Журнал входит в перечень ВАК Минобрнауки РФ

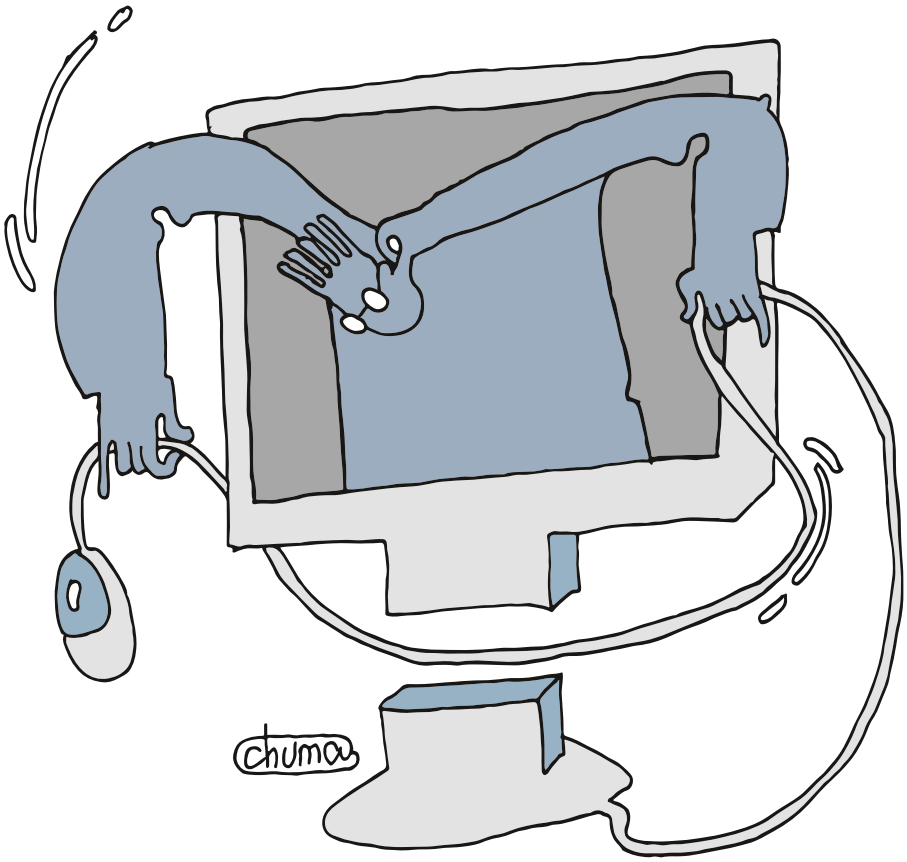
ISSN 1813-5579

9 771813 557005 17004

Системный администратор

ежемесячный журнал www.samag.ru

№04(173)
апрель 2017



Apache POI HSSF
Как «приручить» Excel

Корпоративный Skype
Переход с MS Lync 2013 на Skype for Business

Филиальная сеть
Инструменты техподдержки

Опыт работы в Oracle
с таблицами, содержащими большие LOB-столбцы

Адаптация типовых конфигураций 1С
Применение механизма расширений на практике

Самые частые ошибки
в администрировании межсетевых экранов

Наука и технологии
Представление кодированными деревьями сценариев системы управления интеллектуального здания

Наука и технологии
Создание методов защиты децентрализованных распределенных социальных сетей

Alma mater IT
Андрей Пролетарский: «Информационные технологии – это дорога с двусторонним движением»

16+

Dr.WEB®
с 1992 года

Троянцы-шифровальщики не знают границ

Но и «Доктор Веб» тоже!
Наши специалисты уже помогли пользователям этих стран

Ключ к шифру – в ваших руках

Dr.Web Rescue Pack

Восстановление файлов, зашифрованных троянцем-вымогателем

Закажите расшифровку в «Доктор Веб»

Услуга расшифровки бесплатна
для владельцев действующих коммерческих лицензий Dr.Web Security Space, Dr.Web Enterprise Security Suite (Комплексная защита) и подписчиков услуги «Антивирус Dr.Web» (тарифный пакет Dr.Web Премиум)

Подробнее https://products.drweb.ru/decryption_from_ransomware/

Российский разработчик антивирусов Dr.Web

Опыт разработки с 1992 года

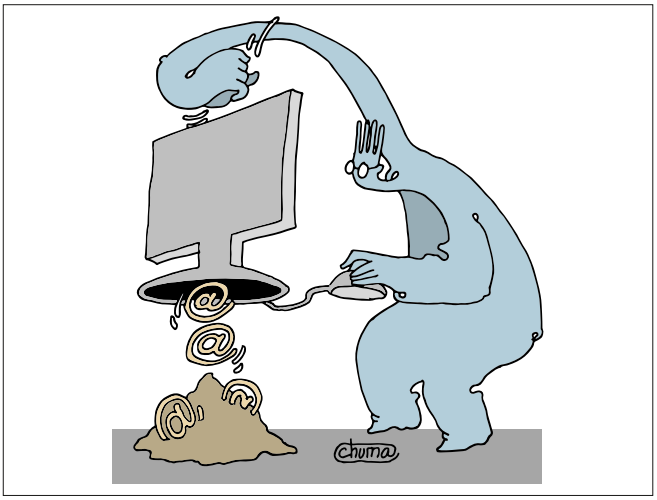
Dr.Web пользуются в 200+ странах мира

Круглосуточная поддержка на русском языке

ООО «Доктор Веб», 125040, Россия, Москва, 3-я улица Ямского поля, вл.2, корп.12А

http://антивирус.рф | https://www.drweb.ru | https://products.drweb.ru/decryption_from_ransomware/

В номере



18

50

ПАО СБЕРБАНК Г. МОСКВА Банк получателя		БИК	044525225
		Сч. №	30101810400000000225
ИНН 9717008803	КПП 771701001	Сч. №	40702810138000086304
ООО "ИЗДАТЕЛЬСКИЙ ДОМ ПОЛОЖЕВЕЦ И ПАРТНЕРЫ" Получатель			

Счет на оплату № 2017СА

Поставщик (Исполнитель): **ООО "ИЗДАТЕЛЬСКИЙ ДОМ ПОЛОЖЕВЕЦ И ПАРТНЕРЫ", ИНН 9717008803, КПП 771701001, 129515, Москва г, Академика Королева ул, дом № Дом 13, корпус Строение 1, квартира Пом. Ii Ком. 63, тел.: 8(499)2771241**

Покупатель (Заказчик):

Основание: **Счет 2017СА**

№	Товары (работы, услуги)	Кол-во	Ед.	Цена	Сумма
1	Журнал "Системный администратор" № 1-2/2017 (январь-февра	1	шт	800,00	800,00
2	Журнал "Системный администратор" № 3/2017 (март)	1	шт	400,00	400,00
3	Журнал "Системный администратор" №4/2017 (апрель)	1	шт	400,00	400,00
4	Журнал "Системный администратор" №5/2017 (май)	1	шт	400,00	400,00
5	Журнал "Системный администратор" №6/2017 (июнь)	1	шт	400,00	400,00
6	Журнал "Системный администратор" № 7-8/2017 (июль-август)	1	шт	800,00	800,00
7	Журнал "Системный администратор" № 9/2017 (сентябрь)	1	шт	400,00	400,00
8	Журнал "Системный администратор" № 10/2017 (октябрь)	1	шт	400,00	400,00
9	Журнал "Системный администратор" № 11/2017 (ноябрь)	1	шт	400,00	400,00
10	Журнал "Системный администратор" № 12/2017 (декабрь)	1	шт	400,00	400,00

Итого: 4 800,00
Без налога (НДС) -
Всего к оплате: 4 800,00

Всего наименований 10, на сумму 4 800,00 руб.
Четыре тысячи восемьсот рублей 00 копеек

Оплата данного счета означает согласие с условиями поставки товара.

Товар отпускается по факту прихода денег на р/с Поставщика.

Руководитель Положевец Г.В. **Бухгалтер** Положевец Г.В.

Общество с ограниченной ответственностью "ИЗДАТЕЛЬСКИЙ ДОМ ПОЛОЖЕВЕЦ И ПАРТНЕРЫ" ИНН 9717008803 КПП 771701001 МОСКВА

Чтобы оформить редакционную подписку на журнал "Системный администратор" 2017 г.:
Шаг 1. Оплатите данный счет.
Шаг.2 Пришлите точный почтовый адрес доставки на e-mail: subscribe@samag.ru с указанием наименования подписчика и реквизитов (если юр. лицо) или Ф,И.О. и точный почтовый адрес доставки (если физ. лицо).
Ваша подписка будет оформлена.
Дополнительная информация на сайте: samag.ru тел. 8 (499) 277-12- 41

С уважением, Издательский дом "Положевец и партнеры".

ПАО СБЕРБАНК Г. МОСКВА Банк получателя		БИК	044525225
		Сч. №	30101810400000000225
ИНН 9717008803	КПП 771701001	Сч. №	40702810138000086304
ООО "ИЗДАТЕЛЬСКИЙ ДОМ ПОЛОЖЕВЕЦ И ПАРТНЕРЫ" Получатель			

Счет на оплату № 2017БИТ

Поставщик (Исполнитель): **ООО "ИЗДАТЕЛЬСКИЙ ДОМ ПОЛОЖЕВЕЦ И ПАРТНЕРЫ", ИНН 9717008803, КПП 771701001, 129515, Москва г, Академика Королева ул, дом № Дом 13, корпус Строение 1, квартира Пом. Ii Ком. 63, тел.: 8(499)2771241**

Покупатель (Заказчик):

Основание: **Счет 2017БИТ**

№	Товары (работы, услуги)	Кол-во	Ед.	Цена	Сумма
1	Журнал «БИТ. Бизнес&Информационные технологии» №1/2017 (февраль)	1	шт	400,00	400,00
2	Журнал «БИТ. Бизнес&Информационные технологии» №2/2017 (март)	1	шт	400,00	400,00
3	Журнал «БИТ. Бизнес&Информационные технологии» №3/2017 (апрель)	1	шт	400,00	400,00
4	Журнал «БИТ. Бизнес&Информационные технологии» №4/2017 (май)	1	шт	400,00	400,00
5	Журнал «БИТ. Бизнес&Информационные технологии» №5/2017 (июнь)	1	шт	400,00	400,00
6	Журнал «БИТ. Бизнес&Информационные технологии» №6/2017 (август)	1	шт	400,00	400,00
7	Журнал «БИТ. Бизнес&Информационные технологии» №7/2017 (сентябрь)	1	шт	400,00	400,00
8	Журнал «БИТ. Бизнес&Информационные технологии» №8/2017 (октябрь)	1	шт	400,00	400,00
9	Журнал «БИТ. Бизнес&Информационные технологии» №9/2017 (ноябрь)	1	шт	400,00	400,00
10	Журнал «БИТ. Бизнес&Информационные технологии» №10/2017 (декабрь)	1	шт	400,00	400,00

Итого: **4 000,00**
Без налога (НДС) **-**
Всего к оплате: **4 000,00**

Всего наименований 10, на сумму 4 000,00 руб.
Четыре тысячи рублей 00 копеек

Оплата данного счета означает согласие с условиями поставки товара.

Товар отпускается по факту прихода денег на р/с Поставщика.

Руководитель Положевец Г.В. **Бухгалтер** Положевец Г.В.

Общество с ограниченной ответственностью ИЗДАТЕЛЬСКИЙ ДОМ ПОЛОЖЕВЕЦ И ПАРТНЕРЫ ИНН 9717008803 КПП 771701001 МОСКВА

Чтобы оформить редакционную подписку на журнал "БИТ.Бизнес&Информационные технологии" 2017 г.:

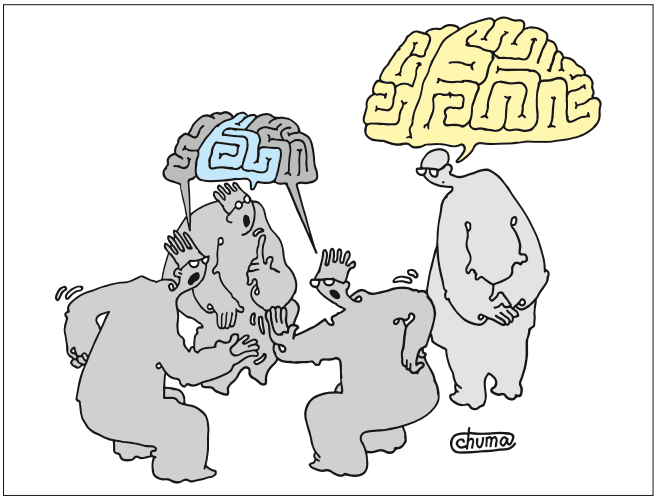
Шаг 1. Оплатите данный счет.

Шаг.2 Пришлите точный почтовый адрес доставки на e-mail: subscribe@samag.ru с указанием наименования подписчика и реквизиты (если юр. лицо) или Ф,И.О. и точный почтовый адрес доставки (если физ. лицо).

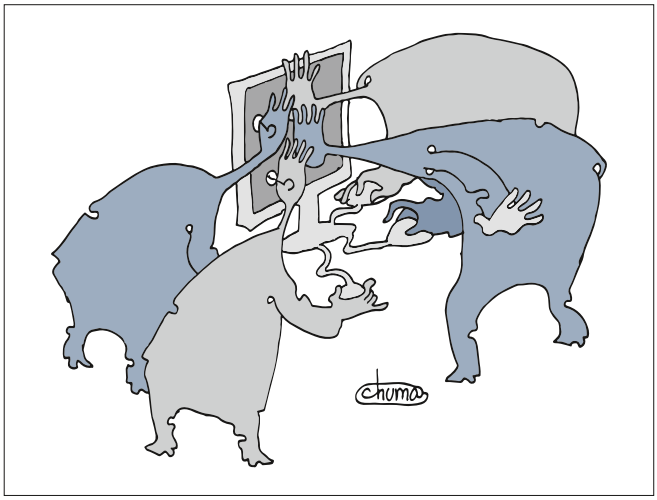
Ваша подписка будет оформлена.

Дополнительная информация на сайте: samag.ru тел. 8 (499) 277-12-41

С уважением, Издательский дом "Положевец и партнеры".



54



66

Прощай, мыщъх!

Николай Лихачев, которого весь айтишный (и не только) мир знает как Криса Касперски, а также мыщъха, ушел из жизни. Ушел безвременно, оставив после себя более 20 книг, множество завершенных и незавершенных ИТ-проектов и признанный авторитет лучшего хакера Земли

«Системный администратор» был первым изданием в России, где Крис из номера в номер печатал свои статьи, сложившиеся позже в книги. И одним из немногих СМИ, которому Касперски, несмотря на свою закрытость и нелюбовь к прессе, никогда не отказывал в интервью – ни в «российский», ни в «американский» период жизни и работы.

Отдавая дань уважения памяти российского обладателя американской визы 01 (для людей с выдающимися способностями, которую дают, к примеру, нобелевским лауреатам), мы публикуем в сокращении некоторые наши архивные материалы.

Романтичный хакер Крис Касперски

«Системный администратор» № 12, 2006 год
http://samag.ru/archive/article/1679

Самородок

Вундеркинд? Сам Касперски так не считает:

– Я смотрю, у многих моих знакомых дети в шесть лет «прогу» уже умеют компилировать. Прогресс... Я же начал тогда, когда появились доступные компьютеры.

Мой первый компьютер – «Правец 8D» – был совершенно несовместим ни с «синклерами», ни с «бэкашками», имевшими хождение в народе, и кучей программного обеспечения. У меня же не было ничего, кроме платы с несколькими микросхемами, и, чтобы вдохнуть в них жизнь, требовалось научиться программировать, это затянуло неожиданно.

Однако после школы Крис пошел учиться на радиофизика. Высшее образование – яркая страница его биографии. Успешно поступив в Таганрогский радиотехнический институт, и, проучившись там три месяца, Крис понял, что он и вуз – вещи несовместимые:

– Я человек неорганизованный, а там много обязаловки. А просто так зубрить, чтобы сдавать, мне не хотелось. Ушел, но на следующий год родители уговорили снова поступать – мол, нельзя ведь без образования остаться. Поступил еще раз, но история повторилась. И в третий раз поступил и опять не выдержал.

Автор

– Последние месяца три у меня такой темп – один день на статью. За один день успеваю статью написать, отредактировать, подготовить картинки. Компоную материал для следующей статьи, после чего падаю замертво на клавиатуру. Утром просыпаюсь и начинаю быстро строчить вторую статью. А еще впереди третья, четвертая, пятая, до бесконечности… А вообще я на статью трачу дня два. Но если серьезная тема, бывает, неделя требуется.

– А что вам нужно, чтобы написать хорошую статью? Вдохновение?

– Не столько вдохновение, сколько материал. Допустим, для «Системного администратора» я писал много статей по файловым системам: NTFS, ext2/3fs, USF. Сначала их исследовал, а потом обобщал в виде статьи. Обычно статье предшествует исследование. Мне нравится писать о том, в чем я разбираюсь, тут я могу принести людям пользу. Любимые темы – дисассемблирование, отладка, защита, взлом. Это мое. Я этим занимаюсь и могу эту тему развить, потому что сам не раз это делал. Имею опыт.

– Самая интересная для вас тема – безопасность?

– Хакерство. Мне интересно заглянуть «под капот» программы, защиты. Тут нетривиальное мышление требуется.

– Вас привлекает тайна?

– Возможность пойти нехоженым путем. Интересно. Смотришь систему защиты даже не ради того, чтобы взломать и бесплатно пользоваться программой, а чтобы себя проверить, выиграть в состязании с машиной. Она меня перехитрит или я ее?

– Вы начали работать на компьютерах, потому что было интересно. А как менялось отношение к информационным технологиям в течение жизни?

– Раньше компьютеры менялись медленно. Я знал о них все. А сейчас – слишком быстро. Мне неинтересно читать о новых железяках. Если ими торгуют, то продавец сам расскажет, если не торгуют, то зачем об этом читать? Раньше процессор расковырял, что-то о нем узнал – и эти знания можно долго использовать, потому что парк компьютеров не обновлялся годами. А теперь все быстро меняется, и смысл детальных раскопок теряется.

– Как вы считаете, какие качества нужны человеку, который решил посвятить жизнь ИТ?

– Во-первых, программист – это инженер, а инженерная профессия – это учет рисков, умение просчитывать, при каких условиях конструкция откажет, когда надо делать дополнительные подпорки, а когда нет. Поэтому хороших программистов не очень много. Нужно иметь инженерное мышление и математическое отчасти. Плюс нужна усидчивость. Приходится читать много документации, на английском языке, постоянно работать над собой. Не бывает так: человек выучился в университете, поступил на работу и может программировать. С другой стороны, все программисты в значительной степени обладают качествами аутистов, т.е. людей, которые стараются не контактировать с окружающими, им интересен только их внутренний мир и дело, которому они посвятили жизнь.

Одиночка

Он честно признается в своих «нелюдимых» качествах, принимает свою природу как данность и бороться с ней не пытается. «Хотя я написал кучу книг, взломал кучу программ, а защитил еще больше, но фактически я ничего не сделал – дерево не посадил, сына не вырастил...»

«Даже когда для других день, я сосредоточен в глубоком колодце своего одиночества, куда редко проникает свет. Ни веревки, ни лестницы внутри колодца нет, а извне о существовании этого самого колодца никто даже и не догадывается, потому что на самом деле никакого колодца нет, есть только сознание, включающее в себя и колодец, и веревку, и лестницу, и чувство глубокого одиночества. Как же это сложно – признаться себе в том, что ты чувствуешь и знаешь, прекратить искать оправдания, а просто принять свою истинную сущность!»

Беседовала Оксана Родионова

Крис Касперски: «Компьютер как средство решения проблем сам стал одной большой проблемой»

«Системный администратор» № 7-8, 2014 год
http://samag.ru/archive/article/2729

– Крис, общеизвестно, вы – образцовый трудоголик. Получается, вся ваша жизнь проходит перед монитором?

– У меня основное время уходит на обдумывание алгоритмов со схематическим изображением квадратиков со стрелочками на бумаге. Вот вам и ответ на вопрос, как не сидеть целый день за компьютером.

Разность ощущений, разумеется, колоссальна. На бумаге значительно меньше знакомест. Даже если брать стандартную консоль (80x25), то нам потребуется целый альбомный лист, а чтобы записать алгоритм в блокноте (не путать с notepad.exe), приходится вспоминать крылатое выражение «словам тесно, а мыслям просторно». Иначе мы будем писать исключительно в write-only-режиме и потом сами не поймем, что это такое и куда оно работает.

Наверное, поэтому я смутно представляю, зачем нужны IDE и рефракторинг, когда есть FAR и – для полного счастья – colorer (впрочем, на Mac я все-таки использую TextMate, но это все же намного ближе к FAR, чем к Visual Studio, тем более что FAR с colorer поддерживает сотни языков, как, впрочем, и TextMate).

– Слагают легенды о вашей верности FAR и его известному плагину, также говорят, что с их помощью вы можете практически все…

– Это не легенды. У меня действительно нет никакого IDE, colorer может не только раскрашивать текст, но и прыгать по парным скобкам, обеспечивает навигацию по функциям (и это еще далеко не все). К тому же он у меня «перепиленный» под себя вдоль и поперек. Вообще у colorer удивительная архитектура, и для расширения функционала даже не обязательно залезать в его исходные тексты.

– Расскажите о вашем рабочем месте и типичном софте. Сколько у вас компьютеров? Каких?

– Сейчас у меня в работе два Mac, восемь «виндовых» лаптопов, стопка «никсовых» серверов, и они укомплектованы обычно следующим образом:

> Под Win: FAR + colorer, HIEW, OllyDbg, IDA-Pro, Python, MS VC.
> Под Mac: TextMate, SynalyzeIt, IDA-Pro, Python, GCC.
> Под Linux: vim, NetBeans, IDA-Pro, Python, GCC.

Мой стандартный комплект влезает на флешку и работает на любой системе, включая *nix и Mac. Компилятор локально не нужен – он стоит за тридевять земель и всегда доступен по ssh. Почему некоторые считают Visual Studio вершиной прогресса, и, кстати, что это за странное слово «рефракторинг»?

Мой вам совет – Festina Lente («поспешай медленно»). Не делай наспех, чтобы потом гарантированно не переделывать. Сначала думай, а потом пиши (программный код), и не пиши заведомо абы как, утешая себя тем, что потом «отрефакторишь». Такой подход формирует привычку, а привычка – это вторая натура.

Впрочем, не нужно зацикливаться на инструментарии и любимых компьютерах. К примеру, я выиграл международный конкурс по обфусцированию JavaScript, сидя при этом в кафе, где у меня с собой был только телефон BlackBerry. Так что не рабочее окружение делает человека. Один хороший знакомый недавно набросал прототип будущей системы и заключил контракт на несколько миллионов долларов на салфетке в таком же прибрежном кафе. Потому что никакой другой бумаги просто не было под рукой. А вот плохому танцору вечно что-то мешает…

– Вы владеете широким спектром языков программирования. Какова разница между низкоуровневым программированием на ассемблере и на высокоуровневых языках?

– Грубо говоря, отличие между языками программирования разных уровней как между тактикой и стратегией. При этом многие люди владеют одним из этих умений и немногие – двумя. Я работаю на низком (тактическом) уровне. На уровне архитектуры движка. Рядом со мной работают стратегические архитекторы, потому как движок без колес и руля никому не интересен.

– Как ассемблерщик и кодокопатель со стажем скажите, есть ли особая романтика у тех ядерных глубин, «куда не ступала нога джависта»?

– Ядро Linux доступно в исходных текстах, исходные тексты ядра Windows сегодня есть уже практически у всех, кому они нужны, потому для этого совершенно необязательно обращаться к дизассемблеру.

Все это кажется волшебством только до тех пор, пока не понимаешь, как оно работает, но, чтобы не понять, нужно очень сильно постараться. Достаточно лишь прочитать Modern Operating Systems by Andrew Tanenbaum и Windows Internals by Mark Russinovich.

– Одна из ваших основных специализаций – анализ вирусов и самого разного malware. Вначале были стелз-вирусы, затем пришла эпоха полиморфов, а что потом?

– ...а потом «замысловатые слова» посыпались как из рога изобилия. Advanced persistent threat (или сокращенно APT) обычно включает в себя сокрытие факта своего присутствия в системе (он же Stealth, он же Root-Kit), активное/пассивное противодействие обнаружению и удалению и т.п.

Полиморфизм – это частный случай метапрограммирования. В computer science под метапрограммированием обычно подразумевают программу, результатом работы которой является другая программа. Пассивные детекторы сканируют файлы в поисках уникальных последовательностей символов. Активные (или как их принято называть проактивные) детекторы работают по принципу поведенческого анализа. Грубо говоря, последовательность вызова API функций – это метрика. Поведенческий анализ распознает определенные сценарии (например, инъекцию кода в доверенный процесс) безотносительно того, как именно они реализованы, и последние несколько лет идут кровопролитные бои за видоизменение поведенческих сценариев до состояния, когда они становятся практически неотличимы от легитимных сценариев популярных программ.

Изменились и угрозы. Если во времена MS-DOS вирусы были «проблемой грязных рук» и не затрагивали тех, кто пользовался лицензионным ПО, то сейчас основная масса вредоносных программ распространяется через документы, эксплуатируя ошибки проектирования.

Дороже всего приходится расплачиваться за ошибки в сетевом стеке. Чтобы подхватить заразу, достаточно всего лишь интернет-подключения, даже браузер запускать необязательно, хотя ошибки в сетевом стеке – большая редкость, и гораздо чаще хакеры проникают через святую троицу – pdf, jar, swf. По умолчанию браузер загружает их автоматически, и, если не установлены обновления, ждите проблем.

– То есть полиморфическим технологиям сейчас переломили хребет?

– Отнюдь. Во времена MS-DOS вирусы включали в себя генератор кода, доступный для анализа. Сейчас же код генерируется удаленно на хакерском сервере и отдается по HTTP-запросу. Или... не отдается. Сервер проверяет IP-источник запроса, и в случае каких-либо подозрений последующие ответы возвращают 404 или чистую страницу. К тому же хакеры обязательно проверяют IP на принадлежность к антивирусным компаниям и разным правительственным лабораториям. Да и сам генератор в любом случае остается недоступен. В лучшем случае вы можете его купить на черном рынке за наличные деньги, но чаще всего такая возможность недоступна, а потому в распоряжении аналитиков есть лишь отдельные экземпляры работы генератора, в которых необходимо выделить неизменную часть, что существенно затрудняет разработку детектора.

К тому же централизованный генератор хакеры могут обновлять так часто, как им вздумается. Прошли времена, когда вирусы работали только под MS-DOS и только под Intel x86. Сейчас необходимо распознавать не только машинный код x86, ARM, PowerPC, не только байт-код (Java, Flash), но и бесчисленное множество скриптовых языков (JavaScript, VBScript, Python). Например, на Mac Python идет предустановленным, что открывает для хакеров новые перспективы. Кстати, Python замечательно распространяется не только в виде скриптов, но и байт-кода.

> Мой вам совет — Festina Lente («поспешай медленно»). Не делай наспех, чтобы потом гарантированно не переделывать. Сначала думай, а потом пиши (программный код), и не пиши заведомо абы как, утешая себя тем, что потом «отрефакторишь»

– И каковы перспективы традиционного автоматического лечения вирусов?

– Автоматическое лечение (удаление троянцев) неуклонно сдает свои позиции, и зачастую оно сводится к переустановке системы. Кроме того, лечение возможно только на endpoints. Типичный IPS в лучшем случае предотвращает атаку, но не в состоянии обезвредить уже атакованные системы, поскольку IPS находится между атакуемым и атакующим.

Вообще сейчас у хакера другой приоритет – любой ценой передать управление на свой код, например, расположенный в файле документа и не рассчитанный на исполнение. Эта новая доминанта содействовала развитию веера новых технологий от NOP Slides до Heap-Spay и Return oriented programming (оно же ROP).

– Как антивирусная индустрия вообще справляется с огромным потоком новых зловредов? Сколько «дохлых тушек», положенных реверсеру на стол, реально обработать за сутки?

– Этим занимаются специально обученные люди и машины, причем машины все более активно вытесняют людей. Все, что можно автоматизировать, давно автоматизировано. Сейчас этих тушек столько, что никаких человеческих ресурсов на них не хватит. В качестве примера устройства этого процесса могу посоветовать интересную презентацию, ищите ее по ключевым словам: Adobe Malware Classifier.

Вообще дизассемблировать каждую тушку зловреда – это все равно, что хватать вражеских солдат по одному

Это говорил Крис Касперски

Первые строки автобиографии на четыре страницы: «Небрежно одетый мыщъх, не обращающий внимание ни на мир, ни на тело, в котором живет, и обитающий исключительно в дебрях машинных кодов и зарослях технических спецификаций».

О первой написанной игровой программе: «Нолик и знак «больше» символизировали рыбку, она бегала по экрану взад-вперед, а в центре был рыбак в виде знака вопроса. Чтобы поймать рыбку, нужно было нажать пробел. После рыбки пошли лабиринты. Здесь нужно было рассчитывать алгоритмы, а это уже математика».

О своих собеседованиях при приеме на работу (в США): «В России на собеседованиях часто пытаются раздавить, любой ценой показать, что ты ничего не понимаешь – чтобы снизить зарплату. Людей там не ценят так, как деньги. Здесь, в США, чаще всего наоборот: если видят, что ты стоящий специалист, в тебя вцепляются мертвецкой хваткой и больше не отпускают, предлагая лучшие условия на рынке и идя во всем навстречу».

О жизни и смерти: «В определенном смысле я никогда не умру, потому что частицы моего Я, мои статьи и книги, разлетелись осколками на века, попав на плодородную почву молодых пытливых умов, изменив ход их бытия, и теперь уже непросто провести границу, где они, а где Я. Во мне живут осколки тех, кто вспыхнул до меня. И так по эстафете. А потому мы приходим к тому, что вначале было слово. Именно слово делает людей бессмертными. Пока кто-то грезит о возможности скопировать свое сознание в компьютер будущего, другие копируют свое сознание посредством письменности».

Цитаты собрал Владимир Гаков

и допрашивать. Оно, конечно, полезно. Добыть языка. Одного. А лучше двух. Но что они могут рассказать? Стратегические планы верховного командования им все равно не известны.

Сегодня зловреды – они уже не сами по себе. Они – пушечное мясо на поле кибервойн, сегодня от них зависит чуть больше, чем ничего. Сейчас важно суметь понять устройство хакерской экосистемы – круговорота машинного кода и наличных денег.

– Вы упомянули о тотальной автоматизации как единственном способе выжить, и я сразу вспомнил о вашем патенте, который получен как раз на тему автоматизации...

– Было время, работал я удаленно. Ну как работал? Анализировал огромное количество сплоитов, причем анализировал медленно, потому что навыка не было. Порядочно устав это делать, я написал программу, которая автоматически сгенерировала другую программу. И вот эта другая программа анализировала сплоиты со скоростью один гигабайт в секунду. Запустил ее и улетел в Берген (Норвегия) на встречу со знакомой немкой, с которой у меня тогда был роман.

И когда дней через десять вернулся, программа уже завершала анализ, но у меня хватило ума никому об этом не говорить и до конца года получать «убитых енотов» автоматом. А за пунктуальность и следование намеченным планам на работе мне еще бонусы давали. В конце концов меня заела совесть, и я выслал результаты машинного анализа одним и очень большим куском. В результате эта фирма надолго встала, и теперь мне же пришлось писать еще одну программу, чтобы автоматизировать труд тех, кто разгребал эти результаты, писал к ним тесты и заносил в базу. Собственно, так я и получил свой первый (и пока единственный) софтверный патент.

– Перейдем непосредственно к вашей специализации – информационной безопасности. Каковы сейчас самые общие тренды в этой области?

– Отвечая коротко – основные «тренды» уже сидят, причем сидеть им долго. Лет двадцать, а то и больше. На помощь антивирусам пришли FBI, CIA, US Secret Service и другие страшные слова. Поэтому сейчас маржа везде падает, а посадки растут.

Самый последний писк моды – в прицел атаки попали встраиваемые устройства. В первую очередь это, конечно, роутеры. Зловредный код в роутере очень сложно обнаружить. А тем временем хакеры нашли способ проникнуть внутрь камер наблюдения, подключенных к Ethernet, например, используя процессорные мощности для майнинга битконов. На очереди умная бытовая техника (например, холодильники), а также атаки на бортовой компьютер автомобиля – это фантастика новой реальности.

– Куда идет современный рынок коммерческих решений в области ИБ? Насколько я знаю, это одно из самых быстрорастущих и популярных направлений ИТ?

– У меня двадцатилетний опыт работы в индустрии безопасности, в том числе и на позиции архитектора. Я хорошо знаю рынок и видел множество примеров успешных начинаний, впрочем, неуспешных примеров было еще больше. Рынок систем безопасности действительно очень быстро растет. И растет он потому, что совсем недавно вирусами занимались школьники, «падонки» и прочие «креативные» личности. Затем персональные компьютеры подключили к банкам, и тут оказалось, что на троянах можно делать деньги.

Рекорд в этом деле – двадцать лет отсидки за шесть доказанных нулей. Накинем еще один нуль за счет недоказанных, но... когда к интернету подключили госучреждения, когда спецслужбы полностью компьютеризировались, внезапно выяснилось, что хакеры – это не просто «оболтусы с Дерибасовской», а угроза национальной безопасности. Сейчас все крупные игроки, ну то есть абсолютно все, купили огромное количество решений безопасности, и на гребне следующей волны пришли системные интеграторы, пытающиеся собрать эту груду разрозненного барахла воедино.

Но и этот гребень уже пошел на спад, а на горизонте маячит новый, третий. В практическом плане это означает: скоро предстоят сделки на миллионы и миллиарды долларов, но «повезет» здесь только тем, кто к этому уже готов и у кого уже есть готовые решения.

Напомню, что в свое время антивирусы для ПК дали рождение многим нынешним компаниям-миллиардерам, возникшим буквально на пустом месте без каких-либо инвестиций. Но это было относительно давно, в девяностых. Впрочем, суть осталась неизменной – большие деньги зарабатывает тот, кто первым предлагает «спасительную» услугу, когда еще никто толком не осознал своих потребностей и необходимостей.

– Можно ли привести примеры пока не заполненных ниш, чтобы наши читатели, молодые и амбициозные специалисты по ИБ, могли увидеть, где же лежит этот новый и такой вожделенный для многих Клондайк?

– Что только ни ломают хакеры сегодня. И если на POS-терминале антивирус еще можно представить (хотя с большим трудом), то, например, на surveillance camera антивирус тупо не встанет, потому что это конструктивно не предусмотрено. Хотя де-факто там скорее всего ARM и портированный Linux.

Такая камера вещает потоковое видео, и там хакеры уже нашли дыры, позволяющие заливать шелл-коды со всеми вытекающими последствиями.

Вот мой личный пример из этой оперы. Недавно я прикупил пару Ethernet-камер для своего дома. С камерами идут аккаунты на сервере их производителя с персональным доменом третьего уровня – заходи себе через браузер, введи пароль и смотри удаленно, что там дома у тебя происходит. Два сервомотора обеспечивают свободу наведения, а ИК-подсветка видит даже в темноте – все было бы хорошо, если бы не было так плохо.

Жизнь показала, что эти камеры оказались дырявые, и в них уже поселился ботнет. Сетевым червям даже мозги напрягать не нужно: ваш домен третьего уровня (точка входа в контрольную панель камеры) – это, грубо говоря, число (в данном случае) очень короткое, а потому все камеры сканируются перебором влет и тут же автоматически взламываются. А вот обнаружить такую атаку затруднительно. Ну то есть не то, чтобы совсем затруднительно... Например, если в камере не включен HTTPS, то шелл-коды ловятся сниффером. А если включен? Мне повезло, что в моем случае производитель сделал фейковый HTTPS (ну практически фейковый – у моей камеры нет ресурсов для шифрования видео, и потому по HTTPS она только пароль с логином передает, а все остальное гонит через HTTP).

Поэтому мне пришлось после работы самолично поковырять такую камеру из-за ее заражения, и я обнаружил, что ботнет откликается на определенные http-запросы к камере. Детектор зараженности, быстро написанный мною на «питоне», укладывается меньше, чем в сотню строк. Если накинуть еще пару сотен, то можно на Squid proxy через ICAP-фильтры давить попытки таких червей проникнуть в камеру, заворачивая их «на юг».

Еще личный пример. Видел в местном магазине микроволновку с Ethernet. По сети она сама выкачивает из интернета время и режимы приготовления тех или иных блюд, используя сканер штрих-кода с упаковки товара. От наших электронщиков слышал, что там при старте прошивка грузится в ПЗУ, распаковываясь в память, и что холодный рестарт, возможно, спасет домохозяек. Но что такое холодный рестарт для микроволновки, особенно в США? Если черви будут атаковать потоково, просто устанешь перезагружаться.

Подведем итог: через несколько лет на рынке бытовой электроники будут миллиарды (!) подобных «умных» устройств, подключенных к интернету.Но известные мировые производители бытовой электроники разбираются в безопасности, как «Тузик в апельсинах» (смотрите два моих личных примера выше). И потому они будут вынуждены выкупать сторонние решения. Все это огромный, только зарождающийся рынок. И он просто гигантский! Поверьте, что рынок ПК в сравнении с ним «нервно курит в сторонке».

– Что лично вас восхищает в современных программировании и ИТ, что заставляет двигаться вперед?

– И вырубает, и восхищает одновременно то, что компьютер как средство решения проблем сам по себе стал одной большой проблемой, и все попытки решения этих проблем лишь порождают новые. Это как на месте срубленной головы Лернейской гидры вырастают две новые.

Беседовал Игорь Савчук

Статьи, опубликованные впервые в журнале «Системный администратор»

Технические:

[1] Глубоководное погружение в чипсет Intel 875P – http://samag.ru/archive/article/144 (№ 6, 2003).
[2] Неявный самоконтроль как средство создания неломаемых защит – http://samag.ru/archive/article/153 (№ 7, 2003).
[3] Восстановление данных с лазерных дисков – http://samag.ru/archive/article/161 (№ 8, 2003).
[4] Могущество кодов Рида-Соломона, или Информация, воскресшая из пепла – http://samag.ru/archive/article/173 (№ 8, 2003).
[5] Искажение TOC как средство борьбы с несанкционированным копированием диска – http://samag.ru/archive/article/178 (№ 9, 2003).
[6] Борьба с вирусами. Опыт контртеррористических операций – http://samag.ru/archive/article/197 (№ 10, 2003).
[7] Полиномиальная арифметика и поля Галуа, или Информация, воскресшая из пепла II – http://samag.ru/archive/article/199 (№ 10, 2003).
[8] Коды Рида-Соломона в практических реализациях, или Информация, воскресшая из пепла III – http://samag.ru/archive/article/211 (№ 11, 2003).
[9] Практические советы по восстановлению системы в боевых условиях – http://samag.ru/archive/article/214 (№ 12, 2003).
[10] Вирусы в UNIX, или Гибель «Титаника» II – http://samag.ru/archive/article/223 (№ 1, 2004).
[11] Жизненный цикл червей – http://samag.ru/archive/article/249 (№ 2, 2004).
[12] Ошибки переполнения буфера извне и изнутри как обобщенный опыт реальных атак – http://samag.ru/archive/article/258 (№ 3, 2004).
[13] Ошибки переполнения буфера извне и изнутри как обобщенный опыт реальных атак. Часть 2 – http://samag.ru/archive/article/266 (№ 4, 2004).
[14] Побег через брандмауэр плюс терминализация всей NT – http://samag.ru/archive/article/285 (№ 5, 2004).
[15] Путь воина – внедрение в pe/coff-файлы – http://samag.ru/archive/article/297 (№ 6, 2004).
[16] Техника внедрения кода в PE-файлы и методы его удаления – http://samag.ru/archive/article/315 (№ 7, 2004).
[17] Разгон и торможение Windows NT – http://samag.ru/archive/article/327 (№ 8, 2004).
[18] Восстановление данных на NTFS-разделах – http://samag.ru/archive/article/342 (№ 9, 2004).
[19] Восстановление данных на NTFS-разделах. Часть 2 – http://samag.ru/archive/article/359 (№ 10, 2004).
[20] Файловая система NTFS извне и изнутри. Часть 1 – http://samag.ru/archive/article/375 (№ 11, 2004).
[21] Файловая система NTFS извне и изнутри. Часть 2 – http://samag.ru/archive/article/395 (№ 12, 2004).
[22] Как защищают программное обеспечение – http://samag.ru/archive/article/404 (№ 1, 2005).
[23] Восстановление NTFS – undelete своими руками – http://samag.ru/archive/article/414 (№ 1, 2005).
[24] Техника оптимизации под Linux – http://samag.ru/archive/article/428 (№ 2, 2005).
[25] Unformat для NTFS – http://samag.ru/archive/article/431 (№ 2, 2005).
[26] Восстановление удаленных файлов под Linux – http://samag.ru/archive/article/440 (№ 3, 2005).
[27] Техника оптимизации под Linux. Часть 2. Ветвления – http://samag.ru/archive/article/449 (№ 3, 2005).
[28] Техника оптимизации под Linux. Часть 3. Оптимизация циклов – http://samag.ru/archive/article/465 (№ 4, 2005).
[29] Восстанавливаем удаленные файлы под BSD – http://samag.ru/archive/article/474 (№ 5, 2005).
[30] Насколько неуязвима ваша беспроводная сеть? – http://samag.ru/archive/article/498 (№ 6, 2005).
[31] Модифицируем BIOS – http://samag.ru/archive/article/501 (№ 6, 2005).
[32] Удаленно управляем BIOS Setup – http://samag.ru/archive/article/518 (№ 7, 2005).
[33] CD, не подвластный копированию – http://samag.ru/archive/article/532 (№ 8, 2005).
[34] Как спасти данные, если отказал жесткий диск – http://samag.ru/archive/article/555 (№ 9, 2005).
[35] Linux/BSD как бастион на пути вирусов – http://samag.ru/archive/article/565 (№ 10, 2005).
[36] Антиотладка: старые приемы на новый лад – http://samag.ru/archive/article/568 (№ 10, 2005).
[37] Судьба shell-кода на системах с неисполняемым стеком – http://samag.ru/archive/article/615 (№ 1, 2006).
[38] Можно ли защититься от переполнения буферов? – http://samag.ru/archive/article/615 (№ 2, 2006).
[39] Генная инженерия на службе распаковки PE-файлов – http://samag.ru/archive/article/673 (№ 5, 2006).
[40] Техника снятия дампа с защищенных приложений – http://samag.ru/archive/article/689 (№ 6, 2006).
[41] Волшебство с паяльником в руках – http://samag.ru/archive/article/693 (№ 6, 2006).
[42] Аудит и дизассемблирование эксплоитов – http://samag.ru/archive/article/721 (№ 8, 2006).
[43] Упаковщики исполняемых файлов в Linux/BSD – http://samag.ru/archive/article/731 (№ 9, 2006).
[44] Как обнаружить malware-программы? Универсальный метод – http://samag.ru/archive/article/734 (№ 9, 2006).
[45] Многоядерные процессоры и проблемы, ими порождаемые, в ОС семейства NT – http://samag.ru/archive/article/1646 (№ 10, 2006).
[46] Ошибки синхронизации открывают большие возможности для хакеров. Каковы механизмы защиты? – http://samag.ru/archive/article/1659 (№ 11, 2006).
[47] Как надо и как не надо защищать веб-контент от кражи – http://samag.ru/archive/article/1681 (№ 12, 2006).
[48] Потоковое аудио/видеовещание с VideoLAN – http://samag.ru/archive/article/2136 (№ 2, 2008).
[49] Поиск malware на Server 2003/XP своими руками – http://samag.ru/archive/article/1590 (№ 3, 2008).
[50] Жучки в электронных письмах – http://samag.ru/archive/article/804 (№ 4, 2008).
[51] Дефекты проектирования Intel Core 2 Duo. Аналитический обзор с точки зрения безопасности – http://samag.ru/archive/article/809 (№ 6, 2008).

Общие:

[52] Рецепты правильного трудоустройства – http://samag.ru/archive/article/187 (№ 9, 2003).
[53] Как зарабатывают на Open Source – http://samag.ru/archive/article/1641 (№ 10, 2006).
[54] Чего ждать от удаленной работы? – http://samag.ru/archive/article/1709 (№ 1, 2007).
[55] Бессистемные заметки о поиске работы за рубежом – http://samag.ru/archive/article/1670 (№ 5, 2008).

Визитка

ВЛАДИМИР СТОЛЯРОВ,
юрисконсульт ИД «Положевец и партнеры», law@samag.ru

Внедрение электронного трудового договора в РФ: плюсы и сложности

На недавнем российском инвестиционном форуме, проходившем в Сочи, Роструд в лице своего руководителя Всеволода Вуколова заявил, что работает совместно с Министерством труда РФ над упрощением заключения трудового договора в электронном виде

В будущем (вполне вероятно, уже через год-полтора) документ можно будет оформить дистанционно, посредством телефона или компьютера. Предоставление возможности оформлять свои трудовые отношения, используя мировую электронную сеть, очень важно в современных условиях. Введение электронного договора, конечно, поможет снизить издержки компаний, вкладывающих большое количество своих средств в документооборот. И, естественно, такая форма оформления трудовых отношений будет удобнее для дистанционных работников. В правовых аспектах данного вопроса мы будем разбираться в нашей новой статье

В первую очередь прорабатывается упрощение заключения трудового договора в электронном виде. Предполагается, что это позволит повлиять на легализацию трудовых отношений. В основном эта тема касается тех, кто работает удаленно, особенно на малых предприятиях. Но и крупным компаниям также электронные трудовые договоры не помешали бы. И в принципе они тоже готовы создавать ресурсы для хранения электронных документов и доступа работников к ним, чтобы человек из другого региона мог спокойно распечатать экземпляр. И вот уже в нынешнем году все эти идеи будут обсуждаться с экспертами и ведомствами.

Роструд, по их собственным словам, уже обладает нужным опытом работы создания ИТ-продукта. По инициативе ведомства был разработан сайт с бесплатной базой вакансий «Работа в России», с помощью которого любой желающий может найти работу и даже пройти предварительное собеседование.

Как заявил президент РФ В.В. Путин, тема цифровой экономики сегодня имеет особую актуальность и есть необходимость запустить системную программу ее развития.

Динамично развивающейся современной экономике все меньше соответствуют такие атрибуты, как бумажный трудовой договор, хранящийся у работодателя и отправляемый в трудовые инспекции по почте традиционным способом. Необходимость изменения рынка труда в направлении его цифровизации (оцифрования) позволит повысить производительность труда, избежать целого ряда административных издержек, повысить роль неформальной занятости. Переход к цифровому формату оформления трудовых отношений можно сравнить с процессом перехода «от счет к компьютерам» в бухгалтерии – «в свое время это тоже многих ставило в тупик, однако уже сейчас все понимают, что это в сотни раз повышает производительность труда».

Переход на электронную форму трудового договора позволит значительно сократить административные издержки, повысить эффективность труда, обеспечить прозрачность трудовых отношений. Особую значимость эта инициатива будет иметь для дистанционных и сезонных работников, уезжающих трудиться на дальние расстояния вахтовым методом, а также для работников, трудоустроенных у физических лиц.

Взаимодействие в электронном виде между работником и работодателем при заключении трудового договора может быть реализовано через специализированную подсистему существующей государственной информационной системы. При этом обмен кадровыми документами может осуществляться с использованием различных сервисов. Таким образом, «умный» трудовой договор позволит обеспечить постоянный автоматический мониторинг начисления заработной платы, а также выплат и взносов в ПФР и ФОМС.

По мнению противников изменений, переход с бумаги «на цифру» должен быть постепенным и аккуратным. В любом случае, этот процесс займет какое-то время. Психологически многие люди, особенно старшего поколения, будут не готовы отказаться от привычного для них бумажного носителя. Исходя из этого было предложено на первых этапах внедрения этой инициативы вести параллельный документооборот – и в электронном, и в привычном для многих бумажном формате. Однако со временем это позволит привыкнуть, доработать систему и в конечном итоге полностью перевести трудовые договоры в электронный вид. Техническая возможность для реализации инициативы существует, однако как это будет сделано с точки зрения законодательных норм, пока неизвестно.

Полноценный законопроект об электронном трудовом договоре будет разработан в 2017 году, для того чтобы вступить в силу уже в 2018-м.

Итак, нормы законодательства, разрешающие использование электронных трудовых договоров и электронных трудовых книжек, могут принять уже в этом году.

По мнению руководства Роструда, в первую очередь допустить возможность заключать трудовые договоры в электронном виде стоит дистанционным и сезонным работникам, а также вахтовым, которых, по разным оценкам, от 5 до 7 млн человек. Ориентировочно дистанционным образом трудятся порядка 5 млн россиян, примерно половина из них, к большому сожалению государственных служб, в частности налоговиков, нелегально. Отчасти это связано со сложностями в оформлении трудовых отношений, поскольку, как отмечают специалисты, пересылка документов миллионам работников обходится работодателям в крупную сумму: от 5 до 10 млрд руб. в год. А приобретение электронных подписей стоит работникам от 3,6 до 5 млрд руб. в год.

В настоящее время закон не запрещает заключать трудовые договоры с использованием электронной подписи, но после этого в любом случае работники и работодатели должны обменяться и бумажными экземплярами договора. Это важный пункт, поскольку в случае возникновения судебного спора между сторонами бумажный документ признается доказательством того, что между сторонами были достигнуты некие договоренности по правам, обязанностям, условиям работы и прочим нюансам.

Инициатива государственных органов подразумевает, что электронный договор будет иметь безусловную юридическую силу. Его обязательность нельзя будет оспорить в суде.

Трудовые споры разрешают районные суды, а они далеко не всегда воспринимают электронные договоры и понимают, как проверить электронную подпись. Так что прежде, чем отказаться от обмена бумажными трудовыми договорами, органам власти придется тщательно проработать этот вопрос.

Основные определения

- **Электронный трудовой договор** – договор, заключенный между работодателем и дистанционным сотрудником не в бумажном, а в электронном виде с использованием усиленной электронной подписи.
- **Дистанционные работники** – лица, заключившие трудовой договор о выполнении работы вне места расположения работодателя (за исключением надомников), в том числе в другой местности, с использованием сети Интернет и других видов связи.
- **Усиленная квалифицированная электронная подпись** – информация, созданная с использованием средств электронной подписи, позволяющая определить лицо, подписавшее документ, и отследить внесенные в него изменения. Ключ проверки такой электронной подписи указан в специальном квалифицированном сертификате.

Так зачем же нужен электронный трудовой договор?

С помощью электронного трудового договора работодатели смогут существенно упростить процедуру оформления сотрудника на работу. Именно это и является основной целью законодателя. Но есть при этом одна существенная оговорка – заключать такой договор можно будет исключительно с сотрудниками, выполняющими работу дистанционно. При этом законопроект исключает из числа таких работников надомников – с ними по-прежнему нужно заключать трудовые договоры в бумажном виде. Целью авторов законопроекта является желание улучшить труд дистанционных работников, предоставить им полный спектр положенных прав и гарантий, а также перевести отношения с ними в правовое поле.

Особенности заключения электронного трудового договора

Порядок заключения электронного трудового договора имеет свои особенности и существенно отличается от привычного:

- при трудоустройстве работник может представить необходимые документы в электронном виде;
- чтобы заверить такой договор, стороны обязаны использовать усиленные квалифицированные электронные подписи (их должен будет получить и оформить работодатель); заключив с сотрудником электронный трудовой договор, работодатель обязан направить ему по почте надлежащим образом заверенную копию этого документа не позднее трех календарных дней;
- ознакомиться с приказом о приеме на работу, правилами внутреннего трудового распорядка, иными локальными актами и коллективным договором сотрудник может в электронном виде. Чтобы подтвердить, что он ознакомился с документом, работник должен поставить на нем усиленную квалифицированную электронную подпись;
- дистанционные работники самостоятельно и по своему усмотрению распределяют свое рабочее время и время отдыха;
- в трудовую книжку по желанию работника можно не вносить сведения о дистанционной работе. В этом случае документами, подтверждающими период дистанционной работы, будут являться копии трудового договора и приказа о прекращении трудового договора.

В чем актуальные преимущества электронного трудового договора?

- **Работодателю гораздо проще оформить трудовые отношения с сотрудником, работающим удаленно** (для примера, существенно снижается бумажная волокита, появляется возможность в считанные минуты оформить сотрудника на работу, не прибегая к помощи почтовых отправлений).
- **Значительно упрощается кадровый документооборот между сторонами** (например, сотрудник может направить документы для трудоустройства в электронном виде и в таком же порядке знакомиться с локальными нормативными актами компании и другими документами).
- **Электронный документооборот позволяет компании снизить затраты на ведение кадрового делопроизводства** (например, это помогает экономить на бумаге и на шкафах, в которых хранятся бумажные документы, освободить офисное пространство и т.д.).

