Известия Гомельского государственного университета
имени Ф. Скорины, № 6 (93), 2015

**ИНФОРМАТИКА**

*УДК 004.052*

# Изменение подходов к безопасной загрузке операционных систем

А.В. ВОРУЕВ, В.И. РАГИН, А.И. КУЧЕРОВ, В.Д. ЛЕВЧУК

Рассмотрено предложение об организации безопасного вычислительного процесса и регулирование профилактики компьютерных вирусов в сетевых структурах. Описан ряд вариантов организации загрузки операционных систем из фиксированных образов. Описывается применение программных и аппаратных средств для реализации управляемой сетевой загрузки.
**Ключевые слова:** операционная система, сервер, тонкий клиент, ЛВС.

The proposal to organize safe computing process and control of prevention from computer viruses in the network structures are considered. A number of options for booting operating systems from fixed images are described.The use of software and hardware to implement a managed network boot is shown.
**Keywords:** operating system, server, thin client, LAN.

**Введение.** Точкой отсчета в появлении компьютерных вирусов можно считать труды известного ученого Джона фон Неймана по изучению самовоспроизводящихся математических автоматов, о которых стало известно в 1940-х гг. В 1951 г. он предложил способ создания таких автоматов. А в 1959 г. журнал Scientific American опубликовал статью Л.С. Пенроуза, посвященную самовоспроизводящимся механическим структурам. В ней была описана простейшая двумерная модель самовоспроизводящихся механических структур, способных к активации, размножению, мутациям, захвату. Позднее другой ученый Ф.Ж. Шталь реализовал данную модель на практике с помощью машинного кода на IBM 650 [1].

Основными путями проникновения вирусов в компьютеры, используемые в учебном процессе, являются сменные диски (гибкие и лазерные), а также компьютерные сети. Заражение жесткого диска вирусами может произойти и при загрузке программы с твердотельного накопителя (флешки), содержащей вирус. Вирус может попасть на сам носитель, даже если носитель просто подключили к системе зараженного компьютера и, например, прочитали ее оглавление.

Поскольку, компьютеры, применяемые для решения задач предприятия, объединены в компьютерные сети и имеют однотипные настройки операционной системы дальнейший сценарий развития событий чаще всего попадает в схему, удобную для распространения вирусов типа *сетевой червь*.

В 1988 г. Робертом Моррисом-младшим было анонсировано перед научным сообществом понятие «массовый сетевой червь». Прототип программы разрабатывался в конце 1990 г. с расчётом на поражение операционных систем UNIX Berkeley 4.3. Вирус изначально разрабатывался как безвредный и имел целью лишь скрытно проникнуть в вычислительные системы, связанные сетью ARPANET, и остаться там необнаруженным. Вирусная программа включала компоненты, позволяющие раскрывать пароли, имеющиеся в инфицированной системе, что, в свою очередь, позволяло программе маскироваться под задачу легальных пользователей системы, на самом деле занимаясь размножением и рассылкой копий. Вирус не остался скрытым и полностью безопасным, как задумывал автор, в силу незначительных ошибок, допущенных при разработке, которые привели к стремительному неуправляемому саморазмножению вируса [1].

По самым скромным оценкам инцидент с червём Морриса стоил свыше 8 миллионов часов потери доступа и свыше миллиона часов прямых потерь на восстановление работоспо-

собности систем. Общая стоимость этих затрат оценивается в 96 миллионов долларов (в эту сумму также включены затраты по доработке операционной системы). Ущерб был бы гораздо больше, если бы вирус изначально создавался с разрушительными целями.

Червь Морриса поразил свыше 6200 компьютеров. В результате вирусной атаки большинство сетей вышло из строя на срок до пяти суток. Компьютеры, выполнявшие коммутационные функции, работавшие в качестве файл-серверов или выполнявшие другие функции обеспечения работы сети, также вышли из строя.

Для решения вопросов, связанных с регулярным перезаражением операционных систем и предотвращением их несанкционированного использования в целях злоумышленников, высокую эффективность показало применение следующих технологий:

- применение «тонких клиентов»;
- реализация централизованной загрузки клиентских операционных систем.

