УДК 004.451.5

WINDOWS POWERSHELL

Безручко А.Н.
Научный руководитель – Щербаков А.В., к.ф-м.н., доцент

Windows PowerShell — расширяемое средство автоматизации от Microsoft, состоящее из оболочки с интерфейсом командной строки и сопутствующего языка сценариев.

Впервые публично язык был продемонстрирован на Профессиональной конференции разработчиков в сентябре 2003 года под кодовым названием Monad. Версия 1.0 выпущена в 2006 году и сейчас доступна для Windows XP SP2\SP3, Windows Server 2003, Windows Vista, и встроена в Windows Server 2008 как необязательный компонент. На данный момент доступна версия 4.0.

Запустив PowerShell, вы не обнаружите поначалу никаких различий между ним и cmd.exe (разве что цвет фона окна у PowerShell по умолчанию - синий). Но первое впечатление о схожести этих оболочекне совсем соответствует действительности.

То обстоятельство, что работа оболочки PowerShell основана на .NET Framework, является главным ее отличием от предыдущих командных оболочек Windows. PowerShell полностью объектно-ориентирована. Результатом выполнения команды в PowerShell является не некий текст, а объект класса .NET. Этот объект представляет собой собственно данные и имеет набор присущих ему свойств и методов.

Внутренние команды для работы с объектами в PowerShell называются командлетами. Для них придумано специальное единообразное именование в виде комбинации действие-цель. Например, для получения данных используется действие “get”, для вывода - “out” и т. д. Цель – это тип объекта, к которому будет применено действие. Командлеты можно рассматривать как мини-программы, исполняемые в среде PowerShell. Для повышения функциональности можно создавать собственные командлеты, устанавливать командлеты сторонних разработчиков. Кроме того, PowerShell позволяет выполнять функции, внешние сценарии и внешние исполняемые файлы. Так же в сценариях можно использовать WMI и Com объекты, а так же классы .Net, для которых предусмотрен специальный систаксис.

Оболочка имеет встроенный механизм псевдонимов команд. С одной стороны, псевдонимы используются для упрощения написания сценариев. Как правило, в этом случае в качестве псевдонима используется сокращенное наименование командлета (например, gc для Get-Content или fl для Format-List). С другой стороны, этот механизм обеспечивает совместимость интерфейсов различных командных интерпретаторов. К

Репозиторий БНТУ

примеру, имея опыт работы с cmd.exe, вы привыкли выводить содержимое папки с помощью команды dir. Выполнение этой команды в PowerShell приведет к тому же результату, хотя на самом деле оболочка вместо псевдонима dir будет выполнять командлет Get-ChildItem.

В области безопасности PowerShell значительно отличается от других оболочек. Использование сценариев VBScript/Jscript/CMD представляет потенциальную опасность для системы - для их исполнения достаточно щелкнуть по значку мышью. Опасность еще более возрастает, если пользователь вошел под учетной записью, входящей в группу администраторов. В PowerShell скрипт с расширением ps1 невозможно запустить на исполнение с помощью мыши - в системе такой файл будет открыт не в командной оболочке, а в Блокноте. Для запуска сценария необходимо запустить саму оболочку PowerShell, ввести имя файла и нажать клавишу Enter.

В новой оболочке так же невозможна подмена команд. Суть этого приема, применяемого злоумышленниками, заключается в следующем. Обычно у пользователя, не имеющего прав администратора, есть некоторые папки с разрешениями на запись и выполнение файлов. Характерный пример - папка C:\Documents and Settings\имя_пользователя. Вредоносная программа создает в такой папке исполняемый файл с именем, совпадающим с именем команды оболочки или именем исполняемой системной программы. При вызове команды запускается вредоносная программа, а не команда из system32

С PowerShell такой фокус не пройдет - для вызова скрипта, путь к которому не совпадает с путями, заданными в системной переменной %Path%, необходимо явно указать его расположение. Даже в том случае, когда скрипт расположен в папке, являющейся для оболочки текущей, необходимо указать путь в таком виде: .\имя_файла. Точка с обратным слешем указывают интерпретатору на текущую папку.

Еще одним механизмом обеспечения безопасности является политика выполнения сценариев. Изначально оболочка настроена так, что даже при правильном вызове сценария его выполнение будет запрещено, а пользователь получит соответствующее сообщение. Политика выполнения может переключаться в один из четырех режимов:

– Restricted - настройка по умолчанию, запуск любых сценариев запрещен;

– AllSigned - разрешен запуск сценариев, имеющих цифровую подпись надежного издателя; сценарии, созданные пользователем, также должны быть заверены центром сертификации;

– RemoteSigned - разрешен запуск сценариев, если они не являются доверенными, но созданы локальным пользователем. Сценарии, загруженные из Интернета, не имеющие подписи, не исполняются;

– Unrestricted - разрешен запуск любых сценариев.

Репозиторий БНТУ

Благодаря обширным возможностям и гибкости сценариев, PowerShell нашел применение в различных направлениях IT.

Особенно часто сценарии PowerShell применяются для автоматизации процессов администрирования Windows и ее компонентов. С помощью оболочки очень удобно управлять объектами Active Directory, администрировать Microsoft Exchange, работать с виртуальными машинами Microsoft Hyper-V.

Сценарии PowerShell часто используются в комплекте с планировщиком задач windows. Такое использование позволяет выполнять многие операции администрирования (бэкапирование, очистку) без участия человека.

Благодаря активному развитию PowerShell его возможности постоянно расширяются. В этом процессе участвуют как разработчики Microsoft так и сторонние компании.

Литература

1. Википедия [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Windows_PowerShell – Дата доступа 18.04.2014

2. WindowsFAQ. [Электронный ресурс]. – Режим доступа: http://all-ht.ru/inf/vpc/p_0_0.html – Дата доступа 19.04.2014

3. TexhNet [Электронный ресурс]. – Режим доступа: http://technet.microsoft.com – Дата доступа 18.04.2014

УДК 004.45

ИСПОЛЬЗОВАНИЕ ВИРТУАЛЬНЫХ МАШИН ДЛЯ УПРОЩЕНИЯ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Безручко А.Н., Бокшиц Е.А.
Научный руководитель – Белова С.В.

Виртуальная машина – это программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы и исполняющая программы на ней, или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы.

Система виртуализации – это специализированное программное обеспечение, используемое для имитации работы одной или нескольких реальных ЭВМ.

Впервые термин «Виртуальная машина» появился в конце шестидесятых годов прошлого века, когда электронные вычислительные машины стали неотъемлемой частью жизни человека. Однако, они разительно отличались от современных персональных ЭВМ и представляли собой огромные и

Репозиторий БНТУ