При переходе на электронные трудовые договоры властям необходимо позаботиться о защите личной информации

> **Работнику не нужно тратить время и средства, чтобы посетить офис работодателя и получить подписанный экземпляр трудового соглашения или ознакомиться с приказом.**

> **Сотрудник получает право выбора – поручить работодателю внести изменения в его трудовую книжку об удаленной работе или нет** (таким образом, дистанционным работникам предоставят альтернативу, тогда как для других категорий такую возможность трудовое законодательство не предлагает).

Имеются ли недостатки у электронного трудового договора?

> **Трудовой кодекс не предусматривает возможности заключать электронный трудовой договор.** На данный момент есть только законопроект № 88331-6 «О внесении изменений в Трудовой кодекс Российской Федерации и статью 1 Федерального закона «Об электронной подписи». Поправки гласят, что электронный трудовой договор можно будет заключать только сотрудниками, трудящимися дистанционно.

> **Работники уже привыкли к бумажным документам.** Ведь их в случае трудового спора всегда можно использовать в суде. Порядок же, в соответствии с которым можно будет использовать электронные документы в судах, пока на законодательном уровне не урегулирован. Поэтому подавляющее большинство сотрудников не поддержат инициативу работодателя заключать с ними трудовые договоры в электронном виде.

> **Чтобы подписать электронный трудовой договор, требуется электронная подпись.** При этом такая подпись должна быть как у работника, так и у работодателя. А оформлять ее всем принимаемым на работу работникам довольно сложно. Это связано с тем, что возможность использовать такую подпись предоставляют специализированные операторы, с которыми работодателю нужно будет заключить соответствующий договор.

> **Если договор с работником и будет подписан в электронном виде, то другие кадровые документы останутся на бумажных носителях** (приказы о приеме на работу, о прекращении трудового договора, о переводе и др., листок временной нетрудоспособности, заявления работника и т.д.). Даже если стороны подпишут трудовой договор в электронном виде, это не решит вопрос использования компанией электронных документов в остальных случаях.

> **Если компания собирается заключать с работниками электронные трудовые договоры, ей нужно будет разработать порядок хранения не только бумажных, но и электронных документов, указать особенности их хранения, а также обеспечить безопасность персональных данных, содержащихся в таких трудовых договорах.**

Электронные сервисы

Роструд готовит запуск нового сервиса, который позволит работникам в режиме онлайн проверить, правильно ли был составлен их трудовой договор. Сервис «Проверь трудовой договор!» уже в скором времени будет доступен на портале Онлайнинспекция.рф, а также пользователям сайта Роструда «Работа в России» (trudvsem.ru), где соискатель вакансии сможет проверить проект договора, который ему предлагает подписать работодатель.

Сервис позволит сократить число нарушений в области трудового права. В ходе проверок в 2016 году инспекторы труда выявили порядка 20 тыс. нарушений, которые связаны с оформлением трудовых отношений, что составляет 10% от числа всех выявленных нарушений. В Роструде также подсчитали, что по итогам 2015 года было выявлено 91 тыс. нарушений правил оформления трудовых отношений.

Планируется, что новый онлайн-сервис по проверке договоров заработает уже в этом году. Когда конкретно – информации нет. Как он будет выглядеть, тоже неизвестно. Причем он будет полезен и работодателям, поскольку за ненадлежащее оформление трудового соглашения компании грозит штраф по ст. 5.27 КоАП в размере от 50 000-100 000 руб.

В презентации концепции говорится, что ГИС «Электронный трудовой договор» даст возможность заключать

трудовые договоры в электронном виде при добровольном согласии работника и работодателя. Заключить такой договор можно будет через портал госуслуг. Сведения о заключенных договорах будут вноситься в специальный реестр.

Система также предполагает возможность обмениваться и другими кадровыми документами в электронном виде, но их перечень в презентации не указан. Для заключения таких договоров уже есть IT-инфраструктура: это может быть ЕСИА, портал «Работа в России».

Каковы перспективы нововведений?

В первую очередь это работа без офиса. Практика оказания услуг на расстоянии распространена среди множества профессий: юристов, журналистов, переводчиков, программистов и представителей других специальностей, которые могут выполнять свои задачи удаленно.

Такой формат работы удобен для каждой из сторон: наниматель не беспокоится об обеспечении сотрудника рабочим местом, о компенсации расходов на проезд, проживание и т.п. А дистанционный сотрудник самостоятельно организует режим своего рабочего времени, ориентируясь на достижение главной цели – своевременное и качественное исполнение заданий.

Для заключения трудового контракта в письменном виде человек должен приехать к работодателю для подписания документов или же отправить их по почте. Но на больших расстояниях это занимает много времени и приводит к увеличению издержек и соискателя, и нанимателя. Поэтому в таких ситуациях на практике трудовые договоры зачастую вообще не заключают. И сотрудничество происходит только на основании устных соглашений.

Так работают примерно около половины людей, которые предоставляют свои услуги дистанционно. Это создает определенные риски как для компаний, так и удаленных работников:

- бывает сложно отследить выполнение взятых обязательств каждой из сторон;
- в случае их невыполнения сложно защитить права пострадавшего;
- работник лишен социальных гарантий, которые мог бы получить в случае официального оформления.

Рост числа незарегистрированных или полулегальных сотрудников приводит к сокращению поступления налога в казну и взносов во внебюджетные фонды. А трудовой договор в электронной форме мог бы снять эти проблемы.

Решение проблемы оформления удаленных сотрудников

Регистрация сторон будет происходить на портале государственных услуг (www.gosuslugi.ru) при взаимном согласии каждой. Сначала наниматель оформляет договор в государственной информационной системе (ГИС) «Электронный трудовой договор», а затем работник подписывает файл усиленной квалифицированной электронной подписью при помощи единой системы идентификации и авторизации на этом сайте.

Для учета заключенных контрактов создадут специальный реестр. При необходимости получить заверенный бумажный экземпляр договора можно будет обратиться в МФЦ («Мои документы»).

В перспективе с помощью электронного документооборота будут передавать и другие документы (например, трудовые книжки). Это значительно упростит делопроизводство и кадровый учет.

Предполагается, что создание системы электронных трудовых договоров:

- мотивирует нанимателя к оформлению трудовых отношений с дистанционными специалистами;
- значительно сократит временные и финансовые издержки обеих сторон;
- оформление контрактов через портал государственных услуг приведет к увеличению количества его пользователей.

Чтобы механизм заработал, законодателям придется внести поправки в Трудовой кодекс и Закон об электронной подписи.

Одно из важнейших положений – заключение трудового договора в электронной форме между работником и нанимателем должно быть добровольным.

Некоторые особенности заключения договоров в электронной форме

Люди часто интересуются возможностью заключения электронного договора с дистанционными работниками.

Особенности отношений, возникающих между работодателем и дистанционным работником, рассматриваются в главе 49.1 Трудового кодекса РФ (далее – ТК РФ).

Согласно статье 312.2 ТК РФ трудовой договор о дистанционной работе и соглашения об изменении условий трудового договора могут заключаться путем обмена электронными документами.

В таком случае работодатель не позднее трех календарных дней со дня заключения данного трудового договора **обязан направить дистанционному работнику по почте заказным письмом с уведомлением оформленный надлежащим образом экземпляр данного трудового договора на бумажном носителе.**

Исходя из буквального толкования нормы речь идет именно об экземпляре, а не о копии. При этом нигде законодательно не установлено обязательное использование конкретного вида электронной подписи и оператора электронного документооборота для обмена такими документами. Однако, учитывая важность кадровых документов, мы бы рекомендовали использовать усиленную квалифицированную электронную подпись. Данный вид ЭП не требует заключать дополнительное соглашение об ЭДО между сторонами и позволяет отследить возможное изменение документа после его подписания.

Отметим, что с 01.01.2017 года для заключения трудового договора с работником микропредприятия можно использовать типовую форму договора. Возможно, что в будущем законодатель попытается решить вопрос использования электронной формы кадровых документов.

Насколько возможно использование международных электронных договоров?

Законом РФ не запрещено заключать электронные договоры с международными субъектами. Поэтому основной трудностью будет реализация такого взаимодействия сторон.

Юридическая сила подписанных электронной подписью документов зависит от законодательства стран, в которых зарегистрированы подписавшие его стороны. Согласно положениям Федерального закона «Об электронной подписи» электронная подпись и подписанный ею электронный документ не могут считаться не имеющими юридической силы только на том основании, что сертификат ключа проверки электронной подписи выдан в соответствии с нормами иностранного права (ст. 7).

Стоит напомнить, что иностранное лицо может обратиться в аккредитованный удостоверяющий центр за получением квалифицированного сертификата (ст. 17 Федерального закона «Об электронной подписи»). Однако возможность представить все необходимые для получения сертификата сведения такого лица также спорна.

Из этого следует, что однозначный ответ о возможности международных субъектов права обмениваться электронными документами (и заключать электронные договоры) можно дать только после проведения широкого анализа законодательства тех государств, где зарегистрированы контрагенты.

Возможные проволочки и вопросы, которые могут возникнуть, при оформлении электронного трудового договора

Вопрос: *Будут ли установлены сроки перевода с материального бумажного договора на электронный?*
Ответ: Договоры оформляются и заключаются по взаимному согласию сторон и в той форме, которая удобна обеим сторонам. Следовательно, сроки перехода на электронную форму договора обычно не регламентируются.

Вопрос: *Соглашение об использовании простой электронной подписи должно быть составлено только в бумажном виде и подписано собственноручно?*
Ответ: Оно может быть составлено как на бумаге, так и путем создания электронного документа. В последнем случае его подписывают усиленной квалифицированной электронной подписью (УКЭП).

Вопрос: *Для публикации на сайте госзакупок, каким образом показать, что договор подписан? Сейчас используется скан с подписью. А в случае с электронным договором?*
Ответ: Могут быть различные варианты представления (визуализации) наличия ЭП на договоре:

- В карточке договора на площадке информация об ЭП может быть указана вместе с другими атрибутами договора и его содержанием. При этом площадка должна уметь работать с файлами ЭП.
- Если функциональность площадки не позволяет это сделать, то необходимо представить копию договора в формате PDF со штампом Оператора ЭДО СФ в виде отдельного файла. В штампе содержится исчерпывающая информация о подписантах, дате подписания и идентификаторе документа в сервисе обмена.

Вопрос: *При обмене документами в формате .doc, .docx и использовании квалифицированной цифровой подписи защищенность остается на том же уровне, как и с .pdf?*
Ответ: Да, если электронный документ подписан усиленной квалифицированной электронной подписью, то всегда можно узнать о внесении в него изменений вне зависимости от формата документа. В случае внесения изменений в документ, подписанный УКЭП, подпись будет недействительна.

При переходе на электронные трудовые договоры властям необходимо позаботиться о защите личной информации. Электронный оборот – это вещь очень удобная и оперативная. Но здесь необходимо разработать жесткие нормативные требования касаемо тайны хранения личных данных.

Заключение трудового договора в электронной форме между работником и нанимателем должно быть добровольным

Можно ли будет оформлять кадровые документы посредством ЭЦП?

На Федеральном портале по проектам нормативных правовых актов вынесен в обсуждение новый законопроект. Он направлен на урегулирование норм использования документации при оформлении кадров в электронной форме.

По этому законопроекту использование электронной подписи при оформлении кадровых документов теперь допускается. Но работодатели в таком случае должны соблюдать ряд различных условий:

- работодателем принят локальный нормативный акт, устанавливающий порядок использования электронной подписи и случаи признания электронных документов, подписанных электронной подписью, равнозначными документам на бумажных носителях;
- работник ознакомлен с данным локальным нормативным актом под роспись;
- работодатель принял организационно-технические меры для использования электронной подписи, в том числе: установлен порядок проверки электронной подписи, хранения электронных документов, обеспечивается целостность и неизменность документов, подписанных электронной подписью.

В случае несогласия работника с порядком использования электронной подписи, установленным локальным нормативным актом, по письменному заявлению работника использование электронной подписи в трудовых отношениях с данным работником не применяется.

В целях недопущения нарушения прав работников предлагается установить, что не допускается использование электронной подписи при оформлении документов, связанных с приемом на работу, заключением, изменением, прекращением трудового договора, документов, связанных с материальной и дисциплинарной ответственностью работника, документов, на основании которых производятся удержания из заработной платы работника, а также в иных случаях, прямо предусмотренных трудовым законодательством и иными нормативными правовыми актами. EOF

Визитка

ДЕНИС СИЛАКОВ,
к.ф.-м.н., старший системный архитектор Virtuozzo, dsilakov@virtuozzo.com

Virtuozzo PowerPanel

Веб-интерфейс управления ВМ для пользователей

Virtuozzo Automator, предоставляющий веб-интерфейс для управления парком машин, рассчитан в первую очередь на обслуживающих эти машины системных администраторов, но что если необходимо предоставить пользователям возможность управлять своими контейнерами и ВМ через веб?

Подобная задача актуальна, например, для хостинг-провайдеров, строящих свои решения на системах виртуализации Virtuozzo и желающих предоставлять клиентам удобный интерфейс к их VPS. Конечно, гибкая система пользователей и ролей Virtuozzo Automator (VA) [1] позволяет так настроить права пользователей, чтобы им был доступен только ограниченный набор действий и только над их собственными системами, но такой подход напоминает стрельбу из пушки по воробьям – львиная доля функционала VA при этом останется невостребованной.

К тому же вряд ли провайдеры сочтут разумным с точки зрения безопасности выставлять «наружу» веб-интерфейс Automator – несмотря на встроенные ограничения прав для внешних пользователей, всегда есть опасность эскалации привилегий и не обязательно за счет ошибок ПО. Надежнее не иметь даже гипотетического шанса допустить сторонних лиц к внутренней инфраструктуре через интерфейс ее управления. Поэтому в линейку Virtuozzo 7 включен отдельный продукт, ориентированный исключительно на пользователей – Virtuozzo PowerPanel (сокращенно PP). Именно его мы и рассмотрим в данной статье.

Архитектура

Основой PowerPanel является контроллер, устанавливаемый на отдельную машину (реальную либо виртуальную) под управлением Virtuozzo Linux 7 либо CentOS 7. Контроллер осуществляет управление серверами, на которых установлен Virtuozzo; в терминах PowerPanel такие машины называются вычислительными узлами, compute nodes.

На контроллере находится база данных со сведениями обо всех вычислительных узлах, расположенных на них виртуальных окружениях, а также о пользователях PP. На каждом из вычислительных узлов необходимо установить и запустить сервис vzapi-compute, фактически являющийся «агентом» PowerPanel.

Обязательным требованием как к контроллеру, так и к вычислительным узлам является наличие у каждого полноценного доменного имени (fully qualified domain name, FQDN), а также их согласованность по времени (рекомендуется на всех машинах включить NTP). Помните об этом, если решите поэкспериментировать с PP в «песочнице».

Установка контроллера

Для установки контроллера в CentOS 7 необходимо отключить SELinux (выставить SELINUX=disabled в файле /etc/sysconfig/selinux) и перезагрузить систему. В Virtuozzo Linux этого делать не требуется, а в будущем разработчики обещают «подружить» PP и SELinux и в CentOS.

Также в CentOS 7 необходимо импортировать ключ, которым подписаны RPM-пакеты Virtuozzo (в Virtuozzo Linux этот ключ присутствует по умолчанию):

```
# rpm --import http://repo.virtuozzo.com/vzlinux/security/ ↵
        VIRTUOZZO_GPG_KEY
```

После чего необходимо установить пакеты pp-release и vzapi-installer:

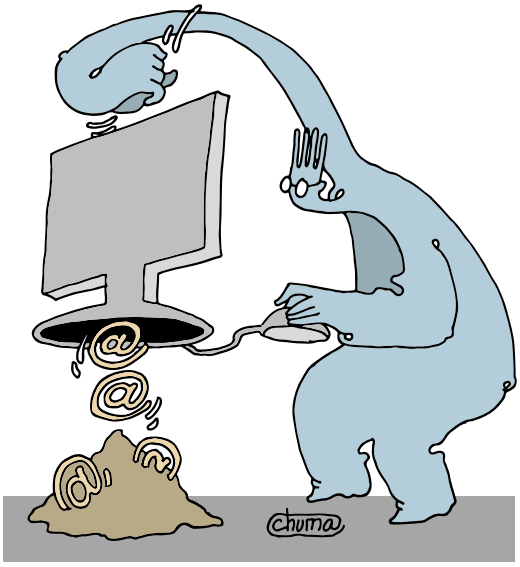
```
# yum install http://repo.virtuozzo.com/pp/releases/2.0/ ↵
        x86_64/os/Packages/p/pp-release-2.0-3.vl7.noarch.rpm
# yum install vzapi-installer
```

Первый пакет добавит в систему конфигурационные файлы с репозиториями PowerPanel, а второй поставит программу установки vzapi-installer.

Установка контроллера запускается следующей командой:

```
# vzapi-installer controller --ask-passwd ↵
        --private-ip <IP_address>
```

Опция --private-ip служит для указания адреса сетевого интерфейса, через который контроллер будет общаться с вычислительными узлами. Эта опция предусмотрена для конфигурации, в которой у сервера есть несколько сетевых интерфейсов, смотрящих в разные подсети. Использование опции --private-ip указывает компонентам PowerPanel (в частности, серверу СУБД MariaDB и системе обмена



Virtuozzo PowerPanel позволяет дать пользователям удобные средства управления их виртуальными окружениями

сообщениями RabbitMQ) слушать только запросы с заданного интерфейса и игнорировать остальные.

Таким образом исключается умышленное или непреднамеренное обращение к PP от машин, которые по определению таких обращений делать не должны. Опция --private-ip является обязательной; если у вашей машины только один сетевой интерфейс, то укажите его адрес.

Опция --ask-passwd говорит программе установки, что пароль администратора PowerPanel необходимо запросить в процессе развертывания контроллера. В случае вызова установщика из других программ либо скриптов можно воспользоваться альтернативной опцией -p, при указании которой пароль считывается со стандартного входа программы без каких-либо вопросов пользователю.

Пароль администратора сохраняется в файле /var/lib/vzapi-installer/group_vars/all. Здесь же хранятся автоматически сгенерированные пароли для внутреннего общения различных компонентов PP между собой.

При обнаружении работающего сервиса firewalld vzapi-installer открывает необходимые для функционирования PP порты: 80, 443, 3306, 5671, 6556, 6557, 35357. Если в момент установки у вас не было брандмауэра, а потом вы решите его включить, то не забудьте открыть указанные порты.

Все общение между машинами в кластере PowerPanel осуществляется по зашифрованным каналам (в частности, по HTTPS). Сертификаты, используемые для настройки такого общения, по умолчанию генерируются установщиком при развертывании контроллера и сохраняются в директории /var/lib/vzapi/. Эти сертификаты являются самоподписанными, поэтому для беспроблемного доступа к веб-интерфейсу PP из современных браузеров необходимо добавить в браузер сертификат /var/lib/vzapi/vzapi_rootCA.crt с контроллера. Сгенерированный сертификат удобен при тестировании, однако в реальной жизни рекомендуется передать установщику настоящие сертификаты и ключ с помощью опций --ssl-cert-file и --ssl-key-file.

Если установка пройдет успешно, то в конце vzapi-installer сообщит, по какому адресу развернут веб-интерфейс PowerPanel.

Из предварительных настроек стоит обратить внимание на параметры создания резервных копий. По умолчанию каждому пользователю разрешается создавать не более трех резервных копий каждой ВМ или контейнера. Этот параметр можно изменить в файле /etc/vzapi/vzapi.conf (опция backup_limit), но помните, что это изменение будет работать только для вновь добавляемых виртуальных окружений. Поэтому если вы изначально хотите задать другое значение этого параметра, то сделайте это после установки контроллера, но до подключения вычислительных узлов.

Еще одним немаловажным параметром является время бездействия, по прошествии которого закрывается сессия пользователя в веб-интерфейсе PP. По умолчанию это время составляет 24 часа. Изменить его можно в файле /etc/keystone/keystone.conf, параметр expiration. Обратите внимание, что время задается в секундах.

Подключение вычислительных узлов

Рекомендуемым способом установки компонентов PP на вычислительные узлы является запуск vzapi-installer на контроллере с указанием списка машин под управлением Virtuozzo 7, которые необходимо контролировать.

На вход vzapi-installer необходимо подать файл со списком машин и указанием пользователя и пароля для доступа к каждой из них. Каждая строка файла (назовем его nodes.lst) должна иметь вид:

```
<user>:<password>@<node_hostname>
```

Развертывание PP подразумевает выполнение различных действий, требующих привилегий суперпользователя, поэтому рекомендуется в этом файле указывать пользователя root.

Процесс установки на контроллере также необходимо запускать с правами суперпользователя.

Поскольку в файле лежат сведения о доступе к другим машинам, в целях безопасности рекомендуется сделать его владельцем пользователя root и запретить другим пользователям доступ даже на чтение:

```
# chown root nodes.lst
# chmod 600 nodes.lst
```

Убедившись в безопасности данных, можно запускать подключение вычислительных узлов:

```
# vzapi-installer computes --nodes=nodes.lst
```

Программа vzapi-installer автоматически сгенерирует новую пару ключей для доступа по SSH (ключи хранятся на контроллере в директории /var/lib/vzapi) и разложит публичный ключ на целевые машины. Если у вас уже настроен доступ по SSH с помощью ключей, вы можете отменить их генерацию с помощью опции --skip-key-deploy. Пароли в файле nodes.lst в этом случае указывать не надо; вместо них необходимо указать парольную фразу ключей, если они есть.

Разложив ключи по удаленным машинам, vzapi-installer запускает установку необходимых программных компонентов с помощью Ansible. Все добавленные вычислительные узлы при этом сохраняются в файле /var/lib/vzapi-installer/inventory.json, в секции computes. Если в будущем заходите переустановить либо обновить компоненты PowerPanel на этих машинах, достаточно запустить команду vzapi-installer computes без аргументов.

При вызове с опцией --nodes vzapi-installer удалит всю имеющуюся информацию из файла inventory.json и заменит ее новой (на основе нового файла). Так что если планируете добавлять новые серверы в придачу к уже имеющимся, то сохраняйте файл nodes.lst, добавляйте в него серверы по мере необходимости и перезапускайте vzapi-installer. Для машин, на которых уже стоят компоненты PP, никакие действия осуществляться не будут.

Запустить установку компонентов вычислительного узла можно и с самого узла, установив на нем vzapi-installer (так же, как это было сделано на контроллере) и запустив установку с явным указанием адреса контроллера:

```
# vzapi-installer computes --controller-node ↵
        <controller_address>
```

Однако учтите, что при таком подходе информация об узле не будет добавлена в файл /var/lib/vzapi-installer/inventory.json на контроллере. Соответственно, последующие вызовы vzapi-installer с контроллера никак этот узел не затронут.

Удаление вычислительного узла

Для удаления машины из пула, контролируемого контроллером, необходимо сначала остановить на этой машине сервис vzapi-compute и удалить соответствующий пакет:

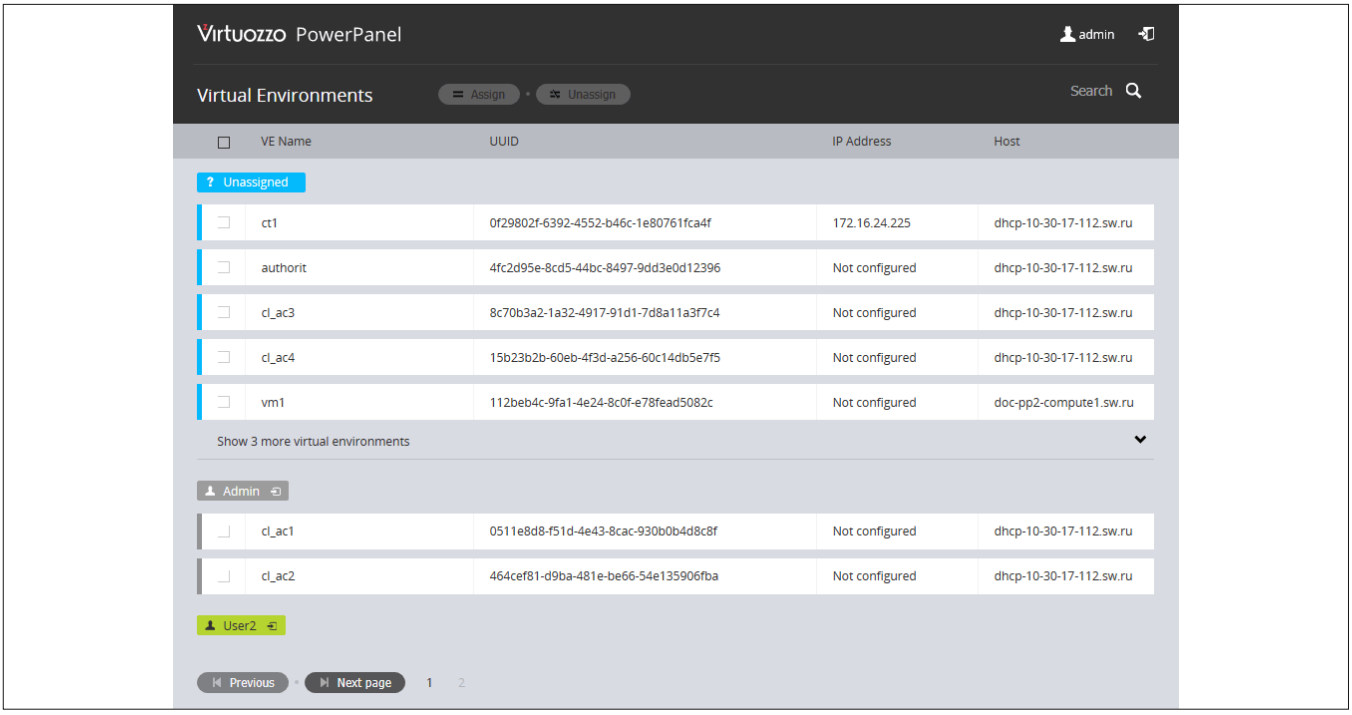
```
# systemctl stop vzapi-compute
# systemctl disable vzapi-compute
# yum remove vzapi-compute
```

После чего необходимо удалить всю информацию об узле из базы данных PP на контроллере. Для этого нужно сначала узнать идентификатор узла в базе, найдя его в выводе команды vzapi host list:

```
# vzapi host list
```

```
<...>
"hostname": "compute4.example.com",
"id": "6ddbb2c1632a42ce9d2d8a83a7c93e04"
<...>
```

Рисунок 1. Веб-интерфейс администратора PowerPanel позволяет для каждого пользователя указать список виртуальных окружений, которыми он может управлять



А затем передать идентификатор команде host delete:

```
# vzapi host delete 6ddbb2c1632a42ce9d2d8a83a7c93e04
```

Также рекомендуется удалить информацию об узле из файлов /var/lib/vzapi-installer/inventory.json и nodes.lst, чтобы случайно не подключить его снова при подключении новых машин.

Обратите внимание, что если вы просто удалите информацию из базы, не отключив сервис vzapi-compute, то при первом же перезапуске сервиса вся информация будет снова добавлена в базу.

Обновление на новый релиз

Переход на новую версию PowerPanel – процесс более сложный, чем просто обновление пакетов. Поэтому при выходе новой версии ее пакеты кладутся в отдельный репозиторий (со своим пакетом pp-release) и не устанавливаются во время штатных обновлений ОС.

Для выполнения обновления в vzapi-installer предусмотрена отдельная команда upgrade. В качестве аргумента этой команде необходимо передать ссылку на пакет pp-release, соответствующий версии PowerPanel, на которую вы собираетесь перейти (информацию о последней версии всегда можно найти на сайте документации Virtuozzo [2]):

```
# vzapi-installer upgrade <upgrade_package>
```

В ходе этого процесса сервисы PowerPanel будут на некоторое время остановлены, что приведет к ее недоступности для конечных пользователей, однако никаких действий со стороны системного администратора не понадобится – все будет сделано автоматически.

Регистрация пользователей

Контроллер PP может быть настроен на использование имеющейся базы LDAP для идентификации пользователей. Для управления данными о пользователях внутри PP применяется OpenStack Keystone, так что интеграция PP с LDAP сводится к интеграции LDAP и Keystone. Подробное описание этого процесса можно найти в документации Openstack [3], здесь же мы сфокусируемся на специфичных для PowerPanel деталях.

Во-первых, в базе LDAP необходимо завести пользователей admin и vzapi с идентификаторами, используемыми в Keystone. Получить идентификатор пользователя admin можно с помощью следующей команды:

```
# openstack --os-cloud local user show admin
```

в выводе которой должна содержаться строка следующего вида:

```
| id | 86921a8ec6a5497895ca07c5d6b738af |
```

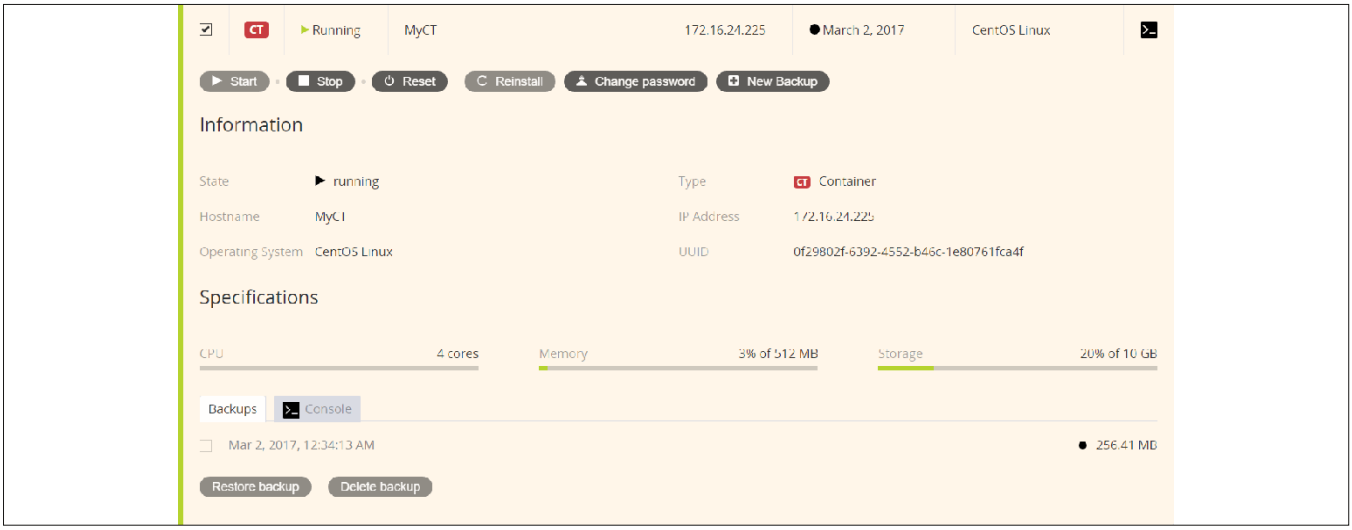
Обозначим полученный идентификатор как <admin_id>. То же самое проделаем для пользователя vzapi, получив его идентификатор <vzapi_id>. После этого необходимо сгенерировать хеш-суммы паролей этих пользователей с помощью команды slappasswd, которая в ответ на ввод пароля должна выдать строку, начинающуюся на {SSHA}. Обозначим полученные хеш-суммы для наших пользователей как <admin_hash> и <vzapi_hash>.

Теперь создадим файл users.ldif следующему шаблону:

```
dn: cn=<vzapi_id>,ou=<PP_users_OU>,dc=<domain>
objectClass: person
cn: <vzapi_id>
sn: vzapi
userPassword: <vzapi_hash>

dn: cn=<admin_id>,ou=<PP_users_OU>,dc=<domain>
objectClass: person
cn: <admin_id>
sn: admin
userPassword: <admin_hash>
```

Здесь <PP_users_OU> – организационная единица (organizational unit) со списком пользователей, которым необходимо дать доступ к PowerPanel, а <domain> – ваш

Рисунок 2. При клике на ВМ или контейнер можно получить детальную информацию и выполнить одно из стандартных действий



контроллер домена (напомним, что формат этой записи для домена типа my.ldap.ru будет иметь вид dc=my,dc=ldap, dc=ru).

Созданный файл необходимо передать утилите ldapadd для добавления необходимых записей в базу LDAP:

```
# ldapadd -x -D cn=Manager,dc=<domain> -W -f users.ldif
```

Пароль пользователя vzapi необходимо дополнительно прописать на контроллере в секции keystone_authtoken файла /etc/vzapi/vzapi.conf.

Также на контроллере необходимо информацию о сервере LDAP внести в секцию ldap файла /etc/keystone/keystone.conf:

```
[ldap]
url = ldap://<ldap_server_address>
user = cn=Manager,dc=<domain>
password = <ldap_admin_password>
suffix = dc=<domain>
user_tree_dn = ou=<PP_users_OU>,dc=<domain>
user_objectclass = person
```

Здесь же в секции identity необходимо изменить значение параметра driver с sql на ldap.

После этого необходимо перезапустить сервис Apache:

```
# systemctl restart httpd
```

а также запустить синхронизацию данных о наших пользователях:

```
# vzapi user sync
```

Синхронизацию следует выполнять после каждой операции добавления или удаления пользователей.

Убедиться, что Keystone видит пользователей LDAP, можно по их наличию в выводе следующей команды:

```
# openstack --os-cloud local user list
```

Альтернативой LDAP является явное создание пользователей с помощью команды vzapi user create, принимающей имя пользователя в качестве аргумента и интерактивно запрашивающей его пароль.

Пользователей LDAP с помощью этой команды создавать нельзя, а если в будущем вы решите переключиться на LDAP, то уже созданные локально пользователи потеряют возможность заходить в PP.

Режимы веб-интерфейса

Virtuozzo PowerPanel имеет два режима работы – администратора и пользователя. В режиме администратора производится ассоциация пользователей и виртуальных окружений, а в режиме пользователя производится непосредственное управление этими окружениями.

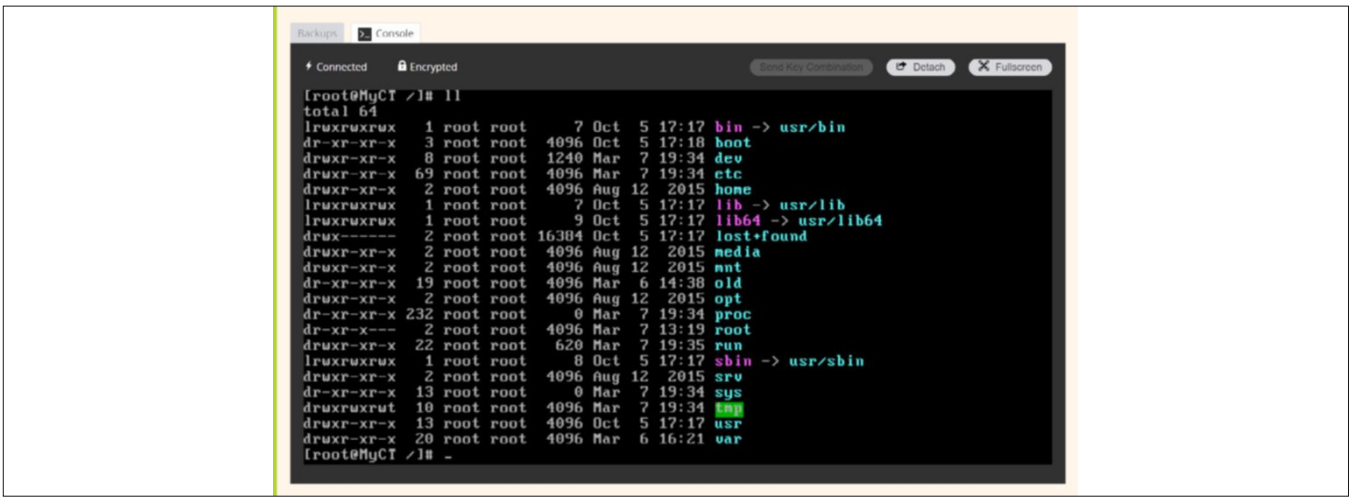
Режим администратора

Основная задача администратора PP в веб-интерфейсе – это прикрепить виртуальные окружения к пользователям, которым разрешено этими окружениями управлять. Для работы в этом режиме необходимо войти в веб-интерфейс под пользователем admin. После входа администратору предоставляется список всех ВМ и контейнеров, находящихся на зарегистрированных в PP вычислительных узлах (см. рис. 1).

При клике на каждую строку можно получить детальную информацию о виртуальном окружении (тип, имя, IP-адрес, конфигурацию оборудования и прочее). Здесь же можно выбрать пользователя, которому будет разрешено управлять данным окружением. Если вам надо отдать одному пользователю сразу несколько окружений, то вы можете выделить их галочками в главном окне и нажать кнопку «Assign» в его верхней части. Отвязать окружение от пользователя можно кнопкой «Unassign».

Администратор может кликнуть на имени пользователя в главном окне и перейти в режим просмотра веб-интерфейса от лица этого пользователя. Для возврата в режим администратора необходимо нажать «Back to Admin» в правом верхнем углу.

Рисунок 3. Вкладка Console отображает ровно то, что обещает, – консоль с полноценным терминалом



Режим пользователя

Конечные пользователи с помощью PowerPanel могут осуществлять следующие действия со своими виртуальными сущностями:

- запускать, останавливать и перезагружать ВМ и контейнеры (под перезагрузкой понимается «жесткая» перезагрузка, равноценная нажатию кнопки «Reset» на физической машине);
- переустанавливать контейнеры (возвращать в исходное состояние со свежеустановленной ОС, но без потери данных пользователя; подробнее этот процесс описан в статье про Virtuozzo Automator [1]);
- изменять пароли пользователей виртуальных окружений;
- управлять резервными копиями;
- подключаться к ВМ и контейнерам по VNC.

Интерфейс PowerPanel для пользователей доступен по тому же адресу, что и для администратора. Зайдя, пользователь увидит все свои виртуальные окружения (см. рис. 2).

Кнопки над списком подконтрольных пользователю объектов используются для действий, которые можно производить с одним либо сразу с несколькими окружениями, – запуск, остановка, перезагрузка (hard reset – как нажатие кнопки «Reset») и создание резервных копий.

Аналогично режиму администратора при клике на конкретное окружение появляется более детальная информация о нем, а также кнопки, позволяющие производить различные манипуляции (запуск, остановку и прочее) с данным окружением. На вкладке Backups можно управлять резервными копиями, а на вкладке Console – подключаться к консоли работающей ВМ или контейнера по VNC (см. рис. 3).

• • •

Virtuozzo PowerPanel позволяет системному администратору дать пользователям удобные средства управления их виртуальными окружениями, не пуская их при этом на внутреннюю кухню серверной инфраструктуры. Инструментарий PP предназначен прежде всего для хостинг-провайдеров, желающих предоставить своим клиентам возможность самостоятельно управлять купленными VPS через веб-интерфейс, но может быть полезна и в других сценариях использования Virtuozzo 7. EOF

[1] Силаков Д. Virtuozzo Automator. Управляем Virtuozzo через веб-интерфейс. // «Системный администратор», № 1-2, 2017 г. – С. 32-35 (http://samag.ru/archive/article/3357).

[2] Официальная документация по продуктам Virtuozzo – http://docs.virtuozzo.com.

[3] Openstack: Integrate Identity with LDAP – https://docs.openstack.org/admin-guide/identity-integrate-with-ldap.html.

Ключевые слова: виртуализация, Virtuozzo.

Визитка

АНДРЕЙ СЕМЕНОВ,
Восточное управление ДВБФ ФГУП «Росморпорт»,
главный специалист ОИТ, avsemenov@gmail.com

Оптимизация с помощью виртуализации

Организуем несколько рабочих мест из одного десктопа. Часть 2

Поддержка функций виртуализации в среднебюджетных рабочих станциях позволяет оптимизировать расходы на рабочих местах пользователей

В первой части статьи [1] был рассмотрен метод разделения ресурсов рабочей станции на два рабочих места с помощью виртуализации на основе гипервизора Xen с использованием в качестве операционной системы хоста ОС Ubuntu 16.04.

Для организации необходимого функционала на основе Xen потребовалась сборка ядра из исходников с определенными опциями, что привело к значительным временным затратам.

Чтобы по возможности упростить настройку и избежать излишних временных затрат, я решил воспользоваться готовым решением для виртуализации на основе KVM – Proxmox VE (Proxmox Virtual Environment, далее PVE [2, 3]) – открытого продукта, распространяемого по свободной лицензии AGPLv3, совместимой с GPLv3.

PVE – это средство управления виртуальным окружением на базе KVM и системы контейнерной изоляции LXC (функционал LXC в рамках рассматриваемой задачи не нужен и не будет задействован). Строго говоря, PVE – это не система виртуализации, а инструментарий управления средой, в которой выполняются виртуальные окружения KVM и LXC. PVE предоставляет удобный графический веб-интерфейс для управления виртуальными машинами, хранилищем и многими другими функциями, что позволит сократить затраты времени на установку и настройку. Продукт распространяется в виде готового дистрибутива на основе Debian, однако имеются репозитории для Debian, также официальную сборку PVE [4] поддерживает разработчик отечественного дистрибутива ALT Linux.

С недавних пор разработчики PVE ввели платную подписку на специальный репозиторий, который рекомендуют использовать в окружениях уровня предприятия и для которого отбираются самые стабильные пакеты (так утверждает разработчик). Подписчикам оказывается своевременная техническая поддержка. Бесплатный репозиторий без подписки не рекомендуется использовать на предприятиях, и те, кого не устраивает такой подход, могут воспользоваться официальным Git [5] от разработчиков PVE и сами собирать пакеты или воспользоваться сборкой от ALT Linux (они также берут за основу исходные коды PVE). Я решил установить PVE из бесплатных репозиториев в заранее установленный дистрибутив Debian 8.7.x, чтобы совместить быстроту установки PVE с гибкостью настроек дистрибутива общего назначения Debian.

Установка PVE как пакета в ОС Debian

Необходимо предварительно установить ОС Debian 8.7.x, после чего подключить репозиторий PVE согласно инструкции на официальном сайте [6]:

```
$ sudo echo "deb http://download.proxmox.com/debian jessie ↵
        pve-no-subscription" > /etc/apt/sources.list.d/ ↵
        pve-install-repo.list
```

Добавляем ключ репозитория:

```
$ wget -O- "http://download.proxmox.com/debian/key.asc" ↵
        | apt-key add -
```

Обновляем список пакетов и устанавливаем PVE и сопутствующие пакеты:

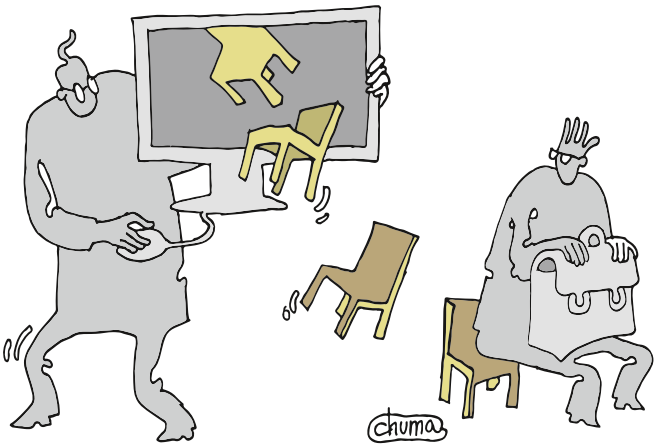
```
$ sudo apt-get update
$ sudo apt-get install proxmox-ve postfix ksm-control-daemon ↵
        open-iscsi systemd-sysv ssh
```

После чего рекомендуется удалить пакет os-prober, так как данное ПО может в определенных сценариях работы привести к порче файловых систем виртуальных машин (см. [6]):

```
$ sudo apt-get remove os-prober
```

Перезагружаем систему. Загрузка произойдет с новым ядром Linux ветки 4.4.x из репозитория PVE. Далее переходим к настройке PVE через веб-интерфейс по адресу: https://ip:8006.

Также нужно будет настроить сетевой мост и, возможно, создать дополнительные хранилища для дисков виртуальных



Proxmox Virtual Environment – это средство управления виртуальным окружением на базе KVM и системы контейнерной изоляции LXC

машин в дополнение к существующим. Процедура первоначальной настройки хорошо описана на сайте [7].

После настройки можно установить гостевую ОС, например Windows 7, и затем переходить к настройке «проброса» видеокарты и USB-устройств.

Вновь напомню о требованиях наличия поддержки технологий Intel VT-x/VT-d или AMD-V/AMD-Vi аппаратными конфигурациями, которые уже были озвучены в первой части данной статьи [1].

Однако в случае с KVM есть особенности, связанные с используемой технологией «проброса» устройств VFIO, пришедшей на смену устаревшей pci-stub, начиная с версии ядра Linux 4.1.

VFIO (Virtual Function I/O) – это фреймворк для драйверов пользовательского уровня (userspace), созданный для организации взаимодействия эмулятора QEMU (Quick EMUlation) непосредственно с «железом» с помощью гипервизора KVM.

VFIO использует возможности IOMMU API type1, поддерживаемые аппаратными платформами Intel с поддержкой VT-d и AMD с поддержкой AMD-Vi, и базируется на группах изоляции IOMMU (IOMMU groups). Подробнее о VFIO можно узнать из презентации разработчика [8], его блога [9], а также на сайте kernel.org [10].

Итак, для «проброса» устройств в виртуальную машину, созданную с помощью гипервизора KVM и эмулятора QEMU, необходимо активировать модули VFIO (далее можно следовать инструкции [11] на сайте PVE).

В /etc/modules добавим модули:

```
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
```

Также указываем ядру, что аппаратная конфигурация поддерживает VT-d/AMD-Vi (подробнее в первой части статьи [1]):

В /etc/default/grub в параметр GRUB_CMDLINE_LINUX_DEAULT добавляем к уже вписанным значениям через пробел значение intel_iommu=on для платформ Intel или amd_iommu=on для AMD:

```
GRUB_CMDLINE_LINUX_DEAULT = "intel_iommu=on"
```

Обновляем загрузчик:

```
$ sudo update-grub
```

и перезагружаем систему.

Теперь можно приступить к проверкам поддержки всего необходимого функционала:

1. Проверить наличие поддержки IOMMU interrupt remapping – части функционала IOMMU, отвечающей за поддержку изоляции и маршрутизации прерываний от устройств и внешних контроллеров прерываний в виртуальную машину [12, раздел 2.5].

На сайте PVE утверждается, что программная поддержка данного функционала для платформы AMD отсутствует, хотя аппаратная реализована. К сожалению, я не могу подтвердить или опровергнуть данное утверждение, так как тестировал функционал только на платформе Intel.

Для проверки необходимо выполнить команду:

```
$ sudo dmesg | grep ecap
```

Вывод команды:

```
[    0.024746] DMAR: dmar0: reg_base_addr fed90000 ver 1:0
cap c0000020660462 ecap f0101a
[    0.024750] DMAR: dmar1: reg_base_addr fed91000 ver 1:0
cap d2008020660462 ecap f010da
```

Если последняя цифра в строке после ecap 8, 9, a, b, c, d, e или f, то технология IOMMU interrupt remapping поддерживается.

Если не поддерживается, можно попробовать разрешить незащищенные прерывания (unsafe interrupts):

```
$sudo echo "options vfio_iommu_type1 ↵
```

```
        allow_unsafe_interrupts =1" ↵
        > /etc/modprobe.d/iommu_unsafe_interrupts.conf
```

Однако работа подобной конфигурации не гарантируется.

2. Проверить IOMMU Isolation (IOMMU Groups). Также необходимо убедиться, что каждое устройство для «проброса» находится в отдельной группе изоляции IOMMU.

Для проверки нужно выполнить команду:

```
$ sudo find /sys/kernel/iommu_groups/ -type l
```

или запустить скрипт, выдающий более удобный для чтения результат:

```
#!/bin/bash

shopt -s nullglob
for d in /sys/kernel/iommu_groups/*/devices/*; do
    n=${d#*/iommu_groups/*}; n=${n%%/*}
    printf 'IOMMU Group %s ' "$n"
    lspci -nns "${d##*/}"
done;
```

Результат работы скрипта (см. рис. 1).

Если в группе IOMMU содержится несколько устройств, то готовить к «пробросу» и изолировать от ОС хоста с помощью модуля vfio-pci нужно все устройства группы. Единственным исключением из этого правила будет PCI-E processor root port, который может находиться в одной IOMMU-группе с видеокартой, но изолировать его не нужно.

Несколько устройств, принадлежащих одной IOMMU-группе, нельзя назначать разным виртуальным машинам – только одной.

Также на большинстве плат порты PCI реализованы через PCI-E-to-PCI Bridge, поэтому все устройства, установленные в слотах PCI, будут находиться в одной IOMMU-группе и их также невозможно будет разделить между разными виртуальными машинами.

Зная такие ограничения, можно приступить к настройке «проброса» видеокарты. Для начала необходимо запретить хостовой ОС использовать драйверы дискретных видеокарт, которые мы хотим задействовать в виртуальных машинах, добавив драйверы в blacklist:

```
echo "blacklist radeon" >> /etc/modprobe.d/blacklist.conf
echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf
echo "blacklist nvidia" >> /etc/modprobe.d/blacklist.conf
```

Далее с помощью команды:

```
$ sudo lspci
```

определяем адрес видеокарт(ы) для «проброса»:

```
01:00.0 VGA compatible controller: Advanced Micro Devices,
Inc. [AMD/ATI] Barts PRO [Radeon HD 6850]
01:00.1 Audio device: Advanced Micro Devices, Inc. [AMD/ATI]
Barts HDMI Audio [Radeon HD 6800 Series]
```

А также deviceid и vendorid видеокарты:

```
$ sudo lspci -n -s 01:00
```

результатом работы команды будут строки:

```
01:00.0 0300: 1002:6779
01:00.1 0403: 1002:aa98
```

где:

- **1002:6779** – vendorid:deviceid видеокарты;
- **1002:aa98** – vendorid:deviceid звукового устройства видеокарты (видеокарта – составное устройство).

Создадим файл vfio.conf и добавим в него vendorid:deviceid имеющихся видеокарт и их звуковых устройств (если есть) для привязки их к модулю vfio-pci и изоляции устройств от ОС хоста:

```
echo "options vfio-pci ids=1002:6779,1002:aa98,10de:0622" ↵
        > /etc/modprobe.d/vfio.conf
```

«Проброс» видеокарт в виртуальные машины может осуществляться несколькими способами:

- **Primary VGA Passthrough** – вывод изображения с момента включения виртуальной машины перенаправляется на физический видеоадаптер.
- **Secondary VGA Passthrough** – при включении виртуальной машины ей назначается виртуальный видеоадаптер, на который осуществляется вывод графической информации, а после загрузки ОС видеовывод перенаправляется на физическую видеокарту.