**Применение «тонких клиентов».** «Тонким» клиентом или терминалом называют пользовательскую вычислительную систему, ресурсов оборудования которой недостаточно для автономной работы. В этом случае обслуживание вычислительного процесса осуществляется удаленной мощной вычислительной системой – сервером [2].

На рисунке 1 отражены некоторые возможные варианты архитектур программных систем с «тонкими клиентами» разного уровня «толщины».

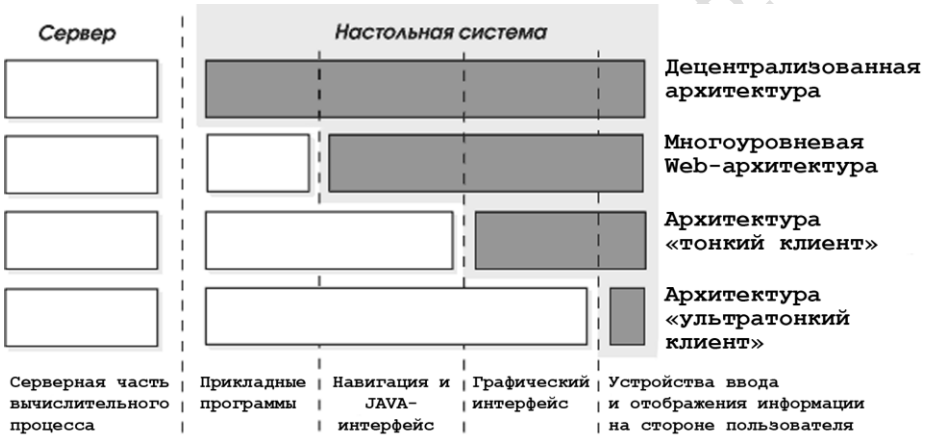


Рисунок 1 – Четыре уровня толщины «тонкого клиента»

Преимущества использования «тонких» клиентов:

- высокий уровень безопасности: непосредственно на пользовательских терминалах отсутствует возможность хранения конфиденциальных данных; все данные хранятся на серверах, где регулярно и централизованно резервируются;
- высокая надежность и длительный срок службы: тонкие клиенты служат дольше и реже выходят из строя; терминалы морально не устаревают рост требований к программному обеспечению вызывает лишь необходимость модернизации ядра системы, то есть сервера;
- уменьшение затрат на обслуживание, администрирование: установка нового и обновление существующего программного обеспечения происходит значительно быстрее и проще; наличие «контролируемой» среды на терминалах не позволяет пользователям запускать неразрешенные администратором приложения.

Низкий уровень требования к оборудованию позволяет разработчикам устройств данного типа до предела их минимизировать и компактно разместить в ограниченном объеме.

Производительность прогнозируется низкая. Из этого положения следует два вывода:

- тепловыделение настолько мало, что устройство практически не нуждается в охлаждении, а, соответственно, не шумит;
- мощность, потребляемая устройством для своей работы, настолько низкая, что обеспечить его достаточным уровнем энергоснабжения можно, используя бытовой блок питания +5V, либо используя технологию PoE (Power over Ethernet).

Работая с тонким клиентом, путь информационного обмена при загрузке операционной системы значительно увеличивается относительно локальной загрузки. Что видно на рисунке 2. Такой метод эффективно применяется в случае низких нагрузок единичного пользователя и при необходимости развернуть большое количество дешевых рабочих мест [3].

