терял многие свои атрибуты, в том числе и ссылочные (характерный пример таковых – членство в группах). Чтобы избежать этого, находились методы вмешательства в схему AD (смотри [2]), после которых атрибуты объектов удаленных и после восстановленных, можно было вернуть обратно. Не очень удобно, учитывая, какое количество пользователей может содержать домен.

Microsoft, наконец, придумала временное хранилище для удаленных объектов – корзину AD (AD Recycle Bin). Теперь удаленный объект можно восстановить при помощи того же ADRestore или LDP вместе со всеми атрибутами, в том числе и паролем.

Корзина может использоваться также для облегченной версии службы каталогов AD LDP. Отмечу, что включение корзины – операция необратимая.

Windows Powershell

С выходом новой версии серверной ОС администраторы получили в составе ее встроенный инструмент командной строки для управления службой каталогов, более совершенный, чем привычные уже dsget.exe, dsmod.exe, dsadd.exe, dsmove.exe и другие. В состав Windows PowerShell входит 85 командлетов, позволяющих выполнять всевозможные манипуляции с объектами. Для получения списка всех командлетов можно использовать командлет Get-Command *-AD*. Помимо этого, как уже писалось в [1], в Windows Server 2008 R2 появилась новая среда разработки (подобная Visual Studio) – Windows PowerShell Integrated Scripting Environment, которая поможет в создании и отладке скриптов.

Создание учетных записей для сервисов

Это еще один шаг в сторону улучшения безопасности. Разъясню вкратце смысл данного нововведения. Оно позволяет не создавать отдельные учетные записи пользователей (для которых могут действовать различные политики паролей) для запуска сервисов в рамках домена. Более того, если в домене действует политика смены пароля через заданный промежуток времени, для такой учетной записи эта операция будет производиться автоматически.

Безусловно, многие, если не все, новшества пойдут на пользу администраторам службы каталогов. Более подробно обо всех новинках обновленной версии Active Directory можно прочитать в библиотеке TechNet [3], а также посмотреть в видеодокладе Ильи Рудь по адресу [4]. EOF

1. Коробко И. Управление корзиной. Новый сервис в Active Directory. //Системный администратор, №8, 2009 г. – С. 46-50.
2. http://itband.ru/2009/03/восстановление-удаленных-доменных-у.
3. Библиотека TechNet – http://technet.microsoft.com/ru-ru/library/cc440482.aspx.
4. http://www.techdays.ru/Lecture.aspx?LID=1381.

Визитка

АЛЕКСАНДР КОСИВЧЕНКО, технический специалист ЗАО «Компьювэй». Занимается внедрением серверных решений на базе ПО Microsoft

Live Migration

Что нужно для ее использования?

Поговорим о возможностях и применении новой технологии, вошедшей в состав Microsoft Windows Server 2008 R2

В Microsoft Windows Server 2008 R2 появилось много новых возможностей. Среди них одной из самых интересных является технология, позволяющая перемещать виртуальные машины между узлами кластера с нулевым временем простоя. В предыдущей моей статье [1] было рассмотрено использование отказоустойчивых кластеров при виртуализации серверов, а здесь мы подробно рассмотрим одну из полезных возможностей такого решения – Live Migration, которая как раз и позволяет такое перемещение осуществлять.

Live Migration: что за зверь и с чем его едят?

Наверняка многим из читателей приходилось сталкиваться с проблемой, когда сервер необходимо на какое-то время выключить или перезагрузить: установка обновлений ОС, замена аппаратных компонентов. Приходилось оповещать всех пользователей, делать свое «черное дело», а потом снова оповещать всех, что-де все в порядке и можно продолжать работу. Иногда же приходилось оставаться на месте после окончания рабочего дня или же приходить в выходные дни. В некоторых случаях, когда требуется работа 24/7, это будет еще проблематичнее. Именно для таких случаев была придумана технология Live Migration, позволяющая переносить виртуальные машины с одного узла кластера на другой незаметно для пользователя.

Как это работает?

Сам процесс миграции может быть инициирован тремя способами:

- вручную через оснастку Failover Clustering;
- с помощью командлета PowerShell;
- с помощью дополнительного ПО, например System Center Virtual Machine Manager.

После инициации процесса миграции файлы конфигурации виртуальной машины передаются на узел назначения и на нем создается пустая виртуальная машина (так называемый скелет), которой выделяется область памяти необходимого объема.

Рисунок 1. Процесс синхронизации содержимого памяти осуществляется итерационно

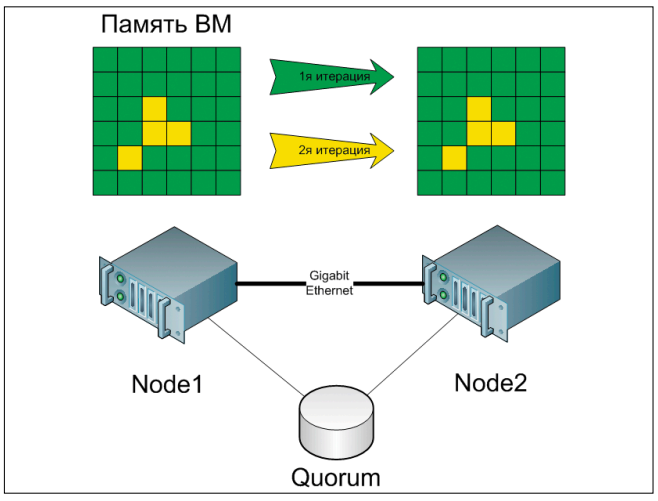
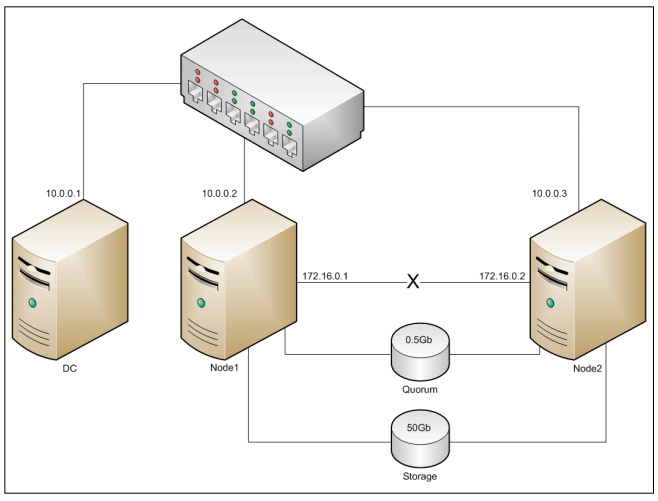
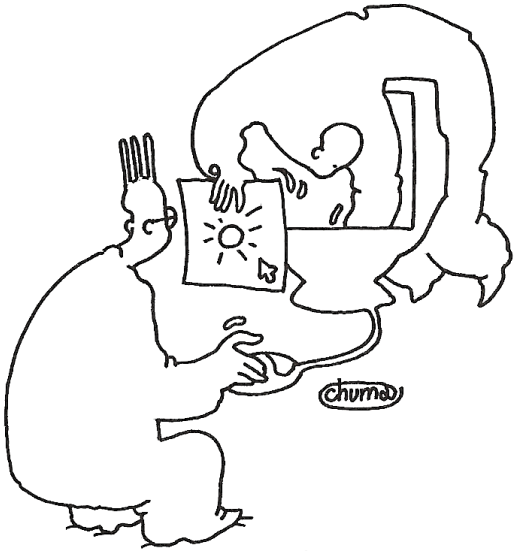


Рисунок 2. Тестовая среда для развертывания кластера





Технология Live Migration позволяет переносить виртуальные машины с одного узла кластера на другой незаметно для пользователя

Затем запускается процесс синхронизации содержимого памяти. В процессе синхронизации виртуальная машина продолжает работать.

На рис. 1 показана область памяти, отведенная некоей виртуальной машине, расположенной на узле Node1. После запуска процесса синхронизации все содержимое памяти передается страницами по 4 Кб на целевой хост, где предварительно была выделена область памяти такого же объема, и содержимое записывается напрямую в память. Это первая итерация. В процессе передачи содержимое некоторых страниц было изменено (желтые ячейки). Поэтому запускается вторая итерация, в процессе которой копируются только измененные страницы. Так как их будет значительно меньше, чем суммарный объем памяти, процесс займет намного меньше времени, и после второй итерации содержимое памяти на обоих хостах будет идентично. На самом деле итераций может быть больше – до 10. Если достичь полной идентичности за 10 итераций не удается, процесс миграции прерывается. Это одно из слабых мест Live Migration: если виртуальная машина слишком активно работает с памятью, существует теоретическая возможность, что перенести ее с помощью Live Migration будет невозможно. В этом случае надо подождать спада активности или использовать другие методы перемещения, например Quick Migration. Сразу же после окончания процесса синхронизации – новой виртуальной машине передается управление всеми необходимыми дисками (как VHD, так и pass-through-дисками). Процесс такого «переключения» происходит практически мгновенно, TCP-соединения не успевают «отвалиться» по тайм-ауту, и пользователи не замечают, что что-то изменилось.

Cluster Shared Volumes

В Windows Server 2008 R2 появились так называемые Cluster Shared Volumes CSV – некий аналог кластерной файловой системы. Cluster Shared Volume (CSV), по сути, представляет собой кластерную файловую систему с единой точкой монтирования для общих дисковых ресурсов на всех узлах кла-

Рисунок 3. Подключение iSCSI-дисков

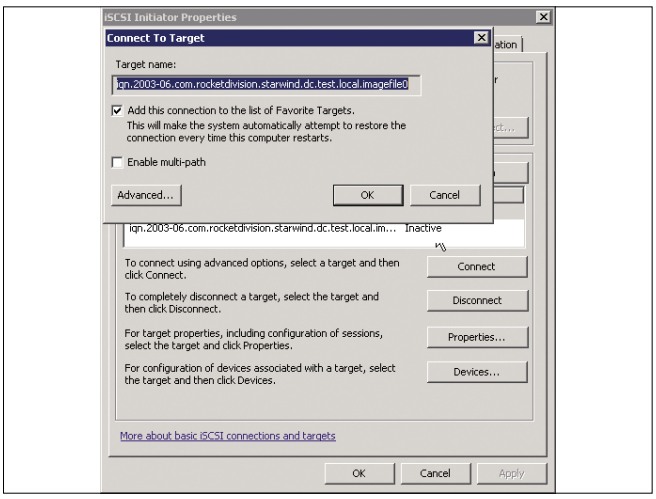
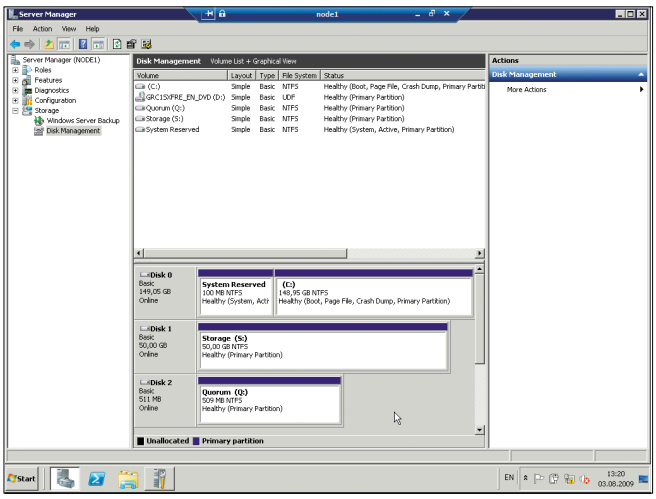


Рисунок 4. Окно оснастки управления дисками после создания разделов



стера. Кроме того, как известно, с каким-либо одним общим дисковым ресурсом кластера до недавнего времени одновременно мог работать только один из узлов. Поэтому, чтобы иметь возможность одновременной работы нескольких виртуальных машин на разных серверах, необходимо было для каждой виртуальной машины создавать отдельный LUN на системе хранения данных. Это могло породить некоторые проблемы при большом количестве виртуальных машин. Дело в том, что существуют технические ограничения на количество создаваемых LUN'ов в разных СХД. Использование Cluster Shared Volume позволяет решить эту проблему: доступ к одному и тому же CSV может осуществляться одновременно со всех узлов кластера, и поэтому количество виртуальных машин ограничивается лишь объемом дискового пространства. Как же выглядит Cluster Shared Volume в системе? На системном разделе (например, диск C:) всех узлов кластера создается папка ClusterStorage, и все дисковые устройства монтируются в ней, как подпапки с именем Volume1, Volume2 и т.д. Таким образом, доступ к файлам на одном и том же блочном устройстве на всех узлах кластера будет осуществляться по одному и тому же пути: C:\ClusterStorage\Volume1\. К сожалению, на данный момент Microsoft разрешает использовать CSV только для задач Hyper-V. Использование CSV в других целях является неподдерживаемым решением, о чем выводится соответствующее предупреждение при включении CSV.

Основное требование для использования CSV – системный раздел на всех узлах кластера должен иметь одинаковую букву диска, например C:.

Требования для использования Live Migration

Существуют требования, которые необходимо соблюсти для использования Live Migration.

- Поддерживаемые версии ОС:
  - Windows Server 2008 R2 64bit Enterprise Edition;
  - Windows Server 2008 R2 64bit Datacenter Edition;
  - Hyper-V Server 2008 R2.
- Все хосты, на которых планируется использовать Live Migration, должны являться узлами Microsoft Failover Cluster.
- Microsoft Failover Clustering поддерживает до 16 узлов в одном кластере.
- Следует создать между узлами отдельную независимую сеть для трафика Live Migration с пропускной способностью 1Gbps и выше.
- Все узлы кластера должны иметь процессоры одного производителя (AMD/Intel).
- Для использования Cluster Shared Volume все узлы кластера должны иметь одинаковую букву загрузочного раздела (например, C:).
- Все хосты должны принадлежать к одной IP-подсети.
- Все хосты должны иметь доступ к общему хранилищу данных.

Помимо обязательных требований, рекомендуется:

- Для хранения файлов виртуальных машин использовать Cluster Shared Volume.
- Конфигурация кластера должна удовлетворять Microsoft Support Policy for Windows Server 2008 Failover Clusters: http://support.microsoft.com/default.aspx?scid=kb;EN-US;943984.

Также обязательно надо учитывать, что при использовании Live Migration любые два узла кластера могут быть задействованы только в одном процессе миграции, и при этом перемещается только одна виртуальная машина. Нельзя одновременно перемещать несколько машин с одного хоста или на один хост. Это означает, что в кластере из N узлов может одновременно происходить N/2 процессов миграции.

Основные сценарии применения

Для чего же можно применять кластеры, и в частности Live Migration? Существует несколько общих сценариев.

Сценарий 1. Техническое обслуживание серверов, установка обновлений

Любой сисадмин периодически сталкивается с необходимостью замены или добавления какого-либо «железа» на сервере или с установкой обновлений ПО, требующих перезагрузки сервера. Так как при использовании виртуализации на одном физическом сервере запущено несколько виртуальных, выключение или перезагрузка физического серве-

Рисунок 5. Проверка конфигурации успешно завершена, можно собирать кластер

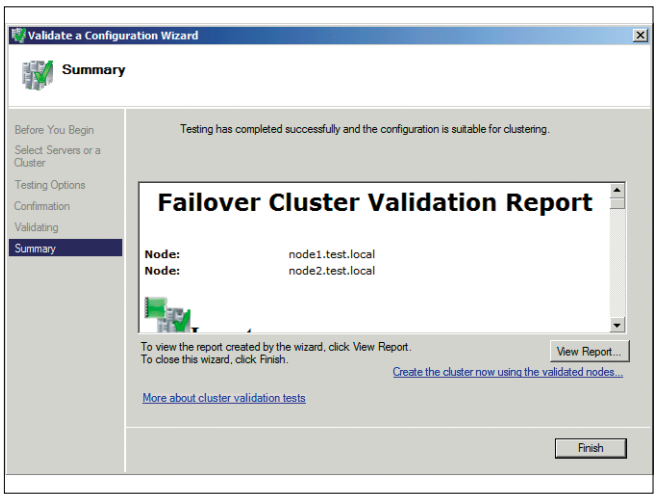
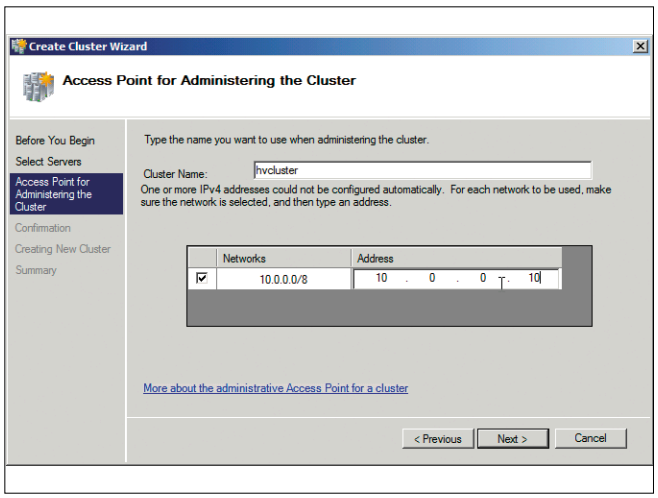


Рисунок 6. Зададим сетевое имя и IP-адрес для нашего кластера



ра неизбежно затронет все виртуальные серверы, запущенные на нем. Поэтому, как правило, такие работы планируют на нерабочее время или на выходные дни, что не очень удобно для администраторов.

Благодаря использованию Live Migration виртуальные машины могут быть перемещены на другой физический хост с нулевым временем простоя, что абсолютно незаметно для пользователя. Это означает, что любые работы, даже связанные с остановом сервера, можно проводить в разгар рабочего дня, просто переместив предварительно все виртуальные машины на другой сервер и не обзванивая пользователей с просьбой «выйти из программы».

Сценарий 2. Динамическая ИТ-инфраструктура

Благодаря использованию технологии Live Migration можно сделать ИТ-инфраструктуру полностью динамической. Очень сильно помогут в этом другие программные продукты – Microsoft System Center Virtual Machine Manager (VMM) и System Center Operations Manager (SCOM). Они позволяют следить за загруженностью всех физических хостов в реальном времени и в зависимости от загруженности перемещать виртуальные машины с более загруженных серверов на менее загруженные, распределяя таким образом нагрузку равномерно.

Возможен и другой сценарий: при низкой загруженности серверов можно автоматически перемещать виртуальные машины, запуская большее число виртуальных машин на меньшем числе физических хостов. Неиспользуемые хосты при этом выключаются, что позволяет снижать энергопотребление и требования к кондиционированию в помещении серверной. При возрастании нагрузки в «часы пик» свободные хосты могут быть снова автоматически включены, и нагрузка может быть вновь равномерно распределена по хостам.

От теории – к практике

Теперь попытаемся «пощупать» Live Migration на практике: развернем кластер на настоящих серверах, создадим высокодоступную (High Available) виртуальную машину и попробуем ее мигрировать.

Краткое описание тестовой среды

Итак, что же мы имеем? Для работы с Live Migration развернута тестовая среда, включающая в себя:

- Виртуальную машину, на которой поднят контроллер домена TEST.LOCAL и iSCSI Target от компании StarWind (бесплатная версия, поддерживает виртуальные диски до 2 Тб).
- Два компьютера, играющих роль узлов кластера. Каждый из них снабжен процессором фирмы Intel, удовлетворяющим требованиям для работы Hyper-V. Оба компьютера снабжены двумя сетевыми адаптерами, один из которых подключается в коммутатор, другой – соединяется с другим компьютером перекрестным патч-кордом.
- Обычный коммутатор Gigabit Ethernet.

Всего создано два iSCSI-устройства. Одно – объемом 0,5 Гб – будет использоваться в качестве кворума. Другое, 50 Гб – для хранения данных. Схема тестовой среды приведена на рис.2.

Начальные настройки узлов кластера

В качестве узлов кластера используются два компьютера – Node1 и Node2. У каждого из них имеется по два интерфейса Gigabit Ethernet – LAN1 и LAN2 (см. таблицу). Оба узла (Node1 и Node2) введены в домен TEST.LOCAL.

Итак, первое, что нам необходимо сделать, – это подключить наши iSCSI LUN'ы. Для этого на первом узле заходим в Administrative Tools – iSCSI Initiator. Сервис iSCSI по умолчанию не запущен, поэтому система спросит, хотим ли мы, чтобы этот сервис запустился и стартовал автоматически. Соглашаемся. В окне настроек инициатора первое, что нам необходимо сделать, – это указать, где находятся наши iSCSI-диски. Заходим на закладку Discovery, выбираем Discover Portal и вводим адрес нашей системы хранения. В данном случае, как уже было написано, это сервер DC, то есть 10.0.0.1. Если все было указано правильно, в списке порталов появится наш сервер. Если были какие-то проблемы (например, порт закрыт файерволом или адрес неправильный), выдаст сообщение об ошибке. Теперь идем на закладку Targets и видим, что у нас появились два таргета со статусом Inactive. Нужно по каждому из таргетов кликнуть и нажать Connect. В появившемся окошке оставляем все по умолчанию (имя таргета должно оставаться, и верхняя галочка должна быть включена – тогда наш диск автоматически подключится при перезагрузке ОС, см. рис. 3).

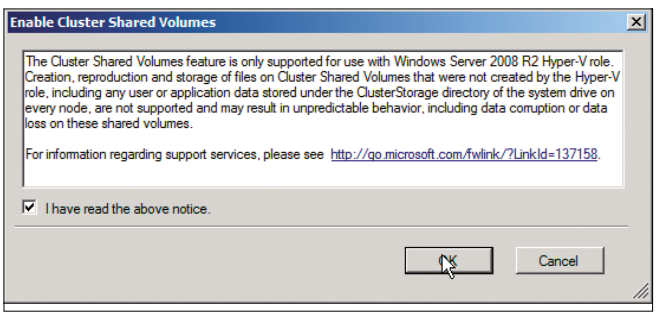
Затем переходим на закладку Volumes and Devices и выбираем Auto Configure. После этого можно закрывать окно свойств инициатора нажатием кнопки OK.

Следующим шагом нам нужно зайти в консоль Server Manager в раздел Storage – Disk Management. Если все было сделано правильно, то там появятся два новых диска и оба будут в состоянии Offline. Переводим их в Online. Затем оба диска надо проинициализировать (Initialize) и создать разделы. Мы создадим на каждом диске по одному разделу объе-

Таблица. Настройки сетевых интерфейсов

Узел/Интерфейс	IP-адрес	Маска подсети	Default Gateway	DNS
DC	10.0.0.1	255.0.0.0	–	127.0.0.1
Node1/LAN1	10.0.0.2	255.0.0.0	10.0.0.1	10.0.0.1
Node1/LAN2	172.16.0.1	255.255.0.0	–	–
Node2/LAN1	10.0.0.3	255.0.0.0	10.0.0.1	10.0.0.1
Node2/LAN2	172.16.0.2	255.255.0.0	–	–

Рисунок 7. При включении Cluster Shared Volumes появляется предупреждение



мом на весь диск и отформатируем в системе NTFS (другие не поддерживаются Microsoft Clustering Services). Присваиваем маленькому диску букву Q: и метку Quorum, а большому – S: и Storage соответственно. Естественно, это делается для удобства. После успешного завершения все будет выглядеть, как на рис. 4.

Затем надо установить роль Hyper-V и Failover Clustering. В оснастке Server Manager идем в раздел Roles, жмем Add Roles, в мастере выбираем роль Hyper-V и устанавливаем ее. Можно и даже желательно поставить галочку напротив сетевого адаптера, «смотрящего» в коммутатор, чтобы не создавать потом виртуальную сеть руками. Потребуется перезагрузка сервера, смиренно соглашаемся и ждем. После перезагрузки снова заходим на сервер. Через какое-то время мастер установки ролей продолжит работу. Надо подождать сообщения об успешной установке Hyper-V. Закрываем мастер, идем в Server Manager → Features → Add Features и выбираем всего одну компоненту: Failover Clustering. Все, больше нам здесь делать нечего. Заходим на второй узел и повторяем все точно так же, за исключением работы с дисками (инициализация-форматирование и т.д.): на втором узле iSCSI-диски должны остаться в состоянии Offline. За сим настройка узлов заканчивается.

Создание отказоустойчивого кластера из двух узлов

После того как мы подключили общие хранилища к узлам будущего кластера и установили необходимые компоненты системы, можно переходить к собственно настройке кластера. С любого из узлов (это не принципиально) или, как в нашем случае, с DC запускаем консоль Failover Cluster Manager (далее – FCM).

Первое, что нам необходимо сделать, – это проверить, получится ли у нас вообще кластер из наших двух серверов. Для этого запускаем проверку: Validate a Configuration. Запустится мастер проверки конфигурации. Вначале выбираем наши оба узла. Далее нам будет предложено выбрать, запустить ли все тесты или только выбранные. Я и Microsoft настоятельно рекомендуем остановить выбор на Run All Tests, потому что только в этом случае решение будет поддерживаться Microsoft. Затем можно идти пить чай: проверка займет какое-то время, порядка 10 минут. Если все OK, мы увидим что-то наподобие рис. 5.

Если были какие-либо проблемы, можно посмотреть отчет (View Report) в виде веб-страницы, где будет подробно объяснено, какие тесты провалились и почему. После исправления всех проблем надо запустить проверку заново.

Итак, проверка успешно завершена, и можно наконец-таки собирать кластер. В FCM выбираем Create a Cluster. Появится мастер, похожий на предыдущий. Надо выбрать наши два узла, а затем ввести DNS-имя и IP-адрес для нашего кластера. В нашем случае – hvcluster с IP-адресом 10.0.0.10 (см. рис. 6).

Теперь кластер собран. Нужно убедиться, что в качестве кворума используется именно наш диск объемом 512 Мб. Заходим FCM – имя_кластера – Storage. В Disk Witness in Quorum должен стоять наш диск Q:, а второй диск (50 Гб) должен находиться в Available Storage.

Использование Cluster Shared Volume

Теперь было бы неплохо сделать так называемый Cluster Shared Volume. Включаем CSV: идем FCM – имя_кластера и выбираем Enable Cluster Shared Volumes. Выводится предупреждение о том, о чем я говорил выше – что CSV может использоваться только для виртуальных машинах и ни для чего более (см. рис. 7).

Ставим галочку, что мы все прочли и поняли, жмем OK. В дереве слева появляется раздел Cluster Shared Volumes. Заходим туда, выбираем Add Storage – и выбираем наш диск, объемом 50 Гб. После этого он пропадает из Available Storage, но появляется в Cluster Shared Volumes и доступен на каждом узле как C:\ClusterStorage\Volume1.

Создание высокодоступной виртуальной машины

Для начала нужно создать саму виртуальную машину. Это можно сделать как непосредственно из консоли FCM, так и из оснастки Hyper-V Manager. В Hyper-V Manager нам заходить все равно придется, так что сделаем оттуда. Итак, на любом из узлов, допустим, на первом, создаем виртуальную машину. Обозвать ее можно как угодно – например TestVM. Память и все остальное – по вкусу. Но есть

Рисунок 8. Файлы виртуальной машины будем хранить на CSV

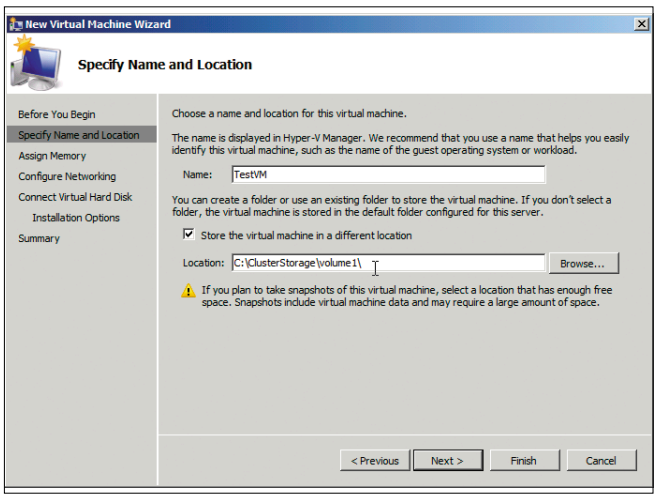
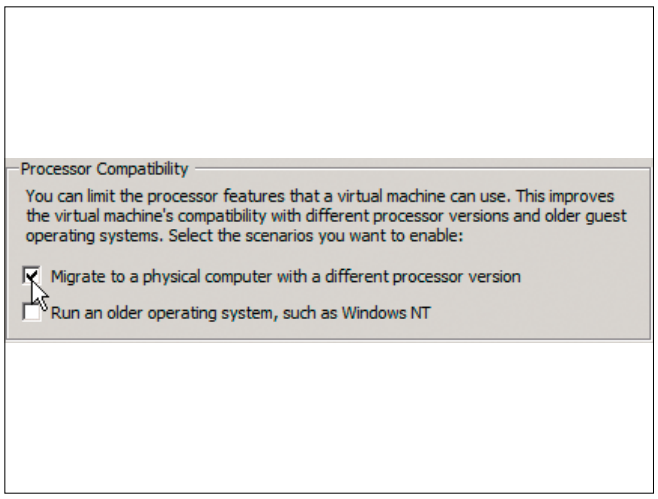


Рисунок 9. Включаем Processor Compatibility Mode



нюансы. Путь для сохранения файлов виртуальной машины не оставляем по умолчанию, а указываем наш кластерный диск (см. рис. 8).

Размер виртуального диска желательно во избежание конфуза указать меньше, чем размер нашего кластерного диска. Например – 20 Гб. Дальше все по вкусу.

Сразу же, если мы собираемся использовать миграцию, а процессоры у серверов одного производителя, но разной модели (как в нашем случае), необходимо включить Processor Compatibility Mode. Заходим в свойства виртуальной машины (Settings) и в разделе Processor устанавливаем галочку Migrate to a physical computer with a different processor version (см. рис. 9).

Теперь нам надо сделать нашу виртуальную машину высокодоступной. Термин «Высокодоступная (High Available)» виртуальная машина» означает, что виртуальная машина сконфигурирована особым образом для работы в кластере, и только в этом случае поддерживаются дополнительные возможности, связанные с кластерами, такие, как миграция между узлами.

Возвращаемся в FCM – имя_кластера. В разделе Services and applications выбираем Configure Servise or Application. В мастере указываем, что это будет виртуальная машина (Virtual Machine внизу списка), и выбираем нашу созданную виртуальную машину (TestVM). Виртуальные машины можно создавать прямо из FCM, и тогда они сразу же автоматически делаются высокодоступными, но тем не менее придется зайти в Hyper-V Manager для настроек процессора (см. выше).

Использование Live Migration

Теперь на нашу готовую и высокодоступную виртуальную машину можно (и нужно!) установить ОС. Какую ОС ставить – это ваш выбор. После установки ОС можно проверить Live Migration. Для этого можно, например, запустить непрерывный пинг нашей виртуальной машины в отдельном окне или качать на нее/с нее какой-нибудь большой файл (фильм в HD весом в 25 Гб, например).

Пока все это безобразие происходит, попробуем тихонечко перенести нашу многострадальную виртуальную машину на другой узел. Кликаем на ней правой кнопкой и выбираем Live Migrate, а в выпадающем меню – тот узел, куда она будет переезжать. Процесс миграции занимает пару секунд – и мы видим, что Current Owner изменился. Если открыть Hyper-V Manager, то можно убедиться, что наша виртуальная машина находится уже на другом узле. При этом во время миграции ни пинг, ни закачка файла, как мы видим, не прерываются. Ура!

***

Итак, в этой статье мы ознакомились с новой «изюминкой» Microsoft Windows Server 2008 R2: Live Migration. Мы узнали, что нужно для ее использования, как происходит сам процесс миграции, и осуществили все описанное на практике. Надеюсь, что вы нашли эту статью полезной. EOF

1. Косивченко А. Комплексное решение: виртуализация + отказоустойчивый кластер. //Системный администратор, №9, 2009 г. – С. 16-19.

Визитка

АНДРЕЙ ПОНАРЕВ, в прошлом системный администратор ЗАО «Нефтепродукт», в настоящее время – фрилансер

Контроль трафика

Прослушиваем и просматриваем вместе с BWMeter

Хотите обеспечить защиту небольшой сети и наблюдать за ее активностью без особых усилий? Контролировать сеть прямо в офисе? Это возможно

> … когда видят солнце и луну, это не считается острым зрением; когда слышат раскаты грома, это не считается тонким слухом.
>
> **Сунь-Цзи, IV, 4**

Компьютеры, а вместе с ними и Интернет, сейчас незаменимы в любой фирме. Но раз есть Интернет, значит, есть и угроза вторжения вредоносного ПО, что уже само по себе неприятно. Нужна надежная защита подобной сети с учетом области ее применения. Но количество ПО, проводящих защиту таких сетей от нежелательных вторжений, настолько велико, что выбрать среди них подходящее – дело не из легких. Я предлагаю вам обратить внимание на BWMeter, легкий в освоении, но в то же время невероятно мощный инструмент.

Возможности BWMeter

В первую очередь BWMeter – это сетевой монитор, способный следить за трафиком и при необходимости сообщать об изменениях в его объеме. Вместе с тем визуальная часть также порадует нас своими возможностями. Сетевую активность можно отобразить в графиках, причем для каждого вида трафика можно завести отдельную диаграмму. Очень выручает возможность ведения статистики и логов, хотя на такое способен практически каждый сетевой монитор.

В свою очередь, программа не потребует мощного компьютера. Единственное, что придется соблюсти, – это поставить Windows. Причем BWMeter поддерживает всю серию систем Windows: от старинных 95-х и 98-х до Vista и новомодной Windows 7, которая выходит совсем скоро. С семеркой, кстати, парни разобрались только в версии 5.1.1.

Чем же так примечательна программа? В первую очередь установкой (см. рис. 1). А если короче, она проста, как апельсин. После запуска инсталлятора можно смело «промотать» первые два окна с лицензионным соглашением. Конечно, их можно и обстоятельно изучить, но в защите сети они вам никак не помогут. В следующем окне вам предложат

Рисунок 1. Установка BWMeter

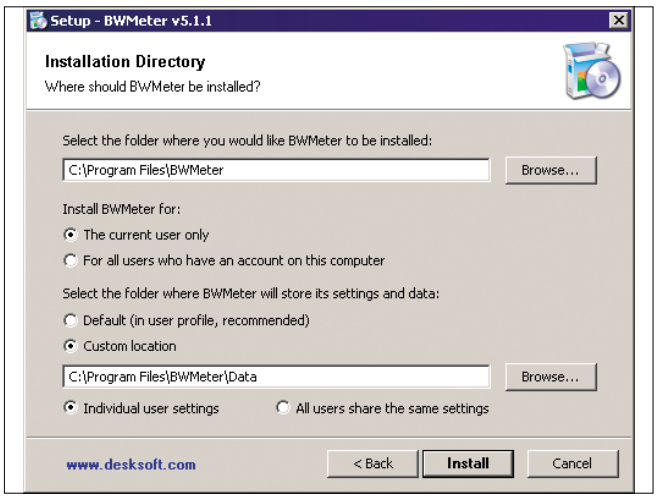
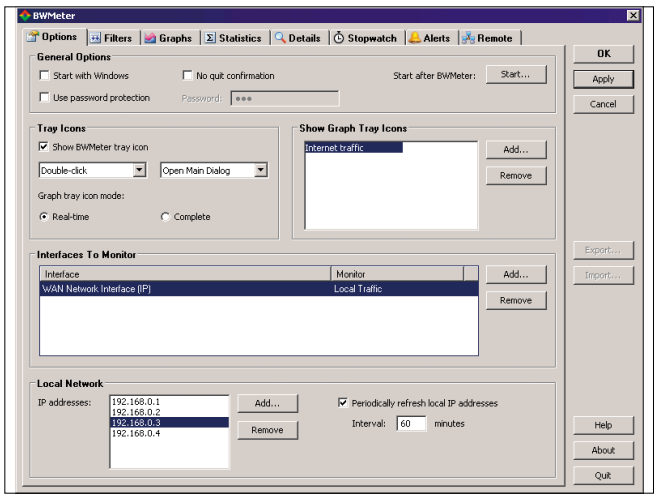


Рисунок 2. Основные настройки сетевого монитора



выбрать директорию установки и директорию для настроек. Вот и все, можете жать Install.

Это, конечно, не самое главное достоинство программы. Далее по шкале «необычности» идет файервол. Он настраивается, если так можно выразиться, с полпинка. Конечно, сначала он кажется неудобным и странным. Но после того как вы в нем разберетесь, а это займет немного времени, я вас уверяю, вы поймете, что ничего лучше и удобнее нет на свете. Однако самое главное в BWMeter – это возможность удаленно следить за трафиком, настройка подобной возможности дело нехитрое. Вообще этот монитор очень прост в использовании, хотя и имеет обширный список возможностей. Не верите? Тогда запустите его и проверьте это.

Если вы запустили BWMeter впервые, то вместе с несколькими диаграммами на экране появится окно настроек межсетевого монитора. Чтобы добраться до настроек, когда вам этого хочется, необходимо в трее найти иконку монитора, щелкнуть по ней правой кнопкой мыши и выбрать пункт Options. То же произойдет после двойного клика по любой из диаграмм или после нажатия <Ctr>+<O> в тот момент, когда одна из диаграмм активна (см. рис. 2).

Раздел General Options кроме привычных пунктов настроек содержит доселе незнакомый Start after BWMeter и Use password protection. Если с последним все ясно, он предоставляет возможность ставить пароль на настройки, то первый немного смущает. Зачем что-то запускать после монитора? Ну мало ли... Конечно, если ваш компьютер постоянно подключен к сети, то эта возможность вам не нужна. А если в сеть вы выходите только для того, чтобы отправить и получить почту? В таком случае постоянно работающий сетевой монитор будет только поглощать ресурсы компьютера, не принося при этом никакой пользы. А если еще учесть, какие компьютеры сейчас стоят на производстве? Тогда удобнее было бы запускать монитор только во время выхода в сеть, и одновременно запускать необходимую программу, для чего выгоднее пользоваться встроенной функцией.

Тонкости фильтрации

BWMeter создан скорее для слежения за трафиком, а не для его фильтрации, но и на это программа способна. Я даже скажу, что следить за трафиком удобно в первую очередь благодаря фильтрам. От слов к делу: перейдем ко вкладке Filters. В одноименном разделе нам предоставляют возможность создавать политики фильтрации. По умолчанию присутствует только две: Internet и Local Network. Назначение кнопок Add и Delete, думаю, и так понятно, а вот с кнопкой Copy стоит разобраться поподробнее. Конечно, с ней тоже все просто, она создает новую политику путем копирования старой. Мне лишь хотелось отметить, что эта элементарно реализуемая функция встречается далеко не у всех брандмауэров, хотя она позволяет экономить ценное время.

Область действия каждой политики поддается очень тонкой, но не всегда доступной для понимания настройке. Поэтому, чтобы не углубляться в теорию, я предлагаю рассмотреть возможности фильтров на практике, создав группу политик для экономии трафика и защиты компьютера от нежелательных вторжений. Итак, ситуация: имеется компьютер, подключенный к локальной сети и Интернету. Во внутренней сети присутствует FTP-сервер, доступ к которому осуществляется штатными средствами Windows. Также на компьютер установлен браузер Mozilla Firefox. А в Интернете тем временем буйствует червь MSBlast! В общем, рядовые будни сисадмина (см. рис. 3).

Начнем с простого – с разграничения локальной сети и Интернета. Цель нашего действия: обеспечить возможность управлять отдельно каждым видом трафика. Я думаю, каждый администратор сталкивался с подобной проблемой. Для решения поставленной задачи нам потребуется две политики: Local traffic и Internet traffic. Я рекомендую удалить стандартные политики с помощью кнопки Delete. Воспользовавшись кнопкой Add, создадим новую политику под названием Local traffic. Теперь в разделе Filter Definition раскроем вкладку General и изучим ее поподробнее.

Галочка Enabled позволяет прекращать и возобновлять работу политику. Список с названием Mode позволяет выбрать режим работы политики: Normal, Firewall (Silent) и Firewall (Interactive).

