МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

WMI как средство обнаружения опасных событий и мониторинга системы и домена

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Афанасенко Кирилла Павловича

Научный руководитель
доцент, к.ю.н., доцент _____ А.В. Гортинский
22.01.2024 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2024 г.

Саратов 2024

ВВЕДЕНИЕ

WMI (Windows Management Instruments) представляет собой набор функций с широким набором возможностей, одна из которых — эффективный мониторинг состояния системы Windows, который, благодаря системе автоматизированных системных подписок, является надёжным и стандартизированным для каждого ПК.

Данная работа ставит целью разработку программного пакета для отслеживания данных, относящихся к локальному ПК и домену, который будет возможен для использования как отдельным пользователем, так и владельцем доменной системы с последующим отчётом в базу данных.

Инструментарий управления Windows (WMI) разработан корпорацией Майкрософт в рамках отраслевой инициативы управления предприятием через Интернет (WBEM), целью которой является создание стандартизированной технологии получения доступа к информации по управлению в среде предприятия. В инструментарии WMI используется модель CIM — отраслевой стандарт, служащий для представления систем, приложений, сетей, устройств и других управляемых компонентов. Модель CIM разрабатывается и обслуживается с помощью распределенной задачи управления Force.

Дипломная работа состоит из введения, 6 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 93 страницы, из них 43 страницы – основное содержание, включая 22 рисунка и 0 таблиц, список использованных источников из 22 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

Раздел 1 содержит описание базовой архитектуры WMI в целом и каждого отдельного её компонента. Описываются компоненты:

Провайдер WMI – COM объект, предназначенный для мониторинга одного или более WMI объектов. Предоставляет WMI информацию об управляемом объекте и передаёт сообщения от WMI к управляемому объекту.

Управляемый объект – логический или физический компонент, имеющий определённое значение для системы.

Ядро WMI – компонент операционной системы Windows, представляющий из себя службу WMI или winmgmt. Ядро WMI состоит из двух компонентов – непосредственного ядра и репозитория WMI. Репозиторий WMI организован по пространствам имён.

Служба WMI создаёт часть пространств (как root\cimv2) при запуске системы и предустанавливает стандартный набор определений классов.

Потребитель WMI – это конечный получатель информации, будь то приложение управления или программа, взаимодействующая с WMI.

Раздел 2 представляет описание различных преимуществ предоставляемых WMI, которые могут быть использованы при разработке приложений взаимодействующих с данным инструментарием. А именно:

Интерфейсы автоматизации. WMI имеет в составе набор заранее составленных интерфейсов автоматизации. Это значительно упрощает разработку провайдеров и проектирование классов.

.NET Интерфейсы управления. Пространство имён System.Management, которое так же используется для управления многими компонентами WMI, основывается на существующей сети COM/DCOM объектов. Как следствие, провайдеры WMI и их наборы классов становятся автоматически доступными для любых приложений .NET независимо от языка программирования (C++, C#, VB.NET или др.).

Программные интерфейсы C / C ++ COM / DCOM. Как и с большей частью компонентов Windows, при разработке COM/DCOM возможно использование функции провайдера на уровне интерфейсов разработки.

Возможности удаленного взаимодействия через DCOM и SOAP. WMI является более продвинутым инструментом чем COM и предлагает большие возможности чем просто локальное взаимодействие с этими интерфейсами, в частности — удалённое взаимодействие через интерфейсы DCOM. Так же инструментарий содержит возможность удалённого взаимодействия через интерфейсы SOAP в серверных версиях Windows, что возможно благодаря инициативе WS-Management.

Поддержка запросов. Одним из преимуществ WMI является поддержка запросов WQL даже в случае, когда этот функционал не входит в изначально разработанный провайдер.

Генератор шаблонов кода. Для ускорения процесса написания новых провайдер WMI, включающих в себя все интерфейсы COM/DCOM и соответствующие определения, команда WMI разработала мастер WMI ATL для создания шаблона кода.

Предсказуемость. Наличие конкретного постоянного набора интерфейсов означает возможность разработки специфичных наборов программных инструментов для налаживания удобного взаимодействия со средой разработки.