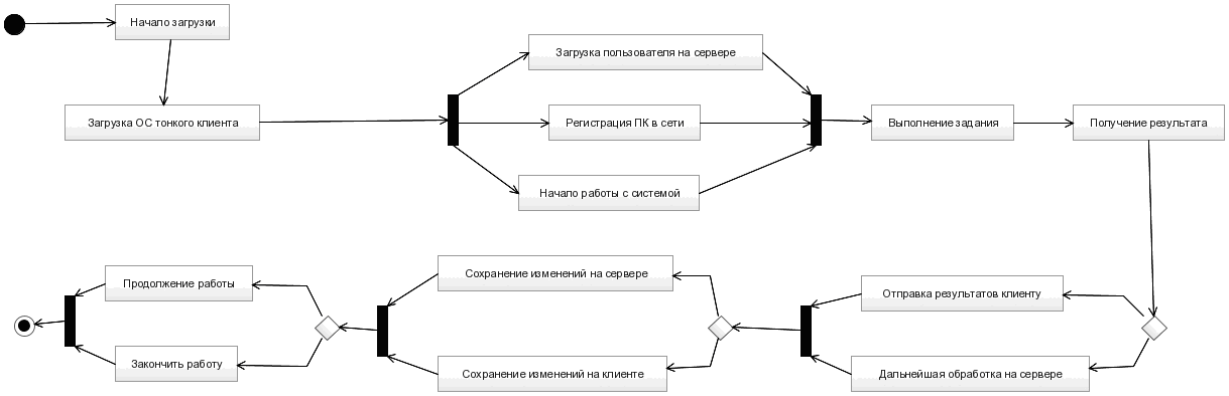


Рисунок 2 – UML диаграмма процесса загрузки тонкого клиента

При типовой работе трафик от клиента к серверу не превышает одного килобайта в секунду. Трафик в обратном направлении (сервер->клиент) составляет несколько десятков килобайт в секунду. Максимальное значение, достигнутое в ходе сеанса при открытии окна IE с графикой и динамическими flash-баннерами на mail.ru – 106664 байт/с. Среднее значение трафика составляет около 5–6 Кбайт/с (работа с браузером, просмотр документов MS Word без графики, открытие и работа программ со стандартными элементами пользовательского интерфейса). Такой низкий трафик достигается не только компрессией передаваемых данных (доходит до 300 %), но и, главным образом, тем, что во время сеанса клиенту передаются только команды на локальное отображение элементов пользовательского интерфейса (окна, кнопки, шрифтовое оформление) вместо их изображения [4].

Превышение максимальной пропускной способности канала не приводит к сбою, а лишь вызывает замедление обновления экрана клиента. Это позволяет при необходимости работать даже через модемное соединение с полосой пропускания 2–5 Кбит/с. Если принять за номинальную рабочую полосу пропускания Ethernet сети 100 Мбит, то данная полоса дает возможность работать либо 20–30 клиентам в режиме серьезной нагрузки без задержки обновления экрана, либо до 500 клиентов в режиме обычной офисной работы без активной динамической графики, требующей постоянной пересылки графических изображений на экран.

Несмотря на то, что использование «тонких клиентов» может серьезно исправить ситуацию с вирусной опасностью в сети учреждений образования, есть в этой технологии ряд серьезных недостатков, ограничивающих их применение для решения поставленной задачи:

- как правило, аппаратная составляющая «тонкого клиента» оптимизирована для работы одного семейства операционных систем, что сужает число дисциплин, которые можно проводить в учебной аудитории;
- при подготовке специалистов в области программирования требуется большое число ресурсов вычислительной системы, что спровоцирует перегрузку оборудования на стороне сервера, увеличит нагрузку на сеть передачи данных и, как следствие, снизит скорость взаимодействия пользователей с интерфейсом на рабочем месте;
- работа с системами обработки графических данных и нелинейной обработки видео будет крайне затруднена, либо требует специализированных программных средств, оптимизированных исключительно для терминальных систем.

Тонкий клиент хорошо подходит для использования в бухгалтерии, офисах и т. п., где нет высоких запросов ко времени обработки данных и прямого доступа к устройствам.

**Сравнение и реализация технологий клиентских станций.** В современном производственном процессе бездисковая загрузка мало представлена, однако на определенные ти-

пы задач (работа с графикой, звуком, обучение студентов) она явятся наиболее выгодной как экономически, так и с точки зрения безопасности.

Для аргументации использования этого метода загрузки, нам необходимо сравнить, насколько различается взаимодействие пользователей с локальной и удаленной машиной.

На рисунке 3 показан процесс взаимодействия с ПК пользователя локальной системы. Это классическая система работы и предполагаемый переход на удаленную загрузку является попыткой оптимизации ресурсов и упрощения администрирования системы в целом [5]. Рисунок 4 показывает ту же самую работу, но с использованием удаленной загрузки.