Normal – это обычный фильтр, он собирает в себе пакеты определенного вида. Firewall отличается от обычного филь-

Рисунок 3. Настройка фильтров

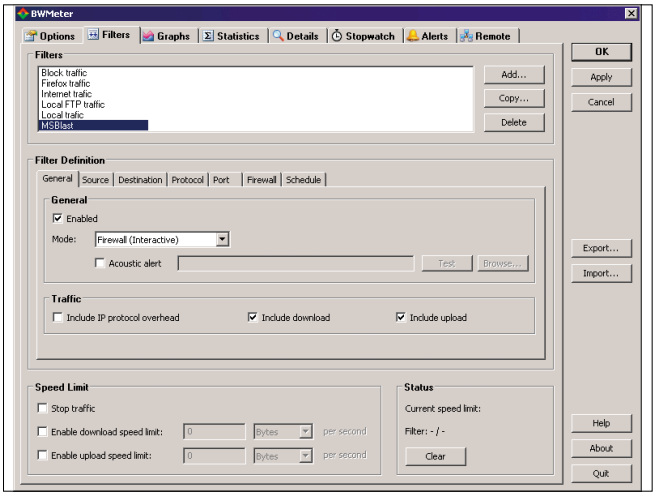
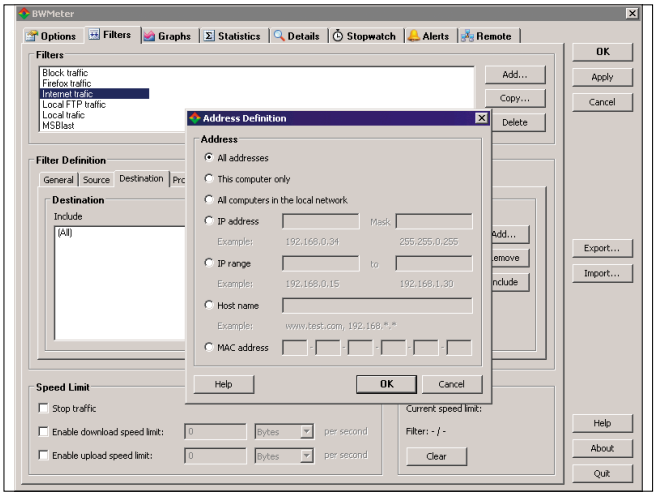


Рисунок 4. Выбираем тип адреса из предложенного списка



тра тем, что вдобавок может еще и блокировать приложения, а в интерактивном случае может даже подать сигнал. Мы к ним еще вернемся. Для отделения входящего трафика от исходящего можно воспользоваться разделом Traffic, поломав голову над галочками Include Download и Include Upload.

Нам же необходимо учесть и входящий, и исходящий трафики, поэтому нужно установить обе галочки в разделе Traffic. Также в списке Mode выберем режим Normal, т.к. нас не интересует блокирование приложений. А теперь перейдем к вкладке Source. Нетрудно догадаться, что эта вкладка предназначена для фильтрации трафика по адресу источника пакета. В одноименном разделе вкладки мы увидим два списка: Include и Exclude. Первый содержит адреса, включенные в фильтр, второй – исключенные из него. Напомню, что если пакет входящий, то в поле Source хранится адрес машины, которой он был послан, если же пакет исходящий, то там хранится адрес машины, которая его отослала. Исходя из этого, в список Include нам необходимо включить адрес данной машины. После нажатия кнопки Add появится диалог ввода адреса (Address Definition) на выбор: все адреса, только адрес этой машины и т.п. Думаю, с этим и так все понятно (см. рис. 4).

Теперь перейдем ко вкладке Destination. Здесь нам предстоит отбирать пакеты по их целевому адресу. Два уже знакомых нам списка и аналогичный способ ввода данных. Опять же напомню, что если пакет исходящий, то целевой адрес – это адрес машины, которой он был отправлен, а если входящий, то адрес машины, которая отправила пакет. Следовательно, в список Include нужно добавить все адреса локальной сети. В списке адресов для подобных целей присутствует отдельный пункт – All computers in the local network. Этот список редактируется во вкладке Options, в разделе Local Network.

Политика Internet traffic настраивается аналогично, только во вкладке Destination в список Include нужно включить все адреса (All addresses в диалоге Address Definition), а в список Exclude включить все адреса локальной сети. Кстати, перемещение значения из одного списка в другой можно осуществить нажатием одноименных кнопок.

Настало время настроить фильтр для отсеивания трафика FTP. Для этого создадим политику Local FTP traffic путем копирования политики Local traffic. И сразу же обратимся ко вкладке Port. Нас встретят уже знакомые списки Include и Exclude. Кликнув по кнопке Add, мы увидим диалог Port Definition. В этом окне две вкладки – Simple и Advanced. В первой вкладке мы вводим один порт, а во второй целых два (см. рис. 5). Как так?

Допустим, мы отслеживаем FTP-трафик и нас не интересует, с какого порта пришел пакет. Но возможны ситуации, в которых от этого зависит отсеивание трафика. Тогда нужно пользоваться вкладкой Advanced. Таким образом, в список Include нужно включить 21 TCP-порт. Но вот незадача: во вкладке Port настраивается только внешний порт. Не беда, тут нам поможет вкладка Protocol, в которой можно аналогичным образом настроить внутренний порт. Нам нужен 6-й, он как раз соответствует TCP.

Также не помешало бы настроить приложения, для которых будет активна данная политика. За это отвечает вкладка Application. И снова два списка. По нажатию на кнопку Add нам предложат либо указать приложение вручную, либо выбрать одно из стандартных. Стандартные – это System и All. Нас, как нетрудно догадаться, интересует первое.

Настройка «Огненного Лиса» тоже проста. Достаточно скопировать политику Internet Traffic и добавить к фильтру два порта: 80-й и 443-й, HTTP и HTTPS соответственно. Ну и, конечно, приложение выбрать. В принципе порты можно и не выбирать, ограничившись лишь выбором приложения.

А как теперь заблокировать нежелательный интернет-трафик? Достаточно скопировать все ту же политику Internet Traffic и во вкладке Application в раздел Include вписать все приложения, а в раздел Exclude – системные приложения и «лиса». Но это лишь отследит нежелательный трафик. А как его «зарубить»? Обратим внимание на раздел Speed Limit. Там можно установить галочку Stop traffic. Но есть и альтернатива, чтобы подобный трафик не забивал канал (когда всякие «винампы» обновления ищут), можно ему задать скоростной лимит. Для этого можно установить галочками Enable download/upload speed limit и выбрать нужное значение.

Рисунок 5. Вводим порт для «Огненного Лиса»

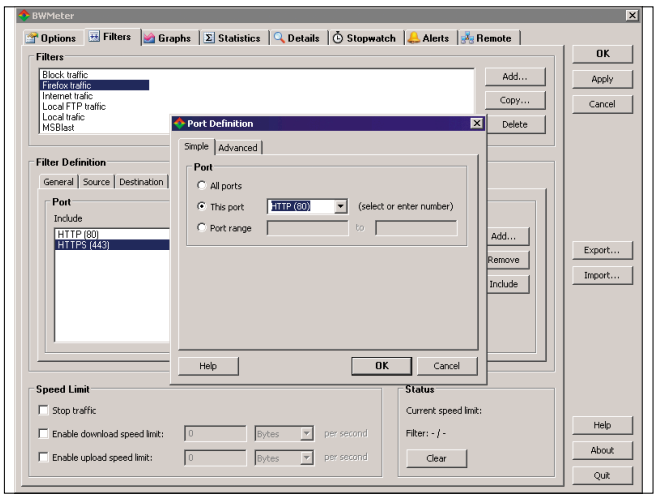
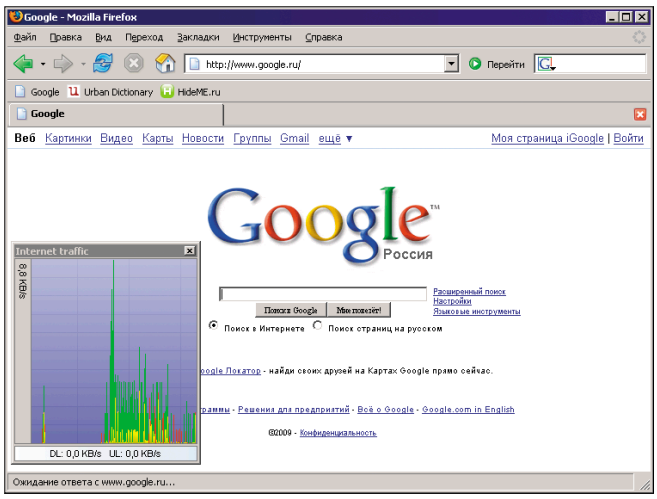


Рисунок 6. Следим за активностью браузера



Но мы забыли про MSBlast, действия которого тем временем приобретают масштабы наполеоновских завоеваний. Как известно, данный червь ломится через 135 TCP-порт, который открыт службой DCOM RPC. Можно, конечно, скачать заплатку, если эта служба необходима. Если же она не представляет интереса, то проще создать политику, блокирующую данный вид трафика. Мы так и поступим.

Создадим новую политику, и в списке Mode выберем Firewall (Interactive). В политику нужно включить все адреса, TCP-протокол и 135-й порт. А вот в разделе Firewall нужно все значения из списка Allow перевести в список Forbid (там должны быть системные приложения и svchost.exe). И MSBlast успешно остановлен!

Возможности мониторинга

Наверное, BWMeter – одна из самых удобных программ для слежения за трафиком. Она является таковой благодаря тому, что следит не за всем трафиком, а только за тем, который проходит через фильтры. С одной стороны, это неудобно, т.к. фильтры вряд ли покроют весь трафик. Но ведь всегда можно создать отдельный фильтр, через который проходит весь трафик, и при желании отслеживать и его. А представьте, если программа жестко следит за всем трафиком, не давая ее грамотно настроить. Тогда вся информация месится в кучу, и разобрать что-то в этой куче представляется невозможным. Конечно, «грамотный» администратор потратит целый день на разбор логов и таки найдет нужную информацию. Но есть ли у меня, «обычного» администратора, столько времени? Вряд ли.

Возможно, при запуске программы вы заметили несколько диаграмм, отображающих активность сети. Эти графики отображали активность двух стандартных фильтров. Для настройки подобных графиков обратимся ко вкладке Graphs (см. рис. 6).

Создание и удаление графиков происходит в одноименном разделе. Все аналогично созданию фильтров. Я предлагаю для начала создать два графика: Local Traffic и Internet Traffic. Если теперь выделить один из графиков и обратить внимание на раздел Show Activity of These Filters, то мы увидим пустой список. А что там должно быть? Там должен быть список фильтров, активность которых отображается на графике. Причем, заметьте, на одном графике могут отображаться несколько фильтров, однако я не советую вам так делать, если речь идет о вашей машине. Если же вы удаленно отслеживаете активность еще нескольких машин, то было бы удобнее на одном графике отображать несколько фильтров, но об этом позже.

Разделы Scale, Update Interval, Visual Style и Position & Size не представляют сложности в понимании, да и особенностями, о которых стоило бы рассказать, они не блещут. А вот раздел Visibility стоит того, чтобы о нем поговорить отдельно. Галочка Visible позволяет, как нетрудно догадаться, скрывать или показывать график. Нижележащие галочки Show if activity rise above и Hide if activity falls below позволяют настроить автоматическое скрытие или появление графика. Вот, допустим, активность почтового клиента сейчас нулевая, что, спрашивается, смотреть на ее графике? Ответ – ничего. Поэтому рекомендую для нее установить обе галочки. Первая отображает график, если активность поднимется выше одного процента на графике, вторая спрячет график, если активность упадет ниже одного процента. Разумеется, проценты можно настроить вручную.

Ну это все, конечно, хорошо. Можно визуально отобразить активность каждого фильтра, что, скорее, удобно администратору. Все перед глазами, в логи лезть не надо. А что сказать начальнику, если он попросит отчет о сетевой активности? Показать ему графики? Боюсь, он не поймет шутки. Поэтому какой-нибудь отчет-таки придется сделать. В этом нам поможет вкладка Statistics (см. рис. 7).

Как сразу видно, BWMeter ведет статистику по каждому фильтру, причем записывает статистику не чаще чем раз в час. Сроки обновления статистики хранятся во вкладке Schedule настройки фильтра. С помощью кнопки Overview можно увидеть статистику всех фильтров сразу. Причем всю эту информацию можно экспортировать в формат .csv, который распознается Excel'ем, ввиду чего, немного поломав голову и приведя данные в удобочитаемый вид, можно все это вывести на бумагу и передать начальнику. Также, если вас интересует детальный отчет фильтра, его мож-

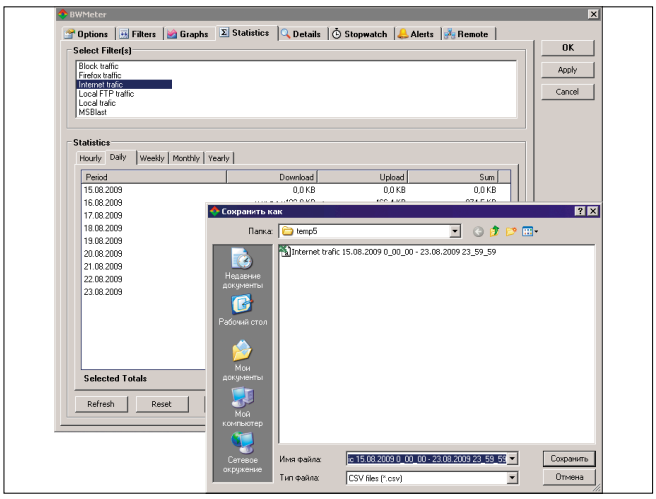
А что еще?

ПО, проводящего выслеживание и отсеивание сетевого хлама, сейчас очень много. Среди них есть достойные внимания. Это, во-первых, Kerio Winroute. Это просто мастодонт, настоящий сетевой монитор. У него есть все для слежения за корпоративной сетью. По своим возможностям он не то что догоняет BWMeter, он перегоняет его с безумной скоростью. Но… какой же он огромный и сложный! Разобраться в этом творении – воистину задача не из легких. А после этого его еще нужно настроить! В общем, для небольших сетей – это сплошной минус.

Не менее достойным претендентом является Kaspersky Internet Security. Всем хорош, много настроек, и все понятны. Настраивается достаточно быстро, хотя и придется попотеть. Особенно порадовала функция «Родительского контроля», ограничивая доступ к некоторым ресурсам. Одно вот только беспокоит: защищать придется каждый компьютер в отдельности, а не всю сеть сразу. А после BWMeter уже так привычно следить за активностью сети удаленно.

Одним словом, выбор всегда за администратором, нужно лишь твердо верить, что решение придет. А искать есть где. Бегло осмотрев рынок мониторов, можно отметить Comodo, Outpost, Sygate, Jetico. Их много, достаточно выбрать подходящий. Дерзайте!

Рисунок 7. Сохраняем статистику в файл



но записать, пользуясь вкладкой Details. В разделе Control указанной вкладки располагаются две кнопки: Start и Clear. Первая начинает запись лога указанного фильтра, а вторая его очищает. Сам лог можно настроить с помощью раздела Destination (см. рис. 8).

Ну и напоследок еще одна положительная сторона BWMeter – возможность настраивать систему оповещения. Для таких дел предназначена вкладка Alerts. В разделе Filters to Watch можно включать и исключать фильтры, за которыми производится слежение. Здесь все понятно, думаю, не стоит на этом останавливаться.

В разделе Condition настраивается ситуация, в случае которой выдается сообщение. А именно: в случае превышения лимита на суммарный трафик, в случае критического уменьшения/увеличения количества трафика за заданный промежуток времени и в случае передачи заданного объема информации. Пункты Reaches, Traffic и Each Time соответственно. Способов сообщить о случившемся представляется несколько, можно как подать звуковой сигнал, так и запустить какую-нибудь программу и даже отправить сообщения на заданный e-mail-адрес! Все это можно найти в разделе General, галочка Notification.

Контроль за сетью

Ну и наконец самая впечатляющая часть BWMeter – это возможность следить за трафиком удаленно, физически находясь хоть у себя дома. Для настройки подобных возможностей существует вкладка Remote.

Основную часть окна во вкладке занимает список Remote Filters, который содержит те фильтры, которые мы отслеживаем удаленно. После нажатия на кнопку Add перед нами появляется окно Add Remote Filters. Здесь мы выберем фильтр, за активностью которого желаем следить. В разделе Select Location нужно ввести адрес машины, на которой установлен BWMeter. Если с адресом трудновато, то можно воспользоваться кнопкой Browse... и выбрать компьютер из представленного списка. В разделе Choose Filter отображается список фильтров удаленной машины, из которого вам нужно выбрать требуемый. Чтобы получить список, нужно нажать кнопку List Filters. Чтобы не тратить время на получения списка фильтров, можно ввести название фильтра сразу, если, конечно, вы его помните (см. рис. 9).

В разделе Remote Features мы настраиваем, какую информацию и как часто хотим получать от удаленной машины. Нам представляется возможность получать информацию о фильтрах и использовать ее в графиках (галочка Enable remote filters in local graphs). Благодаря этой возможности мы сможем отображать активность фильтров удаленной машины на наших локальных диаграммах. Также можно получать статистику фильтров (галочка Enable remote statistics). То есть для составления отчета не придется бегать по кабинетам. Раздел Remote Options позволяет настроить параметры подключения. Это, как всегда, порт (по умолчанию 24810) и адрес компьютера, который имеет право получать информацию от нашего брандмауэра.

Что дальше? Идем во вкладку Graph и создаем диаграммы, отображающие активность удаленных фильтров. Я бы рекомендовал настроить по одной диаграмме на каждую машину, в которой отображать общую активность приложений. Это легко сделать, если во вкладке Graph в список Show Activity of These Filters включить несколько схожих фильтров. И получается такой себе «хакерский» рабочий стол. Есть, конечно, и недостатки у этой возможности. Скажем, детальную статистику с удаленного фильтра записать не получится. Нельзя также удаленно настраивать фильтры. А как всего этого хотелось бы! Но не надо забывать, что проект не заброшен и разработка новых возможностей продолжается. Так что, быть может, мы увидим воплощение нашей мечты.

Слово о лицензии

Я думаю, что проблема выбора сетевого монитора – одна из самых актуальных на данный момент. Количество же подобного ПО растет как грибы после дождя, тем самым выбор усложняется в несколько раз. Надеюсь, я облегчил этот выбор, и BWMeter будет практиковаться теперь не только мной и моими знакомыми. А не практиковать решительно нельзя. Ведь стоимость данного продукта крайне мала – всего 30$. Поэтому я рекомендую попробовать BWMeter в действии. EOF

Рисунок 8. Детальный отчет об активности «Огненного Лиса»

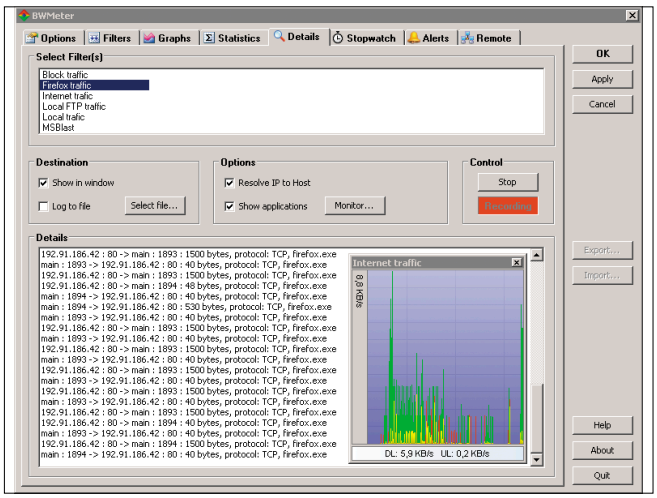
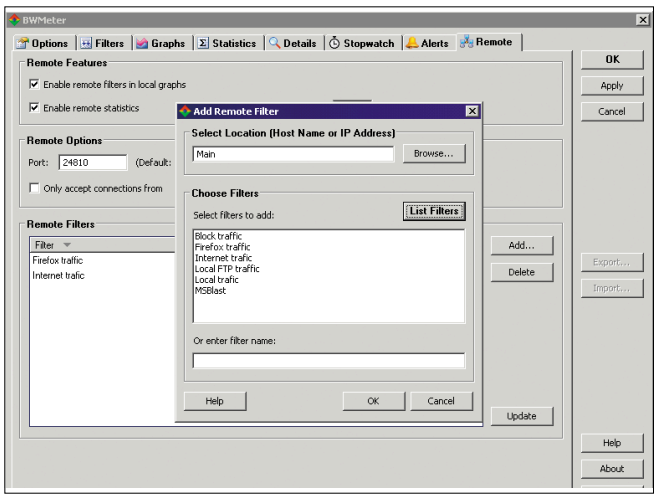


Рисунок 9. Выбираем фильтр для удаленного слежения



Аппаратные устройства Panda:
GateDefender**Performa**

Сделай подарок для себя!

Купи
Panda GateDefender Performa 9100/9500

(устройство + все модули защиты) на 1 год
и сэкономь до **250 000 руб.** за 6 месяцев!

* Акция действует
с 1 октября 2009 года
до 1 марта 2010 года.

Решение: Panda GateDefender Performa

Panda GateDefender Performa — это полноценное, высокопроизводительное SCM-устройство для обеспечения безопасности периметра сети, предоставляющее проактивную защиту корпоративной сети от всех контентных угроз, легко интегрируемое в сеть любого размера без потребности в адаптации существующей архитектуры.

Устройство сочетает проактивную защиту с Коллективным разумом для более эффективной борьбы со всеми входящими и исходящими угрозами на периметре корпоративной сети. Panda GateDefender Performa содержит пять типов защиты:

- **Антивирус:** Автоматическая защита от всех типов вредоносных программ.
- **Контент-фильтр:** Позволяет определить различные корпоративные политики безопасности.
- **Антиспам:** Обеспечивает защиту электронной почты от спама.
- **Блокировка IM/P2P:** Предотвращает злоупотребление корпоративными сетевыми ресурсами.
- **Веб-фильтрация:** Ограничивает доступ к непродуктивному веб-контенту.

	Поток трафика	GD 9100	GD 9500 Lite	GD 9500 Large
HTTP	Мбит/сек	400	500	700
	Транзакций / сек	2950	4726	4726
SMTP	Сообщений / сек	220	400	550
TCP	Соединений / сек	1008	3028	3028
	Параллельных соединений	9800	18000	18000

Основные преимущества

- **Улучшает корпоративный имидж и репутацию** путем обнаружения инфицированных компьютеров, которые могли бы рассылать спам или инфицировать клиентов и партнеров
- **Обеспечивает постоянную доступность Интернета и электронной почты** с помощью всех моделей Performa без потери трафика в случае повреждений или инцидентов
- **Увеличивает продуктивность сотрудников,** удаляя спам из почтовых ящиков, ограничивая использование IM/P2P приложений и контролируя веб-контент, к которому они имеют доступ
- **Обнаруживает почти 100%** спама благодаря новым, улучшенным антиспамовым техникам в новых устройствах Panda GateDefender Performa
- **Позволяет «подключить и защитить»** сеть без необходимости вносить изменения в архитектуру сети
- **Предоставляет высоко превентивное** обнаружение и дезинфекцию неизвестных угроз без вмешательства со стороны администратора, что позволяет значительно усилить корпоративную систему управления рисками
- **Снижает потребление ресурсов, увеличив при этом производительность и юзабилити.** Значительно увеличивает продуктивность компании и непрерывность бизнеса.

Ключевые характеристики

- **Интегрированные проактивные технологии.** Сочетание эвристики, коллективного разума и карантина позволяет обнаружить угрозы и получить важную информацию.
- **Полная защита.** Защита периметра от контентных угроз содержит защиту от вредоносного ПО и потенциально опасного контента (Panda), спама (Cloudmark) и нежелательного веб-контента (Cobion) ограничивает использование P2P и IM.
- **Гибкие политики безопасности.** Можно определить различным пользователям или группам специальные политики безопасности. Любой входящий и исходящий контент анализируется в соответствии с типом контента и специфичностью профиля каждого отправителя или получателя.
- **Обнаружение зомби** на персональных компьютерах и почтовых серверах с помощью идентификации исходящей SMTP-почты
- **Гарантия непрерывности трафика.** Функция транзитной передачи bypass позволяет продолжить передачу трафика вне зависимости от возможных повреждений системы
- **Автоматическая защита.** Постоянные обновления сигнатур и отчеты в режиме реального времени демонстрируют работу устройства в автоматическом режиме.
- **Централизованный мониторинг и управление.** Конфигурирование всех установленных в сети устройств, а также доступ к отчетам и графикам осуществляются через единую консоль

PANDA SECURITY 20 годовщина 1990-2010

www.viruslab.ru

Визитка

ВАДИМ ШПУРИК, разработчик программного обеспечения, фрилансер. Область интересов и приложения творческих усилий: создание интеллектуальных мультиагентных систем; изучение интеллектуального поведения искусственных объектов

Мониторинг состояния
температурных датчиков с помощью протокола SNMP

Как разработать пользовательское расширение к ядру Net-SNMP-демона (SNMP-агент) для сбора данных с температурных датчиков

Использование SNMP-технологии требует, чтобы все объекты были подключены к физической сети, имели стек протоколов TCP/IP и соответствующий SNMP-агент. Случается, что контролируемые объекты не имеют не только необходимого программного обеспечения, но даже сетевого интерфейса. Для связи системы мониторинга с такими объектами необходимо создание промежуточного программного обеспечения – SNMP-агента, который, с одной стороны, обеспечивает связь с SNMP-менеджером, установленным на станции мониторинга, а с другой – опрашивает контролируемые объекты (датчики). Информация, которую система мониторинга запрашивает от агентов, хранится в специальной информационной базе данных MIB (Management Information Base). Обычно производитель аппаратного или программного обеспечения разрабатывает MIB-файл со своим собственным набором переменных, которые должны быть присоединены к дереву MIB. В SNMP каждой переменный присваивается уникальный идентификатор объекта (Object Identifier, OID). Пространство имён OID является иерархическим и контролируется организацией по распределению номеров в Интернете (Internet Assigned Numbers Authority, IANA).

SNMP используется для получения от сетевых устройств информации об их статусе и других характеристиках, которые хранятся в базе данных MIB. SNMP позволяет применять дополнительные агенты для поддержки баз MIB различных производителей. Агент взаимодействует с контролируемым ресурсом по нестандартному интерфейсу, а с менеджером – по стандартному протоколу через сеть. Таким образом, основная задача при создании SNMP-агента состоит в реализации функции, поддерживающей обработку запросов к заданному OID.

Генерирование кода шаблона

Предполагается, что перед началом работ произведена установка пакета Net-SNMP версии 5.4.2.1 (http://www.net-snmp.org/download.html).

Создадим новый каталог для кода шаблона SNMP-агента (далее Агент):

```
$ cd ~
$ mkdir sensTable
$ cd sensTable
```

Переменные, доступные через SNMP, организованы в иерархии и описаны соответствующим MIB. Вопросы, связанные с использованием языка абстрактной синтаксической нотации ASN.1 (http://www.amazon.com/s/ref?url=search-alias%3Dstripbooks&field-keywords=ASN.1) (стандарт ISO 8824:1987, рекомендации ITU-T X.208) и написанием MIB, выходят за рамки данного повествования, поэтому будем считать, что есть некий заранее созданный MIB-файл, например, SENSOR-MIB.mib.txt, в котором нас будет интересовать объект с именем sensTable. Об этом объекте речь пойдет ниже.

Для целей создания Агента в качестве временной замены файла SENSOR-MIB.mib.txt был использован MIB-файл реального устройства Poseidon 3262 (http://www.hw-group.com/products/poseidon/poseidon_3262_en.html).

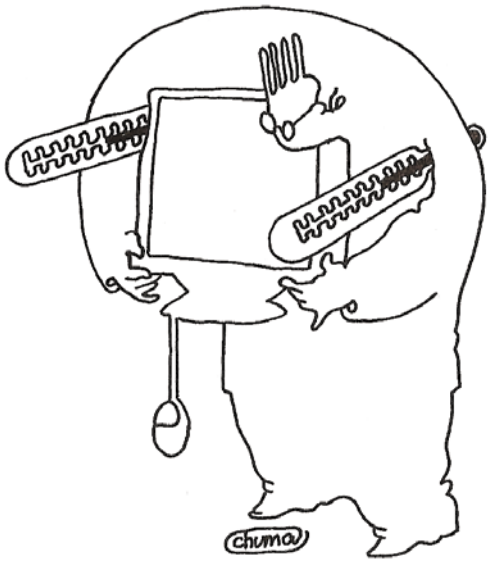
Воспользуемся утилитой mib2c (http://www.net-snmp.org/docs/man/mib2c.html) пакета Net-SNMP для генерации шаблона Агента:

```
$ env MIBS="/usr/share/snmp/mibs/SENSOR-MIB.mib.txt" ↵
        mib2c -c mib2c.mfd.conf \ sensTable
```

На все вопросы программы mib2c следует давать ответы, предусмотренные в ней по умолчанию (DEFAULT). Единственное исключение составляет вопрос Which method would you like to use to gather data about available rows? (какой метод вы хотели бы использовать, чтобы накапливать данные о доступных строках таблицы). На этот вопрос следует выбрать ответ «2) container-cached», который сообщит программе mib2c о вашем желании использовать в Агенте поддержку кэширования.

После ответа на все вопросы программа mib2c сгенерирует шаблон кода Агента:

```
writing to defaults/table-sensTable.m2d
Starting MFD code generation...
writing to sensTable.h
| +-> Processing table sensTable
```



Это «скелетный» проект агента, который может быть использован как основа для развития

```
writing to defaults/node-sensName.m2d
writing to defaults/node-sensState.m2d
writing to defaults/node-sensString.m2d
writing to defaults/node-sensValue.m2d
writing to defaults/node-sensValueRaw.m2d
writing to defaults/node-sensID.m2d
writing to defaults/node-sensUnit.m2d
writing to defaults/node-sensIndex.m2d
writing to sensTable.c
writing to sensTable_data_get.h
writing to sensTable_data_get.c
| |   +-> Processing nonindex sensName
| |   +-> Processing nonindex sensState
| |   +-> Processing nonindex sensString
| |   +-> Processing nonindex sensValue
| |   +-> Processing nonindex sensValueRaw
| |   +-> Processing nonindex sensID
| |   +-> Processing nonindex sensUnit
writing to sensTable_data_set.h
writing to sensTable_data_set.c
writing to sensTable_oids.h
writing to sensTable_enums.h
writing to sensTable_interface.h
writing to sensTable_interface.c
writing to sensTable_data_access.h
writing to sensTable_data_access.c
writing to sensTable-README-FIRST.txt
writing to sensTable-README-sensTable.txt
writing to sensTable_Makefile
writing to sensTable_subagent.c
running indent on sensTable_data_access.h
running indent on sensTable_subagent.c
running indent on sensTable_interface.c
running indent on sensTable_enums.h
running indent on sensTable_data_access.c
running indent on sensTable_interface.h
running indent on sensTable_data_set.c
running indent on sensTable_data_get.h
running indent on sensTable.h
running indent on sensTable.c
running indent on sensTable_data_set.h
running indent on sensTable_data_get.c
running indent on sensTable_oids.h
```

Результатом диалога с mib2c будет сгенерированный этой утилитой набор файлов, среди которых в первую очередь заслуживают внимания sensTable-README-FIRST.txt и sensTable-README-sensTable.txt.

Файл sensTable-README-FIRST.txt содержит указания о том, какие последующие действия следует предпринять, а из файла sensTable-README-sensTable.txt можно получить подробные сведения о тех шагах, которые были проделаны на этапе генерации шаблона.

В частности, следуя указаниям из файла sensTable-README-FIRST.txt, получим TODO-лист необходимых, по мнению mib2c, работ, которые нужно выполнить для получения работоспособного Агента:

```
$ grep -n "TODO:" *.[ch] | sed 's/\([^ ]*\) ↵
       \(.*\)TODO\(.*\)/\3 (\1)/' | sort -n
```

```
:099:x: ******************************** (sensTable.h:205:)
:100:r: Review all context structures (sensTable.h:62:)
:101:o: |-> Review sensTable registration context.
(sensTable.h:65:)
:110:r: |-> Review sensTable data context structure.
(sensTable.h:71:)
:120:r: |-> Review sensTable mib index. (sensTable.h:121:)
:121:r: |   |-> Review sensTable max index length.
(sensTable.h:135:)
:130:o: |-> Review sensTable Row request (rowreq) context.
(sensTable.h:143:)
:131:o: |   |-> Add useful data to sensTable rowreq
context. (sensTable.h:164:)
:180:o: Review sensTable cache timeout.
(sensTable_data_access.h:40:)
:199:x: ********************************(sensTable.h:206:)
:200:r: Implement sensTable data context functions.
(sensTable_data_get.c:41:)
:210:o: |-> Perform extra sensTable rowreq initialization.
(eg DEFVALS) (sensTable.c:134:)
:211:o: |-> Perform extra sensTable rowreq cleanup.
(sensTable.c:151:)
:230:M: Implement sensTable get routines.
(sensTable_data_get.c:18:)
:231:o: |-> Extract the current value of the sensID data.
(sensTable_data_get.c:476:)
:231:o: |-> Extract the current value of the sensName data.
(sensTable_data_get.c:167:)
:231:o: |-> Extract the current value of the sensState
data. (sensTable_data_get.c:235:)
:231:o: |-> Extract the current value of the sensString
data. (sensTable_data_get.c:303:)
:231:o: |-> Extract the current value of the sensUnit data.
(sensTable_data_get.c:528:)
:231:o: |-> Extract the current value of the sensValue
data. (sensTable_data_get.c:371:)
:231:o: |-> Extract the current value of the sensValueRaw
data. (sensTable_data_get.c:424:)
:240:M: Implement sensTable mapping routines (if any).
```

```
(sensTable_data_get.c:19:)
:299:x: ****************************** (sensTable.h:207:)
:300:o: Perform sensTable one-time module initialization.
(sensTable.c:43:)
:301:o: Perform sensTable one-time table initialization.
(sensTable.c:78:)
:302:o: |->Initialize sensTable user context
(sensTable.c:82:)
:303:o: Initialize sensTable data.
(sensTable_data_access.c:57:)
:345:A: Set up sensTable cache properties.
(sensTable_data_access.c:132:)
:350:M: Implement sensTable data load
(sensTable_data_access.c:173:)
:351:M: |-> Load/update data in the sensTable container.
(sensTable_data_access.c:244:)
:352:M: |   |-> set indexes in new sensTable rowreq
context. (sensTable_data_access.c:280:)
:352:r: |   |-> populate sensTable data context.
(sensTable_data_access.c:299:)
:380:M: Free sensTable container data.
(sensTable_data_access.c:419:)
:390:o: Prepare row for request.
(sensTable_data_access.c:444:)
:399:x: ****************************** (sensTable.h:208:)
:499:x: ****************************** (sensTable.h:209:)
:510:o: Perform sensTable pre-request actions.
(sensTable.c:168:)
:511:o: Perform sensTable post-request actions.
(sensTable.c:194:)
```

Следует подчеркнуть, что данный документ никоим образом не подменяет собой официальной документации на пакет Net-SNMP, а предполагает и настоятельно рекомендует предварительное с ней ознакомление.

С общим описанием процесса написания Агента можно ознакомиться здесь http://www.net-snmp.org/wiki/index.php/TUT:Writing_a_Subagent.

Итак, у нас теперь есть шаблон Агента и перечень необходимых работ.

Структура данных

В реальных условиях температурные датчики (могут быть) подключены к Ethernet посредством шлюза Modbus/TCP. Показания температурных датчиков запрашивает и получает отдельный процесс, который сохраняет полученные значения в базе данных. Будем считать, что текущие показания записываются в файл в формате csv и с заданной периодичностью обновляются.

Структура записи csv-файла отображает поля таблицы sensTable:

```
$ snmptranslate -m /usr/share/snmp/mibs/ ↵
        SENSOR-MIB.mib.txt -Tp -IR sensTable
```

```
+--sensTable(3)
   |
   +--sensEntry(1)
      |  Index: sensIndex
      |
      +-- ---- INTEGER   sensIndex(1)
      |        Textual Convention: PositiveInteger
      |        Range: 1..2147483647
      +-- -R-- String    sensName(2)
      |        Textual Convention: SensorName
      |        Size: 0..15
      +-- -R-- EnumVal   sensState(4)
      |        Textual Convention: SensorState
      |        Values: invalid(0), normal(1),
      |        alarmstate(2), alarm(3)
      +-- -R-- String    sensString(5)
      |        Textual Convention: SensorString
      |        Size: 0..10
      +-- -R-- INTEGER   sensValue(6)
      |        Textual Convention: SensorValue
      +-- -R-- INTEGER   sensValueRaw(7)
      |        Textual Convention: SensorValue
      +-- -R-- INTEGER   sensID(8)
      |        Textual Convention: SensorID
      |        Range: 0..65535
      +-- -R-- EnumVal   sensUnit(9)
               Textual Convention: UnitType
               Values: celsius(0), fahrenheit(1),
               kelvin(2), percent(3), volt(4),
               miliamper(5), nounit(6), pulse(7), switch(8)
```

Например, строка csv-файла:

```
1,"Sensor 241",1,"22.6",226,0,14991,0
```

содержит следующие данные:

sensIndex – 1;
sensName – Sensor 241;
sensState – 1 (normal);
sensString – 22.6;
sensValue – 226;
sensValueRaw – 0;
sensID – 14991;
sensUnit – 0 (celsius).

Предположим, что в нашем распоряжении есть шесть датчиков и файл sensor.csv, хранящий их текущие значения:

```
1,"Sensor 241",1,"22.6",226,0,14991,0
2,"Sensor 242",1,"11.4",114,0,14992,0
3,"Sensor 243",1,"2.2",22,0,14993,0
4,"Sensor 244",1,"21.5",215,0,14994,0
5,"Sensor 245",1,"20.6",206,0,14995,0
6,"Sensor 246",1,"24.0",240,0,14996,0
```

Далее необходимо обеспечить периодическое обновление данных в файле sensor.csv. Для отладки Агента нет необходимости использовать аппаратные датчики и шлюз. Их вполне можно заменить программной симуляцией. Другими словами, необходима программа, которая будет имитировать выдачу текущих значений датчиков путем генерации значений температур в заданном диапазоне и записывать эти значения в файл sensor.csv.

Для решения этой задачи можно использовать следующий скрипт:

```
#!/bin/bash
CSVPATH="/home/snmp/sensTable/"
CSVFILE=${CSVPATH}"sensor.csv"

SENSOR[1]="1,\"Sensor 241\",1,\"24.5\",245,0,14991,0"
SENSOR[2]="2,\"Sensor 242\",1,\"36.6\",366,0,14992,0"
SENSOR[3]="3,\"Sensor 243\",1,\"12.2\",122,0,14993,0"
SENSOR[4]="4,\"Sensor 244\",1,\"21.5\",215,0,14994,0"
SENSOR[5]="5,\"Sensor 245\",1,\"20.6\",206,0,14995,0"
SENSOR[6]="6,\"Sensor 246\",1,\"31.7\",317,0,14996,0"

if [ ! -f ${CSVFILE} ]
then
  for index in 1 2 3 4 5 6    # Six lines.
  do
    printf "%s\n" "${SENSOR[index]}" >> ${CSVFILE}
  done
else
cat $CSVFILE | awk -F"," '
BEGIN {
    srand(systime())
    #         [min,max] values
    SENS[1] = "21,34"  # 21.0 .. 34.9
    SENS[2] = "10,15"  # 10.0 .. 15.9
    SENS[3] = "1,6"    # 1.0 .. 6.9
    SENS[4] = "0,0"    # do nothing
    SENS[5] = "0,0"    # do nothing
```

```
    SENS[6] = "23,29"  # 23.0 .. 29.9
}

function randint(n) { return int(n * rand()) }

{
   split(SENS[NR], temp, ",")
   min = temp[1]
   max = temp[2]
   if (min == max) {
      SENS[NR] = $0
   } else {
        grad = min + randint(max-min)
        fraction = randint(9)
        split($0, sens, ",")
        gsub("\"", "", sens[2])
        SENS[NR] = sprintf("%s,\"%s\",%s,\"%d.%1d\", ↵
                %d%1d,%s,%s,%s",
                  sens[1],sens[2],sens[3],
                  grad,fraction,grad,fraction,
                  sens[6],sens[7],sens[8])
   }
}
END {
    for (x in SENS) {
      printf "%s\n", SENS[x] | "sort > '$CSVFILE'"
    }
}'
fi
```

Текст скрипта следует поместить в файл sensor.sh. Вариантов реализации данного скрипта может быть множество, поэтому простор для творчества оставлен открытым.

Периодичность обновления (цикл опроса датчиков) можно обеспечить при помощи cron:

```
$ crontab -e
```

В файл добавим следующие строки:

```
SHELL=/bin/bash
*/10 * * * *   /home/snmp/sensTable/sensor.sh > /dev/null ↵
       2>&1
```

В результате программа sensor.sh будет выполняться каждые 10 минут, что соотвествует длительности реального цикла опроса температурных датчиков.

Задача Агента состоит в том, чтобы по запросу прочитать этот csv-файл и выдать запрашиваемые менеджером данные.

Взаимодействие с контролируемым ресурсом

Теперь необходимо расширить сгенерированный программой mib2c код шаблона Агента для чтения данных из файла sensor.csv. Для этого потребуется написать несколько фрагментов кода на языке C и добавить их в файл sensTable_data_access.c. Полный текст этого файла находится в архиве, который можно скачать по адресу: www.samag.ru раздел «Исходный код».

Требуемые изменения были сохранены в файле patchfile.patch:

```
$ diff sensTable_data_access.c.orig ↵
       sensTable_data_access.c > patchfile.patch
```

где sensTable_data_access.c.orig – исходный файл, сгенерированный программой mib2c, а sensTable_data_access.c – модифицированная версия.