Раздел 3 является кратким описанием типов событий WMI, которые являются неотъемлемой частью работы с данным инструментарием в целом и мониторинга в частности. WMI имеет следующие типы событий:

Внутренние события WMI. Отслеживают изменения в состоянии тех управляемых ресурсов, которые выделены в отдельный класс WMI и хранятся в репозитории CIM, а также изменения в структуре самого репозитория CIM.

Внешние события WMI. WMI имеет возможность отслеживания внешних событий. Примером является изменение значения определенного ключа в реестре. Чтобы создать подписку внешнего события необходима её поддержка соответствующим провайдером WMI.

События таймера. Происходят либо один раз в заранее определённое время, либо несколько раз через заранее определённые временные интервалы, настраиваемые самим запрашивающим события пользователем.

Временные события WMI — самый простой, но самый эффективный тип событий для одноразового получения данных, так как функционируют один единственный раз после создания.

Постоянные потребители WMI событий регистрируются в самом репозитории CIM. При этом, активности скрипта не требуется, то есть, даже после перезагрузки системы будет происходить слежка за объектом, до тех пор, пока событие не будет удалено из CIM.

Раздел 4 содержит описание структуры запросов WMI, а так же объясняет различие между разовыми запросами и запросами событий.

Раздел 5 содержит описание использованных в работе классов WMI. В работе используются следующие классы:

Класс CIM_LogicalFile можно использовать как для получения общей информации о файле, так и используя как класс, на который идёт ссылка в классе событий.

Win32_SystemConfigurationChangeEvent используется как целевой для класса __InstanceOperationEvent.

Класс __InstanceOperationEvent - это класс событий, отвечающий за большую часть процессов в системе. Через свойство TargetInstance можно выбрать определённый тип операций. В работе используется совместно с другими классами для отслеживания непосредственных изменений в системе. Является классом событий и требует уточняющего слова ISA в запросе. Этот класс используется в качестве основного в работе для отслеживания работы программ и событий в папках.

Класс Win32_Process используется как целевой для класса __InstanceOperationEvent для определения работы программы на устройстве.

Класс Win32_PerfRawData_Tcpip_IP используется в совокупности с последними двумя свойствами для регулярного получения данных о трафике на устройстве.

Класс Win32_NetworkAdapter используется для получения данных о подключениях к сети на конкретном устройстве.

Класс Win32_NetworkAdapterConfiguration используется для получения сетевых адресов на конкретном устройстве.

Класс Win32_NetworkLoginProfile используется для получения локальных/использованных доменных пользователей на устройстве.

Класс Win32_IP4RouteTable используется для получения данных о IP таблицах на конкретных устройствах.

Класс Win32_Group используется для получения вышеописанных данных о группах Windows с устройств.

Класс Win32_DCOMApplication используется для получения комбинации этих данных на устройстве.

Раздел 6 содержит описание написанного программного пакета и его работы. Является основной частью работы.

Программный пакет состоит из двух компонентов — основного на языке C++ и дополнительного на языке Java.

Основной компонент представляет из себя программу на языке C++ для отслеживания событий при помощи WMI на локальном и доменных ПК.

Программа имеет следующие функции:

Функция «Программы» использует класс __InstanceOperationEvent в качестве основного в запросе получения событий и класс Win32_Process в качестве аргумента конструкции ISA. Данный запрос получает на вход путь к процессу Windows за действиями и взаимодействиями, с которым будет производиться наблюдение.

Функция «Папки» позволяет составить набор папок, за которыми будет вестись наблюдение в области изменения файлов. Для этого используется запрос мониторинга событий к классу __InstanceOperationEvent с классом CIM_DataFile в качестве аргумента для конструкции ISA.

Функция «Данные TCP» делает запрос к классу Win32_PerfRawData_Tcpip_IPv4, из которого в дальнейшем выбираются данные о количестве отправленных и полученных датаграмм за секунду.

Функция «Данные портов» делает запрос к классу MSFT_NetTCPConnection, из которого в дальнейшем выбирается поле LocalPort, которое и отвечает за локальные порты, зарегистрированные в системе.