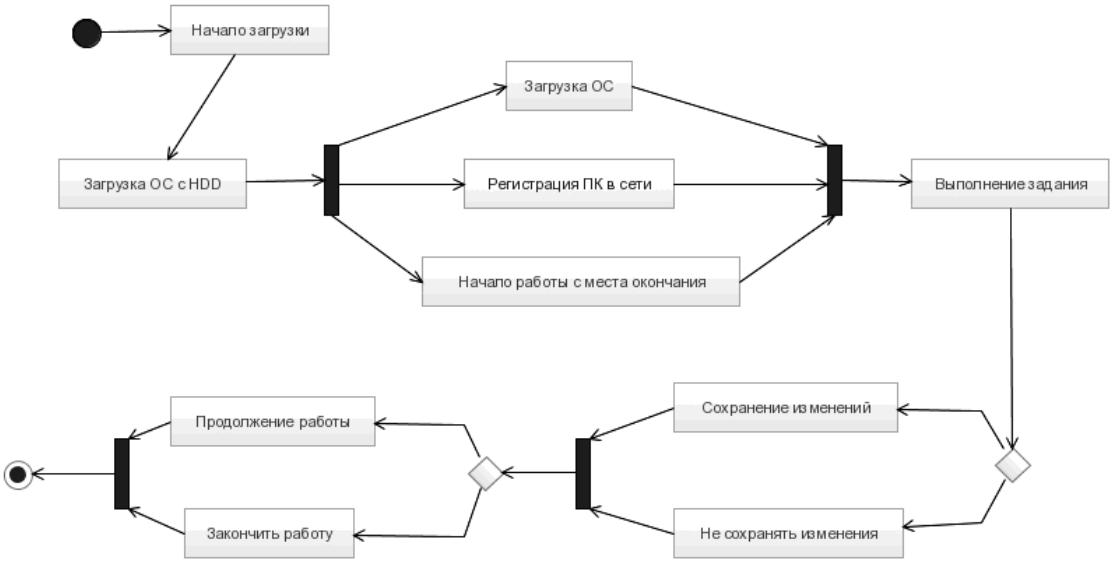


Рисунок 3 – UML диаграмма процесса локальной загрузки

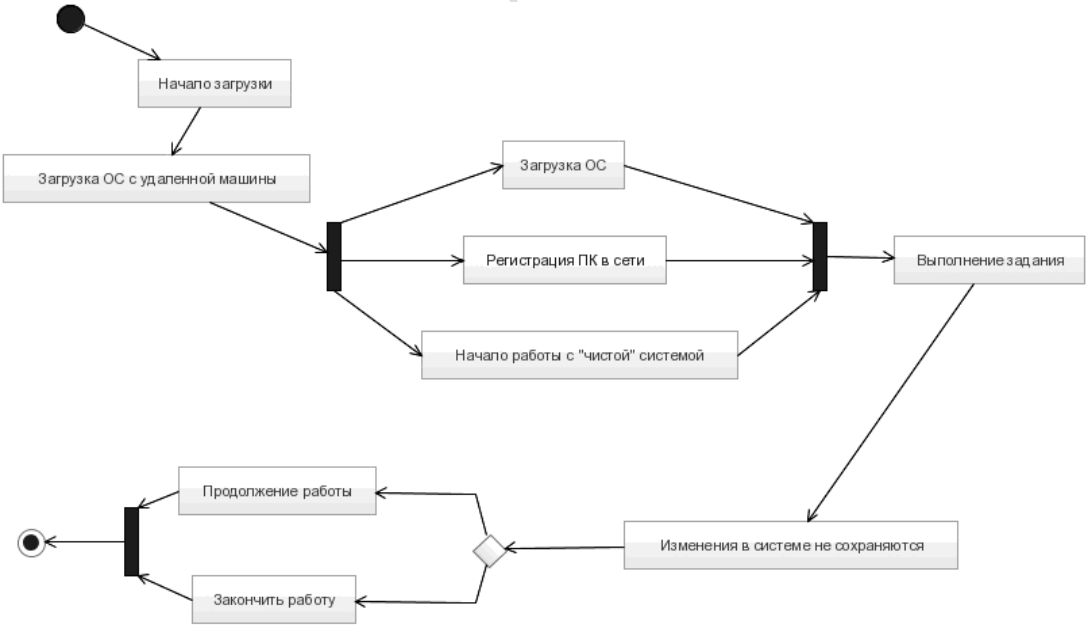


Рисунок 4 – UML диаграмма процесса удаленной загрузки

Как видно, принципиальных отличий не существует. Для пользователя, разница заключаются лишь в невозможности сохранить результат изменения самой системы (хотя и это ограничение является гибко настраиваемым). Это же ограничение распространяется на вирусы или любой другой нелегитимный программный код.