Файл patchfile.patch можно использовать для получения обновленной версии файла sensTable_data_access.c:

```
$ patch sensTable_data_access.c.orig -i ↵
       patchfile.patch -o sensTable_data_access.c
```

После внесения изменений проведем сборку Агента:

```
$ make -f sensTable_Makefile
```

В файле sensTable_Makefile есть две заслуживающие особого внимания команды:

- net-snmp-config --base-cflags;
- net-snmp-config --agent-libs

которые выдают полезную для дальнейшего совершенствования Агента информацию об используемых библиотеках и файлах заголовков:

```
$ net-snmp-config --base-cflags
```

```
-DNETSNMP_ENABLE_IPV6 -march=i586 -mtune=i686
-fmessage-length=0 -O2 -Wall
-D_FORTIFY_SOURCE=2 -fstack-protector -funwind-tables
-fasynchronous-unwind-tables -g -fno-strict-aliasing
-fstack-protector-all -Ulinux
-Dlinux=linux -I/usr/include/rpm -D_REENTRANT -D_GNU_SOURCE
-DPERL_USE_SAFE_PUTENV
-DDEBUGGING -fno-strict-aliasing -pipe -D_LARGEFILE_SOURCE
-D_FILE_OFFSET_BITS=64
-I/usr/lib/perl5/5.10.0/i586-linux-thread-multi/CORE
-I/usr/include
```

```
$ net-snmp-config --agent-libs
```

```
-L/usr/lib -lnetsnmpagent -lnetsnmphelpers -lnetsnmpmibs
-lnetsnmp -lz -Wl,-E -Wl,
-rpath,/usr/lib/perl5/5.10.0/i586-linux-thread-multi/CORE
```

Перечень динамических библиотек, используемых «свежеиспеченным» Агентом:

```
$ ldd sensTable
```

```
linux-gate.so.1 =>  (0xffffe000)
libnetsnmpagent.so.15 => /usr/lib/libnetsnmpagent.so.15
(0xb7f75000)
libnetsnmphelpers.so.15 => /usr/lib/libnetsnmphelpers.so.15
(0xb7f4b000)
libnetsnmpmibs.so.15 => /usr/lib/libnetsnmpmibs.so.15
(0xb7d7d000)
libnetsnmp.so.15 => /usr/lib/libnetsnmp.so.15 (0xb7cb3000)
libz.so.1 => /lib/libz.so.1 (0xb7c9d000)
libc.so.6 => /lib/libc.so.6 (0xb7b41000)
libcrypto.so.0.9.8 => /usr/lib/libcrypto.so.0.9.8
(0xb79da000)
libwrap.so.0 => /lib/libwrap.so.0 (0xb79cf000)
libperl.so => /usr/lib/perl5/5.10.0/i586-linux-thread-
multi/CORE/libperl.so (0xb775b000)
libnsl.so.1 => /lib/libnsl.so.1 (0xb7741000)
libdl.so.2 => /lib/libdl.so.2 (0xb773c000)
libm.so.6 => /lib/libm.so.6 (0xb7713000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0xb76dc000)
libutil.so.1 => /lib/libutil.so.1 (0xb76d8000)
libpthread.so.0 => /lib/libpthread.so.0 (0xb76be000)
librpm-4.4.so => /usr/lib/librpm-4.4.so (0xb7627000)
librpmio-4.4.so => /usr/lib/librpmio-4.4.so (0xb7545000)
libpopt.so.0 => /lib/libpopt.so.0 (0xb753c000)
libsensors.so.3 => /usr/lib/libsensors.so.3 (0xb7504000)
/lib/ld-linux.so.2 (0xb7fcc000)
librpmdb-4.4.so => /usr/lib/librpmdb-4.4.so (0xb73fb000)
librt.so.1 => /lib/librt.so.1 (0xb73f1000)
libbz2.so.1 => /lib/libbz2.so.1 (0xb73e1000)
libselinux.so.1 => /lib/libselinux.so.1 (0xb73c4000)
libsysfs.so.2 => /lib/libsysfs.so.2 (0xb73b7000)
```

Конфигурирование и запуск Агента

Оригинальная документация на пакет Net-SNMP гласит:

```
Enable AgentX in your Net-SNMP (UCD-SNMP) daemon
```

```
configuration (in SuSE it is /etc/snmp/snmpd.conf).
There should be line "master agentx". Restart snmpd.
```

После добавления строки «master agentx» в файл /etc/snmp/snmpd.conf остается только перезапустить snmpd:

```
# service snmpd start
```

Проверим результат запуска:

```
# ps fax | grep snmpd | grep -v grep
```

```
18959 ?        SNl    0:21 /usr/sbin/snmpd -r -A -LF d
/var/log/net-snmpd.log -p /var/run/snmpd.pid
```

Запуск Агента в режиме отладки:

```
# cd /home/snmp/sensTable
# ./sensTable -f ↵
        -DsensTable,verbose:sensTable,internal:sensTable
```

Запуск Агента в нормальном режиме:

```
# ./sensTable -f &
```

Проверим результат запуска:

```
# ps fax | grep sensTable | grep -v grep
```

```
10033 ?        S        0:02 ./sensTable -f
```

Запросим у Агента информацию:

```
# snmpwalk -v2c localhost -c public  ↵
        .1.3.6.1.4.1.21796.3.3.3
```

```
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.1 =
STRING: "Sensor 241"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.2 =
STRING: "Sensor 242"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.3 =
STRING: "Sensor 243"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.4 =
STRING: "Sensor 244"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.5 =
STRING: "Sensor 245"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.6 =
STRING: "Sensor 246"
SNMPv2-SMI::enterprises.21796.3.3.3.1.4.1 = INTEGER: 1
SNMPv2-SMI::enterprises.21796.3.3.3.1.4.2 = INTEGER: 1
SNMPv2-SMI::enterprises.21796.3.3.3.1.4.3 = INTEGER: 1
SNMPv2-SMI::enterprises.21796.3.3.3.1.4.4 = INTEGER: 1
SNMPv2-SMI::enterprises.21796.3.3.3.1.4.5 = INTEGER: 1
SNMPv2-SMI::enterprises.21796.3.3.3.1.4.6 = INTEGER: 1
SNMPv2-SMI::enterprises.21796.3.3.3.1.5.1 = STRING: "24.5"
SNMPv2-SMI::enterprises.21796.3.3.3.1.5.2 = STRING: "36.6"
SNMPv2-SMI::enterprises.21796.3.3.3.1.5.3 = STRING: "12.2"
SNMPv2-SMI::enterprises.21796.3.3.3.1.5.4 = STRING: "21.5"
SNMPv2-SMI::enterprises.21796.3.3.3.1.5.5 = STRING: "20.6"
SNMPv2-SMI::enterprises.21796.3.3.3.1.5.6 = STRING: "31.7"
SNMPv2-SMI::enterprises.21796.3.3.3.1.6.1 = INTEGER: 245
SNMPv2-SMI::enterprises.21796.3.3.3.1.6.2 = INTEGER: 366
SNMPv2-SMI::enterprises.21796.3.3.3.1.6.3 = INTEGER: 122
SNMPv2-SMI::enterprises.21796.3.3.3.1.6.4 = INTEGER: 215
SNMPv2-SMI::enterprises.21796.3.3.3.1.6.5 = INTEGER: 206
SNMPv2-SMI::enterprises.21796.3.3.3.1.6.6 = INTEGER: 317
SNMPv2-SMI::enterprises.21796.3.3.3.1.7.1 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.7.2 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.7.3 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.7.4 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.7.5 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.7.6 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.8.1 = INTEGER: 14991
SNMPv2-SMI::enterprises.21796.3.3.3.1.8.2 = INTEGER: 14992
SNMPv2-SMI::enterprises.21796.3.3.3.1.8.3 = INTEGER: 14993
SNMPv2-SMI::enterprises.21796.3.3.3.1.8.4 = INTEGER: 14994
SNMPv2-SMI::enterprises.21796.3.3.3.1.8.5 = INTEGER: 14995
SNMPv2-SMI::enterprises.21796.3.3.3.1.8.6 = INTEGER: 14996
SNMPv2-SMI::enterprises.21796.3.3.3.1.9.1 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.9.2 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.9.3 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.9.4 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.9.5 = INTEGER: 0
SNMPv2-SMI::enterprises.21796.3.3.3.1.9.6 = INTEGER: 0
```

Результаты наших усилий

Service Status Details
For Host 'poseidon'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
poseidon	Sensor_1	OK	06-17-2009 12:14:58	81d 2h 0m 22s	1/3	SNMP OK - "31.1"
	Sensor_2	OK	06-17-2009 12:16:31	81d 1h 59m 10s	1/3	SNMP OK - "14.4"
	Sensor_3	OK	06-17-2009 12:18:01	81d 1h 56m 3s	1/3	SNMP OK - "2.7"
	Sensor_4	OK	06-17-2009 12:20:39	81d 1h 55m 2s	1/3	SNMP OK - "21.5"
	Sensor_5	OK	06-17-2009 12:21:40	81d 1h 54m 1s	1/3	SNMP OK - "20.6"
	Sensor_6	OK	06-17-2009 12:22:41	81d 1h 53m 0s	1/3	SNMP OK - "28.1"

```
# snmpwalk -v2c localhost -c public  ↵
        .1.3.6.1.4.1.21796.3.3.3.1.2
```

```
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.1 =
STRING: "Sensor 241"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.2 =
STRING: "Sensor 242"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.3 =
STRING: "Sensor 243"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.4 =
STRING: "Sensor 244"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.5 =
STRING: "Sensor 245"
SNMPv2-SMI::enterprises.21796.3.3.3.1.2.6 =
STRING: "Sensor 246"
```

```
# snmpwalk -v2c localhost -c public  ↵
        .1.3.6.1.4.1.21796.3.3.3.1.6.1
```

```
SNMPv2-SMI::enterprises.21796.3.3.3.1.6.1 = INTEGER: 245
```

Разумеется, что значение OID в команде snmpwalk от .1.3.6.1.4.1.21796.3.3.3. специфично исключительно для используемого здесь MIB-файла.

Конфигурирование Nagios

Процесс установки и конфигурирования Nagios (http://www.nagios.org) подробно рассматривался во многих публикациях, поэтому здесь предполагается, что Nagios (включая Nagios check_snmp plugin) уже установлен в системе, и нам остается только внести необходимые изменения в его конфигурационные файлы.

В файле /etc/nagios/nagios.cfg необходимо раскомментировать строку:

```
cfg_file=/etc/nagios/objects/switch.cfg
```

и внести необходимые изменения. Содержимое файла /etc/nagios/objects/switch.cfg:

```
define host{
        use             generic-switch
        host_name       poseidon
        alias           Poseidon 3262
        address         127.0.0.1
```

```
        hostgroups      switches
        }
define hostgroup{
        hostgroup_name  switches
        alias           Network Switches
        }
define service{
        use                     generic-service
        host_name               poseidon
        service_description     Sensor_1
        check_command           check_snmp!-C public -o ↵
                .1.3.6.1.4.1.21796.3.3.3.1.5.1
        }
define service{
        use                     generic-service
        host_name               poseidon
        service_description     Sensor_2
        check_command           check_snmp!-C public -o ↵
                .1.3.6.1.4.1.21796.3.3.3.1.5.2
        }
define service{
        use                     generic-service
        host_name               poseidon
        service_description     Sensor_3
        check_command           check_snmp!-C public -o ↵
                .1.3.6.1.4.1.21796.3.3.3.1.5.3
        }
define service{
        use                     generic-service
        host_name               poseidon
        service_description     Sensor_4
        check_command           check_snmp!-C public -o ↵
                .1.3.6.1.4.1.21796.3.3.3.1.5.4
        }
define service{
        use                     generic-service
        host_name               poseidon
        service_description     Sensor_5
        check_command           check_snmp!-C public -o ↵
                .1.3.6.1.4.1.21796.3.3.3.1.5.5
        }
define service{
        use                     generic-service
        host_name               poseidon
        service_description     Sensor_6
        check_command           check_snmp!-C public -o ↵
                .1.3.6.1.4.1.21796.3.3.3.1.5.6
        }
```

Перезапустим Nagios, кликнем по ссылке Service Detail и посмотрим на результаты наших усилий (см. рисунок).

В столбце Status Information можно наблюдать значения температур, полученные от каждого датчика.

В основе системы мониторинга лежит элементарная схема взаимодействия агента и менеджера. На основе этой схемы могут быть построены системы практически любой сложности с большим количеством агентов и менеджеров разного типа.

Пользуясь API, предоставляемым Net-SNMP, разработчик может создавать законченную систему мониторинга, которая может контролировать специфическое оборудование.

Агент в SNMP – обрабатывающий элемент, который обеспечивает менеджерам, входящим в систему мониторинга, доступ к значениям переменных MIB и тем самым дает им возможность реализовать функции по наблюдению за устройствами. EOF

Визитка

АНТОН БОРИСОВ, специализируется на экспертизе аппаратных и программных решений

Веб-топ-решение
на основе Ulteo Virtual Desktop

Для организации работы сервера, предоставляющего инфраструктуру веб-доступа к приложениям компании, вовсе не обязательно использовать платные коммерческие разработки

В качестве аналога Sun Secure Global Desktop можно попробовать развернуть и GPL-решение – Open Source Enterprise Virtual Desktop от компании Ulteo. Чем хороши коммерческие решения? В первую очередь тем, что стараются предоставить пользователям максимум возможностей, которые по-большому счету многим потребителям и не потребуются. Сравните, например, поддержку работы в Secure Global Desktop (см. №9 за 2009 год) терминалов стандарта TN3270/TN5250, которые используются при коммуникации с серверами IBM iSeries и AS/400. Много ли у нас в стране предприятий, у которых установлен хотя бы один сервер с логотипом IBM? Когда стоит выбор, а что использовать в качестве серверного решения для веб-топов, не последнюю роль играет его стоимость. И хотя для варианта Open Source-затраты на внедрение не являются нулевыми – помните, что затраты косвенно включены в стоимость работы ваших инженерных кадров, – они все же значительно предпочтительнее коммерческих решений. В силу того, что упомянутые кадры, которые разбираются в технологиях, смогут решить нетривиальную задачу, которая может появиться на предприятии при эксплуатации веб-топ-решения. При наличии GPL-продукта решение будет найдено намного быстрее.

Что такое Open Virtual Desktop?

Это связка из двух центров – сервер на основе LAMP (Linux/Apache/MySQL/PHP) для авторизации и управления пользователями и приложениями – Session Manager, и сам сервер приложений – Application Server. И хотя их рекомендуется размещать на разных физических машинах, функционально их можно объединить и разместить на одной платформе (см. рис. 1).

Пользователям, которые зарегистрированы на Open Virtual Deskop, предоставляется возможность запустить в браузере как приложения с Linux-сервера приложений, так и приложения с сервера приложений на Windows-платформе. К сожалению, такое стороннее ПО, как плагин от Java, должно быть установлено на стороне клиента – иначе вся визуализация, которая реализована на основе VNC-протокола, будет недоступна.

На сервере приложений (Application Server) запускается реализация X-сервера – Xtightvnc. Браузер пользователя исполняет java-апплет, в котором реализован VNC-протокол, и, авторизовавшись на сервере SessionManager, получает доступ к виртуальному столу.

Для авторизации пользователей в Windows-системах может использоваться протокол от Active Directory либо от LDAP-хранилища. Либо, когда развернута информационная среда для супермалой компании, по самому простому варианту – добавить одноименные учетные записи как в Linux, так и в Windows-сервер.

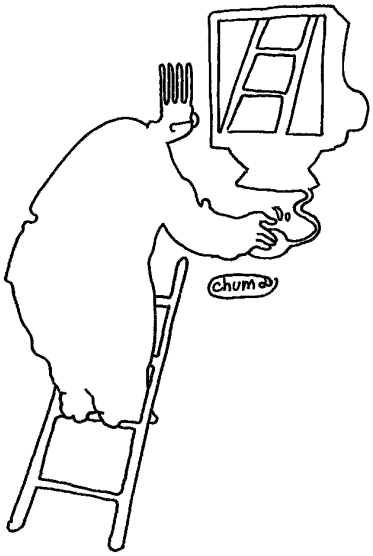
Компания Ulteo распространяет серверные компоненты в качестве RPM-пакетов для RHEL/CentOS/Fedora либо через собственный репозиторий для Debian-систем (или основанных на нем). Можно также взять tarball с исходными текстами. Или же скачать готовый, со всеми компонентами, ISO-образ с Ubuntu. В качестве обзорного примера предлагаю воспользоваться именно этим вариантом, т.к. для выделенной машины выбор дистрибутива не принципиален, а настройка обновлений для системы и приложений в Ubuntu абсолютно аналогична той, что и в Debian.

Сервер приложений

Скачиваем ISO, устанавливаем на выделенную машину, делаем несколько шагов по регистрации пользователей – и через несколько минут Ulteo Virtual Desktop готов и работает. Теперь можно приступить к более тонкой настройке.

Заходим через браузер по адресу http://ovd-server/sessionamanager/admin, где ovd-server – IP-адрес только что развернутого решения. По умолчанию используется внутренний список пользователей. Зайдем под зарезервированной учетной записью – admin/admin. В дальнейшем рекомендуется поменять имя/пароль от администратора. Эти данные хранятся в файле /etc/ulteo/sessionmanager/config.inc.php, причем пароль – это хеш-запись от настоящего пароля, закодированного по методу MD5 (см. рис .2).

Первым делом необходимо будет зарегистрировать Application Server (который в нашем случае находится по такому же адресу, что и OVD-server) – вкладка Servers →



Когда стоит выбор, а что использовать в качестве серверного решения для веб-топов, не последнюю роль играет его стоимость

Unregistered Servers. Нажимаем на кнопку Register, а затем переводим в состояние Production – кнопка Switch to Production. В теории сервер готов. Однако стоит зайти в его свойства и вместо имени localhost внести либо его настоящее DNS-имя, либо прописать IP-адрес (см. рис. 3).

И можно расширить количество одновременных подключений к этому серверу приложений – вместо используемых по умолчанию 20 подключений расширить до необходимых для вашего предприятия.

В этой же вкладке, чуть ниже, также видны те приложения, что будут доступны для запуска. Но о них позже.

И заключительный штрих – не забудем добавить пользователей, которые будут обслуживаться данной веб-топ-системой. Перейдем во вкладку Users и создадим наш собственный список пользователей – радиокнопка I want to create my own users. Так как в моем окружении не использовался ни Active Directory, ни LDAP-сервер, я выбрал вариант хранения учетных записей пользователей в собственной базе Ulteo (см. рис. 4).

Добавление учетных записей также достаточно тривиальное занятие – для этого предназначена вкладка Users. Последовательно проходите все регистрационные шаги – присваиваете уникальные ID пользователю, включаете его учетную запись в определенную пользовательскую группу (например, Department1) и в группу приложений (например, BusinessApps). Чем указанные группы отличаются друг от друга? Последняя, т.е. группа приложений, – это список из пользовательских программ, которые зарегистрированы на сервере приложений Ulteo и которые можно запустить. В то время как пользовательская группа – это список зарегистрированных пользователей, которым назначена та или иная группа приложений. Таким образом можно выделить группы пользователей не только по организационному признаку (подразделения, департаменты), но и по функциональному (выполнение определенного текущего проекта). После этого довольные пользователи могут заходить браузером на Ulteo Virtual Desktop, причем отличить виртуальный рабочий стол, который находится

Рисунок 1. Архитектура веб-топ-решения от Ulteo

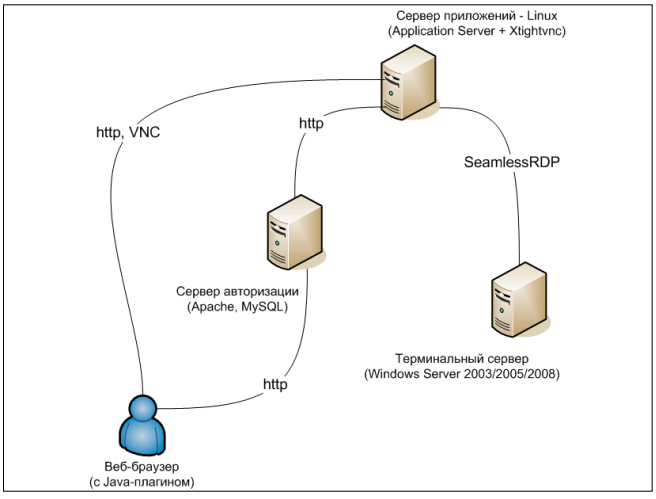
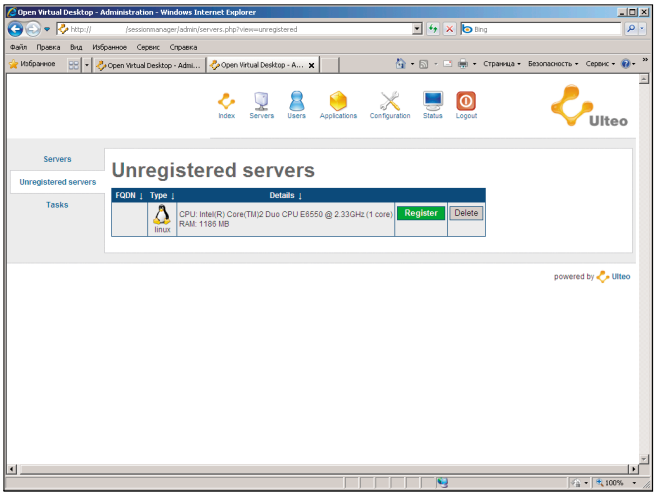


Рисунок 2. Сервер приложений найден, но пока не зарегистрирован



где-то там, на сервере, от реального рабочего места будет сложновато (см. рис. 5).

Больше приложений, красивых и полезных

Как вы могли заметить, пользователю на данном этапе предлагается сравнительно небольшой выбор по программному обеспечению – только те приложения, что были установлены внутри Application Server, который в свою очередь основан на Linux. Что делать, когда требуется запустить Win32-приложения? Используется ли какая-либо виртуализация или запуск Windows-приложений происходит несколько иначе?

В Ulteo решили не усложнять с виртуализацией и поступили намного проще. Для запуска Windows-приложений все-таки требуется терминальный сервер с включенной RDP-службой. Для экстремально ограниченного варианта, когда нет терминального сервера Windows, можно попробовать выделить машину на базе Windows XP Professional с включенным Remote Desktop Connection (дистанционное управление рабочим столом). В этом случае работа приложений будет возможна только в однопользовательском режиме, т.е. запустить любое Windows-приложение, что зарегистрировано на Windows XP, сможет только один зарегистрированный пользователь Ulteo. На этот сервер устанавливается легкий клиент от Ulteo, который соединяется с главным сервером приложений – Application Server – и «заворачивает» вывод произвольного приложения с терминального сервера в seamless RDP-сессию. То есть разработчики взяли открытый код от проекта SeamlessRDP компании Cendio AB и модифицировали его под задачи виртуального веб-десктопа. Для пользователя, работающего с виртуальным десктопом, никакой принципиальной разницы между приложениями, запущенными из Linux-среды и из Windows, нет – есть, правда, небольшое визуальное отличие в оформлении окон. Вопросы могут появиться, когда необходимо будет переписать/передать/сообщить о расположении сохраненного документа коллегам. И здесь придется подумать, как лучше организовать общее сетевое рабочее пространство – использовать единое SAN-хранилище или что-то еще.

Приступим к практической реализации. Загружаем Windows-агент и устанавливаем на терминальный сервер. В файле C:\Program Files\Ulteo\Open Virtual Desktop\ulteo-ovd.conf вставляем нужные адреса как самого терминального сервера (параметр SERVERNAME), так и сервера приложений Ulteo (IP-адрес в параметре SESSION_MANAGER_URL). Теперь остается перезапустить службу Ulteo Open Virtual Desktop agent и зарегистрировать появившийся сервер в системе Ulteo. И, как это было описано выше с Linux-сервером, перевести его в production mode.

Листинг 1. Пример конфигурационного файла Windows-агента Ulteo

```
SERVERNAME=10.10.250.10
SESSION_MANAGER_URL=http://10.10.2.10/sessionmanager
LOG_FLAGS="info warn error"
WEBPORT=8082
```

После этой операции имеет смысл завести новую группу для Windows-приложений и для нужных пользователей подключить эту группу. Тогда у этих пользователей на виртуальном рабочем столе появятся ярлыки от Windows-приложений с терминального сервера.

Однако остановимся на этом шаге. Для развитых инфраструктур, где развернуто множество серверов, встретить доменную или LDAP-авторизацию вполне обычное дело. Иное дело, когда рассматривается малое предприятие – объективных причин заводить домен ради 10 пользователей нет. Проще и быстрее зарегистрировать их вручную на нескольких серверах. Можно ли разворачивать Ulteo в ситуации, когда нет ни LDAP, ни Active Directory? Можно – внесем пару строк в модуль авторизации – файл /usr/share/ulteo/sessionmanager/startsession.php.

Найдите блок авторизации пользователей (строки 1-18) и добавьте новый код default (строки 19-21):

```
1) switch ($prefs->get('UserDB', 'enable')) {
2)     case 'activedirectory':
3)         $prefs_ad = $prefs->get('UserDB', ↵
               'activedirectory');
4)         $windows_login = $user->getAttribute( ↵
               'real_login').'@'.$prefs_ad['domain'];
6)         break;
7)     case 'ldap':
```

Рисунок 3. Тонкая настройка Application Server позволит оптимизировать его работу

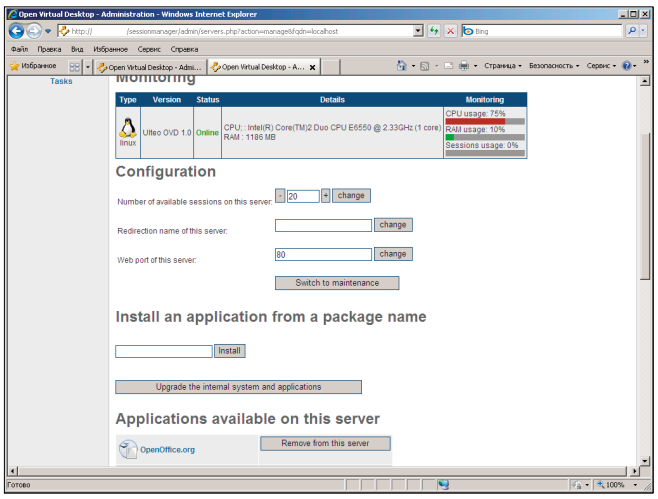
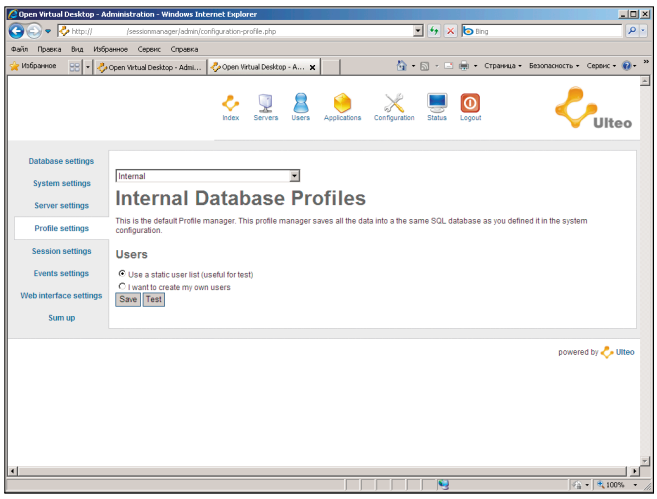


Рисунок 4. Учетные записи пользователей могут храниться либо во встроенной базе, либо в LDAP/AD-хранилище



```
8)          $prefs_ldap = $prefs->get('UserDB', 'ldap');
9)          if ($prefs_ldap['ad'] == 1) {
10)             $buf = $prefs_ldap['suffix'];
11)             $suffix = suffix2domain($buf);
12)             if (! is_string($suffix)) {
13)                 Logger::error('main', 'LDAP suffix ↵
                        is invalid for AD usage : '.$buf);
14)                 break;
15)             }
16)             $windows_login = $user->getAttribute( ↵
                        'login').'@'.$suffix;
17)         }
18)         break;
19)     default:
20)         $windows_login = $user->getAttribute('login');
21)         break;
22) }
```

Данная процедура позволит добавить на виртуальный рабочий стол ярлыки приложений с терминального сервера. Не забывайте только одинаково именовать учетные записи пользователей во внутренней базе Ulteo и в терминальных серверах на Windows-платформе. И соответственно, у них должны быть одинаковые пароли (см. рис. 6).

Что делать, когда нужны другие Linux-приложения, помимо уже установленных внутри Ulteo? С терминальными Windows-серверами понятно – поставил, и они посредством работающего Windows-агента появляются в администраторской панели. С сервером приложений – Application Server – чуть-чуть сложнее. Если вы зайдете по SSH на работающий сервер Ulteo, вы увидите, что это обычный LAMP-сервер. А вот сам Application Server прячется в chroot-окружении в директории /opt/ulteo/ и является обычной Ubuntu-системой. Думаю, что проблем с добавлением источников обновлений пакетов в файл /opt/ulteo/etc/apt/sources.list и дальнейшей работой в chroot-среде не возникнет.

***

Если сравнивать Ulteo Virtual Desktop с такими коммерческими решениями, как Presentation Server (Citrix) и Secure Global Desktop (Sun), то Ulteo будет казаться этаким пигмеем на фоне гигантов. Количество и качество возможностей этих систем превосходят аналогичные от Ulteo. Однако последний даст значительную фору, когда речь заходит о стоимости разворачивания, установки и лицензирования. В целом, используя Ulteo, можно быстрее получить отдачу вложенных в него средств (Return of investment – ROI) по сравнению с любым коммерческим решением и его проще интегрировать с существующими на предприятии системами. Судя по количеству загрузок, это прекрасно понимают в Европе, Азии и в обеих Америках – доля скачиваний из каждого региона примерно треть от общего числа.

Сами разработчики из Ulteo признают, что их продукт используется не только в публичном секторе (в министерствах и ведомствах, администрациях городов, а также в школах и университетах), но также в торговле. И что самое, интересное – некоторые хостинговые компании разворачивают проекты по предоставлению услуг удаленного доступа, посредством веб-топа, для работы в офисных программах. Такие проекты в первую очередь нацелены на малые и средние предприятия. С одной стороны, пользователям, работающим и привыкшим только к Windows-среде, ненавязчиво предлагается целый пакет Linux-приложений. И в то же время Linux-пользователи обеспечены доступом к специфическим Windows-продуктам, альтернатив которым пока нет.

С точки зрения производительности видеоподсистемы следует помнить, что VNC-протокол не самый быстрый и он не ориентирован на просмотр видеофильмов. Работать же в приложениях, нацеленных на офисную работу, также просто и привычно – обновление экрана происходит в незаметном для глаза режиме.

Думаю, что для 80% пользователей будет абсолютно неважно, что же именно скрывается за табличкой Ulteo Virtual Desktop – главное, что все те программы, с которыми они привыкли работать, остались на местах, правда, переместились в окно браузера. А уж сами технические работники, ответственные за поддержание этой виртуальной инфраструктуры, и так знают, сколько они потратили усилий и средств на ее развертывание. EOF

1. http://www.ulteo.com/home/en/ovdi/openvirtualdesktop/downloadnow.
2. http://www.cendio.com/seamlessrdp.
3. http://www.ulteo.com/main/downloads/ulteo-ovd-win.php.

Рисунок 5. Виртуальный рабочий стол GNOME в окне браузера

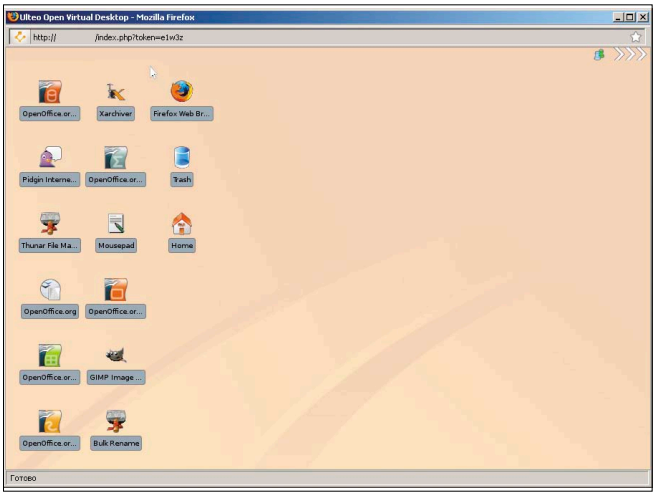
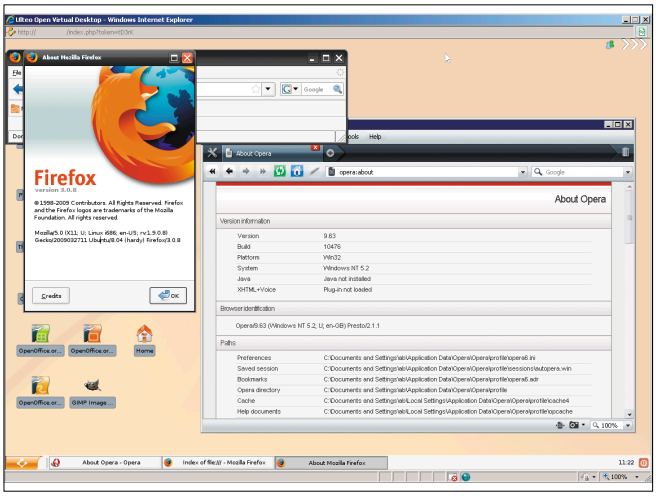


Рисунок 6. С помощью SeamlessRDP добавляется возможность работы и с Windows-программами



Визитка

АНДРЕЙ БИРЮКОВ, специалист по информационной безопасности. Работает в крупном системном интеграторе. Занимается внедрением решений по защите корпоративных ресурсов

Linux Live CD

Создаем загружаемый диск

Это удобный и быстрый инструмент для решения различных задач системного администрирования. Рассмотрим процесс создания собственного Live CD

Какие бывают Live CD?

Загружаемые дистрибутивы Live CD в последние годы стали необходимым в повседневной работе инструментом для системного администратора. Ведь дистрибутив Live CD не требует установки на жесткий диск, умещается на одном компакт-диске, не требователен к ресурсам сервера, но при этом обладает достаточным функционалом, необходимым для решения базовых задач. К тому же всегда удобно, когда под рукой имеются все необходимые программы и утилиты, и для того чтобы воспользоваться ими, достаточно просто загрузиться с компакт-диска, не прибегая к подчас нетривиальной процедуре установки и настройки этих программ.

На сегодняшний день существуют сотни Live CD-дистрибутивов, построенных на основе различных редакций Linux: Debian, Red Hat, Slax и других. Есть также Live CD-версии для BSD-систем, например, Frenzy, операционная система, образ которой умещается на маленьком компакт-диске 210 Мб. Диск с Frenzy я всегда ношу с собой, и он несколько раз выручал меня при решении различных проблем с загрузкой рабочей операционной системы. Но не стоит думать, что мир загружаемых дистрибутивов ограничивается только UNIX-решениями. Существуют Live CD-версии Windows. Например, загружаемая версия Windows XP с набором необходимых приложений официально используется правоохранительными органами Германии для сбора сведений о данных, хранящихся на компьютере подозреваемого. Но операционная система Windows, как и большинство продуктов Microsoft, не является бесплатной, соответственно, для разработки собственного Live CD-дистрибутива необходимо приобретать лицензию. Поэтому в данной статье мы не будем рассматривать создание Live CD-дистрибутивов на основе ОС Windows.

А зачем нам свой Live CD-дистрибутив?

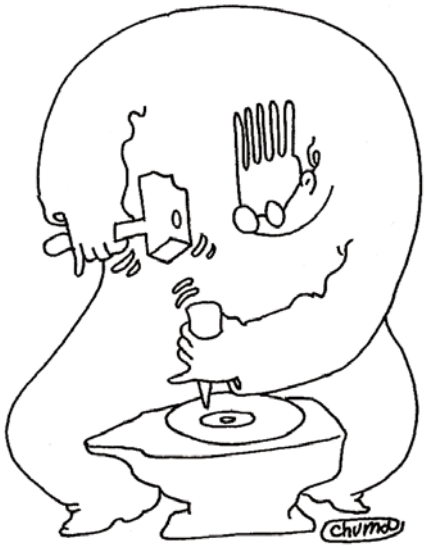
Итак, ранее мы выяснили, что существует множество различных Live CD-дистрибутивов, предназначенных для решения разнообразных задач, таких как использование для системного администрирования, на рабочих станциях, задач информационной безопасности и других. Получить полный список дистрибутивов вы можете по адресу [1]. Конечно, замечательно, когда кто-то уже столкнулся с аналогичной проблемой и уже предоставил на всеобщее обозрение ее решение. Например, создал Live CD-дистрибутив, содержащий установленный межсетевой экран. Но вот проблема, в вашей компании межсетевой экран интегрирован с системой учета трафика. В случае сбоя сервера с межсетевым экраном, даже если вы загрузитесь с соответствующего Live CD и быстро экспортируете необходимые сетевые настройки с флеш-карты, вам все равно для интеграции с системой учета трафика необходимо компилировать соответствующие приложения, что требует дополнительных затрат времени. В противном случае администратор информационной безопасности не будет какое-то время получать информацию о посещаемых сотрудниками веб-ресурсах. А для крупных госучреждений это, как правило, очень плохо. А как было бы удобно, в случае сбоя жесткого диска на сервере, где установлен программный межсетевой экран, просто загрузиться с Live CD, который автоматически продолжит работу в штатном режиме.

Другой пример, вам часто приходится посещать филиалы компании, и вам необходим набор нестандартных приложений, используемых в корпоративной сети. Вместо того чтобы таскать все время с собой ноутбук, можно просто воспользоваться Live CD-диском с набором необходимых программ.

И наконец мой случай. Для решения задач информационной безопасности мне потребовались некоторые утилиты, которые не входили в состав основных Security Live CD, таких как Bug Track, Auditor, и других. Компилировать их каждый раз при загрузке с Live CD было не слишком удобно, поэтому я задумался над сборкой собственного Live CD. Наличие готового набора нужных утилит существенно упрощает проведение аудита и исследование информационной безопасности.

Приступаем к работе

Надеюсь, в предыдущем разделе я достаточно наглядно пояснил, зачем нужно собирать свой собственный дистри-



Сегодня существуют сотни Live CD-дистрибутивов, построенных на основе различных редакций Linux

бутив Live CD. Теперь приступим непосредственно к разработке. Для этого нам потребуется базовый дистрибутив Linux, виртуальная машина, например, VMware, и программа для записи компакт-дисков. В качестве базового дистрибутива я буду использовать Knoppix, так как данная версия Linux наилучшим образом приспособлена для различных модификаций.

Загружаемый дистрибутив Knoppix является одним из наиболее популярных дистрибутивов для создания загружаемых операционных систем. Причин этому несколько. Прежде всего это поддержка по умолчанию большого количества оборудования от различных производителей, благодаря которому вам не нужно начинать работу с системой с поиска и настройки различных драйверов. Еще одной причиной является то, что в Knoppix можно модифицировать практически все – от ядра операционной системы до компонентов рабочего стола. Также данная система имеет удобный интерфейс, позволяющий даже плохо знакомому с операционными системами семейства Linux пользователю без особых сложностей работать в X Window. В состав Knoppix по умолчанию входит множество различных утилит и приложений, таких как OpenOffice, различные медиаплееры, средства работы с сетью и другие программы. В общем, для разработки своего первого загружаемого дистрибутива Knoppix более чем подходит, если в дальнейшем вы захотите создать свой загружаемый дистрибутив на основе другой редакции Linux, то по адресу [2] вы можете найти несколько статей, посвященных работе с другими дистрибутивами.

Для создания дистрибутива вам необходим компьютер со следующими аппаратными характеристиками: от 1 Гб RAM, не менее 3 Гб свободного дискового пространства. Далее я приведу описание процесса создания и модификации собственного дистрибутива на основе Knoppix.

Подготовив машину, приступим непосредственно к модификации дистрибутива:

1. Скачайте последнюю версию дистрибутива с официального сайта knoppix.net. На момент написания этой статьи на сайте была доступна как CD, так и DVD-образ Knoppix, воспользуемся CD-версией.

2. Загрузитесь с компакт-диска Knoppix. Проследите за тем, чтобы операционная система правильно определила используемые аппаратные компоненты. При использовании VMware Workstation с этим проблем не возникает, однако при использовании других систем виртуализации возможны определенные трудности.

3. В открывшемся окне необходимо выбрать значок консоли (Konsole), и введите команду su чтобы стать суперпользователем. По умолчанию в Knoppix учетная запись root использует пустой пароль. Вся дальнейшая процедура создания загружаемого дистрибутива должна производиться под учетной записью root.