Рисунок 1. Вывод IOMMU Groups

```
IOMMU Group 0 00:00.0 Host bridge [0600]: Intel Corporation 4th Gen Core Processor DRAM Controller [8086:0c00] (rev 06)
IOMMU Group 10 00:1c.1 PCI bridge [0604]: Intel Corporation 82801 PCI Bridge [8086:244e] (rev d5)
IOMMU Group 10 03:00.0 PCI bridge [0604]: Intel Corporation 82801 PCI Bridge [8086:244e] (rev 41)
IOMMU Group 11 00:1c.3 PCI bridge [0604]: Intel Corporation 8 Series/C220 Series Chipset Family PCI Express Root Port #4 [8086:8c16] (rev d5)
IOMMU Group 12 00:1c.4 PCI bridge [0604]: Intel Corporation 8 Series/C220 Series Chipset Family PCI Express Root Port #5 [8086:8c18] (rev d5)
IOMMU Group 13 00:1d.0 USB controller [0c03]: Intel Corporation 8 Series/C220 Series Chipset Family USB EHCI #1 [8086:8c26] (rev 05)
IOMMU Group 14 00:1f.0 ISA bridge [0601]: Intel Corporation Q87 Express LPC Controller [8086:8c4e] (rev 05)
IOMMU Group 14 00:1f.2 SATA controller [0106]: Intel Corporation 8 Series/C220 Series Chipset Family 6-port SATA Controller 1 [AHCI mode] [8086:8c02] (rev 05)
IOMMU Group 14 00:1f.3 SMBus [0c05]: Intel Corporation 8 Series/C220 Series Chipset Family SMBus Controller [8086:8c22] (rev 05)
IOMMU Group 15 05:00.0 USB controller [0c03]: VIA Technologies, Inc. VL80x xHCI USB 3.0 Controller [1106:3432] (rev 03)
IOMMU Group 16 06:00.0 VGA compatible controller [0300]: Advanced Micro Devices, Inc. [AMD/ATI] Caicos [Radeon HD 6450/7450/8450 / R5 230 OEM] [1002:6779]
IOMMU Group 16 06:00.1 Audio device [0403]: Advanced Micro Devices, Inc. [AMD/ATI] Caicos HDMI Audio [Radeon HD 6400 Series] [1002:aa98]
IOMMU Group 1 00:01.0 PCI bridge [0604]: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor PCI Express x16 Controller [8086:0c01] (rev 06)
IOMMU Group 1 01:00.0 VGA compatible controller [0300]: NVIDIA Corporation G94 [GeForce 9600 GT] [10de:0622] (rev a1)
IOMMU Group 2 00:02.0 VGA compatible controller [0300]: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor Integrated Graphics Controller [8086:0412] (rev 06)
IOMMU Group 3 00:03.0 Audio device [0403]: Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor HD Audio Controller [8086:0c0c] (rev 06)
IOMMU Group 4 00:14.0 USB controller [0c03]: Intel Corporation 8 Series/C220 Series Chipset Family USB xHCI [8086:8c31] (rev 05)
IOMMU Group 5 00:16.0 Communication controller [0780]: Intel Corporation 8 Series/C220 Series Chipset Family MEI Controller #1 [8086:8c3a] (rev 04)
IOMMU Group 5 00:16.3 Serial controller [0700]: Intel Corporation 8 Series/C220 Series Chipset Family KT Controller [8086:8c3d] (rev 04)
IOMMU Group 6 00:19.0 Ethernet controller [0200]: Intel Corporation Ethernet Connection I217-LM [8086:153a] (rev 05)
IOMMU Group 7 00:1a.0 USB controller [0c03]: Intel Corporation 8 Series/C220 Series Chipset Family USB EHCI #2 [8086:8c2d] (rev 05)
IOMMU Group 8 00:1b.0 Audio device [0403]: Intel Corporation 8 Series/C220 Series Chipset High Definition Audio Controller [8086:8c20] (rev 05)
IOMMU Group 9 00:1c.0 PCI bridge [0604]: Intel Corporation 8 Series/C220 Series Chipset Family PCI Express Root Port #1 [8086:8c10] (rev d5)
```

Первый метод используется в KVM VFIO и позволяет увидеть вывод изображения на «проброшенную» видеокарту сразу при старте виртуальной машины, но сопряжен с некоторыми ограничениями VGA BIOS.

Основное ограничение VGA BIOS при использовании нескольких видеокарт для одновременной работы – это использование фиксированных участков памяти с одинаковым адресом 0xa0000-0xbffff, который используется для инициализации видеоустройства.

Для обработки данной коллизии был разработан функционал vga arbitration, который призван обрабатывать ошибки в ОС хоста. Коллизии возникают, когда Xorg при наличии двух видеоадаптеров в системе отключает DRI (Direct Rendering Interface), используемый для аппаратной поддержки OpenGL, что приводит в итоге к сильному замедлению 3D графики на хосте или в редких случаях вообще к отключению графического режима и выходу в консоль. Подробнее об этом – в статье основного разработчика VFIO [13].

Второй метод «проброса» видеокарты используется в гипервизоре Xen. Этот подход позволяет избавиться от проблем использования одних и тех же участков памяти устаревшим программным кодом VGA BIOS, так как начальная инициализация при загрузке виртуальной машины выполняется на виртуальном видеоадаптере, а «проброшенный» видеоадаптер подключается после загрузки ОС.

Однако существуют и другие подходы. Разработчик VFIO рекомендует применять новые возможности VFIO, позволяющие избавиться от использования устаревшего VGA BIOS и связанных с ним проблем [13, 14].

Для этого предлагается для запуска виртуальных машин вместо классического SeaBIOS (Open Source-реализация 16-битного x86 BIOS) использовать прошивку OVMF (Open Virtual Machine Firmware – UEFI совместимая прошивка, разработанная для загрузки виртуальных машин в QEMU) и современные дискретные видеокарты с UEFI-прошивкой вместо классического VGA BIOS.

Описанный выше способ можно использовать, с условием, что операционная система виртуальной машины имеет поддержку UEFI (это Windows 8+ и большинство современных Linux-дистрибутивов).

Один из простых способов узнать, поддерживает ли видеоадаптер UEFI, – запустить программу GPU-Z и проверить, установлен ли флажок на соответствующем пункте главного окна программы (см. рис. 2).

Если Windows под рукой нет, то можно узнать это и в Linux, следуя инструкции на сайте PVE [6].

В зависимости от типа прошивки, с которым запущена виртуальная машина (SeaBIOS или OVMF), и от типа прошивки видеокарты (классический VGA BIOS или UEFI), а также от используемого в виртуальной машине набора системных устройств (эмулируемого чипсета), доступно четыре варианта организации «проброса» с помощью VFIO.

GPU OVMF PCI PASSTHROUGH (рекомендуемый)

Загрузка UEFI. Необходима ОС с поддержкой такой загрузки (Windows 8 и новее).

В данном режиме можно добавить параметр disable_vga=1 в модуль vfio-pci, при этом видеокарта будет по возможности исключена из vga arbitration. Для виртуальной машины будет эмулироваться оборудование на чипсете i440fx (чипсет по умолчанию в QEMU, нет поддержки PCI-E и USB 3.0).

```
echo "options vfio-pci ids=1002:6779,1002:aa98 ↵
        disable_vga=1" > /etc/modprobe.d/vfio.conf
```

Для данного режима необходимо, чтобы у видеокарты была UEFI-прошивка.

В конфигурационном файле виртуальной машины /etc/pve/qemu-server/<vmid>.conf должны быть указаны следующие параметры:

```
bios: ovmf
hostpci0: 01:00,x-vga=on
```

GPU OVMF PCI Express PASSTHROUGH

Также загрузка UEFI. Необходима ОС Windows 8 и новее. Для виртуальной машины будет эмулироваться оборудование на чипсете Q35 (чипсет Intel Q35 с поддержкой PCI-E в QEMU).

Аналогично предыдущему режиму для данного режима также необходимо, чтобы у видеокарты была UEFI-прошивка.

В конфигурационном файле виртуальной машины /etc/pve/qemu-server/<vmid>.conf должны быть указаны следующие параметры:

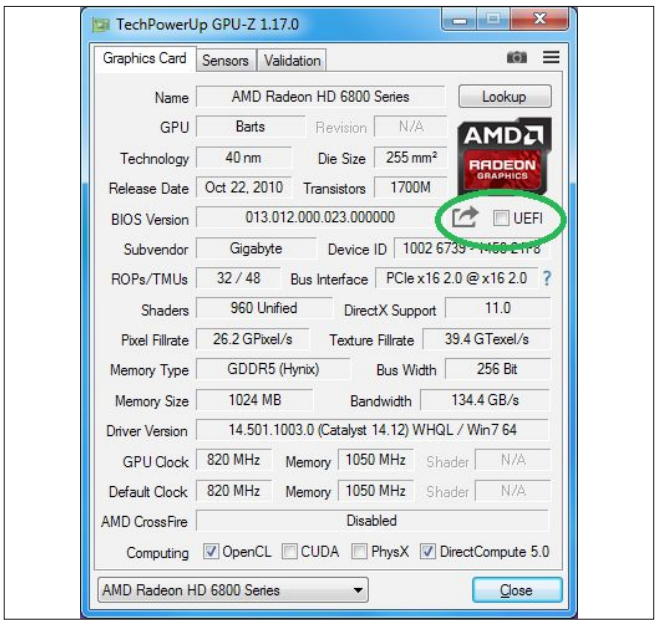
```
bios: ovmf
machine: q35
hostpci0: 01:00,pcie=1,x-vga=on
```

GPU SeaBios PCI PASSTHROUGH

В этом режиме возможно в качестве гостя использовать Windows 7, загрузка будет осуществляться с помощью SeaBios (Open Source-реализация 16-битного x86 BIOS).

Для виртуальной машины будет эмулироваться чипсет i440fx (чипсет по умолчанию в QEMU).

Рисунок 2. Окно программы GPU-Z без флага поддержки UEFI



Кстати

Для успешного «проброса» видеокарт в две и более виртуальные машины, которые будут работать одновременно, необходимо убедиться, что порты PCI-E, в которые вставлены видеокарты, не находятся на шине процессора.

Дело в том, что если процессор не поддерживает технологию ACS (Access Control Services), то все порты PCI-E на шине процессора будут состоять в одной iommu-группе, что не позволит использовать видеоадаптеры, находящиеся в этих портах в разных гостевых ОС одновременно. Функционал ACS встречается лишь в серверных CPU (Xeon E5, E3 не поддерживает ACS) и Hi-End Desktop CPU Core i7 для разъемов LGA2011 и LGA2011-3 [15].

В бюджетных материнских платах несколько портов PCI-E на шине процессора встречаются не часто (в основном PCI-E-порт только один, обычно самый быстрый), а остальные порты находятся на шине чипсета PCH (Platform Controller Hub). В таком случае проблем возникнуть не должно, и все видеокарты скорее всего попадут в разные iommu-группы изоляции. На рис. 3 видно, что на PCI-E-шине процессора находится один скоростной PCI-E-порт (PCI-E x16), второй PCI-E-порт для видеокарты (PCIe x4) находится на шине чипсета. В такой конфигурации проблем с «пробросом» видеокарт в разных гостей не возникнет независимо от наличия поддержки ACS-процессором.

Если же в материнской плате на шине процессора находится более одного порта PCI-E и их нужно задействовать для разных виртуальных машин и при этом процессор не поддерживает ACS, то можно указать в параметрах загрузки ядра опцию VFIO

```
pcie_acs_override= value
```

где value может принимать следующие значения:

- **downstream** – поддержка ACS для всех портов;
- **multifunction** – поддержка ACS для многофункциональных устройств (например, видеокарты со звуковым устройством);
- **id:nnnn:nnnn** – поддержка ACS для конкретного устройства в формате vid:did (vendor/device ID) в шестнадцатеричном формате.

Так как на самом деле поддержка ACS для PCI-E-устройств включается в таком случае без подтверждения поддержки со стороны «железа», такая конфигурация может не заработать. По возможности лучше не использовать для видеокарт более одного порта PCI-E на шине процессора, если он не поддерживает изоляцию устройств (ACS), а воспользоваться портами PCH.

Возможны проблемы, связанные с VGA arbitration [13].

В конфигурационном файле /etc/pve/qemu-server/<vmid>.conf указать только:

```
hostpci0: 01:00,x-vga=on
```

GPU Seabios PCI Express PASSTHROUGH

В этом режиме также возможно в качестве гостя использовать Windows 7. Для виртуальной машины будет эмулироваться чипсет Q35.

В конфигурационном файле /etc/pve/qemu-server/<vmid>.conf указать:

```
machine: q35
hostpci0: 01:00,pcie=1,x-vga=on
```

Одна из видеокарт на бюджетном чипсете AMD HD6450 поддерживала UEFI, и я смог протестировать и Windows 7 с использованием SeaBios и Windows 10 с использованием OVMF.

После успешного «проброса» видеоадаптера(ов) в виртуальные машины можно заняться периферийными USB-устройствами.

Чтобы использовать в виртуальной машине не отдельные устройства или порты, а USB-контроллер целиком, нужно также добавить его vednorid:deviceid в /etc/modprobe.d/vfio.conf, аналогично тому, как это сделано с видеокартами, чтобы изолировать данное устройство от ОС хоста.

В моем случае в виртуальную машину «прокидывается» PCI-E USB-контроллер на чипсете VIA:

```
IOMMU Group 15 05:00.0 USB controller [0c03]:
VIA Technologies, Inc. VL80x xHCI USB 3.0 Controller
[1106:3432] (rev 03)
```

Файл /etc/modprobe.d/vfio.conf будет выглядеть следующим образом:

```
options vfio-pci ids=1002:6779,1002:aa98,10de:0622,1106:3432
```

В опциях модуля vfio-pci перечислены две видеокарты (AMD HD6450 и Nvidia 9600GT) и PCI-E USB-контроллер.

Для проверки необходимо убедиться, что в выводе команды:

```
$ sudo lspci -v
```

у указанных в модуле vfio-pci устройств есть строчка:

```
Kernel driver in use: vfio-pci
```

Также необходимо добавить видеокарты и USB-контроллер в соответствующие конфигурационные файлы /etc/pve/qemu-server/<vmid>.conf виртуальных машин.

В одну из виртуальных машин я добавлю видеокарту, а в другую – видеокарту и USB-контроллер.

Для первой машины строка с видеокартой будет такой:

```
machine: q35
hostpci0: 01:00,pcie=1,x-vga=on
```

а для второй, кроме видеокарты, добавился USB-контроллер:

```
hostpci0: 06:00,x-vga=on
hostpci1: 00:14.0
```

Кроме возможности «проброса» целого USB-контроллера, существует также возможность передавать виртуальной машине конкретный USB-порт или же конкретное USB-устройство [16].

USB-устройства передаются в виртуальную машину по vendor:product-id, а USB-порты – по host bus и port.

Если устройство, указанное в конфигурационном файле виртуальной машины, отсутствует на момент ее старта, то оно без проблем появится позже, когда устройство подключат.

Для удобного просмотра нумерации USB-шин и портов можно воспользоваться командой:

```
$ sudo qm monitor vmID
qm>info usbhost
```

После выполнения нашей команды появится подобный вывод:

```
  Bus 4, Addr 15, Port 2, Speed 5000 Mb/s
    Class 00: USB device 174c:5106, StoreJet Transcend
  Bus 3, Addr 2, Port 13, Speed 1.5 Mb/s
    Class 00: USB device 1220:0008
```

По адресу Bus 4 Port 2 подключен USB-диск.

По адресу Bus 3 Port 13 подключена радиоклавиатура

Данную клавиатуру и именно указанный жесткий диск можно подключить к виртуальной машине, добавив в файл /etc/pve/qemu-server/<vmid>.conf следующие строки:

```
usb0: 174c:5106
usb1: 1220:0008
```

Более подробно разобраться, как организовано подключение USB-портов в виртуальную машину, можно, выполнив команду:

```
lsusb -t
```

```
/:  Bus 06.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p,
5000M
/:  Bus 05.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/1p,
480M
/:  Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/6p,
5000M
    |__ Port 2: Dev 15, If 0, Class=Mass Storage, Driver=usb-
storage, 5000M
/:  Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_
hcd/14p, 480M
    |__ Port 13: Dev 2, If 0, Class=Human Interface Device,
Driver=usbfs, 1.5M
    |__ Port 13: Dev 2, If 1, Class=Human Interface Device,
Driver=usbfs, 1.5M
```

Чтобы пробросить в виртуальную машину не конкретное USB-устройство, а определенный USB-порт, в файл /etc/pve/qemu-server/vm_id.conf нужно добавить строки (см. выше вывод команды lsusb -t):

```
usb0: host=4-2
usb1: host=3-13
```

Таким образом, виртуальной машине будут доступны два USB-порта с подключенными к ним устройствами.

Аналогичным образом в виртуальные машины подключаются все необходимые USB-устройства.

• • •

Следующий этап – сравнение производительности различных подсистем виртуальных машин с ОС Windows на основе Xen/KVM с установленной непосредственно на «железо» Windows, а также мои субъективные впечатления от использования технологий виртуализации Xen и KVM в разрезе разделения ресурсов компьютера на несколько рабочих мест с «пробросом» устройств. EOF

[1] Семенов А. Оптимизация с помощью виртуализации. Организуем несколько рабочих мест из одного десктопа. // «Системный администратор», № 3, 2017 г. – с. 22-28 (http://samag.ru/archive/article/3385).

[2] Официальный сайт PVE – http://pve.proxmox.com/wiki/Main_Page.

[3] О PVE на сайте Wikipedia – http://ru.wikipedia.org/wiki/Proxmox_Virtual_Environment.

[4] Сборка PVE от Alt Linux – http://www.altlinux.org/Starterkits/server-pve.

[5] Исходный код PVE – http://git.proxmox.com.

[6] Инструкция по установке PVE – http://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_Jessie.

[7] Настройка PVE – http://serveradmin.ru/ustanovka-i-nastroyka-proxmox.

[8] Обзор VFIO от разработчика – http://www.youtube.com/watch?v=WFkdTFTOTpA.

[9] Блог разработчика VFIO – http://vfio.blogspot.ru.

[10] Описание технологии VFIO – http://www.kernel.org/doc/Documentation/vfio.txt.

[11] «Проброс» PCI(e)-устройств – http://pve.proxmox.com/wiki/Pci_passthrough.

[12] Спецификация Intel VT-d – http://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/vt-directed-io-spec.pdf.

[13] VGA arbitration – http://vfio.blogspot.ru/2014/08/whats-deal-with-vga-arbitration.html.

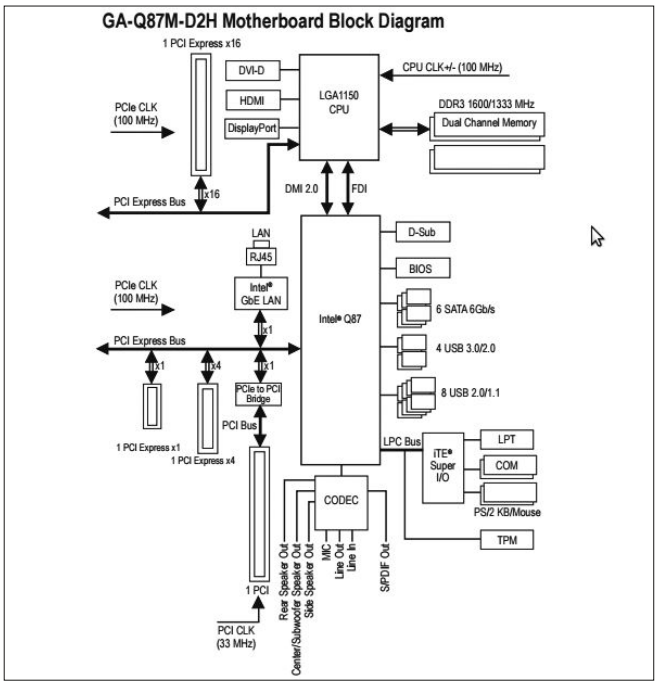
[14] VFIO проблемы «проброса» GPU – http://www.youtube.com/watch?v=NhZ9eIpg2nM.

[15] Процессоры с поддержкой ACS – http://vfio.blogspot.ru/2015/10/intel-processors-with-acs-support.html.

[16] «Проброс» USB-портов – http://pve.proxmox.com/wiki/USB_physical_port_mapping.

Ключевые слова: виртуализация, Linux, ProxmoxVE, Xen.

Рисунок 3. Схема шины процессора и чипсета (PCH) с расположением портов PCI-E на примере материнской платы Gigabyte GA-Q87M-D2H



Визитка

СЕРГЕЙ ИЛЬИЧЕВ,
ГК «Лаборатория Интеллект», г. Тула,
инженер-программист, sergil68@mail.ru

Apache POI HSSF – как «приручить» Excel

Не стоит недооценивать свободные генераторы отчетов. Библиотека с открытым исходным кодом Apache POI доказывает свою состоятельность

Постановка задачи

К теме генерации отчетов время от времени приходится обращаться практически каждому программисту и системному администратору. Получив очередное техзадание сделать отчет для информационной системы предприятия именно в Microsoft Excel и имея некоторый опыт построения отчетов с помощью технологий, описанных в [1], решено было исследовать, какой из современных генераторов отчетов (желательно бесплатный) подошел бы для решения этой задачи. При этом совершенно не хотелось лезть в поднадоевшие механизмы OLE-DDE, привязываясь к одной из операционных систем.

Использование электронных таблиц для формирования отчетов – трюк давний и вполне оправданный. Во-первых, современные динамические таблицы – мощные программные комплексы, способные на математические вычисления вплоть до аппроксимации. Во-вторых, рабочие единицы в электронных таблицах – ячейки, строки, столбцы, – и с ними легко работать алгоритмически, например, используя циклы. И напоследок: подобные программы установлены практически на любом персональном компьютере.

В процессе поиска информации заставил «потереть руки» в предвкушении чего-то интересного один проект – Apache POI, выпущенный под лицензией Apache License. Официальный сайт проекта, в котором, кстати, может принять участие любой желающий, – [2]. Аббревиатура POI (и другие – см. ниже) на титульной странице сайта никак не расшифровывается, однако в документации [3] даются некоторые разъяснения.

Согласно информации из документации эта Java-библиотека разделена на несколько пакетов: если необходимо прочитать или записать файл Excel (.xls), то можно использовать пакет HSSF, если нужно прочитать или записать файл OOXML Excel (.xlsx), можно использовать XSSF. Комбинированный интерфейс SS позволяет легко считывать и записывать все виды файлов Excel (.xls и .xlsx). Кроме того, существует специализированная реализация SXSSF, которая позволяет работать с очень большими Excel (.xlsx) файлами в памяти с оптимизацией.

С первого взгляда возможности HSSF с лихвой покрывали потребности нашего отчета. После анализа техзадания выполнение задачи было разбито на два этапа: 1-й этап – данные из информационной системы с помощью ее инструментов должны быть выгружены в текстовый файл с символами «#» в качестве разделителей такого вида:

```
Поле 1#Поле 2#Поле 3#Поле 4#Поле 5#Поле 6
Поле 1#Поле 2#Поле 3#Поле 4#Поле 5#Поле 6
Поле 1#Поле 2#Поле 3#Поле 4#Поле 5#Поле 6
```

Затем 2-й этап – файл должен подхватиться программой на Java и после обработки вывестись пользователю на экран уже в Excel. Подобные поэтапные решения применяются в так называемой лоскутной автоматизации, описаны в [1] и довольно часто применяются в информационных системах.

Целью статьи будет описание процесса создания программы на языке программирования Java для решения задач 2-го этапа, а именно открытие текстового файла с разделителями и формирование многостраничного отчета с помощью библиотеки POI, способного открыться в Microsoft Excel. Программировать будем в JDK 6 (Linux Mint 13 LTE, версия Java – OpenJDK 1.6), сама программа, что естественно для языка программирования Java, после компиляции должна выполняться и в Linux и в Windows.

Установка библиотеки

Итак, скачиваем стабильную версию POI для Linux – 3.15 на момент написания статьи (см. [4]).

Распаковываем архив. У меня на компьютере в домашней директории место нашлось в папке java/poi-3.15:

```
$ cd ~/java
$ tar xzvf poi-bin-3.15-20160924.tar.gz
```

Для работы с библиотекой нужно показать путь к ее файлам в переменной окружения CLASSPATH, для этого заходим в файл .profile в домашнем каталоге

```
$ vim .profile
```

в конце файла пишем

```
export POI_HOME=$HOME/java/poi-3.15
export POI_CLASSES=$POI_HOME/poi-3.15.jar
export CLASSPATH=$CLASSPATH:$POI_CLASSES:.
```

Следите за путями, они зависят от того, куда вы разместите файл библиотеки poi-x.xx.jar.

Далее нужно применить изменения в .profile:

```
$ source .profile
```

или, на худой конец, перезагрузиться, после чего можно проверить правильность пути к библиотеке с помощью команды:

```
echo $CLASSPATH
```

Контрольный пример

Теперь необходимо убедиться, что все работает, библиотека установлена правильно, т.е. файлы скопированы, нужные пути прописаны в переменной окружения CLASSPATH, и заодно попрактиковаться в работе с библиотекой.

Повторив пример Writing a new file (см. ссылку [5]), убеждаемся, что это так. Чтобы этот пример был работоспособен, необходимо добавить открытый (public) класс, функцию main() и следующий импорт:

```
import java.io.FileOutputStream;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFCellStyle;
import org.apache.poi.hssf.usermodel.HSSFDataFormat;
import org.apache.poi.hssf.usermodel.HSSFFont;
import org.apache.poi.hssf.usermodel.HSSFRichTextString;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.BorderStyle;
import org.apache.poi.ss.util.CellRangeAddress;
```

При компиляции этого примера в том виде, в котором он есть на сайте, правда, получаем предупреждения об использовании устаревших (deprecated) методов (см. рис. 1), однако для любителей «чистой» компиляции есть информация [6, 7], где указано, чем эти методы заменить.

После запуска программы (допустим, программный файл называется Generator.java) командой:

```
$ java Generator
```

В текущем каталоге обнаруживается файл workbook.xls (см. рис. 2), анализируя который можно понять, что делает пробная программа и какие свойства ячеек Excel она изменяет.

Что еще может POI как генератор отчета? Да практически все, что нужно самому требовательному заказчику: от изменения параметров ячейки таблицы (высоты, ширины, используемого в ячейке шрифта, выравнивания) до размещения диаграмм на листе рабочей книги.

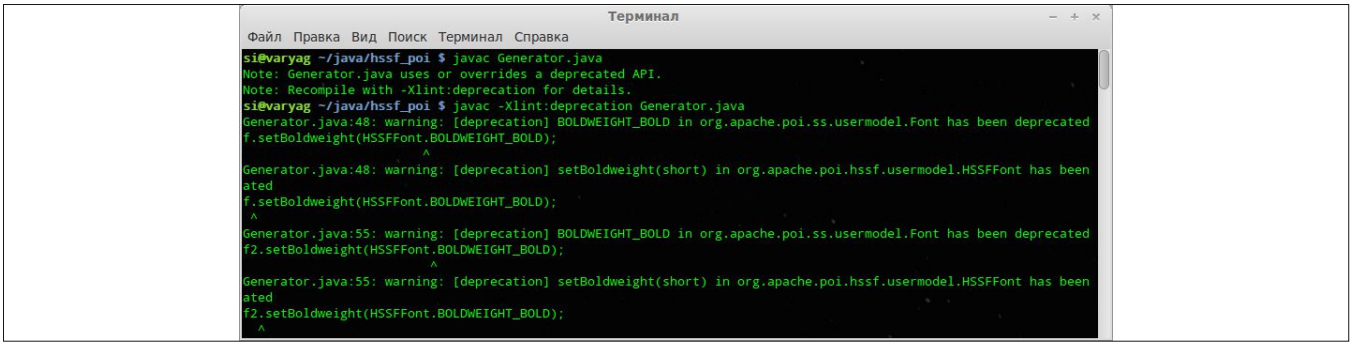
Описание программы для решения технического задания

Ну и, конечно, приведем описание кода, необходимого для решения нашей технической задачи, в котором читается файл и информация из него попадает в ячейки Excel. Такая программа может послужить заготовкой для нужного руководству или коллеге отчета и использует в качестве входного параметра текстовый файл с разделителем «#».

Создадим класс POIExample и одноименный .java-файл, формирующий книгу example.xls из текстового файла с разделителями. В нем можно определить переменные класса и конструктор, в котором и разместится основная часть кода. В конструкторе и определим необходимые классы Apache POI:

```
public class POIExample {
   // вспомогательные переменные
    private StringTokenizer st;
    // для строки текстового файла
    private String line = null;
    // для строки и ячейки Excel
    private short rownum;
    private short cellnum;
  // Конструктор класса
  POIExample(String fileName) throws Exception {
    // выходной поток - новый файл .xls
    FileOutputStream out = new FileOutputStream("example.xls");
    // создаем новую книгу
    HSSFWorkbook wb = new HSSFWorkbook();
    // создаем новый лист
    HSSFSheet s = wb.createSheet();
    // объявляем объект строки
    HSSFRow r = null;
    // объявляем объект ячейки
    HSSFCell c = null;
```

Рисунок 1. Предупреждения при компиляции



Необходимо также определить стили ячеек. Стиль ячейки в библиотеке описывает такие параметры, как: выравнивание текста, его положение, шрифт, обрамление ячейки, цвет заполнения и т.д.

```
// создаем 3 объекта стилей
HSSFCellStyle cs = wb.createCellStyle();
HSSFCellStyle cs2 = wb.createCellStyle();
HSSFCellStyle cs3 = wb.createCellStyle();
```

Создадим объекты для шрифтов, которые используются в стилях:

```
HSSFFont f = wb.createFont();
HSSFFont f2 = wb.createFont();

// устанавливаем размер первого шрифта 14 пунктов
f.setFontHeightInPoints((short)14);
// тип шрифта
f.setFontName("TimesNewRoman");
// делаем шрифт полужирным
f.setBold(true);
```

Устанавливая параметры для второго шрифта, можно изменить, например, цвет методом setColor() или установить тип шрифта «курсив» с помощью метода setItalic(true).

Затем к стилю применяем следующие свойства:

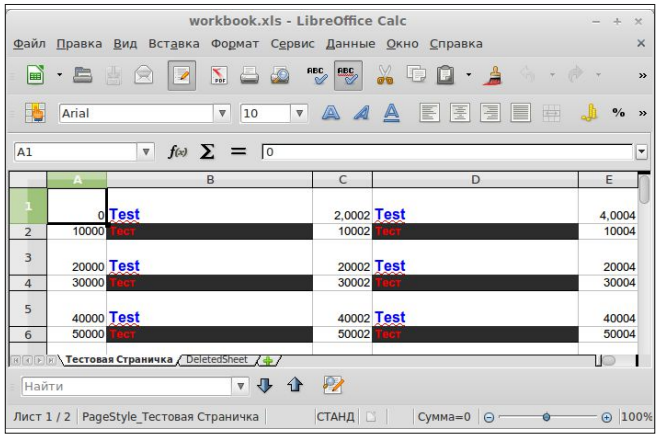
```
// для первого стиля устанавливаем шрифт f2
cs.setFont(f2);
// выравнивание
cs.setAlignment(cs.ALIGN_LEFT);
// обрамление
cs.setBorderBottom(cs2.BORDER_THIN);
cs.setBorderTop(cs2.BORDER_THIN);
cs.setBorderLeft(cs2.BORDER_THIN);
cs.setBorderRight(cs2.BORDER_THIN);
```

Используя свойства стиля можно задать формат ячейки, например «text» для текста или «0.00» для чисел с плавающей запятой. Для этого в пакете org.apache.poi.ss.usermodel есть класс BuiltinFormats:

```
cs2.setDataFormat(HSSFDataFormat.getBuiltinFormat("text"));
```

В классе HSSFWorkbook можно использовать метод, который задает имя листа:

Рисунок 2. Файл workbook.xls




```
wb.setSheetName(0, "Отчет за V квартал");
```

Для созданного входного потока можно явно указать кодировку Windows 1251

```
BufferedReader in = new BufferedReader(new InputStreamReader(
        new FileInputStream(fileName), ↵
            Charset.forName("CP1251")));
```

Нам понадобится также счетчик строк, который начинается с нулевого значения

```
rownum = (short) 0;
```

Заголовок отчета можно сделать, используя свойства строки и уже знакомые свойства ячеек:

```
r = s.createRow(rownum);
cellnum = (short) 0;
c = r.createCell(cellnum);
// устанавливаем высоту ячейки заголовка
r.setHeight((short) 450);
// устанавливаем стиль для ячейки
c.setCellStyle(cs3);
// текст для заголовка
c.setCellValue("Заголовок отчета за V квартал");
```

Основной цикл программы сделан так, что может обработать строки с любым количеством полей, разделенных «#»:

```
// идем по строкам текстового файла
while( (line = in.readLine()) != null) {
  if(line.trim().length()==0) break;

  // создаем новую строку
  r = s.createRow(rownum);
  //уст. высоту
  r.setHeight((short) 400);

  // разбиваем строку на токены, разделитель "#"
  st = new StringTokenizer(line, "#");

  String[] a = new String[n];

  for (int j = 0; j < st.countTokens(); j++) {

    a[j] = st.nextToken();
    cellnum = (short) j;
    // создаем ячейку
    c = r.createCell(cellnum);
    // первая ячейка пошире и шрифт выровнен по центру
       // (используем стиль cs)
    if (j == 0) {
            c.setCellStyle(cs);
            s.setColumnWidth((short) cellnum, ↵
              (short) 14000);
    }
    //остальные используют стиль cs2
    else {
            c.setCellStyle(cs2);
            s.setColumnWidth((short) cellnum, ↵
              (short) 3500);
    }
    // устанавливаем значение ячейки
    c.setCellValue(a[j]);

  }
  // переходим к следующей строке
  rownum++;

}
```

В конце программного модуля закрываем поток чтения файла и выходной поток. Остается в функции main() вызвать конструктор POIExample() и в качестве параметра передать ему имя файла, в котором находится информация:

```
    in.close();

    // записываем информацию и закрываем выходной поток
    wb.write(out);
    out.close();

    return;
  }

  public static void main (String args[]) throws Exception {
        String file = "";

        for(int n = 0; n < args.length; n++) {
            if (args[n].equals("-f")) file = args[++n];
            else throw new IllegalArgumentException( ↵
                  "Неверный аргумент!");
        }
        new POIExample (file);
  }
}
```

После компиляции в ОС Windows, например, программу можно запустить таким командным файлом, задав имя файла для обработки в параметре командной строки:

```
cd path\to\program
SET CLASSPATH=.;path\to\poi-3.15\poi-3.15.jar
java your.packet.POIExample -f %1
start /b path\to\program\example.xls
```

Полный код этого примера можно скачать с сайта samag.ru.

• • •

Мы рассмотрели ключевые моменты использования технологии POI для формирования отчета и выяснили, что эта библиотека с открытым исходным кодом помогает быстро и качественно «приручить» Excel. За остальным обращайтесь к справке [3], которая, кстати, хорошо документирована. Отмечу также скорость формирования отчета с помощью Apache POI, она приятно удивляет! А вам желаю удивить коллег отличным отчетом! EOF

[1] Java-отчет с помощью OpenOffice.org Writer – http://www.learn2prog.ru/ooo-report.

[2] Сайт проекта Apache POI – https://poi.apache.org.

[3] Страница документации Apache POI – https://poi.apache.org/apidocs/index.html.

[4] Страница загрузки (download) библиотеки –.https://poi.apache.org/download.html#POI-3.15.

[5] Описание API-библиотеки для работы с электронными таблицами – https://poi.apache.org/spreadsheet/how-to.html.

[6] Интерфейс Font-библиотеки, поля и методы – https://poi.apache.org/apidocs/org/apache/poi/ss/usermodel/Font.html.

[7] Устаревшие методы Apache POI – https://poi.apache.org/apidocs/deprecated-list.html.

Ключевые слова: создание отчетов, Java, Excel, Apache POI, HSSF.

Визитка

ОЛЕГ ФИЛИППОВ,
АНТ-Информ, заместитель начальника отдела разработки, comol@mail.ru

Инструменты технической поддержки филиальной сети

В основе статьи — накопленный опыт использования различных систем в отделе технической поддержки филиальной сети

Техническая поддержка филиальной сети – достаточно типовая задача для ИТ-специалиста. За годы работы в ретейле мне пришлось столкнуться с очень большим количеством систем, которые помогали службе технической поддержки выполнять свои функции. Часто это были весьма специфичные решения: начиная от использования собраний в Outlook как HelpDesk-системы и заканчивая использованием антивируса в качестве системы мониторинга удаленных хостов.

Так или иначе, за некоторое время удается подобрать системы, которые наилучшим образом подходят для решения задач подразделения технической поддержки. В моем случае эти задачи главным образом связаны с поддержкой распределенной фронт-офисной системы на платформе 1С.

Как правило, в технической поддержке нужны следующие категории систем:

- Система HelpDesk
- Система базы знаний
- Система мониторинга удаленных рабочих станций
- Мессенджер
- Система удаленного управления рабочими станциями
- Система учета ИТ-оборудования
- Система управления конфигурациями
- Система развертывания релизов (обновления конфигурации)
- Система централизованной установки софта и административных действий на рабочих станциях

Часто это неполный перечень информационных систем, которые используются в компаниях. Можно вспомнить, что есть стек продуктов HP, которые охватывают весь спектр подобных задач, являясь, по сути, одной системой. Но в ретейле, когда бизнес считает деньги и придерживается соображений экономии, дорогие продукты от HP редко используются. Кроме того, на мой взгляд, продукты HP достаточно сложны в освоении и администрировании, а интерфейс оставляет желать лучшего. Универсальные решения от вендоров (Microsoft/1С/SAP) достаточно хорошо выполняют одни задачи и абсолютно не проработаны для других. Поэтому со временем я пришел к выводу, что для каждой группы задач нужно подбирать свою систему, которая подходит для них наилучшим образом. Часто подобные узкоспециализированные системы не стоят денег, т.к. являются достаточно простыми Open Source-решениями. Если подходящей системы не существует, бывает проще вложить силы в разработку, чем использовать некоторые половинчатые решения. Далее рассмотрим отдельно системы каждой категории.

Система HelpDesk

Является, по сути, центральной системой для технической поддержки. От ее выбора зависит очень много. У меня была возможность протестировать многие решения подобного рода:

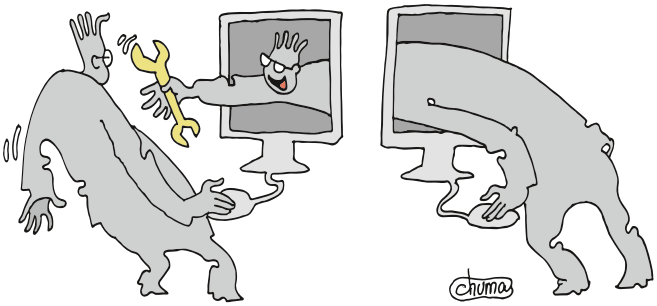
- HP OpenView
- ManageEngine
- Mantis
- Redmine
- 1С Итилиум
- 1С ITIL
- OTRS

Как правило, мой выбор всегда останавливается на решениях от «1С» – по элементарно простым причинам: вся техподдержка умеет с ними работать (1С-ный интерфейс знаком весьма неплохо) и кастомизация систем ограничена только личными фантазиями (в штате есть 1С-программисты).

Наиболее выгодным образом выглядит 1С:ITIL. Данная система является достаточно функциональной, на рынке существует давно, весьма проработана. Исходный код полностью открыт, лицензии нужны только обычные 1С. В части функциональности при этом ничем особо не уступает дорогим продуктам от HP. Кроме всего прочего, содержит в себе модули управления конфигурациями и учета оборудования.

Но в настоящий момент мы используем OTRS [1]. Перед 1С ITIL у нее есть ряд преимуществ:

- Полноценный веб-интерфейс, который позволяет с ней работать без проблем с удаленных компьютеров без установленного клиента 1С (веб-интерфейс от 1С все-таки оставляет желать много лучшего).



Для каждой группы задач нужно подбирать свою систему, которая подходит для них наилучшим образом

> Удобный функционал очередей для распределения заявок (у 1С все-таки все в куче – другой принцип).

Для нас нормальный веб-интерфейс (см. рис. 1) оказался критичен – много сотрудников, не знакомых с 1С, работают удаленно. Кроме всего прочего, сама система очень неплоха и бесплатна.

Система базы знаний

Каждая уважающая себя HelpDesk-система имеет внутри себя базу знаний. Использовать базу знаний внутри HelpDesk-системы кажется логичным на первый взгляд. К сожалению, только на первый. Мне не встречалось ни одной удачной реализации базы знаний внутри HelpDesk-системы. Это все-таки должен быть специализированный продукт, решающий подобные задачи.

База знаний – это не набор неструктурированных документов Word, описывающих разные процессы, это не папка с инструкциями, это не набор бессвязных ответов на обращения.

База знаний – некоторое структурированное хранилище информации, актуальность которого поддерживается и отслеживается, по нему должен быть доступен поиск.

Разумеется, идеальная база знаний должна быть в формате Wiki, который уже давно стал стандартом для подобного рода решений. Из Wiki-движков как наиболее функциональные можно выделить следующие:

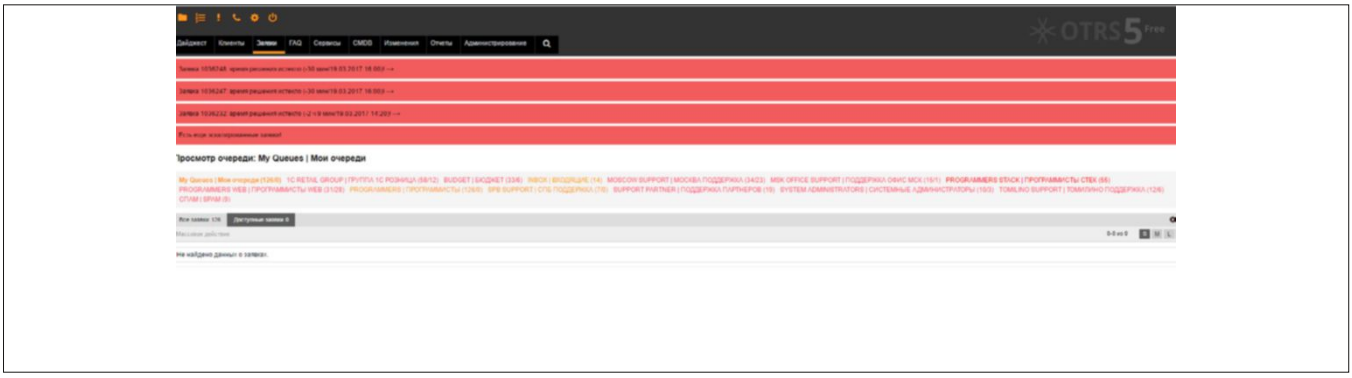
> MediaWiki
> DokuWiki
> Sharepoint Wiki

MediaWiki, естественно, вне конкуренции по технической реализации и набору функциональности, но для корпоративной базы знаний, как оказалось на практике, он подходит не лучшим образом:

> Файлы хранятся в БД, соответственно, конвертировать из Word нет никакой возможности.
> Достаточно долго и проблемно работать с рисунками.
> Трудно использовать внешние ресурсы.
> Внешний вид практически не кастомизируется от стандартного.

Как вы догадались, всех этих недостатков лишен движок DokuWiki [2], на мой взгляд, конечно. DokuWiki – единственный мне известный движок, который не использует хранение статей непосредственно в БД. Это можно рассматривать как недостаток для современных систем, но,

Рисунок 1. Система OTRS



как показала моя практика, для корпоративных баз знаний хранение статей в виде файлов имеет ряд неоспоримых преимуществ:

> **Простота конвертации из страниц других форматов.** Перед тем как у вас появится Wiki, вы наверняка уже имеете кучу инструкций в формате Word, PDF или аналогичных. Собственно, первичная задача конвертировать их в Wiki; вот тут внедрение базы знаний часто и заканчивается. В случае с DokuWiki вы просто преобразовываете все эти файлы в html, а затем применяете обычный конвертер Wiki-разметки.

Открытый API для создания ботов сделал мессенджер внутри отдела технической поддержки чрезвычайно популярным среди ИТ-команд

> **Простота формирования разделов.** При формировании разделов для переноса групп страниц и их переименования достаточно просто выполнить операции с файлами, что обычно делается быстро и проблем не вызывает.

> **Простота настройки.** В отличие от той же MediaWiki, DokuWiki ориентированана внутреннее использование, что позволяет сделать более простые настройки безопасности, более простое и понятное формирование разделов.

Мессенджер

Как это ни странно, мессенджер технической поддержке необходим. Для коммуникации с сотрудниками компании мы чаще всего используем Skype – не корпоративный, т.к. он вызывает много проблем с привязкой к домену, который есть не во всех удаленных узлах. Skype в последнее время стал намного хуже, и настало время подыскивать альтернативы.

Хорошей альтернативой был корпоративный Jabber – открытые протоколы позволяют достаточно удобно с ним взаимодействовать, кроме того, обслуживают мессенджер серверы внутри компании. Но для использования Jabber его нужно централизованно внедрять, в то время как Skype возникает «стихийно».

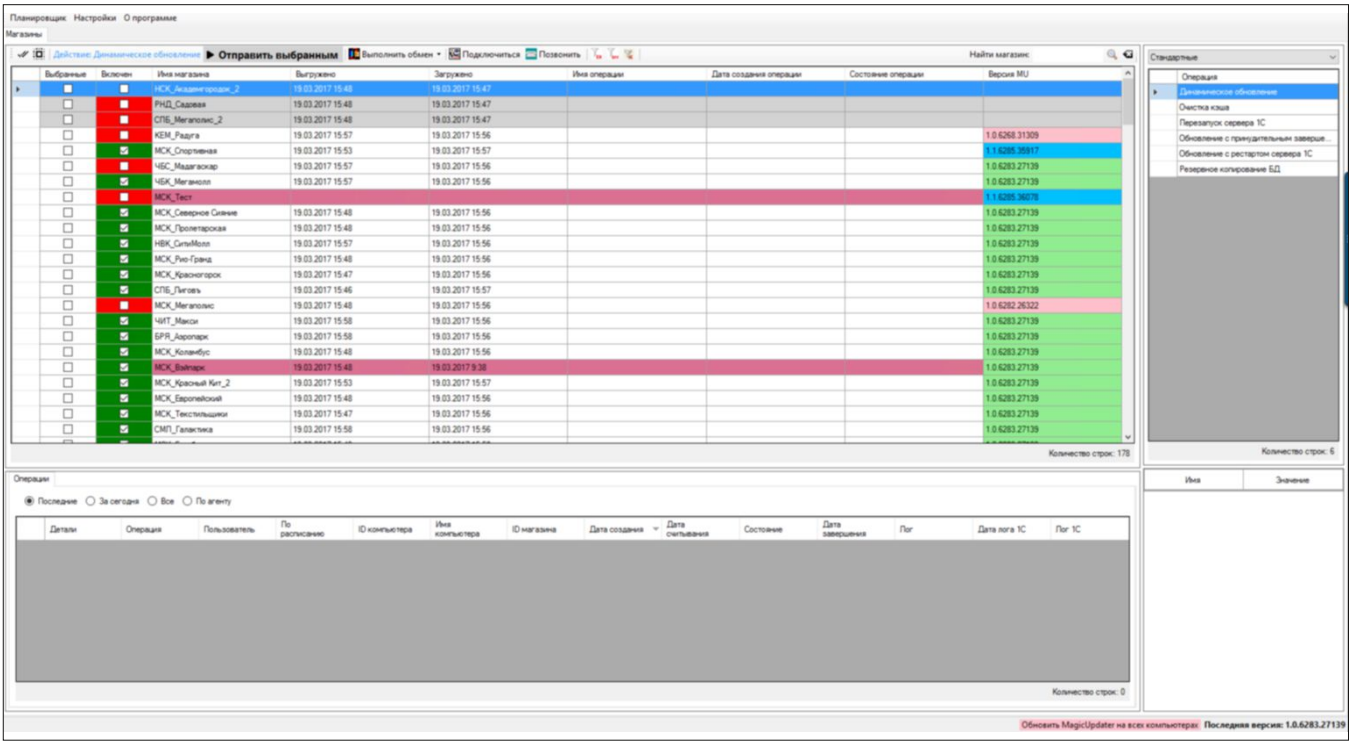
Существует также мессенджер внутри отдела – общий чат технической поддержки. В современном мире это стало уже «стандартом». Мы для него используем Telegram [3] – пожалуй, наиболее быстрый, удобный, безопасный мессенджер. Для него существует нормальный (быстрый, удобный, не подвисающий, в отличие от того же Viber) десктопный клиент. Возможности передачи больших файлов тоже бывают полезны. Самое главное, почему мы выбрали Telegram, – это возможность разрабатывать диагностические боты. Открытый API для создания ботов сделал данный мессенджер чрезвычайно популярным среди ИТ-команд.

Система удаленного управления рабочими станциями

Набор данных систем достаточно узок. Скорее всего с большей частью из них вы уже сталкивались:

> Ammy Admin
> Radmin
> TeamViewer
> TightVNC

Рисунок 2. Система собственной разработки для развертывания релизов, мониторинга и административных действий на рабочих станциях



Сразу оговорюсь, что основным требованием к этой системе была ее цена и открытость, поэтому TightVNC [4] была вне конкуренции. TightVNC – бесплатное решение, кроме того, в отличие от AmmyAdmin она имеет достаточно широкий набор функций (передача файлов, служебные комбинации клавиш и т.п.), позволяет создавать ярлыки с автоматическим подключением и работает как служба.

Для нее необходимо иметь внешний IP в удаленных точках; это, конечно, неприятно, но не так критично – практически каждый интернет-провайдер предоставляет данную услугу.

Система мониторинга рабочих станций

Для решения этих задач у себя мы используем систему собственной разработки. К сожалению, перечень задач, которые приходится решать на рабочих станциях, часто слишком широк. Объединение их в рамках Active Directory накладывает некоторые ограничения и не представляется возможным.

Кроме того, даже возможностей AD было бы недостаточно. Выполнять удаленно скрипты умеет множество разнообразного софта, включая, к примеру, корпоративный антивирус, но часто перед административным действием на рабочей станции нужно организовать диалог с пользователем, получить статус этого действия, выполнять цепочку действий – каждое последующее в случае успешного завершения предыдущего.

В нашем случае также важен единый перечень рабочих станций, которые находятся под управлением. Ежедневно появляются новые или закрываются существующие. Данный список актуален в системе 1С:Розница, в которой каждая рабочая станция – это узел обмена. Плотная интеграция с системой развертывания релизов, мониторинга и административных действий – одно из основных требований. Существующие системы, как правило, имеют закрытые протоколы, поэтому для подобной интеграции понадобилось разрабатывать индивидуальные решения.

Еще одной необходимостью была возможность разработки скриптов административных действий на языке C#. Он имеет достаточно мощные средства взаимодействия с ОС, к слову, намного большие, чем у того же PowerShell. Нормальный синтаксис и удобство отладки делают скорость разработки подобных скриптов куда более высокой, чем на традиционных скриптовых языках.

Поэтому мы реализовали административные действия в виде отдельных плагинов к данной системе. Действия из каждого плагина можно выполнить по расписанию, включить в очередь, посмотреть состояние, прочитать лог и т.п.

Таким образом, развертывание релиза – это просто последовательность операций, которые выполняются последовательно, каждая следующая – в случае неуспешного выполнения предыдущей:

- резервное копирование,
- динамическое обновление,
- условное статическое обновление (ожидающее подтверждения пользователя),
- безусловное статическое обновления (с завершением работы),
- статическое обновление с перезапуском сервера.

Естественно, в системе есть очередь операций – если она не была выполнена из-за выключенной рабочей станции, она выполнится при ее включении.

Установка дополнительного софта происходит примерно по тому же принципу. Резервное копирование – тоже одна из операций, которая добавлена в расписание и включена в процедуры обновления в самое начало.

> Плотная интеграция с системой развертывания релизов, мониторинга и административных действий — одно из основных требований к системе мониторинга

Мониторинг состояния процессора, памяти, жесткого диска и наличия интернет-соединения происходит просто при периодическом опросе агентов. В других показателях с рабочих станций из нашей практики не было необходимости.

Внешний вид системы представлен на рис. 2.

Система учета ИТ-оборудования

Как было упомянуто выше, 1С ITIL [5] мы не используем в качестве HelpDesk-системы из-за не очень подходящего веб-интерфейса, но есть задачи, для которых системы на платформе 1С просто незаменимы, – это учетные задачи внутри ИТ. В нашем случае это вопросы учета ИТ-оборудования. Для учетных задач 1С подготовлена как нельзя лучше, и 1С ITIL в частности. Решать задачи учета оборудования и конфигурации каждого ПК достаточно легко и удобно. Частично конфигурацию можно получить из AD или использовать тот же Everest. 1С ITIL поддерживает интеграцию и с тем, и с другим, но мы вполне обошлись ручными операциями.

• • •

Не возьмусь говорить, что тот набор систем, которые используем мы, самый верный и правильный, но повседневные задачи решает вполне успешно. Знать об их существовании, функциях бывает весьма полезно. Если не удается найти подходящую систему для ваших задач, не нужно бояться собственной разработки – самый простой путь не всегда самый верный. BOF

[1] Система HelpDesk OTRS – https://www.otrs.com/otrs-free-help-desk.

[2] Система базы знаний DokuWiki – https://www.dokuwiki.org/dokuwiki#.

[3] Мессенджер Telegram – https://telegram.org.

[4] Система удаленного управления TightVNC – http://www.tightvnc.com.

[5] Система 1С ITIL – http://1c-itil.ru.

Ключевые слова: HelpDesk, поддержка, администрирование.

Визитка

СЕРГЕЙ БОЛДИН,
системный администратор НЭК «Укрэнерго», bsergey2@gmail.com

«Раздача» программ в SCCM 2012 R2
Способ 2

В Configuration Manager имеется второй тип распространения программного обеспечения – Application (приложений). Рассмотрим отличия от типа Package (пакетного), новые возможности, настройки, достоинства и недостатки

В Configuration Manager централизованное распространение программного обеспечения можно осуществлять двумя способами – с помощью пакетов и приложений. Самое главное отличие между ними заключается в количестве возможностей. Для пакетов данный список является более ограниченным по сравнению с приложениями. Применяя второй способ «раздачи» программ и утилит, сотрудники предприятия получают дополнительные удобства в работе, ведь он больше нацелен на взаимодействие с конечным пользователем [1].