Функция «Устройства» делает запрос к классу Win32_DeviceChangeEvent, из которого выбирается поле EventType отвечающее за тип события.

Подфункция «Сетевые адреса» обращается к классу Win32_NetworkAdapterConfiguration с целью выборки сетевой конфигурации. В частности, выбираются следующие поля: Description или «Описание», IPAddress или «IP адрес», MACAddress или «MAC адрес».

Подфункция «Локальные аккаунты» обращается к классу Win32_NetworkLoginProfile с целью выборки аккаунтов входа в сеть. В частности, выбираются следующие поля: Caption или «Реальное имя», Name или «Техническое имя», UserId или «ID пользователя».

Подфункция «Настройки сети»обращается к классу Win32_NetworkAdapter с целью выборки соединений с сетью. В частности, выбираются следующие поля: Availability или «Доступность соединения», Caption или «Заголовок соединения», Name или «Формальное имя соединения», NetConnectionID или «Тип соединения», NetConnectionStatus или «Состояние соединения».

Подфункция «IP таблицы» обращается к классу Win32_IP4RouteTable с целью выборки IP таблиц. В частности, выбираются следующие поля: Name или «Имя», Caption или «Заголовок», Description или «Описание», Information или «Информация», Protocol или «Протокол», Type или «Тип».

Подфункция «DCOM» обращается к классу Win32_DCOMApplication с целью выборки объектов DCOM. В частности, выбираются поля Name или «Имя приложения» и AppID или «ID приложения».

Подфункция «Группы Windows» обращается к классу Win32_Group с целью выборки групп Windows. В частности, выбираются поля: Caption или «Полное имя», SID, Status или «Статус».

Важной особенностью программы является возможность работы с другими ПК в домене. При известном имени устройства, пространства имён и

требуемых классов, подключение не требует никаких дополнительных данных. Строка подключения составляется из имени устройства и пространства имён.

В процессе работы программы все события записываются в технический файл «report.txt». Данный файл содержит в себе все используемые поля классов, к которым производится запрос. Также программа делает вывод полных результатов запросов в файл «output.txt».

Технический файл может быть передана дополнительному компоненту на языке Java. Данный компонент передаёт данные в базу данных SQL. Данный компонент был отделён от основной программы и написан на языке Java. Данному решению способствовали два фактора.

Фактор первый — язык Java куда лучше интегрирован с базами данных SQL и имеет готовые библиотеки с возможностью доступа к базе данных через специально отведённое имя для входа, которому возможно выдать крайне ограниченные права, вплоть до только возможности заносить новые строки в специально отведённую базу данных.

Фактор второй — разделение программы получения данных и скрипта вноса изменений в базу данных. Это приводит к возможности различных вариантов работы с ними, с переносом данных в техническом файле между устройствами при необходимости.

Связующим звеном между всеми таблицами является графа «Имя_Устройства». В случае, если работа ведётся на локальном ПК, перед непосредственной работой, программа делает отдельный запрос WMI при помощи класса Win32_ComputerSystem, из которого выделяется свойство Name. В случае, если работа ведётся на доменном ПК, имя устройства должно быть известно заранее для подключения, соответственно, его можно взять из уже введённого поля.

ЗАКЛЮЧЕНИЕ

В ходе работы был разработан двухкомпонентный программный пакет для отслеживания доменных и локальных данных.

Основным компонентом пакета является программа на языке C++, отслеживающая события и получающая данные на локальном или доменном устройстве при помощи WMI. При этом отслеживание событий может производиться по следующим параметрам: работа программы, изменение файлов в папке, количество полученных/отправленных пакетов данных, используемые устройством порты, подключение/отключение внешних устройств. Получение данных может производиться по следующим параметрам: локальные группы Windows, настройки сетевых соединений, адреса сетевых соединений, IP таблицы, DCOM объекты и локальные/использованные доменные аккаунты входа.

Дополнительным компонентом пакета является скрипт Java, который используется для внесения полученной информации в базу данных SQL.