4. Обеспечьте доступ к нужному разделу жесткого диска. Для этого необходимо заранее определить, какой из разделов диска будет использоваться для этой процедуры. Как уже упоминалось ранее, на диске требуется от 3 до 5 Гб дискового пространства. Открыть раздел необходимо с правами «чтение-запись». Чтобы это сделать, нужно выбрать на рабочем столе значок, который относится к необходимому разделу, щелкнуть по нему правой кнопкой мыши, демонтировать, а затем снова смонтировать с правами «чтение-запись». В качестве файловой системы будем использовать ext3. Хотя для многих более удобным будет использование следующих команд:

```
# mke2fs -j /media/hda1
# umount /media/hda1
# mount -o rw /media/hda1
```

5. Создайте рабочий каталог, то есть каталог, который будет использоваться для создания нового дистрибутива. Здесь и далее будем предполагать, что рабочим разделом у нас является /hda1. Для создания каталога используйте следующую команду:

```
# mkdir /media/hda1/knoppix
```

6. Теперь необходимо проверить доступную память. Для этого воспользуйтесь командами:

```
# df -h /media/hda1
```

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/root             2.5M   20K  2.4M   1% /
```

```
# free -m
```

```
             total       used       free     shared    buffers     cached
Mem:           503        237        266          0          8        143
-/+ buffers/cache:         85        418
Swap:            0          0          0
```

7. Если у вас менее 1 Гб свободной оперативной памяти, то лучше создать файл подкачки. Для этого выполните следующие команды:

```
# dd if=/dev/zero of=/media/hda1/knoppix/myswap bs=1M ↵
       count=1000
# mkswap /media/hda1/knoppix/myswap
# chmod 0600 /media/hda1/knoppix/myswap
# swapon /media/hda1/knoppix/myswap
```

Результатом выполнения данных команд будет создание, разметка и подключение пустого файла подкачки размером 1000 Мб.

8. Создайте главный и исходный каталоги. Нужно создать два отдельных каталога, в которых будет вестись основная работа. Каталог master/KNOPPIX содержит файлы, необходимые, чтобы запустить компакт-диск, а каталог source/KNOPPIX содержит файлы, которые образуют файловую систему Knoppix, которые впоследствии будут сжаты в единый образ с рабочим названием Knoppix. Чтобы создать эти два рабочих каталога, введите следующее:

```
# mkdir -p /media/hda1/knoppix/master/KNOPPIX
# mkdir -p /media/hda1/knoppix/source/KNOPPIX
```

9. Скопируйте файлы Knoppix. Каталог KNOPPIX на запущенном загружаемом компакт-диске содержит структуру файловой системы Knoppix, которую вам требуется настроить. Выполните следующую команду:

```
# cp -Rp /KNOPPIX/* /media/hda1/knoppix/source/KNOPPIX
```

Учтите, что выполнение данной команды может занять несколько минут, так как производится копирование большого количества файлов. Можно добавить ключ -v, и тогда весь процесс копирования будет отображаться в консольном окне.

10. Теперь скопируйте стартовые страницы, которые отображаются при загрузке Linux. Для этого введите следующее:

```
# cp /cdrom/index.html /media/hda1/knoppix/master
# cp /cdrom/autorun.* /media/hda1/knoppix/master
# cp /cdrom/cdrom.ico /media/hda1/knoppix/master
```

11. Скопируйте файлы с компакт-диска. Для этого нужно скопировать в свой рабочий каталог все остальные файлы, необходимые при запуске компакт-диска. Не нужно копировать только файл образа KNOPPIX размером 600 Мб. Сделать это можно вручную или с помощью следующей команды:

```
# cp /cdrom/KNOPPIX && find . -size -10000k -type f -exec ↵
       cp -p -parents '[]' ↵
       /media/hda1/knoppix/master/KNOPPIX/ \;
```

12. Теперь, когда все необходимые файлы скопированы в рабочий каталог, необходимо изменить свой корневой каталог (выполнить chroot) на каталог source KNOPPIX и изменить структуру этого каталога так, чтобы он содержал приложения, файлы и каталоги, которые вы хотите включить в свой загружаемый компакт-диск. Для этого используйте команду:

```
# chroot /media/hda1/knoppix/source/KNOPPIX
# mount -t proc /proc proc
```

В случае появления ошибки раздела /etc/fstab продолжайте работу, это не критично и никак не влияет на результат. Также можно его просто скопировать.

13. На этом шаге необходимо произвести модификацию приложений, входящих в состав Knoppix. Думаю, что это самый важный шаг, так как именно на нем мы производим модификацию исходного дистрибутива, заменяя в нем ненужные приложения на свои. Ненужных приложений в дистрибутиве довольно много, и удаление каждого из них может занять продолжительное время, поэтому я предлагаю удалить только те из них, которые занимают больше всего места. Для этого прежде всего необходимо узнать, какие пакеты установлены в системе на данный момент. Сделать это в дистрибутиве Knoppix можно с помощью следующих команд:

```
# dpkg-query -W -showformat ='$(Installed-Size) ↵
       $(Package)/n' | sort -nr | less
```

или проще:

```
# dpkg-query -l | sort -nr | less
```

В результате выполнения этой команды, на экран будет выведен список установленных пакетов:

```
…
ii  outguess     0.2-5    Universal Steganographic tool
ii  openvpn      2.0.6-1  Virtual Private Network daemon
ii  openssl      0.9.8a-8 Secure Socket Layer (SSL) binary
and related c
ii  openssh-server        4.3p2-1  Secure shell server,
an rshd replacement
ii  openssh-client        4.3p2-1  Secure shell client,
an rlogin/rsh/rcp replace
ii  openoffice.org-debian-menus    2.0.2-3  OpenOffice.org
desktop integration
ii  openoffice-de-en      2.0.2-1  The OpenOffice suite,
see http://www.openoffic
ii  openhackware 0.4.1-2  OpenFirmware emulator for PowerPC
ii  nxviewer     1.4.0.2-0alpha4   NoMachine NX - nesting
X server with roundtrip
ii  nxtunnel-server       1.4.0-m2-1        server portion
of nxtunnel
ii  nxtunnel-client       1.4.0-m2-1        client portion
of nxtunnel
ii  nxssh        1.4.0-m2-1        NoMachine NX - custom
and modified OpenSSH
ii  nxsetup-knoppix       0.3-1    Simple frontend for
(almost) automatic nxserve
ii  nxproxy      1.4.0.2-0alpha4   NoMachine NX -
X protocol compression proxy
ii  nxlibs       1.4.0.2-0alpha4   NoMachine NX - common
agent libraries
ii  nxdesktop    1.4.0.2-0alpha4   NoMachine NX - nesting X
server with roundtrip
ii  nxclient     1.4.0-91.2        NoMachine NX - NX Client.
ii  nxagent      1.4.0.2-0alpha4   NoMachine NX - nesting
X server with roundtrip
ii  nvtv         0.4.7-3  tool to control TV chips on NVidia
cards under
ii  ntpdate      4.2.0a+stable-8.1 The ntpdate client for
setting system time
…
```

14. Как видно из списка, в составе дистрибутива имеется множество различных пакетов, естественно, многие из них нам не понадобятся. В качестве примера удалим OpenOffice как приложение, занимающее больше всего места. А также приложения, предназначенные для игр (мы же делаем диск для работы, а не для развлечений). Начнем с OpenOffice, для этого сначала нужно увидеть, что содержит данный пакет:

```
# dpkg-query -s openoffice-de-en
```

```
knoppix@2[~]$ dpkg-query -s openoffice-de-en
Package: openoffice-de-en
Status: install ok installed
Priority: optional
Section: unknown
Installed-Size: 353204
Maintainer: Klaus Knopper <knoppix@knopper.net>
Architecture: i386
Version: 2:2.0.2-1
Replaces: openoffice-de-en
Depends: libnspr4
Description: The OpenOffice suite, see
http://www.openoffice.org/
 This release uses a quick&dirty hack to support english
 as well as german layout and templates in KNOPPIX.
```

На экран будет выведена статистика по данному пакету. Из полученного вывода статистики можно узнать как объем и версию установленного приложения, так и его зависимости.

15. Теперь удалим OpenOffice. Для этого необходимо выполнить следующее:

```
# apt-get remove openoffice-de-en
```

В процессе выполнения команды вам будет необходимо подтвердить удаление данного пакета.

И проделаем то же самое для игр на примере kmahjongg:

```
knoppix@2[~]$ dpkg-query -s kmahjongg
```

```
Package: kmahjongg
Status: install ok installed
Priority: optional
Section: games
Installed-Size: 1584
Maintainer: Debian Qt/KDE Maintainers
<debian-qt-kde@lists.debian.org>
Architecture: i386
Source: kdegames
Version: 4:3.5.2-1+b2
Depends: kdelibs4c2a (>= 4:3.5.2-1), libc6 (>= 2.3.6-6),
libgcc1 (>= 1:4.1.0), libkdegames1 (>= 4:3.5.2), libqt3-mt
(>= 3:3.3.6), libstdc++6 (>= 4.1.0)
Description: the classic mahjongg game for KDE project
 Your mission in this game is to remove all tiles from
 the game board. A matching pair of tiles can be removed,
 if they are 'free', which means that no other tiles block
 them on the left or right side.
```

Удалим ненужные пакеты группой:

```
# aptitude purge openoffice-de-en kmahjongg
```

```
Reading package lists... Done
Building dependency tree... Done
The following packages will be REMOVED:
  openoffice-de-en
0 upgraded, 0 newly installed, 1 to remove and 1 not
upgraded.
Need to get 0B of archives.
After unpacking 362MB disk space will be freed.
Do you want to continue [Y/n]? y
(Reading database ... 103699 files and directories
currently installed.)
Removing openoffice-de-en ...
Removing `diversion of /usr/bin/soffice to
/usr/bin/soffice.distrib by openoffice-de-en'
...
```

16. В качестве примера также добавим новый пакет в дистрибутив. Для этого необходимо сначала скопировать нужный пакет на виртуальную машину под управлением Knoppix. Нужный пакет можно скачать напрямую из Интернета или же переписать на виртуальную машину. Для того чтобы переписать с основной машины, необходимо сначала на Knoppix запустить сервер Samba. Сделать это можно следующим образом: выберите раздел Knoppix → Services → Start Samba Server. Затем укажите пароль для пользователя knoppix. Теперь ваша виртуальная машина Knoppix доступна для обмена файлами с основной машиной.

Но вернемся к установке пакетов. Установим текстовый процессор abiword, для этого необходимо указать следующую команду:

```
# apt-get install abiword
```

17. Теперь, когда мы удалили ненужное, необходимо удалить мусор, дабы не занимать лишнее место на нашем загружаемом диске. Просмотрим «бесхозные» пакеты:

```
# deborphan | less
```

18. Выбранные для удаления пакеты можно удалить с помощью следующей команды:

```
# deborphan | xargs apt-get -y remove
```

19. Теперь очистим кэш, в котором также могут находиться удаляемые пакеты:

```
# apt-get clean
```

20. Удалим директории, имеющие временное или динамическое наполнение:

```
# umount /proc
```

21. Когда все необходимое для создания собственного дистрибутива подготовлено, необходимо сделать сжатый образ, полученный в результате всех преобразований файловой системы:

```
# mkisofs -R -U -V "My Knoppix" ↵
        -publisher "Ivan I. Ivanov" ↵
        -hide-rr-moved -cache-inodes -no-bak -pad ↵
        /media/hda1/knoppix/source/KNOPPIX | ↵
        nice -5 /usr/bin/create_compressed_fs - 65536 > ↵
        /media/hda1/knoppix/master/KNOPPIX/KNOPPIX
```

Параметр -R добавляет записи SUSP и RR, которые должны иметь атрибуты файловой системы, рассчитанные на систему Linux (например, длинные имена файлов, символические связи и т.д.).

Параметр -U разрешает имена файлов, которые могут включать в себя символы, не соответствующие стандарту ISO 9960 (начальные точки, многоточия и т.д.).

Параметр hide-rr-moved скрывает каталог RR_Moved в образе, переименовывая его в .rr_moved.

Cache-inodes позволять сберечь пространство на компакт-диске.

Параметр no-bak исключает файлы резервных копий.

Наконец, параметр -pad служит для добавления служебной информации в конец образа.

Следует также отметить, что кроме изменения состава пакетов, в Knoppix можно также модифицировать и другие параметры системы, такие как предустановленный сетевой адрес, имя хоста и другие настройки. Однако в рамках этой статьи мы такие модификации рассматривать не будем.

22. Последний шаг в создании своего дистрибутива – это выполнение команды mkisofs для создания образа всего диска. Добавим параметры в командную строку, чтобы сделать возможной загрузку содержимого компакт-диска:

```
# cd /media/hda1/knoppix/master
# mkisofs -pad -l -r -J -v -V "MyKnop" -no-emul-boot ↵
        -boot-load-size 4 -boot-info-table ↵
        -b boot/isolinux/isolinux.bin ↵
        -c boot/isolinux/boot.cat -hide-rr-moved ↵
        -o /media/hda1/knoppix/knoppix.iso ↵
        /media/hda1/knoppix/master
```

23. В результате выполнения данной команды будет создан файл knoppix.iso в каталоге /media/hda1/knoppix. Данный iso-образ представляет собой файловую систему ISO 9660 CD-ROM, которая разрешает длинные имена файлов (опция -l), расширения и права доступа SUSP, полезные для систем Linux или UNIX (-r) и каталожные записи Joilet (-J). Этот образ готов к проверке с помощью системы виртуализации и записи на компакт-диск.

24. В простейшем случае для записи образа на компакт-диск достаточно воспользоваться командой cdrecord:

```
# cdrecord -v -data knoppix.iso
```

25. Теперь запустите сервер Samba так, как это описано в пункте 16, и скопируйте созданный iso-образ на основную машину.

26. Создайте новую виртуальную машину и загрузите ее с iso-образа.

27. После успешной загрузки системы (что уже является хорошим знаком) убедимся в отсутствии удаленных пакетов. Для этого нажмите Knoppix → Office. Как видите, OpenOffice отсутствует, зато появился abiword. Модификация дистрибутива прошла успешно.

***

Итак, ваш первый загружаемый дистрибутив Knoppix был успешно создан. Теперь вы можете с помощью описанных выше действий вносить изменения в свой дистрибутив и производить его тестирование, однако советую не увлекаться чрезмерным удалением приложений, так как это может привести к неработоспособности всей операционной системы из-за существующих зависимостей некоторых приложений и системных файлов.

Завершая статью, хочу отметить, что помимо приведенной методики модификации Knoppix, существуют также и другие способы, один из которых приведен в [3]. EOF

1. Список существующих дистрибутивов Linux Live CD – http://www.livecdlist.com/?sort=&showonly=.
2. Статьи по модификации дистрибутивов Linux – http://linux-livecd.ru/doc_main.html.
3. Модификация Knoppix – http://stirnimann.com/mystuff/doc/knoppix.txt.

Решите проблемы **лицензирования ПО** с помощью профессионалов!

Операционная система GNU/Linux и свободное программное обеспечение помогут вам с минимальными затратами решить проблему лицензирования программного обеспечения, повысить безопасность и надежность вашей компьютерной сети.

Компания ГНУ/Линуксцентр предлагает вам внедрение ОС GNU/Linux и свободного программного обеспечения, реализацию и техническую поддержку сложных технических решений на базе свободного ПО, обучение ваших сотрудников — как пользователей, так и технических специалистов.

С НАШЕЙ ПОМОЩЬЮ ВЫ СМОЖЕТЕ:

- оптимизировать затраты на лицензирование ПО за счет максимально возможного использования свободного ПО;
- существенно сократить время системных администраторов, затрачиваемое на устранение последствий деятельности вирусов и сбоев в программном обеспечении.

ТИПОВЫЕ ПРОЕКТЫ:

- миграция рабочих станций и серверов с Microsoft Windows на GNU/Linux;
- установка 1С на серверах и рабочих станциях под управлением GNU/Linux;
- миграция с Microsoft Windows Active Directory на Mandriva Directory Server;
- миграция с Microsoft Exchange на Zimbra;
- внедрение интернет-телефонии на базе Asterisk;
- внедрение свободной CRM-системы SugarCRM;
- создание кластеров высокой доступности;
- реализация терминальных решений;
- создание порталов любой сложности на базе свободных CMS-систем — Joomla, Drupal, Plone;
- внедрение защищенных систем на основе сертифицированного ФСТЭК ПО.

СРЕДИ НАШИХ КЛИЕНТОВ:

- Правительство Московской области;
- Правительство Нижегородской области;
- администрация Черниговского района Приморского края;
- Министерство финансов республики Саха (Якутия);
- Владивостокский государственный университет экономики и сервиса;
- группа компаний «ИМАГ»;
- компания «Азбука мебели»;
- компания «Бестли — выставочные материалы»

и другие организации различного профиля.

Наш опыт внедрения свободного программного обеспечения в компаниях различного профиля поможет выбрать оптимальное сочетание свободного и коммерческого программного обеспечения, подходящее именно для вашей организации, а также поможет избежать техничеких и организационных проблем при внедрении свободного ПО.

Linux center www.linuxcenter.ru

Департамент внедрений компании ГНУ/Линуксцентр
Телефон в Москве: (499) 271-49-54,
в Санкт-Петербурге: (812) 309-06-86

ЗВОНИТЕ СЕЙЧАС!

Реклама

Визитка

АГУНДА АЛБОРОВА, журналист. В последние годы пишет на темы ИТ для банков

Вячеслав Калошин:
«Всегда ищи точки приложения своих сил!»

На вопросы «СА» отвечает технический директор компании PingWin Software Вячеслав Калошин

Вячеслав Калошин, окончил Иркутский государственный технический университет, там же работал техником-лаборантом. Работал – оператором ЭВМ в Иркутском интернет-провайдере, редактором сайта в ASPLinux, начальником отдела техподдержки и системным администратором в VoIP-провайдере в Москве, советником директора по ИТ в Ростелекоме, советником директора по развитию бизнеса в компания ISG, советником/заместителем директора в МПК, техническим директором в КЦ национального домена .ру. Ныне – технический директор компании PingWin Software.

Его домашняя дисковая система объемом в четыре терабайта сопоставима с системой небольшого предприятия. В его персональной библиотеке, которая постоянно пополняется, более 40 тыс. книг, а фонотека занимает порядка 200 Гб. У него нет бороды, он не носит бандану, но пиво любит, как всякий нормальный сисадмин. Вячеслав Калошин, ныне технический директор компании PingWinSoftware, считает, что должность сисадмина – прекрасная возможность постоянно узнавать что-то новое.

– Вячеслав, с чего началась ваша карьера? Как вы попали в компанию, в которой работаете сейчас?

– Компьютерами я занимаюсь с 1985 года. Начало работы было комическим. Я ходил в компьютерный класс, когда учился в университете. Но вдруг моя машина оказалась закрытой. Преподаватель сказал, что там какой-то вирус и вообще «все страшно»! После коротких переговоров я машину починил. В итоге меня стали привлекать «починить». Через некоторое время я начал официально работать техником. Увлечение компьютерами проходило параллельно с учебой на машиностроительном факультете по специальности «робототехнические системы и комплексы». Программирование там преподавалось в очень

незначительных объемах, поэтому приходилось заниматься самообразованием. Потом в университете появилась «лаборатория мультимедиа», и я стал работать там…

Следующее место работы – местный интернет-провайдер в Иркутске. Там я выполнял классические сисадминские функции: поднять, настроить, убрать, ответить на запросы по телефону. Там же стал писать свои первые статьи, я называю их «рассказики».

Мой первый опыт программирования связан с калькулятором Б3-34. В инструкции к нему были программы, которые можно было вводить, чтобы он выполнил какие-то функции. Я с большим любопытством их изучил, и калькулятор стал решать простые уравнения сам.

Потом я стал изучать специальную литературу, пробовать программировать на Бэйсике, позже появился Паскаль. Это были времена первых огромных компьютеров с восьмидюймовыми дискетами. В 1995-1996 годах Иркутский государственный технический университет стал первым заведением в городе, где появилась настоящая волоконно-оптическая сеть. А мне довелось тянуть кабели и выполнять весь цикл работ по их прокладке. Через какое-то время я ушел на системное администрирование полностью.

Довольно быстро сеть разрослась, пришлось иметь дело с более 500 машинами. По тем временам это была одна из самых больших сетей.

Тогда же я стал выкладывать свои «рассказики» на сайт. Он стал популярным, меня пригласили в Москву делать linuxnews.ru – это было в 1998-1999 годах…

Будучи редактором сайта, мне приходилось активно общаться, работать на выставках. Но хотелось двигаться дальше. И я, оставаясь редактором, ушел на позицию обычного тестировщика в ASPLinux. Кстати, этот опыт мне многое дал в дальнейшем в вопросах локализации ошибок. Сейчас могу смело утверждать, что локализации каких-то неполадок я могу делать, что называется, с закрытыми глазами.

Сначала я выпускал первые релизы ASPLinux, потом начали привлекать к общению с клиентами, я стал объяснять им преимущества Linux. Он тогда уже очень хорошо использовался там, где что-то было связано с Интернетом. Но квалифицированной поддержки пользователям Linux не было. Поэтому возникла идея создать портал для «линуксоидов» и собственный дистрибутив Linux, поэтому он был назван ASPLinux.

Собственно, эта идея актуальна и по сей день. Для меня наиболее интересной задачей является выстраивание полноценной технической поддержки по Linux, причем не для одного дистрибутива, а для многих. Таких сервисов на рынке у нас пока нет, поэтому это для меня некий вызов. В PingWinSoftware, где я работаю с мая, этим и занимаюсь.

– Какие, на ваш взгляд, вопросы сейчас наиболее актуальны для системных администраторов?

– Как ни странно, но те же самые, что и лет десять назад. Во-первых, это коммуникационные навыки. Суметь объяснить пользователю на пальцах, что надо делать, а что не надо. Суметь доказать начальнику, что необходимо сделать. Рассказать кому-нибудь другому, насколько ты хорош собой.

Главный вопрос, который мучает новичка-сисадмина: «Откуда берутся эти тупые пользователи?!»

Во-вторых – получение новых знаний и умений. Умеете рулить почтой – учитесь управлять базами данных. Умеете управлять базами – учитесь строить кластеры, тренироваться с виртуализацией. Возможностей для самосовершенствования много.

Когда я работал в ASPLinux, мне приходилось отвечать на многочисленные телефонные звонки пользователей. Как-то один из них предложил мне перейти на работу в провайдер IP-телефонии. Было интересно, и я пошел. Через какое-то время нам удалось добавить к существующему новое «железо», сконфигурировать и состыковать все с телефонной станцией, аву-

Блиц-опрос

– Кем вы хотели стать в детстве?

– Официальная версия, что в детстве я хотел быть космонавтом. Сейчас с высоты того опыта, который у меня есть, я могу сказать, что мне, в общем-то, практически все равно, в какой области работать. Большинство систем одинаково устроены, различия лишь в наборе параметров и тонкостях управления ими.

– Как вы относитесь к знакомствам через Интернет?

– Хорошо. Свою жену так нашел!

– Вы читаете книги в электронном виде?

– Да, предпочитаю электронные или аудиокниги. Я столкнулся с тем, что зачастую обычные книги стали издаваться очень низкого качества и меня это стало раздражать. В последнее время я читаю либо легкую фантастику, либо профессиональную или бизнес-литературу. Недавно перечитал множество книг о фондовом рынке.

– Какие предпочтения в музыке?

– Мне нравится разная музыка. В моей музыкальной библиотеке есть все – от тяжелого рока до классики. По объему – это 3,5 месяца непрерывного звучания без пауз.

– С чем бы вы сравнили компьютер?

– Вы можете смеяться, но, с моей точки зрения, это живое существо! Если сисадмин слабее машины, она начинает капризничать, глупые поломки устраивать, дразниться. А были случаи, когда админ подстраивал ее под себя, тогда компьютер либо начинал работать исправно и надежно, либо вообще ломался. Такие вот мистические штучки встречаются…

– Ваша фирменная рекомендация?

– Ask Google!

Про UNIX и Windows

«Войны не будет. Дискуссии о том, что лучше, всегда были и будут. Но суть в том, что они существуют параллельно. Их архитектуры все больше становятся похожи друг на друга. Кардинальные отличия лежат на большой глубине, на уровне программистов ядерного уровня. А на практике они и взаимодействуют, и стыкуются. Это гетерогенные системы. Готов поручиться прядью своих волос, что в любой большой организации сейчас есть и UNIX, и Windows. И так будет всегда потому, что некоторые специфические задачи хорошо решает только UNIX».

томатизировать многие функции, и наконец все заработало.

Но много времени занимала техническая поддержка пользователей. В то время я написал статью, которая имела большой резонанс в сети, под названием «Каким мне видится хороший системный администратор».

– А, кстати, каким он должен быть?

– Суть можно выразить так. Первое – сисадмин всегда должен собирать статистику. Второе – приходить к руководству подготовленным, с цифрами, с графиками. И третье – пользователь, с которым он работает, должен чувствовать себя, как цыпленок под крылом у курицы...

– С какой проблемой чаще всего сталкиваются сисадмины?

– Проблема довольно простая и общая. В условиях кризиса практически все предприятия экономят средства. В первую очередь режутся бюджеты на информационное обеспечение. Руководство считает каждую копейку, поэтому на любого человека, который приносит внятную идею о сокращении расходов, оно смотрит благосклонно.

Где и за счет чего можно сэкономить в ИТ-департаменте? Если говорить о компаниях, в которых рабочих мест более 300-500, то тут вполне возможен частичный переход или замещение больших программ их Open Source-аналогами. Мы, например, пару раз использовали такой переход и остались довольны. Комплект Microsoft Office в большинстве случаев спокойно заменяется на бесплатный аналог Open Office. Экономия средства идет за счет оплаты необходимых лицензий.

Стоимость миграции с проприетарного софта на Open Source составляет примерно 75-80% от годовой оплаты проприетарных лицензий.

Поэтому мой ответ на вопрос: «Куда сейчас смотреть сисадминам?» – будет такой: посмотрите, что из существующего софта можно заменить на Open Source. Конечно, это потребует времени для того, чтобы опробовать, как будут работать подобные изменения. Но результат может превзойти все ожидания. И экономия оказалась довольно существенной.

Этот подход можно рекомендовать любым компаниям – от маленьких до крупных. Дело в том, что Open Source может быть очень полезен в случаях, когда речь идет об использовании тиражируемого решения.

Основная ошибка многих компаний состоит в их решении полностью отказаться от лицензионного софта. Этого делать не надо. Сейчас компания Microsoft все чаще предлагает аренду лицензии, то есть программное обеспечение не приобретается, а арендуется. Для большинства компаний это выгодно и удобно. Поэтому оптимально сделать так: ту часть, которая нуждается в лицензионном проприетарном софте, не трогать, но все другие места, где возможно обойтись без него, перевести на Open Source. Уверяю вас, экономия средств на сезонных выплатах будет существенная.

Есть еще одна выгода от подобного перехода. Open Source – это лицензионные продукты, соответственно решается вопрос и с проверяющими органами.

Если сисадмин решает осуществить подобный переход, ему прежде всего надо провести полную ревизию всего имеющегося софта. И наверняка он найдет участки, на которых вполне могут работать Open Source–решения.

– А где решения Open Source абсолютно неприемлемы?

– В компаниях, где весь компьютерный парк заточен под определенную специализацию. Например, типография или конструкторское бюро с определенной специализацией. А если в компании стандартная бухгалтерия, использующая «1С», пользователи работают с обычными программами, один сервер, который обеспечивает выход в Интернет и т.д., – использование Open Source в ней совершенно оправданно и может дать реальную экономию. Обычно компьютер бухгалтера не трогают, остальные компьютеры можно спокойно перевести под Open Source.

– Какой ваш фирменный совет сисадминам?

– Надо искать то, что интересно. У меня получается так: я достигаю чего-то в одной области, потом мне приходится наступать на горло собственной песне и идти младшим помощником старшего повара в другой области. Зато появляются новые навыки. Учиться надо все время.

Для получения новых знаний есть только одно безотказное правило – ask Google! (Смеется.)

Если хочешь быть как все – будь там, где все. Если тебе нужно что-то уникальное – вероятность того, что об этом кто-то напишет на форуме, чрезвычайно мала. Лично я на форумах ответы на свои вопросы не ищу, меня там нет. В основном там решаются довольно простые практические вопросы, которые задают новички. Причем обычно, как правило, им выдается всего лишь man (справочные команды UNIX), по которым можно что-то узнать.

– Чем, как вам кажется, психология сисадмина отличается от психологии хакера?

– По большому счету ничем. Я проверяю свою сеть типичными хакерскими приемами. Любой сисадмин без здорового любопытства, без вопросов, а как же все это работает и как это все можно поломать, разобрать, а потом собрать, долго не проживет. Различия же в том, что задача хакера – разломать и попользоваться в своих целях, а у сисадмина – разломать, починить и закрыть, чтобы другим непо-

Про Windows Vista

«У каждой компании определенный цикл: хороший продукт – плохой продукт. Так и здесь. Когда появилась Windows XP – много было недовольных, но потом ее доделали. А Windows Vista – просто очередной этап, который надо пережить. Уже вышла Windows 7 – по всеобщему мнению, это та Vista, которая должна быть, со всеми необходимыми доработками. Такой цикл есть практически в любом софте – продукт проходит определенные стадии, дорабатывается и совершенствуется. И относиться к этому надо спокойно».

вадно было ломать. Методология одна и та же, а мотивации разные.

– Вы интуитивно чувствуете угрозы? Стало ли их больше в последнее время?

– Я себя отношу к категории профессиональных параноиков. Вопрос «а что, если?..» всегда вертится в моей голове, когда работаю. И должен сказать, что в определенные моменты это помогало – 95% угроз благодаря такому подходу закрывалось. А те немногие проколы, которые случались, были по причине того, что лень было что-то лишний раз перепроверить. И, кстати, могу сказать, что 90% взломов происходит только потому, что сисадмин поленился что-то проверить или перестраховаться.

С угрозами, как мне кажется, ситуация такая: если раньше давили интеллектом, то сейчас – количеством. Если раньше заражали пять систем, но хитро, вызывая проблемы по обнаружению и обезвреживанию, то сейчас тупо заражают 100 тыс. систем. С одной стороны, это страшнее, но с другой – если пользователи и провайдеры настроены защищаться правильно, такое явление гораздо проще гасится. Качественный уровень угроз сейчас невысокий.

– В чем главная психологическая сложность профессии сисадмина?

– Для начинающего сисадмина главный вопрос, который его мучает: «Откуда берутся эти тупые пользователи?» Главное – перебороть это отторжение и не подхватить звездную болезнь. Нужно научиться понимать точку зрения другого человека. И всегда помнить о том, что даже если ты хорошо разбираешься в компьютерах, ты можешь ничего не понимать в той же бухгалтерии, и от того, насколько ты сможешь выслушать и понять проблему пользователя, зависит качество твоей работы и решение этой проблемы.

– Кстати, почему так мало среди сисадминов женщин? Может, это не их работа?

– Хотя женщины-сисадмины и редко встречаются, но не про одну из тех, кого знаю, я не могу сказать, что она не профессионал.

Кстати

Компания «ПингВин Софтвер», в которой сейчас работает Вячеслав Калошин, входит в состав группы компаний «АйТи», в которой помимо системного интегратора находятся также «Академия АйТи», «Аплана», «БОСС. Кадровые системы», «БОСС-Референт», «Мобико», «ПингВин Софтвер». В сентябре «АйТи» завершила создание инфраструктуры центра обработки данных для размещения в Уфе мультисервисного узла связи Mobile WiMAX сети Yota.

Из женщин получается совершенно идеальные сисадмины! Во-первых, они пунктуальны и методичны в работе. Если женщина становится начальником ИТ-департамента, у нее в отделе безупречный порядок, на любой вопрос она знает или ответ, или где можно его найти. Либо все работает, либо ничего не работает по какой-то серьезной причине. Если классический сисадмин сталкивается с подобной руководительницей, он не всегда понимает, как к такой женщине относиться. Когда узнают, что один из крутых специалистов по вирусам – женщина, у многих просто сносит крышу напрочь! EOF

Визитка

ОЛЕГ КУЗЬМИН, создавал систему информбезопасности Северного флота. Развивал в ВС РФ защиту от утечек информации. С 2006 г. – директор департамента информбезопасности ЗАО «Ай-Теко»

Практический инсайд

Основная проблема и причина низкой эффективности борьбы с утечками и действиями инсайдеров в целом — слишком большая теоретизация

Утечка информации: суха теория, мой друг…

В настоящее время стремительными темпами развивается новый сектор рынка информационной безопасности – борьба с утечками информации. Технологии предотвращения утечек информации позиционируются чаще всего под аббревиатурой DLP, хотя сам термин, обозначающий решения по защите конфиденциальных данных от внутренних угроз, еще и не устоялся и трактуется по-разному: и как Data/Information Leak Prevention, и как Data Loss Protection.

Большинство ведущих производителей программных и аппаратных средств ИБ уже предложили линейки своей продукции класса DLP-систем для защиты наиболее чувствительной, важной информации, принадлежащей компании, от ее же собственных сотрудников – инсайдеров. Казалось бы, внедри предложенное на выбор решение, ужесточи организационные меры, и вот оно, счастье, – утечки в компании наконец-то прекратятся! Однако не все так просто, как хотелось бы и может показаться на первый взгляд. Несмотря на принимаемые руководством компаний жесткие меры, утечки данных по вине инсайдеров все же случаются, причем довольно часто.

Основная проблема и причина низкой эффективности борьбы с утечками и действиями инсайдеров в целом – слишком большая теоретизация. Теория в вопросах DLP сейчас шагает семимильными шагами, при этом, к сожалению, зачастую расходясь с нынешними реалиями. В чем это выражается? Сегодня все решения по борьбе с инсайдерами строятся прежде всего на предположениях, что сотрудник:

- целенаправленно охотится за конфиденциальной или любой другой чувствительной информацией и занимается ее несанкционированным сбором;
- случайно или преднамеренно пытается выполнить с конфиденциальной информацией какие-либо действия, направленные на ее утечку;
- с использованием и/или превышением имеющихся у него прав пытается извлечь материальную выгоду лично для себя;
- намеревается предпринять противоправные действия, чтобы кому-то в компании отомстить, насолить, создать проблемы.

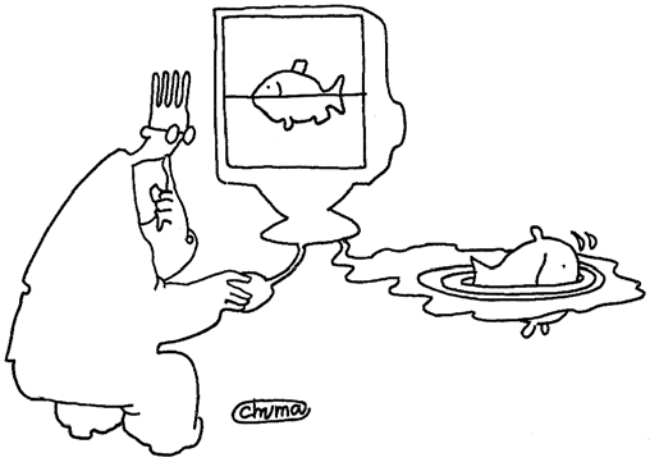
Такова теория, а что в это время происходит в реальной жизни?

Герой не нашего романа

А с практикой понятно далеко не все. Дело в том, что обычно мы можем оперировать только обработанной аналитиками обобщенной статистикой, основанной на данных анкетирования сотрудников компании, занимающихся ИТ или в лучшем случае еще и сотрудников службы безопасности. Большинство из опрошенных, вероятно, даже не смогут самостоятельно точно определить, что считать утечкой, а что является нормальным рабочим процессом в современной организации, и уж тем более – их общее число в компании. Ведь более-менее уверенно они могут отвечать лишь в рамках своих должностных обязанностей, сферы компетенции и зоны ответственности.

Кроме того, в ходе опроса выясняется, что более 90% из них оказываются попросту незнакомы с перечнем конфиденциальной информации компании, не знают, где и в каком виде она хранится, что с ней в принципе можно сделать. В итоге мы получаем результаты опросов, основанные больше на предположениях, нежели на фактах. Вот и фиксируются у нас лишь последствия громких утечек (в основном благодаря СМИ), приведшие к финансовым потерям, да еще многочисленные случайные факты утечек персональных данных. Соответственно все вышеперечисленные теоретические выкладки на практике оказываются не применимы в полной мере к инсайдерским угрозам как в силу неопределенности самой информации, отнесенной в организации к конфиденциальной, так и всех связанных с ней действий.

В результате нагнетания напряженности вокруг данной проблемы одновременно с ростом озабоченности руководства состоянием ИБ на предприятии в целом наблюдается и рост потерь от зафиксированных СМИ утечек, что отражается в диаграммах ежегодных аналитических отчетов.



Друг тайно доверил копию двум другим проверенным приятелям, у которых были свои надежные друзья

Очевидно, что при таком подходе утечки конфиденциальной информации будут происходить и дальше, а количество связанных с ними громких дел год от года с переменным успехом расти. Внедренные DLP-решения, вероятно, смогут, подобно неводу, выуживать из сетей предприятия различных нарушителей, по неопытности или незнанию слишком вольно обращающихся с доверенной им или случайно попавшей к ним информацией, при этом в ряде случаев не замечая действий настоящих инсайдеров, способных в один прекрасный день полностью разорить свою компанию. Именно такие инсайдеры, а не оступившиеся сотрудники, по неосторожности или в результате случайного стечения обстоятельств ставшие причиной мелких локальных инцидентов, будут объектом рассмотрения в нашей статье, посвященной практическому инсайду и его реальному использованию.

У матросов нет вопросов? Зато есть ответы!

В далеких теперь уже 90-х годах XX века на одном из флотов Вооруженных Сил РФ для борьбы с утечками конфиденциальной и служебной информации на флоте была предложена и опробована на практике система, позволяющая в режиме реального времени останавливать подобные утечки, обнаруживая и пресекая их еще на этапе возникновения предпосылок. Тогда оценивались возможные события, вероятность сочетания их между собой и дальнейшее перерастание уже в реальные угрозы. В качестве основной уязвимости рассматривался прежде всего сам сотрудник, целенаправленные или случайные действия которого могли бы привести к риску утечки информации.

Например, в штабе флота разрабатывался некий серьезный документ для крайне ограниченного круга пользователей, который мог бы быть весьма интересен определенной категории должностных лиц – скажем, командирам кораблей, соединений, объединений флота – для успешной сдачи зачетов на допуск к самостоятельному управлению. В современной терминологии можно это обстоятельство считать событием, по мере окончательной готовности документа переросшим в угрозу, поскольку данный документ был гипотетически востребован достаточно большим кругом лиц. Ознакомление с данным документом, в котором агрегировалась информация и обобщались сведения из большого числа закрытых источников, освобождало таких «лишних читателей» от необходимости самостоятельного поиска и изучения более трех десятков других, не менее важных и нужных для успешной сдачи зачета, документов.

Представьте себе, что вам, по основному своему образованию биологу или филологу, срочно нужно сдать экзамен по сложному предмету, включающему обширные разделы знаний из области математики, физики, химии и биологии, вместе взятых. Согласитесь, это трудноосуществимо и требует существенных затрат, в том числе временных. Вдруг происходит чудо, и у вас в руках оказывается документ, содержащий все экзаменационные вопросы с ответами на них. И беспокоиться больше не о чем – ведь лично вам успешная сдача экзамена в этом случае практически гарантирована!

Сотрудника, разработавшего сей документ, и узкий круг официально к нему допущенных офицеров можно считать потенциальной уязвимостью. Технические уязвимости при этом также принимаются во внимание. Данное сочетание угрозы и уязвимости в то время и приводило к утечке информации и дальнейшему ее неконтролируемому размножению на случайных носителях.

Как именно это происходило? Сотрудник, разработавший документ, под «большим секретом» сделал только одну несанкционированную копию для старого друга, которому всецело доверял. Друг в свою очередь также тайно доверил копию двум другим, безусловно, проверенным приятелям, у которых были свои, конечно же, очень надежные и верные друзья. Итог был предсказуем и весьма показателен: через несколько дней после разработки документа его несанкционированная копия была обнаружена (в ходе рядовой проверки состояния информационной безопасности) на жестком диске компьютера в организации флота, крайне далекой от штаба. Проведенное расследование показало, что после ухода первой несанк-

ционированной копии, о которой знал сам разработчик, остальные – и весьма многочисленные – несанкционированно растиражированные копии попали на личные компьютеры и оказались доступны лицам, не только не сдающим никакие зачеты, но и даже вообще потерявшим связь с флотом…

Кто виноват и что делать?

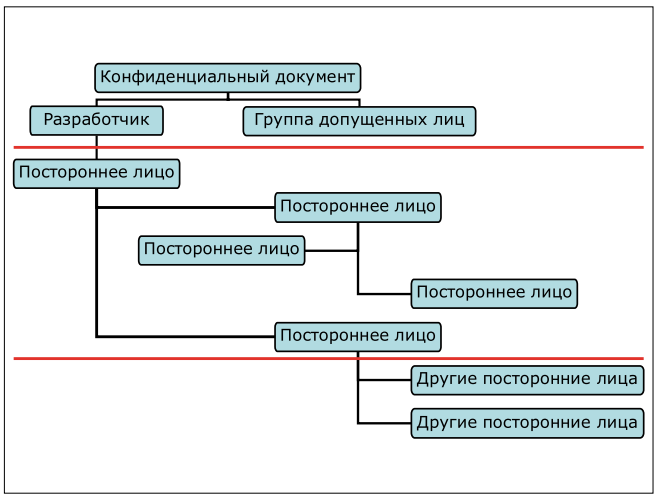
На приведенной схеме (см. рис. 1) над первой красной чертой отмечена допустимая политикой безопасности организация работы с конфиденциальным документом. Блок между первой и второй красной чертой определяет несанкционированное и бесконтрольное распространение документа в пределах организации. Все, что ниже второй красной черты, означает окончательную потерю информации, содержащейся в документе, и возможности каким-либо образом отследить ее дальнейшее использование.

На этом и аналогичных фактах и строились в дальнейшем основы организационной составляющей системы информационной безопасности, позволявшей свести риски утечек важной информации к минимуму.

Как это может быть применено сегодня, надеюсь, большинство читателей уже и сами догадались. При внедрении DLP-решения в компании необходимо оценивать реальные возможности гипотетических действий сотрудников с информацией, а также учитывать сочетания возможностей двух и более сотрудников компании либо при наличии сговора, либо возможного случайного стечения обстоятельств. Сам объект защиты – конфиденциальная информация – должен быть досконально изучен, четко определен и выражен в конкретных документах или данных. При этом различного рода размытости, неточности формулировок и двусмысленности в их определении должны быть полностью исключены. Места хранения конфиденциальной информации и документов должны четко фиксироваться, контролироваться и анализироваться на предмет возникновения аномалий, включая возрастающую активность допущенных к ним сотрудников.

Действия инсайдера должны не только перехватываться в момент совершения утечки важной информации, как это, например, происходит сейчас: они должны быть заранее спрогнозированы на случай как самой возможности, так и мотивации сотрудника, по вине которого конфиденциальная информация может уйти на сторону. Это очень тонкая и последовательная многоэтапная работа, подчас не доступная ни сотрудникам службы безопасности компании, ни тем более ИТ-службам.

Рисунок 1. Схема распространения конфиденциальной информации в организации



Генеральная уборка

Вот еще характерный пример, раскрывающий одну из форм практического инсайда, на этот раз позитивную. Внедряется большой комплекс средств автоматизации (БКСА) АСУ «Океан». Как грибы после дождя автоматизированные рабочие места (АРМ) пользователей появляются в управлениях и отделах, где немедленно забиваются разнообразной закрытой информацией. Организация обучения фактически отдана на откуп самим пользователям, которым даны инструкции и руководство по работе с сетью и конкретным АРМ. В БКСА применяется система защиты информации, сертифицированная по 3-му классу защиты от НСД, и обслуживает ее ни много ни мало 12 сотрудников службы безопасности.

Начальник средств автоматизации уверяет, что закрыто всё, что только возможно, и первая проведенная очередь государственных испытаний это подтверждает: все условия по ИБ, оговоренные в техническом задании, действительно выполнены. Начальник средств автоматизации торжествует, принимает поздравления разработчиков комплекса и «признания в любви» от руководства отделов и управлений, до которых очередь установки АРМ еще не дошла. По понятным причинам не дошла она и до ответственного за ИБ флота, в результате чего возможность проверки системы защиты вне рамок госиспытаний отсутствовала.

Однако вскоре офицеру, ответственному за ИБ, случайно становится известно, что на нескольких технологических АРМах по окончании испытаний разработчиками БКСА неофициально были оставлены дисководы. После этого офицер флота по ИБ, изучив схему функционирования БКСА и распорядок работы управлений флота, решился на практический инсайд, заручившись предварительно соответствующим разрешением своего руководителя.

В роли инсайдера выступил сам офицер, выбрав в качестве модели возможного поведения самую низкоуровневую фигуру в штабе флота – уборщика производственных помещений. В назначенный час, по согласованию с начальником технологического подразделения, с ведром и шваброй он прибыл в одно из технологических помещений, попросив всех присутствующих сотрудников удалиться на время уборки, по регламенту на 10-15 минут. Обрадованные возможностью покинуть на время душное помещение, сотрудники поспешно удалились на перекур, оставив шесть включенных в сеть АРМов. При этом ни один из них не закрыл паролем доступ к своему компьютеру! На одном из АРМов обнаружился рабочий дисковод, в который «уборщиком» и была вставлена дискета (заранее надлежащим образом зарегистрированная). Далее последовал быстрый серфинг по сети, обнаружение в ней АРМов, принадлежащих наиболее привлекательным – с точки зрения конфиденциальности обрабатываемой информации – управлениям флота. Быстрый просмотр расшаренных папок (по версии

разработчика, это были хранилища почтовых ящиков) дал поразительную картину: сотни важных документов оказались доступны для немедленного их копирования, что и было сделано «уборщиком». Спустя 15 минут прибывшие с перекура сотрудники застали его вытирающим влажный пол и уже готовым удалиться.

В ожидании реакции службы безопасности БКСА на несанкционированные действия с технологического АРМа была взята 24-часовая пауза. И поскольку реакции не последовало, скопированные документы были частично распечатаны и предъявлены начальнику штаба флота с предложением поручить начальнику средств автоматизации разобраться, каким образом они могли «уйти» из защищенного БКСА. Последовал ряд «боевых» действий, а закончилась история тем, что при попытке разработчиков убрать открытые почтовые ящики весь комплекс стал буквально разрушаться, и в итоге БКСА АСУ «Океан» так и не был принят флотом, оставшись существовать в разобранном на разрозненные локальные сети виде.

Так, используя методы практического инсайда, удалось доказать несостоятельность системы защиты в деле сохранения важной конфиденциальной информации. Надо понимать, что дисковод в данной ситуации был инструментом для сбора доказательств и являлся дополнительной уязвимостью, а воспользоваться основной уязвимостью мог любой легальный пользователь с любого АРМа, или же такие действия мог выполнить любой посторонний человек, случайно оказавшийся возле оставленного без присмотра АРМ. О том, как бы он мог распорядиться полученной информацией, лучше даже и не думать.

За примером далеко ходить… надо

Перенесемся за океан. Лето 2008 года, США. Терри Чайлдс, администратор городской сети г. Сан-Франциско FiberWAN, 13 июля получил контроль над сетью FiberWAN. Узнав о планах начальства по его увольнению, Чайлдс создал учетную запись, которая наделяла его исключительными правами доступа. Заблокировав доступ к FiberWAN всем остальным пользователям, Чайлдс в итоге стал единственным администратором городской сети.

Сеть FiberWAN обошлась Сан-Франциско в несколько миллионов долларов. Она содержит сотни тысяч конфиденциальных документов, включая платежные ведомости и переписку служащих. Сотрудники муниципалитета еще не до конца оценили нанесенный Чайлдсом ущерб. В полиции опасаются, что сисадмин мог оставить лазейку для третьих лиц, чтобы те могли уничтожить сотни тысяч важных документов.

Если сотрудник – инсайдер, допущенный или случайно получивший доступ к конфиденциальной информации, ограничен в возможностях ее несанкционированного копирования, изменения, уничтожения или отправки, это вряд ли его остановит. Он в конце концов найдет новое нестандартное решение, направленное на одно или сразу несколько перечисленных выше действий с конфиденциальной информацией. Или даже просто выучит ее и восстановит по памяти на компьютере у себя дома. Но даже на этот случай должны быть разработаны превентивные меры и меры, сводящие ущерб от подобных действий к его допустимому минимуму.

Практический инсайд – это не только тематическая область для постоянного изучения и выработки мер противодействия и предупреждения. Это еще и хороший инструмент в умелых руках для убеждения собственного руководства в недостаточной защищенности жизненно важных для компании информационных ресурсов.

А как дела обстоят в Европе? Январь 2008 года, Франция. Банк Societe Generale прогремел на весь мир в связи с потерей почти 5 млрд евро из-за спланированных действий его сотрудника Жерома Кервьеля.

Этот сотрудник не являлся топ-менеджером, не принадлежал к руководящему персоналу банка. Не обладая особыми коммерческими и финансовыми талантами, к своим 30 годам Жером смог подняться с позиции скромного клерка отдела контроля до брокера, став торговцем. Трейдер Кервьель не придумал ничего уникального и не изобрел новый гениальный способ наживы. Благодаря точному знанию правил и условий контроля и мониторинга в банке он смог успешно обойти все установленные заслоны, похитить коды доступа операторов банка и умело их использовать для осуществления операций с финансовыми транзакциями в компьютерной системе банка. Хотя в этом банке наверняка были выполнены все требования акта Sarbanes Oxley (SOX) и других обязательных актов в вопросах управления операционными, инвестиционными и кредитными рисками. Да и вопросы информационной безопасности в этом банке были решены не самым худшим образом. Однако последствия практического инсайда не заставили себя долго ждать. Помимо сразу выявленных прямых убытков в 4,9 млрд евро, банк был оштрафован еще на 4 млн евро, а его акции рухнули на 3,64%. Причиной столь значительных потерь стали незаконные операции, которые производил на бирже трейдер Кервьель, открыв позиции на фондовой бирже на сумму, в несколько раз превосходящую собственный капитал Societe Generale. Если в январе 2008 года рейтинг банка был «AA», то уже 18 февраля 2008 года рейтинговое агентство Standard & Poor's понизило долгосрочный кредитный рейтинг банка до «AA-» и оставило негативный прогноз. Некоторое время спустя, узнав о своих дополнительных потерях, банку пришлось пойти на продажу части своих акций с 40% скидкой. Столь высокой оказалась цена действий лишь одного инсайдера.

Следствие по этому делу зашло в тупик, и отпущенный на свободу после полуторамесячного заключения Ж. Кервьель сразу получил выгодное предложение о работе от компании, занимающейся информационной безопасностью. Главный вывод расследования, проведенного банком с привлечением сторонних экспертов, был сформулирован так: «Банк недооценивал риски, связанные с возможностью фиктивных сделок и свободным перераспределением лимитов на операции между трейдерами». По утверждению авторов доклада, именно благодаря этому Кервьель смог открыть фиктивные позиции на большие суммы и даже создать внутри банка собственную организацию, которая позволяла ему получать доход за счет использования ресурсов Societe Generale. Этот пример приведен в качестве иллюстрации наиболее острой формы практического инсайда – злонамеренного.

К сожалению, печальный опыт не всегда оказывается полезным и учитывается в дальнейшем. Иначе чем объяснить

тот факт, что спустя всего несколько месяцев после нашумевшей истории с Ж. Кервьелем кассир Румынского банка развития, принадлежащего Societe Generale, Михаэла Росана Биту была арестована за попытку перевести на свой счет 3,2 млн долларов. Для доступа к счетам госпожа кассир воспользовалась паролями высокопоставленных сотрудников этой кредитной организации. По сообщению источника в Румынском банке развития, банку удалось своевременно заблокировать перевод большей части и вернуть около 2,8 млн долларов. Вместе с мошенницей арестованы еще три человека, всего же в число подозреваемых попало семь лиц.

Стоит ли упоминать, что и в России аналогичных примеров предостаточно...

Существует и еще одна, третья форма практического инсайда – скрытый острый негатив. Эта форма инсайда предполагает, что инсайдер подготовился и выполнил заранее все необходимые действия, в результате чего не только получил доступ к интересующей его информации, но и совершил ее хищение (в виде копий). Убедившись, что его действия не раскрыты, инсайдер может пересмотреть конечную их цель с точки зрения получения им существенно больших, чем ожидалось ранее, дивидендов. Например, к ним может быть отнесена продажа инсайдером не одной, а сразу нескольких копий конфиденциальной информации с предварительным шантажом руководства компании и получением выкупа или же сдвиг активных действий с информацией по времени до наступления более выгодных для инсайдера условий. И такие случаи сейчас далеко не редкость, особенно в небольших и средних по размеру компаниях. Остановка действий инсайдера на этом этапе еще может помочь сократить размер убытков и последствий для компании в целом.

Инсайт vs инсайд

Таким образом, можно выделить три вида практического инсайда (см. рис. 2).

Позитивный – это наиболее гуманный вид инсайда, носящий предупредительный характер и не приводящий в большинстве случаев к серьезным реальным потерям в организации.

Злонамеренный инсайд – тот, который выполняется с целью получения немедленной выгоды в результате тщательно продуманных действий злоумышленника.

Остро негативный инсайд – наиболее изощренный в плане нанесения компании максимально возможных убытков, вплоть до ее полного разорения или поглощения. Этот вид инсайда может быть растянут во времени до неопределенных пределов, конечная цель инсайдера может делиться на этапы, представленные разными целями.

Практический инсайд сегодня – это реальный инструмент, который может одинаково эффективно сработать как во благо компании в единичных случаях, так и – в абсолютном их большинстве – во вред организации. Насколько хорошо и, главное, кем он может быть применим – основной вопрос для руководства любой компании, которой есть что защищать от посторонних в целях безопасности своего бизнеса и самого существования организации. Целенаправленная деятельность по построению полноценной системы защиты от практического инсайда включает в себя хорошо организованную работу с важной конфиденциальной и другой чувствительной информацией и допущенными к ней в компании людьми, а также предполагает разработку эффективных организационных мер и использование различных технических средств для осуществления действенного контроля за работой этой системы. EOF

Рисунок 2. Виды и последствия практического инсайда

Множественные уязвимости в ATL в IBM Java

Программа: IBM Java 5.0 SR10 и более ранние версии.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибки в обработчике регулярных выражений при сравнении общих имен в сертификатах. Удаленный пользователь может с помощью специально сформированного сертификата, подписанного CA, или сертификата, принятого пользователем, вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.

2. Уязвимость существует из-за ошибки при обработке определенных полей в сертификате. Злоумышленник может обманом заставить пользователя принять специально сформированный сертификат.
URL производителя: www.ibm.com.
Решение: В настоящее время способов устранения уязвимости не существует.

Выполнение произвольного кода в Kaspersky Online Scanner

Программа: Kaspersky Online Scanner 7.0, возможно, другие версии.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за неизвестной ошибки, которая позволяет удаленному пользователю скомпрометировать целевую систему. Подробности уязвимости не разглашаются.
URL производителя: www.kaspersky.com/virusscanner.
Решение: В настоящее время способов устранения уязвимости не существует.

Множественные уязвимости в Wireshark

Программа: Wireshark 1.2.1 и более ранние версии.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибки в диссекторе OpcUa. Удаленный пользователь может с помощью специально сформированного Service CallRequest-пакета потребить все доступные ресурсы процессора на системе. Уязвимость распространяется на версии 0.99.6 по 1.0.8, и 1.2.0 по 1.2.1.

2. Уязвимость существует из-за ошибки в диссекторе GSM A RR. Удаленный пользователь может аварийно завершить работу приложения.

3. Уязвимость существует из-за ошибки в TLS-диссекторе. Удаленный пользователь может с помощью специально сформированного TLS 1.2-сетевого пакета аварийно завершить работу приложения на некоторых системах (например, Windows).
URL производителя: www.wireshark.org.
Решение: Установите последнюю версию 1.0.9 или 1.2.2 с сайта производителя.

Выполнение произвольного кода в реализации TCP/IP в Microsoft Windows

Программа: Microsoft Windows Vista; Microsoft Windows 2008.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за недостаточной очистки информационного статуса TCP-пакетов в TCP/IP-стеке. Удаленный пользователь может с помощью специально сформированного TCP/IP-пакета заставить TCP/IP-стек ссылаться на поле в качестве указателя функции и выполнить произвольный код на целевой системе.
URL производителя: www.microsoft.com.
Решение: Установите исправление с сайта производителя.

Выполнение произвольного кода в nginx

Программа: nginx версии до 0.7.62, 0.6.39 и 0.5.38.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за ошибки при обработке ссылок в функции ngx_http_parse_complex_uri(). Удаленный пользователь может с помощью специально сформированного запроса вызвать потерю значимости буфера и выполнить произвольный код на целевой системе.
URL производителя: nginx.net.
Решение: Установите последнюю версию 0.7.62, 0.6.39 или 0.5.38 с сайта производителя.

Множественные уязвимости в VMware Workstation Movie Decoder

Программа: VMware Workstation Movie Decoder версии до 6.5.3 build 185404; VMware Workstation версии до 6.5.3 build 185404; VMware Player версии до 2.5.3 build 185404.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибки проверки границ данных в кодеке VMnc (vmnc.dll версии 6.5.2.7026). Удаленный пользователь может с помощью специально сформированного видеофайла с некорректным разрешением вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.

2. Уязвимость существует из-за ошибки в VMnc-кодеке при обработке видеоданных с высотой ниже 8 пикселей. Удаленный пользователь может с помощью специально сформированного видеофайла вызвать повреждение динамической памяти и выполнить произвольный код на целевой системе.
URL производителя: www.vmware.com.
Решение: Установите последнюю версию с сайта производителя.

Составил Александр Антипов

Визитка

СЕРГЕЙ ЯРЕМЧУК, инженер автоматизации. Автор более 800 статей и 4 книг. В «Системном администраторе» публикуется с первого номера. Интересы: сетевые технологии, защита информации, свободные ОС

OSSEC-HIDS
Установка и настройка

Эта хостовая система обнаружения атак позволяет защитить компьютеры, работающие под управлением разных ОС

Сегодня в UNIX используется большое количество систем защиты (logcheck, tripware, tiger, chkrootkit, rkhunter), использующих разные принципы – контроль целостности файлов, анализ журналов, поиск сигнатур и так далее. В отдельности каждый из этих продуктов имеет вполне достаточные возможности, позволяющие защитить доверенную систему, которые после настройки могут работать автономно, лишь оповещая администратора о появившихся проблемах. Но при развертывании их на большом количестве систем в сети администратору приходится искать оптимальный метод развертывания и, главное, получения и анализа отчетов.

OSSEC-HIDS [1] (Open Source Host-based Intrusion Detection System) – хостовая система обнаружения атак, в которой для определения атак используется несколько методов – анализ журналов систем и сервисов, проверка целостности файлов и реестра Windows, обнаружение rootkit, имеющая функциональность SIM (Security Information Management, Управление информацией безопасности). Собственно OSSEC и появился как решение, адаптированное на проверку большого количества систем.

Первая версия, называвшаяся Syscheck, была написана Даниелем Сидом (Daniel B.Cid) и обнародована в 2004 году. Через некоторое время к Syscheck добавился Rootcheck, который позволял управлять обнаружением руткитов с централизованной консоли. И, наконец, в 2005 году вышел OSSEC, объединивший функции Syscheck и Rootcheck.

Сегодня проект поддерживают большое количество сторонних разработчиков, которые вносят изменения в код, создают правила, участвуют в переводе интерфейса. Но основная работа выполняется Даниелем. В июне 2008 года права были переданы корпорации Third Brigade, Inc., которая обязалась продолжить разработку и коммерческую поддержку. В мае 2009 права перешли к Trend Micro, которая также обязалась сохранить его бесплатным и продолжать разработку по Open Source-лицензии. В настоящее время OSSEC-HIDS распространяется по лицензии GNU GPL v3.

OSSEC состоит из трех компонентов:

- базового приложения, которое устанавливается в UNIX-системах, обеспечивает основную функциональность, то есть сервер для сбора и анализа данных, и агента (для UNIX-систем);
- агента Windows;
- веб-интерфейса, позволяющего просмотреть оповещения в удобном виде.

Для сбора информации на удаленных системах используются агенты, хотя возможна работа и без агента (например, на некоторых маршрутизаторах). В настоящее время агенты реализованы для Linux, Solaris, *BSD, Mac и Windows 2000/XP/2003/Vista/2008. Для связи агентов с сервером используется 1514 UDP-порт, подключение производится по зашифрованному SSH-каналу. Веб-интерфейс устанавливается на Linux, Solaris, *BSD и Mac. Количество поддерживаемых OSSEC-форматов журналов достаточно велико – Asterisk, Apache, IIS, Postfix, Courier, ProFTPd, Pure-FTPd, PostgreSQL, MySQL Checkpoint, Sonicwall, VMware ESX и многие другие. Полученные данные обрабатываются на сервере на основании правил в формате XML, и в случае обнаружения попыток модификации важных файлов (анализируются MD5 и SHA1, права доступа) и реестра, и прочих потенциально опасных действий OSSEC выдает оповещение, а при высоком уровне реагирует на попытку взлома.

Типичное правило (в текущей версии их 896) заключено в теги rule и состоит из описания (description), контролируемого параметра и его значения (url, same_source_ip, regex, match), группы (group), действия (action), уровня серьезности (level) и имеет свой уникальный номер (id).

Например, описание одного из правил из файла web_rules.xml.