Примечание. В данной статье понятия «приложение» и «программа» не являются синонимами. Приложение здесь – это контейнер, «обертка» (как и в случае с Package), в которую и заворачивается программа или утилита с установочным EXE, MSI или другим файлом.

Package vs Application

Принцип распространения программ как у Package, так и у Application одинаков – это подготовка на сервере (создание «обертки», выбор установочного файла, прописывание ключей тихой установки, указание группы компьютеров или пользователей для распространения), доставка их на компьютер сотрудника (доставка «завернутого» контента на клиентскую машину в папку C:\Windows\ccmcache) и установка (процесс распаковки, то есть запуск установочного файла с применением ранее указанных настроек и ключей тихой установки).

Первое отличие этих типов – использование программ с определенными расширениями. Для Package характерны следующие:

- **exe** – файл запуска или установки программы;
- **vbs** – скриптовый файл;
- **bat** – командный файл;
- **cmd** – командный файл.

А для типа Applications такие:

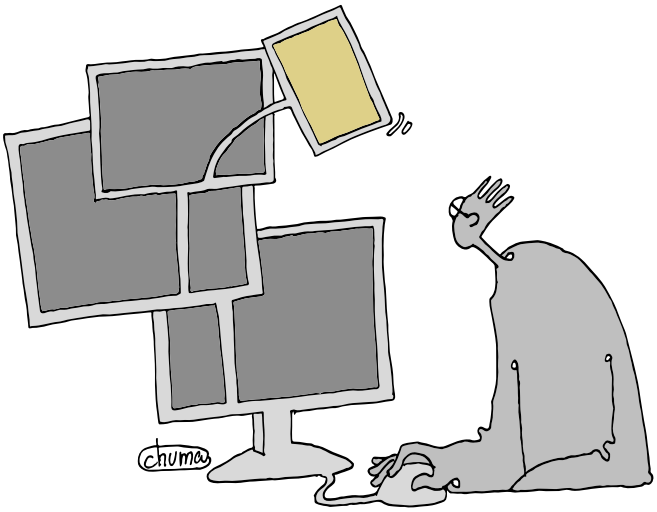
- **msi** – установочный файл (Windows Installer);
- **appx, appxbundle** – для Windows;
- **xap** – для Windows Phone;
- **cab** – для Windows Mobile;
- **ipa** – для iOS;
- **apk** – для Andriod;
- **sis** – для Nokia;
- **для веб-приложений;**
- **для Windows Application Virtualization 4 и 5;**
- **для Mac OS X.**

Второе отличие заключается в наличии нескольких этапов проверок (см. рис. 1) для приложений перед его развертыванием на машину сотрудника.

Следующие отличия относятся больше к преимуществам работы с приложениями:

- для выбора устанавливаемых программ пользователю доступен веб-каталог;
- можно настроить автоматическое обновление (замену) версий ранее установленных программ;
- указав ключи тихой деинсталляции, сотрудники смогут самостоятельно удалять программное обеспечение;
- в одно приложение можно добавить несколько программ, например, для мобильного устройства, компьютера и Application Virtualization.

Рисунок 1. Процесс развертывания приложения



Уровень работы с приложениями как минимум на один порядок выше, нежели с пакетами

Установка ролей

Веб-каталог представляет собой сервис, предоставляющий сотрудникам предприятия право на установку/удаление доступного программного обеспечения прямо из браузера. Но есть нюанс – приложения в веб-каталоге отобразятся только в том случае, если они будут распространяться для пользователя, а не на устройство.

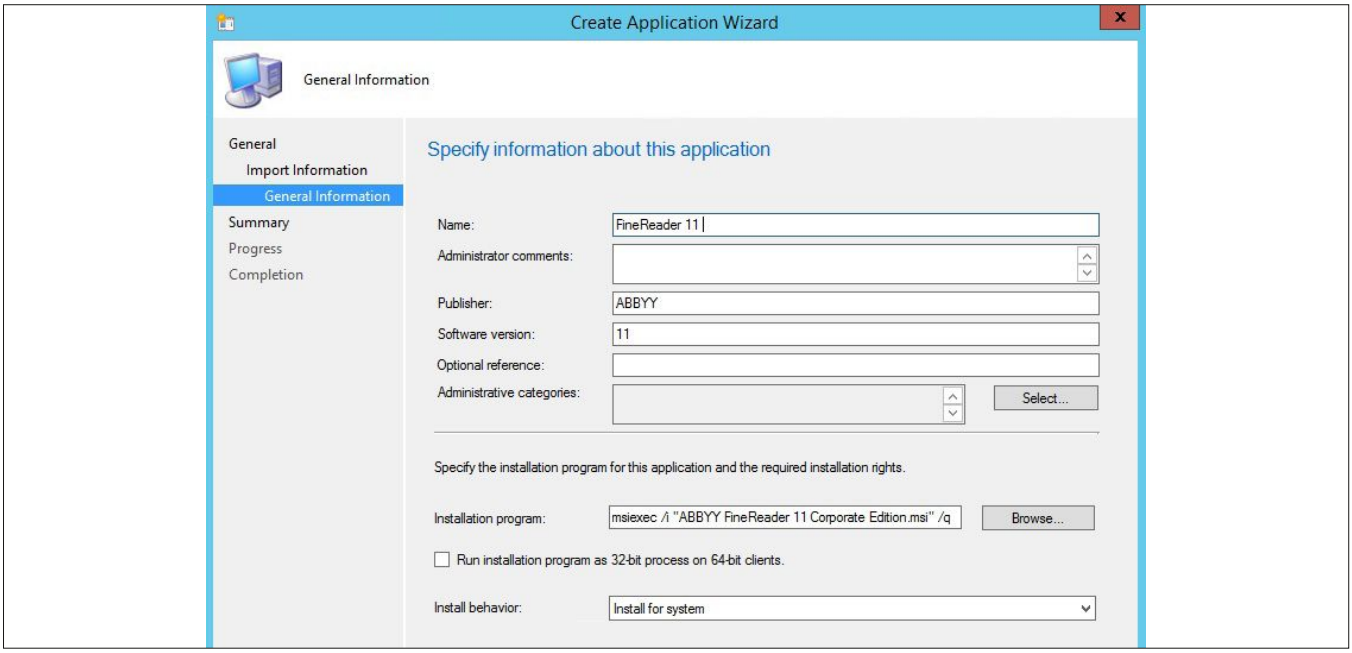
Чтобы такой сервис стал доступным, нужно установить на SCCM-сервере две роли – Application Catalog web service point и Application Catalog web server point [2]. Для этого проходим путь Administration → Overview → Site Configuration → ervers and Site System Roles, жмем правой кнопкой мыши и в меню выбираем пункт Add Site System Role, в списке активируем две соответствующие галочки. После этого веб-каталог будет доступен по адресу: http://<сервер>/CMApplicationCatalog.

Информация в каталоге обновляется по умолчанию каждые пять минут. Но данный интервал можно изменить, зайдя в Administration → Overview → Site Configuration → Sites, выбрав Settings → Site Maintenance и в появившемся окне в конце списка выбрав Update Application Catalog Tables.

Создание приложения на основе msi-файла

Когда программы имеют установочный файл с расширением msi, их упаковка в приложение занимает минимум времени, так как в этом случае можно положиться на мастера с автоматическим режимом конфигурирования.

Рисунок 2. Автоматическая настройка приложений



Для инсталляции программ с msi-файлом [3] нужно зайти в раздел Software Library, далее Overview → Application Management → Applications, нажать правой кнопкой мыши и в меню выбрать Create application. В появившемся мастере на первом шаге выбрать первый вариант Automatically detect information about this application from installation files. Далее следует выбрать тип файла msi и UNC-путь к нему. На следующем шаге необходимо проверить и по необходимости исправить заполненные поля: имя, ключи тихой установки программы и другое (см. рис. 2), затем Next → Next → Finish.

Теперь созданное приложение нужно распространить. Для этого на нем жмем правой кнопкой мыши и в меню выбираем пункт Deploy. В появившемся мастере выбираем нужную коллекцию пользователей, точку распространения, далее выбираем Available (означает необязательную установку, режим «доступно») и время установки. После этого в веб-каталоге появится новое приложение (см. рис. 3) для дальнейшей его инсталляции.

Создание приложения на основе exe-файла

Большая часть программного обеспечения включает в себя установочный exe-файл, для их упаковки в приложения придется пройти более длительный путь конфигурирования, так как в мастере автоматический режим не предусмотрен.

Итак, снова создаем приложение, на первом шаге мастера выбираем Manually specify the application information, на шаге General Information вписываем имя, на третьем шаге – Application Catalog – выбираем заранее подготовленную и соответствующую программе иконку, добавляем краткое описание программы и Next → Next → Finish. Тем самым мы создали болванку, или «обертку», в которую теперь нужно поместить программу.

Для упаковки программы на созданном приложении жмем правой кнопкой мыши и в меню выбираем пункт Create Deployment Type. На первом шаге мастера в выпадающем списке (в перечне доступных типов фалов) видим пункт Script Installer, его и выбираем. Затем вводим имя, указываем путь к папке, выбираем установочный файл и дописываем ключи тихой установки [4], а также тихого удаления (по необходимости).

На следующем шаге – Detection Method – требуется назначить правило проверки соответствующего критерия (требования). Здесь имеются три варианта: относительно файловой системы (папки или файлы), запросы в реестр и файлы MSI. Часто такая проверка используется для установленной программы, чтобы в дальнейшем ее обновить до новой версии или заменить другой. То есть если такая проверка имеет положительный результат, то процесс развертывания приложения далее выполняться не будет (см. рис. 1). Если же впервые программа будет устанавливаться, то можно создать и какое-то простое правило проверки, например, на существование папки Program Files. Для этого выбираем File System и указываем место расположения папки.

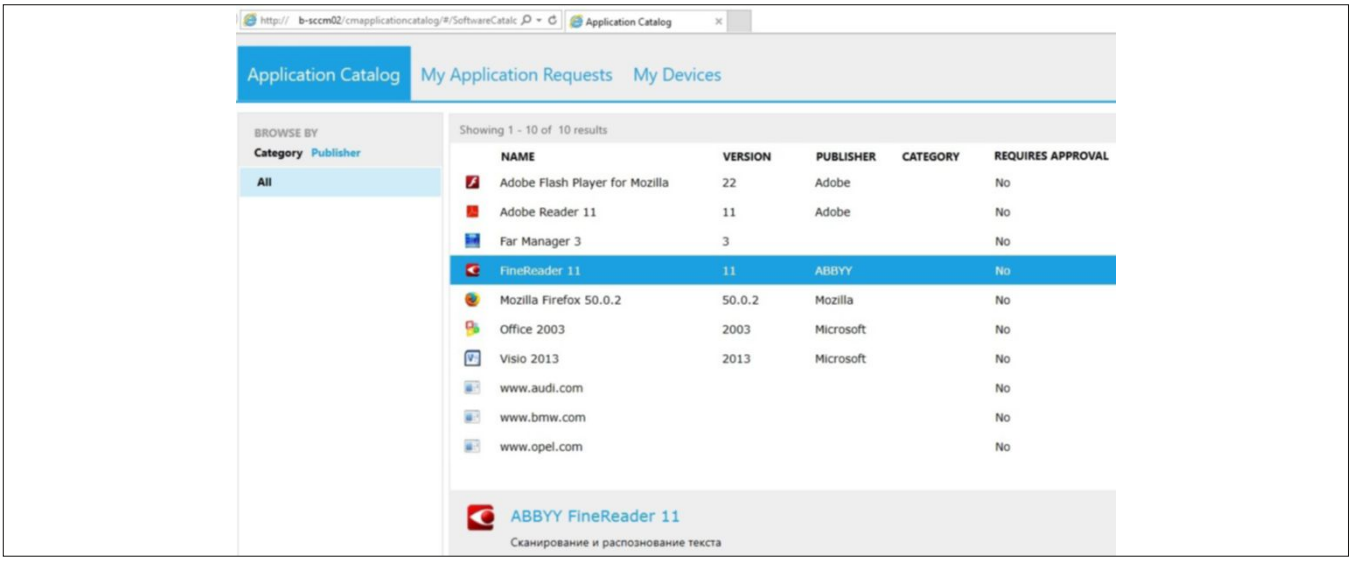
В User Experience мы даем предпочтение hidden-режиму, Whether or not a user is logged on и Install for System, что позволит нам отказаться даже от кратковременного появления каких-либо окон, развертывание приложения производить с полными правами и независимо от того, залогинен ли пользователь или нет.

На шаге Requirements можно задать параметры компьютеров, для которых будет происходить распаковка приложения, – это количество памяти, скорость процессора, ОС и другое. Например, укажем, что участвуют только машины с ОС Windows7 x32 и объемом памяти не менее 2 Гб (см. рис. 4). Если из стандартного набора правил ничего не подходит, то можно задать свои правила опросов [5] Active Directory, IIS, реестра, SQL, WMI, файловой системы и другие.

Последний шаг – Dependences – дает возможность указать приложения, которые будут дополнительно инсталлироваться вместе с настраиваемым, ведь некоторые программы перед своей инсталляцией требуют наличия каких-то компонентов, например Framework или Visual C++ Redistributable. Этот шаг позволяет не устанавливать зависимости заранее вручную.

После длительной конфигурации приложения вручную можно выполнить имитацию его установки, при которой

Рисунок 3. Каталог пользователя



не копируется контент на клиентскую машину, не берутся во внимание ключи тихой установки, а проверяются зависимости, требования и методы обнаружения. То есть имитация – это промежуточная проверка правильности настроек, применив которую системный администратор может понять, нет ли каких-то препятствий на пути к распространению приложения. Чтобы запустить ее, нужно на приложении нажать правой кнопкой мыши и в меню выбрать пункт Simulate Deployment.

Осталось созданное приложение распространить, после чего оно будет доступно в веб-каталоге пользователя (см. рис. 3) для дальнейшей установки.

Замена старой версии программы на новую

Разберемся с одной интересной и важной особенностью приложений – замена старой версии программы на новую [6]. Вручную переустанавливать каждый раз программы и утилиты на компьютерах сотрудников проблематично, даже имея удаленный доступ.

Чтобы автоматизировать процесс замены программ, нужно сначала создать новое приложение, упаковать в него программу с новой версией, затем зайти в параметры во вкладке Supersedence, нажать кнопку Add и указать приложение с ненужной программой (см. рис. 5). Если же активировать галочку Uninstall, то старое приложение будет удаляться, а новое – устанавливаться на его место, а не поверх него. Затем необходимо зайти в свойства Deployment Types и во вкладке Detection Method задать какое-либо условие, например указать существующую папку, файл, ветку реестра или другое. Последним действием является распространение нового приложения.

Сотрудник компании в веб-каталоге старой версии программы уже не увидит, а чтобы она исчезла с его компьютера, нужно будет нажать кнопку Install и процесс обновления/замены пройдет автоматически.

Создание контейнера для веб-приложения

Появление новых ярлыков в Пуске и/или на рабочем столе для многих не относится к удобству, ведь ссылки можно расположить прямо в браузере на панели Избранное. Однако в нашем предприятии достаточное количество сотрудников, у которых весь рабочий стол забит ярлыками и ссылки на сайты тоже должны быть в таком виде.

Веб-приложения представляют собой контейнер с ссылкой на ресурс сайта. Здесь не нужно выбирать какой-то файл и производить множество настроек. Достаточно только на первом этапе из списка выбрать тип Web Application и ввести адрес сайта, на втором шаге указать какую-то дополнительную информацию.

После этого созданное приложение остается только распространить. Веб-приложение отобразится в веб-каталоге (см. рис. 3). После его установки в меню Пуск появится ярлык, при запуске которого в браузере откроется указанный сайт.

При установленной ОС Windows 10 не получится удалить ярлык из меню Пуск или перетащить его на рабочий стол. Чтобы произвести какие-то манипуляции с ярлыком, нужно попасть в его месторасположение: C:\Users\<пользователь>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs.

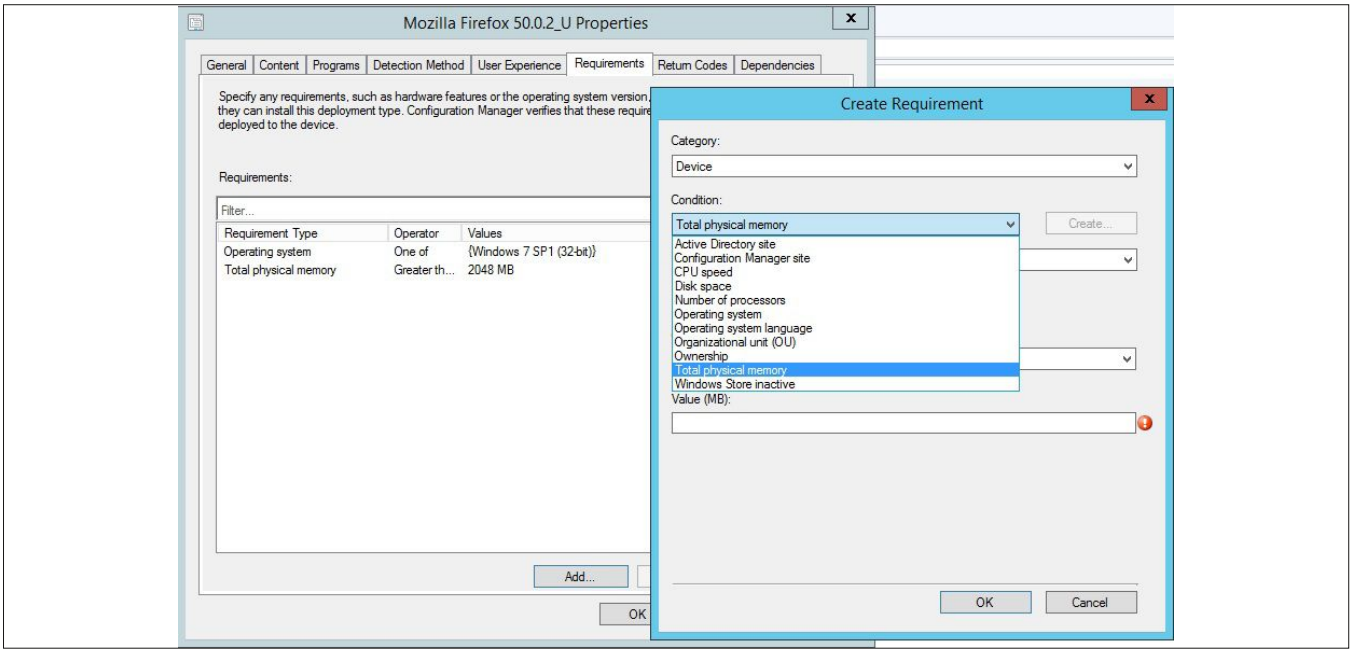
С веб-приложениями можно производить такие же операции, как и с приложениями на основе установочных файлов – устанавливать зависимые компоненты, например Silverlight или FlashPlayer, выставлять требования к компьютеру, заменять один ярлык (сайт) другим.

Экспорт/импорт приложений

Экспорт и дальнейший импорт приложений (как и пакетов) может пригодиться, когда нужно перенести приложения на другой сайт или переустановить свой.

Чтобы экспортировать приложение, нужно нажать правой кнопкой мыши, в меню выбрать пункт Export и указать

Рисунок 4. Настройка приложения с инсталляционным exe-файлом



путь для сохранения. В указанную папку попадет приложение в виде zip-архива, а также программа в первоначальном виде со всеми библиотеками, установочным и вспомогательными файлами, скопированная из места своего расположения.

История ревизий приложения

Когда производится конфигурирование приложения, каждое изменение записывается, и можно просмотреть его историю, а также при необходимости перейти на более нужный момент времени, то есть откатиться назад.

Чтобы просмотреть историю ревизий, нужно на приложении нажать правой кнопкой мыши и в меню выбрать Revision History.

Troubleshooting

При работе с приложениями в SCCM могут появиться какие-то проблемы. Ведь неверно сконфигурированные приложения приведут к тому, что пользователь получит ошибку о невозможности установить/удалить программу или утилиту.

Первым делом разобраться с проблемой ИТ-специалисту помогут лог-файлы AppDiscovery.log, AppEnforce.log, AppIntentEval.log, располагающиеся на клиентском компьютере в C:\Windows\Ccm\Logs.

Второй вариант поможет просмотреть состояния приложения [7]. Для этого нужно зайти в Monitoing → Overview → Deployments, где отображается все распространенное программное обеспечение, его цель или назначение (доступное или по требованию), тип (программа, приложение или последовательность задач), коллекции, действие (установка или удаление) и другое.

Достоинства: дополнительные возможности, которые удобны для конечного пользователя.

Недостатки: больше тратится времени на настройку приложений, чем при «завертывании» программ в пакет.

• • •

При использовании приложений системному администратору приходится осуществлять огромное количество настроек, в результате чего сотрудники предприятия сами могут инсталлировать нужные им программы прямо из браузера. Произведя конфигурацию на сервере SCCM по замене старой программы на новую, получаем значительную экономию времени за счет автоматизации такого процесса, а также отсутствие лишних вопросов и раздражающих телефонных звонков. Подводя итоги, можно с уверенностью сказать, что уровень работы с приложениями как минимум на один порядок выше, нежели с пакетами. EOF

[1] Описание приложений – https://www.navus.kz/microsoft/voprosy-i-otvety-po-system-center-2012-configuration-manager-sccm-2012-chast-6.html.

[2] Установка ролей – https://blog.it-kb.ru/2012/09/24/system-center-2012-configmgr-sccm-application-catalog/.

[3] Настройка приложений – https://technet.microsoft.com/ru-ru/library/gg682159.aspx, https://blog.eaglenn.ru/sozdanie-prilozheniya-v-system-center-configuration-manager-2012-r2.

[4] Болдин С. Тихая установка программ в SCCM. // «Системный администратор», № 1-2, 2017 г. – С. 24-27 (http://samag.ru/archive/article/3355).

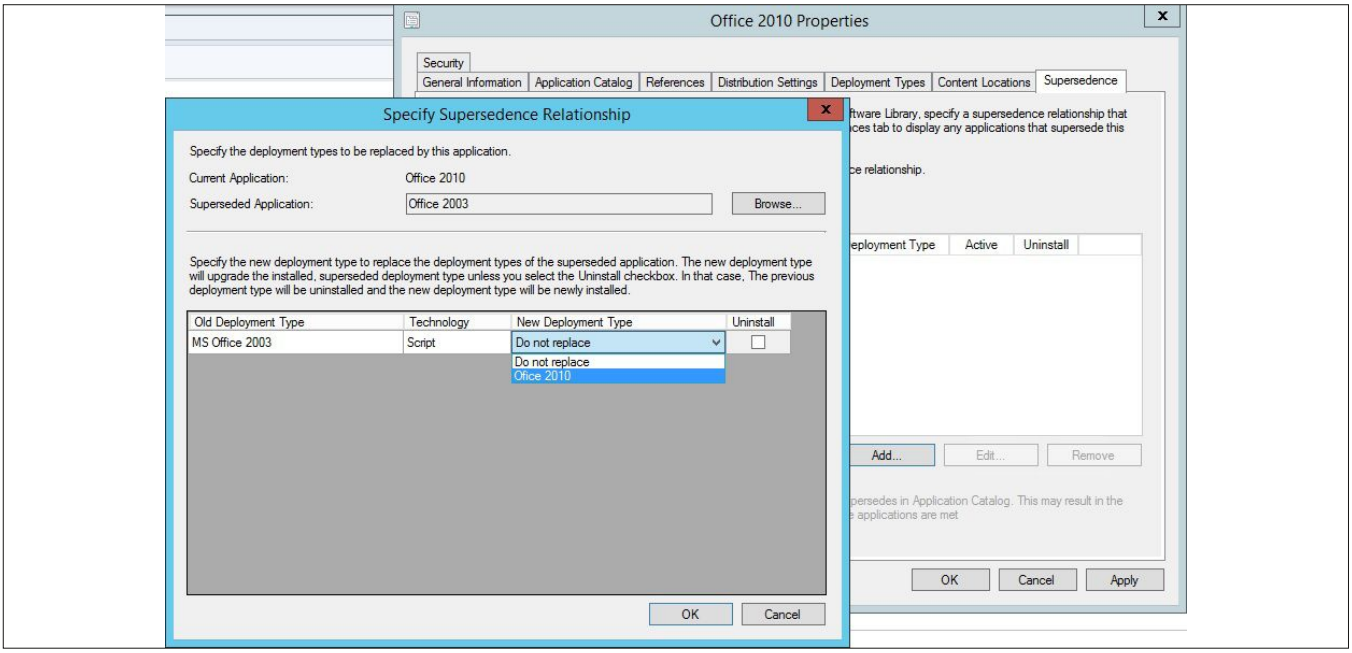
[5] Создание глобальных условий – https://technet.microsoft.com/ru-ru/library/gg682048.aspx.

[6] Замена приложений – https://technet.microsoft.com/ru-ru/library/gg682071.aspx; https://mwesterink.wordpress.com/2012/12/11/sccm-2012-superseding-applications-a-real-world-example.

[7] Мониторинг состояния приложений – https://technet.microsoft.com/ru-ru/library/gg682201.aspx.

Ключевые слова: пакет, программа, способ, роли, приложение, развертывание, установка, распространение, веб-каталог, файл, расширение, возможности, метод, зависимости, старая версия, тип, требования, обнаружение, правило, «раздача».

Рисунок 5. Настройка замены программ



Визитка

ЮРИЙ СЛОБОДЯНЮК,
О13 Netvision, инженер ИТ, yuri@yurisk.info

Самые частые ошибки в администрировании межсетевых экранов Check Point. Как их избежать

За 10 лет ежедневной работы с МСЭ Check Point мне довелось увидеть немало неисправностей. Разные версии, топологии, но что оставалось неизменным, так это неисправности вследствие ошибок самих администраторов. В статье расскажу о самых часто совершаемых ошибках и как их избежать

Удаление объекта, который используется

Особенно характерно для сисадминов, пришедших из мира MS Windows и привыкших подтверждать все сообщения системы на уровне предупреждения (Warning). Из всех ошибок эта, пожалуй, может оказаться самой губительной для вашей карьеры. Check Point позволяет удалить объект, который используется в политиках безопасности, часто даже предупредив о последствиях удаления. К сожалению, не все и не всегда читают/понимают эти предупреждения.

Моя рекомендация: получив сообщение о том, что объект используется, ни в коем случае не удалять его, а пройти по всем указанным в сообщении местам и удалить оттуда этот объект, пользуясь здравым смыслом, конечно. Дело в том, что при удалении объекта система автоматически заменяет его объектом Any, что в 99% случаев не то, что администратор хотел, чтобы произошло.

Для иллюстрации приведу пример из моего опыта. Клиент жаловался на безумно медленный интернет во всей организации: веб-страницы открываются с трудом, почта посылается/принимается с задержкой в полчаса. После нескольких проверок стало ясно, что Check Point грузит канал на все 100%, причем происходит это и при полностью отключенной внутренней сети.

Логи в SmartView Tracker показывали необычно большое количество исходящих соединений SSH на различные адреса во внешней сети. Посмотрев через CLI на содержание директорий этого МСЭ, обнаружил файлы с именами bruter.sh, uploader.sh и т.д., а также файлы больших размеров с именами фильмов, на то время идущих в кинотеатрах. Стало ясно, что МСЭ клиента взломали и используют как хранилище вареза и SSH-сканнер хостов. После просмотра журналов SmartView Tracker Audit (в новых версиях называется Management) стало понятно, что произошло.

Рисунок 1. Предупреждение об изменении правила

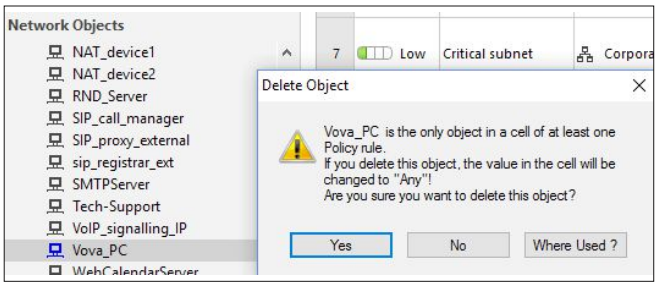


Рисунок 2. Правило политики безопасности

No.	Hits	Name	Source	Destination	VPN	Service	Action	Track
Limit Access to Gateways Rule (Rule 1)								
1	Med	Stealth	Vova_PC	Corporate-gw Management	Any Traffic	CPMI ssh https	accept	Log

Рисунок 3. Правило после изменения

No.	Hits	Name	Source	Destination	VPN	Service	Action	Track
Limit Access to Gateways Rule (Rule 1)								
1	Med	Stealth	Any	Corporate-gw Management	Any Traffic	CPMI ssh https	accept	Log

Было такое правило в политике безопасности, необходимое для работы админа с МСЭ: объекту Vova_PC разрешен доступ к МСЭ по протоколам ssh/https/CPMI (протокол соединения между МСЭ и SmartDashboard) (см. рис. 2).

> **Vova_PC** – хост во внутренней сети.
> **Corporate-gw, Management** – соответственно сам МСЭ и его SmartCenter.

Затем пришел на работу новый админ и по своей инициативе решил сделать «чистку» политики безопасности.

Одним из его действий было удаление объекта Vova_PC, содержащего адрес внутреннего хоста, принадлежавшего его предшественнику на этом посту. Хотя система предупредила его о возможных последствиях, а именно что этот объект используется в политике безопасности и после его удаления он будет заменен объектом Any (см. рис. 1).

Админ проигнорировал его и, подтвердив удаление, установил политику безопасности, что превратило это правило в правило, разрешающее доступ к МСЭ по протоколам ssh/https/CPMI отовсюду (см. рис. 3).

По плохому стечению обстоятельств аккаунт admin (с правами root) операционной системы МСЭ имел пароль qwe123. Судя по логам, взломали их Check Point меньше чем за час после изменения и установки политики безопасности с IP в Румынии. Им, можно сказать, «повезло», так как взломщики не поняли, куда попали, и не пошли дальше внутрь сети, просто использовали МСЭ как сервер Linux для раздачи вареза и брутфорса хостов в интернете.

А почему я назвал эту ошибку самой губительной для вашей карьеры? Потому что этот новый администратор там больше не работает.

Использование Dynamic Object в политике безопасности для блокирования доступа на веб-сайты

Ошибка, которая гарантированно перегрузит/завалит МСЭ, всегда происходит в такой ситуации: администратор МСЭ получает задание блокировать доступ к ресурсу в интернете, у которого нет постоянного IP. Check Point, конечно, умеет это делать, но… в соответствующем модуле – URL Filtering/App Control, требующем своей лицензии. Лицензия = деньги. Но что делать, если нет денег? Админ продолжает искать в SmartDashBoard, пока не находит объект Dynamic Object – как ему кажется то, что искал, так как позволяет использовать FQDN в его создании (см. рис. 4).

Рисунок 4. Создание нового динамического объекта

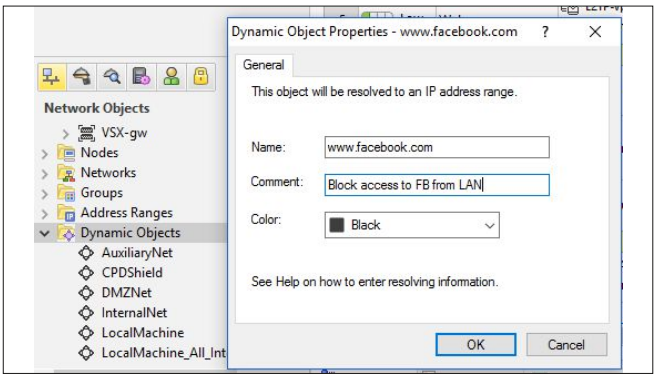
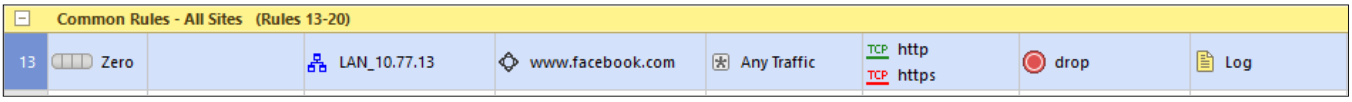


Рисунок 5. Правило с использованием динамических объектов



Admin создает его, конфигурирует согласно просьбе, добавляет правило в политике безопасности, в котором блокирует внутренней сети доступ к ресурсу в интернете по протоколам http/https (см. рис. 5).

Устанавливает политику на МСЭ… и, как правило, вся организация теряет интернет, а админ доступ к МСЭ. А происходит следующее: согласно этому правилу безопасности каждый пакет, потенциально подходящий под него, перехватывается, МСЭ посылает запрос серверу DNS, получает адрес IP (в примере www.facebook.com), вносит эту пару адресов (source IP/destination IP) в свои таблицы и только после этого выпускает пакет в интернет. Теперь умножьте этот процесс на десятки тысяч пакетов, проходящих через МСЭ, и получите CPU 100%.

Моя рекомендация: не использовать Dynamic Object вообще, так как он предназначен для других целей – для работы с удаленными МСЭ, не имеющими постоянного адреса, работы с SmartProvisioning и других ситуаций, довольно редко встречаемыми в основной работе.

Не проверять наличие свободного места на жестком диске

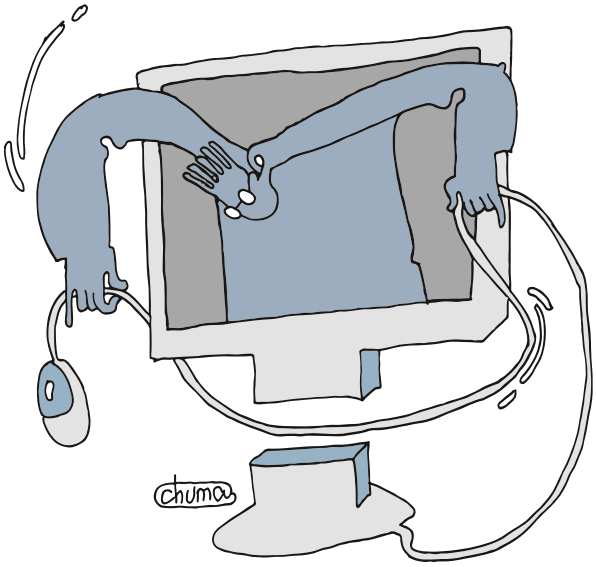
Первое, что я делаю, если есть проблема и надо делать дебаг, – проверяю наличие места на жестком диске. Дело в том, что проблемы с местом на диске могут выражаться в совершенно, казалось бы, не связанных поломках:

> Невозможно установить политику безопасности, а получаем ошибку типа:

```
Could not install policy, error 0x36748956 …
```

> Невозможно открыть логи в SmartView Tracker, или открываются, но получаем пустой лог.
> Невозможно обновить IPS/URL Filtering/App Control – и опять же выдает ошибку, которая ничего не говорит администратору.
> Невозможно подсоединиться к SmartCenter в SmartDashboard.
> Я могу продолжать и продолжать …

Тут надо помнить, что Check Point МСЭ основан на сервере Linux RHEL (а именно R77 на RHEL 5 и новые версии R80.10 и выше на основе RHEL 7). Как любому серверу, ему требуется свободное место на диске для исправной работы – для загрузки файлов обновлений из интернета,



Познакомимся с «граблями», на которые легко наступить при управлении сетевыми экранами Check Point

для обработки и консолидации логов, для шифрования/дешифрования файлов, для обработки внутренних баз данных, которые в большинстве своем файлы на диске.

Поэтому наличие достаточного свободного места критично, по своему опыту скажу, что не стоит опускаться меньше 1 Гб, особенно в разделе «/», где установлена сама операционная система.

Проверить наличие места можно командой (см. рис. 6):

```
df -h в expert mode
```

Использование простых паролей, легких для взлома

Это настолько элементарно, что вы можете не поверить, что это встречается в жизни – легкие пароли, да еще и пароли МСЭ? Неприятно удивлю, что не только встречается, но и нередко.

Обычно это начинается с интегратора, который устанавливает или обновляет версию существующего МСЭ и как всегда спешит на еще две-три установки в тот день в других местах.

И так как есть необходимость использовать пароль аккаунта операционной системы много раз во время установки и первичной конфигурации, то многие облегчают себе работу тем, что выбирают легкие для введения пароли, такие как qwe123/1q2w3e/123456, сказав себе: «нет проблем, как закончу, поменяю на более сложные», и, конечно, забывают сделать это. Я видел МСЭ, которые установили 10 лет назад еще в версии R55 с таким легким паролем и 10 лет после этого обновляли и обновляли, не меняя пароль, так как боялись потерять доступ к нему или трогать то, что работает.

Как я рассказывал выше, такие пароли могут легко привести к взлому самого МСЭ.

Поэтому моя рекомендация: изменить при установке нового МСЭ пользователя admin на другое имя – не бойтесь, ничего плохого не произойдет, или, если пользователь уже существует и боитесь его удалить, просто поменяйте его пароль на что-то длинное и очень сложное, сохраните этот пароль в программе или в месте, где вы храните все свои остальные пароли, и больше никогда им не пользуйтесь для входа в систему, а вместо этого создайте каждому администратору свой индивидуальный пользовательских аккаунт.

Забыть отключить акселерацию (SecureXL) перед началом дебага

Это распространенная ошибка, которая случалась и у меня, и у техподдержки самого Check Point. Когда со всех сторон давят на нас, чтобы решить проблему как можно скорее, естественна такая ошибка.

Когда-то это не было настолько важно, но сегодня 90% МСЭ используют компонент акселерации, называемый на языке Check Point «SecureXL». Этот компонент позволяет сократить время обработки проходящих через МСЭ пакетов тем, что пакеты, не требующие проверки модулем брандмауэра, рассматриваются и проверяются модулем SecureXL.

Например, если говорим о TCP, то создание сессии проходит полную проверку модулем брандмауэра, продолжение же трафика такой сессии передается модулю SecureXL. Как следствие, делая дебаг в снифере fw monitor, мы увидим только начало сессии TCP SYN/ACK. И, конечно, делать дебаг, видя только часть сетевого трафика, невозможно.

Поэтому первое, что надо сделать до начала какого-либо дебага в современных МСЭ, – это проверить, включен

Рисунок 6. Проверка наличия свободного места

```
smartcenterr77> expert
Enter expert password:

Warning! All configuration should be done through clish
You are in expert mode now.

[Expert@smartcenterr77:0]#
[Expert@smartcenterr77:0]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_splat-lv_current
                      7.0G  4.6G  2.9G  62% /
/dev/hda1             289M   24M  251M   9% /boot
tmpfs                 980M     0  980M   0% /dev/shm
/dev/mapper/vg_splat-lv_log
                      3.0G  524M  2.3G  19% /var/log
[Expert@smartcenterr77:0]#
```

ли этот компонент акселерации, и если да, то отключить его временно и включить после окончания дебага.

Кстати, обратите внимание, что сделать это можно двумя способами. Через cpconfig не делайте этого, так как это отменяет акселерацию на постоянной основе и вдобавок просит сделать рестарт всему МСЭ.

И весьма просто с командной строки следующим образом:

> проверить, включена ли акселерация:

```
fwaccel stat
```

> если включена, временно отключить:

```
fwaccel off
```

> по окончании дебага включаем обратно:

```
fwaccel on
```

Рисунок 7. Включение Database Revision Control

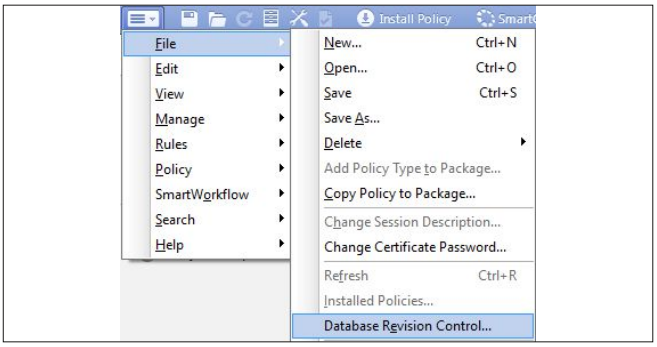


Рисунок 8. Отмечаем Create a new Database version...

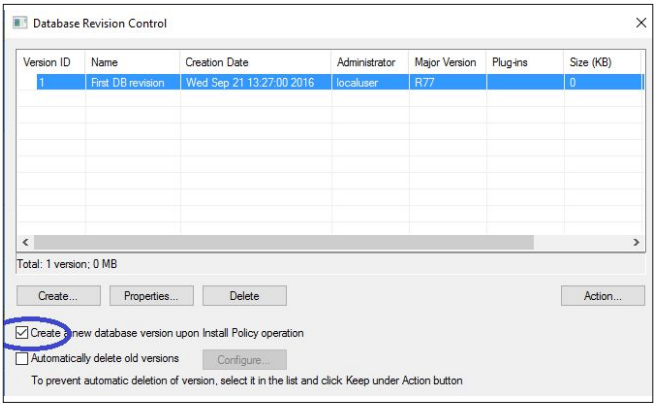


Рисунок 9. Лучше Reject заменить на Drop

Пометка: отключение акселерации увеличит нагрузку на процессор МСЭ, поэтому стоит проверить до того, что процессор не перегружен, и если перегружен, то уменьшить эту нагрузку до включения дебага.

Не пользоваться «страховкой» против ошибок конфигурации – Database Revision Control

С самого своего начала Check Point дает возможность сохранить конфигурацию для резервного копирования, включая все объекты и политику безопасности. Можно сделать копию в любой момент по желанию или установить автоматическое сохранение перед каждой установкой политики безопасности. К моему удивлению, не более 10-15% МСЭ, которые я видел, используют эту функцию.

И зря, это спасает ситуацию при случайном удалении объекта или правила или в случае возникновения проблем в сети, когда не ясно, какое изменение в МСЭ привело к этому и надо срочно сделать откат к проверенной политике.

Сам процесс восстановления прост до банальности – несколько кликов, и все вернулось к прежнему состоянию. Единственным недостатком можно назвать использование этими копиями места на диске, но даже тут можно установить, сколько копий хранить. Активируем функцию резервных копий Database Revision Control через Launch Menu → File → DataBase Revision Control (см. рис. 7).

И в открывшемся окне помечаем Create a new Database version upon Install Policy operation (см. рис. 8).

Для восстановления конфигурации нажмите Action → Restore Version…

Использовать Reject вместо Drop в правилах политики безопасности

Тут важно понимать разницу между двумя возможностями блокировки трафика. Reject не только блокирует пакеты, но и посылает инициатору уведомление об этом. В случае TCP это TCP Reset, что не только добавляет нагрузку на МСЭ, но и сообщает второй стороне, что их, собственно, заблокировал МСЭ. Drop, с другой стороны, блокирует пакеты, не давая потенциальному атакующему никакой информации. Не вижу никакой причины использовать Reject (см. рис. 9).

Перезапуск всего МСЭ, когда требуется перезапуск только Management/SmartCenter

Часто администраторы забывают, что программное обеспечение фильтрующего модуля МСЭ и программное обеспечение управления Management/SmartCenter – это два самостоятельных компонента, даже когда они установлены на одном сервере. Поэтому, когда по какой-либо причине

хотят перезагрузить SmartCenter, делают это перезагрузкой всего сервера.

Проблему это может и решит, но и весь файрвол перегрузит без всякой на то необходимости. Намного проще перегрузить только сервис SmartCenter командами в expert mode:

> Закрыть сервис:

```
cpwd_admin stop -name FWM -path "$FWDIR/bin/fw" ↵
         -command " fw kill fwm"
```

> Загрузить его снова:

```
cpwd_admin start -name FWM -path "$FWDIR/bin/fwm" ↵
         -command "fwm"
```

Не синхронизировать время МСЭ через ntp

Не раз мне приходилось участвовать в изнуряющем дебаге, чтобы позже понять, что время МСЭ и, соответственно, его логов ошибочное. МСЭ создает много логов – политики безопасности всех его сервисов (логи с расширением .elg), и это очень помогает в дебаге и анализе происшествий. Все логи, создаваемые МСЭ, имеют дату и время.

Когда часы сервера, на котором установлен МСЭ, отклоняются от точного времени, это делает логи ненадежными и даже вводящими в заблуждение.

Из моего опыта могу с уверенностью утверждать: не важно, насколько продвинутый и дорогой сервер, его часы будут отклоняться со временем. И не только это – отклонение оно нелинейное. То есть, если я вижу, что сегодня часы отстают на 15 минут, невозможно сказать, на сколько они отставали неделю или месяц назад. Были случаи, когда из-за этой неизвестной неточности логи становились бесполезными.

Решение: синхронизируйте МСЭ с внутренним или внешним сервером NTP. Возможна конфигурация двух серверов NTP – одного главного и одного вторичного. Сделать это можно или в Gaia GUI (System Management → Time → Set Time and Date, см. ниже), или на командной строке (см. рис. 10).

```
smartcenterR77> set ntp server primary 13.13.13.1 version 2
smartcenterR77> set ntp server secondary 23.23.23.1 version 2
smartcenterR77> save config
```

Не проверять резервные копии конфигурации на работоспособность

Check Point имеет несколько способов сохранить конфигурацию – через Gaia GUI, на командной строке, сделать одноразово или автоматически.

Важнее всего, конечно, сохранять SmartCenter, содержащий все объекты политики безопасности и аккаунты пользователей. Конфигурацию модуля тоже стоит сохранять, но не так критично, в нем интересуют только адреса и таблица маршрутизации. Часто SmartCenter установлен на VMware или другой инфраструктуре виртуализации. Тогда легче – просто периодически делать Snapshot. Но если бэкап делается средствами самого Check Point или своими скриптами, то обязательна проверка работоспособности этих копий. То есть делать в лабораторных условиях полное восстановление МСЭ из резервной копий конфигурации.

Иллюстрация о том, как важно проверять копии. Обратился к нам как-то клиент с проблемой SmartCenter – ОС не поднимается с ошибками диска.

Не страшно, на этом SmartCenter раз в неделю запускался бэкап, который по завершении Check Point поднимал по FTP на внутренний сервер. Клиент с его интегратором установили новый физический сервер, инсталлировали МСЭ без конфигурации и начали пробовать восстановить из backup саму конфигурацию с помощью утилиты upgrade import самого МСЭ.

Кинули upgrade import один файл бэкапа – ошибка, файл поврежден и не может быть открыт. Второй файл – то же самое, третий файл… в общем, они перебрали более 20 файлов бэкапа, и все оказались повреждены.

В итоге клиент был вынужден пригласить специалистов, которые восстановили данные SmartCenter напрямую с железа проблематичного диска.

Так что, если вы делаете бэкап средствами Check Point или любой другой автоматизированной системой, включая собственные скрипты, всегда проверяйте свой бэкап. Нужно не забывать, что Check Point – это сервер Linux, и, чтобы сделать бэкап, как любой другой сервер, он собирает важные файлы, сжимает и архивирует их в архив tgz и сохраняет на внешнем носителе или сервере. И тут много шансов чему-то пойти не так – в папке /tmp не было достаточно места для работы tar/gzip, принимающей сервер FTP повредил файлы при получении и так далее. И в большинстве таких сбоев мы не получаем об этом никакой индикации.

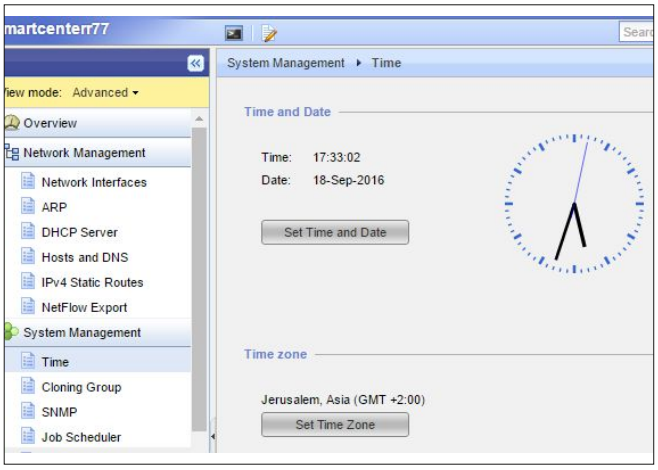
• • •

Спасибо за внимание, надеюсь, статья познакомила вас с теми «граблями», на которые легко наступить, поддавшись видимой дружелюбности и легкости управления сетевыми экранами Check Point, и сделала вас более уверенными в работе с ними. EOF

[1] Официальный сайт Check Point – https://www.checkpoint.com/ru.

Ключевые слова: администрирование, МСЭ, Check Point, Linux.

Рисунок 10. Настройка NTP



Облака на горизонте:
как и почему нужно переходить на облачные платформы

В наши дни существует немало организаций, которые недоверчиво относятся к переводу своих систем на облачные сервисы. В основном их беспокоят вопросы безопасности

Это неудивительно – в новостях появляется масса сообщений о взломах и утечках критических данных. Раскрываются планы правительств, публикуются личные фотографии известных людей… Такие события удерживают от перехода в облака даже самых уверенных руководителей.

Однако, по мере развития облачных технологий, становится все сложнее отказаться от их очевидных преимуществ. Специалисты, принимающие решения, и передовые руководители уже понимают – сегодня некоторые системы гораздо эффективнее держать в облаке.

Одна из таких систем – корпоративная система объединенных коммуникаций, позволяющая быстро перевести компанию с устаревшей АТС и задействовать новейшие технологии связи.

Система объединенных коммуникаций в облаке призвана упростить ИТ-процессы и сократить рутинное администрирование. 3CX предлагает свою облачную АТС с философией «zero-admin» (нулевое администрирование). Наши разработки позволяют штатному системному администратору развернуть облачный сервер АТС для компании за считанные минуты.

PBX Express – это онлайновый инструмент (Мастер конфигурирования), позволяющий интеграторам 3CX и конечным пользователям запустить полнофункциональную АТС за восемь несложных шагов. В АТС присутствуют самые современные технологии объединенных коммуникаций, поддерживаются распространенные IP-телефоны, VoIP-шлюзы и операторы – без неудобств, связанных с традиционной телефонией.

Максимальная экономия

Главное достоинство облачных систем, которое, в общем, вытекает из простой установки и нулевого обслуживания – гораздо более низкая стоимость. Экономят все – и небольшие компании, и средние бизнесы. А в крупных организациях появляется еще одно преимущество – лицензирование системы не по количеству сотрудников, а по параллельным коммутациям (вызовам).

Небольшие организации особенно оценят корпоративные функции, возможности объединенных коммуникаций, отличную систему безопасности и минимальное администрирование – по цене, составляющей небольшую часть цены альтернативных решений. А если прибавить к этому высвободившиеся рабочие часы ИТ-персонала, установка 3CX будет иметь долгосрочный экономический эффект для всего бизнеса.

Поскольку облачные системы не требуют приобретения и установки физических серверов, компании не нужно выделять крупные ресурсы на создание инфраструктуры, которая, к слову, может быть совсем непростая. Вполне подойдет существующая сеть, при этом вы не только экономите на затратах, но и принципиально сокращаете время запуска системы. Это позволяет переключиться на новую систему практически незаметно.

В случае с 3CX PBX Express интегратор или пользователь просто заполняет поля в Мастере первоначальной настройки и получает рабочую функциональную облачную UC-систему быстрее, чем сходить на обед.