Несмотря на действия защиты методом запрета, у пользователя остаётся широкий круг возможностей. Раньше, при реализации политики доменов в локальных машинах, требова-

лось ограничивать пользователей в административных правах. В данном случае этот пункт является условным. Никакие изменения, сделанные в ходе сеанса работы, не сохранятся при последующей перезагрузке.

**Использование ISCSI для решения задач толстого клиента.** Для реализации прототипа удаленной загрузки был выбран StarWind iSCSI SAN с бесплатной лицензией [6]. За серверную платформу был взят MS Server 2012, бесплатный для студентов, преподавателей и учебных заведений. MS Server включает в себя большинство необходимых служб и настроек:

- встроенный TFTP сервер из состава компонент служб развертки;
- DHCP и NAT. При настройке особое внимание необходимо обратить на дополнительные параметры DHCP, параметры 66, 67, с их помощью мы задаем имя загружаемого файла и IP сервера TFTP;
- DNS и Active Directory для соединения с сетевыми учетными записями, если данные службы не активированы во внешней сети.

Образ системы устанавливается на виртуальной машине и настраивается, обязательна установка CCboot инициатора. По завершению настройки система конвертируется в .img образ с помощью StarWind V2V Image Converter. Полученный образ добавляется в StarWind iSCSI SAN и настраивается таргет. Раздача образов определяется MAC адресом машины.

Подключенный образ можно настроить как на запись, так и на отклонения сделанных изменений в системе. Во время первой загрузки необходимо обновить драйвера и ПО, а также произвести персональную настройку рабочего места. Сделанные изменения сохраняются в файле snapshot и могут быть использованы при следующей загрузке системы, не внося изменения в основной образ. После настройки файл snapshot также блокируется для записи.

**Пример реализации удаленной загрузки.** При разработке прототипа, сеть реализации загрузки размещалась в пределах одного помещения и состояла из сервера, маршрутизатора, персональных ПК для удаленной загрузки, а также подключения к общей сети и раздаче сервиса Wi-Fi [7]. Сервер используется для задач DHCP, NAT, PXE, ISCSI. Во всей схеме, сервер одно из «узких мест», соответственно, необходимо сохранить максимум ресурсов на основные задачи. Вторым «узким местом» является пропускная способность коммутатора, что крайне сказывается на первичной загрузке, это можно отследить на рисунке 5.

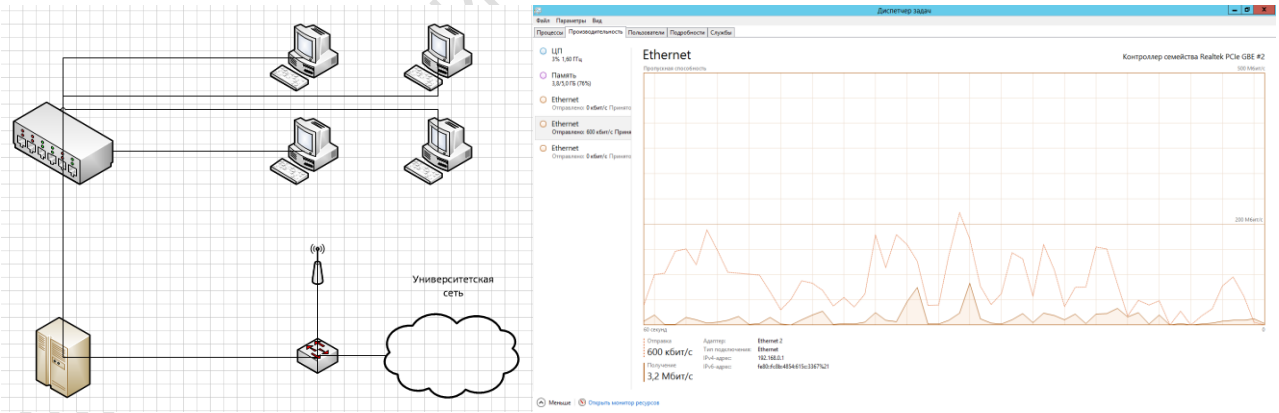


Рисунок 5 – Компоновка и график загруженности прототипа сети для загрузки с образов