```
<rule id="31105" level="6">
<if_sid>31100</if_sid>
<url>%3Cscript|%2Fscript|script>|script%3E| ↵
      SRC=javascript|IMG%20|</url>
<url>%20ONLOAD=|INPUT%20|iframe%20</url>
<description>XSS (Cross Site Scripting) ↵
      attempt.</description>
<group>attack,</group>
</rule>
```

При помощи <if_sid> одни правила могут использовать условия из других правил и на их основе устанавливать свои действия. Например, несколько записей с id 31105 с разных IP-адресов свидетельствуют о распределенной XSS-атаке.

В правилах возможно использование регулярных выражений, поэтому при определенных навыках можно описать практически любую ситуацию. Администратор может легко указать на события, которые его интересуют, и получать оповещения через сервисы SMTP, SMS и Syslog, а также настроить ответную реакцию, заблокировав удаленный узел через /etc/host.deny или при помощи пакетного фильтра – iptables (Linux), ipfilter (FreeBSD, NetBSD, Solaris). При активации по умолчанию производится блокировка всех удаленных источников с уровнем опасности начиная от 6. Предусмотрено выполнение и произвольных команд пользователя, хотя, чтобы реализовать такую возможность, требуется уже некоторый опыт.

Чтобы уменьшить вероятность случайной блокировки своих ресурсов в случае ошибки OSSEC, разработчики предлагают готовые сценарии, используются белые списки и тайм-аут, после которого ресурс будет разблокирован. Вывод может быть направлен в Prelude IDS [2] (сейчас она позиционируется тоже как SIM), оповещения по умолчанию сохраняются в текстовых файлах, но для этой цели можно подключить базу данных MySQL.

Установка OSSEC

Одной из приоритетных задач при разработке OSSEC являлась и является простота в установке. Действительно, многие отзывы сводятся к одному – OSSEC очень прост не только в инсталляции, но и в работе. В настоящее время ведутся работы над универсальным скриптом, который позволит еще более облегчить задачу развертывания системы контроля на компьютерах, работающих под управлением разных версий ОС.

Актуальной на момент написания статьи была версия 2.1.1, о которой мы и будем в дальнейшем говорить. Выход следующей версии OSSEC 2.2 ожидается в ближайшее время, правда, точная дата пока неизвестна. В статье будет описана установка основного приложения и веб-интерфейса на Ubuntu Server 8.04.1 LTS, а также агента, работающего в Windows XP. В других системах процесс практически аналогичен.

На странице закачки доступны три вида пакетов, которые реализуют компоненты, описанные выше. Учитывая, что OSSEC является узловой системой, агент должен быть установлен в идеальном случае на каждом компьютере в сети. Основной пакет (ossec-hids) реализует как серверную часть (server), так и клиентскую часть (agent) для UNIX, нужный тип установки необходимо будет выбрать уже в процессе. Если OSSEC будет использоваться на единственной UNIX-системе в сети, можно использовать локальную (local) установку.

Для установки OSSIM понадобится компилятор.

```
$ sudo apt-get install build-essentials
```

Среди файлов на странице закачки имеется ossec-hids-latest.tar.gz, но он указывал на версию 2.1, хотя уже доступна 2.1.1. Скачиваем файл, проверяем контрольную сумму, распаковываем.

```
$ wget -c http://www.ossec.net/files/ossec-hids-2.1.1.tar.gz
$ wget -c http://www.ossec.net/files/ ↵
      ossec-hids-2.1.1_checksum.txt
$ cat ossec-hids-2.1.1_checksum.txt
$ md5sum ossec-hids-2.1.1.tar.gz
$ sha1sum ossec-hids-2.1.1.tar.gz
$ tar xzvf ossec-hids-2.1.1.tar.gz
```

Запускаем установочный скрипт, находящийся внутри архива.

```
$ cd ossec-hids-2.1.1/
$ sudo ./install.sh
```

Скрипт предлагает выбрать язык установки, в списке есть и русский. Вводим ru. Далее выводятся данные по системе, проверяем и нажимаем <Enter>. Выбираем тип установки, введя по подсказке – сервер, агент или локальный (на русском), указываем, куда установить OSSEC (по умолчаниию /var/ossec).

Третьим пунктом идут настройки OSSEC, в которых подтверждается:

- активация уведомления по электронной почте – вводим почтовый адрес дважды, указываем SMTP-сервер;
- запуск сервиса проверки целостности;
- запуск сервиса обнаружения руткитов;
- активация системы активного реагирования – разрешаем использование пакетного фильтра (блокировка с level >=6), указываем «белые» адреса;
- сохранение журналов на удаленном syslog.

Установочный скрипт автоматически заносит в белый список локальную систему и DNS-сервера. Рекомендую сразу же указать и адрес системы, с которой производится удаленное управление сервером, так как после запуска OSSIM запросто может заблокировать доступ.

```
$ sudo cat /var/ossec/logs/active-responses.log
```

```
…
Птн Авг 28 04:24:00 MSD 2009 /var/ossec/active-response/
bin/host-deny.sh add - 192.168.1.58 1251419040.1029 40101
Птн Авг 28 04:24:00 MSD 2009 /var/ossec/active-response/
bin/firewall-drop.sh add - 192.168.1.58 1251419040.1029 40101
```

Далее скрипт автоматически заносит в список контролируемых журналы установленных сервисов. В дальнейшем файл журнала можно легко добавить, отредактировав па-

раметр localfile в конфигурационном файле /var/ossec/etc/ossec.conf, который и создается впоследствии установочным скриптом на основе полученных ответов. Далее происходит компиляция компонентов OSSEC.

В процессе будет создан init-скрипт /etc/init.d/ossec, который можно использовать для запуска сервиса. Как вариант указать скрипт напрямую:

```
/var/ossec/bin/ossec-control start|stop
```

О смене владельца говорит надпись при загрузке.

```
Starting OSSEC HIDS v2.1 (by Trend Micro Inc.)...
```

Обновление происходит аналогичным образом, скрипт обнаруживает установленный OSSEC и предлагает обновить установку.

Файл ossec.conf представляет обычный XML-файл, разобраться в его структуре довольно просто. Например, каталоги, в которых проверяется целостность файлов, описываются следующим образом:

```
<syscheck>
 <!-- проверка каждые 22 часа -->
     <frequency>79200</frequency>
 <!-- каталог для проверки -->
<directories check_all="yes">/etc,/usr/bin, ↵
       /usr/sbin</directories>
<!-- игнорируемые файлы -->
<ignore>/etc/hosts.deny</ignore>
</syscheck>
```

Уровни оповещения:

```
 <alerts>
     // Запись в журнал
     <log_alert_level>1</log_alert_level>
     // Отправка почтового сообщения
     <email_alert_level>7</email_alert_level>
 </alerts>
```

Настройка ответа:

```
 <active-response>
 <!--использование host.deny для блокировки IP
     на 60 секунд при level >= 6
 -->
     <command>host-deny</command>
     <location>local</location>
     <level>6</level>
     <timeout>600</timeout>
  </active-response>
```

Сервер уже контролируется, просмотреть оповещения можно в журнале:

```
$ sudo tail -f /var/ossec/logs/ossec.log
```

Добавление агентов

Каждый агент для получения доступа к серверу должен указать свой уникальный ключ. Для создания такого ключа или удаления агента на сервере используется менеджер агентов.

```
$ sudo /var/ossec/bin/manage_agents
```

```
****************************************
* OSSEC HIDS v2.1 Agent manager. *
* The following options are available: *
****************************************
  (A)dd an agent (A).
  (E)xtract key for an agent (E).
  (L)ist already added agents (L).
  (R)emove an agent (R).
  (Q)uit.
Choose your action: A,E,L,R or Q:
```

Далее нажимаем клавишу, соответствующую нашей задаче:

A – добавить агента;

E – получить ключ для агента;

L – просмотреть список известных агентов;

R – удалить агента.

Нажимаем «A», вводим по запросу имя агента (от 2 до 32 знаков), IP-адрес системы, на которой будет установлен агент, затем ID агента (скрипт предложит автоматически следующий номер, по порядку начиная от 001). Подтверждаем установки. Теперь выбираем «E», менеджер выведет список агентов, указываем ID агента, ключ которого необходимо получить.

```
Choose your action: A,E,L,R or Q: E

Available agents:
```

Рисунок 1. Работа скрипта установки OSSIM

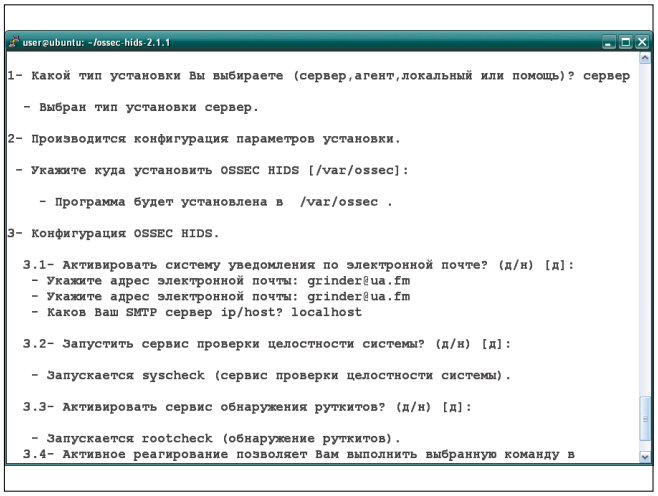
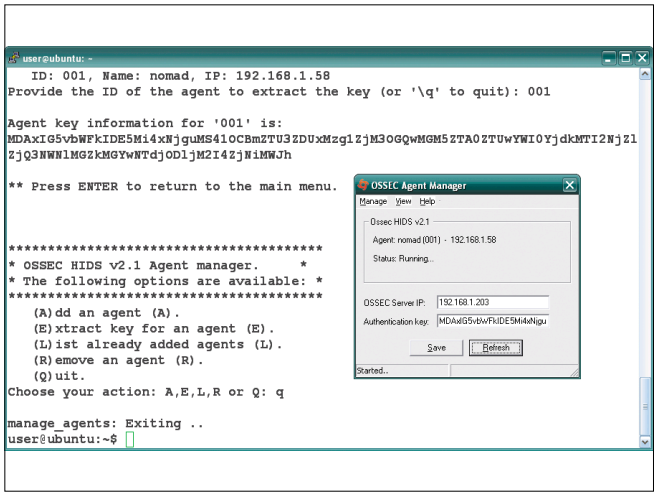


Рисунок 2. Подключение Windows-агента и интерфейс OSSEC Agent Manager



```
ID: 001, Name: agent1, IP: 192.168.1.58
Provide the ID of the agent to extract the key
(or '\q' to quit): 001

Agent key information for '001' is:
MDAxIG5vb……..2I4ZjNiMWJh
```

Запускаем установку ossec-hids-2.1.1, как показано выше (для UNIX-систем), выбрав вариант «агент», по окончанию установки также запускаем manage_agents, в котором выбирается импорт ключа сервера – «I».

При установке агента для Windows, дополнительно можно активировать сканирование журналов IIS и проверку целостности. На последнем этапе, в окне OSSEC Agent Manager вводим IP-адрес сервера и ключ доступа. После подключения к серверу в журнале агента должна появиться соответствующая запись.

```
2009/08/29 09:13:59 ossec-agent:
INFO: Trying to connect to server (192.168.1.203:1514).
2009/08/29 09:14:09 ossec-agent(4102):
INFO: Connected to the server (192.168.1.203:1514).
```

В дальнейшем управление работы агентов производится для UNIX-систем стандартным способом, через init скрипт, в Windows при помощи графического OSSEC Agent Manager. Конфигурационный файл агента также называется ossec.conf и по своей структуре практически полностью аналогичен ossec.conf сервера, о котором говорилось выше.

В версии 2.1 появилась возможность удаленного управления агентами. Для этого после подключения агента на сервере создается единый конфигурационный файл /var/ossec/etc/shared/agent.conf, в котором описываются настройки для конкретного агента:

```
<agent_config name=" agent1|agent2">
```

или в зависимости от ОС.

```
<agent_config os="Linux">
```

Затем при помощи команды agent_control с указанием ID агента файл отправляется на удаленную систему.

```
$ sudo /var/ossec/bin/agent_control -i 001
```

В подкаталоге contrib архива с исходными текстами находится ряд скриптов, позволяющих упростить работу с OSSIM или расширить его возможности.

Например, ossec-batch-manager.pl предназначен для упрощенного подключения агентов. В строке запуска можно указать сразу несколько систем:

```
$ sudo ./ossec-batch-manager.pl -a --ip 192.168.1.1 ↵
       -n agent_1 -a --ip 192.168.1.2 -n agent_1
```

Затем получаем ключ и подключаем агентов.

Все сообщения о тревогах будут собраны в подкаталоге /var/ossec/logs/alerts в отдельном файле.

Например, для 28 августа 2009 года это будет /var/ossec/logs/alerts/2009/Aug/ossec-alerts-28.log.

```
$ cat /var/ossec/logs/alerts/2009/Aug/ossec-alerts-28.log

** Alert 1251464430.42143: - syslog,sudo
2009 Aug 28 17:00:30 ubuntu->/var/log/auth.log
Rule: 5402 (level 3) -> 'Successful sudo to ROOT executed'
```

Расширенная установка

Максимальное число агентов, поддерживаемое сервером по умолчанию, равно 256. О чем свидетельствует запись в журнале после его старта:

```
INFO: Maximum number of agents allowed: '256'.
```

Изменить это значение можно, использовав перед запуском установочного скрипта параметр setmaxagents.

```
$ cd src; make setmaxagents

Specify maximum number of agents: 4096
Maximum number of agents set to 4096

$ cd ..; sudo ./install.sh
```

Теперь максимальное количество агентов будет равно 4096. Но этот предел ограничен еще и настройками ОС, в частности, количеством одновременно открытых файлов, по умолчанию в Linux это число равно 1024.

```
$ ulimit -a

open files          (-n) 1024
```

При добавлении 1025 агента получим:

```
** ID '1025' already present. They must be unique.
```

Изменим число:

```
$ sudo ulimit -n 4096
```

или

```
$ sudo sysctl -w kern.maxfiles=4096
```

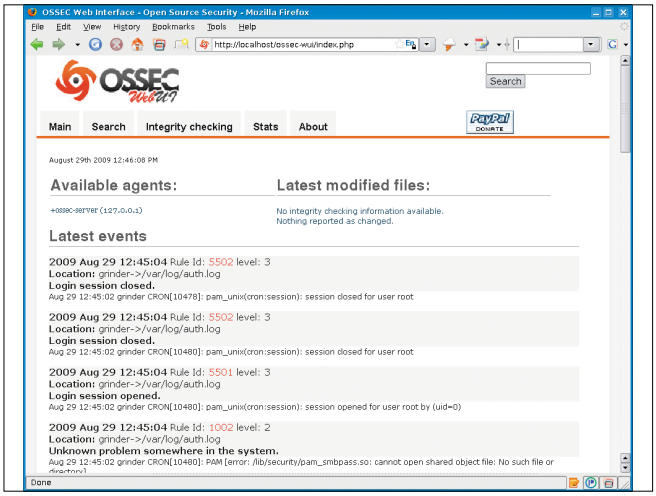
После чего следует перезапустить OSSIM.

Присутствие компилятора на сервере нежелательно, поэтому разработчики предусмотрели возможность бинарной установки. Для этого вначале на другой системе следует собрать OSSIM (результат будет сохранен в подкаталоге bin), затем перенести исходные тексты на сервер. Теперь в файле etc/preloaded-vars.conf снимаем комментарий со строки:

```
USER_BINARYINSTALL="x"
```

И запускаем установку обычным образом.

Рисунок 3. Веб-интерфейс OSSEC-WUI



```
Src IP: (none)
User: user
Aug 28 17:00:28 ubuntu sudo: user : TTY=pts/1 ;
PWD=/home/user/; USER=root ; COMMAND=/bin/ls
/var/ossec/logs/alerts/2009/Aug/ossec-alerts-28.log
```

Скрипт ossectop.pl позволяет увидеть список самых активных тревог за текущий день. Единственный минус – для определения текущей даты используется код:

```
my $datepath=`date "+%Y/%b/ossec-alerts-%d.log"`;
my $LOG='/var/ossec/logs/alerts/'. $datepath;
```

При включенной русской локали скрипт будет искать файл совсем в другом месте: /var/ossec/logs/alerts/2009/Авг/. Исправить можно разными способами. Например, так:

```
$ sudo env LC_ALL=POSIX ./ossectop.pl
```

Скрипт ossec_report_contrib.pl позволяет получать форматированные отчеты.

```
$ sudo cat ossec-alerts-28.log | ↵
       ./ossec_report_contrib.pl -s | ↵
       mail root -s 'OSSEC summary report'
```

Или через cron:

```
00 01 * * * (cat ossec-alerts-28.log | ↵
       ./ossec_report_contrib.pl -s)
```

И, наконец, используя набор файлов ossec2mysql.*, можно подключить вывод OSSEC в MySQL. Процесс достаточно прост. Создаем новую базу данных и затем таблицы при помощи шаблона ossec2mysql.sql, копируем contrib/ossec2mysql.conf в /etc/ossec2mysql.conf с 0600 правами. Редактируем файл, указав параметры доступа к базе данных.

Установка веб-интерфейса

Веб-интерфейс написан на PHP. Для его работы разработчики рекомендуют Apache с поддержкой PHP (>= 4.1 или >= 5.0) или Lighttpd с FastCGI (php4-cgi, php5-cgi). При их наличии процесс установки достаточно прост.

```
$ wget -c http://www.ossec.net/files/ui/ossec-wui-0.3.tar.gz
$ tar xzvf ossec-wui-0.3.tar.gz
$ sudo mv ossec-wui-0.3 /var/www/ossec-wui
```

Далее можно создать нужные настройки вручную, воспользовавшись инструкциями на сайте, но проще запустить установочный скрипт.

```
$ /var/www/ossec-wui
$ sudo chmod 770 /var/www/ossec-wui/tmp
$ sudo chgrp www-data /var/www/ossec-wui/tmp
$ sudo ./setup.sh
```

Скрипт запросит имя пользователя и пароль для доступа к веб-интерфейсу (создается при помощи htdigest), в результате его работы получим файл .htaccess. Делаем учетную запись, от имени которой работает веб-сервер, членом группы ossim.