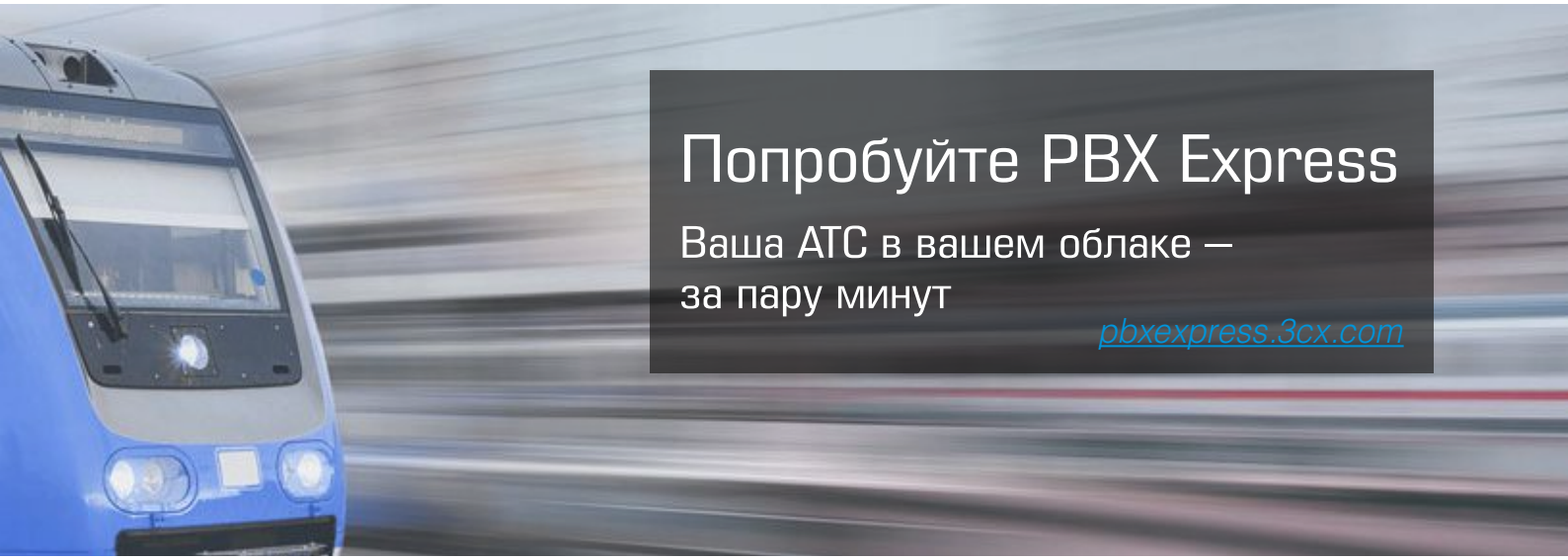
Гибкая настройка и контроль своих данных

Традиционная облачная модель подразумевает, что вы полагаетесь на возможности облачного провайдера и доверяете ему свои данные.

Современные технологии АТС предполагают не только быструю установку и простое сопровождение, но и гибкую функциональность, и, безусловно, полную конфиденциальность ваших данных.

К сожалению, многие провайдеры облачных АТС ограничивают возможности конфигурирования системы для пользователя. Но заказчикам важно иметь полный доступ к настройкам для реализации своих собственных решений, и в то же время максимально ограничить доступ к конфигурации для третьих лиц.

Идеальный баланс – это облачный хостинг полнофункциональных АТС в виде индивидуальных экземпляров, полностью изолированных друг от друга, на облачном провайдере по вашему выбору. Благодаря такому подходу каждый заказчик получает свой индивидуальный экземпляр АТС, изолированный от некорректных действий других пользователей.



Возможность настройки АТС – то свойство, которое позволяет создавать коммуникационные решения, оптимальные именно для вашей организации. Гибкость системы как раз и определяет долгосрочный экономический эффект от внедрения, поскольку позволяет перенастраивать АТС под изменяющиеся нужды организации без дополнительных вложений: выбирать облачного провайдера, подключать различные телефоны и SIP-транки по вашему выбору и т.п. Вы также не связаны никакими долгосрочными контрактами, вынуждающими вас доплачивать за каждого нового пользователя.

3CX предлагает максимальную гибкость настроек, не ограничивая вас количеством пользователей. Расширяйте АТС вместе с бизнесом, переносите ее из облака на локальный сервер и обратно – все в ваших руках.

Объединенные коммуникации для всех

Современная компания должна использовать последние технологические новшества, ведь это залог успешной конкуренции. В качестве коммуникационной системы недопустимы компромиссы – независимо от того, где она работает – локально или в облаке.

Выбирая облачное UC-решение, ИТ-директор должен разобраться, настолько ли оно функционально и надежно, как заявлено, и насколько его функциональность отличается от локальных аналогов. Просто протестируйте работу самых важных для вас функций АТС в облаке и сравните с локальными альтернативами.

Система 3CX имеет широкие возможности объединенных коммуникаций: видеоконференции, индикацию присутствия, чат и т.д. Она содержит модули интеграции с популярными CRM-системами, повышая производительность сотрудников и лояльность клиентов. Все это работает сразу «из коробки», после завершения Мастера PBX Express.

Почему 3CX

Любая UC-система, на которую стоит обращать внимание, должна обладать тремя преимуществами: высокая надежность, безопасность, простота внедрения – независимо от того, где она расположена.

Если один из этих факторов отсутствует, все остальные преимущества теряют свою ценность. 3CX в облаке или локально предлагает интегратору надежное и весьма простое в сопровождении UC-решение. При этом она обладает возможностями коммуникационной системы корпоративного класса.

Корпорации выбирают решение, поддерживаемое постоянными обновлениями с самыми последними технологическими инновациями. Это гарантирует, что коммуникационная система будет расти вместе с компанией, реализуя новые бизнес-модели.

Выбирая UC-решение, организации нужно принять во внимание уже существующую инфраструктуру и установленное программное обеспечение. Облачная АТС 3CX сохраняет вложения в инфраструктуру, предлагая средства совместимости с имеющимися IP-телефонами, SIP-шлюзами и транками и уже работающими CRM-приложениями.

Переходим на PBX Express

PBX Express исключает сложный, затратный и длительный процесс внедрения. Пользователи и интеграторы запускают АТС за несколько минут, пройдя восемь шагов Мастера первоначальной настройки. После этого система сразу готова к работе. Выбирайте международный хостинг: Google Cloud, Amazon, OVH или Openstack; используйте свой собственный хостинг; пробуйте систему на демо-хостинге 3CX. А если вы устанавливаете 3CX для клиента, используйте PBX Express для развертывания 3CX под своим облачным аккаунтом интегратора.

И напоследок – после запуска АТС вы получаете бесплатную систему 3CX PBX Edition на восемь одновременных вызовов. Бесплатная лицензия действует год и обеспечивает поддержку DNS-имени сервера 3CX, бесплатный доверенный SSL-сертификат, одного SIP-транк, пяти одновременных участников веб-конференции, бесплатные обновления системы и техподдержку через наш форум. ADV

Дополнительная информация: www.3cx.ru
Попробуйте PBX Express: pbxexpress.3cx.com

Визитка

СЕРГЕЙ БОЛДИН,
системный администратор НЭК «Укрэнерго», bsergey2@gmail.com

Корпоративный Skype

Переход с MS Lync 2013 на Skype for Business

Skype for Business – это обновленная версия системы MS Lync 2013, в которой главным новшеством является одинаковый интерфейс со Skype, а основной функционал остался без изменения

В центральном офисе произошло обновление продукта MS Lync 2013 до версии Skype for Business 2015 (SfB). Такие действия предстоит проделать и в нашем региональном представительстве, ведь сервер Lync развернут в отдельном (ресурсном) лесу для связи с филиалами по более быстрым каналам, нежели с головным офисом [1].

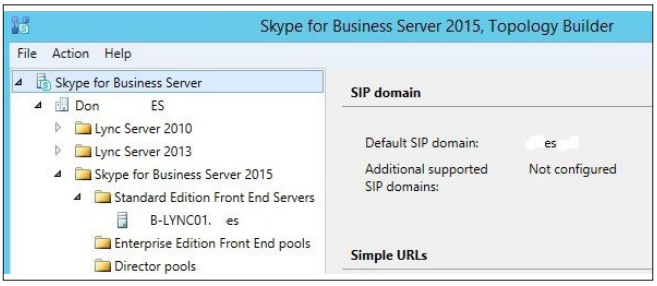
С имеющимися доработками Skype for Business можно ознакомиться на официальном сайте [2].

Одни сотрудники получили разрешение иногда работать удаленно из дома, другие стали чаще ездить в длительные командировки, третьи предоставляют консультационные услуги, а связь по телефону не всегда удобна для обеих сторон. В связи с появлением новых задач появились и потребности добавлять внешние контакты – из личного Skype в корпоративный.

Обновление до новой версии

Первым делом обновляется топология (по рекомендациям делается это на отдельном сервере). Для этого запускаем Skype for Business Server Deployment Wizard (мастер установки), жмем Install Administrative Tools (установить административные утилиты), затем запускаем Skype for Business Server Topology Builder (построитель топологии) и выбираем xml-файл с топологией, с появившимся сообщением «обновить» соглашаемся. В результате появится новый пункт Skype for Business 2015 (см. рис. 1) [3]. Для завершения первого этапа публикуем топологию, проделав путь «Action (Действие) → Topology (Топология) → Publish (Опубликовать)».

Рисунок 1. Изменения в построителе топологии



Затем следует остановить сервисы Lync. Сделать это можно в командной строке:

```
Stop-CsWindowsService
```

На следующем шаге запускаем установочный файл SfB и ожидаем, пока пройдет весь процесс обновления (удаление ролей, отсоединение базы данных, установка и настройка основных компонентов, установка средств администрирования, присоединение и обновление базы данных, включение реплики, установка ролей и другое).

По окончании включаем сервисы из командной строки:

```
Start-CsWindowsService
```

или в мастере развертывания SfB и проверяем их работоспособность, зайдя в «Computer Management (Управление компьютером) → Services and Applications (Сервисы и приложения) → Services (Сервисы)». В моем случае служба Skype for Business Server Front End не запускалась, пока я не применил все обновления для новой Skype-системы и для Windows Server 2012 R2 (KB2982006).

Настройка федераций

Настройка федераций [4] в системе MS Lync\Skype for Business позволяет сотрудникам разных компаний, а также временным/удаленным сотрудникам (фрилансерам) осуществлять между собой связь для ведения совместной работы. Будем считать, что сервер Edge для внешних соединений настроен (это тема отдельной статьи) и правильно функционирует.

Для управления совместной работой между пользователями Skype и Skype for Business заходим в раздел Federation and External Access (Федерации и внешний доступ), в закладке External access policy (Внешняя политика доступа) активируем необходимые галочки. В моем случае выбраны все (см. рис. 2).

Обновление с Lync 2013 до Skype for Business проходит достаточно легко

Далее в закладке Access Edge Configuration (конфигурация пограничного доступа) необходимо настроить политику взаимодействия с внешним миром, также выбрав нужные флажки (только вторую оставляем пустой):

- **Enable federation and public IM connectivity** – разрешить федерацию и подключение к общедоступной системе обмена мгновенными сообщениями;
- **Enable partner domain discovery** – разрешить обнаружение домена партнера;
- **Send archiving disclaimer to federated partners** – отправлять федеративным партнерам заявление об отказе относительно архивации;
- **Enable remote user access** – разрешить удаленный доступ пользователей;
- **Enable anonymous user access to conferences** – разрешить анонимный доступ пользователям к конференциям.

Осталось удостовериться в наличии поставщика SIP – Skype. Переходим в закладку SIP Federated Provides (Федеративные поставщики SIP). Если такой отсутствует, то необходимо его добавить с помощью командной строки PowerShell таким образом:

```
New-CsPublicProvider -Identity Skype ↵
         -ProxyFqdn federation.messenger.msn.com ↵
         -IconUrl https://images.edge.messenger.live.com/ ↵
         Messenger_16x16.png ↵
         -NameDecorationRoutingDomain msn.com ↵
         -NameDecorationExcludedDomainList ↵
         "msn.com,outlook.com,live.com,hotmail.com" ↵
         -Enabled $true -EnableSkypeIdRouting $true ↵
         -EnableSkypeDirectorySearch $true
```

Результат можно проверить, введя командлет Get-CsPublicProvider (см. рис. 3).

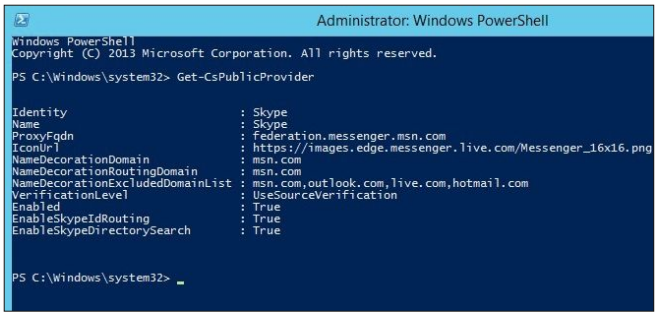
На этом конфигурирование серверной части завершилось. Теперь перейдем непосредственно к клиентам Skype for Business.

После обновления сервер SfB сам будет определять несоответствие версий клиентской части и автоматически осуществлять переход на «правильную» версию. После этого сотрудники увидят интерфейс скайпа – цветовое оформление, фото в кружке, элементы управления, такие же смайлики, статус присутствия.

Чтобы внести на клиенте корпоративного Skype контакт из личного, нужно знать SkypeID (логин) коллеги либо его учетную запись Microsoft [5]. Если таковая информация имеется, то справа жмем на кнопку «+», в появившемся меню проходим «Добавить контакт не из моей организации →

Рисунок 2. Настройка федераций

Рисунок 3. Вывод данных о поставщике



Skype», в строке поиска вводим почтовый ящик собеседника или SkypeID и ожидаем, пока в личном Skype примется приглашение на посланный запрос.

После появления контакта в списке выводится информация «откуда он взят», также виден статус присутствия (см. рис. 4), работники могут обмениваться текстовыми сообщениями с такими же звуковыми эффектами, изменять место вывода оповещения (угол экрана) о входящем звонке или сообщении, осуществлять аудиовидеозвонки. В SfB также можно искать внешние контакты Skype по отображаемому имени или по номеру телефона.

Утилита Profiles for Lync [6], предназначенная для переключения между разными учетными записями (серверами), после обновления сервера Lync до более новой версии работает без проблем и вмешательства в перенастройку не требует.

Несмотря на имеющиеся положительные изменения в федерации, присутствует большое количество недоработок при совместном использовании Skype и Skype for Business:

- не поддерживается пересылка файлов;
- в обоих Skype фото (аватарки) контакта не отображается;
- не получится предоставить коллеге доступ к своему рабочему столу;
- нет возможности осуществлять групповые голосовые звонки;
- нельзя принимать участие в конференциях или собраниях;
- не всегда выходит второй раз добавить внешний контакт Skype, ранее удаленный из SfB. Это было выявлено практическим путем и в документации не упоминается.

Если углубиться в работу федерации, то можно увидеть, что у Microsoft имеется дополнительный домен skypeids.net, который необходим для более упрощенного подключения внешних контактов Skype, не используя учетную запись Microsoft, а только SkypeID [7].

Достоинства: обновление с Lync 2013 до Skype for Business проходит легко.

Недостатки: много нереализованных функций при совместной работе Skype и SfB, о которых официально говорится в документации, а также имеются недоработки, о которых просто умалчивается.

• • •

И даже зная, что функционал федерации на данный момент работает нестабильно, решение по обновлению Lync 2013 является оправданным, т.к. в компании запрещено использовать личный Skype, а также потому, что система Скайп для бизнеса является коммерческой, нацелена на крупный корпоративный сектор и следующая версия может избавить от имеющихся недоработок. Хоть недостатков при работе SfB с внешними контактами из Skype достаточное количество, но нас даже устраивают звонки тет-а-тет и ведение с ними переписки, с этим пока проблем не выявлено. EOF

[1] Болдин С. Настройка MS Lync 2013. Часть 1 // «Системный администратор», №9, 2016 г. – С. 52-57 (http://samag.ru/archive/article/3273).

[2] Новшества в Skype for Business – https://technet.microsoft.com/ru-ru/library/dn933785.aspx.

[3] Обновление сервера Lync – https://blog.eaglenn.ru/poshagovoe-obnovlenie-lync-2013-do-skype-for-business-server-2015.

[4] Настройка федераций – https://technet.microsoft.com/ru-ru/library/dn440170.aspx, https://alekssh.com/2015/05/18/s4b-fed-skype , https://technet.microsoft.com/ru-ru/library/gg398161.aspx.

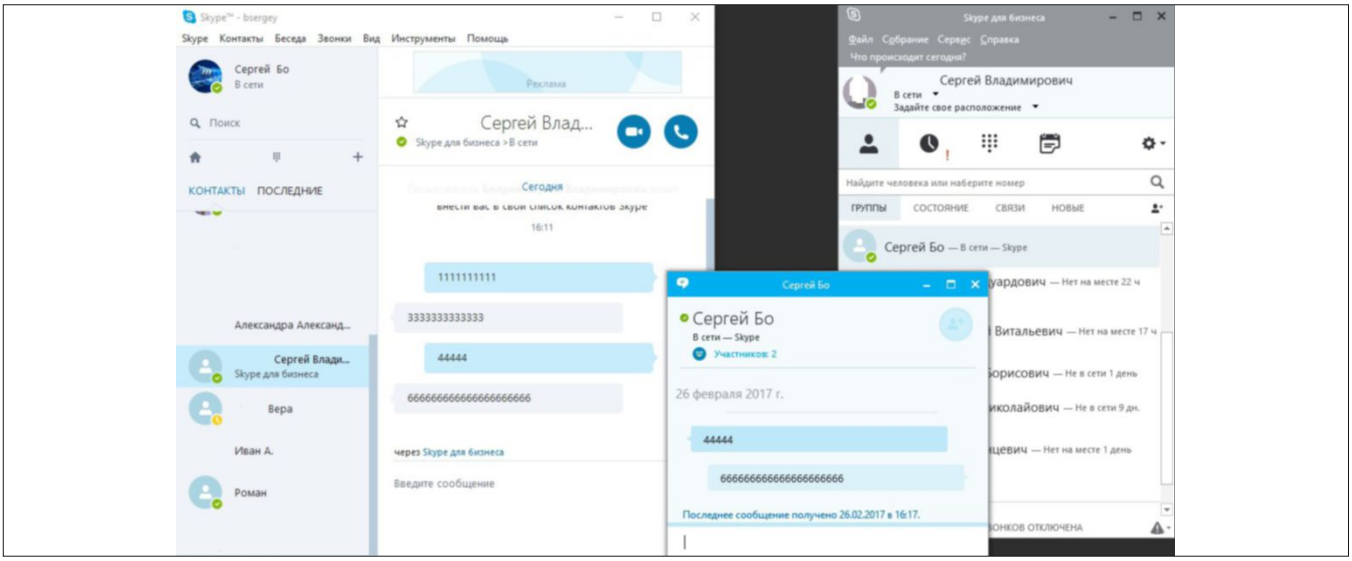
[5] Описание учетной записи Microsoft – https://support.skype.com/ru/faq/FA12059/chto-takoe-uchetnaya-zapis-maykrosoft.

[6] Описание утилиты Profiles for Lync – https://gallery.technet.microsoft.com/office/Profiles-For-Lync-2013-1b193329.

[7] Дополнительный домен для федераций – https://www.veritech.ru/blog/zvonki-iz-lync-na-obychnye-akkaunty-skype.html?tmpl=component&print=1, http://blog.enowsoftware.com/solutions-engine/lync-skype-for-business-to-skype-connectivity.

Ключевые слова: настройка, обновление, Skype, контакты, добавление, федерации, сервер, версия, командлет.

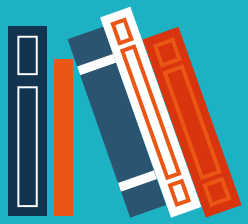
Рисунок 4. Добавление контактов Skype в S4B



Платформа для хранения и агрегации профессиональных знаний, аналитики и экспертных материалов IT-отрасли.

Prof-IT / WIKI РАЭК

Платформа запускается в бета версии в апреле 2017



При реализации используются средства государственной поддержки, выделенные в качестве гранта в соответствии с распоряжением Президента Российской Федерации от 05.04.2016 No 68-РП и на основании конкурса, проведенного Общероссийской общественной организацией «Российский союз ректоров».

Визитка

ВАЛЕРИЙ МИХЕИЧЕВ,
эксперт Oracle, СПАО «Ингосстрах»,
Valery.Mikheitchev@ingos.ru

Опыт работы в Oracle с таблицами, содержащими большие LOB-столбцы

Рассмотрим вопросы работы в Oracle с таблицами, в которых имеются большие LOB-столбцы. Особое внимание уделим особенностям интервального секционирования в Oracle 11g и Oracle 12c для таблиц, содержащих LOB-столбцы

Существует несколько типов LOB-объектов:
- **BLOB** – двоичный большой объект;
- **CLOB** – символьный большой объект;
- **NCLOB** – национальный символьный большой объект;
- **BFILE** – внешний двоичный файл.

LOB-объекты делятся на внутренние и внешние.

Внутренние большие объекты Internal LOB – хранятся в базах данных (БД), к ним относятся BLOB, CLOB и NCLOB. Внутренние большие объекты могут быть постоянными или временными:
- **постоянные LOB** – создаются как столбцы таблицы БД командой Create таблицы, при этом данные LOB могут храниться как в самой таблице, так и вне таблицы;
- **временные Temporary LOB** – создаются для использования только в пределах конкретного приложения. Для создания временного LOB используется процедура dbms_lob.createtemporary.

Внешние большие объекты External LOB – вид данных, который хранится в файлах операционной системы, вне базы данных, а в базе данных на них хранятся ссылки (локаторы). Внешние LOB используют тип данных BFILE.

Каждый LOB-объект состоит из двух частей:
- **локатор-указатель,** который специфицирует местонахождение контента (т.е. местонахождение данных LOB-объекта);
- **контент** – набор двоичных или символьных байтов, составляющих LOB.

Инструментом работы с LOB выступает пакет DBMS_LOB. Он предоставляет методы манипулирования внутренними и внешними LOB. В данной статье рассматриваются только постоянные внутренние LOB-объекты.

Сегменты LOB-объектов Oracle

Для объектов Oracle, таких как таблица, индекс, LOB-объект и др., создается структура, называемая сегментом. При этом имеется особенность: при создании сегментов для таблицы и индекса создается один сегмент, в то же время для каждого LOB-столбца таблицы создается не один, а два сегмента. Один сегмент – это LOB-сегмент для хранения данных и второй – это LOB-индексный сегмент. LOB-индексный сегмент создается автоматически при создании LOB-сегмента и служит для навигации по LOB-сегменту. Увидеть созданные сегменты объектов Oracle (при наличии данных) позволяет Oracle представление DBA_SEGMENTS. Это представление замечательно тем, что можно увидеть (в том числе в текущий момент) число байт в объекте Oracle.

LOB-объекты не секционированных таблиц

Порядок работы с обычными не секционированными таблицами, имеющими LOB-столбцы, рассмотрим на примере таблицы AIF.SERVICEMSGXML, функционирующей в нашей системе. В таблице имеются два LOB-столбца: XMLOUT и XMLIN – оба типа CLOB, при этом таблица будет размещаться в табличном пространстве AIFDATA.

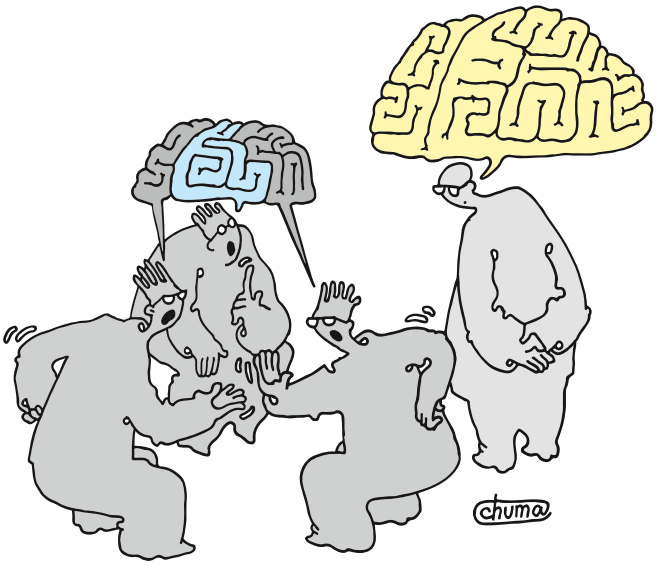
Команда создания таблицы может иметь простой вид:

```
CREATE TABLE AIF.SERVICEMSGXML
( ISN           NUMBER,
  UPDATED       DATE,
  XMLOUT        CLOB,
  XMLIN            CLOB
)
TABLESPACE AIFDATA;
```

В процессе создания таблицы AIF.SERVICEMSGXML командой Create в ней автоматически для каждого LOB-столбца таблицы создается конструкция LOB (имя LOB-столбца) STORE AS, где важным элементом является STORE AS.

Таким образом, в результате выполнения команды Create получится следующая структура таблицы:

```
CREATE TABLE AIF.SERVICEMSGXML
(  ISN           NUMBER,
  UPDATED       DATE,
   XMLOUT       CLOB,
   XMLIN           CLOB
)
```



Рассмотрим особенности интервального секционирования таблиц с LOB-столбцами

```
LOB (XMLOUT) STORE AS (
  TABLESPACE  AIFDATA
  ENABLE      STORAGE IN ROW
  CHUNK       8192
  )
LOB (XMLIN) STORE AS (
  TABLESPACE  AIFDATA
  ENABLE      STORAGE IN ROW
  CHUNK       8192
  )
TABLESPACE AIFDATA
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;
```

Как видно, LOB-объекты находятся в том же табличном пространстве AIFDATA, что и сама таблица. При этом в конструкции STORE AS автоматически создается ряд атрибутов, например, таких как TABLESPACE, ENABLE STORAGE IN ROW и CHUNK (их расшифровка приведена ниже).

Задание табличного пространства для LOB-столбцов таблицы

Как было показано выше, по умолчанию LOB-объект для каждого LOB-столбца создается в том же табличном пространстве (ТП), что и таблица, в которой находится LOB-столбец. При этом весьма важный момент: LOB-сегмент и LOB-индекс всегда находятся в одном и том же ТП.

Вместе с тем порой целесообразно помещать LOB-объект в ТП, отличное от ТП таблицы. При этом LOB-объекты таблицы могут размещаться в различных ТП (т.е. каждый LOB-столбец может иметь свое ТП). Для размещения LOB в других ТП необходимо в команде Create создания таблицы вручную прописать конструкцию STORE AS. Начиная с Oracle 11g можно использовать новую конструкцию STORE AS SECUREFILE в команде создания таблицы (ее особенность рассматривается ниже).

Замечание. Конструкцию STORE AS можно указать только при создании таблицы, т.е. эту фразу нельзя вставить в процессе модификации таблицы.

Так, в таблице AIF.SERVICEMSGXML было принято решение, что один LOB-объект, построенный на столбце XMLOUT, попадет в ТП с именем LOBTBS, а второй, построенный на столбце XMLIN, – в ТП с именем LOBTBS2. В результате команда Create таблицы примет вид:

```
CREATE TABLE AIF.SERVICEMSGXML
(  ISN            NUMBER,
  UPDATED        DATE,
  XMLOUT         CLOB,
  XMLIN             CLOB
)   TABLESPACE ADATA
LOB (XMLOUT) STORE AS
      ( TABLESPACE LOBTBS
        ENABLE       STORAGE IN ROW
        CHUNK     16384)
LOB (XMLIN) STORE AS
      ( TABLESPACE LOBTBS2
        ENABLE   STORAGE IN ROW
        CHUNK   8192 );
```

В команде Create таблицы не только указали другое ТП для LOB-объектов, но еще изменили значения атрибута CHUNK со стандартного 8192 на другое значение 16384.

LOB-объекты (созданные для них сегменты) этой таблицы можно увидеть из представления DBA_LOBS запросом:

```
Select * From DBA_LOBS Where owner='AIF' ↵
         and table_name = 'SERVICEMSGXML';
```

В результате получаем:

```
COLUMN_NAME        SEGMENT_NAME                 TABLESPACE_NAME
INDEX_SEGMENT_NAME
XMLOUT             SYS_LOB0001677384C00014$$  LOBTBS
SYS_IL0001677384C00014$$
XMLIN              SYS_LOB0001677384C00015$$  LOBTBS2
SYS_IL0001677384C00015$$
```

Как видно, в именах сегментов имеются символы SYS_LOB для LOB-сегментов и SYS_IL для индексных сегментов.

Используя представление Oracle DBA_SEGMENTS (при наличии данных в LOB), можно получить размер

в байтах, занимаемых LOB-сегментами и LOB-индексными сегментами. Например, для LOB-столбца XMLOUT число байтов, занимаемых сегментом, определяется запросом, указанным ниже. При этом в запросе в поле segment_name подставляется имя сегмента, полученного из представления DBA_LOBS.

```
Select owner, segment_name, bytes, tablespace_name ↵
         From DBA_SEGMENTS Where segment_name = ↵
         'SYS_LOB0001677384C00014$$';
```

Для работы с LOB удобным является запрос, в котором используются оба указанных выше представления: DBA_SEGMENTS и DBA_LOBS. Данный запрос покажет для каждого LOB-столбца таблицы не только имена их LOB-сегментов и LOB-индексов, но и их размеры в байтах.

```
Select s.owner, d.table_name, d.column_name, ↵
         s.segment_name, s.segment_type, s. bytes, ↵
         s.tablespace_name From DBA_SEGMENTS s, DBA_LOBS d ↵
         Where s.segment_name in (d.segment_name,d.index_name) ↵
         and s.owner=d.owner and d.owner='AIF' ↵
         and d.table_name = 'SERVICEMSGXML' order by 3;
```

```
COLUMN_NAME        SEGMENT_NAME                SEGMENT_TYPE
BYTES              TABLESPACE_NAME
XMLIN              SYS_LOB0001677384C00015$$   LOBSEGMENT
655360             LOBTBS2
XMLIN              SYS_IL0001677384C00015$$    LOBINDEX
65536              LOBTBS2
XMLOUT             SYS_LOB0001677384C00014$$   LOBSEGMENT
29746003968        LOBTBS
XMLOUT             SYS_IL0001677384C00014$$    LOBINDEX
25165824           LOBTBS
```

Иногда необходимо посмотреть объем, занимаемый самой таблицей (без LOB-объектов). Это можно увидеть запросом:

```
Select s.owner, s.segment_name, s.segment_type, s.bytes, ↵
         s.tablespace_name From DBA_SEGMENTS s ↵
         Where owner='AIF' and segment_name = 'SERVICEMSGXML';
```

```
OWNER     SEGMENT_NAME      SEGMENT_TYPE      TABLESPACE_NAME
BYTES
AIF       SERVICEMSGXML     TABLE             AIFDATA
32505856
```

Как видно, сама таблица составляет порядка 32 505 856 байт, в то время как один из LOB-столбцов занимает 29 746 003 968 байт. В силу чего таблица была размещена в табличном пространстве ADATA, а LOB-объекты – в других более значительных по объему табличных пространствах LOBTBS и LOBTBS2.

Атрибуты конструкции STORE AS

Фраза TABLESPACE. Указывает, в каком табличном пространстве создается LOB-сегмент (соответственно и его LOB-индексный сегмент).

Фраза ENABLE STORAGE IN ROW. Создается по умолчанию, что означает, что LOB размером до 4000 байт будут храниться в самой таблице (сегменте таблицы), а LOB, которые больше 4000 байт, будут храниться в LOB-сегменте вне таблицы. При DISABLE STORAGE IN ROW все значения LOB попадают в LOB-сегменты, т.е. вне таблицы (что успешно используется в наших таблицах).

При этом предпочтительнее ставить ENABLE STORAGE IN ROW, если в основной массе случаев размер LOB меньше 4000 байт, т.е. LOB умещаются в таблице. В отсутствии кэширования LOB каждое обращение к LOB (чтение или запись) задействует физическую операцию ввода-вывода. Таким образом, хранение LOB в таблице позволяет существенно сократить операции физического ввода-вывода на извлечение, запись и модификацию LOB. При этом ENABLE STORAGE IN ROW позволяет также уменьшить число логических операций ввода-вывода, обусловленных обращением к LOB-индексу.

Замечание. Для принятия решения, какой метод принять: ENABLE STORAGE IN ROW или DISABLE STORAGE IN ROW, может использоваться функция GETLENGTH пакета DBMS_LOB.

Данная функция определяет размер LOB в байтах для BLOB и BFILE или длину LOB в символах для CLOB. Например, для столбца XMLIN типа CLOB число символов, занимаемое соответствующими строками таблицы SERVICEMSGXML, определяется запросом

```
Select DBMS_LOB.GETLENGTH(XMLIN) From AIF.SERVICEMSGXML ↵
         order by 1 desc;
```

Фраза CHUNK. Определяет размер порции данных, размещенных в LOB. В отличие от таблицы, где минимальной единицей размещения данных является блок, в LOB данные хранятся в виде порций – логически непрерывного набора блоков, являющихся минимальной единицей размещения LOB. Таким образом, каждое хранимое значение LOB потребляет минимум один CHUNK. Размер CHUNK задается кратным размеру блока данных db_block_size базы данных и определяется из запроса:

```
Select value from v$parameter where name ='db_block_size'.
```

Обычно CHUNK задается равным db_block_size, например 8192, но может быть равным 16384, 24576, 32768 и т.д. В приведенной выше таблице один из LOB имеет значение 16384.

При этом значение CHUNK больше, чем 8192, должно быть обоснованным, поскольку, с одной стороны, это может повысить производительность работы с LOB, а с другой – может привести к не эффективному расходованию ТП в силу его частичного заполнения. При выборе CHUNK целесообразно сначала определить процент строк, которые умещаются в 8192. Если этот процент составляет большинство, то CHUNK принимается значение 8192, если нет, то значение 16384 и т.д.

Ниже предлагается запрос, позволяющий оценить распределение размеров LOB-строк в байтах по таблице:

```
Select DBMS_LOB.GETLENGTH(XMLOUT) bytes, count(1) cnt ↵
         from AIF.SERVICEMSGXML m group by ↵
         DBMS_LOB.GETLENGTH(XMLOUT) order by 2 desc;
```

Фраза NOCACHE. По умолчанию принимает значение NOCACHE, т.е. запрещает кэшировать LOB. Поскольку LOB, как правило, велики, то в большинстве случаев целесообразно отключать кэширование LOB. CACHE включает

кэширование LOB. При этом кэширование относится только к LOB-объектам, не затрагивая саму таблицу.

Фраза LOGGING. По умолчанию имеет значение LOGGING, которое говорит, что изменения данных в LOB-объектах записываются в redo-журнал для восстановления данных LOB в случае сбоя базы. Однако, поскольку данные в LOB-объектах обычно велики, может быть включен режим NOLOGGING, позволяет отключить формирование redo-информации.

В Oracle 11g структура STORE AS получила дальнейшее развитие, так появилась возможность сжимать данные LOB и исключить их дублирование. Для этого вместо конструкции STORE AS пишется конструкция STORE AS SECUREFILE. В ней дополнительно к существующим атрибутам добавились новые атрибуты, среди них:

Фраза COMPRESS. Задает режим сжатия LOB-объектов: HIGH – высокое сжатие , MEDIUM – среднее (по умолчанию) и LOW – низкое. Сжатие происходит внутри LOB, каждый LOB сжимается независимо от других. При этом сжатие LOB не зависит от сжатия таблиц. Так если задали режим сжатия таблицы, то это не приведет автоматически к сжатию ее LOB и наоборот. При выборе режима сжатия надо учитывать, что оно требует работы процессора, так в режиме LOW хотя и получаем минимальное сжатие, однако потребуется намного меньшее потребление процессорных ресурсов, кроме того, обеспечивается более высокая скорость процесса.

Фраза DEDUPLICATE. Запускает режим дедубликации. В этом режиме повторяющиеся значения в LOB заменяются указателями, что экономит табличное пространство.

Выводы. При создании таблицы и задании фразы STORE AS следует оценить, что эффективнее: хранить LOB в таблице или вынести их из таблицы (режим ENABLE или DISABLE), обосновать целесообразность кэширования LOB и определить более эффективный размер CHUNK.

Модификация атрибутов конструкции STORE AS

Ряд атрибутов STORE AS могут быть добавлены или модифицированы в любой момент после создания таблицы командой ALTER TABLE (не модифицируются такие атрибуты, как CHUNK и ENABLE/DISABLE STORAGE IN ROW).

Например, для столбца XMLIN таблицы SERVICEMSGXML модификация атрибутов NOCACHE на CACHE осуществляется по команде ALTER:

```
ALTER TABLE AIF.SERVICEMSGXML MODIFY LOB (XMLIN) (CACHE);
```

Для режима STORE AS SECUREFILE можно модифицировать или добавлять новые атрибуты. Например, добавить DEDUPLICATE и модифицировать COMPRESS с LOW на HIGH по команде:

```
ALTER TABLE AIF.SERVICEMSGXML MODIFY LOB (XMLIN) ↵
         (DEDUPLICATE);
ALTER TABLE AIF.SERVICEMSGXML MODIFY LOB (XMLIN) ↵
         (COMPRESS HIGH);
```

Секционированная таблица с LOB-столбцами

В секционированных таблицах с LOB-столбцами (так же как и для не секционированной таблицы) создаются по два сегмента для каждого LOB-столбца таблицы.

Рассмотрим особенности секционирования в Oracle 11g и 12c на примере таблицы AIF.SVLOG с двумя LOB-столбцами с именами REQUESTHEADER и REQUESTPARAMS. LOB-сегменты с LOB-индексным сегментами можно увидеть также из представления DBA_SEGMENTS (для каждого LOB-столбца):

```
OWNER  TABLE_NAME  LOB_COLUMN_NAME  SEGMENT_NAME
AIF    SVLOG       REQUESTHEADER    SYS_LOB0001163299C00018$$
AIF    SVLOG       REQUESTHEADER    SYS_IL0001163299C00018$$
AIF    SVLOG       REQUESTPARAMS    SYS_LOB0001163299C00019$$
AIF    SVLOG       REQUESTPARAMS    SYS_IL0001163299C00019$$
```

Секционированные таблицы с LOB-столбцами создаются по команде Create, используя те же конструкции SRORE AS или STORE AS SECUREFILE для описания свойств LOB-столбца.

Следует заметить, что при секционировании таблицы каждой секции таблицы будет соответствовать LOB-секция и LOB-индексная секция для каждого LOB-столбца таблицы. Например, при двух LOB-столбцах в таблице AIF.SVCLOG каждой секции таблицы будут соответствовать две LOB-секции и две LOB-индексные секции для каждого LOB-столбца. Так для секции PT_SVLOG_1 таблицы для LOB-столбца REQUESTHEADER создаются LOB-секция SYS_LOB_P9769 и LOB-индексная секция SYS_IL_P9806, аналогично для той же секции, но другого LOB-столбца REQUESTPARAMS создаются соответственно SYS_LOB_P9843 и SYS_IL_P9880 секции:

```
TABLE_NAME  PARTITION_NAME  COLUMN_NAME      LOB_PART_NAME
LOB_INDPART_NAME
AIF.SVLOG   PT_SVLOG_1      REQUESTHEADER    SYS_LOB_P9769
SYS_IL_P9806
AIF.SVLOG   PT_SVLOG_1      REQUESTPARAMS    SYS_LOB_P9843
SYS_IL_P9880
```

Особенности интервального секционирования для таблиц с LOB-столбцами

При интервальном секционировании (т.е. при использовании фразы Interval, доступной для Oracle 11g и выше), при вводе данных автоматически создаются новые секции.

Вместе с тем при интервальном секционировании таблиц с LOB-столбцами появились некоторые особенности. Особенности интервального секционирования таблицы с двумя CLOB-столбцами рассмотрим на примере AIF.SVLOG. Таблица размещается в ТП AIFDATA, ключ секционирования – столбец CREATED, секционирование по дням.

Команда Create для создания таблицы имеет вид (ряд столбцов для сокращения опущены):

```
CREATE TABLE AIF.SVLOG
( ISN    ,
  REQUESTHEADER      CLOB,
  REQUESTPARAMS      CLOB,
  CREATED            DATE)
TABLESPACE AIFDATA
PARTITION BY RANGE (CREATED)
INTERVAL (INTERVAL '1' DAY)   STORE IN (IMAGE)
(PARTITION  PART_01012017  VALUES LESS THAN ↵
         (TO_DATE('01.02.2017','DD.MM.YYYY'))
LOB (REQUESTHEADER) STORE AS SECUREFILE (
          TABLESPACE  IMAGE
          DISABLE STORAGE IN ROW
          COMPRESS MEDIUM
```

```
        DEDUPLICATE )
LOB (REQUESTPARAMS) STORE AS SECUREFILE (
        TABLESPACE  IMAGE
        ENABLE STORAGE IN ROW
        COMPRESS HIGH
        DEDUPLICATE )
  )
ENABLE ROW MOVEMENT;
```

Замечание. Структура команды Create данной таблицы может использоваться как типовое решение при интервальном секционировании таблиц с LOB-столбцами, поскольку в ней отображены наиболее характерные и новые решения, появившиеся начиная с Oracle 11g.

До использования интервального секционирования разнос секций таблицы и секций с LOB-объектами успешно реализовывался

В таблице используется конструкция STORE AS SECUREFILE, что позволяет ввести режим сжатия LOB-столбцов COMPRESS MEDIUM и HIGH, а также режим DEDUPLICATE для исключения дублирования значений LOB-данных. Кроме того, в первом LOB-столбце используется атрибут DISABLE STORAGE IN ROW в режиме DISABLE, в результате чего все LOB попадают в табличное пространство для LOB-объектов, при этом другой столбец находится в режиме ENABLE STORAGE IN ROW, разрешающем запись LOB-данных в таблицу. Имеется также очень полезная фраза ENABLE ROW MOVEMENT, которая позволяет перескакивать строкам из секции в секцию при изменении значения ключа секционирования (для данной таблицы это изменение столбца CREATED).

В нашем случае имеем интервальное секционирование по типу Range (секционирование за период по дням). Интервал секционирования задается фразой INTERVAL (INTERVAL '1' DAY) . В результате при выполнении команды Insert при вводе новых строк, которые должны попасть в новую секцию, такая секция создается автоматически.

Однако при использовании Interval появляются особенности, если планируем разнести секции с LOB-объектами и секции таблицы по разным ТП.

Ранее (до использования интервального секционирования) разнос секций таблицы и секций с LOB-объектами успешно реализовывался. Для этого достаточно было при создании таблицы в конструкции STORE AS, описывающей LOB-объект, указать другое ТП. Однако при использовании фразы Interval создавать новые LOB-секции в другом ТП (отличном от таблицы) можно только для секций, описанных вручную при создании таблицы (в нашем случае это одна секция PARTITION PART_01012017).

Остальные автоматически созданные секции (как LOB-секции, так и секции таблицы) при использовании фразы Interval будут создаваться в одном табличном пространстве. Это ТП определяется при создании таблицы (в нашем примере это ТП AIFDATA, указанное при создании таблицы фразой TABLESPACE AIFDATA).

Замечание. Если ТП таблицы не задано, то новая сессия по Interval будет создаваться в ТП, определенном по умолчанию для таблицы на основании имени схемы, в которой создается таблица. Увидеть это default ТП можно запросом:

```
Select default_tablespace, username from dba_users ↵
         where username='AIF';
```

Чтобы однозначно указать, в какое ТП должна попадать вновь создаваемая секция, целесообразно при создании таблицы к фразе Interval добавить фразу STORE IN (имя ТП). В нашем примере написать INTERVAL (INTERVAL '1' DAY) STORE IN (IMAGE), тогда LOB-секции и секции таблицы будут попадать не в ТП AIDDATA, а ТП IMAGE.

При этом во фразе STORE IN (IMAGE) может быть не только одно ТП, а и перечень табличных пространств (через запятую). Например, STORE IN (IMAGE, IMAGE2, IMAGE3), в которых по очереди будут создаваться новые секции по Interval (в цикле, т.е. вначале в IMAGE, далее в IMAGE2, затем в IMAGE3 и снова IMAGE и т.д.).

Замечание. Если таблица уже создана, то изменить ТП, в котором будет создаваться новая секция при Interval, можно (в любой момент без остановки работы с таблицей) командой:

```
ALTER TABLE схема.имя таблицы SET STORE IN (имя ТП);
```

Для нашего примера это будет:

```
ALTER TABLE AIF.SVLOG SET STORE IN (IMAGE);
```

Замечание. В случае необходимости замены интервала в существующей секционированной таблице, например месячного, INTERVAL (INTERVAL '1' MONTH) на другой интервал (например, ежедневный), следует выполнить две команды:

```
ALTER TABLE AIF.SVLOG SET INTERVAL ();
ALTER TABLE AIF.SVLOG SET INTERVAL (INTERVAL '1' DAY);
```

где первая команда удаляет существующую фразу Interval из таблицы, а вторая вставляет новый интервал.

На основании вышесказанного можно сделать выводы об особенностях использования Interval для секционированных таблиц с LOB-столбцами:

- При создании новой секции по Interval секция таблицы и соответствующая ей LOB-секция попадают в одно и то же ТП.
- Табличное пространство, где создается новая секция по Interval, является ТП, заданное при создании таблицы (фразой TABLESPACE AIFDATA). Если ТП при создании таблицы не определено, то создается в default ТП таблицы.
- Для однозначного задания имени ТП, в котором планируется создание новых секций по Interval, используется фраза STORE IN (имя ТП).

Таким образом, при использовании интервального секционирования целесообразно при создании таблицы задать фразу STORE IN (имя ТП), например:

```
INTERVAL (INTERVAL '1' DAY) STORE IN (IMEGE)
```

В случае необходимости переноса секции таблицы в другое ТП, используется команда MOVE. Например, для переноса секции таблицы SYS_P156318 в пространство LOBT используется команда:

```
ALTER TABLE AIF.SVLOG MOVE PARTITION SYS_P156318 ↵
         TABLESPACE LOBT PARALLEL 32 NOLOGGING COMPRESS ↵
         UPDATE GLOBAL INDEXES;
```

где применяется режим сжатия COMPRESS переносимых данных, а также используется для ускорения работы распараллеливание операции (PARALLEL степени 32), кроме того, фраза UPDATE GLOBAL INDEXES обеспечивает при переносе секции исправность глобальных индексов.

Для переноса из той же секции SYS_P156318 соответствующей ей LOB-секции LOB-столбца REQUESTHEADER в другое ТП используется команда:

```
ALTER TABLE AIF.SVLOG MOVE PARTITION SYS_P156318 ↵
         LOB (REQUESTHEADER) STORE AS (TABLESPACE LOBT) ↵
         PARALLEL 32 NOLOGGING COMPRESS UPDATE GLOBAL INDEXES;
```

где в команде добавляется фраза LOB (REQUESTHEADER) с указанием имени переносимого LOB-столбца и фраза STORE AS (TABLESPACE LOBT) с указанием ТП, куда перенести LOB-секцию для данного LOB- столбца.

Замечание. Если в таблице несколько LOB-столбцов и все их хотим перенести в другое ТП, то в вышеуказанную команду ALTER следует добавить через запятую перечень LOB-столбцов:

```
ALTER TABLE AIF.SVLOG MOVE PARTITION SYS_P156318 ↵
         LOB (REQUESTHEADER, REQUESTPARAMS) STORE AS ↵
         (TABLESPACE LOBT) PARALLEL 32 NOLOGGING COMPRESS ↵
         UPDATE GLOBAL INDEXES;
```

Увидеть в каком табличном пространстве создается новая секция таблицы и ее LOB-секции позволяют указанные ниже запросы.

Запросы, определяющие размер секций секционированных таблиц

Определить табличное пространство для LOB-секций и их размер (в mb), а также общее число mb, занимаемое всеми LOB-секциями таблицы, позволяет запрос:

```
Select s.tablespace_name lob_tablespace_name, s.owner, ↵
    d.table_name, d.column_name lob_column_name, ↵
    p.high_value, l.partition_position pos, l.partition_name, ↵
    l.lob_partition_name, l.lob_indpart_name, ↵
    s.segment_name, round(s.bytes/1024/1024) lob_part_mb, ↵
    round(sum(s.bytes/1024/1024) over (partition by ↵
    s.segment_name, d.column_name)) lob_sum_mb, s.segment_type ↵
    From dba_segments s,dba_lobs d,dba_lob_partitions l, ↵
    dba_tab_partitions p Where l.table_owner=d.owner ↵
    and l.table_name=d.table_name ↵
    and l.column_name=d.column_name and s.partition_name ↵
    in (l.lob_partition_name,l.lob_indpart_name) ↵
    and s.segment_name in (d.segment_name, d.index_name) ↵
    and s.owner=d.owner and p.table_name=d.table_name ↵
    and p.table_owner=s.owner ↵
    and p.partition_name=l.partition_name ↵
    and d.owner='AIF' and d.table_name ='SVLOG' ↵
    and s.segment_type='LOB PARTITION' order by d.column_name, ↵
    s.segment_type, l.partition_position, l.partition_name;
```

Определить ТП секций таблицы, число строк и размер (в mb), занимаемый каждой секцией таблицы (без LOB), а также размер всей таблицы (без LOB) позволяет запрос:

```
Select p.tablespace_name, p.table_owner owner, p.table_name, ↵
    p.high_value, p.partition_position pos, ↵
    p.partition_name, s.segment_name, ↵
    p.num_rows,round(s.bytes/1024/1024) mb, ↵
    round(sum(s.bytes/1024/1024) over (partition by ↵
    table_name)) sum_mb ↵
    From dba_tab_partitions p,dba_segments s ↵
    Where s.owner=p.table_owner ↵
    and s.segment_name=p.table_name ↵
    and s.partition_name=p.partition_name ↵
    and p.table_owner='AIF' ↵
    and P.table_name ='SVLOG' order by pos;
```

Особенности очистки табличного пространства, занимаемого LOB-объектами

В отличие от обычных таблиц удаление данных из таблицы с LOB не освобождает табличное пространство, занимаемое LOB. Для освобождения табличного пространства используется команда SHRINK либо TRUNCATE таблицы или секции для секционированных таблиц.

Сжатие пространства для не секционированной таблицы осуществляется по команде

```
ALTER TABLE SERVICEMSGXML MODIFY LOB (XMLIN) ↵
         (SHRINK SPACE CASCADE);
```

Для секционированной таблицы команды с shrink и truncate имеют вид:

```
ALTER TABLE имя таблицы MODIFY PARTITION имя секции ↵
         LOB (имя LOB столбца) (SHRINK SPACE);
ALTER TABLE имя таблицы TRUNCATE PARTITION имя секции ↵
         UPDATE GLOBAL INDEXES;
```

Фраза UPDATE GLOBAL INDEXES в TRUNCATE обеспечивает исправность глобальных индексов после очистки секции по Truncate.

Выводы

- Для каждого LOB-столбца таблицы автоматически создают два LOB-сегмента.
- По умолчанию LOB-сегменты попадают в то же ТП, что и таблица. Если LOB-столбцы требуется разместить в другом ТП, то используется конструкция STORE AS или STORE AS SECUREFILE (кроме случая интервального секционирования).
- Использование интервального секционирования (т.е. фразы Interval) приводит к созданию новой секции, в которую попадают как секции таблицы, так и ее LOB-секции. Задать имя ТП для новых секций позволяет использование фразы STORE IN (имя ТП).
- Возможна модификация ряда атрибутов конструкции STORE AS или STORE AS SECUREFILE.
- Для сжатия табличного пространства LOB-объектов используются команды SHRINK или TRUNCATE. EOF

Ключевые слова: Oracle, СУБД, LOB.

Визитка

ИГОРЬ АНТОНОВ,
руководитель отдела разработки ПО,
страховая компания АО «ДальЖАСО», a@iantonov.me

Адаптация типовых конфигураций

Применение механизма расширений на практике

Адаптация типовых решений от «1С» под требования заказчика чаще всего сопровождается сложностями. Вместе с новым функционалом приходят проблемы дальнейшего сопровождения в виде трудностей установки обновлений от поставщика

Применение хорошо зарекомендовавших себя практик доработки решает проблему частично – как не старайся, а переопределить типовой код без снятия конфигурации с поддержки невозможно. Помочь справиться с надоевшей проблемой призвана новая технология – механизм расширений.

В прошлом номере журнала публиковалась статья «Правильная доработка типовых решений от 1С» [1]. В ней мы рассмотрели практики доработки типовых решений, позволяющих максимально сократить сложности с последующей установкой обновлений от поставщика. К сожалению, при глобальных доработках, затрагивающих основополагающие алгоритмы типового решения, предложенные методы не смогут полностью решить все проблемы.

Разработчику придется снимать конфигурацию с поддержки и вносить изменения напрямую, модифицируя типовой код. Единственный способ не потеряться в собственных изменениях – включить на полную катушку документирование и с помощью подручных технологий автоматизировать проверку возможности объединения серьезно переработанных участков. Шаг влево, шаг вправо – и прощай выполненные изменения.

Особые сложности возникают при росте участков с изменениями типового кода. В моей практике адаптации типовой конфигурации «Бухгалтерия 2.0» изменения затрагивали десятки участков кода. Каждый процесс обновления требовал особой внимательности, чтобы не затереть наработки. Наиболее трудно сращивать изменения после серьезного рефакторинга кода со стороны поставщика. При сопровождении «молодого» типового решения сращивание кода доставляет особые проблемы. Поставщик активно меняет код, и надо успевать подстраивать свои наработки.

Механизм расширений

Компания «1С» в курсе перечисленных выше трудности. Разработчики платформы потратили большое количество времени, чтобы привнести в систему новую технологию, способную решить наболевшие проблемы. Таким решением стал механизм расширений конфигурации.

Как и все новое, с момента первого релиза технология подвергалась переосмысливанию, доработкам и пересмотру функционала. За пару лет существования концепт технологии устоялся, и сегодня ее можно применять без особого страха и рисков. Основные возможности прекрасно работают, и прикладным разработчикам пора примять ее в проектах.