В сети присутствует соединение 100 Мб/с с общей сетью и соединение 100 Мб/с с пользовательскими станциями. Между коммутатором и сервером канал 1 Гб/с. На графике видно, что даже одна пользовательская станция в такой конфигурации сети при загрузке полностью использует 100 Мб/с канал, т. е. всю полосу пропускания. Этот факт сказывается на скорости загрузки операционной системы - примерно от 90 до 120 секунд.

В измененной схеме (рисунок 6) сеть лишается основных недостатков.

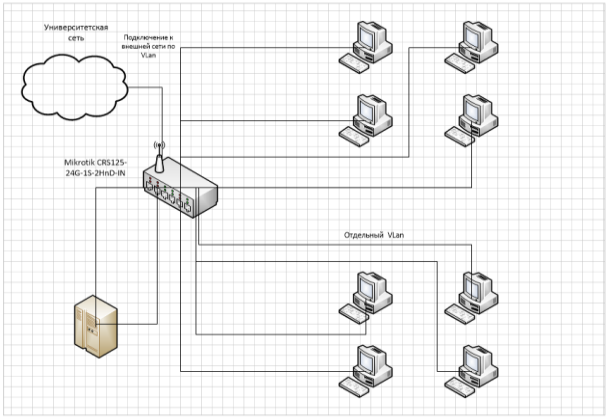


Рисунок 6 – Изменение компоновки сети для загрузки пользовательских станций с образов

Теперь надобность в NAT отпадает, что освобождает ресурсы сервера. За разделение и передачу трафика между VLAN, используется маршрутизатор Mikrotik CRS125-24G-1S-2HnD-IN. Каналы связи по всем направлениям поддерживают скорость 1 Гб/с, также маршрутизатор является точкой Wi-FI. На сервере остается только нагрузка DHCP, ISCSI, PXE.

## Литература

1. История компьютерных вирусов [Электронный ресурс]. – 2015. – Режим доступа : http://ru.wikipedia.org/wiki/История_компьютерных_вирусов. – Дата доступа : 20.02.2015.

2. Воруев, А.В. Архитектура ЭВМ / А.В. Воруев, О.М. Демиденко, А.И. Кучеров, В.Н. Кулинченко, В.Н. Леванцов // Учебно-методическое пособие Рекомендовано УМО вузов Республики Беларусь по образованию в области информатики и радиоэлектроники в качестве учебно-методического пособия для студентов учреждений, обеспечивающих получение высшего образования «Автоматизированные системы обработки информации. – ГГУ им. Ф. Скорины. – Гомель : 2011. – 192 с.

3. Официальный сайт программы управления данными Starwind [Электронный ресурс]. – 2014. – Режим доступа : http://ru.starwindsoftware.com/. – Дата доступа : 25.11.2014.

4. Коллективный блог [Электронный ресурс]. – 2014. – Режим доступа : http://habrahabr.ru/. – Дата доступа : 25.12.2014.

5. Моримото, Р., Ноэл, М. Microsoft Windows Server 2012. Полное руководство / Р. Моримото, М. Ноэл. – М. : ООО «Вильямс», 2013. – 1456 с. : ил.

6. Оффициальный сайт Starwind [Электронный ресурс]. – 2014. – Режим доступа: http://ru.starwindsoftware.com/. – Дата доступа : 25.11.2014.

7. Демиденко, О.М., Левчук, В.Д., Кучеров, А.И. Функциональные возможности программного комплекса адаптивной идентификации пользователей корпоративной сети / О.М. Демиденко, В.Д. Левчук, А.И. Кучеров // Проблемы, физики, математики и техники. – 2010. – № 3 (4). – С. 69–73.

Гомельский государственный
университет им. Ф. Скорины

Поступила в редакцию 30.06.2015