```
$ sudo vi /etc/group
```

```
ossec:x:1001:www-data
```

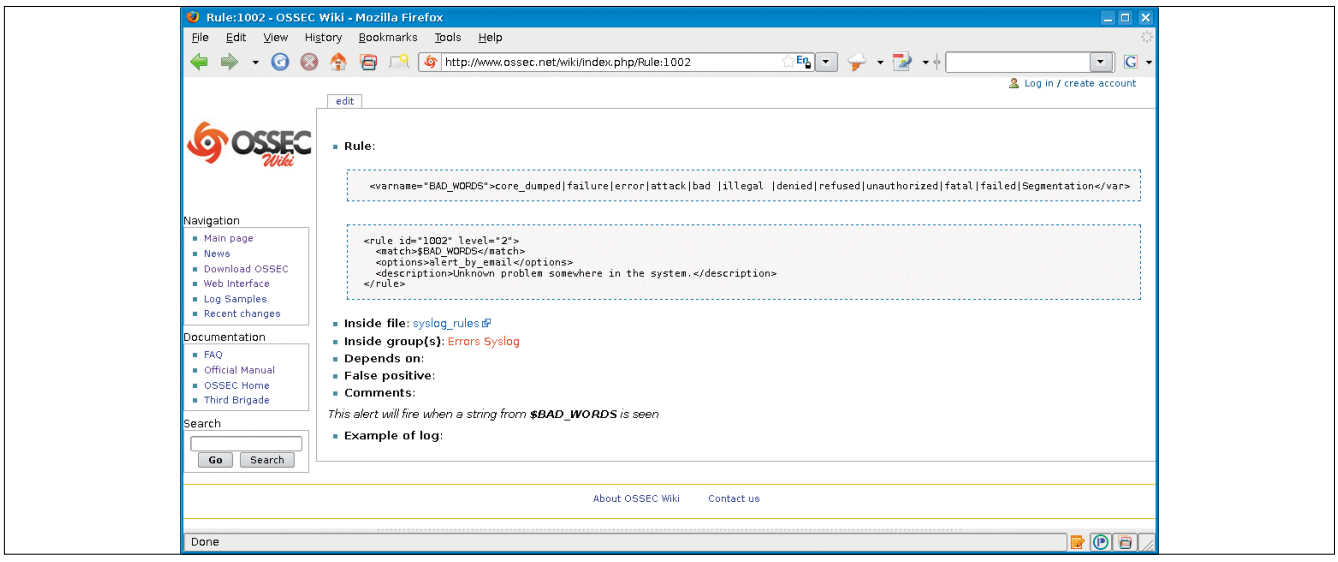
Если используется Apache, это все. Для Lighttpd настроек больше, все они расписаны на Wiki проекта. Теперь можно заходить по адресу http://server/ossec-wui, где будут доступны последние события, список измененных файлов, данные об агентах. Реализован поиск, для каждого события можно просмотреть правило, которое активировано событием.

***

OSSEC позволяет контролировать системы в сети, при этом достаточно прост в установке и управлении. При определенных навыках можно самостоятельно сконфигурировать правила и изменить настройки для любых условий. Для дополнительной информации обращайтесь к документации проекта, также проект имеет свою группу Google [3]. EOF

1. Сайт проекта OSSEC-HIDS – http://www.ossec.net.
2. Сайт проекта Prelude IDS – http://www.prelude-ids.com.
3. Группа OSSEC в Google – http://groups.google.com/group/ossec-list.

Рисунок 4. Просмотр правила на Wiki проекта



Множественные уязвимости в PHP

Программа: PHP версии до 5.2.11.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за неизвестной ошибки при проверке подлинности сертификатов в php_openssl_apply_verification_policy.

2. Уязвимость существует из-за неизвестной ошибки, относящейся к индексу цветов в функции imagecolortransparent().

3. Уязвимость существует из-за неизвестной ошибки при обработке exif-данных.
URL производителя: www.php.net.
Решение: Установите последнюю версию 5.2.11 с сайта производителя.

Уязвимость при обработке SMB-пакетов в Windows Vista и Windows 2008

Программа: Microsoft Windows Vista, Microsoft Windows 2008.
Опасность: Средняя.
Наличие эксплоита: Да.
Описание: Уязвимость существует из-за ошибки индексирования массива в драйвере ядра srv2.sys. Удаленный пользователь может с помощью специально сформированного пакета SMB v2 вызвать повреждение памяти и выполнить произвольный код на целевой системе.

Уязвимость не распространяется на Windows Server 2008 R2 для x64-битных платформ и Windows Server 2008 R2 для платформ Itanium.
URL производителя: www.microsoft.com.
Решение: В настоящее время способов устранения уязвимости не существует. В качестве временного решения производитель рекомендует отключить протокол SMB v2. С сайта производителя можно скачать утилиту, которая отключает протокол: go.microsoft.com/?linkid=9683379.

Выполнение произвольного кода в службе Wireless LAN AutoConfig в Microsoft Windows

Программа: Microsoft Windows Vista, Microsoft Windows 2008.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за ошибки в службе Wireless LAN AutoConfig Service (wlansvc) при обработке определенных фреймов, полученных по беспроводной сети. Удаленный пользователь может с помощью специально сформированных фреймов вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.
URL производителя: www.microsoft.com.
Решение: Установите исправление с сайта производителя.

Множественные уязвимости в Cisco Unified Communications Manager

Программа: Cisco Unified Communications Manager 5.x.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибки при обработке SIP INVITE-сообщений. Удаленный пользователь может с помощью специально сформированного SIP INVITE-сообщения аварийно завершить работу важных процессов.

2. Уязвимость существует из-за ошибки при обработке сетевых подключений. Удаленный пользователь может создать большое количество TCP-подключений к уязвимой системе и тем самым предотвратить возможность создания новых подключений.

3. Уязвимость существует из-за ошибок при обработке SIP- и SCCP-пакетов. Удаленный пользователь может отправить системе большое количество TCP-пакетов и закрыть SIP-парты (5060/TCP и 5061/TCP) и SCCP-порты (2000/TCP и 2443/TCP).
URL производителя: www.cisco.com.
Решение: В настоящее время способов устранения уязвимости не существует.

Неавторизованный доступ к зашифрованным данным в устройствах QNAP

Программа: QNAP TS-239 Pro версия прошивки 3.1.1 0815, 3.1.0 0627 и 2.1.7 0613, возможно, другие версии; QNAP TS-639 Pro версия прошивки 3.1.1 0815, 3.1.0 0627 и 2.1.7 0613, возможно, другие версии.
Опасность: Низкая.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за небезопасного хранения резервного ключа в процессе шифрования данных на жестком диске. Локальный пользователь может дешифровать диск с использованием ключа, сохраненного в flash-памяти устройства.
URL производителя: www.qnap.com.
Решение: В настоящее время способов устранения уязвимости не существует.

Уязвимость форматной строки в httpdx

Программа: httpdx версии до 1.4.1.
Опасность: Высокая.
Наличие эксплоита: Да.
Описание: Уязвимость существует из-за ошибки форматной строки в функции h_readrequest() в файле httpdx_src/http.c. Удаленный пользователь может с помощью специально сформированного HTTP-заголовка «Host:» выполнить произвольный код на целевой системе.
URL производителя: httpdx.sourceforge.net.
Решение: Установите последнюю версию 1.4.1 с сайта производителя.

Составил Александр Антипов

ПАВЕЛ ПЛОТНИКОВ, социальный педагог. Имеет награды за продвижение современных электронных технологий в обучающий процесс. С 2008 года работает в компании «Доктор Веб», занимаясь дизайном взаимодействия ПО

Dr.Web для Mac OS X
Как рождался интерфейс

Работая над интерфейсом для продукта, который предполагается использовать на платформе, принципиально отличной от Windows, мы столкнулись с целым рядом интересных задач

И дело здесь не только в специфике «маковских» интерфейсов – при работе над «лицом» Dr.Web для Mac OS X удалось реализовать и несколько новых идей. Я расскажу вам о том, как появился пользовательский интерфейс первого антивируса для защиты «маков», созданного в России.

К началу разработки интерфейса рабочий прототип антивируса был уже практически написан. Концепция на тот момент была такова: продукты для различных платформ должны быть схожи и объединяться под одним названием – «КонтролЦентр». Что и отразилось на дизайне прототипа (см. рис. 1).

Поэтому первое, с чем мы столкнулись при работе с прототипом, – его внешний вид имел мало общего со стилем Mac OS X.

Иконки оживают

Для начала мы решили немного оживить иконки, благо сама Mac OS X располагает к красоте и жизнерадостности, а формат и размеры «маковских» иконок позволяют развернуться «на полную» (размеры могут достигать огромных размеров – 512x512 точек).

Образ компьютера мы решили привести к образу Mac. Однако пока мы его рисовали, Apple успела выпустить новый iMac с глянцевыми экранами, так что иконку пришлось переделывать.

Изображая угрозы, мы сразу отвергли пошловатый прием использования значков радиационной и биологической опасности. Хотелось чего-нибудь «эдакого». И «эдакое» родилось – в виде молнии.

Нелегко дались и графики, тем более что до последнего не было ясно, будут ли реализованы графики результатов сканирования в самой программе (см. рис. 2).

Одно потянуло за собой другое: раз мы изменяем иконку в приложении – надо изменить ее и для панели Dock. Соответственно нарисовали кнопку агента и изменили согласно новому образу анимацию сканирования.

Рисунок 1. Прототип

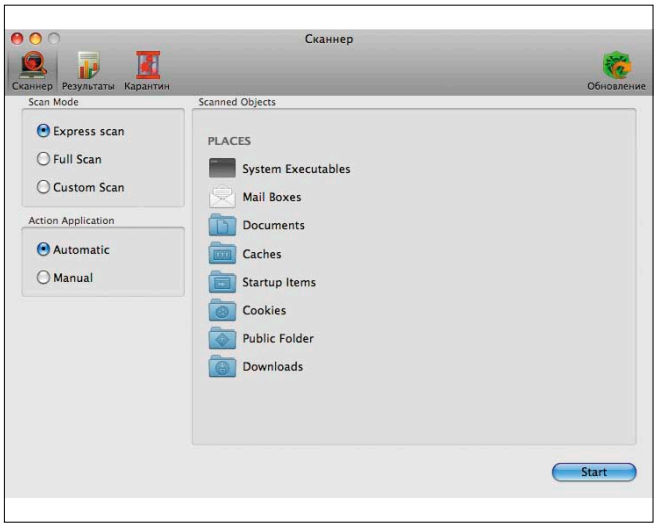
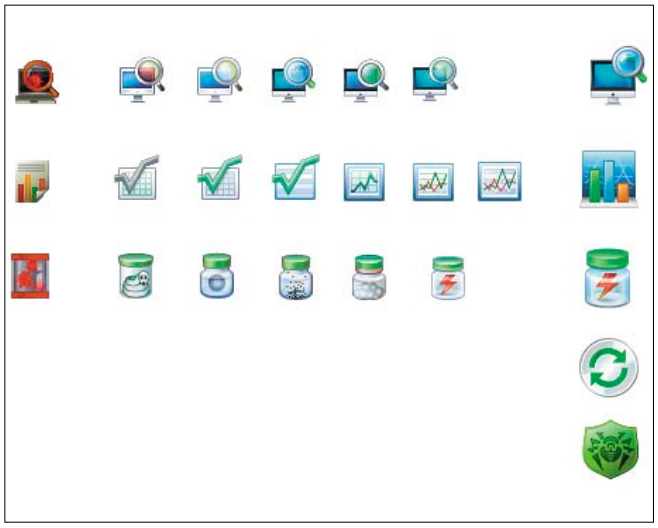


Рисунок 2. Эволюция иконок



Вид из окон становится интереснее

Окно сканирования имело один большой недостаток: кнопки управления сканирования находились далеко внизу, слева от прогресс-бара. Мы перенесли эти кнопки наверх.

Чтобы добиться явного и однозначного отображения завершения процесса, были добавлены графики по окончании сканирования.

Попутно из правого верхнего угла была убрана кнопка «Обновить». Тем самым мы избежали двусмысленности: кнопки слева управляют переключением вкладок, в то время как кнопка «Обновить» сразу запускала процесс обновления.

Пульт управления

Главной вкладкой у нас стал пульт управления всем антивирусом. Здесь пользователь сразу видит состояние компонентов продукта и получает информацию о безопасности системы.

На вкладке отображается процесс запуска задач на сканирование или обновления, время последнего обновления, статус файлового монитора, количество угроз, обнаруженных каждым модулем, не обработанных автоматически и ожидающих действий пользователя. Кстати, иконка обновления при запуске процесса обновления крутится по часовой стрелке (см. рис. 3).

Для удобства пользователей мы сочли полезным сделать размер главного окна изменяемым, и разработчики поддержали эту идею. Кстати, Dr.Web для Mac OS X стал первым антивирусом с такой возможностью!

Сканер

Окно сканера наследовало старую идеологию, и мы полностью его модернизировали. Впервые пользователи антивируса получили возможность создавать неограниченное количество профилей проверок.

В этом же окне мы поместили кнопку для ввода администраторского пароля – ранее сканер, не получив доступа, в самом конце проверки запрашивал пароль, что могло вызвать раздражение пользователя, замедлить процесс проверки и создать угрозу безопасности системы. Параллельно на все формы был установлен значок вызова контекстной справки (см. рис. 4).

Файловый монитор

Это единственный файловый монитор для Mac OS X, который позволяет отложить выполнение действий над обнаруженными угрозами.

Если пользователь настроил антивирус на информирование об угрозах, происходит следующее: после того как файловый монитор детектирует угрозу, всплывает виджет с информацией о количестве обнаруженных угроз.

Если произойдёт ещё один детект, отображаемое число угроз увеличится. Пользователь наводит курсор мыши на виджет, тот плавно раскрывается и трансформируется в таблицу с возможностью выделить все или отдельные файлы (угрозы) и обезвредить их автоматически или выбрать необходимое действие.

Мастер

Для полноценной работы антивируса необходимо получить лицензионный ключевой файл. Поскольку вариантов развития событий может быть несколько, мы решили пойти по пути создания Мастера.

Стоит отметить, что в Dr.Web для Mac OS X нет ни одного всплывающего окна или многоуровневой вложенности в настройках. Единственный случай, когда программа требует у пользователя подтверждения действий, – это момент безвозвратного удаления файлов из Карантина.

Задумались мы и о том, откуда пользователю удобнее вызывать антивирус: из Dock-панели, из системного меню или лучше комбинировать оба способа. Остановились на комбинированном способе и, судя по отзывам, не прогадали. Однако в следующей версии мы хотим предоставить пользователю самому решать, как именно вызывать антивирус.

Так что работа над интерфейсом продолжается. И для того, чтобы он стал еще удобнее, нам очень важно знать мнение пользователей! EOF

Рисунок 3. Пульт управления

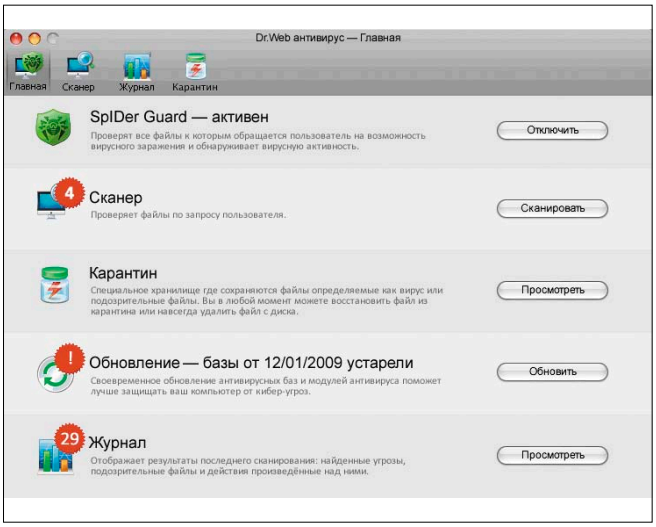
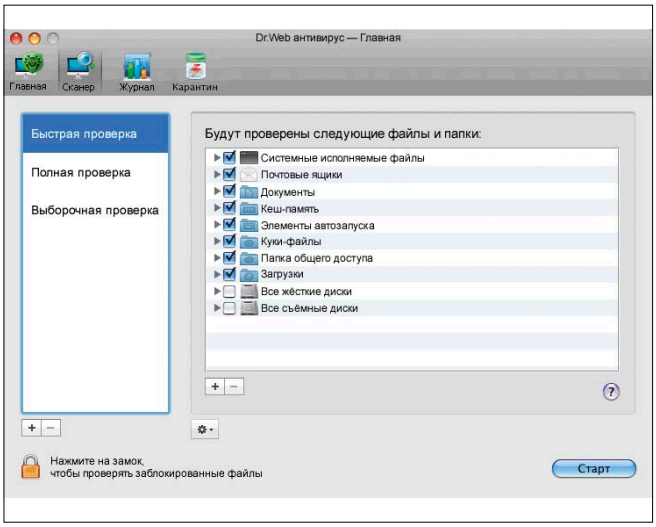


Рисунок 4. Окно сканирования



Визитка

АНДРЕЙ ЛУКОНЬКИН, ведущий инженер-программист ОАО «НижегородАвтоДор». Занимается автоматизацией производства, бухгалтерского, управленческого и кадрового учета

Библиотека стандартных подсистем

Обсуждаем плюсы и минусы новшества

Фирма «1С» объявила о выпуске ознакомительной версии инструментария для разработчика «1С:Библиотека стандартных подсистем 8.2». Попробуем вникнуть в суть новинки

Теперь помимо платформы «1С:Предприятие» фирма «1С» предлагает готовую базовую функциональность и технологию для разработки прикладных решений, использование которой позволит уменьшить время на создание и внедрение программного продукта.

«1С:Библиотека стандартных подсистем 8.2» (сокращенно БСП) содержит набор универсальных функциональных подсистем, которые могут быть использованы в разрабатываемой конфигурации как все вместе, так и по отдельности. Для чего это нужно? Предполагается, что с помощью БСП можно будет быстро создавать новые конфигурации с уже готовой базовой функциональностью, а также включать готовые функциональные блоки в существующие конфигурации. Проще говоря, теперь есть некий набор готовых подсистем, которые может использовать программист, не тратя время на изобретение велосипеда.

Состав подсистем ознакомительной версии

Стандартные подсистемы. Процедуры и функции общего назначения, настройка заголовка окна программы, сервисные возможности администратора системы.

Адресный классификатор. Предоставление адресного классификатора (КЛАДР), загрузка адресного классификатора с диска ИТС, из файла и с пользовательского раздела сайта фирмы «1С», периодическая проверка наличия обновлений адресного классификатора на сайте.

Бизнес-процессы и задачи. Информирование пользователя о его текущих задачах, интерактивный ввод задач, мониторинг и контроль исполнения задач со стороны заинтересованных лиц, базовая функциональность для разработки произвольных бизнес-процессов.

Валюты. Хранение и предоставление доступа к списку и курсам валют, загрузка курсов валют с веб-сайта rbc.ru, выбор валют из общероссийского классификатора ОКВ.

Версионирование объектов. Хранение истории изменений справочников и документов, сравнение произвольных версий объектов, просмотр ранее сохраненных версий объекта.

Завершение работы пользователей. Установка блокировок новых соединений с информационной базой, завершение существующих соединений.

Запрет редактирования реквизитов объектов. Проверка обязательного заполнения некоторых реквизитов объектов, запрет редактирования «ключевых» реквизитов записанных объектов, проверка возможности изменения «ключевых» реквизитов пользователем, имеющим на это права.

Контактная информация. Расширение состава реквизитов справочников конфигурации произвольным набором реквизитов, предназначенных для ввода контактной информации, использование функциональности подсистемы «Адресный классификатор».

Контроль динамического обновления конфигурации. Проверка факта динамического изменения конфигурации, оповещение пользователя с предложением перезапустить программу.

Контроль журнала регистрации. Просмотр журнала регистрации в режиме «1С:Предприятие», формирование отчета о критичных записях в журнале регистрации и периодическая рассылка отчета заданным получателям.

Печать. Формирование печатных форм объектов в виде табличных документов, вывод на печать и отправка печатных форм по электронной почте.

Полнотекстовый поиск. Настройка и использование полнотекстового поиска данных.

Получение файлов из Интернета. Программный интерфейс для получения файлов из Интернета, получение файла из сети на клиенте, сохранение файлов на клиентском компьютере, в информационной базе, запрос и хранение параметров прокси-сервера.

Пользователи. Просмотр и редактирование списка пользователей системы, управление правами пользователей, определение текущего пользователя при запуске системы.

Работа с файлами. Присоединение файлов к произвольным объектам конфигурации, присоединение фай-

лов из файловой системы или создание файлов по шаблону, коллективное редактирование файлов, хранение и предоставление доступа к версиям файлов, поддержка полнотекстового поиска по присоединенным файлам.

Регламентные задания. Отображение и настройка параметров регламентных заданий, поддержка выполнения регламентных заданий в файловой версии информационной базы.

Свойства. Создание и редактирование дополнительных свойств документов и справочников, хранение свойств в объекте (дополнительные реквизиты) и вне объекта в специальном регистре сведений (дополнительные сведения), возможность использования свойств в отчетах, поддержка возможности задавать разные наборы свойств для разных объектов одного и того же типа.

Сохранение настроек. Сохранение и загрузка вариантов отчетов и пользовательских настроек отчетов, доработка формы отчетов для удобного переключения пользовательских настроек и вариантов отчета.

Физические лица. Хранение и предоставление доступа к информации о физических лицах.

Функции отчетов. Программный интерфейс для настройки форм отчетов СКД.

По замыслу разработчиков этой библиотеки, использование БСП при написании приложений на платформе «1С:Предприятие 8.2» позволит сократить время, затрачиваемое на разработку, повысить качество прикладных решений, а также стандартизовать конфигурации.

Планируется, что фирма «1С» разрешит использовать БСП зарегистрированным пользователям системы «1С:Предприятие 8», имеющим действующую подписку на ИТС, без дополнительной оплаты.

Всё, казалось бы, хорошо и замечательно. Но только до тех пор, пока новость не стала обсуждаться специалистами на интернет-форумах. Приведу здесь некоторые отзывы (разделив их на положительные и отрицательные), после чего попробую подвести некоторые итоги обсуждений.

Положительные отзывы

- «Например, чтобы в 8.1 внедрить в свою конфигурацию адресный классификатор, загрузку валют, банков и прочие «хотелки», приходилось заморочиваться вырезанием этого функционала из типовой. Сейчас вроде будет все проще».
- «Если понадобился блок характеристик, например, можно скопировать его в конфигурацию, и через 10 минут все работает. По-моему, хорошая мысль».
- «Никто не обещал, что эта библиотека будет набором полностью готовых подсистем. Задача совершенно иная – дать разработчику возможность быстро решать часто возникающие конкретные задачи (включить функционал использования электронной почты или свойств и характеристик). Не более того. Если что-то не устраивает – напишите самостоятельно или переделайте предложенные».
- «Приходилось до этого вытаскивать код из типовых, что не всегда удобно, и не всегда ты сможешь подтянуть все что нужно, и ошибки вываливаются в самый неподходящий момент».
- «А зачем изобретать велосипед? Скажем, есть типовой механизм работы с пользователями, когда я пишу свою конфигурацию, у меня стоит куча задач – я возьму этот механизм из БСП и забуду о нем, сосредоточившись на выполнении основной задачи».
- «Я не понимаю воинствующих противников. Они никогда не использовали стандартных библиотек в других языках? Или они просто не знают о существовании таких языков и библиотек?»
- «Цель у БСП – облегчить жизнь как разработчикам, так и пользователям. И это касается в первую очередь тиражных решений. Разработчику не придется изобретать велосипед, а пользователю, перейдя с одной конфигурации на другую, не придется изучать заново работу стандартных механизмов».

Отрицательные отзывы

- «Так скоро и программисты не нужны будут... Потихоньку «1С» движется к настраиваемой системе, а не конфигурируемой. Уже меньше программиста, больше настройки».
- «В типовых конфигурациях характеристики, контактная информация и даже адресный классификатор сделаны методологически неоптимально. Теперь это навязывается всем. Теперь эти недоработки толкают в массы».
- «Раз это стандарт, он должен делаться хорошо, раз и навсегда».
- «Есть подозрение, что если разработчики новой конфигурации не использовали эти библиотеки, то конфигурация не пройдет в будущем сертификацию на «1С:Совместимо». Иначе зачем фирме «1С» унификация общих механизмов?»

***

Итак, что мы имеем? С одной стороны, есть готовые блоки, которые можно вставлять в свою конфигурацию, акцентируя внимание на решении более важных задач, чем управление пользователями или работа с файлами. С другой стороны, появляются опасения на счет неоптимальности предоставляемых типовых решений, как следствие – необходимость частого внесения изменений (дополнительные трудозатраты).

Если взвесить все «за» и «против», то лично я склоняюсь к тому, что плюсов все-таки гораздо больше. Образно говоря, фирма «1С» предоставляет «секции для забора», которые нужно лишь расставить по периметру и приступать уже непосредственно к строительству дома. Идеально написанную программу сложно себе представить, и даже если допустить, что в БСП какие-то методы неоптимальные, это всё равно удобнее, чем написание «с нуля». Выгодно это и конечным пользователям, т.к. им не придется оплачивать работу программиста по написанию адресного классификатора и тому подобных типовых вещей.

Библиотека стандартных подсистем – новое явление в мире «1С», но уже сейчас оно имеет своих сторонников и противников. Как показывает практика, критика только способствует совершенствованию продукта. Поэтому обсуждение всех новшеств – важный этап в становлении действительно хорошего решения. EOF

Визитка

ИГОРЬ ШТОМПЕЛЬ, инженер, системный администратор. Сфера профессиональных интересов – GNU/Linux, функциональное программирование

Open Web Tools Directory

Собрать все значимые инструменты для веб-разработчика в одном месте всегда являлось привлекательной идеей. Недавно такую попытку предприняла компания Mozilla

Новый проект Mozilla Labs

6 июля 2009 года в блоге Mozilla Labs появилась информация о запуске нового проекта – Open Web Tools Directory. Основной целью реализации каталога стало желание произвести централизованное индексирование средств веб-разработчика. Как писал в сообщении от имени команды разработчиков Бен Гэлбрэйт (Ben Galbraith): «Есть огромное количество инструментов, которые были созданы, чтобы помочь веб-разработчикам. К сожалению, вам сложно узнать обо всех них, так как нет их централизованного индексирования» [1]. Новый каталог стал доступен по адресу http://tools.mozilla.com. Данная страница, как показано на рис. 1, отличается оригинальным дизайном, хотя и может вызывать определенные задержки в отображении даже у пользователей, использующих ADSL-соединение (64, 128 Кбит/сек). Впрочем, внизу страницы имеется ссылка «Plain HTML Version», которая отобразит «облегченный» вариант каталога, как показано на рис. 2, без дополнительных эффектов.

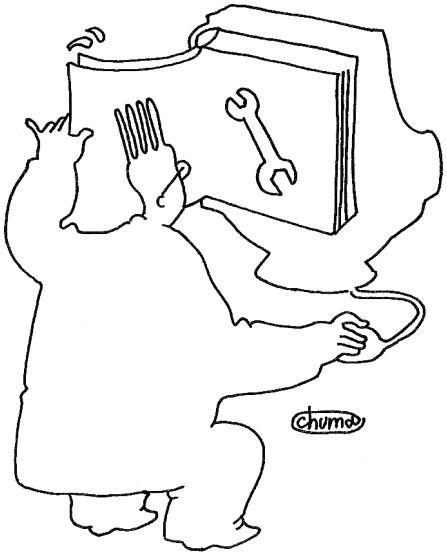
Также в нижней части главной страницы каталога доступен поисковый инструмент. Он имеет шесть тематических вкладок: Design, Code, Debug, Test, Deploy, Docs. Design.

Рисунок 1. Сайт проекта Open Web Tools Directory

Выбор вкладки осуществляется щелчком по ней – будут отображены значки проектов данной тематики. Например, выбираем вкладку Design – отображаются значки проектов, тематически относящихся к данному разделу. Кроме этого щелчок на значке проекта приведет, как показано на рис. 3, к отображению краткой информации о данном инструменте. Повторный щелчок по активной вкладке отменит выбор – будут вновь отображены значки всех инструментов каталога.

Сам поисковый механизм отличается продуманной реализацией. Так, по мере ввода букв в строке поиска автоматически отбираются соответствующие текущему запросу проекты. Например, осуществим поиск инструмента Aptana, вводя в строку поиска буквы постепенно, наблюдая за изменениями каталога. На рис. 4 показано как изменилось отображение каталога после ввода apt, а на рис. 5 – после ввода aptana.

В Mozilla не собираются останавливаться на достигнутом и продолжают развитие проекта. Еще во время его запуска были очерчены ближайшие перспективы развития. Так, планируется исследование возможности реализации различных социальных аспектов (social participation features) для входящих в каталог средств разработки. Например, ранжирование и комментарии. Кроме того, ведется работа над усовершенствованием механизма поиска Open Web Tools Directory, который был рассмотрен выше.



Есть много инструментов, которые созданы, чтобы помочь веб-разработчикам. Но сложно узнать обо всех…

Помочь развитию проекта могут все желающие. Для этого, если вам известен какой-либо инструмент, отсутствующий в каталоге, вы можете заполнить специальную форму [2].

Инструменты Open Web Tools Directory

Рассмотрим, какие средства доступны для разработчиков веб-проектов. Начнем обзор с такого инструмента, как **ajaxload.info** (http://www.ajaxload.info). Данный проект ориентирован на создание иконки – индикатора загрузки на базе технологии Ajax. Ajaxload.info дает возможность выбора цвета иконки и ее фона. Инструмент доступен бесплатно.

Aptana Studio (http://www.aptana.com/studio) – среда веб-разработки, основанная на Eclipse с поддержкой HTML, CSS, JavaScript и DOM. Дополнительные плагины позволяют добавить в Aptana Studio возможность работы с PHP, Python, Ruby on Rails, Jaxer, Adobe AIR, Apple iPhone и Nokia S60. Данная среда разработки имеет поддержку следующих AJAX-библиотек: jQuery, Prototype, script.aculo.us, Dojo, Ext, YUI, MochiKit, MooTools, SPRY и Aflax.

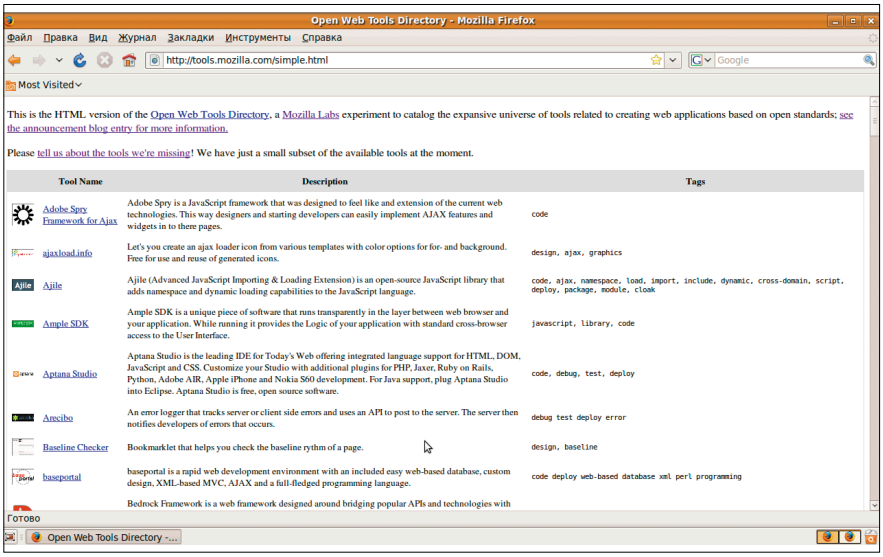
Baseline Checker (http://scribu.net/util/baseline-checker-bookmarklet.html) представляет собой специальную закладку, осуществляющую проверку расположения базовых линий (baseline rythm of a page) на веб-странице. При использовании утилиты страница становится похожа на «тетрадь в линию». Под линиями, расположенными на определенном расстоянии друг от друга, видны все элементы веб-страницы.

Bespin (http://bespin.mozilla.com) является основанным на веб-технологиях расширяемым редактором кода, разработанным с использованием технологий HTML 5. Данный продукт является детищем Mozilla Labs. Для работы с Bespin требуется регистрация.

Color Scheme Designer (http://colorschemedesigner.com) – это инструмент с богатыми возможностями, который поможет подобрать цветовую схему. Среди последних, например, имеется возможность экспорта созданной цветовой схемы в HTML+CSS, XML, Text, ACO (палитра Photoshop), GPL (палитра GIMP).

CSS Sprite Generator (http://spritegen.website-performance.org) позволяет автоматизировать процесс создания css-спрайтов. Напомню, css-спрайты – это механизм, ориентированный на сокращение количества HTTP-запросов для графических ресурсов. Утилите указывается расположение двух или более графических файлов (в форматах .png, .gif или .jpeg), а она генерирует sprite-изображения и соответствующий css-код для их отображения. Исходный код инструмента доступен для загрузки под лицензией BSD, а сайт доступен на русском языке.

Рисунок 2. «Облегченный» вариант Open Web Tools Directory



CSS Text Wrapper (http://www.csstextwrap.com) дает возможность работать с обтеканием HTML-текста. Можно добиться того, чтобы текст обтекал кривые, зигзаги или другие формы. Для этого разработчик управляет левой и правой линией инструмента, а последний генерирует соответствующий код. Пример управления обтеканием текста показан на рис. 6.

DHTMLX Suite (http://dhtmlx.com) представляет собой набор компонентов для создания «богатых» интерфейсов на базе технологии Ajax. Последний включает наиболее часто используемые компоненты пользовательского интерфейса: grid, treegrid, treeview, combobox, tabbar, menu, windows, calendar, layout и другие. Все компоненты набора написаны на чистом JavaScript. Они совместимы с таким браузерами, как Mozilla Firefox, Google Chrome. Opera, Safari, Internet Explorer. Набор доступен под GPL (Standart Edition), коммерческой и enterprise (Professional Edition) лицензиями. В коммерческой лицензии в отличие от enterprise имееются ограничения – например, максимальное число разработчиков равно пяти.

Doodle.js (http://www.lamberta.org/blog/doodle) – это проект, реализующий спрайт-библиотеку для Canvas. Библиотека содержит ряд примитивов, реализующих полноценное API для рисования (a full-fledged drawing api), а также, является основой для создания спрайтов и взаимодействия с ними. Doodle.js лицензирована под MIT License.

Ряд инструментов из Open Web Tools Directory ориентирован на создание иконок (favicons) – специальных значков, которые отображаются в адресной строке (могут быть анимированными). Среди них – **Favicon.cc** (http://www.favicon.cc)и**FaviconGenerator** (http://favicon-generator.org/editor). Оба инструмента обладают возможностью импорта изображения (первый из форматов .png, .jpg, .jpeg, .gif, .bmp, .ico, а второй – .png, .jpeg и .gif) и содержат галереи уже созданных иконок.

Firebug (http://getfirebug.com) представляет собой утилиту, которая выполнена в виде расширения для Mozilla Firefox (последняя версия Firebug на сегодняшний день 1.4.2, работает с данным браузером версии 3.*). Она позволяет редактировать, производить отладку и контролировать код CSS, HTML, JavaScript на веб-страницах.

Кроме того, имеется ряд утилит расширяющих функциональные возможности Firebug. Это и **FirePython** (http://firepython.binaryage.com), предоставляющий возможность работы с кодом на языке Python; и **FireQuery** (http://firequery.binaryage.com) – специализированный набор для работы с Jquery; и **FireRainbow** (http://firerainbow.binaryage.com), обеспечивающий подсветку синтаксиса Java Script; и **FireUnit** (http://fireunit.org), который предназначен для тестирования приложений JavaScript. Firebug распространяется по лицензии BSD.

Ряд разработок Google также вошли в каталог от Mozilla. **Google Plugin for Eclipse** (http://code.google.com/eclipse) позволяет осуществлять разработку с использованием Google Web Toolkit и приложений App Engine.

Google Web Toolkit или GWT (http://code.google.com/webtoolkit) ориентирован на создание высокопроизводительных приложений AJAX. Разработчик создает код интерфейса на Java, а GWT компилирует исходный код в оптимизированный JavaScript.

PageSpeed (http://code.google.com/speed/page-speed) представляет собой дополнение для Firefox/Firebug. Веб-разработчики могут использовать PageSpeed с целью оценки эффектив-

Рисунок 3. Отображение краткой информации об инструменте каталога

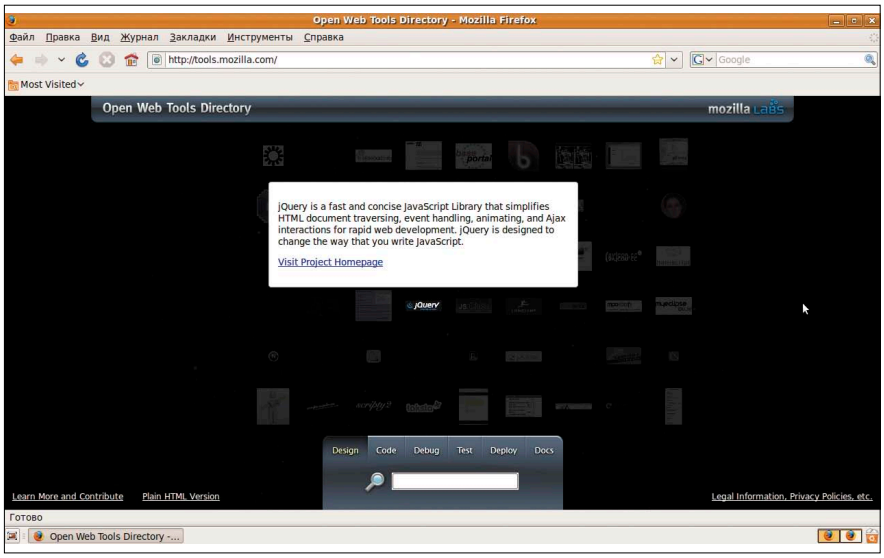
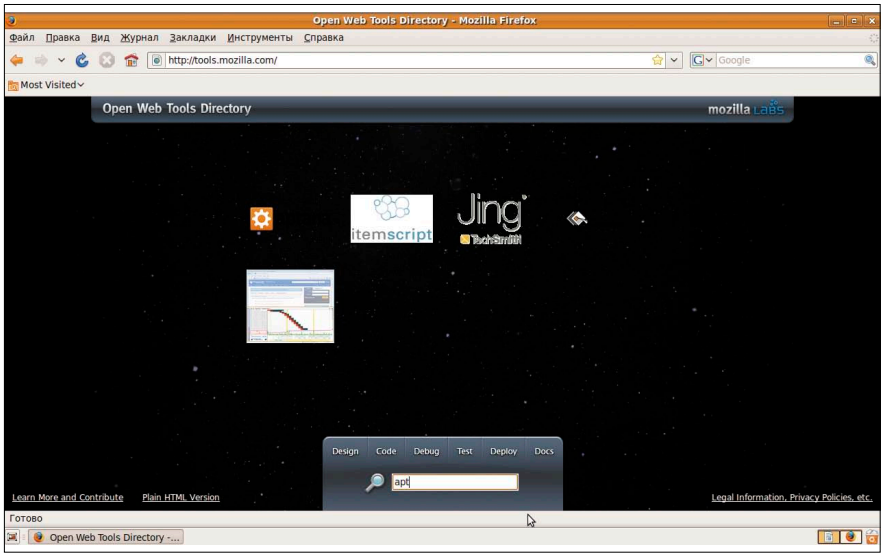


Рисунок 4. Состояние каталога после ввода в строке поиска apt



ности своих веб-страницах, а также получения рекомендаций по их улучшению.

JQuery (http://jquery.com) – это библиотека JavaScript, которая упрощает обработку событий, работу с анимацией и взаимодействие с AJAX для быстрой веб-разработки. Библиотека имеет двойное лицензирование под MIT и GPL.

Возможности работы с jQuery расширяют **jQuery Tools** (http://flowplayer.org/tools/index.html) – набор наиболее важных компонентов пользовательского интерфейса для веб-сайтов и **JQuery Values** (http://code.google.com/p/jquery-values) – инструмент, который позволяет использовать HTML как синтаксис для двунаправленных (bi-directional) шаблонов JavaScript.

OpenKomodo (http://www.openkomodo.com) – среда разработки с использованием динамических языков. Она основана на веб-технологиях и легко расширяется с использованием XML, JavaScript, Python и C++.

Pencil (http://www.evolus.vn/Pencil) – это инструмент, который используется для создания диаграмм и прототипирования GUI. Он доступен под лицензией GPL. Может быть установлен как расширение для Mozilla Firefox 3.*, а также как самостоятельное приложение на базе XULRunner.

Prototype (http://prototypejs.org) представляет собой фреймворк JavaScript, созданный с целью упрощения разработки динамических веб-приложений. Фреймворк отличается простотой в использовании. Prototype доступен под лицензией MIT для исходного кода и CC BY-SA для документации.

Pylot (http://www.pylot.org) ориентирован на тестирование производительности и масштабируемости веб-сервисов и является свободным программным обеспечением (лицензия GPL). Pylot генерирует для обеспечения требуемой нагрузки HTTP-запросы, проверяет ответы сервера, и создает отчеты с метриками. Набор тестов может выполняться и контролироваться с использованием графического интерфейса или shell.

SpourtCore (http://www.sproutcore.com) – это Open Source-фреймворк для создания облачных приложений на базе стандартов HTML 5 и JavaScript. Он включает в себя фреймворк JavaScript, набор утилит для сборки.

Web Developer (http://chrispederick.com/work/web-developer) представляет собой плагин для Mozilla Firefox/Flock/Seamonkey с большим количеством полезных инструментов для веб-разработчиков. Например, разметка документов, отображение разнообразной информации (CSS, о стилях CSS и другой), валидация. Web Developer – это свободное программное обеспечение, лицензированное под GPL.

***

Были рассмотрены наиболее интересные из инструментов на субъективный взгляд автора. Стоит отметить, что проект отличает широта предоставленного программного обеспечения, а также простота использования. Приятное впечатление производит и реализованный механизм поиска. Как представляется, данный каталог ждет дальнейшее развитие и расширение, в том числе при участии сообщества. Хочется пожелать увеличения количества инструментов, представляющих собой свободное программное обеспечение. EOF

1. https://labs.mozilla.com/blog/2009/07/open-web-tools-directory.
2. http://spreadsheets.google.com/viewform?formkey=cjJTMzFrOGUtcXRYRm9rcUQtTDd4UkE6MA.

Рисунок 5. Изменение состояние каталога после ввода в строке поиска aptana

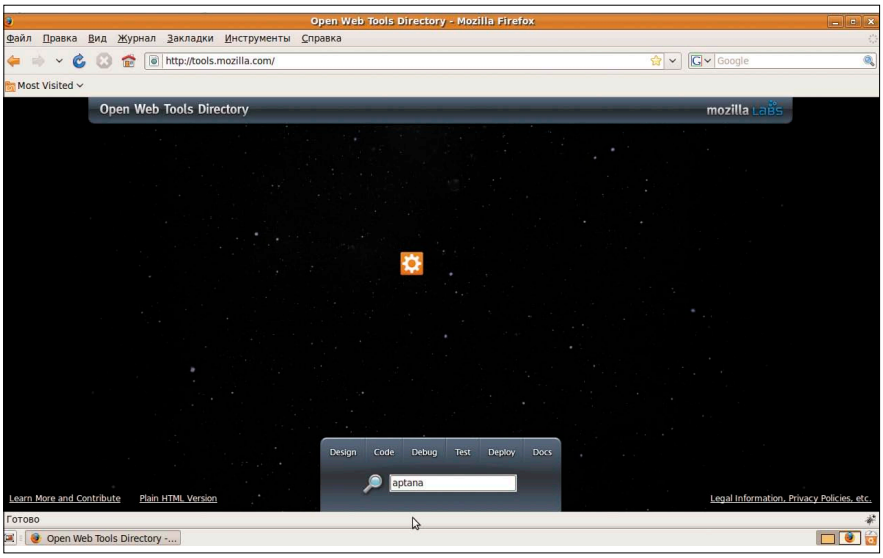
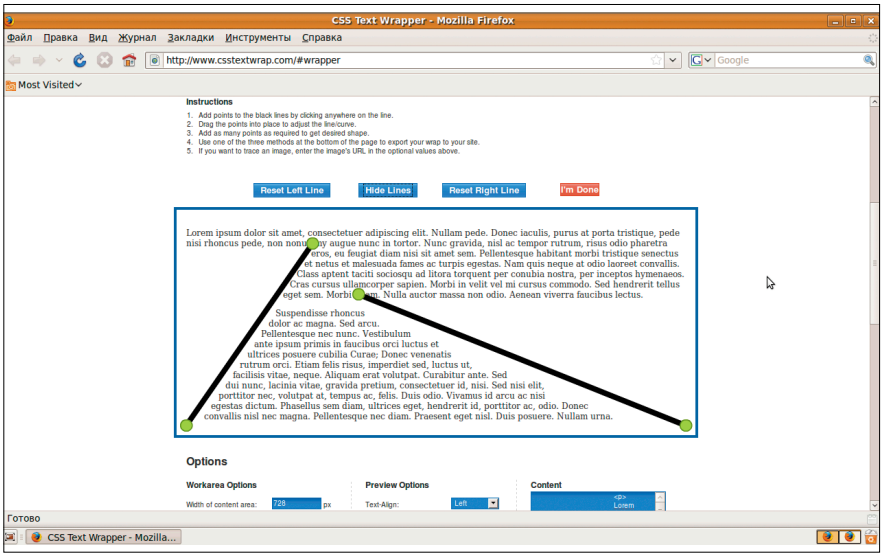


Рисунок 6. Управление обтеканием текста с использованием CSS Text Wrapper



Визитка

АНТОН ПИЩУЛИН, главный специалист ОАО «ОТП Банк», занимается разработкой и интеграцией корпоративных информационных систем

DBMS_SHEDULER

Запуск последовательных и зависимых заданий

В корпоративных информсистемах бывает необходимо выполнить расчеты для отчетов и витрин данных для конечных пользователей

Зачастую эти все вычисления нельзя запустить одним процессом, так как в ходе выполнения расчетов необходимо дождаться ряда событий (например, окончания выгрузки из одной системы в другую для какой-то определенной задачи). Можно, конечно, попытаться спрогнозировать, когда произойдут все необходимые события, и запустить весь расчет одним процессом, но в этом случае конечный пользователь увидит результат позже, что может повлиять на оперативность принимаемых решений особенно в условиях перерасчета большого объема данных.

В статье демонстрируется одно из возможных решений задачи оптимизации запуска связанных между собой процессов на платформе Oracle и объединенных в единую систему управления на основе пакета DBMS_SCHEDULER. В примере используется встроенный в Oracle механизм связанных задач (цепочки), объединенных вместе в одну пакетную задачу, которая запускается по сообщениям от клиента. Сообщения записываются и читаются клиентами из очереди Oracle. Все примеры кода приведены на PL/SQL, а все выполненные действия можно сделать как с помощью кода, так и с помощью Oracle EM (DBCONSOLE).

Немного теории

Для того чтобы приступить к демонстрации технологии связанных цепочек задач Oracle, приведем немного теории. В основе данной технологии лежит планировщик задач с расширенными функциями (расписаниями). Расписание в Oracle 10g является отдельной сущностью. Расписания бывают стандартными (старый вариант планировщика на основе пакета DBMS_JOB), на основе календаря (с возможностью задания расширенных правил на календарь), на основе событий (основан на чтении пользовательских сообщений из очереди). Последний вариант мы будем рассматривать далее, так как это соответствует нашей изначальной постановке задач.

Любая задача (Job) Oracle может быть привязана к расписанию. В свою очередь одна задача (Job) Oracle 10g может быть нескольких типов. Нас интересует прежде всего пакетный тип, когда в качестве запускаемого кода выбирается цепочка процессов (CHAIN). Процессы в цепочке могут быть простыми программами (процедурами) Oracle или расписаниями, или другими цепочками. Мы будем рассматривать самый простой – первый вариант. То есть предположим, что у нас есть три хранимые процедуры. Две первые из них должны запускаться параллельно при наступлении некоторого события, а третья дожидается выполнения их, только затем начинает свою работу. Для практического выполнения пользователю необходима привилегия dba.

Создание необходимых объектов очереди

Поскольку события запуска и остановки цепочки будут находиться в очереди сообщений Oracle, то на первом этапе нам необходимо создать необходимые для этого объекты (в Oracle EM Maintenance → Management → Messaging): тип сообщения и таблицы очереди, а также непосредственно самой очереди.