Перед тем как погрузиться в технические нюансы и рассмотрение практических примеров, давайте немного поговорим об общем механизме системы расширения конфигурации. Каким образом новая технология решает стек закоренелых проблемам?

Первая и важная особенность расширений – работа поверх стоковой конфигурации. Разработчику не требуется снимать конфигурацию поддержки. В контексте механизма расширений необходимые доработки выносятся прямо в расширение. Технически расширение представляет собой мини-конфигурацию, которая при подключении объединяется с основной. Объединение выполняется автоматически сразу после подключения. Непосредственное подключение происходит прямо из режима «предприятие». Достаточно выбрать файл-расширение (*.cfe) и подключить его через менеджер расширений. Потребуется перезапуск «предприятие», и после него расширение будет готово к работе. Процесс подключения и активации расширения выполняется максимально прозрачно.

Механизм расширений практически полностью решает проблему сопровождения доработанных конфигураций. В идеальных случаях конфигурацию не требуется снимать с поддержки, и последующие обновления выполняются стандартным образом. После установки обновления от поставщика добавленные расширения автоматически подключатся к обновленной информационной базе.

Как разработчика вас никто не ограничивает в количестве расширений. Хотите – дробите функционал на отдельные расширения или держите доработки в одном. Главное, придерживайтесь принципа «все расширения должны быть к месту». Чрезмерное количество подключенных расширений может сыграть злую шутку и нанести урон производительности.

Хорошо, а что если поставщик конфигурации внесет глобальные изменения, подобно упомянутым выше, – проведет серьезный рефакторинг кода? В этом случае фатального ничего не случится. Расширение просто перестанет работать, и вы сможете заняться адаптацией к новым требованиям.

Механизм расширений покрывает сценарии адаптации типовых решений, но важно запомнить, что расширение не является полноценной конфигурацией. Разработчик может создавать дополнительные объекты метаданных (в расширении), но перечень поддерживаемых для создания объектов ограничен. Не буду перечислять список поддерживаемых объектов (см. официальную документацию), скажу лишь, что в расширениях отсутствует возможность создания объекта для хранения информации. Добавить новый справочник или реквизит документа не получится.

Это одновременно ограничение и защита данных. Если разработчик имеет возможность создавать в расширении объекты для хранения информации, а потом расширение случайно отключается от информационной базы, то данные улетят в трубу, что недопустимо.

Хорошо, а как тогда быть, если доработка подразумевает ввод в конфигурацию дополнительных объектов метаданных (справочников, документов)? Снимать с поддержки? Нет, в этом случае активируется возможность внесения изменений (поддержка остается на месте), создаются новые объекты, а вот взаимодействие или переопределение типовых вещей выполняется в расширении. Схема работы расширений приведена на рис. 1.

Создаем первое расширение

Для создания нового расширения требуется воспользоваться менеджером расширений – «Конфигурация → Расширения конфигурации». Открываем менеджер (см. рис. 2). В окне менеджера вы можете управлять доступными расширениями (удалять, тестировать на возможность применения к конфигурации, выгружать в файлы и т.д.) и создавать новые. Для создания расширения нажимаем кнопку «Добавить». В появившемся окне (см. рис. 3) требуется заполнить:

- **Имя расширения.** К именованию применяются стандартные правила именования новых метаданных объектов.
- **Синоним.**
- **Префикс.** Ко всем процедурам/функциям добавляется указанный префикс. Префикс желательно выбирать исходя из контекста расширения.

Двойной клик мышкой по расширению откроет дерево конфигурации расширения. В нем вы можете создавать новые объекты или помещать объекты основной конфигурации для последующего расширения. Последнюю операцию удобно выполнять через контекстное меню объекта метаданных в дереве рабочей конфигурации (см. рис. 4).

Далее выполняется стандартный цикл разработки – пишете бизнес-логику расширения и запускаете режим «предприятие» для тестирования. После завершения разработки выполняется выгрузка (через менеджер расширений) расширения в файл («Конфигурация → Сохранить конфигурацию в файл») и последующее подключение к рабочей базе. Подключение готового расширения реализуется в режиме «предприятие» («Все функции → Стандартные → Управление расширениями конфигурации»).

Расширяем модули

Переходим непосредственно к практической части и начнем рассматривать расширения в контексте модернизации модулей. Начиная с релиза 8.3.9 механизм расширений серьезно обновился и привнес новый функционал в работе с модулями. Разработчики получили возможность изменять любые типы модулей, определенные в конфигурации.

Функционал, направленный на расширение модулей, покрывает задачи, которые раньше невозможно было решить без изменения. В расширении разработчик может создавать новые общие модули и, самое главное, переопределять существующие путем перехвата вызова функций/процедур (далее будем использовать слово «метод»). Переопределяемые методы могут вызывать: перед типовыми, после типовых или вместо типовых.

Рассмотрим практический кейс расширения модулей. Представим, в модуле «СуперМодуль» есть функция «РассчитатьСебестоимость(Товар)». Не будем вдаваться в детали,

Рисунок 1. Схема работы расширений

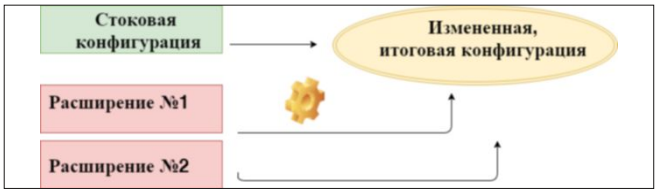


Рисунок 2. Менеджер расширений в режиме «Конфигуратор»

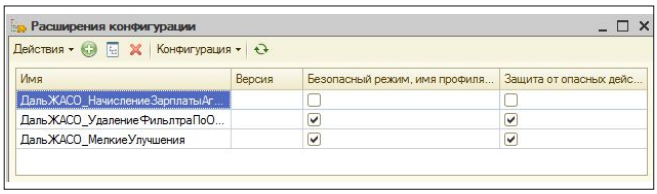


Рисунок 3. Создание нового расширения

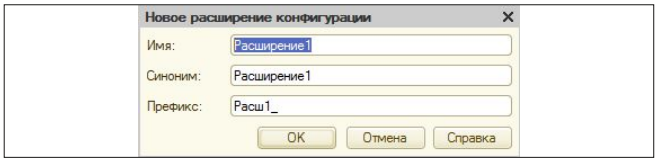
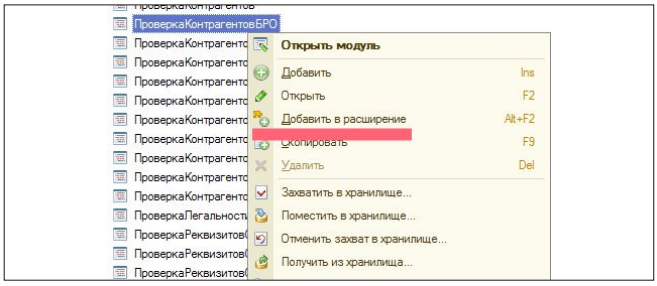


Рисунок 4. Добавляем в расширение общий модуль



мы знаем лишь, что она возвращает себестоимость переданного в качестве параметра товара. Поступает задача переделать функцию так, чтобы к себестоимости каждого товара добавлялась магическое число 100.

Проще всего задача решается добавлением строчки кода в конец функции «РассчитатьСебестоимость()», но тогда в будущем на нас упадут все проблемы сопровождения. В контексте технологии расширения модулей задача сведется к переопределению функции «РассчитатьСебестоимость()» в новом расширении.

Порядок решения будет таким. Создаем новое расширение и через контекстное меню дерева конфигурации помещаем в него «СуперМодуль». Затем в дереве конфигурации расширения переопределяем интересующий нас метод:

```
&Вместо(«РассчитатьСебестоимость»)
Функция ПрефиксРасширения_РассчитатьСебестоимость(Товар)
        Результат = ПродолжитьВызов(Товар);
        Возврат Результат + 100;
КонецФункции
```

Переопределяя в расширении типовой метод, в первую очередь мы должны определиться с вариантом его вызова. Порядок вызова описывается при помощи добавленных в язык препроцессора новых аннотаций:

- &Вместо («ИмяПроцедуры/Функции»)
- &После («ИмяПроцеудры/Функции»)
- &Перед («ИмяПроцедуры/Функции»)

Рисунок 5. Конфигурация нового расширения

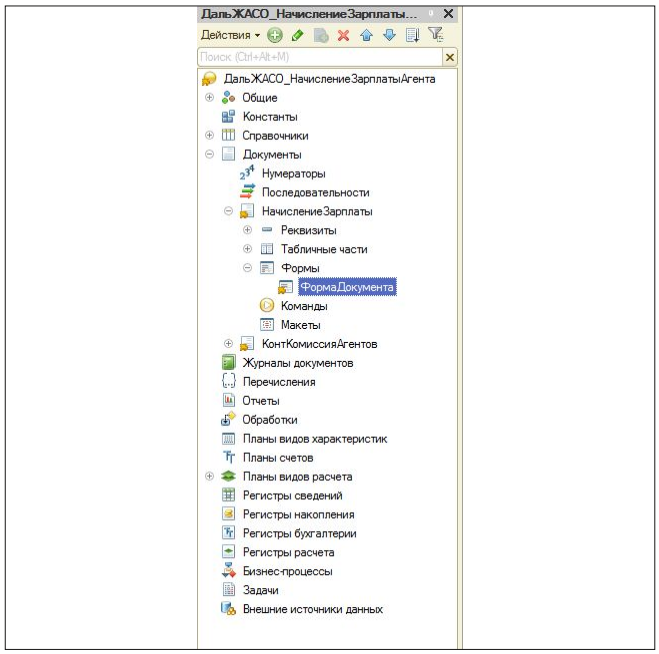
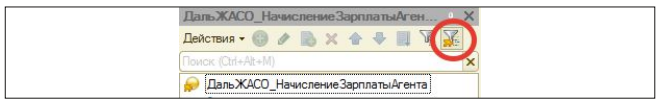


Рисунок 6. Управление видимостью измененных объектов



В примере выбор был сделан в пользу аннотации «&Вместо». Таким образом, после применения расширения типовой метод «РассчитатьСебестоимость» становится «мертвым». Вместо него всегда будет вызываться наша реализация. На практике это представляет опасную ситуацию, поскольку мы ставим крест на возможных обновлениях реализации оригинального метода. Применять полное перекрытие стоит только в крайних случаях.

Поскольку полное перекрытие несет ряд ненужных сложностей, в теле метода я объявляю переменную и присваиваю в нее результат выполнения функции «ПродолжитьВызов(Товар)». Эта встроенная функция передаст управление оригинальному методу, и результат выполнения запишется в переменную «Результат». Получив результат выполнения оригинальной функции, выполняем прибавление числа 100.

Модифицировать результат требуется далеко не всегда. Например, при необходимости проведения документа по новому регистру подойдет переопределение метода с аннотацией «После». В этом случае сначала отработает оригинальный код, а потом вызовется наша процедура. Механизм расширения модулей прекрасно подходит как для адаптации решений, так и для создания экстренных заплаток (кейс рассмотрим ниже).

Модернизация форм

Теперь взглянем, как выглядит процедура модернизации типовых форм посредством расширений. Для демонстрации примера я возьму документ «НачислениеЗарплаты» типовой конфигурации «Бухгалтерия некредитной финансовой организации КОРП» (вы можете повторить на любом другом) и доработаю основную форму.

Согласно техническому заданию необходимо вывести на форму значение нового реквизита «КомиссияАгента» и добавить команду, автоматизирующую заполнение в табличной части колонки «Дата выплаты».

Создаем новое расширение и переносим в него основную форму документа «НачисленияЗарплаты». Помимо интересующей нас формы, в дерево конфигурации расширения попадут связанные сущности (документы, справочники и т.д.). На них не стоит обращать внимание (таков принцип работы расширений). Нас интересует форма документа, которую надо немного доработать (см. рис. 5).

Я вынес на форму отображение необходимого реквизита («КомиссияАгента») и добавил команду «Проставить дату оплаты». Чтобы не запутаться в создаваемых элементах, каждому сразу присваивается префикс в виде наименования организации. Например, «ИмяОрганизации_КомиссияАгента», «ИмяОрганизации_ВыполнитьПростановкуДатыОплаты».

Обратите внимание на маленькую особенность. Если в оригинальной форме не было ссылок на некоторые реквизиты объекта, то они автоматически не перенесутся в расширение. Их придется добавлять вручную таким же образом, как и форму. Только после этой операции реквизит будет доступен для закрепления в форме расширения.

При модификации сложных форм (содержащих большое количество реквизитов разного типа) в дереве конфигурации расширения начнется легкий бардак в виде появившегося огромного числа ссылок на объекты конфигурации. Чтобы не затеряться в нем, воспользуйтесь кнопкой

«Измененные и добавленные в расширение» на панели инструментов дерева окна конфигурации (см. рис. 6).

Команды формы создаются стандартным способом. В примере я добавляю одну новую команду «Проставить дату оплаты» и в обработчики события пишу код:

```
&НаСервере
Процедура НЗА_ДальЖАСО_ПроставитьДатуВыплатыПередНаСервере()

        Для Каждого ЭлементНачисления Из
                ОБъект.Начисления Цикл
            ЭлементНачисления.ПланируемаяДатаВыплаты = ↵
                            ДальЖАСО_ДатаВыплаты;
        КонецЦикла;

        Модифицированность = Истина;

КонецПроцедуры
```

В модуле формы расширения вы видите только свой код. Программный код из оригинальной формы не переносится, но ваш код выполняется в контексте модуля оригинальной формы. Все методы формы, описанные в конфигурации поставщика, доступны из расширения (см. рис. 7, 8).

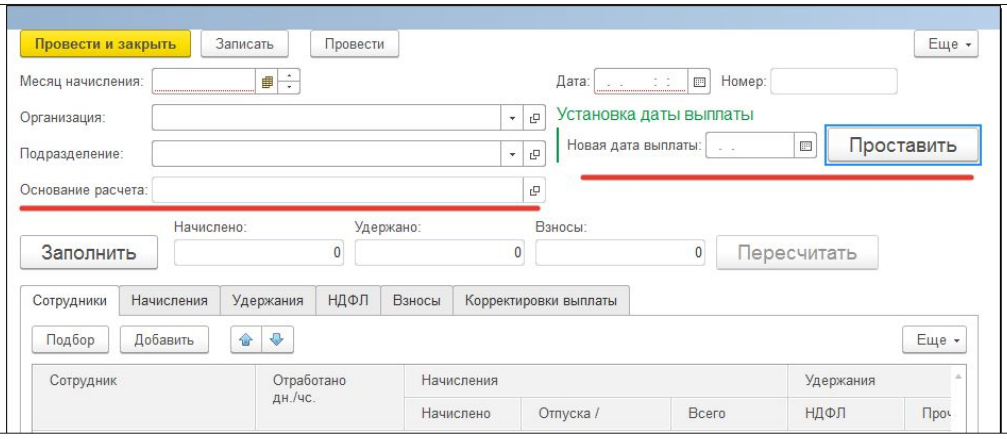
Если при изменении формы требуется переопределить логику обработчиков существующих элементов, то следует применять уже разобранный процесс перехвата методов. Принцип тот же, но тип вызова указывается не аннотацией, а в конструкторе обработчика.

На практике этот процесс выглядит следующим образом. Переходите в инспекторе свойств команды (например, «Заполнить») и нажимаете на кнопку определения обработчика. Перед вами должно отобразиться диалоговое окно (см. рис. 9) с вопросом выбора контекста исполнения обработчика (клиент, сервер) и типа вызова.

Рисунок 7. Модуль формы расширения

```
//*************************************************************
//* ВСПОМОГАТЕЛЬНЫЕ ПРОЦЕДУРЫ (СЕРВЕР)
//*************************************************************

&НаСервере
Процедура НЗА_ДальЖАСО_ПроставитьДатуВыплатыПередНаСервере()

    Для Каждого ЭлементНачисления Из ОБъект.Начисления Цикл
        ЭлементНачисления.ПланируемаяДатаВыплаты = ДальЖАСО_ДатаВыплаты;
    КонецЦикла;

    Модифицированность = Истина;

КонецПроцедуры

//*************************************************************
//* ОБРАБОТЧИКИ СОБЫТИЙ (КЛИЕНТ)
//*************************************************************

&НаКлиенте
Процедура НЗА_ДальЖАСО_ПроставитьДАтуВыплатыПередЗавершение(РезультатВопроса, ДополнительныеПараметры) Экспорт

    Если РезультатВопроса = КодВозвратаДиалога.Нет Тогда
        Возврат;
    КонецЕсли;

    НЗА_ДальЖАСО_ПроставитьДатуВыплатыПередНаСервере();

КонецПроцедуры

&НаКлиенте
Процедура НЗА_ДальЖАСО_ПроставитьДатуВыплатыПеред(Команда)

    ОписаниеОповещения = Новый ОписаниеОповещения("НЗА_ДальЖАСО_ПроставитьДАтуВыплатыПередЗавершение", ЭтаФорма);
    ПоказатьВопрос(ОписаниеОповещения, "Вы действительно хотите проставить новую даты выплаты?", РежимДиалогаВопрос.ДаНет,,, "Внимание!")

КонецПроцедуры
```

Рисунок 8. Результат расширения формы



Тип вызова позволяет задать порядок выполнения переопределяемого обработчика. Варианты аналогичны доступным аннотациям: перед, после и вместо.

После создания обработчика в инспекторе свойств (см. рис. 10) появится дополнительная иконка, отражающая порядок исполнения переопределенного обработчика события.

Расширение как заплатка

Само слово «расширение» заставляет воспринимать технологию как средство для модернизации типового функционала. Не нужно загонять себя в рамки и откидывать альтернативные идеи применения расширений. Один из интересных кейсов использования технологии расширений – создание заплаток.

Вычистить полностью код от ошибок сложно, и, как обычно, они возникают в самых неожиданных местах. Разработчики привыкли ругать программистов «1С» за многочисленные баги в типовых конфигурациях, но не все отдают себе отчет в масштабности проекта и размере кодовой базы.

Ошибки всегда имеют место быть, и далеко не всегда заплатки от поставщика приходят оперативно. Что делать в таких случаях? Самый верный вариант – откатиться на предыдущую версию. Только сделать это удается не всегда. Ошибка может долго не проявляться, а возникнуть при выполнении регламентных операций.

Например, представим, что после очередного обновления возникла ошибка в документе начисления заработной платы. Если обновление с ошибкой появилось после ежемесячного начисления заработной платы, то о проблеме узнают только к следующему расчету зарплаты. За это время информационная база пополнится данными, и просто так, без их потери, откатиться к резервной копии не получится.

В таких ситуациях механизм расширений может сослужить хорошую службу. Ничего не мешает вам создать новое расширение и внести в него исправление ошибок типового механизма, получив тем самым своеобразную заплатку. Вы останетесь со всеми свежими изменениями и примените элегантный патч на код, содержащий ошибки. Как только разработчики выпустят официальное исправление, потребуется просто отключить временное расширение.

Рисунок 9. Определяем порядок вызова обработчика

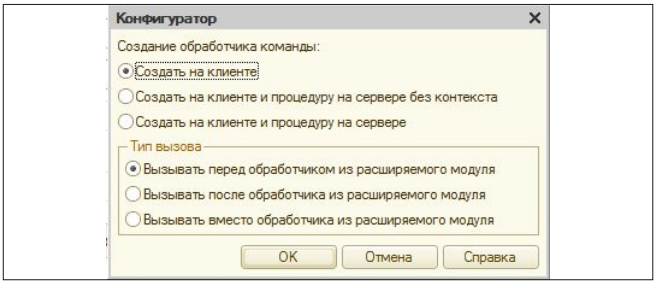
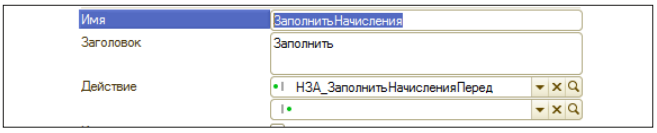


Рисунок 10. После создания обработчика формы



Озвученная методика будет актуальна не только для разработчиков, занимающихся сопровождением типовых решений, но для тех, кто занят в разработке собственных решений. Если вы выпустили обновление с проблемной функцией, которой пользуются только несколько ваших клиентов, то, возможно, проще будет поставить лично им заплатки, чем выкатывать новый релиз и принуждать к обновлению остальных.

Расширение как плагин

Компания «1С» пока не спешит агитировать разработчиков за создание полноценных тиражных решений на основе механизма расширений, мотивируя молодостью технологии и не совсем подходящей архитектурой. Несмотря на это, уже сейчас вполне реально применять технологию расширений для создания небольших универсальных плагинов.

Безусловно, проблемы тиражирования имеют место быть (как правило, речь идет об организационных моментах), но во многих случаях они решаемы. Я долго думал над примером для демонстрации кейса универсального расширения. Мне в голову не пришло ничего лучше, чем разработка плагина, проверяющего уникальность контрагентов при записи. Если в базе существует контрагент с таким же ИНН, то плагин запретит запись.

Создаем новое расширение стандартным способом. Затем добавляем в него форму элемента справочника «Контрагенты». Проверку на дубли будем выполнять в интерактивном режиме, т.е. во время взаимодействия пользователя с формой справочника. Для реализации защиты от дублей нам требуется расширить обработчик события формы «ПередЗаписью».

Создаем для расширяемой формы обработчик события «ПередЗаписью» и в качестве типа вызова выбираем «Вызывать перед обработчиком из расширяемого модуля». В теле обработчика пишем код из листинга 1.

Листинг 1. Запрещаем запись дублированных контрагентов

```
ЭтоФизическоеЛицо = Объект.ЮридическоеФизическоеЛицо = ↲
        ПредопределенноеЗначение("Перечисление. ↲
        ЮридическоеФизическоеЛицо.ФизическоеЛицо");
        Если (Объект.ИННВведенКорректно И ↲
                Объект.КППВведенКорректно) ИЛИ
            (Объект.ИННВведенКорректно И ↲
                ЭтоФизическоеЛицо) Тогда

КоличествоДублейКонтрагентов = ВыполнитьПоискДублейСервер( ↲
        Объект.ИНН, Объект.КПП, Объект.Ссылка);

Если КоличествоДублейКонтрагентов > 0 Тогда
Отказ = Истина;
ТекстСообщенияПользователю = СтрШаблон("В справочнике →
        ""Контрагенты"" существуют элементы с таким ↲
        ИНН/КПП - %1/%2.
        |Запись элемента не может быть выполнена.", ↲
        Объект.ИНН, Объект.КПП);

ПоказатьПредупреждение(, ТекстСообщенияПользователю);
  КонецЕсли;
КонецЕсли;
```

Перед записью элемента мы выполняем проверку на корректность введенных ИНН и КПП. Причем мы ставим два условия – в первом проверяем пару ИНН + КПП,

а во втором – ИНН и признак физического лица. Если условие выполняется, то приступаем к поиску дублей.

Сам алгоритм поиска дублей мы не реализовываем – пользуемся готовой функцией «ВыполнитьПоискДублейСервер». Но где описана данная функция? Она описана в модуле оригинальной формы, к которому мы можем обратиться исходя из правила контекста.

Результатом выполнения функции «ВыполнитьПоискДублейСевере» будет число – количество обнаруженных дублей. Если оно больше нуля, то в базе есть уже контрагент с введенными ИНН/КПП и запись надо пресекать (Отказ = Истина), не забыв при этом известить пользователя.

Код расширения получился достаточно простым, но наглядно продемонстрировал легкость и практическую пользу применения расширений в виде плагинов (см. рис. 11). Плагин получился универсальным, и его без особо труда легко подключить к похожей типовой конфигурации.

Слабые стороны

Механизм расширений привносит в процедуру адаптации типовых решений новые богатые возможности и изменяет устоявшиеся правила игры. Доработать конфигурации становится проще, и новый функционал не препятствует последующей поддержке типового решения.

Как у любой технологии, у механизма расширений есть минусы. Первый и самый весомый – технические ошибки. Технология новая, затрагивает множество участков платформы, и ошибки имеют место быть. Их немного, они правятся от релиза к релизу, но они есть. По опыту использования механизма расширений еще с версии 8.3.6 могу сказать, что в 8.3.9 проблем стало значительно меньше. Именно с этого релиза мы начали применять механизм расширений в рабочем окружении.

Второй важный минус – дополнительная сложность доработки конфигураций с большим количеством подключенных расширений. Пока нет никаких встроенных средств, позволяющих понять, какое из расширений переопределяет метод оригинальной конфигурации. Из-за неготовности внутреннего инструментария могут возникать трудности при изучении механизма работы уже применяемых в базе расширений.

Третий весомый минус кроется в режиме совместимости. Я говорил, что самые интересные нововведения механизма расширений появились в последних версиях платформы. Так вот, чтобы воспользоваться ими, необходимо соблюсти условие: расширяемая конфигурация не должна работать в режиме совместимости с более ранними версиями.

В противном случае придется выбирать: довольствоваться функционалом механизма расширений, актуальным для версии совместимости, или самовольно его снимать. Типовые конфигурации в этом плане медлительны и многие только недавно перешли на режим совместимости с 8.3.8.

Вместо заключения

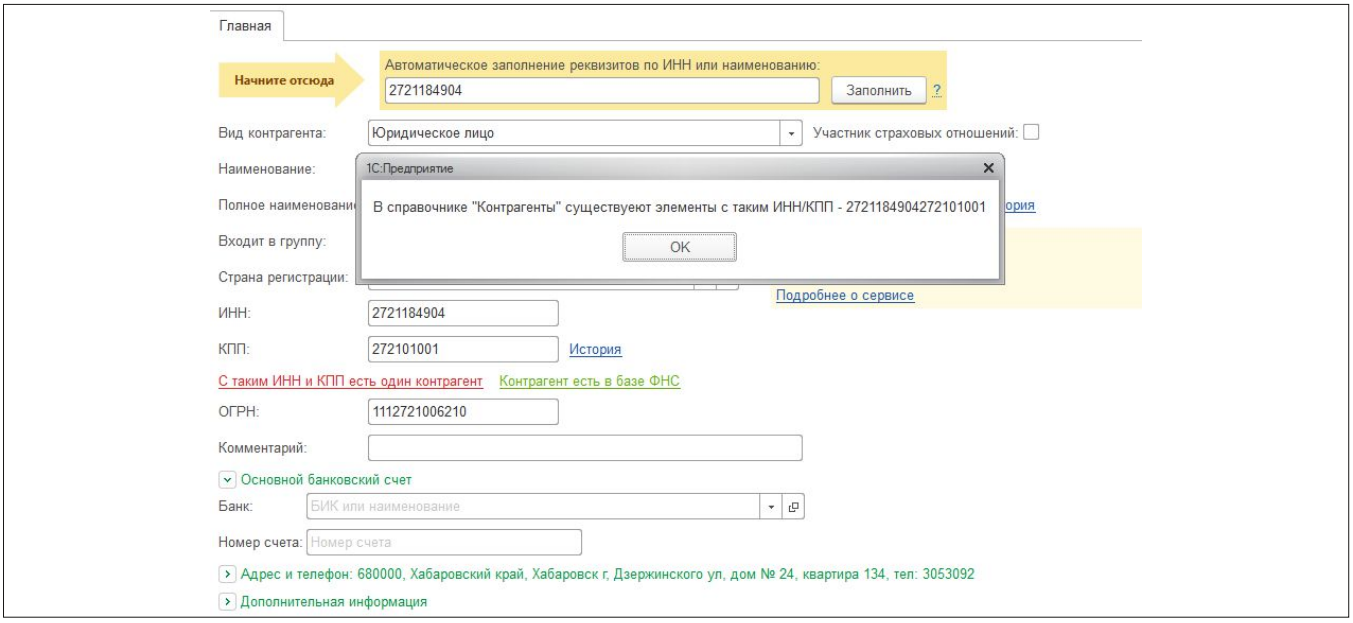
Можно долго рассуждать о сильных и слабых сторонах механизма расширений, но, так или иначе, уже сейчас эта технология позволяет решить ряд исторических проблем. Не все разработчики охотно ее встречают: есть замечания к недоработкам концепции и общей сырости технологии. Все это временные трудности.

Механизм расширений стоит применять уже сегодня, чтобы завтра быть готовым к новым правилам адаптации типовых решений. «1С» взяла строгий курс на развитие этой технологии. Следовательно, разработчики прикладных решений в обозримом будущем получат еще больше возможностей бесшовной доработки функционала типовых решений. EOF

[1] Антонов И. Правильная доработка типовых решений от 1С. // «Системный администратор», № 3, 2017 г. – С. 63-69 (http://samag.ru/archive/article/3393).

Ключевые слова: 1С, разработка, расширения.

Рисунок 11. Тест расширения запрета записи контрагентов с дублями



Визитка

ИГОРЬ ОРЕЩЕНКОВ,
инженер-программист, iharsw@tut.by

О совместном доступе к файлам в PHP

Работа с общими ресурсами в конкурентной среде является одной из самых сложных и интересных задач параллельного программирования. В статье рассматриваются некоторые аспекты совместного доступа к файлам на языке программирования PHP

Язык программирования PHP на сегодняшний день занимает лидирующие позиции в области веб-разработок. На нем написаны популярнейшие универсальные CMS (WordPress, Joomla!, Drupal), с помощью которых в сжатые сроки создаются как персональные блоги, так и интернет-магазины. Язык PHP используется в разнообразных интернет-проектах, разработку которых упрощают богатые функциональностью фреймворки (Symphony, Laravel, Yii).

Однако сугубо практическая направленность этого языка программирования, лежащая в его основе и сопутствующая всему процессу развития, явилась поводом к формированию субъективных мнений в сообществе программистов, которые со временем приобрели популярность. Низкий порог вхождения – простота установки интерпретатора и создания простейшей работающей программы – привели к возникновению иллюзии о том, что язык программирования автоматически решает проблему сложности мира, в котором он применяется. В некоторой степени этому способствовали книги, на страницах которых процесс веб-разработки освещался в популярной форме [1].

Из-за этого начинающие программисты иногда забывают о многогранности контекста, в котором предстоит выполняться их разработкам, что в конечном счете приводит к ошибкам, выливающимся в нестабильную работу программного продукта. А опытные коллеги, которые смотрят на подобные фиаско со стороны, преувеличивают вину языка программирования, на котором велась разработка.

Что говорит теория

Одним из наиболее сложных и трудновоспроизводимых факторов, который оказывает влияние на работу веб-приложения, является возможность одновременного выполнения сценариев (одного или разных) по запросам, поступающим от пользователей программного продукта, или другим событиям среды выполнения.

Сценарии для совершения своей работы обращаются к ресурсам: памяти, базам данных, файлам. И если области памяти у каждого выполняющегося PHP-сценария свои и не пересекаются (это обеспечивается средой выполнения), то исключить одновременное использование разными сценариями одних и тех же файлов или баз данных нельзя.

Одновременные обращения к одному и тому же ресурсу приводят к конкуренции между сценариями за доступ к этому ресурсу. В случае баз данных она регулируется сервером баз данных. Однако игнорирование конкуренции при работе с файлами приводит к конфликтам.

Рисунок 1. Разделяемая блокировка запрещает доступ к ресурсу только для процессов, которые запрашивают исключительную блокировку

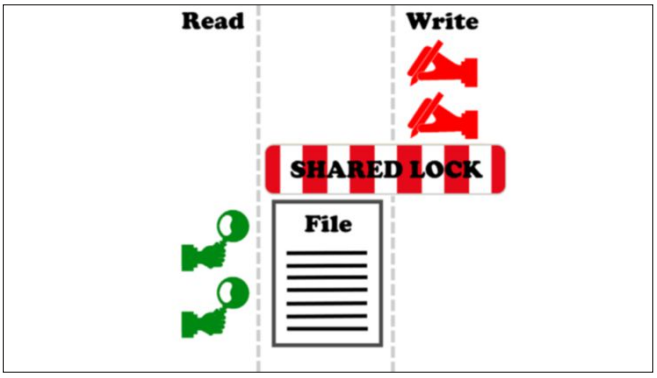
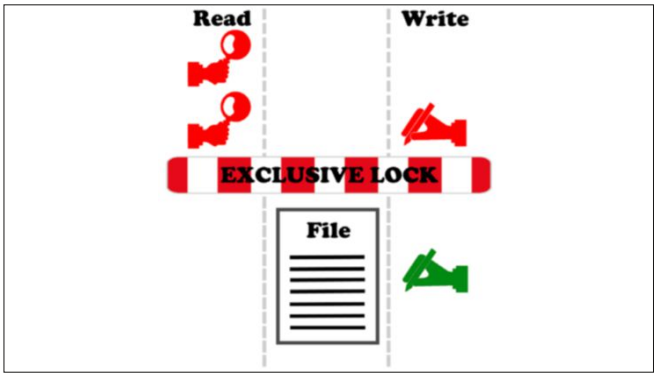
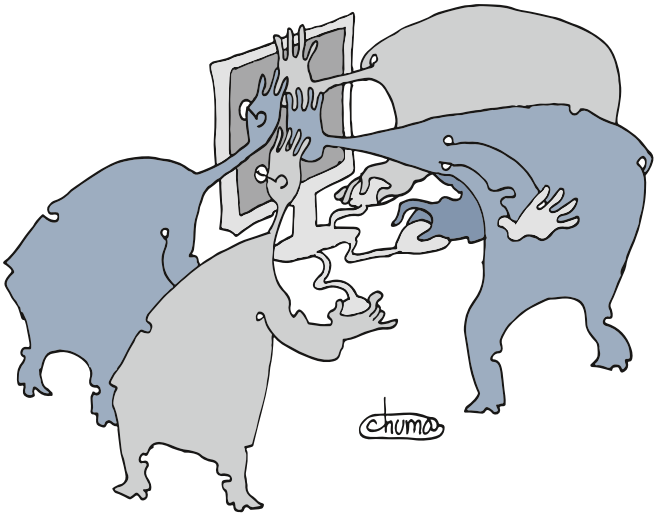


Рисунок 2. Исключительная блокировка запрещает доступ к ресурсу со стороны любых процессов, кроме установившего эту блокировку





Использование операций с файлами в PHP-приложениях не утрачивает своей актуальности

В теории параллельного программирования выделяются два способа работы с общим ресурсом:

- получение от ресурса информации, при которой его состояние остается неизменным (например, чтение содержимого файла);
- изменение состояния ресурса (например, запись информации в файл).

Очевидно, что в первом способе работы может одновременно участвовать несколько процессов, что никак не испортит результат этой работы. Каждый процесс получит одну и ту же информацию. Важно лишь, чтобы в это время не модифицировалось состояние используемого ресурса.

Чтобы выполнить это условие, каждый процесс устанавливает на ресурс «разделяемую» (SHARED) блокировку. Она не мешает чтению состояния ресурса, но запрещает его изменение (см. рис. 1).

Другое дело, когда процесс модифицирует общий ресурс. Несмотря на то что эта операция может записываться одной командой языка программирования, в реальных системах ее выполнение будет состоять из нескольких отдельных шагов. Любая попытка доступа к общему ресурсу в процессе его изменения приведет к получению ошибочных результатов. В случае чтения состояния ресурса будет получена информация о частично измененном состоянии. А в случае записи состояние ресурса может быть просто испорчено.

Таким образом, модифицировать ресурс одновременно может только один процесс, причем в ходе модификации не допускается даже чтение состояния этого ресурса. Для выполнения этого условия процесс, приступающий к изменению состояния, устанавливает на ресурс «исключительную» (EXCLUSIVE) блокировку. Она запрещает доступ к общему ресурсу со стороны всех остальных процессов (см. рис. 2).

Сказанное выше о блокировках является теоретическими сведениями, которые по-разному воплощаются в реальные программные продукты – операционные системы и языки программирования.

Какова реализация на практике

При работе с файлами в языке программирования PHP использование блокировок носит «договорной» характер. Это означает, что каждый участок кода, в котором может возникнуть конкуренция за одновременный доступ к файлу, должен быть оформлен специальным образом. Ни в коем случае нельзя полагаться на то, что разделение доступа к файлу будет решено на уровне операционной системы или где-нибудь еще.

Например, мало установить исключительную блокировку перед записью в файл новой информации. Это не воспрепятствует чтению из файла в другом сценарии, если в нем перед чтением не будет затребована разделяемая блокировка. Другими словами:

Участок кода № 1

```
...
if (flock ($fp, LOCK_EX)) {
        fwrite ($fp, ...);
}
...
```

Ошибочный участок кода № 2

```
...
$data = fread ($fp, ...);
...
```

Исправленный участок кода № 2

```
...
if (flock ($fp, LOCK_SH)) {
        $data = fread ($fp, ...);
}
...
```

Что происходит, если в момент выполнения функции flock (...) файл, в отношении которого запрашивается блокировка, «занят» другим процессом и блокировка не может быть установлена? Стандартное поведение PHP – приостановить работу сценария до тех пор, пока процесс, использующий

файл, не «освободит» его, то есть не снимет свою блокировку. А если не освободит? Тогда сценарий «зависнет».

Чтобы сценарий не останавливался в ожидании блокировки, нужно тип блокировки (LOCK_SH или LOCK_EX) комбинировать с флагом LOCK_NB, например: flock ($fp, LOCK_SH | LOCK_NB). Если в момент выполнения этой функции файл $fp будет «занят», то она сразу вернет FALSE, что позволит выполнить какие-нибудь другие операции перед очередной попыткой.

В большинстве практических применений стандартное поведение вполне приемлемо. С учетом изложенного можно рекомендовать следующий способ работы с файлами:

- Открыть файл для чтения или записи.
- Установить на файл разделяемую или исключительную блокировку.
- Выполнить запланированные файловые операции чтения/записи.
- Закрыть файл.

Тогда, чтобы правильно прочитать содержимое файла, на PHP потребуется написать такой текст (все листинги можно скачать на сайте журнала samag.ru).

Листинг 1. Участок кода, иллюстрирующий чтение из файла в конкурентной среде

```
<?php
/*==================================================*/
/* Участок кода для конкурентного чтения из файла. */
/* $fn - имя файла (file name),                     */
/* $fp - манипулятор файла (file pointer).          */
/*==================================================*/
$fn = 'file.txt';
$fp = @fopen ($fn, 'r');
if ($fp !== FALSE):
    if (flock ($fp, LOCK_SH)):
         $text = fread ($fp, 8192);
         flock ($fp, LOCK_UN);
    else:
         echo "ERROR: can't shared lock file $fn\n";
    endif;
    fclose ($fp);
else:
    echo "ERROR: can't open file $fn\n";
endif;
?>
```

Текст блока для записи информации в файл будет несколько сложнее из-за того, что операция открытия файла для записи немедленно очищает его содержимое, что недопустимо до успешного получения блокировки. Поэтому в зависимости от факта существования файла он открывается либо для записи «w» (и при этом создается новый пустой файл), либо для чтения/записи «r+» (при этом его содержимое сохраняется). В последнем случае содержимое файла перед записью очищается отдельной командой ftruncate () только после того, как будет успешно получена блокировка.

Листинг 2. Участок кода, иллюстрирующий запись в файл в конкурентной среде

```
<?php
/*==================================================*/
/* Участок кода для конкурентной записи в файл. */
/* $fn - имя файла (file name),                     */
/* $fp - манипулятор файла (file pointer).          */
/*==================================================*/
$fn = 'file.txt';
$fp = @fopen ($fn, file_exists ($fn)? 'r+': 'w');
if ($fp !== FALSE):
    if (flock ($fp, LOCK_EX)):
         ftruncate ($fp, 0);
         fwrite ($fp, 'Text, text, text, text...');
         flock ($fp, LOCK_UN);
    else:
         echo "ERROR: can't exclusive lock file $fn\n";
    endif;
    fclose ($fp);
else:
    echo "ERROR: can't open file $fn\n";
endif;
?>
```

Завершая обсуждение типовых методов работы с файлами в PHP, нужно отметить, что вся логическая транзакция решаемой задачи должна выполняться в пределах одной непрерывной блокировки.

Например, транзакция учета очередного посетителя сайта состоит из следующих шагов:

- Прочитать из файла текущее значение счетчика посетителей.
- Увеличить значение счетчика на 1.
- Записать новое значение счетчика в файл.

Рисунок 3. Схема веб-приложения с прямым взаимодействием диспетчера и контроллеров

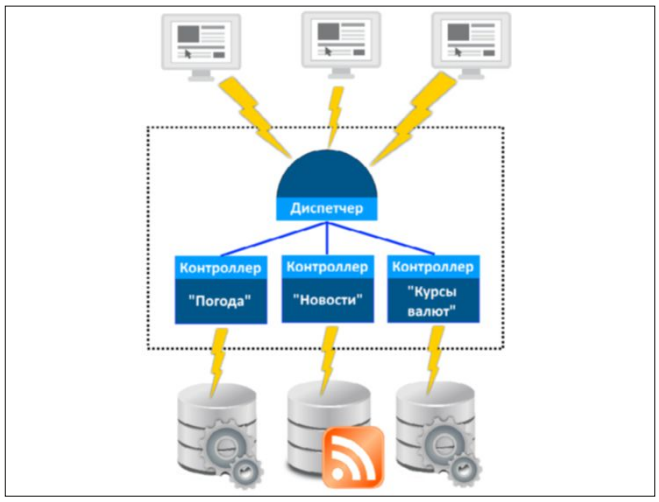
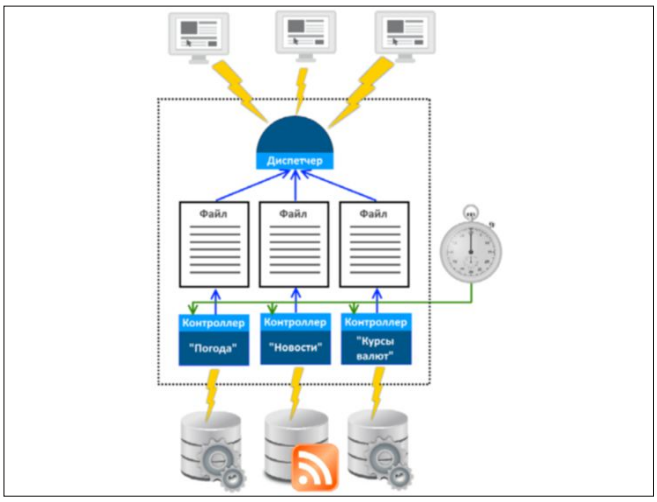


Рисунок 4. Схема веб-приложения с опосредованным взаимодействием диспетчера и контроллеров через файлы



Казалось бы, можно на первом шаге открыть файл для чтения с разделяемой блокировкой, прочитать его содержимое, закрыть файл, открыть его заново для записи с исключительной блокировкой, записать увеличенное на единицу значение счетчика и закрыть файл окончательно. Такой подход будет ошибочным, потому что между сеансами чтения и записи файла существует интервал времени, в котором файл не заблокирован и может быть модифицирован другим процессом.

Надо сказать, что начиная с версии PHP 5.3.2 закрытие файла само по себе уже не приводит к снятию блокировки, как это было раньше. В современных версиях блокировку нужно снимать явным вызовом функции flock ($fp, LOCK_UN).

Правильное решение задачи на языке PHP записано ниже.

Листинг 3. Участок кода, иллюстрирующий изменение файлового счетчика в конкурентной среде

```
<?php
/*==================================================*/
/* Участок кода для изменения значения файлового */
/* счётчика в конкурентной среде.                 */
/* $fn - имя файла (file name),                   */
/* $fp - манипулятор файла (file pointer).        */
/*==================================================*/
$fn = 'counter.txt';
$fp = @fopen ($fn, file_exists ($fn)? 'r+': 'w');
if ($fp !== FALSE):
    if (flock ($fp, LOCK_EX)):
          $counter = fread ($fp, 8192);
          $counter += 1;
          ftruncate ($fp, 0);
          rewind ($fp);
          fwrite ($fp, $counter);
          flock ($fp, LOCK_UN);
   else:
          echo "ERROR: can't exclusive lock file $fn\n";
   endif;
   fclose ($fp);
else:
   echo "ERROR: can't open file $fn\n";
endif;
?>
```

Отметим особенность работы с файловым указателем. После чтения содержимого файла (текущего значения счетчика) указатель перемещается на первую позицию, следующую за прочитанными данными. Функция ftruncate (...) обнуляет размер файла, но оставляет файловый указатель на прежнем месте. Если после нее сразу выполнить запись в файл, то начало файла до текущей позиции указателя заполнится нулевыми байтами, после которых будет записано модифицированное значение счетчика. Чтобы этого не происходило, нужно перед записью в файл явным образом переместить его указатель на начало, что выполняется функцией rewind (...).

В примере с файловым счетчиком можно было бы вообще обойтись без использования функции ftruncate (...), потому что новое значение счетчика занимает в файле не меньше места, чем старое, и при перезаписи оно всегда будет перекрывать предыдущее значение.

О чем молчит документация

Рассмотрим два принципиально различных подхода к построению веб-приложений [2].

При первом диспетчер сервера получает запрос от клиентского приложения (веб-браузера), анализирует его и запускает на выполнение соответствующий контроллер, который динамически формирует ответ, для чего выполняет запросы к базе данных, опрашивает API вспомогательных сервисов и производит другие необходимые действия (см. рис. 3).

При втором подходе большая часть информации готовится контроллерами заранее в асинхронном по отношению к диспетчеру режиме. Задачей же диспетчера является разбор запроса и выдача уже подготовленного статического ответа (см. рис. 4).

Каждый подход имеет свои достоинства, недостатки и область применения, но в контексте вопроса, освещаемого в этой статье, рассмотрим возможную реализацию второго метода. Для выдачи содержимого файлов язык PHP предлагает три очень удобные функции, которые появились еще в четвертой версии интерпретатора:

- **file (...)** – загрузка содержимого файла в массив строк;
- **file_get_contents (...)** – загрузка содержимого файла в строковую переменную;
- **readfile (...)** – вывод содержимого файла на стандартное устройство.

Реализация задачи вывода файла по запросу не представляла бы никакого интереса, если бы файлы были абсолютно статичными. Однако это не всегда так. Файл с информацией о курсах валют должен обновляться раз в сутки, а с информацией о погоде и новостные – как минимум с интервалом в несколько минут. В такой ситуации вполне возможен конфликт между диспетчером, читающим запрошенный файл, и контроллером, который должен обновить содержимое файла. А значит, необходимо использовать механизмы блокировки, о которых говорилось выше.

Блокировку в языке программирования PHP можно установить только с использованием манипулятора файла – значения, возвращаемого функцией fopen (...). Но перечисленные выше функции работают на более высоком уровне абстракции, ограничиваясь именами файлов. Механизм низкоуровневого доступа к файлу скрыт от разработчика. Может быть, блокировки тоже реализованы где-то в недрах этих функций? В официальной документации [3] отсутствует ответ на этот вопрос.

Остается только исследовать поведение функций на тестовом стенде. Для этого воспользуемся двумя PHP-сценариями, имитирующими работу веб-сайта.

Сценарий update.php (см. листинг 4) осуществляет запись в файл index.html. Чтобы «растянуть» процесс во времени, запись информации осуществляется в два этапа, между которыми включена 10-секундная задержка. На всем протяжении записи файл index.html находится под исключительной блокировкой.

Листинг 4. Модуль update.php имитирует формирование HTML-файла контроллером

```
<?php
/*==================================================*/
/* Модуль, имитирующий изменение файла контроллером. */
/* $fn - имя файла (file name).                     */
/* $fp - манипулятор файла (file pointer).          */
/*==================================================*/
$pause = 10;
```

```
$fn = 'index.html';
$part1 = "<html><head><title>Test</title></head>";
$part2 = "<body><h1>Hello, world!</h1></body></html>";
$fp = fopen ($fn, file_exists ($fn)? 'r+': 'w');
if ($fp !== FALSE):
    if (flock ($fp, LOCK_EX)):
         echo "File $fn locked. Please wait...\n";
         ftruncate ($fp, 0);
         fputs ($fp, $part1);
         while ($pause > 0):
            echo "$pause ";
            sleep (1);
            $pause--;
         endwhile;
         fputs ($fp, $part2);
         flock ($fp, LOCK_UN);
         echo "\nFile $fn unlocked.\n";
   endif;
   fclose ($fp);
else:
   echo "Can't open file $fn.\n";
endif;
?>
```

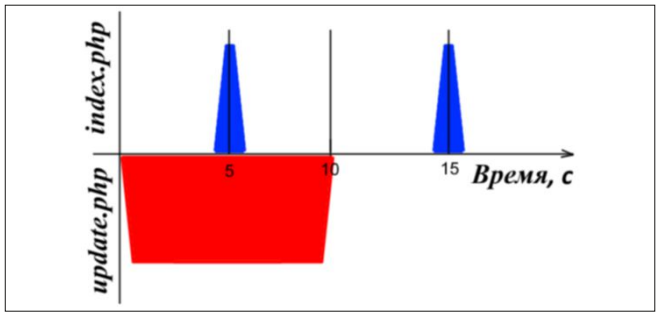
Сценарий index.php (см. листинг 5) производит чтение информации из файла index.html с помощью исследуемых функций file (...), file_get_contents (...) и readfile (...). Результат считывания выводится с помощью функции var_dump (...), позволяющей установить как состав данных, так и их тип.

Листинг 5. Модуль index.php имитирует чтение HTML-файла диспетчером

```
<?php
/*=============================================*/
/* Модуль, имитирующий чтение файла диспетчером */
/* без использования механизма блокировок.      */
/* $fn - имя файла (file name).                 */
/*=============================================*/
$fn = "index.html";
if (file_exists ($fn)):
    echo "1) file (...)\n";
   $r = file ($fn);
   var_dump ($r);
   echo "2) file_get_contents (...)\n";
   $r = file_get_contents ($fn);
   var_dump ($r);
   echo "3) readfile (...)\n";
   $r = readfile ($fn);
   var_dump ($r);
else:
    echo "Internal server error. File not found: \"$fn\".";
endif;
?>
```

При тестировании под операционной системой Windows 10 (Apache 2.4.23, PHP 7.0.16 и PHP 5.6.23) был использован сценарий операционной системы test.bat (см. листинг 6), автоматизирующий запуск модулей update.php и index.php в соответствии с планом тестирования, который изображен на диаграмме. Ход тестирования протоколировался в файл index.log.

Рисунок 5. Временная диаграмма процесса тестирования



Листинг 6. Сценарий test.bat для автоматизации совместного тестирования модулей update.php и index.php под Windows

```
@ECHO OFF
SET INDLOG=index7.log
SET PHPEXE=C:\EXE\PHP7\php.exe
ECHO TEST: %PHPEXE% > %INDLOG%
START %COMSPEC% /C %PHPEXE% update.php
TIMEOUT /T 5 /NOBREAK
ECHO *************** >> %INDLOG%
ECHO *** PHASE 1 *** >> %INDLOG%
ECHO *************** >> %INDLOG%
%PHPEXE% index.php >> %INDLOG%
TIMEOUT /T 10 /NOBREAK
ECHO.
ECHO *************** >> %INDLOG%
ECHO *** PHASE 2 *** >> %INDLOG%
ECHO *************** >> %INDLOG%
%PHPEXE% index.php >> %INDLOG%
```

Из анализа протокола (см. листинг 7) видно, что функции file (...) и file_get_contents (...) при чтении из файла, заблокированного другим процессом, вернули пустые значения: первая – пустой массив, вторая – пустую строку. Эти значения не являются признаками ошибки чтения, потому что по документации таковым является логическое значение FALSE. Похоже, что внутри реализации этих функций совершается попытка получить блокировку файла, и в случае неудачи они возвращают пустое значение.

Листинг 7. Под Windows функции file (...) и file_get_contents (...) используют разделяемую блокировку без ожидания ее получения, а функция readfile (...) не использует блокировок

```
TEST: C:\EXE\PHP7\php.exe
***************
*** PHASE 1 ***
***************
1) file (...)
array(0) {
}
2) file_get_contents (...)
string(0) ""
3) readfile (...)
<html><head><title>Test</title></head>int(38)
***************
*** PHASE 2 ***
***************
1) file (...)
array(1) {
  [0]=>
  string(80) "<html><head><title>Test</title></
head><body><h1>Hello, world!</h1></body></html>"
}
2) file_get_contents (...)
string(80) "<html><head><title>Test</title></
head><body><h1>Hello, world!</h1></body></html>"
3) readfile (...)
<html><head><title>Test</title></head><body><h1>Hello,
world!</h1></body></html>int(80)
```

Функция же readfile (...) возвращает частично записанную в файл информацию, что говорит о том, что она даже не пытается заблокировать файл, а читает его «как есть».