```
--Создание типа сообщения
create or replace type dwh_que_type as object
(
  msg_id number,
  msg varchar2(50)
);

--Создание таблицы DWH_QUE_TEST
begin
    dbms_aqadm.create_queue_table(
    queue_table => 'DWH_QUE_TEST',
    queue_payload_type => 'DWH.DWH_QUE_TYPE',
    sort_list => 'ENQ_TIME',
    multiple_consumers => true,
    message_grouping => sys.dbms_aqadm.transactional);
end;

--Создание очереди DWH_QUE_TST
begin
    dbms_aqadm.create_queue(
    queue_name => 'DWH_QUE_TST',
    queue_table => 'DWH_QUE_TEST',
    queue_type => sys.dbms_aqadm.normal_queue,
    max_retries => 500,
    retry_delay => 0,
    retention_time => 0);
end;
```

Создание расписания, цепочки и рабочих процедур

Далее переходим к созданию цепочки и расписания, а также макета рабочих (полезного бизнес кода) процедур, которые и будут участвовать в данном примере. А именно: будут созданы три процедуры, две из которых стартуют сразу, а третья дожидается выполнения первой и второй процедуры и начинает свою работу.

```
--Создание расписания TEST_SCH
--(в Oracle EM Administration → Schedules)
BEGIN
    dbms_scheduler.create_event_schedule(
    event_condition => 'tab.user_data.msg = ''TST_BEGIN''',
    queue_spec => '"DWH"."DWH_QUE_TST"',
    start_date => systimestamp at time zone 'Europe/Moscow',
    schedule_name => '"DWH"."TEST_SCH"');
END;

--Создание пустой цепочки DWH_CH_TST
--(в Oracle EM Administration → Chains)
BEGIN
sys.dbms_scheduler.create_chain(
chain_name => '"DWH"."DWH_CH_TST"');
END;

--Создания рабочих процедур
create or replace procedure p1
is
n number;
BEGIN
   n:=0;
   --Здесь нужно вставить рабочий код.
   --Если возникает ошибка, то ее обязательно нужно
   --обработать и сгенерить новый exception
exception when others then
   raise_application_error(-20112, 'p1');
END p1;

create or replace procedure p2
is
n number;
BEGIN
   n:=0;
   --Здесь нужно вставить рабочий код.
   --Если возникает ошибка, то ее обязательно нужно
   --обработать и сгенерить новый exception
exception when others then
   raise_application_error(-20112, 'p2');
END p2;

create or replace procedure p3
is
n number;
BEGIN
   n:=0;
   --Здесь нужно вставить рабочий код.
   --Если возникает ошибка, то ее обязательно нужно
   --обработать и сгенерить новый exception
exception when others then
   raise_application_error(-20112, 'p3');
END p3;
```

Создание программ Oracle

Теперь объединяем рабочие процедуры в программы, которые будут участвовать в качестве процессов в цепочке. Последняя программа pr3, кроме запуска рабочей процедуры p3, записывает сообщение (TST_END) окончания работы цепочки в очередь Oracle. Это сообщение будет прочитано клиентом, запустившим цепочку.

```
--Создание программ (в Oracle EM Administration → Programs)
BEGIN
DBMS_SCHEDULER.CREATE_PROGRAM(
program_name=>'"DWH"."pr1"',
program_action=>'begin
       p1();
       end;',
program_type=>'PLSQL_BLOCK',
number_of_arguments=>0,
enabled=>TRUE);
END;

BEGIN
DBMS_SCHEDULER.CREATE_PROGRAM(
program_name=>'"DWH"."pr2"',
program_action=>'begin
       p2();
       end;',
program_type=>'PLSQL_BLOCK',
number_of_arguments=>0,
enabled=>TRUE);
END;

--Последняя программа отличается от других
BEGIN
DBMS_SCHEDULER.CREATE_PROGRAM(
program_name=>'"DWH"."pr3"',
program_action=>'
       begin
         begin
         p3();
         end;
         --Далее идет сервисная часть, посылающая клиенту
         --сообщение о завершении работы цепочки
         DECLARE
           enqueue_options DBMS_AQ.enqueue_options_t;
           dequeue_options DBMS_AQ.dequeue_options_t;
           message_properties DBMS_AQ.message_properties_t;
           message_handle RAW(16);
           message DWH_QUE_TYPE;
         BEGIN
           message := DWH_QUE_TYPE(1, ''TST_END'');
           DBMS_AQ.ENQUEUE(
                queue_name => ''DWH_QUE_TST'',
                enqueue_options => enqueue_options,
                message_properties => message_properties,
                payload => message,
                msgid => message_handle);
           COMMIT;
         END;
       end;
       ',
program_type=>'PLSQL_BLOCK',
number_of_arguments=>0,
enabled=>TRUE);
END;
```

Создание шагов цепочки Oracle

Завершаем создание цепочки, то есть создаем шаги цепочки (S1, S2, S3) из созданных программ выше и определяем связи (правила) между ними. Чтобы цепочка начала свою работу, необходимо вначале (rule1) установить правило true для первых запускаемых программ. Второе правило (rule2) запускает шаг S3 при условии, что шаги S1 и S2 завершили свою работу. В конце создания цепочки необходимо ее активировать.

```
--Создание шагов цепочки
BEGIN
   dbms_scheduler.define_chain_step(
    chain_name => '"DWH"."DWH_CH_TST"',
    step_name => '"S1"',
    program_name => '"DWH"."pr1"');
END;
BEGIN
   dbms_scheduler.define_chain_step(
    chain_name => '"DWH"."DWH_CH_TST"',
    step_name => '"S2"',
    program_name => '"DWH"."pr2"');
```

```
END;
BEGIN
   dbms_scheduler.define_chain_step(
    chain_name => '"DWH"."DWH_CH_TST"',
    step_name => '"S3"',
    program_name => '"DWH"."pr3"');
END;

--Создание правил в цепочке
BEGIN
dbms_scheduler.define_chain_rule(
   chain_name => '"DWH"."DWH_CH_TST"',
   condition => 'true',
   rule_name => 'rule1',
   action => 'START "S1","S2"');
END;

BEGIN
dbms_scheduler.define_chain_rule(
   chain_name => '"DWH"."DWH_CH_TST"',
   condition => 'S1 COMPLETED AND S2 COMPLETED',
   rule_name => 'rule2',
   action => 'START "S3"');
END;

BEGIN
dbms_scheduler.define_chain_rule(
   chain_name => '"DWH"."DWH_CH_TST"',
   condition => 'S3 COMPLETED',
   rule_name => 'rule3',
   action => 'END');
END;

--Активация цепочки
BEGIN
dbms_scheduler.enable('"DWH"."DWH_CH_TST"');
END;
```

Создание пакетной задачи Oracle

Осталось только создать пакетную задачу определенного типа (chain), указать ей расписание (test_sch) и добавить пользователя dwh (под которым велась вся работа) в качестве подписчика очереди сообщений (dwh_que_tst). Затем остается только стартовать очередь и перейти к тестированию всей работы цепочки.

```
--Создание пакетной задачи
--(в Oracle EM Administration → Jobs)
BEGIN
dbms_scheduler.create_job(
   job_name => '"DWH"."JOB_TST"',
   job_type => 'CHAIN',
   job_action => '"DWH"."DWH_CH_TST"',
   schedule_name => '"DWH"."TEST_SCH"',
   job_class => '"DEFAULT_JOB_CLASS"',
   auto_drop => FALSE,
   enabled => FALSE);
   dbms_scheduler.set_attribute( name => '"DWH"."JOB_TST"',
        attribute => 'logging_level',
        value => DBMS_SCHEDULER.LOGGING_FULL);
   dbms_scheduler.enable( '"DWH"."JOB_TST"' );
END;

--Добавление пользователя dwh как подписчика очереди
--сообщений(в Oracle EM Maintenance → Management →
--Messaging)
declare
  subscriber sys.aq$_agent;
begin
  subscriber := sys.aq$_agent('"DWH"', '', 0);
  dbms_aqadm.add_subscriber(
        queue_name => '"DWH"."DWH_QUE_TST"',
        subscriber=> subscriber, rule=> '' ,
        transformation=> '');
end;
```

```
--Запуск очереди
begin
DBMS_AQADM.START_QUEUE(
   'DWH_QUE_TST');
end;
```

Тестирование работы цепочки

После помещения TST_BEGIN в очередь сообщений срабатывает расписание TEST_SCH и запускает пакетную задачу JOB_TST. Затем клиент читает очередь и ждет сообщения TST_END и завершает свою работу.

```
--Запуск цепочки
DECLARE
       enqueue_options     DBMS_AQ.enqueue_options_t;
       dequeue_options     DBMS_AQ.dequeue_options_t;
       message_properties  DBMS_AQ.message_properties_t;
       message_handle      RAW(16);
       message             DWH_QUE_TYPE;
    BEGIN
       message := DWH_QUE_TYPE(1, 'TST_BEGIN');

       --Помещение в очередь сообщения TST_BEGIN
       DBMS_AQ.ENQUEUE(
                 queue_name => 'DWH.DWH_QUE_TST',
                 enqueue_options => enqueue_options,
                 message_properties => message_properties,
                 payload => message,
                 msgid => message_handle);
       COMMIT;
    /*Ожидание сообщения о завершении работы цепочки*/
     DECLARE
       enqueue_options     DBMS_AQ.enqueue_options_t;
       dequeue_options     DBMS_AQ.dequeue_options_t;
       message_properties  DBMS_AQ.message_properties_t;
       message_handle      RAW(16);
       message             DWH_QUE_TYPE;
       s                   varchar(20);
       s2                  varchar2(20);
     BEGIN
       s2 := 'TST_END';
       s  := ''''||s2||'''';
       dequeue_options.consumer_name := 'DWH';
       dequeue_options.wait := DBMS_AQ.FOREVER;
       dequeue_options.navigation := DBMS_AQ.FIRST_MESSAGE;
       dequeue_options.deq_condition := ↵
                 'tab.user_data.msg = '||s;

       --Читаем очередь, ожидаем сообщения TST_END
       DBMS_AQ.DEQUEUE(
                 queue_name => 'DWH.DWH_QUE_TST',
                 dequeue_options => dequeue_options,
                 message_properties => message_properties,
                 payload => message,
                 msgid => message_handle);
       COMMIT;
    END;
END;
```

***

В статье была продемонстрирована возможность по оптимизации запуска связанных между собой процессов Oracle. В основе демонстрации был использован функционал Oracle 10g (пакет dbms_scheduler), позволяющий управлять цепочкой задач при помощи определенных правил и очередей сообщений Oracle. За более подробной информацией (описание пакета, атрибутов, свойств и т.д.) необходимо обращаться к официальной документации Oracle [1]. EOF

1. http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14258/d_sched.htm.

Множественные уязвимости в PostgreSQL

Программа: PostgreSQL версии до 8.4.1, 8.3.8, 8.2.14, 8.1.18, 8.0.22 и 7.4.2.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибки при подключении к LDAP-серверу с пустым паролем. Удаленный пользователь может обойти механизм аутентификации. Для успешной эксплуатации уязвимости требуется определенная настройка LDAP-сервера. Уязвимость распространяется на ветки 8.2 и 8.3.

2. Уязвимость существует из-за отсутствия защиты от выполнения RESET SESSION AUTHORIZATION в функциях индексирования. Удаленный пользователь может повысить свои привилегии в приложении. Уязвимые ветки: 8.4, 8.3, 8.2, 8.1, 8.0 и 7.4.

3. Уязвимость существует из-за ошибки, которая позволяет завершить работу сервера путем перезагрузки библиотек из $libdir/plugins. Для успешной эксплуатации уязвимости требуется, чтобы библиотеки присутствовали в каталоге $libdir/plugins. Уязвимые ветки 8.4, 8.3 и 8.2.
URL производителя: www.postgresql.org.
Решение: Установите последнюю версию 8.4.1, 8.3.8, 8.2.14, 8.1.18, 8.0.22 или 7.4.26 с сайта производителя.

Множественные уязвимости в Novell eDirectory

Программа: Novell eDirectory 8.7.3, возможно, другие версии.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за неизвестной ошибки в ndsd-демоне. Удаленный пользователь может аварийно завершить работу демона. Уязвимость распространяется на Novell eDirectory 8.7.3 SP10b.

2. Уязвимость существует из-за неизвестной ошибки в ndsd-демоне. Удаленный пользователь может вызвать переполнение стека и выполнить произвольный код на целевой системе. Уязвимость распространяется на Novell eDirectory 8.7.3 SP10b.

3. Уязвимость существует из-за неизвестной ошибки, которая позволяет удаленному пользователю остановить работу ndsd-демона. Уязвимость распространяется на Novell eDirectory 8.7.3.

4. Уязвимость существует из-за неизвестной ошибки, которая позволяет аварийно завершить работу ndsd-демона. Уязвимость распространяется на Novell eDirectory 8.7.3 SP10b.
URL производителя: www.novell.com/products/edirectory.
Решение: В настоящее время способов устранения уязвимости не существует.

Множественные уязвимости в ядре Linux

Программа: Linux kernel 2.6.x.
Опасность: Низкая.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибки в функции find_ie() в файле net/wireless/scan.c. Удаленный пользователь может с помощью специально сформированного пакета вызвать зацикливание. Уязвимость не распространяется на версии ядра до 2.6.30.

2. Уязвимость существует из-за ошибки проверки границ данных в функции perf_copy_attr() в файле kernel/perf_counter.c. Локальный пользователь может передать специально сформированные данные системному вызову perf_counter_open() и вызвать переполнение буфера. Уязвимость не распространяется на версии ядра до 2.6.31.

3. Уязвимость существует из-за того, что функция kvm_emulate_hypercall() в файле arch/x86/kvm/x86.c не отвергает гипервызовы MMU для непривилегированных гостевых ОС. Локальный пользователь может аварийно завершить работу ядра гостевой ОС.
URL производителя: www.kernel.org.
Решение: Установите исправление из GIT-репозитория производителя.

Отказ в обслуживании в реализации TCP/IP в продуктах Cisco

Программа: Cisco Adaptive Security Appliance (ASA) 7.x, 8.x; Cisco CATOS 7.x, 8.x; Cisco IOS 12.x, R12.x, XE 2.1.x, XE 2.2.x; Cisco NX-OS 4.x; Cisco PIX 7.x, 8.x.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибок в реализации протокола TCP при обработке состояний соединений. Удаленный пользователь может установить длительные TCP-соединения и потребить большое количество ресурсов.

2. Уязвимость существует из-за ошибки в Cisco NX-OS for Nexus 5000. Удаленный пользователь может с помощью специально сформированной последовательности TCP-пакетов вызвать отказ в обслуживании.
URL производителя: www.cisco.com.
Решение: Установите исправление с сайта производителя.

Уязвимость в реализации TCP/IP в Sun Solaris

Программа: Sun Solaris 8, 9, 10.
Опасность: Низкая.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за различных ошибок в реализации протокола TCP при обработке ресурсов. Удаленный пользователь может вызвать отказ в обслуживании системы.
URL производителя: www.sun.com.
Решение: В настоящее время способов устранения уязвимости не существует.

Составил Александр Антипов

Визитка

ДМИТРИЙ ВАСИЛЬЕВ, больше 10 лет профессионально занимается разработкой ПО, принимает активное участие в различных проектах с открытым исходным кодом

Основные грани Ruby

От главных конструкций до приложения

Ruby исповедует философию интуитивно понятного языка программирования, поддерживающего высокую производительность разработчика приложений

Особенности Ruby

Объектно-ориентированный язык. Каждый элемент в языке является объектом, и результат манипуляций с объектами также является объектом.

Поддержка автоматической «сборки мусора». Программисту не надо заботиться об освобождении памяти, она освобождается автоматически.

Поддержка исключений. Исключения позволяют изменять основной путь выполнения программы в случае возникновения каких-либо ошибочных ситуаций.

Поддержка итераторов и генераторов. Итераторы позволяют перебирать элементы коллекции вне зависимости от ее конкретной реализации. Генераторы, в свою очередь, позволяют создавать функции, возвращающие итератор, без необходимости создания какого-либо массива, а возвращая по одному элементу за раз.

Регулярные выражения на уровне языка. Регулярные выражения предоставляют стандартный способ работы с шаблонами для текста. Ruby позволяет создавать регулярные выражения на уровне синтаксиса языка, как, например, Perl или JavaScript.

Мультиплатформенность. Интерпретатор Ruby можно скачать практически для всех основных платформ, включая Windows, Linux, Mac OS X. Или он может быть собран из исходного кода.

Поддержка интроспекции и изменения объектов. Ruby позволяет исследовать и изменять объекты во время выполнения. В том числе, есть возможность изменять даже встроенные в язык классы.

Удобный синтаксис. Как и SmallTalk, Ruby позволяет не писать скобки при вызове методов, что делает его удобным для создания микроязыков.

Большая стандартная библиотека. Как и Python, Ruby поставляется с большим количеством стандартных пакетов.

Проекты, использующие Ruby

Ruby используется как скриптовый язык многими коммерческими компаниями, например NASA, Motorola и Lucent. Многие сайты используют Ruby как основной язык разработки, например:

Basecamp (http://www.basecamphq.com) – веб-приложение для управления проектами.

Twitter (http://twitter.com) – сервис микроблоггинга.

GitHub (http://github.com) – хостинг проектов, использующих систему контроля версий Git.

Также Ruby используется для многих проектов с открытым исходным кодом, например:

Ruby on Rails (http://www.rubyonrails.ru) – фреймворк для создания веб-проектов.

Capistrano (http://www.capify.org) – инструментарий для запуска скриптов на нескольких серверах.

RubyGems (http://docs.rubygems.org) – менеджер пакетов для Ruby.

Интерактивная оболочка

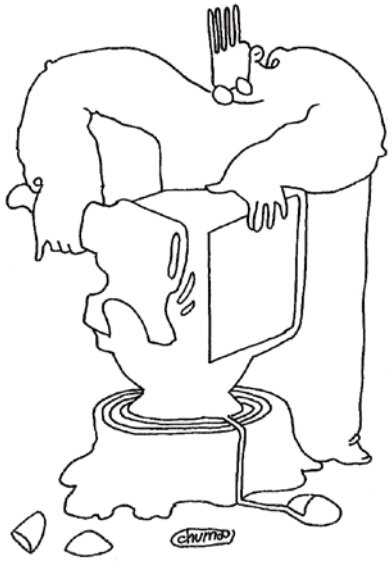
В примерах интерактивных сессий ниже будет использоваться интерактивная оболочка для Ruby – Interactive Ruby Shell (irb). Она входит в официальный дистрибутив Ruby, но в случае использования стандартной системы пакетов операционной системы irb может быть в отдельном от основного интерпретатора пакете и может потребовать отдельной установки. Попробуйте запустить оболочку командой irb (на Windows путь к оболочке должен быть установлен в переменной среды PATH), если вы увидите следующее приглашение, то интерпретатор и оболочка Ruby уже установлены в системе:

```
irb(main):001:0>
```

В противном случае установите интерпретатор и оболочку либо с помощью стандартной системы пакетов операционной системы, либо скачав инсталлятор или исходный код с официального сайта: http://www.ruby-lang.org/en/downloads.

Основные типы данных

Запустим интерактивную оболочку Ruby и начнем экспериментировать с основными типами данных.



Ruby используется как скриптовый язык многими компаниями – NASA, Motorola и Lucent

Числа и логические операции

Рассмотрим пример:

```
irb(main):001:0> (10 == 012) && (10 == 0xa) && (10 == 0b1010)
=> true
irb(main):002:0> 12345678901234567890 / 4.5
=> 2.74348420027435e+18
irb(main):003:0> -33 == 33
=> false
```

Надо сразу заметить, что трехзначное число в приглашении оболочки показывает номер текущей строки, в нашем случае – это 001, 002 и 003. В первой строке показаны различные способы записи числа 10:

012 – в восьмеричной системе счисления;

0xa – в шестнадцатеричной системе счисления;

0b1010 – в двоичной системе счисления.

В примере числа в каждой из систем счисления сравниваются на равенство с числом 10, и в результате возвращается (объект после знака =>) логическое true «истина».

Во второй строке длинное целое (Ruby автоматически поддерживает работу с целыми, больше размера машинного слова) делится на вещественное число и в результате получается вещественное число, которое выводится в экспоненциальной форме.

В третьей строке мы сравниваем отрицательное целое с положительным и, естественно, получаем в итоге логическое false «ложь».

В логических операциях в качестве значения false «ложь» могут выступать только предопределенные константы false и nil «нулевой объект». Все остальные объекты будут представлять из себя «истину».

Строки

Пример со строками:

```
irb(main):001:0> ("test" == 'test') && ("test" == %q/test/)
=> true
irb(main):002:0> <<END
irb(main):003:0" test
irb(main):004:0" END
=> "test\n"
irb(main):005:0> "test\n" != 'test\n'
=> true
irb(main):006:0> "#{10 * 10} times"
=> "100 times"
irb(main):007:0> '#{10 * 10} times'
=> "\#{10 * 10} times"
```

В первой строке показаны три способа создания строк – двойные кавычки, одинарные кавычки и %q. Последний способ %q позволяет использовать практически любые ограничители для строк и соответствует одинарным кавычкам (%Q соответствует двойным кавычкам). Строки в двойных кавычках позволяют использовать специальные символы, например перевод строки – \n. Строки в одинарных кавычках передают все символы «как есть». Различие между использованием одинарных и двойных кавычек показано в строках с 5-й по 7-ю. Для многострочных строк удобно использовать способ, показанный в строках со 2-й по 4-ю. Текст-ограничитель после символов << должен совпадать с текстом в конце строки и не входит в состав результирующей строки. Внутрь строк удобно включать небольшие выражения с помощью конструкции #{выражение}, как показано в строке 6. Но, как было описано выше, этот способ не работает для строк в одинарных кавычках.

Массивы

Обычные массивы индексируются по номерам элементов и создаются с помощью квадратных скобок:

```
irb(main):001:0> [1, 4.5, "test"]
=> [1, 4.5, "test"]
irb(main):002:0> [[1, 2], [3, 4]]
```

```
=> [[1, 2], [3, 4]]
irb(main):003:0> [[1, 2], [3, 4]][0]
=> [1, 2]
irb(main):004:0> [[1, 2], [3, 4]][-1]
=> [3, 4]
```

Как видно из примеров, массивы могут содержать смешанные элементы. В первой строке мы создали массив, содержащий целое и вещественные числа и строку. Во второй строке мы создали массив, содержащий два других массива, которые в свою очередь содержат целые числа.

Элементы в массивах доступны по индексам, при этом индексы могут быть отрицательными, и в этом случае отсчет будет идти с конца массива. Примеры в строках 3 и 4 иллюстрируют доступ к элементам массива.

Ассоциативные массивы

Ассоциативные массивы (хэши) индексируются не по номеру элемента, а практически любым объектом-ключом и создаются с помощью фигурных скобок:

```
irb(main):001:0> {"one" => 1, 2 => "two"}
=> {2=>"two", "one"=>1}
irb(main):002:0> {"one" => 1, 2 => "two"}["one"]
=> 1
irb(main):003:0> {"one" => 1, 2 => "two"}[2]
=> "two"
irb(main):004:0> {"one" => 1, 2 => "two"}["2"]
=> nil
```

В первой строке мы создали хэш, состоящий из двух элементов 1 и «two», которые индексируются соответственно строкой «one» и целым 2. Примеры в строках 2, 3 и 4 показывают доступ к элементам хэша. При этом запрос несуществующего элемента «2» (Ruby – это язык со строгой типизацией и не преобразует автоматически типы данных) возвращает «нулевой объект» nil.

Диапазоны значений

Для определения диапазонов значений предназначен тип, описывающий набор последовательных значений между двумя объектами:

```
irb(main):001:0> 1..10
=> 1..10
irb(main):002:0> "a"..."z"
=> "a"..."z"
irb(main):003:0> (1..10).to_a
=> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
irb(main):004:0> ("a"..."f").to_a
=> ["a", "b", "c", "d", "e"]
```

При этом, если начало и конец диапазона разделены двумя точками, то последнее значение будет включено в диапазон, а в случае трех точек – исключено.

Немного забегая вперед, в строках 3 и 4 мы используем вызов метода to_a(), преобразующего диапазон в массив. Здесь видно, что для доступа к методам объектов используется оператор «.» и для метода необязательно указывать скобки.

Переменные и управляющие конструкции

Для имен переменных Ruby использует префиксы, помогающие определить использование (или область видимости) имени:

- Локальные переменные, параметры методов и имена методов все должны начинаться с маленькой латинской буквы или знака подчеркивания.
- Глобальные переменные должны начинаться со знака $.
- Переменные экземпляров классов должны начинаться со знака @ и доступны только внутри объекта-экземпляра.
- Переменные классов должны начинаться с двух знаков @@ и разделяются между всеми экземплярами класса.
- Имена классов, модулей и константы должны начинаться с большой латинской буквы.

Рассмотрим небольшой пример:

```
irb(main):001:0> PI = 3.1415
=> 3.1415
irb(main):002:0> PI = 3.1415
(irb):2: warning: already initialized constant PI
=> 3.1415
irb(main):003:0> $count = 0
=> 0
irb(main):004:0> (1..10).each {|i| $count += i}
=> 1..10
irb(main):005:0> $count
=> 55
```

В первой строке мы создаем константу PI. В случае ее переопределения выводится предупреждение, хотя ее все еще можно переопределить.

В третьей строке мы создаем глобальную переменную $count. В четвертой строке используется новая конструкция Ruby – блок. В левой части вызывается метод-итератор each уже известного нам типа данных – диапазона значений. В правой части в фигурных скобках описывается блок кода, где между вертикальными чертами указывается переменная для подстановки, и затем код для выполнения. Полное выражение делает следующее: метод each вызывает блок кода для каждого значения в диапазоне от 1 до 10 и подставляет значение в локальную переменную i, после чего переменная складывается с глобальной переменной $count. Как мы видим, в пятой строке значение глобальной переменной содержит сумму всех элементов в диапазоне от 1 до 10.

Кроме фигурных скобок блоки кода могут быть определены с помощью конструкции do/end, удобной для многострочных блоков:

```
irb(main):001:0> (1..10).each do |i|
irb(main):002:1*     printf "#{i} "
irb(main):003:1> end
1 2 3 4 5 6 7 8 9 10 => 1..10
```

Заметьте, что у выражения есть также и возвращаемое значение (1..10).

Условные выражения

Выражение if..then/end позволяет выполнять ветки кода в зависимости от условий:

```
irb(main):001:0> n = 100

=> 100

irb(main):002:0> if n == 100 then
irb(main):003:1*     "hundred"
irb(main):004:1> elsif n == 10 then
irb(main):005:1*     "ten"
irb(main):006:1> else
irb(main):007:1*     "one"
irb(main):008:1> end

=> "hundred"
```

Также есть выражение «обратное» if – unless..then/end:

```
irb(main):001:0> n = 100

=> 100

irb(main):002:0> unless n == 100 then
irb(main):003:1*     "ten"
irb(main):004:1> else
irb(main):005:1*     "hundred"
irb(main):006:1> end

=> "hundred"
```

Выражение case/end сравнивает переменную по нескольким условиям:

```
irb(main):001:0> n = 100

=> 100

irb(main):002:0> case n
irb(main):003:1>     when 1...10 then 1
irb(main):004:1>     when 10...100 then 2
irb(main):005:1>     when 100...1000 then 3
irb(main):006:1> end

=> 3
```

Циклы

Кроме использования итераторов и блоков кода, Ruby предоставляет и более «классические» конструкции для создания циклов.

Конструкция while/end выполняет тело цикла, пока выражение не вернет «ложь» (необходимо помнить, что в качестве значения «ложь» можно использовать только false или nil):

```
irb(main):001:0> while line = gets
irb(main):002:1>     printf "Line: #{line}"
irb(main):003:1> end

line1
Line: line1
line2
Line: line2
=> nil
```

Здесь в качестве выражения используется функция gets, читающая данные со стандартного устройства ввода. Завершить ввод данных можно набрав «конец файла» – <Ctrl>+<D> (<Ctrl>+<Z> для Windows).

Конструкция until/end является «обратной» while и выполняет тело цикла, пока не будет получено значение «истина»:

```
irb(main):001:0> i = 0

=> 0

irb(main):002:0> until i > 5
irb(main):003:1>   i += 1
irb(main):004:1> end

=> nil

irb(main):005:0> i

=> 6
```

И, наконец, конструкция for..in/end позволяет выполнить итерацию по какому-либо объекту:

```
irb(main):001:0> for i in [100, 10, 1]
irb(main):002:1>     printf "#{i} "
irb(main):003:1> end

100 10 1 => [100, 10, 1]
```

Классы, объекты и методы

Рассмотрим классы, объекты и методы на примере небольшой иерархии, состоящей из одного абстрактного базового класса Shape и двух его наследников – Circle и Square. Создадим файл area.rb со следующим содержимым:

```
class Shape
    def area()
        raise "Method not implemented"
    end
end

class Circle < Shape
    attr_reader :radius
    def initialize(radius)
        @radius = radius
    end
    def area()
        Math::PI * @radius
    end
end

class Square < Shape
    attr_reader :side
    def initialize(side)
        @side = side
    end
    def area()
        @side * @side
    end
end
```

Базовый класс Shape содержит один метод area(), который при вызове поднимает исключение RuntimeError с параметром Method not implemented.

Класс Circle наследуется от класса Shape. Первая строка класса (attr_reader) служит для автоматического создания метода, возвращающего значение переменной экземпляра @radius (по умолчанию переменные доступны только внутри экземпляра класса). Метод initialize – это конструктор класса, получающий переменную radius и сохраняющий ее значение в переменной экземпляра @raidus. Метод area использует константу из стандартного модуля Math для вычисления площади круга.

Класс Square так же наследуется от класса Shape и похож на класс Circle, за исключением реализации метода area.

Рассмотрим работу классов в интерактивной сессии:

```
irb(main):001:0> require "area"
```

```
=> true
irb(main):002:0> s = Shape.new
=> #<Shape:0x7f82a7812028>
irb(main):003:0> s.area
RuntimeError: Method not implemented
        from ./area.rb:3:in `area'
        from (irb):3
        from :0
irb(main):004:0> c = Circle.new 10
=> #<Circle:0x7f82a77ff860 @radius=10>
irb(main):005:0> c.radius
=> 10
irb(main):006:0> c.radius = 20
NoMethodError: undefined method `radius=' for
#<Circle:0x7f82a77ff860 @radius=10>
        from (irb):6
        from :0
irb(main):007:0> c.area
=> 31.4159265358979
irb(main):008:0> sq = Square.new 10
=> #<Square:0x7f82a77e43a8 @side=10>
irb(main):009:0> sq.side
=> 10
irb(main):010:0> sq.area
=> 100
```

В первой строке мы загружаем наш файл с помощью метода require, который возвращает true в случае успеха. Затем мы создаем экземпляр класса Shape с помощью его метода new. В строке 3 мы пробуем выполнить метод area для экземпляра класса Shape, но, как и ожидалось, метод выкидывает исключение RuntimeError. В строке 4 мы создаем экземпляр класса Circle и в следующей строке запрашиваем значение атрибута radius. Конечно, атрибут нельзя присвоить, так как не был создан соответствующий метод присвоения. Метод area возвращает площадь круга в строке 7. Строки с 8-й по 10-ю создают экземпляр класса Square и вызывают его методы.

Создаем приложение

После того как мы создали несколько классов, воспользуемся ими для создания небольшого приложения, которое позволит вычислять площадь объектов в интерактивном режиме. Добавим в файл area.rb следующий код:

```
@shapes = {
    "circle" => Circle,
    "square" => Square
}

def area(line)
    parts = line.split
    raise "Wrong number of parameters" if parts.length < 2
    shape = @shapes.fetch parts[0], nil
    raise "Unknown shape" if !shape
    arity = shape.allocate.method(:initialize).arity
    sizes = parts[1..-1].map {|i| i.to_f}
    raise "Wrong number of arguments" if arity != ↵
        sizes.length
    shape.new(*sizes).area
end

def main()
    begin
        print "> "
        begin
            break if !line = gets
            puts area line
        rescue Interrupt
            break
        rescue RuntimeError => why
            print "?#{why}\n"
        end
    end while true
end

main
```

Этот код добавляет один хэш-массив, который сопоставляет имена объектов с их классами, и две функции:

main – основная функция приложения, вызов которой производится в самом конце файла. Функция содержит бесконечный цикл, выход из которого осуществляется оператором break, либо в случае признака «конец файла» (<Ctrl>+<D>, или <Ctrl>+<Z> для Windows), либо в случае прерывания выполнения с помощью <Ctrl>+<C>. Внутри бесконечного цикла функция gets читает строку, введенную пользователем, и вычисляет площадь объекта с помощью функции area. В случае ошибки в функции area она перехватывается оператором rescue RuntimeError и ее описание выводится на экран.

area – вычисляет площадь объекта по строке, введенной пользователем. Сначала строка разбивается на части по пробелам, или табуляциям. При этом число частей должно быть больше двух. Затем первый элемент массива используется для получения нужного класса вычисления площади из хэш-массива @shapes. После этого все остальные аргументы преобразуются в вещественные числа с помощью метода массива map и их количество сравнивается с количество аргументов конструктора выбранного класса. В итоге выбранный класс и аргументы используются для вычисления площади.

Запустим получившееся приложение и попробуем несколько примеров:

```
$ ruby area.rb
> circle 1
3.14159265358979
> circle 40.3
126.606183939669
> square 20
400.0
> unknown 30
?Unknown shape
```

Вы можете продолжить знакомство с языком по материалам следующих сайтов:

- http://preview.ruby-lang.org/ru – официальная страница языка по-русски;
- http://www.ruby.inuse.ru – блог о Ruby по-русски;
- http://rubyflow.ru – новости по Ruby на русском. EOF

Визитка

СЕРГЕЙ СУПРУНОВ, инженер электросвязи «широкого ИТ-профиля». В свободное время изучает FreeBSD и Python и пытается осмыслить свою нелюбовь к KDE

Язык Python

Дополнительные типы коллекций

Возможность разрабатывать свои собственные типы данных делает его гибким и мощным. Но прежде чем писать что-то своё, загляните в стандартную библиотеку

Коллекциями в языке Python принято называть типы данных, в том или ином виде хранящие ссылки на ряд других объектов (не обязательно однотипных). Коллекции делятся на последовательности, множества и отображения. Среди встроенных типов данных к первым относятся списки (тип данных list) и кортежи (tuple), ко вторым – обычные (изменяемые) и фиксированные множества (set и frozenset), и, наконец, к третьим – словари (dict).

Для большинства решаемых задач этих типов коллекций более чем достаточно, но иногда всё же возникает потребность в нестандартных типах. И по мере развития языка всё реже приходится прибегать к собственным реализациям, поскольку с каждым новым релизом стандартная библиотека Python предлагает всё больше дополнительных типов данных.

В данной статье мы рассмотрим типы коллекций, которые доступны в модуле collections стандартной библиотеки Python 3.1.

Именованные кортежи

Класс collections.namedtuple позволяет создать тип данных, ведущий себя как кортеж, с тем дополнением, что каждому элементу («полю») присваивается имя, по которому можно в дальнейшем получать доступ к нему как к атрибуту:

```
>>> import collections
>>> User = collections.namedtuple('User', ↵
        'name phone login speed')
>>> vasya = User('Петров В. И.', 22343, 'vasya', 128)
>>> for field in vasya:
...   print(field) # поведение аналогично кортежу
```

```
Петров В. И.
22343
vasya
128
```

```
>>> print(vasya[3]) # работает как обычный кортеж
```

```
128
```

```
>>> print(vasya.login, vasya.speed)
```

```
vasya 128
```

```
>>> tvasya = tuple(vasya) # если нужен простой кортеж
>>> tvasya
```

```
('Петров В. И.', 22343, 'vasya', 128)
```

Первым параметром функции namedtuple() передаётся имя создаваемого типа данных, которое будет возвращаться функцией type() и фигурировать, например, в сообщениях об ошибках. Второй параметр – разделённый пробелами список «полей» в том порядке, как они будут следовать в кортеже. С помощью дополнительного аргумента rename=True (появился в Python 3.1) можно включить автоматическое переименование тех полей, для которых задано недопустимое с точки зрения Python имя (например, не соответствующее правилам именования идентификаторов или одно из зарезервированных слов) – такие поля автоматически получат имена вида _0, _1 и т.д., число соответствует порядковому номеру поля:

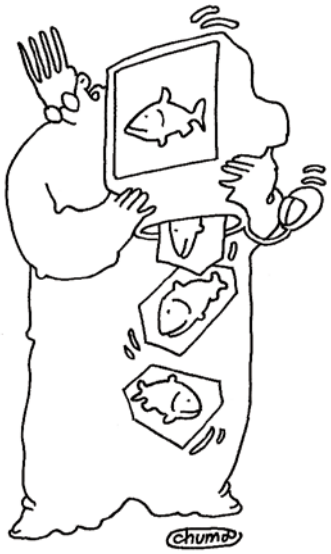
```
>>> Bool = collections.namedtuple('Bool', 'False True', ↵
        rename=True)
>>> test = Bool('2 > 3', '7 < 8')
>>> test
```

```
Bool(_0='2 > 3', _1='7 < 8')
```

Данный пример, конечно, искусственный. Реальная потребность в переименовании может понадобиться, если имена полей берутся из «внешних источников», например формируются на основании заголовка CSV-файла. Использование этого типа данных позволяет повысить удобочитаемость кода, поскольку не нужно постоянно вспоминать, какой смысл несёт значение под тем или иным индексом.

Раньше аналогичное поведение можно было получить, создав собственный класс с соответствующими атрибутами. Иногда, чтобы сделать код более понятным, использовался и такой приём:

```
NAME, PHONE, LOGIN, SPEED = (0, 1, 2, 3)  # или range(4)
vasya = ('Петров В. И.', 22343, 'vasya', 128)
print(vasya[LOGIN])
```



С каждым релизом стандартная библиотека Python предлагает все больше типов данных

Теперь всё доступно в стандартной библиотеке, и вам нет необходимости прикладывать лишние усилия.

Словари со значением по умолчанию

Рассмотрим небольшой пример:

```
>>> counter = {}
>>> sample = [1,2,1,4,2,3,4]
>>> for dig in sample:
...     counter[dig] += 1
...
```

```
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
KeyError: 1
```

Знакомая ситуация? Да, мы хотели в словаре counter посчитать число вхождений каждого элемента в списке sample. Но на первой же итерации возникла ошибка, так как словарь пуст и элемента с ключом 1 ещё нет – прибавлять единицу не к чему. В обычных словарях эта проблема решается либо предварительной проверкой на вхождение, либо с помощью методов словаря get() или setdefault(), либо обработкой соответствующего исключения:

```
# Первый способ
if dig in counter:
   counter[dig] += 1
else:
   counter[dig] = 1

# Второй способ
counter[dig] = counter.get(dig, 0) + 1

# Третий способ
counter[dig] = counter.setdefault(dig, 0) + 1

# Четвёртый способ
try:
   counter[dig] += 1
except KeyError:
   counter[dig] = 1
```

Однако в модуле collections стандартной библиотеки есть тип данных defaultdict, отличающийся от стандартного словаря как раз способностью автоматически создавать элементы для отсутствующих ключей:

Встроенные типы коллекций

Кратко напомню основные особенности встроенных коллекций:

Список (list) – тип данных, хранящий упорядоченный список ссылок на другие объекты (элементы списка). Обычно создается с использованием квадратных скобок: [1, 'два', 3]. (Замечание: помимо приведённого здесь и далее синтаксиса, объект любого типа данных в Python может быть создан вызовом соответствующей (одноимённой имени типа) функцией. Например, int(5) создаст целое число со значением 5, list() создаст пустой список, и т.д.)

Кортеж (tuple) – неизменяемый набор элементов. Создается с использованием круглых скобок: (1, 'два', 3). В отличие от списка, кортеж представляет из себя единый объект, состоящий из нескольких элементов, и соответственно элементы нельзя изменять, удалять, добавлять иначе, как путём создания нового объекта. Благодаря неизменяемости операции с кортежем выполняются несколько быстрее, чем со списком, и его можно использовать там, где требуется «хэшируемый» объект, например, ключ словаря (см. далее).

Множество (set) – неупорядоченный список уникальных элементов. Создается с использованием фигурных скобок: {1, 'два', 3}; пустое множество можно создать только вызовом функции set(). Все дублирующиеся элементы удаляются. Элементами могут быть только хэшируемые (неизменяемые) объекты – числа, строки, кортежи, фиксированные множества (frozenset). Благодаря использованию хэш-функции множества поддерживают очень быструю проверку на вхождение элемента оператором in (в списках и кортежах оператор in работает методом прямого перебора). Множества (как «обычные», так и фиксированные, о которых речь идёт в следующем абзаце) по своему поведению аналогичны математическим множествам и поддерживают методы, реализующие основные операции над ними – объединение, пересечение, разность, проверка на включение и т.д.

Фиксированное множество (frozenset) – неизменяемое множество (создаётся только вызовом функции frozenset()). Благодаря неизменяемости операции с ними выполняются ещё быстрее, и их можно использовать там, где требуется хэшируемый тип данных (как элементы множеств или ключи словарей).

Словарь (dict) – тип данных, хранящий пары «ключ – значение». Создается с использованием фигурных скобок: {'one':1, 'two': 2, 'three': 3}. Ключами могут быть только хэшируемые объекты, значениями – объекты любых типов. Порядок следования элементов не определён.

```
counter = collections.defaultdict(int)
for dig in sample:
   counter[dig] += 1
```

Эта конструкция уже не вызовет никаких ошибок – если элемент с нужным ключом в словаре отсутствует, он будет создан автоматически, а значением по умолчанию будет результат выполнения «фабричной функции», переданной в качестве параметра при инициализации словаря. В качестве «фабричной» можно использовать любую функцию, возвращающую какое-либо значение. В нашем примере мы используем встроенную функцию int(), которая, будучи вызвана без аргумента, возвращает 0. Именно это значение и будет присвоено вновь создаваемому элементу.

Аналогично вы можете использовать собственную функцию (она должна уметь выполняться без аргументов) или lambda-функцию: collections.defaultdict(lambda: 'default') создаст словарь, в котором автоматически создаваемые элементы будут получать в качестве значения строку default. Учтите, что конструктору defaultdict() должна передаваться именно ссылка на функцию (то есть её имя), а не возвращаемое значение. Поэтому имя нужно указывать без скобок.

Только не увлекайтесь этим типом данных – иногда лучше получить ошибку исполнения из-за отсутствия ключа в словаре, чем логическую ошибку, которую будет гораздо сложнее отловить.

Копирование коллекций

Затронем и один «тонкий» вопрос реализации языка Python – копирование объектов. С «бытовой» точки зрения присваивание «b = a» должно бы породить объект b, являющийся копией a. Но на практике мы видим совсем другое:

```
>>> a = [1,2,3]
>>> b = a
>>> b[2] = 5
>>> a

[1, 2, 5]

>>> b

[1, 2, 5]

>>> a is b

True
```

Почему так происходит? Дело в том, что имя переменной содержит ссылку на объект (можно сказать, его адрес). Записывая «b = a», мы копируем в переменную b этот адрес, но все обращения по нему ведут к тому же объекту (обратите внимание, что выражение «a is b» возвращает True).

С неизменяемыми объектами – числами, строками, кортежами - этого не происходит, поскольку при любом «изменении» такого объекта фактически создаётся новый, и переменная b начинает указывать на него, в то время как a продолжает ссылаться на первоначальный объект.

Так как же создать копию изменяемой коллекции? Есть три основных способа: использование «полного» среза (b = a[:]), явное создание нового объекта на базе существующего (b = list(a)) и использование функции copy модуля copy (b = copy.copy(a)).

Всё это относится к так называемому «поверхностному» копированию – например, если копируемый список содержит вторым элементом словарь (точнее, ссылку на словарь), то список-копия будет ссылаться на тот же самый словарь. В итоге изменения в списках a и b не будут влиять друг на друга, а вот изменения в словаре a[1] будут видны и по ссылке b[1]. Чтобы создать абсолютно независимые копии, применяется «глубокое копирование», выполняемое функцией copy.deepcopy().

«Двусторонние очереди»

Тип deque позволяет очень просто имитировать такие последовательности, как очереди и стеки.

Например:

```
>>> fifo = collections.deque()
>>> stack = collections.deque()
>>> fifo.appendleft(1)
>>> fifo.appendleft(2)
>>> fifo.appendleft(3)
>>> fifo.pop()

1

>>> fifo.pop()

2

>>> stack.append(1)
>>> stack.append(2)
>>> stack.append(3)
>>> stack.pop()

3

>>> stack.pop()

2
```

В объект fifo мы добавляем элементы слева, а извлекаем справа, реализуя тем самым очередь типа FIFO (First In – First Out, «первый пришёл – первый вышел»). Объект stack ведёт себя по принципу LIFO (Last In – First Out, «последний пришёл – первый вышел»). Хотя в принципе ничто не мешает добавлять и извлекать элементы с любой из сторон. Соблюдение соглашений здесь целиком и полностью ложится на программиста.

Объекты класса deque, как и другие коллекции, являются итерируемыми – при желании вы всегда можете выполнить их обход в цикле. Можно обращаться к элементам и по индексу.

По большому счёту, продемонстрированное выше поведение неплохо обеспечивается и обычным списком: методы append() и pop() ведут себя точно так же, а работу с началом списка можно реализовать методами pop(0) и insert(0), то есть передавая в качестве аргумента индекс извлекаемого элемента (кстати, метод pop() объектов deque «умеет» извлекать только последний элемент, а метода insert() нет вообще).

Хотя методы appendleft() и popleft() класса deque, на мой взгляд, более очевидны. Также скорость их выполнения не зависит от количества элементов в очереди (в терминах оценки времени исполнения алгоритмов O(), вставка и извлечение элементов как сначала очереди, так и с конца, имеет производительность O(1)) – например, при большом числе элементов метод appendleft() выполняется в разы быстрее, чем метод insert() для списков.

Но зато тип deque предоставляет метод rotate(N), переносящий N последних элементов в начало очереди (для отрицательного значения N – с начала в конец; по умолчанию N=1). Так, последовательно применяя этот метод к очереди deque([1, 2, 3, 4, 5]) , мы будем получать deque([5, 1, 2, 3, 4]) , deque([4, 5, 1, 2, 3]) и так далее. Как пример использования, можно привести реализацию «русской рулетки»:

```
#!/usr/local/bin/python3.1

import collections, random

baraban = collections.deque((1,0,0,0,0,0))
baraban.rotate(random.randint(5,99))

for shot in range(len(baraban)):
   input('Чья очередь - нажмите Enter... ')
   if baraban[shot]:
      print('БА-БАХ!!!')
      break
   else:
      print('клац')

print('Финита ля комедия, господа!')
```

Упорядоченные словари

Класс OrderedDict появился в Python 3.1 и позволяет создавать словари с фиксированным порядком элементов. В стандартных словарях, как известно, порядок элементов не определён – они хранятся на основании хэш-функции, и любое изменение словаря может изменить и порядок, в котором элементы будут доступны при переборе (например, в цикле for ... in).

Если же порядок заполнения словаря для реализуемого вами алгоритма важен, можете воспользоваться классом OrderedDict – он гарантирует, что при переборе элементы словаря будут возвращаться строго в том порядке, в каком в словарь помещались (аналогично спискам):

```
>>> od = c.OrderedDict()
>>> od['w'] = 123
>>> od['a'] = 34
>>> od.keys()
```

```
KeysView(OrderedDict([('w', 123), ('a', 34)]))
```

При обновлении значения элемента он остаётся на своём месте, если же элемент удалить и тут же добавить вновь, он попадёт в конец списка. С помощью этого типа данных удобно, например, вести подсчёт некоторых событий, если помимо их количества важно знать и порядок их возникновения.

Счётчики

Ещё один тип данных, появившийся в модуле collections в Python 3.1, – Counter. Чуть выше, когда мы обсуждали словари со значением по умолчанию, мы в качестве примера решали задачу подсчёта количества одинаковых элементов в некотором списке. Благодаря классу Counter (и разработчикам языка Python) это теперь можно сделать одной строчкой:

```
>>> sample = [1,2,1,4,2,3,4]
>>> cnt = collections.Counter(sample)
>>> cnt
```

```
Counter({1: 2, 2: 2, 4: 2, 3: 1})
```

```
>>> cnt[4]
```

```
2
```

Столь же просто можно выполнить подсчёт любых элементов в любой последовательности, например посчитать частоту появления различных символов в файле (строка текста не вполне вписывается в понятие коллекции, поскольку является одним объектом, а не набором ссылок на другие, но везде, где подразумевается итерируемый объект, строка рассматривается как последовательность отдельных символов):

```
>>> text = open('/home/amsand/reconf.py').read()
>>> cnt = c.Counter(text)
>>> cnt.most_common(3)
```

```
[(' ', 1162), ('e', 397), ('s', 284)]
```

Метод most_common() возвращает указанное число наиболее часто встречающихся элементов, упакованных в кортежи (ключ, значение). Здесь «золото» взял пробел.

***

Как видите, в модуль collections добавлено уже довольно много типов «нестандартных» коллекций (а ведь ещё в версии 2.5 там были представлены только defaultdict и deque), которые могут в ряде случаев заметно упростить вашу программу и сделать её более понятной. Помимо рассмотренных «готовых к употреблению» типов данных, модуль collections предоставляет и ряд абстрактных классов и «классов-заготовок», на базе которых вы можете создавать свои собственные типы коллекций.

Подробную документацию к модулю collections вы найдёте по следующей ссылке: http://docs.python.org/3.1/library/collections.html. Также обратите внимание на модули array, queue и heapq – в ряде случаев они могут предоставить более соответствующие вашим задачам специализированные типы данных. EOF

ИЛЬЯ АЛЕКСАНДРОВ, постоянный автор ряда ИТ-журналов, студент исторического факультета СПбГУ. Специально – для «Системного администратора»

Лидер всегда щедр

За что конкуренты любили Sun?

Есть компании, которые больше, чем просто бизнес-проект. В их стенах не только зарабатывают деньги — там придумывают мир, в котором мы живем. Одна из таких компаний — Sun Microsystems

Скотт и другие

Скотт Макнили (Scott McNealy) родился в 1954 году, недалеко от Детройта. Его отец, как и многие в городе, работал на автозаводе, где дослужился до управляющего. Высокая должность позволила дать сыну образование в элитарной школе. Юный Скотт мечтал стать врачом, но уже тогда демонстрировал способности к математике и другим точным наукам. В итоге родители и учителя убедили его поступать на экономический факультет Гарварда.

После окончания университета Скотт принимает решение получить престижную степень MBA, для чего отправляется в Стэнфордский университет. Поступить ему удалось только с третьей попытки, но это того стоило: именно в Стэнфорде он знакомится с Винодом Хослой (Vinod Khosla) и Энди Бехтольшеймом (Andy Bechtolsheim), будущими основателями Sun Microsystems.

В ту пору Макнили всерьез собирался заниматься работой в оборонной области, на танковом заводе. Однако бывший школьный учитель экономики приглашает его в Onyx, компанию, производящую рабочие станции. Но спустя год трое выпускников Стэндфорда основали свою фирму Stanford University Network (Sun).

Стартап вчерашних студентов

Винод Хосла и Энди Бехтольшейм тогда, в 1982 году, работали инженерами. Им постоянно требовались компьютеры для CAD/CAM-программ. В те годы компьютеры для вычислений стоили колоссальные деньги, и техники на всех не хватало просто физически. Часто приходилось работать по очереди. Амбициозных молодых специалистов такое положение дел устраивать не могло. Они решили создать собственную компанию, которая вместо дорогих рабочих станций должна была производить более дешевые, но почти не проигрывающие по функционалу. С этой целью и была основана Sun Microsystems. Перед компанией стоял вопрос: за счет чего можно убавить стоимость компьютеров, сделать их более доступными, чем у конкурентов?

Ведь на рынке компьютеров для вычислений тогда существовала компания Apollo. Сейчас ее название уже никому ничего не говорит, а когда-то на рынке ИТ Apollo считалась одной из самых перспективных. Первое решение нашлось быстро – Sun стала разрабатывать компьютеры, которые могли работать на дешевом, а лучше вообще бесплатном программном обеспечении.

Так, в новую компанию был приглашен Билл Джой. В отличие от своих коллег – еще никому не известных специалистов и вчерашних выпускников – Джой пришел в Sun, уже имея лавры одного из создателей операционной системы BSD. Именно доработанную под нужды компании систему BSD стали устанавливать на новые рабочие станции. Из-за этого стало возможно продавать новые компьютеры всего за …20 тысяч долларов. Аналогичная продукция Apollo стоила на 5 тысяч дороже. Apollo разрабатывала собственную операционную систему, что обходилось компании очень дорого.

Про свою первую модель инженеры Sun вспоминают с большой охотой и ностальгией: 4 Мб оперативной памяти и стандартный шестнадцатиразрядный процессор MC68010 (см. рис. 1) от Motorolla, способный выполнять около 800 тысяч операций в секунду. А главное – наличие карты Ethernet. В то время локальные сети еще считались экзотикой, и возможность объединить компьютеры в сеть стала серьезным аргументом за Sun, особенно у клиентов, приобретающих большое количество компьютеров для офисов. Поддержка сети в совокупности с бесплатной открытой операционной системой сделала компьютер главным событием выставки Comdex в 1983 году.

Энди, Билл, Скотт и Винод получают свой первый большой контракт – компания Computervision (специализирующаяся на CAD-системах) закупает у них партию компьютеров на 40 миллионов долларов.

Сеть – это компьютер

В 80-е годы у Sun Microsystems развивалась в двух направлениях: создавала технологии для работы в сети и разрабатывала собственный процессор.

Успеху первых компьютеров компании немало способствовала встроенная сетевая карта, и это достижение нужно было развивать. В 1984 году Sun анонсирует технологию Network File System. С помощью NFS пользователи могли работать с информацией и фай-

лами на всех компьютерах, объединенных в сеть. Это была одна из самых лучших реализаций протокола для работы в сети. Похожие протоколы были разработаны и конкурентами, но Sun выгодно отличалась от них мультиплатформенностью. NFS позволяла объединять в сеть не только компьютеры от SUN Microsystems, она работала и с MS DOS, IBM DOS, с операционками от Apple. До этого подобное не делал никто.

Но революционным стала даже не сама разработка, а то, как ее представили другим компаниям. Любой мог за символическую цену купить лицензию на NFS, получить исходники и сколько угодно пользоваться наработками в своих проектах. В 1984 году компании очень бережно хранили свои проекты и делиться исходным кодом не считали нужным.

Этот жест символизирует идеологию Sun: отдать свои достижения конкурентам, пусть те сами развивают технологию. Но при этом Sun все равно умудрялась оставаться лидером в своей области, используя уже и то, что сделали другие компании. «Это не мы делаем им подарки, это конкуренты работают на нас», – шутили в компании.

Сетевые компоненты развивались и на благо разработчиков. В 1987 году публике был представлен Open Network Computing, самый совершенный в то время механизм для реализации распределенных вычислений. Теперь в научных центрах для особо сложных и ресурсоемких задач появилась возможность объединять мощь всех имеющихся компьютеров.

В 1986 году происходит еще одно важнейшее событие в истории компании. Инженеры представляют свой процессор, который должен заменить исчерпавшие свой потенциал модели Motorolla. Новый 32-разрядный процессор работал на RISC-архитектуре, что упрощало усовершенствование старых моделей и увеличивало производительность. Новая архитектура была названа SPARC.

SPARC превосходил процессоры от Motorolla технически и при этом обходился компании дешевле. Первые модели SPARC (см. рис. 2) работали на тактовой частоте 16,6 МГц и позволяли установить на один сервер сразу несколько процессоров. Sun осталась верна своим традициям, и технология производства была доступна всем желающим. Лицензии на процессоры SPARC получили четыре компании.

Правда, относительно удачным из всех четырех был только опыт Cray. Они продавали дорогие серверы самых продвинутых конфигураций, а их заказчиком выступали государственные компании. Руководство Sun Microsystems даже пыталась приобрести в собственность конкурента, но переговоры зашли в тупик – собственники Cray соглашались только на продажу всей компании, а Sun нуждалась лишь в серверном производстве. Впрочем, не было бы счастья, да несчастье помогло. Cray через несколько лет оказалась в собственности SGI, а эта компания в производстве серверов на базе SPARC заинтересована не была. Так Sun получила желаемое даже дешевле, чем когда-то хотела.

Возможность объединить компьютеры в сеть стала серьезным аргументом за Sun, особенно у тех, кто покупал много компьютеров для офисов

Солярка и офис

Главной сферой деятельности Sun изначально было производство серверов и рабочих станций. Эта сфера остается профилирующей и самой доходной для компании по сей день. Но главное, что можно отметить в истории Sun в 90-е, это то, что компания стала уделять огромное внимание не только железу, но и программным продуктам.

Основной такой продукт – операционная система Solaris. Изначально на компьютеры компании ставилась доработанная Биллом Джоем BSD, получившая название SunOS. Но в 1991 году появляется отдельное подразделение компании (логично названное SunSoft), главная цель которого – развитие операционной системы. В сентябре 1991 года была выпущена первая версия Solaris, ставшая симбиозом из BSD и AT&T System V. «Солярка» приобрела популярность в научной и технической среде, снискала славу безопасной системы для серверов. Создатели Solaris особенно гордились высокими показа-

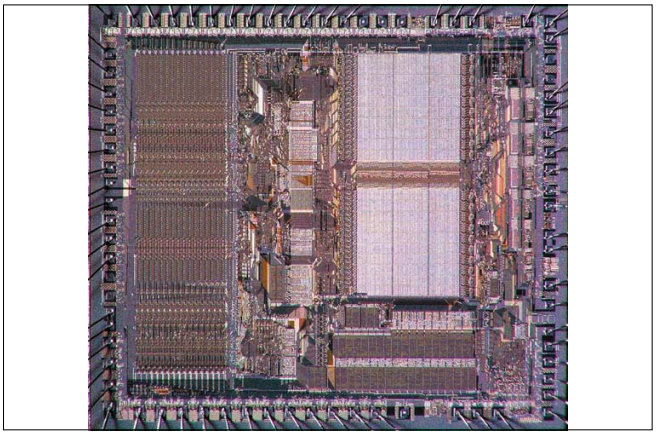


Рисунок 1. Микропроцессор Motorolla MC68010, установленный в первом компьютере компании

Рисунок 2. Рабочая станция Sun Ultra 10 на базе процессора SPARC

телями масштабируемости, что важно при большом объеме вычислений.

Следующим успехом должно было стать создание собственной графической системы. Но, увы, сегодня про Network/extensible Window System (NeWS) никто не вспоминает. NeWS была объективно слабее X Window. Именно последней было суждено воцариться на десктопах *nix-пользователей.

Одним из тех, кто провалил работу над графической системой, был Патрик Нойтон. Он даже собирался уходить из компании, но его настойчиво попросил остаться Макнили.

Во время беседы с Макнили Патрик Нойтон заявил, что в разросшейся Sun стало слишком много бюрократизма, который тормозит развитие компании. В ответ Скотт выделил Патрику нескольких разработчиков и предложил придумать идею нового продукта. Идея нашлась у Билла Джоя, он предложил создать программное обеспечение, которое смогло бы расширить функции бытовой электроники. Для этого традиционные языки программирования, вроде C и C++ не подходили.

Программисты решают создать свой язык. Сначала он получает кодовое название Oak, в честь дуба рядом с офисом. А потом его переименовывают в честь любимой марки кофе программистов Sun – Java. Через несколько лет большинство программ для мобильных телефонов начнут создавать именно на этом языке, его активно будут использовать в веб-разработках.

К концу 90-х Sun стала интересоваться не только серверным и узкоспециализированным рынком, но и проектами для обычных пользователей. В 1999 году происходит покупка компании StarVision, а вместе с ней и офисного пакета StarOffice (см. рис. 3). Исходные коды StarOffice были открыты Sun, хотя сам офисный пакет и продается за деньги. Впрочем, на базе открытых исходников был создан проект OpenOffice, который также долгое время финансировался из средств Sun Microsystems. И если коммерческий успех StarOffice можно оценить как весьма средний, то OpenOffice завоевывает все большую популярность, причем не только в UNIX-среде. По функционалу он мало чем уступает аналогу от Microsoft, зато не стоит и цента – важный аргумент для стран, где решили бороться с пиратством.

Крушение легенды

Такова история успешной компании. Типичная для многих. Собрались однокурсники, что-то придумали, дело пошло, и вот через несколько лет просто хорошая компания превращается в экономическую империю.

Так и шло в Sun Microsystems по нарастающей вверх вплоть до 2001 года. До того, как лопнул «пузырь» дотком. Пресловутый «пузырь» рос с 1995 года – стоимость акций интернет-компаний увеличивалась, в них было престижно и, как тогда казалось, выгодно вкладывать деньги.

Многие проекты получили большой финансовый приток, львиная часть сетевых проектов работали на серверах от Sun. Компании стали закупать все больше продукции. Sun расширяла производство, нанимала новых сотрудников, капитализация росла. Но потом вдруг оказалось, что бизнес в онлайне не приносит мгновенного обогащения. А зачастую не приносит денег вообще. Предприниматели прекратили инвестировать в интернет-компании.

Огромное количество серверов, которое производила к 2001 году Sun, оказалось никому не нужно. Производство пришлось сокращать, а готовую продукцию распродавать за полцены. Как итог – в 2002 году Sun стоила столько же, сколько в 1998 году, ее выручка продолжала падать. Компании предпочитали теперь закупать обычные серверы на базе Intel и AMD, с установленными на них Linux или FreeBSD.

Чтобы переломить негативную тенденцию, Sun заключила контракты с главными производителями процессоров и помимо серверов на SPARC стала делать сервера с Intel Xeon и AMD Opteron внутри. Также Sun приняла решение об открытии исходных кодов своей операционной системы – так в 2005 году увидел свет OpenSolaris.

Неизвестно, сумела бы компания решить свои проблемы и вернуть былые позиции, потому что грянул новый кризис – уже общемировой. От него Sun пострадала больше всех своих конкурентов. Убытки компании с сентября 2008 по июнь 2009 года составили астрономическую сумму, почти два миллиарда долларов! И, судя по всему, скоро Sun уже будет не совсем Sun. В апреле нынешнего года компания Oracle изъявила желание купить Sun Microsystems. Совет директоров последней предложение принял. Сумма сделки должна составить до 7,5 миллиардов долларов. Ожидается, что официально продажа будет оформлена уже осенью.

Что будет с Sun, которая была такой яркой страницей в истории информационных технологий? Как ее изменят новые владельцы? Как бы там ни было, это будет уже другая Sun Microsystems. Ее история только начинается. EOF

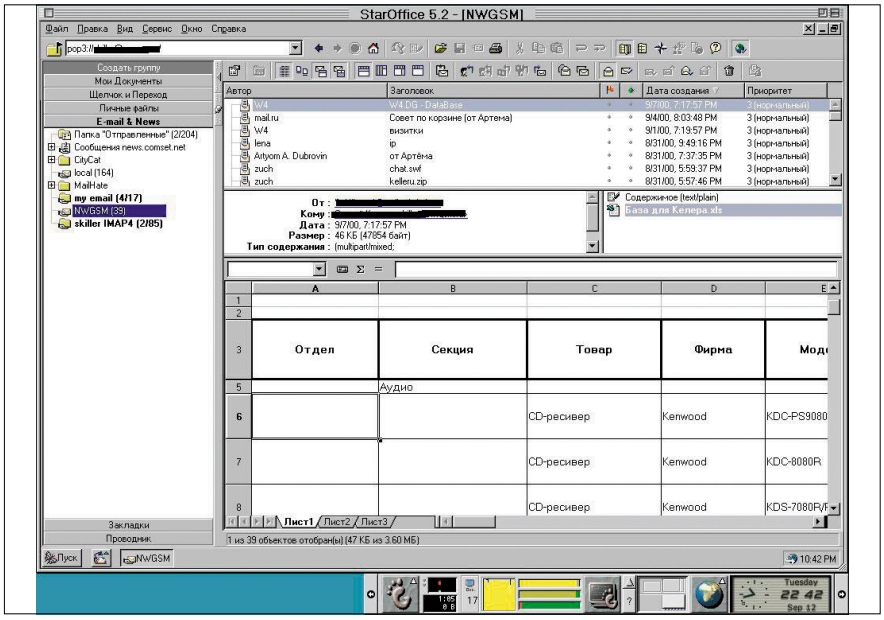


Рисунок 3. Одна из старых версий StarOffice

Алексей Барабанов:
«Кто «хозяин» компьютеров? Сисадмин!»

На вопросы «СА» отвечает наш постоянный автор и читатель Алексей Барабанов

– Как давно вы читаете «СА»?
– Первым в моей коллекции идет октябрьский номер 2002 года, за №1. Достал его, перечитал оглавление и отметил, что за редким исключением почти все статьи актуальны до сих пор. Вообще, я всегда старался читать периодику интересных для меня направлений. Но многие из этих изданий закрылись. Потому когда в мои руки попал «Системный администратор» я был внутренне готов к принятию активной авторской позиции, так как не желал, чтобы этот замечательный журнал повторил историю других.

Хорошо улететь в латинские державы, взять ренткар и поколесить среди «хуанов»

– Помогает ли он в вашей работе?
– Да еще как! Кроме редкой пока аналитики мне очень нравятся статьи Сергея Яремчука, которые буквально на 90% воспринимаются мною как руководство к действию. Так что у меня к журналу корыстные чувства. Скажу по секрету, принимая участие в редактировании новых статей, я иногда ловлю себя на мысли, что непонятно, почему я сам за это не приплачиваю... Шутка, конечно.

– Что вы хотели бы изменить в нем?
– Мне кажется, главное, чтобы журнал увеличил интерактивность. Сильной стороной журнала является его целевая аудитория. Системные администраторы являются грамотной и технически оснащенной частью общества. Журнал в идеале должен издаваться в виде электронного информационного блока в формате web 3. Такой журнал сможет увеличить как печатную, так и рекламную емкость.

– Как вы стали системным администратором?
– Мне очень нравились компьютеры. Первая книга на эту тему, которая разбудила мой интерес, сборник рассказов Айзека Азимова о робототехнике. Прочитав ее, я тривиально пошел в книжный магазин и купил книгу издательства «Связь» о логических элементах ЭВМ: транзисторная логика, диодная логика... Когда появились микропроцессоры, я вдруг понял, что могу обладать собственной ЭВМ. К тому моменту я уже успел соприкоснуться с кибернетическими монстрами своего времени и в ФМШ МГУ, и на спецкурсе московской школы №1140, затем поработал, будучи студентом, на ЕС-совместимой ЭВМ и продолжил работу на той самой СМ4, начальником которой в конце концов и стал. Но главное, я начал собирать самодельные компьютеры! Все! После этого профессия стала призванием – ни о чем ином, кроме компьютеров, я и слышать не хотел. А кто хозяин компьютеров? Сисадмин!

– За что любите свою работу?
– Тридцать лет назад ответил бы: а за то, что я знаю, а больше никто! Двадцать лет назад сказал бы: да за то, что я умею, а больше никто! Десять лет назад признался бы: так за то, что я понимаю, а больше никто! Сейчас скажу: ну, за то, что я замечаю, а больше никто! Еще лет через десять, предположу, придется согласиться с иной редакцией ответа: лишь за то, что я догадываюсь, а больше никто...

– Собираетесь ли расти дальше в своей карьере?
– Нет! Карьера – это «восточный дракон», одолев которого, не только понимаешь, что сам теперь вынужден занять его место, но так же видишь, что это далеко не последний «дракон» впереди, и, самое страшное, осознаешь, что «драконы» теперь и сверху, и снизу! Но я нашел иную формулу роста – стал аутсорсером. Правда, переведя работу в виртуальный режим, однажды обнаружил, что меня поразила болезнь сисадмина – желание захватить ноутбук в отпуск.

– Чем увлекаетесь?
– Самое любимое занятие – улететь куда-нибудь в латинские державы, взять там ренткар и поколесить среди «хуанов» в краю перманентной сиесты: я за рулем, а любимая жена – штурман с картой. И пока отдыхаешь, аптайм серверов все растет и растет... EOF

Редакционнная подписка
для физических лиц

> Вы можете оформить подписку только на **российский адрес.**
> При заполнении квитанции обязательно **разборчиво укажите фамилию, имя, отчество полностью, почтовый индекс и адрес получателя (область, город, улица, номер дома, номер квартиры), контактный телефон.**
> Журнал высылается почтой заказной бандеролью только после поступления денег на расчетный счет и **копия заполненного и оплаченного бланка, отправленная в редакцию по факсу: (495) 628-8253, (доб. 120) или на emil: subscribe@samag.ru**

ИЗВЕЩЕНИЕ

Кассир

ООО "С 13" Форма № ПД-4
ИНН 7708654814/ КПП 770801001
Р. сч. 40702810300080001868 К. сч. 30101810100000000787
ОАО «УРАЛСИБ» г. Москва БИК 044525787
Коды: по ОКПО 84027582, по ОКОПФ 65

Вид платежа: Редакционная подписка на журнал "Системный администратор" за 2010 г.

01	02	03	04	05	06	07	08	09	10	11	12
X	X	X	X	X	X	X	X	X	X	X	X

Дата_____ Сумма платежа: 2400 руб. 00 коп.

Информация о плательщике:

_____
(Ф. И. О. почтовый индекс, адрес и телефон)

_____

_____

Подпись _____

КВИТАНЦИЯ

Кассир

ООО "С 13" Форма № ПД-4
ИНН 7708654814/ КПП 770801001
Р. сч. 40702810300080001868 К. сч. 30101810100000000787
ОАО «УРАЛСИБ» г. Москва БИК 044525787
Коды: по ОКПО 84027582, по ОКОПФ 65

Вид платежа: Редакционная подписка на журнал "Системный администратор" за 2010 г.

01	02	03	04	05	06	07	08	09	10	11	12
X	X	X	X	X	X	X	X	X	X	X	X

Дата_____ Сумма платежа: 2400 руб. 00 коп.

Информация о плательщике:

_____
(Ф. И. О. почтовый индекс, адрес и телефон)

_____

_____

Подпись _____

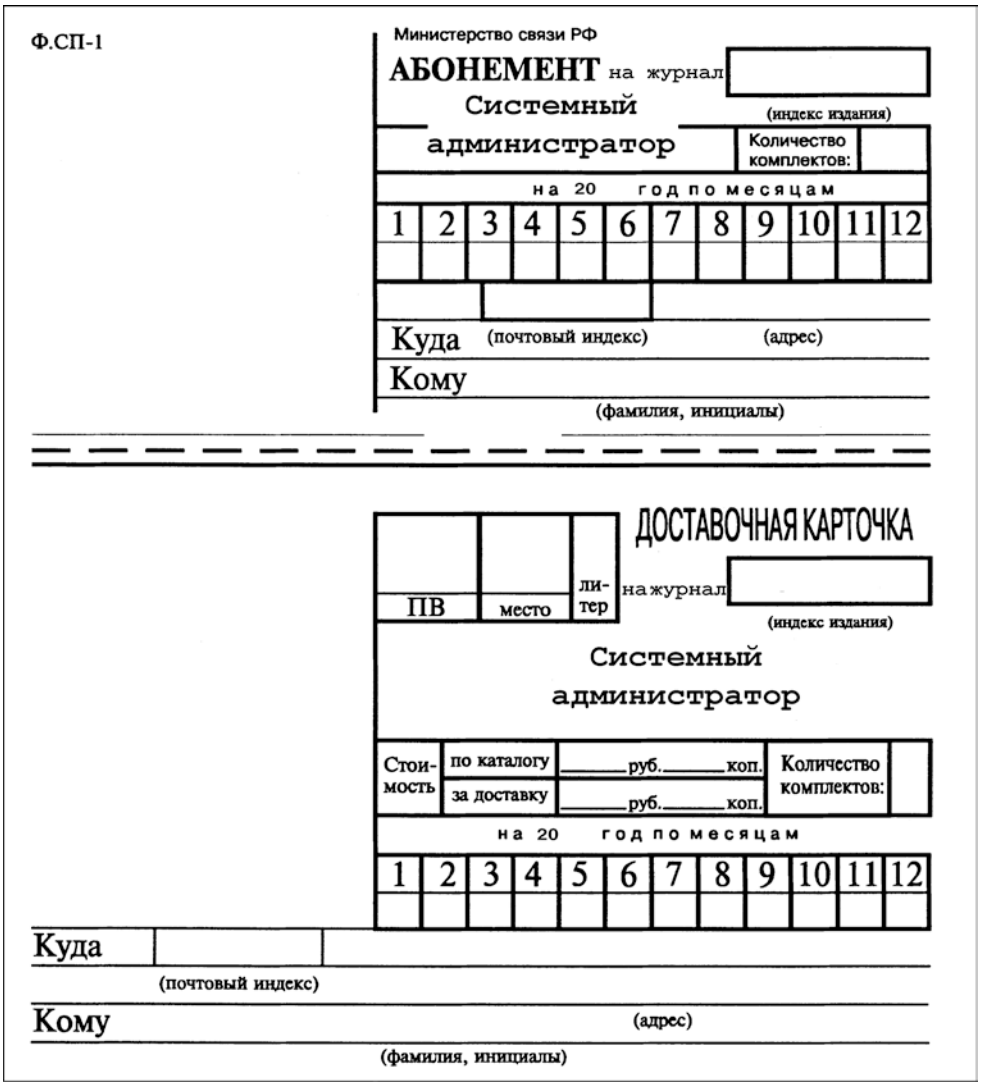
Российская Федерация

> Подписной индекс годовой – **20780**, полугодовой – **81655**
Каталог агенства **«Роспечать»**
> Подписной индекс годовой – **88099**, полугодовой – **87836**
Объединенный каталог **«Пресса России»**
Адресный каталог **«Подписка за рабочим столом»**
Адресный каталог **«Библиотечный каталог»**
> Альтернативная подписка агенства:
«Интер-Почта» (495) 500-00-60, курьерская доставка по Москве
«Вся Пресса» (495) 787-34-47
«Курьер-Прессервис»
«ООО Урал-Пресс» (343) 375-62-74
ЛинуксЦентр www.linuxcenter.ru
> Подписка On-line:
http://www.arzi.ru
http://www.gazety.ru
http://www.presscafe.ru

СНГ

В странах СНГ подписка принимается в почтовых отделениях по национальным каталогам или по списку номенклатуры **«АРЗИ»**:

> **Азербайджан** – по объединенному каталогу российских изданий через предприятие по распространению печати **«Гасид»** (370102, г. Баку, ул. Джавадхана, 21)
> **Казахстан** – по каталогу **«Российская Пресса»** через ОАО **«Казпочта»** и ЗАО **«Евразия пресс»**
> **Беларусь** – по каталогу изданий стран СНГ через РГО **«Белпочта»** (220050, г. Минск, пр-т Ф. Скорины, 10)
> **Узбекистан** – по каталогу российские издания через агентство по распространению печати **«Davriy nashrlar»** (7000029, г. Ташкент, пл. Мустакиллик, 5/3, офис 33)
> **Армения** – по списку номенклатуры **«АРЗИ»** через ГЗАО **«Армпечать»** (375005, г. Ереван, пл. Сасунци Давида, д. 2) и ЗАО **«Контакт-Мамул»** (375002, г. Ереван, ул. Сарьяна, 22)
> **Грузия** – по списку номенклатуры **«АРЗИ»** через АО **«Сакпресса»** (380019, г. Тбилиси, ул. Хошараульская, 29) и АО **«Мацне»** (380060, г. Тбилиси, пр-т Гамсахурдия, 42)
> **Молдавия** – по каталогу через ГП **«Пошта Молдовей»** (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
по списку через ГУП **«Почта Приднестровья»** (MD-3300, г. Тирасполь, ул. Ленина, 17)
по прайс-листу через ООО Агентство **«Editil Periodice»** (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
> **Украина** – Киевский главпочтамт
Подписное агентство **«KSS»**, тел./факс (044)464-0220



Подписные индексы:

20780*
+ диск с архивом статей 2009 года

81655**
без диска

по каталогу агентства «Роспечать»

88099*
+ диск с архивом статей 2009 года

87836**
без диска

по каталогу агентства «Пресса России»

* Годовой
** Полугодовой
*** Диск вкладывается в февральский номер журнала, распространяется только на территории России

Системный администратор

УЧРЕДИТЕЛИ ИЗДАНИЯ
Частные лица

Генеральный директор
Владимир Положевец
Главный редактор
Галина Положевец
chief@samag.ru
Шеф-редактор
Наталья Хвостова
sekretar@samag.ru
Технический директор
Владимир Лукин
Главный редактор электронного приложения «Open Source»
Дмитрий Шурупов
osa@samag.ru
Дизайн-макет
Марина Рязанцева
Дмитрий Бессонов
Иллюстрации
Виктор Чумачев

Над номером работали:
Алексей Барабанов, Александр Емельянов, Олег Щербаков, Кирилл Сухов, Дмитрий Васильев, Александр Тышенко, Андрей Бешков

Реклама и PR-служба
Дарья Зуморина, reklama@samag.ru,
Полина Гвоздь, pr@samag.ru,
тел./факс: (495) 628-82-53 (доб.120)

Распространение
Светлана Зобова
(495) 628-82-53 (доб.120)

Адрес редакции
107045, г. Москва, Ананьевский переулок, дом 4/2, стр.1,
тел./факс: (495) 628-82-53 (доб.120)
Сайт журнала: www.samag.ru

Издатель
ООО «С 13»

Отпечатано в типографии
ООО «Периодика»
Тираж 17000 экз.
Тираж электронной версии 62000 экз.

Вы знаете, как бороться с «Просачивающейся Адварью»? Применяете «Чарующий скрипт»?

Редакция журнала «Системный администратор» представляет вам новый админский сувенир для истинных знатоков своего дела – карточную игру «АУТСОРСЕР».

В ходе игры участники тянут из колоды карты «Проблем», с которым им предстоит бороться один на один или с помощниками, используя подручные средства. Успешное решение «Проблемы» добавляет игроку уровни. Если вы не считаете себя добрым и милым, то для вас в игре предусмотрена специальная возможность – сделать гадость другому участнику и обойти его в погоне за уровнями.

Победителем становится тот, кто быстрее всех доберется до 10 уровня. Остальные подробности об игре, «Чарующем скрипте», «МегаУтилите» и «Клановом коктейле» вы сможете узнать из правил игры.

«АУТСОРСЕР» – это пародия на жизнь, которая позволит вам ощутить всю прелесть аутсорсинга... но без всей словесной мишуры, типа, «утром стулья, вечером деньги...»!

Приобретайте игру «АУТСОРСЕР» в редакции.