При проведении аналогичных испытаний на веб-сервере Ubuntu 16.10 (Apache 2.4.18 и PHP 7.0.8) с использованием

сценария test.sh (см. листинг 8) были получены неожиданные результаты. Все три функции вели себя так, как будто внутри них отсутствует механизм работы с блокировками файлов. Они вернули частичное содержимое заблокированного файла (листинг 9).

Листинг 8. Сценарий test.sh для автоматизации совместного тестирования модулей update.php и index.php под Linux

```
#!/bin/sh
INDLOG=index7.log
php update.php &
for i in 1 2 3 4 5; do echo -n "$i"; sleep 1; done
echo "****************" >>$INDLOG
echo "*** PHASE 1 ***" >>$INDLOG
echo "****************" >>$INDLOG
php index.php >>$INDLOG
for i in 1 2 3 4 5; do echo -n "$i"; sleep 1; done
echo "****************" >>$INDLOG
echo "*** PHASE 2 ***" >>$INDLOG
echo "****************" >>$INDLOG
php index.php >>$INDLOG
```

Листинг 9. Под Linux все тестируемые функции работают без использования блокировок

```
****************
*** PHASE 1 ***
****************
1) file (...)
array(1) {
  [0]=>
  string(38) "<html><head><title>Test</title></head>"
}
2) file_get_contents (...)
string(38) "<html><head><title>Test</title></head>"
3) readfile (...)
<html><head><title>Test</title></head>int(38)
****************
*** PHASE 2 ***
****************
1) file (...)
array(1) {
  [0]=>
  string(80) "<html><head><title>Test</title></
head><body><h1>Hello, world!</h1></body></html>"
}
2) file_get_contents (...)
string(80) "<html><head><title>Test</title></
head><body><h1>Hello, world!</h1></body></html>"
3) readfile (...)
<html><head><title>Test</title></head><body><h1>Hello,
world!</h1></body></html>int(80)
```

Может быть, блокировки консольных процессов PHP работают иначе, нежели блокировки, реализованные в модулях PHP сервера Apache? Однако испытания, проведенные со сценариями index.php и update.php в режиме доступа к ним через веб-интерфейс сервера, показали такие же результаты, как и выполнение их в режиме командной строки.

В очередной раз получено подтверждение известной истины о том, что использование недокументированных возможностей может приводить к непредсказуемым результатам. Так можно ли использовать функции file (...), file_get_contents (...) и readfile (...) для обработки локальных файлов сервера или их единственное предназначение – доступ к удаленным ресурсам по протоколам, которые все равно не поддерживают механизм блокировок?

Пользоваться этими функциями можно, если применять «блокирующую обертку», которая была предложена для функции fread (...) в начале этой статьи. Доработанный вариант сценария index.php приведен в листинге 10.

Листинг 10. Исправленный модуль index.php, использующий разделяемую блокировку

```
<?php
/*==============================================*/
/* Модуль, имитирующий чтение файла диспетчером */
/* с использованием разделяемой блокировки.     */
/* $fn - имя файла (file name).                 */
/*==============================================*/
$fn = "index.html";
if (file_exists ($fn)):
    $fp = fopen ($fn, 'r');
    if ($fp !== FALSE):
         if (flock ($fp, LOCK_SH)):
             echo "1) file (...)\n";
             $r = file ($fn);
             var_dump ($r);
             echo "2) file_get_contents (...)\n";
             $r = file_get_contents ($fn);
             var_dump ($r);
             echo "3) readfile (...)\n";
             $r = readfile ($fn);
             var_dump ($r);
             flock ($fp, LOCK_UN);
         else:
             echo "Can't lock file \"$fn\".\n";
         endif;
         fclose ($fp);
    else:
         echo "Can't open file \"$fn\".\n";
    endif;
else:
    echo "Internal server error. File not found: \"$fn\".\n";
endif;
?>
```

Как дальше быть

Сегодня благодаря современным архитектурным решениям в аппаратуре и программном обеспечении серверов файл уже не должен ассоциироваться с доступом к какой-то «медленной внешней памяти». SSD-диски, многоуровневое кэширование, быстрые интерфейсы привели к тому, что с практической точки зрения файл можно воспринимать как абстрактную структуру данных (наподобие массивов, списков и деревьев), предназначенную для длительного хранения состояния.

Реляционные базы данных, базы данных NoSQL и плоские файлы – все эти способы хранения данных имеют свои преимущества, недостатки и сферы применения. С ростом популярности микросервисной архитектуры веб-приложений у разработчиков появляется больше свободы в выборе хранилища данных для отдельных сервисов, а работа с файлами снижает зависимость от внешних ресурсов.

Все это говорит о том, что использование операций с файлами в PHP-приложениях как минимум не утратит своей актуальности. Поэтому программистам, которые будут с ними работать, важно следовать соглашениям, выработанным для использования разделяемых ресурсов в конкурентной среде. При этом недопустимо полагаться на предположения, которые не описаны явно в документации. Нарушение этих рекомендаций приведет к труднообнаруживаемым ошибкам и низкому качеству разрабатываемых приложений. EOF

[1] Кухарчик А. PHP: обучение на примерах. – Мн.: Новое знание, 2004.

[2] Афонин С. М. Программирование на языке PHP. – М.: НТ Пресс, 2007.

[3] Официальная документация PHP – http://php.net/docs.php.

Ключевые слова: PHP, программирование, файлы, доступ.

Визитка

КИРИЛЛ ТКАЧЕНКО,
инженер 1-й кат., ФГАОУ ВО «Севастопольский государственный университет»,
tkachenkokirillstanislavovich@gmail.com

Программная реализация
энигмоподобной системы в среде 1С

Рассмотрим реализацию узлов инициализации, шифрования и дешифрования для энигмоподобной системы на встроенном языке 1С

Роторная шифровальная машина «Энигма» широко известна из технической литературы [1, 2], художественных произведений и кинематографа. Она и ее клоны оставили глубокий след в криптографии и криптоанализе, вызвав их скачкообразное развитие. Кроме прочего, сведения технического, алгоритмического и конструктивного характера по ней доступны в открытой печати. Поэтому интересно рассмотреть пути ее реализации в современных условиях.

В статье рассмотрим программную реализацию на встроенном языке 1С отдельных узлов несовместимой с оригинальной энигмоподобной роторной шифровальной машины.

> Развитые возможности встроенного языка программирования 1С позволяют обеспечить функционирование всех необходимых возможностей энигмоподобной шифровальной системы

В полной мере воспроизвести шифровальную машину не представляется возможным. Более того, многие ее аппаратные, аппаратно-программные и полностью программные реализации [1, 2] будут отличаться друг от друга и от эталона: по числу колес, алфавиту и числу символов алфавита на колесе ротора; по способу кодирования символов алфавита; по начальному заполнению колес ротора и их начальному повороту; по способу поворота колес; наконец, для программных реализаций, по способу обратного преобразования (построение обратных таблиц для колес, прямой поиск в обратном направлении и прочее).

Прежде чем начинать работу, необходимо обратить внимание на имеющиеся русскоязычные публикации.

Достаточно известна монография [1] с объемными, подробными сведениями исторического характера. Но, к сожалению, программа на языке QBasic, приведенная в листинге этой книги, обладает существенным недостатком. Она работает не со всеми 256 однобайтовыми символами, а с их подмножеством, «срезом» кодировки – прописными русскими буквами. При этом происходит преобразование прописных русских букв в их порядковые номера в алфавите (А – 0, Б – 1 и так далее) и наоборот способом, пригодным только для однобайтовой кодировки OEM-866. Внутренние преобразования с кодированными подобным способом буквами также в некоторой степени основаны на этом преобразующем соответствии. Этот способ хранения символов значительно сдерживает портирование этой конкретной реализации на многие другие окружения, хоть немного отличные от DOS/QBasic.

Усовершенствованная реализация [2] достаточно сильно отличается от классических роторных машин. Например, шифрование и дешифрование выполняются не над конечным подмножеством символов алфавита, а над отдельными байтами входной информации. Добавлены существенные усложнения в узлы инициализации таблиц перестановок. Перенос в актуальные окружения затруднен использованием не только возможностей операционной системы, но и вызовов BIOS.

В современных реалиях можно остановиться на программном комплексе 1С. Развитые возможности его встроенного языка программирования позволяют обеспечить функционирование всех необходимых возможностей энигмоподобной шифровальной системы.

Учитывая вышесказанное, можно, несколько упрощая, описать процесс шифрования и дешифрования сообщений в таких системах.

Перед началом как процесса шифрования, так и процесса дешифрования выполняется настройка колес ротора. Каждое колесо выполняет шифрование отдельного символа сообщения по шифру простой замены. Но выходной зашифрованный символ подается на вход следующего колеса. Получается, что символ последовательно проходит

через цепочку колес и оказывается на выходе системы. После выполнения шифрования отдельного символа колеса поворачиваются. Для дешифрования процесс выполняется в обратном порядке для колес. Происходит выбор расшифрованного варианта символа на основе текущего рассматриваемого.

Чтобы придать происходящим процессам подобие исходных конструкций образца, в прямом виде символы алфавита фигурировать не будут. Будет использоваться их порядковый номер начиная с нуля, в алфавите, для прописных букв: 0 – А, 1 – Б, 2 – В, …

Поэтому построение программы начинается с написания подпрограммы инициализации ротора. Ротор будет помещен в двумерный массив:

```
Ротор = Новый Массив(МаксНомерКолеса + 1, МаксНомерСимвола + 1);
```

Номера строк – 0, 1, …, МаксНомерКолеса, номера – столбцов 0, 1, …, МаксНомерСимвола, где МаксНомерКолеса – максимальный номер колеса, МаксНомерСимвола – максимальный номер символа.

Чтобы иметь возможность шифровать и дешифровать сообщения, но при этом обеспечить псевдослучайный характер начального заполнения колес, создается объект генератора псевдослучайных чисел:

```
ГСЧ = Новый ГенераторСлучайныхЧисел(МойКлюч);
```

МойКлюч является целым положительным числом. Это число будет являться единственным ключом в этой симметричной системе шифрования.

В циклах, просматривая все строки массива, от 0 до МаксНомерКолеса, выполняется заполнение колес. Для этого вначале каждому j-му символу i-го колеса присваивается его порядковый j-й номер. Затем для каждого символа, от 0 до МаксНомерСимвола, происходит его обмен со случайным соседом по строке.

Ограниченное подмножество символов алфавита явно прописывается в программе:

```
Алфавит = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
```

Наибольший номер колеса также прописывается явно, но наибольший номер символа определяется исходя из длины строки с алфавитом. После этого можно задать значение ключа МойКлюч, исходную строку для дальнейшей работы СтрокаИсходная, присвоить пустые значения переменным СтрокаЗашифрованная и СтрокаРасшифрованная – шифрованной и дешифрованной строкам соответственно.

Дальше выделяются две фазы – шифрование и дешифрование.

Шифрование начинается с создания нового ротора, учитывая текущее значение ключа МойКлюч, и границы двумерного массива МаксНомерКолеса, МаксНомерСимвола. Каждый символ шифруется отдельно. Вначале символ преобразуется в его порядковый номер в текущем алфавите: А – 0, Б – 1, В – 2, …:

```
k = Найти(Алфавит, Сред(СтрокаИсходная, i, 1));
k = ?(k > 0, k - 1, 0);
```

Выполняется шифрование отдельного символа. Он прогоняется через колеса, от первого к последнему, каждый раз при этом находится его новый следующий код:

```
Для j = 0 По МаксНомерКолеса Цикл
        k = Ротор[j][k];
КонецЦикла;
```

Наконец, результат шифрования после преобразования в один из допустимых символов алфавита добавляется в конец зашифрованной строки:

```
СтрокаЗашифрованная = СтрокаЗашифрованная + Сред(Алфавит, ↵
        k + 1, 1);
```

После этого производится поворот колес, при этом для каждого колеса его начальный элемент становится последним, следующий за ним – первым:

```
Для j = 0 По МаксНомерКолеса Цикл
        Ротор[j].Добавить(Ротор[j][0]);
        Ротор[j].Удалить(0);
КонецЦикла;
```

Фрагмент кода для поворота ротора нецелесообразно выделять в особую подпрограмму, поскольку длина комментированного или некомментированного программного кода может возрасти по числу строк.

Дешифрование также начинается с создания нового ротора, учитывая текущее значение ключа МойКлюч, и границы двумерного массива МаксНомерКолеса, МаксНомерСимвола. Строка дешифруется посимвольно, также символ для внутренней работы преобразуется в его код в алфавите. Для дешифрования по каждому колесу от последнего к первому производится поиск в колесе номера ячейки с текущим кодом. Этот номер и становится инвертированным следующим номером кода:

```
j = МаксНомерКолеса;
Пока j >= 0 Цикл
        k = Ротор[j].Найти(k);
        j = j - 1;
КонецЦикла;
```

Дешифрованный код символа преобразуется в алфавитный и добавляется в конец расшифрованной строки. Поворот колес реализуется аналогичным образом, что и в шифровании.

Полный листинг предлагаемой программы:

```
// Функция НовыйРотор
//
// Параметры:
//  МойКлюч - целое положительное число, которое будет
//  использоваться для шифрования и дешифрования,
//  как начальное значение генератора
//  МаксНомерКолеса - максимальный номер колеса
//  МаксНомерСимвола - максимальный номер символа
//
// Возвращаемое значение:
//   Созданный ротор
//
Функция НовыйРотор(МойКлюч, МаксНомерКолеса, МаксНомерСимвола)
    // Создается двумерный массив Ротор,
    // с номерами строк 0, 1, ..., МаксНомерКолеса,
```

```
    // номерами столбцов 0, 1, ..., МаксНомерСимвола
    Ротор = Новый Массив(МаксНомерКолеса + 1, ↵
         МаксНомерСимвола + 1);
    // Создается новый объект ГенераторСлучайныхЧисел,
    // начальным значением которого является МойКлюч
    ГСЧ = Новый ГенераторСлучайныхЧисел(МойКлюч);
    // Для всех строк массива, то есть для всех колес ротора,
    // 0, 1, ..., МаксНомерКолеса
    Для i = 0 По МаксНомерКолеса Цикл
         // Для каждого символа, 0, 1, ..., МаксНомерСимвола
         Для j = 0 По МаксНомерСимвола Цикл
             // присваивается его порядковый номер
             // по умолчанию, 0, 1, ..., МаксНомерСимвола
             Ротор[i][j] = j;
         КонецЦикла;
         // Для каждого номера символа, 0, 1, ...,
         // МаксНомерСимвола, производится его обмен
         // с псевдослучайным элементом
         Для j = 0 По МаксНомерСимвола Цикл
             // Выбирается псевдослучайный индекс в массиве
             t1 = ГСЧ.СлучайноеЧисло(0, МаксНомерСимвола);
             // Производится обмен
             t2 = Ротор[i][j];
             Ротор[i][j] = Ротор[i][t1];
             Ротор[i][t1] = t2;
         КонецЦикла;
    КонецЦикла;
    // Возврат созданного ротора
    Возврат Ротор;
КонецФункции

// Используемое подмножество символов алфавита
Алфавит = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
// Наибольший номер колеса, 0, 1, ..., МаксНомерКолеса
МаксНомерКолеса = 6;
// Наибольший номер символа, 0, 1, ..., МаксНомерСимвола
МаксНомерСимвола = СтрДлина(Алфавит) - 1;

// Ключ, который используется для шифрования и дешифрования,
// целое положительное число
МойКлюч = 1;
// Исходная строка для шифрования
СтрокаИсходная = "ААААААААМАМАМЫЛАРАМУАААААААА";
// Ее зашифрованный вариант
СтрокаЗашифрованная = "";
// Расшифрованный вариант зашифрованного текста
СтрокаРасшифрованная = "";

// Процесс шифрования
// Создается новый ротор. Псевдослучайная последовательность
// обменов символов в колесах опеределяется ключом, начальным
// значением генератора
Ротор = НовыйРотор(МойКлюч, МаксНомерКолеса, МаксНомерСимвола);
// Для каждого символа строки, 1, 2, ..., длина строки
Для i = 1 По СтрДлина(СтрокаИсходная) Цикл
    // Получается номер символа в алфавите, в данном случае,
    // А - 0, Б - 1, ...
    k = Найти(Алфавит, Сред(СтрокаИсходная, i, 1));
    k = ?(k > 0, k - 1, 0);
    // Символ шифруется. В каждом колесе от первого к последнему
    Для j = 0 По МаксНомерКолеса Цикл
         // Находится следующий код, соответствующий символу
         k = Ротор[j][k];
    КонецЦикла;
    // Зашифрованный символ, выбранный по номеру из алфавита,
    // добавляется к результату
    СтрокаЗашифрованная = СтрокаЗашифрованная + ↵
         Сред(Алфавит, k + 1, 1);
    // Производится поворот колес. Для каждого колеса
    Для j = 0 По МаксНомерКолеса Цикл
         // Первый элемент - символ - становится последним
         Ротор[j].Добавить(Ротор[j][0]);
         Ротор[j].Удалить(0);
    КонецЦикла;
КонецЦикла;

// Процесс дешифрования
// Создается новый ротор. Псевдослучайная последовательность
// обменов символов в колесах опеределяется ключом, начальным
// значением генератора
Ротор = НовыйРотор(МойКлюч, МаксНомерКолеса, МаксНомерСимвола);
// Для каждого символа строки, 1, 2, ..., длина строки
Для i = 1 По СтрДлина(СтрокаЗашифрованная) Цикл
    // Получается номер символа в алфавите, в данном случае,
    // А - 0, Б - 1, ...
    k = Найти(Алфавит, Сред(СтрокаЗашифрованная, i, 1));
    k = ?(k > 0, k - 1, 0);
    // Символ дешифруется. В каждом колесе от последнего
    // к первому
    j = МаксНомерКолеса;
    Пока j >= 0 Цикл
         // Находится предыдущий код, соответствующий символу
         k = Ротор[j].Найти(k);
         j = j - 1;
    КонецЦикла;
    // Дешифрованный символ, выбранный по номеру из алфавита,
    // добавляется к результату
    СтрокаРасшифрованная = СтрокаРасшифрованная + ↵
         Сред(Алфавит, k + 1, 1);
    // Производится поворот колес. Для каждого колеса
    Для j = 0 По МаксНомерКолеса Цикл
         // Первый элемент - символ - становится последним
         Ротор[j].Добавить(Ротор[j][0]);
         Ротор[j].Удалить(0);
    КонецЦикла;
КонецЦикла;

// Вывод исходной, зашифрованной и расшифрованной строки
Сообщить(СтрокаИсходная);
Сообщить(СтрокаЗашифрованная);
Сообщить(СтрокаРасшифрованная);
```

Примеры выполнения программы:

```
ААААААААААААААААААААААААААА
ГЯОВФДМЩЕХЯОХРКЭВФЕУОЮЩЪЁГЧ
ААААААААААААААААААААААААААА

ББББББББББББББББББББББББББ
МПВЩЫЕЯЮЁЙНЧТНЁУНКЙИЙРЦКГОЙ
ББББББББББББББББББББББББББ

ВВВВВВВВВВВВВВВВВВВВВВВВВВ
ЩЬЙЮСНУКЬМЭИШГЪПНЩШБАЯЕДШУТЦ
ВВВВВВВВВВВВВВВВВВВВВВВВВВ

ГГГГГГГГГГГГГГГГГГГГГГГГГГГ
ТОШИТГВЖНСФХИУЦТЫЪЬНРЗЖЖФТЗЖ
ГГГГГГГГГГГГГГГГГГГГГГГГГГГ

ААААААААМАМАМЫЛАРАМУАААААААА
ГЯОВФДМЩЁХКЯГСЖКЕВЯВУОЮЩЪЁГЧ
ААААААААМАМАМЫЛАРАМУАААААААА
```

• • •

Рассмотренная система, конечно, не сможет найти широкое применение в общей практике 1С-программистов. Но в ряде случаев, например для обфускации каких-либо данных, для предотвращения неправильной и некорректной эксплуатации решений, ее применение может стать приемлемым. EOF

[1] Жельников В. Криптография от папируса до компьютера / В. Жельников. – М.: ABF, 1996. – 335 с. – ISBN 5-87484-054-0.

[2] Текин В.В. Усовершенствованная версия «Энигмы» / В.В. Текин // «Мир ПК», № 6, 2007 г. – С. 64-65.

Ключевые слова: Энигма, 1С.

бизнес & информационные технологии

БИТ

Марина Аншина – известный эксперт в области управления ИТ, автор нескольких книг, преподаватель, председатель Комитета по стандартам СоДИТ, делится своими идеями и наработками в сфере

Чем хороша ее новая книга?

Во-первых, Марина Аншина провела сравнительный анализ разнообразных методов управления ИТ-проектами, что очень ценно.

Во-вторых, издание буквально пропитано обширным опытом автора по выполнению ИТ-проектов.

В-третьи, это подробное практическое пособие – можно открыть его и методично переходить при работе над проектом от этапа к этапу, руководствуясь подсказками автора.

В-четвертых, книга снабжена большим справочным материалом.

И наконец, что немаловажно, читать ее легко и приятно – она написана хорошим русским языком.

Бумажная версия – 550 руб. + доставка

Электронная версия – 275 руб.

Покупайте и читайте на бумаге или экране!

samag.ru/catalogue

Андрей Пролетарский:
«Информационные технологии – это дорога с двусторонним движением»

В гостях у «Системного администратора» Андрей Пролетарский, декан факультета «Информатика и системы управления» Московского государственного технического университета имени Н.Э. Баумана, директор НОЦ «Технопарк», доктор технических наук, доцент, автор более 100 научных трудов по информационным технологиям и системам управления

Подготовила Ирина Ложкина

Андрей Пролетарский, декан факультета «Информатика и системы управления» Московского государственного технического университета имени Н.Э. Баумана, директор НОЦ «Технопарк», доктор технических наук, доцент, автор более 100 научных трудов по информационным технологиям и системам управления

Справка

Бауманский университет – национальный университет техники и технологий – проводит исследования по приоритетным и самым передовым направлениям науки, техники и технологий, базирующиеся на восьми технологических платформах. 32 компании включили МГТУ в свои программы инновационного развития. Сегодня университет реализует 90 крупных научных проектов по различной тематике. Бауманский университет – учредитель фонда «Сколково». Сайт университета: – http://www.bmstu.ru

– «Бауманка» – это бренд, и учиться в МВТУ – мечта многих молодых людей. Насколько реально к вам поступить?

– Вполне реально. У нас довольно большой набор, примерно 700 бюджетных мест. Дополнительных испытаний мы не проводим, хотя, возможно, и придем к этому, принимаем по результатам ЕГЭ. Абитуриенты, имеющие не менее 240 баллов, могут рассчитывать на успех. Ну а ребята, которые учились в школе слабо, наше образование просто не потянут. Есть у нас и платные группы, где требования почти так же высоки.

Сохранили практику «целевого набора» – это когда государство заказывает обучение в своих интересах. Есть отраслевые факультеты на ведущих предприятиях страны. Особенность здесь в том, что, помимо учебы, один день ребята работают на предприятии. Преференции при поступлении получают победители олимпиад.

Я бы сказал, что к нам не столь трудно поступить, сколь сложно учиться. Какое-то количество поступивших не выдерживают напряженного ритма, отсеиваются. Но их не более 5-8 процентов. Кстати, хочу сказать, не знаю случаев исключения студентов из-за того, что они были не в состоянии освоить программу. Основные причины – разгильдяйство и неорганизованность. Так что остаются те, кто умеет вкалывать и планировать свое время. Но результат того стоит.

– ИТ-специальности – на пике востребованности. Но бытует мнение, что институтские знания и навыки сильно отстают от стремительно развивающихся ИТ.

– Невозможно стать классным специалистом, не имея отличной базовой подготовки. Без глубокого знания математики, физики, информатики и досконального понимания сути процессов и технологий нельзя ни освоить

эти процессы и технологии, ни применять, ни тем более их создавать.

Мы организуем процесс обучения студентов таким образом, чтобы создать синергетический эффект. Даем серьезные научные знания и учим применять их на практике. Ведь образование в информационных технологиях – это дорога с двусторонним движением! Я имею в виду совместную работу университета и работодателей.

На нашем факультете 11 кафедр. Только недавно открыли четыре новых направления. Это и big data, и интернет вещей, и интеллектуальные системы, и… Мы растем! Именно потому, что сфера ИТ развивается столь стремительно.

Мы живем в цифровом обществе. Всем нужна информация. И мы готовим специалистов по работе с ней – от ее извлечения, обработки до хранения, защиты и использования. И теоретические знания, полученные от наших преподавателей, а у нас очень сильный педагогический состав, подкрепляют практические знания, которые дают специалисты крупнейших компаний, ведущие у нас занятия. Плюс самостоятельная работа студентов в различных реальных проектах.

Да, наших выпускников не приходится ни переучивать, ни доучивать. Наше образование ничуть не хуже западного. Великолепные возможности и прекрасные результаты дает «Технопарк» – совместный образовательный проект Mail.RuGroup и МГТУ, реализующий практико-ориентированное бесплатное дополнительное обучение студентов для интернет-сферы.

У нас много лабораторий и центров компетенций с ведущими российскими и зарубежными ИТ-компаниями, например КРОК, РТСофт, «Эшелон», «1С», Cisco, IBM, D-Link, National Instruments, и даже с банками. Прекрасные отношения с оборонным сектором.

– В таком случае ваши выпускники, видимо, нарасхват на рынке труда?

– К моменту выпуска расхватывать уже некого. Начиная с третьего курса все ребята уже фактически трудоустроены – как правило, они уже определились с выбором компании, трудятся в том или ином проекте и просто плавно перетекают из вуза на работу. Выгоды ясны и очевидны. Выпускник имеет практически 100% гарантию трудоустройства, интересную и, что немаловажно, хорошо оплачиваемую работу, мы же уверены, что дали молодому человеку надежную путевку в жизнь.

Профессиональная подготовка осуществляется на отраслевых факультетах, созданных на базе крупных предприятиях, организаций и учреждений оборонно-промышленного комплекса, расположенных в Москве и подмосковных городах: Реутове, Красногорске и Королеве, а также в филиале университета в Калуге

Не секрет, что самое сложное – найти свою первую настоящую работу. Да, по специальности трудятся 99,9% наших бывших студентов. И в ИТ-компаниях, и в крупных госкорпорациях, и в банках и финансовых организациях. Кто-то уезжает за рубеж, но отток все время сокращается. Сейчас и на отечественном рынке много достойных вариантов получить увлекательную и перспективную работу и конкурентоспособную заработную плату. Ну и развитие фриланса упрощает ситуацию.

– ИТ уже перестали быть «мужской» специализацией.

– Да, прошли те времена, когда в группе учились 1-2 девочки. Девушек у нас на факультете немало, умных, способных, дисциплинированных, в основном гораздо более дисциплинированных, чем ребята. Да и работодатели пересмотрели свои взгляды и с одинаковой охотой принимают к себе выпускников обоего пола.

Кстати, мало кто из абитуриентов задумывается о таком важном моменте, как создание семьи с достойным

Справка

- **Кафедра ИУ1.** Занимается всем диапазоном исследований и разработок от простых систем управления одиночными объектами до сложнейших космических и производственных комплексов.
- **Кафедра ИУ2.** Занимается разработкой прецизионных элементов приборов и сервисной электроники, систем ориентации, навигации и управления подвижными роботами.
- **Кафедра ИУ3.** Готовит специалистов в области разработки информационных и телекоммуникационных систем.
- **Кафедра ИУ4.** Охватывает радиоэлектронные, электронно-вычислительные, наноэлектронные и наносистемные средства, средства телекоммуникаций, технологические процессы производства электронных средств.
- **Кафедра ИУ5.** Основными направлениями обучения являются программирование, информационное обеспечение, вычислительные сети и телекоммуникации.
- **Кафедра ИУ6.** Девиз кафедры: дать фундаментальную подготовку студенту по широкому спектру направлений вычислительной техники и информационных технологий.
- **Кафедра ИУ7.** Готовит специалистов широкого профиля в области проектирования и разработки программного обеспечения.
- **Кафедра ИУ8.** Основным научным направлением и областью подготовки специалистов является комплексное обеспечение информационной безопасности автоматизированных систем.
- **Кафедра ИУ9.** Подготовка специалистов по математическому и программному обеспечению высокотехнологичных областей техники и современных информационных технологий с уклоном в высокоэффективное программирование.
- **Кафедра ИУ11.** Готовит специалистов в широкой области информационных технологий и систем.

человеком, а зря. У нас год за годом все больше студенческих семей или пар, закончивших наш вуз. Согласитесь, общность интересов и возможность хорошо узнать друг друга за время учебы – серьезная база для прочных отношений.

– Безусловно, не учебой единой... Общежитие предоставляется? Возможности для спорта, творчества, отдыха?

– Да, можно рассчитывать на общежитие, кстати, вуз активно строит новое. Есть спортивный комплекс с бассейном. Нужно сказать, что традиционно в ИТ много разносторонне талантливых ребят. Есть Дворец культуры и самые разнообразные творческие коллективы. Перечисление возможностей творческой самореализации – это тема для отдельной статьи. Есть конкурс «Дебют на Бауманской сцене», где принимают участие индивидуальные исполнители, «Бауманиада» – смотр-конкурс факультетских команд. Всего достаточно.

И в технических конкурсах и форумах, от университетского и российского до международного уровней, участвуют – и побеждают – наши студенты. EOF

Спонсоры нас любят. Почему? Ответ простой – мы обеспечиваем им достойный кадровый резерв

Алексей Попов, доцент кафедры ИУ-6, кандидат технических наук

Интернет вещей – тема сейчас на слуху и, безусловно, многим интересна. Особенно молодым людям, только выбирающим для себя профессию, связанную с ИТ-технологиями. Но еще очень многие ИТ-вузы только пробуют к ней подступиться.

А мы на нашей кафедре (ИУ-6) уже готовим специалистов в этой сфере! Программа обучения – наше ноу-хау – самая передовая в данной области. Она основана на сочетании классического подхода – мы даем фундаментальное образование и проводим прикладные исследования и практические занятия на самом современном оборудовании, предоставленном нам компаниями-спонсорами.

Могу сказать, что все большее количество компаний тесно с нами сотрудничает. Да, спонсоры нас любят. Почему? Ответ простой – мы обеспечиваем им достойный кадровый резерв.

На современном витке развития информационных технологий ни один серьезный проект не может обойтись без использования облаков. Умные технологии, умные города, умные дома, умную технику нужно внедрять уже сегодня. И наши студенты учатся проектированию киберфизических систем, решений интернета вещей, разработке распределенных вычислительных комплексов – тому, что должно управлять большим количеством оборудования.

Ребята разрабатывают самые настоящие проекты, которые далее можно внедрять. Особенно интересными стали дипломные проекты – они являются готовой базой для стартапов. Ну а мы стараемся помочь их авторам в продвижении работы, в поисках инвесторов.

К примеру, есть такая разработка – условная «майка», с системой, позволяющей слабослышащим и слабовидящим людям свободнее ориентироваться во внешней среде. Создан специальный «импульсный алфавит», благодаря которому к человеку поступает необходимая информация в режиме онлайн.

Для контроля обстановки в труднодоступных лесных районах Сибири и Дальнего Востока спроектирована система передачи данных на большие расстояния. Наблюдение за тем, нет ли пожаров или незаконных вырубок леса, становится более точным и экономически выгодным, нежели со спутника.

Ну и наши знаменитые хакатоны! Самая интересная и живая форма обучения. Мы их встроили в учебный процесс. Надо сказать, что способности наших студентов порой бывают недооценены, а в рамках креативной, интересной работы ребята раскрываются самым лучшим образом. Ну а потом уже приходится «держать планку»! EOF

Хакатоны — самая интересная и живая форма обучения. Мы их встроили в учебный процесс. Надо сказать, что способности наших студентов порой бывают недооценены, а в рамках креативной, интересной работы ребята раскрываются самым лучшим образом. Ну а потом уже приходится «держать планку»!

Главное, что дает Бауманка, – это правильный настрой и подход к решению задач и проблем

Марк Макарычев, студент ИУ6-81И

Бауманская система обучения, на мой взгляд, очень правильная и во все времена актуальна.

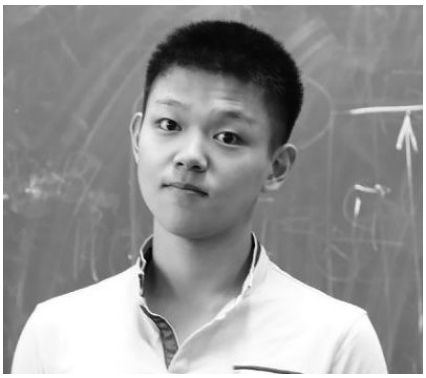
На первых курсах дается хорошая база по фундаментальным предметам и приучают к трудолюбию.

На последних же курсах появляются самые интересные и современные дисциплины наряду с теми, которые на первый взгляд могут показаться устаревшими. Но интересно то, что и эти предметы придают некий шарм учебному процессу.

Помимо общения с крайне интересными одногруппниками и преподавателями, пожалуй, главное, что дает Бауманка, – это правильный настрой и подход к решению задач и проблем. Ведь это то, что не только на любой работе пригодится, но и в жизни в целом. EOF

Стоит отметить вкусную еду в столовых, так как голодный студент – плохой студент

Ли Цзяцзянь, студент ИУ6-81И



В МГТУ им. Н.Э. Баумана я получил не только качественные знания, но и незабываемые эмоции. Здесь и восторг первых дней учебы, и радость от общения с новыми друзьями, и волнения перед экзаменами. Обучение здесь – это возможности: изучение различных дисциплин, чтобы расширить свой кругозор; понимание принципов работы технических средств и шанс создать нечто свое; взаимодействие с крупными компаниями, такими как Samsung и IBM, где я смог пройти практику и получить опыт в работе с передовыми технологиями. Также стоит отметить вкусную еду в столовых, так как голодный студент – плохой студент. Также учеба в МГТУ им. Н.Э. Баумана запомнилась яркими и интересными культурными событиями, проводимыми университетом. EOF

Я попробовал себя практически во всех направлениях современной ИТ-индустрии

Алексей Комаров, студент АК5-81

Обучение в МГТУ им. Н. Э. Баумана дало мне широкий простор для души и для жизни, если можно так выразиться.

За годы в университете я попробовал себя практически во всех направлениях современной ИТ-индустрии: программировал микроконтроллеры, занимался разводкой печатных плат, проектировал локально-вычислительные сети предприятия, собственноручно сверстал с десяток сайтов и выпустил мобильное приложение, и это лишь малая часть того, чем я занимался и продолжаю заниматься, а то ли еще будет!

МГТУ закладывает хорошую базу – фундамент для твоего дальнейшего развития. Учебный процесс построен таким образом, чтобы ты постоянно самосовершенствовался, перенимал опыт у настоящих профессионалов своего дела и сразу же применял полученные знания на практике.

Кроме того, МГТУ – это еще и студенты, которые тебя окружают. Пройдя тернистый путь от первого до четвертого курса, я приобрел настоящих друзей, на которых всегда могу положиться. EOF

Вакансия: веб-аналитик

В эпоху Web 2.0 центральное место в ИТ занимают веб-технологии. Одной из сторон их функционирования стало накопление большого количества информации. Это привело к необходимости оптимизации процессов ее анализа и принятия решений в целом и усовершенствования работы веб-аналитиков в частности. Мы попросили представителей компаний рассказать о знаниях, навыках, опыте, актуальных для ИТ-профессии веб-аналитик

1. **Веб-аналитик: какими знаниями и навыками он должен обладать?**
2. **Каков инструментарий веб-аналитика?**
3. **Каковы требования компании к уровню образования потенциальных сотрудников?**
4. **Какие требования предъявляются к опыту работы?**
5. **Есть ли особые требования, которые обусловлены спецификой деятельности компании?**

Владислав Денисов,
веб-аналитик Sports.ru | Tribuna Digital

1 Профессия находится на стыке продукта и разработки. Необходимо понимать назначение анализируемого продукта и основные сценарии поведения пользователей для грамотного составления и объяснения отслеживаемых метрик. Аналитик должен обладать навыками программирования (для анализа данных отдают предпочтение языкам Python/R); уметь извлекать данные из различных источников (SQL, API, текст, Web Scraping); знать, как работают инструменты, которые использует; и, самое важное, владеть на высоком уровне аппаратом математической статистики и ее прикладных дисциплин.

2 Система аналитики (из бесплатных, например, Google Analytics, Яндекс.Метрика, Piwik); терминал; Python/R и дополнительные библиотеки; инструменты BI.

3 Идеально – математическое, но достаточно и технического. На уровень университета не смотрим, проверяем только знания..

4 На позицию младшего аналитика рассматриваем и людей без опыта работы, в т.ч. еще обучающихся студентов. На данный момент у нас внутри команды уже выросли два специалиста такого профиля. От специалистов с более полным резюме мы ждем навыков владения перечисленными выше инструментами, в больше или меньшей степени – это зависит от продолжительности работы в прошлых компаниях. В целом тут можно отметить, что быстро прокачиваются специалисты в digital, e-commerce областях, т.к. там веб-аналитике действительно уделяют повышенное внимание и знания, там получаемые, релевантны нашим требованиям.

5 Работа предполагает увлечение спортом хотя бы на уровне зрителя. Нужно будет работать с данными спортивной статистики, знать или быстро разбираться в основных метриках и терминах футбола, хоккея, баскетбола и других видов спорта.

Поскольку Tribuna Digital живет не только в вебе, но и в мобайле, нужно понимать особенности мобильной аналитики.

Помимо непосредственной аналитики нужно заниматься различными задачами data engineering, например написанием и поддержкой ETL-системы.

Сергей Ляшенко,
руководитель отдела продуктового анализа в Tutu.ru

1 Аналитик – это прежде всего человек, который работает с данными. А данные в наш век – понятие растяжимое. Это может быть пара чисел из отчета по обзору рынка или несколько терабайт логов – все нужно обработать и выдать в разумные сроки вывод для заказчика. Калькулятором и школьным уровнем математики тут не обойтись – нужна статистика, нужно иметь определенные технические навыки, чтобы обрабатывать большие объемы данных, но это все не главное. Аналитик должен не только уметь обрабатывать данные, но и любить это делать. У нас это называется «копать»: любой технически грамотный человек может посчитать среднее, но аналитик должен понять, что скрывается за этим средним. Простой пример, вы перекрасили

свой сайт из голубого в розовый и в АБ-тесте получили нулевой прирост конверсии. Многие в этом месте скажут: для роста конверсии нужно улучшение функциональности сайта, простым «перекрашиванием» этого не достичь. Но хороший аналитик обязан попытаться найти сегменты, на которых наблюдается улучшение конверсии, и, возможно, дополнительные точки роста для продукта, над которым он работает. Ведь вполне вероятно, что для какого-то сегмента (условных женщин) мы будем наблюдать повышение конверсии, в то время как для другого (условных мужчин) – понижение, что в среднем даст нам ноль.

2 Последние годы мир веба сильно развивается, вместе с ним развивается и инструментарий аналитика. Когда-то было достаточно знания Excel, навыков работы с интернет-счетчиками (LiveInternet, Яндекс Метрика) и немного языка запросов к реляционной БД. Сейчас с ростом объемов данных уже не обойтись без умения работать со специальными хранилищами и средствами обработки Больших Данных. Абсолютно все аналитики в нашей компании знают и используют в работе Python, т.к. он имеет API для большинства современных хранилищ данных. Кроме того, мы используем хранилище на основе технологии ElasticSearch.

3 Необходимый минимум – высшее техническое образование. Мы, как правило, рассматриваем выпускников мехмата, физфака, ВМК МГУ, МФТИ, МИФИ. Но это скорее тенденция, т.к. образование – только некий сигнал о том, кто наш соискатель. Ключевые навыки мы проверяем на собеседованиях.

4 Зачастую мы нанимаем специалистов без опыта работы и в течение нескольких месяцев обучаем их до необходимого уровня. Печально, но факт: курс статистики, который обычно проходят на втором-третьем курсе, новоиспеченные выпускники успевают благополучно забыть за ненадобностью, и их приходится обучать каким-то азам заново. При этом мы отбираем людей, которые могут и хотят учиться, а главное, смогут научиться и начать применять знания на практике.

5 Пожалуй, главное требование – соискателю должна быть интересна предметная область, в которой ему предстоит работать. Мы предпочитаем брать людей, которые любят путешествовать. Но таких, думаю, большинство.

Константин Игнатьев,
руководитель отдела контентных аналитиков «Лаборатории Касперского»

1 Должность предполагает глубокие знания в том, как устроен интернет, как работают веб-серверы, обязательно должен точно понимать, как строятся сайты, как и почему они отображаются в том или ином виде. Очень желателен опыт веб-верстки, веб-разработки. Веб-контент-аналитик должен не только с технической стороны понимать, как все устроено, но и что в интернете происходит, что интересно пользователям именно сейчас, какие есть тенденции в вебе, в соцсетях, что пользователям интересно в данный момент, в прошлом, и, возможно, угадывать, что «выстрелит» в будущем. Однозначно аналитик должен иметь аккаунты во всех соцсетях и представлять, как они работают и какие угрозы там есть.

В своей работе веб-контент-аналитик сталкивается с обработкой большого количества данных, поэтому требуется использование скриптовых языков, например Python, также требуется владение SQL и такими инструментами, как Apache SPARK, Hadoop и подобными.

> Любой технически грамотный человек может посчитать среднее, но аналитик должен понять, что скрывается за этим средним

2 Для каждого проекта у нас есть специально отведенные рабочие кабинеты в вебе, для того чтобы упростить рутинные задачи. Но, как писал выше, часто инструменты – это скриптовые языки программирования, регулярные выражения, фреймворки для обработки большого количества данных и т.д.

3 Высшее техническое или математическое образование – наш приоритет. Но у нас успешно трудятся не только люди с техническим образованием, но и филологи, лингвисты и даже искусствоведы. Важно, чтобы у человека был аналитический склад ума.

4 Очень желателен опыт работы в сходных должностях – аналитик поиска, антиспам-аналитик и подобные должности, которые предполагают работу с большим количеством данных.

5 Я бы тут выделил психологическую устойчивость – веб-контент-аналитики создают решения для родительского контроля в интернете. И аналитикам подчас приходится защищать детей от контента, который непросто обрабатывать, – это порно (включая извращения), убийства, насилие, сцены казней и т.д.

Андрей Порожнетов,
аналитик Lingualeo

1 На мой взгляд, веб-аналитик должен:

> Понимать, как работает веб в целом: что такое референ, зачем нужны куки и т.п. Уметь пользоваться отладчиком браузера, иметь представление об HTML и JS.

> Знать, как работает бизнес и продукт.

> Уметь визуализировать результат своей работы в простом, понятном и убедительном виде. В идеале неплохо бы владеть каким-то инструментом визуализации данных (qlikview, tableau и т.п.).

> Не испытывать проблем со статистикой и математикой. Расчет статистической значимости нужен почти в каждой задаче.

> Быть в курсе про Data engineering. Не обязательно, но хорошо, если аналитик понимает, как работают базы данных, и может сам загрузить в БД данные из любого источника.
> Уметь использовать алгоритмы Machine Learning. Бывают задачи, где это необходимо, но у нас таких пока очень немного.

Кроме этих скилов, есть важные качества, которыми должен обладать аналитик, например критичное отношение к любому экспертному мнению, стремление искать во всем причину и желание разбираться, в чем исходный смысл той или иной задачи.

В своей работе веб-контент-аналитик сталкивается с обработкой большого количества данных, поэтому требуется использование скриптовых языков

2 Наш основной инструмент – это SQL. Все остальное вспомогательно. Периодически смотрим, что интересного появляется в таких сервисах, как Amplitude, AppSee и т.п., но полноценно их не используем.

3 Уровень образования, а иногда и опыт не имеют особого значения для нас. Мы делаем ставку на тестовое задание и на то, как себя человек проявляет на собеседовании.

4 Я стараюсь не иметь предрассудков относительно образования или опыта. Если человек проходит собеседование, значит, его опыт устраивает. Главное, чтобы кандидат разделял ценности компании и демонстрировал, что может и хочет решать те задачи, которые у нас есть.

5 Будет плюсом, если кандидат – активный пользователь образовательных сервисов и мобильных приложений и имеет интерес к тому, как развивается этот рынок.

Роман Прокашев,
специалист отдела интернет-маркетинга в JetBrains

1 Наиболее важные навыки – это умение работать с большим объемом данных и умение сделать из данных вывод, который будет ценным для компании и поможет принять правильное решение. При этом важно видеть и не очевидные выводы, а также уметь извлечь пользу из данных, из которых вообще никакого вывода на первый взгляд нет. Аналитик должен, с одной стороны, быть внимателен к деталям и в то же время быть в курсе новостей компании – релизов продуктов, акций, специальных предложений, – чтобы видеть зависимость между этими событиями, объемом трафика и поведением пользователей. Аналитик должен понимать механику всех основных платных и бесплатных каналов, по которым приходит трафик. Проанализированные им данные должны помогать выработать общую маркетинговую стратегию компании. Следует знать и инструменты, которые используются другими командами, – интеграция веб-аналитики с ними может дать дополнительные возможности. Важно, чтобы аналитик мог не только откликнуться на запрос в данных от других команд, но и мог сам определить, какими данными стоит поделиться. Для этого нужно постоянно смотреть на данные под разным углом, думать о том, могут ли они быть полезны.

2 Основным инструментом является Google Analytics или подобные системы. Важно уметь пользоваться системой управления тегами (например, Google Tag Manager) – она позволяет наиболее гибко и оперативно собирать данные, при этом минимально зависеть от команды разработчиков сайта. Чтобы максимально использовать функционал систем вроде GTM, желательно знать основы HTML, CSS и JavaScript. Дальнейшие инструменты зависят от того, как много данных из разных источников необходимо собрать вместе: если нужны данные из внутренней CRM в Google Analytics или наоборот, то обычно используются API, импорт данных, дополнительные третьи инструменты. Использование простых sql-запросов может пригодиться, если есть необходимость вычленять отдельные данные. Сегодня все больше используются инструменты визуализации данных, но практической пользы от них едва ли больше, чем от самых обычных офисных пакетов. Важно также уметь грамотно формулировать и вести свои задачи в issue трекере, чтобы они были понятны коллегам и легко находились.

3 Строгих требований к образованию нет – работа идет на стыке статистики, маркетинга и основ веб-разработки. В университетах этому направлению не учат, а курсы, как правило, не дают необходимых знаний, особенно для работы на международных рынках. Поэтому в отрасли можно встретить людей с самым разным бэкграундом. Решение практических задач важнее дипломов, хоть веб-аналитика без высшего образования встретишь едва ли.

4 Требования к опыту зависят от сложности задач, с которыми будет работать сотрудник. Для более сложных опыт важен, но не критичен. Можно сказать, что опыт ценен только в связке с такими качествами, как желание, обучаемость, умение разобраться и найти необходимую информацию. Важно и уметь пользоваться своим опытом – переносить и проецировать его на другие задачи, не копирующие друг друга один в один. А для того чтобы начать работать с простыми задачами, хватит мотивации, логического мышления и способности искать информацию в интернете.

5 У нас клиенты почти во всех странах мира, поэтому наш рынок – глобальный. Обязательно знание английского, кроме того, важно понимать специфику продуктов – кто является целевой аудиторией, как люди пользуются продуктом и какие задачи решают. Наши клиенты – это разработчики, и их отличает преимущественно рациональное поведение, в том числе и в использовании сайтов компании. Другой важной особенностью является то, что доля мобильного трафика от общего у нас весьма мала – здесь мы идем немного вразрез с общемировой mobile-first-тенденцией. EOF

Подготовил Игорь Штомпель

РОССИЙСКАЯ НЕДЕЛЯ
ВЫСОКИХ ТЕХНОЛОГИЙ

ТПП РФ

ufi Approved Event

Expo Rating

СВЯЗЬ

Информационные и коммуникационные технологии

25—28 апреля 2017

В НОВЫЕ СРОКИ

29-я международная выставка

Организатор:

ЭКСПОЦЕНТР
МОСКВА

При поддержке:

- Государственной Думы Федерального Собрания РФ
- Министерства связи и массовых коммуникаций РФ
- Министерства промышленности и торговли РФ
- Федерального агентства связи (Россвязь)
- Российской ассоциации электронных коммуникаций (РАЭК)

Под патронатом Торгово-промышленной палаты РФ

Реклама 12+

Россия, Москва, ЦВК «Экспоцентр»

www.sviaz-expo.ru

Визитка

ВЛАДИМИР ГАКОВ, писатель, специалист по научной фантастике, журналист, лектор. Окончил физфак МГУ. Работал в НИИ. С 1984 г. на творческой работе. В 1990-1991 гг. – Associate Professor, Central Michigan University. С 2003 г. читает курс по истории бизнеса в Институте бизнеса и делового администрирования (ИБДА) Российской академии народного хозяйства и государственной службы (РАНХиГС). Автор 8 книг и более 2000 публикаций

Китайская сказка «1001 ночи»

Один из богатейших предпринимателей Китая Джек Ма вспоминает, как в 1999 году, попивая кофе в одной малайзийской кофейне, почему-то вспомнил читанную в детстве сказку о сокровищах Али-Бабы

Спросив подошедшую официантку, знает ли она, кто это и с чем у нее ассоциируется это имя, Ма услышал в ответ: «Ну, как же – сезам, откройся!» Так же ответили и десятки случайных прохожих на улице. В том же году 33-летний бизнесмен основал на родине торговый веб-портал Alibaba Online. Тогда еще, вероятно, даже не мечтая о том, что его сетевой «супермаркет» станет крупнейшим в мире. Всего-то спустя полтора десятка лет.

Маркиз-триллионер

Сегодня уже никого не удивишь строкой из советской песни, ныне относимой, увы, к китайцам: они «рождены, чтоб сказку сделать былью». Во всяком случае, одну из таких воплощенных в быль сказок «сочинил» за считанные годы Джек Ма. В ноябре 2012-го (когда объемы продаж на двух главных торговых площадках Alibaba преодолели символический рубеж в 1 трлн юаней – тогда это равнялось примерно $170 млрд) заслуживший от соотечественников цветистый, как это принято у китайцев, титул «маркиз триллионов» (Trillion Houjuie, а не Trillion Hou, как ошибочно написано в большинстве англоязычных источников).

Один из богатейших предпринимателей Китая Джек Ма

Сегодня интернет-гигант Alibaba Group со штаб-квартирой в Ханчжоу, владелец одноименного веб-портала (и еще кучи всего, перечисленного ниже), обеспечивает онлайн-торговлю по всему миру – а конкретно в 230 странах. По трем главным направлениям электронной торговли: C2C, B2C и B2B, то есть соответственно потребитель – потребитель, бизнес – потребитель и бизнес – бизнес. Три года назад капитализация только что акционированного «китайского Али-бабы» за первый же день торгов на Нью-Йоркской фондовой бирже превысила рекордные $230 млрд – ничего подобного Уолл-стрит до той поры не видывала. Сейчас, ввиду общего спада на рынке, компания стоит чуть меньше, но все равно, как говаривал некогда популярный телеперсонаж: «Внушаеть!»

И произошел этот невиданный взлет – даже по китайским масштабам (в этой удивительной стране всего не много, а очень много!) – повторюсь, за неполные два десятка лет.

Юнь Ма (именно так звали будущего основателя Alibaba) родился 10 сентября 1964 года в Ханчжоу. Он с детства страстно желал выучить английский, практикуясь на иностранцах: на протяжении почти десятка лет Ма регулярно поджидал их у отеля в 160 км от родного дома и устраивал гостям бесплатные экскурсии по городу. Один из англичан (или американцев) после безуспешных попыток правильно произнести имя добровольного «экскурсовода», стал называть его Джеком. Со временем это стало вторым – и основным – именем будущего «маркиза-триллионера».

Четыре года новоокрещенный Джек Ма безуспешно пытался поступить в колледж, пока не прошел по конкурсу в городской педагогический институт, который закончил с дипломом учителя английского. После чего еще несколько лет пытался найти себе работу – с тем же успехом, что и с поисками вуза. Его никуда не брали – даже в полицию и в первое

открывшееся в Ханчжоу заведение американского фастфуда KFC… И еще Джек Ма десять раз подавал документы в американский Гарвардский университет и регулярно получал их обратно с неизменным отказом.

Об интернете будущий «сетевой магнат» впервые услыхал в 1994 году. И спустя год с помощью друга впервые побывал в США, где впервые же сам «побродил» по Всемирной сети. Первым словом, набранным китайцем-англофилом в поисковике, было слово «пиво». К удивлению Ма, среди лавины информации о любимом напитке, обрушившейся на неофита, не было почти ничего о китайском пиве. Более того, начинающий юзер не смог выудить из тогдашнего интернета ничего существенного и о своей родине – Китае! «Так жить нельзя», – решил Ма и с помощью более продвинутых друзей-пользователей создал свой собственный (как он сам признавался – откровенно «кривой») веб-сайт, посвященный исключительно Китаю. И в первые же часы получил множество электронных писем от неизвестных ему соотечественников, желающих познакомиться и проконсультироваться по каким-то конкретным вопросам.


Штаб-квартира Alibaba

Так Джек Ма открыл для себя поле будущей деятельности. В апреле того же 1995 года вместе с женой и другом с помощью того же интернета он собрал $20 тыс. и основал свою первую компанию, назвав ее China Yellow Pages (по аналогии с популярными телефонными справочниками). За первые три года компания принесла основателям $800 тыс. Любопытно, что сам Джек Ма, по его собственному утверждению, за всю жизнь не написал ни единой строчки программы и вообще купил себе первый компьютер для личного пользования в возрасте тех же 33 лет. Когда создал Alibaba.

Сезам открылся

До этого, вернувшись на родину, Ма в течение двух лет возглавлял ИТ-компанию, созданную при министерстве внешней торговли и экономического содружества. А в 1999-м уволился, вернулся в родной Ханчжоу и вместе с 18 друзьями и единомышленниками основал компанию Alibaba с уставным капиталом $80 тыс.

За первый же год работы компании и одноименного веб-портала Ма смог привлечь инвестиций на $50 млн от таких серьезных инвесторов, как, к примеру, Goldman Sachs. Можно сказать, что именно Alibaba стала своего рода стартером к запуску национальной системы электронной торговли как для розничных покупателей, так и для малого и среднего бизнеса. Впрочем, глава компании недолго ограничивал свои амбиции национальными интересами – в 2003 году он основал сразу несколько компаний (Taobao Marketplace, Alipay, Ali Mama, Lynx) в целях выхода на международный уровень.

Успех последних был столь очевиден, что на приобретение торговой веб-площадки Taobao нацелилась сама eBay, сделавшая владельцу Alibaba «предложение, от которого нельзя отказаться»! Однако Ма не стал продавать свое детище. Более того, он получил от другого супертяжеловеса на IT-рынке – совладельца Yahoo Джерри Янга (как китайцу – пусть и с Тайваня – да не поддержать соотечественника с материка!) – $1 млрд дополнительных инвестиций. Этот прилетевший в компанию миллиард был потрачен на создание китайской «дочки» транснационального гиганта Yahoo! Inc. (конкретно компания Джерри Янга и Дэвида Файлоу приобрела 43% акций компании Джека Ма, 20% из которых последний выкупил обратно в 2012 году).

За четыре года, с 2008-го по 2012-й, Alibaba (превратившаяся теперь уже в холдинг Alibaba Group) реорганизовала свое подразделение Taobao, выделив из него самостоятельные компании – розничной торговли Taobao Mall (Tmall.com) и торговый поисковик eTao. И создала собственное исследовательское подразделение – Alibaba Group R&D Institute, подразделение Alibaba Cloud Computing (к 2011 году разработавшее собственную операционную систему для смартфонов – Aliyun OS), а также специальный фонд исследований по охране окружающей среды. В том же 2011 году число зарегистрированных пользователей портала Alibaba увеличилось до 65 млн человек из двух сотен с лишним стран мира. Годом позже к уже освоенным национальным рынкам прибавился и российский, на который Alibaba пришла сначала в партнерстве с платежной системой Qiwi, а затем подключилась к системе WebMoney Transfer и наконец – через Яндекс.Деньги – к терминалам Сбербанка.

Но настоящий прорыв к сказочным сокровищам в пещерах случился в 2014 году. В марте детище Джека Ма объявило о своей готовности выйти на рынок с IPO (первичным публичным предложением акций). Вообще-то все крупные компании рано или поздно, но акционируются, этим трудно кого-либо удивить, но Alibaba, следуя традиции «делать сказку былью», удивила на сей раз саму Уолл-стрит.

Точнее, Нью-Йоркскую фондовую биржу, которая, казалось бы, успела привыкнуть ко всему. Потому что 19 сентября 2014 года в самом знаменитом биржевом здании мира случилась сенсация. Первичное публичное предложение акций Alibaba Group (по цене $68 каждая) принесло компании $21,8 млрд. К моменту закрытия торгов капитализация китайской компании равнялась $231 млрд. 22 сентября Alibaba выбросила на рынок еще 15% от ранее запланированного

количества акций, увеличив сумму привлеченного капитала до $25 млрд – абсолютный рекорд в истории американской финансовой системы!

Председатель Ма – за капитализм без акул

Пост CEO Alibaba Group основатель компании занимал до 2013 года, после чего уступил свой пост Дэниэлу Чжану, оставшись председателем Совета директоров. Ма по-прежнему «рулит» своей компанией (президентом которой, кстати, избрали вообще не китайца – канадского бизнесмена, а по совокупности еще и байдарочника – олимпийского чемпиона Майкла Эванса!), но больше времени уделяет общественной деятельности (а еще занятиям тай чи, одного из восточных боевых единоборств). Выполнять различные представительские функции Джеку Ма позволяет его статус – одного из самых известных и влиятельных китайцев в мире.

Французский замок Джека Ма

Еще в 2005 году Всемирный экономический форум наградил его почетным титулом «Молодой глобальный лидер», влиятельный американский журнал Fortune включил китайского бизнесмена в список «25 самых крупных азиатских бизнес-персон», а журнал Time – в сотню самых влиятельных людей в мире. Аналогичных признаний китайский «сетевой магнат» добился от практически всех авторитетных представителей мировой (не говоря уж о родной китайской) бизнес-периодики – Businessweek, Harvard Business Review, Forbes и других.

Его приглашают выступить в самых знаменитых и престижных университетах Америки, Азии и Европы (в последней, кстати, Джек Ма владеет сразу несколькими замками и виноградниками в провинции Бордо). А гонконгский Университет науки и технологии в 2013 году присудил Ма почетную докторскую степень. Куда более знаменитому Гарварду, ранее десятикратно «футболившему» настойчивого абитуриента, очевидно, остается только кусать локти с досады…

Кроме того, Джек Ма заседает в правлениях японского банка и крупного китайского медиахолдинга. А также в совете попечителей Всемирного фонда сохранения дикой природы (World Wildlife Fund), эмблемой которого, как известно, служит охраняемый представитель китайской фауны и национальный символ страны – панда.

В совете попечителей этой авторитетной международной организации Джек Ма, кстати, единственный китаец. Хотя избран туда не ради представительства от родины панд, а за признанные заслуги в деле охраны дикой природы. Дело в том, что еще в 2007 году компания Alibaba подверглась критике со стороны экологов за… Как вы думаете – за что? За то, что на соответствующих торговых площадках среди прочего активно торговали одним из самых ходовых в Китае товаров… Акульими плавниками, из которых готовят столь почитаемый в национальной кухне суп. А поскольку китайцев на свете очень много и все любят полакомиться вкусным супчиком, то идущих на его изготовление акул в Мировом океане становится все меньше. И всевозможные природоохранительные организации забили тревогу – страшных морских хищников тоже нужно беречь!

Неизвестно, какие душевные муки испытал китайский бизнесмен и китайский гурман Джек Ма, принимая непростое для себя и бизнеса решение. Но китайский любитель дикой природы Джек Ма в январе 2009 года официально заявил, что отныне он вместе с семьей навсегда исключает из домашнего рациона акульи плавники. А на всех торговых площадках его компании последние попадают в раздел запрещенных к продаже товаров.

Кроме того, годом позже руководство Alibaba официально объявило, что отныне 0,3% годовой прибыли будет отводиться на различные экологические проекты. А сам глава компании в позапрошлом году лично потратился на аналогичную инициативу – за $23 млн купил два крупных природоохранных участка земли в американских Национальных парках (заповедниках) – Адирондак и Брэндон в штате Нью-Йорк.

Короче, не только Восток – дело тонкое, но и восточный (а особенно китайский!) бизнес. Чему свидетельством еще одно важное – и неординарное – заявление, сделанное Ма на годовом собрании акционеров в том же 2009-м. Глава Alibaba призвал их кардинально пересмотреть собственные планы создания и ведения бизнеса. А именно: создавать новые компании, заранее приспосабливая их к нынешнему затяжному экономическому кризису, вместо того, чтобы по обыкновению рассчитывать на помощь со стороны властей или другого, крепко стоящего на ногах бизнеса. А значит, искать какие-то нетривиальные, непривычные ходы. Джек Ма напомнил присутствующим, что все великие прорывы в бизнесе совершались теми, кто видел нечто, недоступное зрению остальных. И, соответственно, в нынешней экономической ситуации успех ждет таких же «инакосмотрящих».

Карта пещер с сокровищами

Между тем первое, основное и самое любимое детище Джека Ма – его Alibaba Group – так же бурно развивается и расширяется, как и в первые годы существования.

Только за последние два года китайский суперхолдинг пополнился солидным пакетом (на полмиллиарда долларов) акций местной компании – производителя смартфонов Meizu. И пакетом акций американской компании Groupon, специализирующейся на посредничестве между локальным бизнесом и покупателями. И контрольным пакетом (также на полмиллиарда долларов) немецкой компании Lazada Group, чьи азиатские «дочки» получили название «Amazon.com Юго-Восточной Азии». А также – до кучи – англоязычной гонконгской газетой South China Morning Post и другими активами одноименной медиагруппы SCMP (еще четверть миллиарда).

И даже собственным автопроизводством! Первый «интернет-автомобиль» Roewe RX5, созданный в содружестве с Шанхайской автопромышленной корпорацией (SAIC), был представлен публике в прошлом году. На презентации Джек Ма заявил, что, по его прогнозам, «80% автомобиля будущего вообще никак не будет связано с транспортными функциями, скорее машина превратится в очередного домашнего робота, с которым владелец будет общаться на протяжении целого дня».

Дом Джека Ма в Гонконге

На сегодняшний день обширное и разветвленное «хозяйство» Джека Ма (создавшее, по его собственным подсчетам, более 30 млн рабочих мест в одном Китае) состоит из следующих самостоятельных подразделений:

> **Alibaba.com** – крупнейшая онлайновая торговая площадка для организаций (B2B), включающая три главных сервиса: одноименный англоязычный портал, обслуживающий экспортеров и импортеров в более чем двух сотнях стран и регионов; портал на китайском языке 1688.com для аналогичного обслуживания местного B2B-бизнеса; веб-сайт AliExpress.com, обеспечивающий торговлю в розницу и мелким оптом.

> **AutoNavi** – служба автонавигации и картографии.

> **Taobao Marketplace (или просто Taobao)** – крупнейшая в Китае онлайновая платформа для торговли C2C (потребитель – потребитель), интернет-магазин и интернет-аукцион (аналогичный eBay.com). К началу позапрошлого года – второй по посещаемости веб-сайт в Китае, входит также в первую двадцатку самых посещаемых сайтов в мире.

> **Tmall.com** – онлайновый магазин розничных продаж.

> **Juhuasuan.com** – еще один торговый портал на китайском языке, специализирующийся на flash sales, т.е. «горячих продажах» (критики называют это системой искусственно созданного дефицита), когда предлагаемые товары продаются со скидкой, но время продаж строго ограничено (от суток до месяца).

> **eTao.com** – дополнительная торговая площадка, позволяющая покупателю сравнивать цены на тот же товар на разных сайтах, а также делиться результатами своих ценовых изысканий с другими покупателями.

> **Alipay** – платформа для электронных платежей онлайн (обеспечивает почти половину всех онлайн-платежей в Китае).

> **Alibaba Cloud (Aliyun)** – платформа различных сервисов на базе облачных вычислений.

> **China Yahoo!** – интернет-портал на китайском языке для поиска и чтения новостей, пересылки электронной почты и всего того, что обеспечивает глобальная служба Yahoo!

> **Aliwangwang** – служба мгновенного обмена сообщениями (instant messaging) между покупателями и онлайн-продавцами. С этой целью начиная с 2013 года Alibaba Group разработала собственное приложение – Laiwang.

> **Alibaba Pictures** – кинокомпания.

> **Youku Tudou** – интернет-ТВ-компания.

> **Alibaba Group R&D Institute** – научно-исследовательское подразделение, созданное в 2008-м и за первый год работы принесшее Alibaba Group около 350 патентов и оригинальных приложений.

> **Xiami** – музыкальный онлайн-магазин.

> **365fanyi.com** – платформа для краудсорсинг-перевода (особенно актуального для китайского малого и среднего бизнеса, решившего торговать «с заграницей»).

> **South China Morning Post** – англоязычная газета, издающаяся в Гонконге.

> **Ali Health** – медицинский онлайн-магазин.

> **UCWeb** – ведущий в стране китайский интернет-провайдер.

> **Lazada Group** – «дочка» одноименной немецкой компании, обосновавшаяся в Сингапуре и имеющая отделения в Индонезии, Малайзии, Таиланде, Вьетнаме и на Филиппинах.

Как и всего, повторюсь, в Китае – не много, а очень много… Корпоративных подразделений и служб в частности. Но Джека Ма и его команду это, судя по всему, не пугает. Как-то справляются они со своими несметными сокровищами в пещерах!

А как иначе – покупателей в их стране не счесть, и всех нужно обслужить вовремя и аккуратно. На Востоке к сказкам вообще относятся очень серьезно: если последние не сделать былью – не поймут. EOF

НАУКА И ТЕХНОЛОГИИ

Николаев П.Л., *преподаватель, Московский авиационный институт (национальный исследовательский университет), г. Москва, npavel89@gmail.com*
Хорошко Л.Л., *к.т.н., доцент, Московский авиационный институт (национальный исследовательский университет), г. Москва, khoroshko@mati.ru*

Представление кодированными деревьями сценариев системы управления интеллектуального здания

В работе предложена модель представления сценариев (управляющих алгоритмов), осуществляющих работу системы управления интеллектуального здания. На основе предлагаемой модели в дальнейшем планируется программная реализация среды, предназначенной для визуального программирования алгоритмов функционирования систем умного дома. В рамках исследования детально разобраны определение и структура сценариев на основе примера. Также рассмотрен способ задания сценариев в виде деревьев, закодированных кодом Прюфера

Введение

Одним из важных элементов любой управляющей системы умного дома является программирование сценариев, обеспечивающих его функционирование. В системах управления интеллектуальными зданиями сценариями называются алгоритмы работы как отдельных подсистем, так и всей управляющей системы здания (дома или квартиры). Например, могут существовать отдельные сценарии управления освещением, тепловым режимом как для определенного помещения, сразу для нескольких помещений, так и для всего здания или дома.

В современных домах применяется все больше интеллектуальных устройств, но из-за разнообразия программирование их всех становится трудной задачей **[1]**.

Среднестатистическому пользователю, не умеющему программировать либо не обладающему всеми необходимыми знаниями и навыками, необходим такой инструмент, который позволил бы довольно просто и быстро создавать и редактировать различные сценарии.

Наилучшим способом для этого является визуальное программирование. Под этим термином подразумевается способ образного графического представления (используются графические элементы и фигуры) разрабатываемого алгоритма, который более естественен для восприятия человека **[2, 3]**.

Визуальное программирование позволяет избежать множества ошибок и значительно сократить время реализации алгоритмов. Применительно к интеллектуальным зданиям это позволит уменьшить возможные отказы систем из-за каких-либо ошибок, которые можно допустить при классическом программировании.

Авторами работы планируется программная реализация системы визуального программирования сценариев, в которой программирование будет осуществляться путем правильной расстановки графических блоков.

В данном исследовании мы рассмотрим способ представления сценариев, создаваемых в такой системе визуального программирования.

Структура сценариев

С точки зрения формального строения сценарий – это набор действий, выполняемых при наступлении определенных событий. Приведем пример сценария по управлению тепловым режимом в помещении.

При уменьшении температуры в помещении до определенного значения (допустим, T1 <= 15) и присутствии в комнате человека (срабатывает датчик присутствия, S1 == истина) необходимо выполнить несколько действий:

1. Для начала нужно проверить состояния окна (A1) и кондиционера (A2). Если окно открыто, о чем свидетельствует состояние геркона (S2 == истина) или включен кондиционер (A2 == истина), то необходимо выполнить еще ряд действий.
   - Проверить состояние штор, и если они закрыты (A3 == ложь), то открыть их (A3 = истина), иначе невозможно будет закрыть окно, а затем уже закрыть окно (A1 = ложь).
   - Выключить кондиционер (A2 = ложь).
2. В ситуации, когда окно закрыто и кондиционер выключен, следует проверить состояние обогревателя (A4), и если он выключен, то включить его (A4 = истина).

На псевдокоде данный сценарий можно записать так:

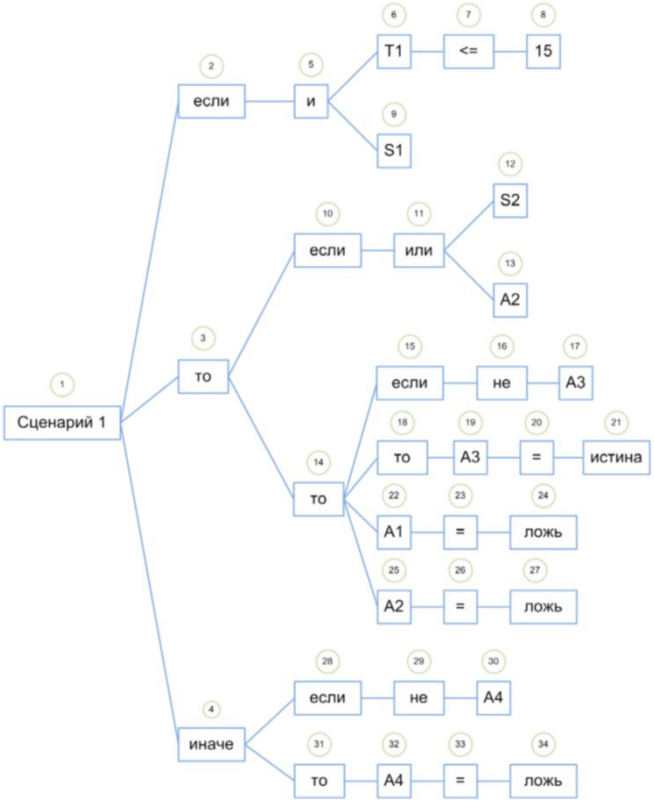
```
сценарий 1
    если (T1<= 15 и S1) то
        если (S2 или A2) то
                если (не A3) то A3 = истина;
                A1 = ложь;
                A2 = ложь;
    иначе
        если (не A4) то A4 = истина;
```

По сути, любой сценарий может записать с помощью языковой условной конструкции «если-то-иначе». Помимо условной конструкции сценарии должны состоять из набора операций, переменных и отношений между ними. В итоге получим следующие возможные составляющие сценария:

- условная конструкция «если-то-иначе»;
- переменные (логические, числовые и текстовые), отображающие состояния устройств и передаваемые ими данные;
- логические операторы (и, или, не, исключающее или);
- операторы сравнения (больше, меньше, больше/меньше или равно, равно, не равно);
- операторы циклов;
- специальные функции (таймер, вывод текстового сообщения, вывод изображения).

В условной конструкции в «если» задается условие выполнения сценария – события, при котором задаваемый сценарий должен активироваться. В «то» задаются действия, которые должны быть запущены при выполнении условия. А в «иначе» (не является обязательным) задаются действия, которые должны быть запущены при невыполнении условия.

Рисунок 1. Представление сценария системы управления интеллектуального здания в виде дерева



Подробнее рассмотрим возможные типы переменных. В качестве логических определим переменные, отображающие состояния устройств, которые могут иметь только два значения: «истина» (устройство находится в активном состоянии) и «ложь» (устройство находится в неактивном состоянии). Это могут быть различные датчики (датчики движения, датчики освещенности, герконы), кнопки и управляемые устройства (устройства освещения, розетки).

В качестве числовых определим переменные, предназначенные для отображения числовых данных, поступающих от измерительных датчиков, а также передаваемых на управляемые устройства. В первом случае это могут быть температурные датчики (передают значения температуры), датчики влажности, концентрации газа и т.п. А во втором случае – к примеру, диммеры, используемые для изменения яркости подключенных устройств освещения в зависимости от изменения электрической мощности.

В качестве текстовых определим переменные, отображающие состояния устройств и передаваемые ими данные в текстовом виде. Например, жалюзи могут передавать следующие данные: «открыты», «закрыты», «открываются», «закрываются».

Представление сценариев

Структурно условную конструкцию можно представить в виде дерева. При визуальном программировании блок будет являться одной из вершин дерева, а связанные блоки будут образовывать ребра дерева. Преимущества представления алгоритма в виде дерева:

- легко читаемое и компактное представление данных;
- уменьшение количества возможных ошибок ввиду наглядного представления данных;
- упрощение внесения изменений в сценарий ввиду того же наглядного представления данных и возможности более быстрого поиска необходимых компонентов;
- также можно отметить, что представленный в форме дерева сценарий в дальнейшем будет легче преобразовать в код, т.к. при генерации кода необходимо строить абстрактное синтаксическое дерево.

Для рассмотренного выше примера сценария построено дерево, представленное на рис. 1.

Порядок нумерации вершин совершенно неважен, будем присваивать номера автоматически в порядке добавления элементов.

Кодирование деревьев

Согласно **[4, 5]** существует несколько способов представления деревьев: представление с помощью матрицы смежности, представление с помощью списков смежности, представление с помощью списка ребер и кода Прюфера, уровневый код, десятичная кодировка. Наиболее оптимальным способом кодировки деревьев является код Прюфера **[4, 5]**. Рассмотрим принцип работы данного способа кодирования.

Дерево T, состоящее из множества вершин $\{v_1, v_2, ..., v_n\}$, где n – количество вершин, задается списком ребер (v_i, v_j), где v_i и v_j – связанные вершины, $i, j \in [1, n]$ $i \neq j$. Код Прюфера – это последовательность $P(T) = [a_1, a_2, \ldots, a_{(n-2)}]$, которая формируется по следующему алгоритму **[5, 6]**:

Повторять от 1 до n-2 раза (пока не останется одно ребро):

1. Найти висячую вершину (лист) с минимальным номером v_i.
2. Записать в код Прюфера вершину v_j, смежную с v_i.
3. Удалить вершину v_i и ребро (v_i, v_j) из дерева.

На основе вышеизложенного сформируем алгоритм создания сценария:

1. Сценарий записывается как множество ребер дерева $T = \{v_1, v_2, ..., v_n\}$.
2. Создается последовательность $B = \{b_1, b_2, ..., b_n\}$, куда записываются значения вершин дерева (блоков) в порядке, соответствующем номерам вершин.
3. Дерево кодируется кодом Прюфера, который записывается в последовательность $P(T)$.
4. Итоговыми выходными данными после завершения процесса создания сценария будут две последовательности: B со значениями блоков, из которых состоит сценарий, и $P(T)$, где записан код Прюфера для этого сценария.

Список ребер дерева для сценария из нашего примера:

- (1, 2)
- (2, 5)
- (3, 1)
- (4, 1)
- (5, 6)
- (6, 7)
- (7, 8)
- (5, 9)
- (3, 10)
- (10, 11)
- (11, 12)
- (11, 13)
- (3, 14)
- (14, 15)
- (14, 18)
- (15, 16)
- (16, 17)
- (18, 19)
- (19, 20)
- (20, 21)
- (14, 22)
- (22, 23)
- (23, 24)
- (14, 25)
- (25, 26)
- (26, 27)
- (4, 28)
- (28, 29)
- (29, 30)
- (4, 31)
- (31, 32)
- (32, 33)
- (33, 34)

Список значений вершин:

- сценарий 1
- если
- то
- иначе
- и
- T1
- <=
- 15
- S1
- если
- или
- S2
- A2
- то
- если
- не
- A3
- то
- A3
- =
- истина
- A1
- =
- ложь
- A2
- =
- ложь
- если
- не
- A4
- то
- A4
- =
- ложь

Для нашего сценария код Прюфера будет следующим

- 7
- 6
- 5
- 5
- 2
- 1
- 11
- 10
- 3
- 16
- 15
- 14
- 20
- 19
- 18
- 14
- 23
- 22
- 14
- 26
- 25
- 14
- 3
- 1
- 4
- 29
- 28
- 4
- 32
- 32
- 33

Далее при помощи специального парсера можно раскодировать дерево и сгенерировать код для интеграции сценария на том языке программирования, на котором реализовано программное обеспечение системы управления интеллектуального здания. Также дерево можно раскодировать для редактирования сценария.

• • •

На основе предложенного в статье способа представления сценариев, обеспечивающих функционирование систем управления интеллектуальными зданиями, планируется разработать среду визуального программирования сценариев. Визуальная среда разработки будет являться частью специализированного облачного сервиса. Данный сервис включает в себя серверную и клиентскую части, что предполагает передачу данных между ними по сети Интернет. Представление сценариев в виде деревьев, закодированных кодом Прюфера, позволит достичь нескольких результатов.

Во-первых, передача данных об алгоритме и его хранение в закодированном виде в базе данных позволят сократить объем передаваемых и хранимых данных. Кодировка деревьев относится к задаче сжатия информации, а компактная запись дерева, полностью описывающая его структуру, может существенно упростить передачу информации о дереве и работу с ним **[5]**.

Во-вторых, кодирование дерева позволит повысить безопасность передаваемых данных. В случае перехвата злоумышленниками передаваемых данных о сценариях с целью повреждения работоспособности систем умного дома или с целью проникновения в сеть здания у них могут возникнуть проблемы с расшифровкой кодированных сценариев из-за неосведомленности об используемых способах представления данных. EOF

[1] Serna M., Sreenan C., Fedor S. A visual programming framework for wireless sensor networks in smart home applications. // Proceedings of the 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 7-9 April 2015. – P. 1-6.

[2] Visual Programming Language (VPL) [Электронный ресурс]. – URL: https://www.techopedia.com/definition/22855/visual-programming-language-vpl.

[3] Коварцев А.Н. Методы и средства визуального параллельного программирования. Автоматизация программирования: учеб. / А.Н. Коварцев, В.В. Жидченко. – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2011. – 168 с.

[4] Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003. – 1104 с.

[5] Кирсанов М. Н. Графы в Maple. Задачи, алгоритмы, программы. – М.: Издательство ФИЗМАТЛИТ, 2007. – 168 с.

Ключевые слова: умный дом, интеллектуальное здание, визуальное программирование, кодирование деревьев, код Прюфера.

Representation of scenarios of intelligent building management system by coded trees

Nikolaev P.L., Lecturer of Moscow Aviation Institute (National Research University), Moscow, npavel89@gmail.com
Khoroshko L.L., Ph.D, associate professor of Moscow Aviation Institute (National Research University), Moscow, khoroshko@mati.ru

Abstract: *In this paper the model of representation scenarios (control algorithms) which are implementing the work of intelligent building management system is offered. On the basis of the proposed model a software implementation of the environment for visual programming of algorithms for the operation of smart home systems is planned further. In this research the definition and structure of the scenarios based on the example are analyzed in detail. Also the way of representation scenarios in the form of trees coded by Prufer code is considered.*

Keywords: *smart home, intelligent building, visual programming, tree coding, Prufer code.*

Богораз А.Г., *кафедра безопасности информационных технологий, Институт компьютерных технологий и информационной безопасности, Южный федеральный университет, Ростовская область, г. Таганрог, bogoraz.a.g@gmail.com*

Создание методов защиты
децентрализованных распределенных социальных сетей

Целью данной работы является определение методов, которые необходимо разработать для повышения уровня безопасности в децентрализованных распределенных социальных сетях (ДРСС). Описана базовая архитектура ДРСС и проблематика отличия способов защиты клиент-серверных сетей и ДРСС, также приведены типы атак, актуальные для ДРСС. Проведен анализ российской и зарубежной научной литературы по тематике. На основе анализа была сформулирована задача разработать методы регистрации нового пользователя, метод «Доверия», позволяющий пользователям «модерировать» других пользователей, и метод аутентификации в ДРСС

Введение

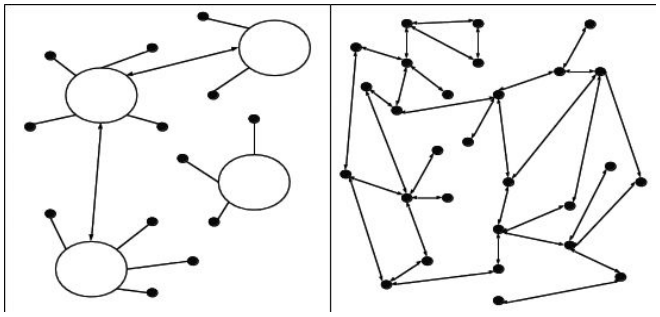
Социальные сети (СС) стали в наше время одними из самых популярных и быстро развивающихся сетевых сервисов. СС – это веб-ориентированная платформа, используемая для создания пользователем своей сетевой идентификации – так называемого профиля, а также для создания списка «Друзей» индивида, с которыми он хочет поддерживать отношения.

Большинство современных популярных СС используют клиент-серверную архитектуру, и во многом это определяет их достоинства и недостатки. Однако существуют также СС, основанные на архитектуре распределенного типа **[1, с. 101]**.

Архитектура

Распределенные социальные сети (РСС) – это сети, изначально созданные прежде всего для повышения уровня конфиденциальности пользовательских данных. Личные данные пользователя хранятся исключительно на его клиентском компьютере. Пользователь целиком и полностью определяет доступ к своим данным для любого другого пользователя РСС. В обычных СС сервер используется для абсолютно всех клиентских действий, таких как общение, хранение информации о других пользователях и прочее. В РСС сервер может либо использоваться для соединения пользователей между собой, либо вообще отсутствовать. Таким образом, можно выделить два типа такого вида СС – клиент-серверные и полностью децентрализованные.

Рисунок 1. Схематическое изображение клиент-серверных и полностью децентрализованных распределенных сетей



Клиент-серверные РСС работают аналогично обычным СС, но с серьезными отличиями. В РСС данного типа сервер используется только для соединения клиентов между собой. Преимуществом данного решения можно назвать отсутствие необходимости хранить информацию пользователей и отсутствие доступа к личным данным пользователя. К минусам можно отнести возможность взлома сервера и последующей прослушки пользовательских коммуникаций, а также отслеживание личных данных пользователей, передаваемых между общающимися через сервер пользователями.

Полностью децентрализованные РСС (ДРСС) отличаются от клиент-серверных сетей полным отсутствием сервера. Клиент такой сети устанавливается на компьютер пользователя и является как клиентом, так и сервером одновременно. Подобные сети называют еще p2p-сетями **[1, с. 106]**.

Проблематика отличия способов защиты клиент-серверных и децентрализованных распределенных социальных сетей

Основной и главной проблемой защиты ДРСС является относительная неприменимость стандартных способов защиты «классических» СС, построенных на основе архитектуры «клиент-сервер». К примеру, основными способами защиты архитектуры сети в «классических» СС являются:

- повышение защищенности серверного обеспечения;
- повышение защищенности серверного оборудования;
- использование нескольких, территориально удаленных друг от друга, серверов;
- повышение уровня защищенности специального приложения или веб-сервиса для взаимодействия с СС.

Как уже описывалось выше, в децентрализованных сетях отсутствует понятие сервера, поэтому в отличие от сетей с традиционной архитектурой отсутствует необходимость в его защите.

Серверное оборудование, с точки зрения прямого назначения, в ДРСС не используется, и, следовательно, отсутствует необходимость в его защите. Также нет необходимости в использовании территориально удаленных серверов, так как изначальная архитектура ДРСС подразумевает, что ее клиенты будут являться узлами одноранговой сети, в архитектуре которой заложена реализация единого отказоустойчивого сервиса.

Таким образом, единственным, полностью применимым и необходимым способом защиты является повышение уровня защищенности веб-сервиса или специального приложения для взаимодействия с ДРСС. Фактически разработка методов защиты для веб-сервиса или приложения является основной задачей при решении проблем безопасности в ДРСС.

Типы атак, актуальные для децентрализованных распределенных социальных сетей

Как и «классические» СС, ДРСС подвержены различным типам атак. В данной секции будут описаны типы атак, которые актуальны для ДРСС. В соответствии с **[2, 3, 4]** необходимо выделить следующие типы атак:

- Сетевые атаки: подслушивание трафика, Man-in-the-Middle, DoS и DDoS-атаки и другие.
- Атаки, связанные с архитектурой ДРСС, к примеру создание множества злоумышленных узлов или «наводнение» нелигитимными запросами.
- Атаки, связанные с криптографией, такие как подбор криптографических ключей, используемых для шифрования трафика в ДРСС или применение уязвимостей используемых протоколов шифрования.
- Социальная инженерия: использование поддельной личности, мошенничество, спам и т.д.
- Злоумышленное ПО: вирусы, троянские кони и т.д.
- Использование уязвимостей архитектуры, ПО и протоколов ДРСС.

Следует отметить, что данная классификация не является исчерпывающей и может изменяться в связи с появлением новых типов угроз.

Обзор российской научной литературы по тематике защиты децентрализованных распределенных социальных сетей

В российской научной литературе тематика защиты децентрализованных распределенных сетей (ДРС) представлена достаточно слабо. В то же время стоит отметить **[1]**, авторы которой предлагают идею разработки одноранговой ДРСС, которая предполагает наличие системы «приглашений» для новых пользователей, наличие доверенного «человека посередине», а также описывают различные аспекты безопасности, которые планируется учесть при создании такой ДРСС.

Авторы **[5]** и **[6]** анализируют наиболее известные РСС Diaspora и Friendica с точки зрения безопасности. В частности, достаточно подробно рассматривается алгоритм работы протокола Diaspora, который используется в одноименной РСС, а также протокола Zot, который используется в РСС Friendica. Они используются в этих РСС для создания безопасных коммуникационных каналов между пользователями этих сетей.

В **[7]** предлагается модифицировать программу Frost, которая работает в ДРС Freenet, с целью встраивания защиты от DoS и спам-атак. В качестве одного из методов защиты авторы предлагают модификацию реализации Сети Доверия, основная идея которой выражается в присваивании пользователями другу другу оценки качества сообщений, что позволяет фильтровать сообщения пользователей с оценкой, выше обозначенной планки. Предполагается, что в результате модификации некоторые пользователи, каким-либо образом подтверждающие, что они являются реальными людьми, смогут регистрироваться в виде агентов, которые смогут оценивать значимость каждого прочитанного сообщения, что позволит повысить общую производительность, в то время как в не модифицированной Сети Доверия каждый пользователь должен присваивать каждому сообщению свою отдельную оценку.

Стоит также отметить **[8]**, в которой предлагается использовать разработанный автором метод маршрутизации трафика в случае необходимости безопасной передачи данных между двумя точками в сети Интернет. Автор предполагает, что в случае периодической передачи данных между заданными точками злоумышленник может определить «оптимальный» маршрут, через который вероятность передачи данных максимально высока, и ожидать возможности для перехвата этих данных. В своей работе автор предлагает использовать множество доверенных серверов распределенной сети, которые могут использоваться как точки случайного маршрута, по которому будут переданы данные.

Обзор зарубежной литературы по тематике защиты децентрализованных распределенных социальных сетей

В зарубежной научной литературе преобладают статьи по анализу уязвимостей, которым подвержены ДРС, а также работы, в которых авторы рекомендуют использовать систему «репутационных» отношений между узлами ДРС, которая позволит узлам определять репутацию каждого отдельного узла в ДРС.

По тематике анализа уязвимостей в ДРС стоит отметить **[2]**, в которой приводится общий список сетевых атак, актуальных для «классических» сетей и ДРС, способы защиты от них, а также перечень атак, актуальных только для ДРС, и также показаны способы противодействия таким атакам.

Стоит отметить **[3]**, в которой представлен свой перечень типов атак, актуальных для ДРС, и описано исполнение фаз некоторых атак в тестовой симуляции ДРС, а также проведено исследование распространения злоумышленного ПО в ДРС.

По системам «доверительных» отношений стоит упомянуть **[9]**, авторы которой предлагают использовать разработанную ими систему PeerTrust, основная идея которой заключается в анализе поведения узлов ДРС с помощью оценки их «репутации» другими узлами. Каждый узел при взаимодействии с другим узлом выставляет ему плохую или хорошую оценку, которая хранится у него и на других узлах. Оценка может быть выставлена исключительно после взаимодействия. В случае, когда узлу A нужно начать взаимодействие с узлом B с неизвестной репутацией, он посылает в сеть запрос о репутации узла B и собирает данные, на основании которых принимает решение.

Также существует возможность использования системы «репутационных» отношений узлов, разработанной на основе теоремы Байеса. Автор **[10]** считает, что возможности любого узла, который выкладывает в ДРС файлы для скачивания, могут быть выражены в компонентах: качество файла, тип файла и скорость скачивания файла. Каждый узел ДРС строит свою сеть Байеса для каждого качества, типа и скорости передаваемого файла для каждого узла, с которым он взаимодействует. Узлы могут опрашивать друг друга для получения рекомендаций относительно других узлов, с которыми они не взаимодействовали.

Формализованная постановка задачи

В ходе анализа российской и зарубежной научной литературы, находящейся в открытом доступе, по тематике защиты ДРС и ДРСС было выявлено, что практически все «репутационные» системы, опубликованные в научной литературе, описывали системы отношений, основанные на узлах ДРС, которые самостоятельно будут определять репутацию каждого узла и принимать решения об использовании или неиспользовании этого узла при работе в ДРС. «Мнение» пользователя относительно «репутации» узла не учитывается или учитывается косвенно.

Системы, описанные в **[9, 10]**, предполагают, что узел может быть использован либо только как узел, с помощью которого можно передать данные далее по сети, либо как узел, обладающий неким файлом, который пользователь хочет скачать. Данные системы не могут быть применены в случае ДРСС, так как пользователи ДРСС проявляют социально активное взаимодействие, которое включает в себя не только передачу трафика и скачивание файлов, но и другую социальную активность.

Таким образом, было предпринято решение разработать такую систему «репутационных» отношений, которая позволит пользователям ДРСС определять «репутацию» каждого пользователя, а следовательно, узла, с помощью сертификатов «Доверия». Это позволит каждому пользователю ДРСС «модерировать» действия пользователей, которые являются его «Друзьями» в такой сети.

Планируется, что такая система «модерации» будет привязана к возможностям пользователя для взаимодействия с ДРСС. К примеру, для того, чтобы пригласить нового пользователя в ДРСС, уже существующий пользователь должен будет иметь определенный уровень «Доверия», который может быть обеспечен его «Друзьями»-пользователями, которые ему доверяют.

В ходе анализа литературы также не было обнаружено работ по методам безопасного приглашения и последующей аутентификации в ДРСС. Таким образом, было также принято решение разработать безопасные методы приглашения и аутентификации, которые будут взаимодействовать между собой и с вышеописанной системой «доверительных» отношений.

Таким образом, предполагается разработка следующих методов безопасного взаимодействия в ДРСС:

1) Метод регистрации нового пользователя в ДРСС должен позволять пользователю, который уже является членом ДРСС, «пригласить» нового пользователя в эту ДРСС таким методом, который позволит однозначно утверждать, что приглашаемый пользователь является именно тем пользователем, которого планировалось пригласить. При условии использования шифрования позволит защищать участников процесса приглашения от сетевых атак, например подслушивания трафика. Так как в методе планируется использование многослойного, с возможностью использования разных типов, шифрования, он также способен защитить от атак, связанных с криптографией, например подбор используемого ключа.

2) Метод «Доверия» должен представлять собой систему, которая позволит:

- проверить уровень «Доверия» пользователя, даже если он на момент проверки отсутствует в ДРСС;
- узнать, кто является «Доверителями» пользователя, которые присвоили ему такой уровень «Доверия», даже если эти «Доверители» отсутствуют в ДРСС на момент проверки;
- не допускать подделки пользователем своего уровня «Доверия»;
- не допускать подделки чужого уровня «Доверия» другими пользователями;
- регулировать возможности пользователя по взаимодействию с ДРСС в зависимости от его уровня «Доверия».

Так как метод фактически является способом формализации отношений пользователей, действующий по принципу «доверяю – подтверждаю свое доверие», следовательно, такая система позволит предотвратить атаки, связанные с архитектурой ДРСС, к примеру множественная регистрация новых узлов, с социальной инженерией, например использование поддельных личностей, и со злоумышленным ПО, которое может распространяться злоумышленниками.

3) Метод аутентификации предоставит возможность безопасной аутентификации пользователя в ДРСС при учете взаимодействия с вышеописанными методами. Должен обеспечивать защиту от атак, аналогичную методу регистрации нового пользователя.

Планируется, что данные методы могут быть применимы к любым ДРСС и позволять встраиваться в любую ДРСС либо в виде отдельных программных модулей, которые могут взаимодействовать между собой, либо в виде единого комплекса методов защиты.

Следует отметить, что методы, которые планируется разработать, не могут обеспечить абсолютную защиту от всех атак любого из вышеописанных типов. В то же время планируется разработать вышеописанные методы таким образом, чтобы обеспечить защиту от максимально возможного количества атак.

• • •

В заключение стоит отметить, что в данной работе присутствуют только формальные требования к разрабатываемым методам регистрации и аутентификации пользователей и системе «Доверия» в ДРСС и атаки, от которых они могут защитить, и в то же время отсутствует описание работы методов, характеристик и особенностей взаимодействия. К примеру, не описана ситуация, связанная с разрабатываемой системой «Доверия» пользователей, при которой уже зарегистрированный пользователь, имеющий высокий уровень «Доверия», может начать совершать злоумышленные действия, или, например, какие возможности на каком уровне «Доверия» доступны пользователю и т.д. Эти и другие вопросы планируется описать в дальнейших работах. EOF

[1] Богораз А.Г., Пескова О.Ю. Централизованные и распределенные социальные сети // Университет ИТМО, Технологии информационного общества в науке, образовании и культуре: сборник научных статей. Труды XVII Всероссийской объединенной конференции «Интернет и современное общество» (IMS-2014), Санкт-Петербург, 19-20 ноября 2014 г.

[2] Khalied Shredeh. Analysis of Attacks and Security Issues on the Peer-to-Peer Networks // International Journal of Computer Applications (0975 – 8887) Volume 138 – No.2, March 2016. URL: http://www.ijcaonline.org/research/volume138/number2/shredeh-2016-ijca-908728.pdf (дата обращения: 04.03.2017).

[3] J.Schäfer, K. Malinka, P. Hanáček. Peer-to-peer Networks: Security Analysis// International Journal On Advances in Security, vol 2 no 1, year 2009, pages 53-61. URL: https://www.iariajournals.org/security/sec_v2_n1_2009_paged.pdf (дата обращения: 04.03.2017).

[4] Анализ угроз сетевой безопасности // Your Private Network Лаборатория сетевой безопасности, Рубрика «Проблемы информационной безопасности сетей», дата публикации: Август 9th, 2009, 14:40. URL: http://ypn.ru/138/analysis-of-threats-to-network-security/2/ (дата обращения: 04.03.2017).

[5] Богораз А.Г., Власов А.С., Пескова О.Ю. Анализ безопасности распределенной социальной сети Diaspora // Материалы VII Международной студенческой электронной научной конференции «Студенческий научный форум». URL: http://www.scienceforum.ru/2015/908/12355 (дата обращения: 04.03.2017).

[6] Богораз А.Г., Власов А.С., Пескова О.Ю. Анализ безопасности распределенной социальной сети FRIENDICA // Материалы VII Международной студенческой электронной научной конференции «Студенческий научный форум». URL: http://www.scienceforum.ru/2015/908/14551 (дата обращения: 04.03.2017).

[7] Шумихин И.И., Моженков В.В. Исследование методов защиты децентрализованных систем от dos и спам-атак (на примере программы frost в сети freenet) // «Современные наукоемкие технологии», № 8-1, 2013 г. – С. 110-110. URL: http://www.top-technologies.ru/ru/article/view?id=32536 (дата обращения: 04.03.2017).

[8] Никонов В.И. Методы защиты информации в распределенных компьютерных сетях с помощью алгоритмов маршрутизации // Доклады ТУСУР, № 1-2 (21), 2010 г. URL: http://cyberleninka.ru/article/n/metody-zaschity-informatsii-v-raspredelennyh-kompyuternyh-setyah-s-pomoschyu-algoritmov-marshrutizatsii (дата обращения: 04.03.2017).

[9] L. Xiong, L. Liu. PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities // IEEE Transactions on Knowledge and Data Engineering, Volume 16 Issue 7, July 2004, pages 843-857. URL: http://www.it.iitb.ac.in/~madhumita/trust/xiong03peertrust.pdf (дата обращения: 04.03.2017).

[10] Yao Wang. Bayesian Network-Based Trust Model in Peer-to-Peer Networks // Second International Workshop, AP2PC 2003, Melbourne, Australia, July 14, 2003, Revised and Invited Papers, pages 23-34. URL: https://pdfs.semanticscholar.org/c42c/483ada58bd29ad6fb0829720ab37c1b2edcf.pdf (дата обращения: 04.03.2017).

Ключевые слова: аспределенные социальные сети, РСС; децентрализованные сети, ДС; социальные сети, СС; методы защиты.

Protection methods of the distributed decentralized social networks

Bogoraz A.G., Department of Information Technologies Security, Institute of Computer Technologies and Information Security, Southern Federal University, Rostov Region, Taganrog, bogoraz.a.g@gmail.com

Abstract: *The main aim of this article is to determine the methods to be developed to improve security in a Decentralized Distributed Social Networks (DDSN). The basic DDSN architecture and differences between way of protection for client-server network and DDSN described and also specified the types of attacks that are relevant to DDSN. The analysis of the Russian and foreign scientific literature on the subject. Based on the analysis has been formulated the task to develop methods for registering a new user, the method of «Trust» that allows users to «moderate» other users, and method of the authentication at the DDSN.*

Keywords: *Decentralized Distributed Social Network, DDSN; Social Networks; Decentralized Networks; protection methods.*

Китайская бизнес-грамота

На сей раз экспонат нашего виртуального музея – снова набор афоризмов. Точнее, ярких и лаконичных бизнес-советов от человека, к советам которого, право, стоит прислушаться. Это китайский предприниматель, основатель и глава Alibaba Group Джек Ма (см. статью о нем в этом номере журнала). Еще один «восточный маг», профессией которого стало превращение сказок в быль. Донельзя материальную, осязаемую и очень «денежную».

«Независимо от того, насколько трудна погоня за мечтой, последняя должна постоянно быть перед глазами – такой, какой вы ее впервые увидели. Только это мотивирует вас и спасет – от любых проявлений слабоволия».

«Помогайте молодым. Помогайте малышам – они когда-нибудь обязательно вырастут. И тогда изменят мир – благодаря посеянным в их головах вашим семенам».

«Я стараюсь быть счастливым, потому что знаю, что если буду несчастлив, то такими же будут и мои коллеги, и мои акционеры, и мои покупатели».

«Без интернета не было бы ни Джека Ма, ни Alibaba или Taobao».

«Я не технократ. Я смотрю на технологии глазами моих покупателей – обычных людей».

«Никогда не ведите дел с правительством. Крутить любовь – пожалуйста, но не доводите дело до брака».

«Как только у нас появляются деньги, мы тут же начинаем делать ошибки».

«Никогда не ставьте себе целью завершить за два года программу, рассчитанную на двадцать лет».

«Если вы не пытаетесь попробовать что-то, откуда вам знать, что шанс на успех все-таки существует?»

«Я – нормальный парень».

«Если не опускать рук, шанс на успех остается. Сдаться – это величайшая ошибка».

«Жизнь так коротка и прекрасна. Не относитесь так серьезно к работе – наслаждайтесь жизнью».

«Жалобы и счастливый случай – соседи».

«Если вы невелики ростом, вам не остается ничего другого, как полагаться на свой мозг, а не на физическую силу».

«Никто не думает о том, что его работа может быть бессмысленна для общества».

«Если мы – сыгранная команда, знающая, что делать, каждый из нас стоит десятка противников».

«Лидер – глава компании – обязан обладать твердостью, выдержкой и упорством и быть способным вынести то, что не смогут вынести его сотрудники».

«Мирные переговоры всегда сложны, всегда замысловаты».

«Мне бы не хотелось, чтобы у моего народа были глубокие карманы, но мелкие мысли».

«Не хочу, чтобы меня любили. Хочу, чтобы меня уважали».

«Никогда не знаешь, как много сможешь сделать в жизни».

«Надо учиться у конкурентов, но никогда их не копировать. Только начнете копировать – и вам конец».

«Неважно, если я ошибусь. По крайней мере я донесу свою идею до других – и кто-то сможет преуспеть».

«Вам приходится заставить покупателя поумнеть. Это не онлайн-торговля продает товары по сниженным ценам, это офлайн-торговля продает их по завышенным».

«Я бы назвал себя слепцом, сидящим верхом на слепом тигре».

«У нас никогда нет дефицита денег. У нас дефицит людей, способных мечтать и способных умереть за свою мечту».

«Очень важная вещь, которой нужно обладать, – это терпение».

«Никогда не соревнуйтесь в ценах, но в сервисе и инновациях».

«Умным нужен предводитель-глупец. Когда команда состоит из одних ученых умников, в качестве предводителя лучше всего выбрать неграмотного крестьянина. Он думает по-другому. Если иметь людей, видящих все с необычной точки зрения, легче выиграть».

«Когда вас полюбит покупатель, полюбит и правительство».

«Очень трудно понять окружающий мир, но свой-то внутренний мы понимаем. Мы знаем, что нам нужно и что мы хотим. Если лучше изучить самого себя, можно себя и изменить, чтобы приспособить к окружающему миру».

«Нужно иметь рядом с собой нужных людей, а не лучших».

«Все, что вам нужно, – это привить вашей команде ценности, чувство новизны и кругозор».

«Мы благодарны дню вчерашнему, но ожидаем лучшего завтрашнего».

«Я хочу изменить историю, сделать что-то важное в жизни и оказать влияние на людей – подобно тому, как мы повлияли на миллионы предпринимателей на Alibaba. Тогда и они в ответ полюбят и зауважают того, кто сделал что-то важное в их жизнях». EOF

Владимир Гаков

Системный администратор

Издается с 2002 года

«Системный администратор» включен в перечень ведущих рецензируемых журналов ВАК Минобрнауки РФ http://vak.ed.gov.ru/87

Включен в Российский индекс научного цитирования www.elibrary.ru

Научный руководитель журнала – председатель Редакционной коллегии
А.И. Аветисян, директор ИСП РАН, д.ф.-м.н., член-корреспондент РАН

Главный редактор
Галина Положевец, chief@samag.ru
Генеральный директор
Владимир Положевец
Шеф-редактор журнала «Системный администратор»
Владимир Лукин, lukin@samag.ru
Заместитель главного редактора
Ирина Ложкина, lozhkina@samag.ru
Заместитель главного редактора, официальный представитель редакции в Украине
Сергей Яремчук, grinder@samag.ru

Главный бухгалтер
Надежда Кан
buch@samag.ru

Юридический отдел
Владимир Столяров
stolyarov@samag.ru

Реклама
reklama@samag.ru

Распространение
Олег Иванов
subscribe@samag.ru

Дизайн обложки
Михаил Лебедев

Дизайн-макет
Марина Рязанцева, Дмитрий Бессонов

Иллюстрация
Виктор Чумачев

Издатель
ООО «Издательский дом Положевец и партнеры»
Адрес редакции
128017, г. Москва, 3-й пр-д Марьиной Рощи, д. 40, стр. 1, офис 606,
тел.: (499) 277-12-41, факс: (499) 277-12-45
Сайт журнала: www.samag.ru

Отпечатано в типографии
Типография «Практика». Тираж 17000 экз.

бизнес & информационные технологии

БИТ

Весь мир ИТ – в журнале «БИТ»

Оформив редакционную подписку на журнал «БИТ. Бизнес&Информационные технологии», вы получите возможность узнавать о событиях в мире информационных технологий из первых уст!

Все об ИТ
в бизнесе
и для бизнеса

Бумажная + электронная версии

5000 руб.

Бумажная версия – 4000 руб.

Электронная версия – 2000 руб.

Подписывайтесь прямо на сайте!

bit.samag.ru/subscribe

ДРУГОЙ УРОВЕНЬ УПРАВЛЕНИЯ

Kaspersky® Endpoint Security Cloud

Kaspersky Endpoint Security Cloud — новое решение «Лаборатории Касперского» для обеспечения безопасности бизнеса, которое сочетает многоуровневую защиту с исключительно простым облачным управлением.

Созданное с учетом потребностей небольших компаний, решение поможет управлять системой безопасности из любой точки мира и с любого устройства, подключенного к интернету. Оно полностью готово к работе и не требует специальных знаний или покупки дополнительного оборудования.

www.kaspersky.ru/cloud

KASPERSKY lab

БОЛЬШЕ ТЕСТОВ*
БОЛЬШЕ НАГРАД*
БОЛЬШЕ ЗАЩИТЫ

*kaspersky.ru/top3