Обеспечение безопасности работы реестра Windows

Набережнев Д.А.
Информатика
Группа КИБ31, Технический колледж ФГБОУ ВО «Тамбовский государственный технический университет», г. Тамбов, Тамбовской области
Научный руководитель: Мосягина Н.Г., Технический колледж ФГБОУ ВО «Тамбовский государственный технический университет», г. Тамбов, Тамбовской области

ВВЕДЕНИЕ

По данным сайта Statcounter [1] за последний год, операционную систему Windows используют более 80% россиян, что увеличивает количество угроз безопасности информации для обычного пользователя данной ОС. Реестр хранит очень много настроек и информации, а разработчики приложений активно им пользуются для хранения не всегда именно той информации, которая должна находиться в реестре, при изменении или при получении доступа к которой злоумышленник может получить доступ к конфиденциальной информации, хранящейся на компьютере.

Данный проект выполнен по заданию Центрально-Черноземного регионального учебно-научного центра по проблемам информационной безопасности – места прохождения производственной практики.

Цели и задачи исследования.

Целью данного исследовательского проекта является изучение реестра операционной системы Windows, а также разработка алгоритмов его использования для обеспечения информационной безопасности.

Для реализации поставленной цели необходимо решить следующие задачи:

- изучить работу и структуру реестра операционной системы Windows;
- разработать алгоритмы для работы с реестром операционной системы Windows;
- изучить средства для работы программ с реестром;
- осуществить выбор средства разработки представленных алгоритмов.

Новизна и научная содержательность проекта.

Проект выполнен с использованием методов объектно-ориентированного программирования и системного программирования. В работе используются новые методы и алгоритмы оптимизации работы с реестром, разработанные

автором. Предложенные алгоритмы позволят сократить время обработки запросов к реестру.

Реестр – иерархическая база данных, которая хранит низкоуровневые настройки операционной системы, такие как настройки IRQ (запросов прерывания), канала DMA (прямого доступа к памяти) и конфигурационные настройки драйверов устройств, а также необходимые данные и настройки приложений. Появившись в Windows 3.1 в основном использовался для хранения информации о конфигурации COM-объектов, которые были нужны для создания системных компонентов. С появлением Windows 95 и Windows NT функционал был расширен для рационального и централизованного хранения информации, в отличии от использовавшихся тогда INI-файлов, которые хранили информацию каждой индивидуальной программы и находились в разных местах, что увеличивало время обращения аппаратного обеспечения к файлу.

Реестр состоит из двух базовых элементов: ключей (разделов) и параметров. Ключи регистра являются контейнерами, похожими на папки, которые хранят параметры и подключи (подразделы). Параметры могут быть следующих типов:

– REG_NONE — без типа;

– REG_BINARY — содержат бинарные (двоичные) данные;

– REG_SZ — строковое значение, обычно хранится и отображается в кодировке UTF-16, а также оканчивается NULL-символом;

– REG_EXPAND_SZ — “расширяемое” значение строки, которое может хранить переменные среды окружения, обычно хранится и отображается в кодировке UTF-16, а также оканчивается NULL-символом;

– REG_DWORD — значение в виде двоичного слова, 32-битное беззнаковое целое число (числа в диапазоне от 0 до 4,294,967,295);

– REG_LINK — символьная ссылка на другой ключ реестра, определяющая корневой ключ и путь к целевому ключу;

– REG_MULTI_SZ — мультистроковое значение, которое представляет собой упорядоченный список непустых строк, обычно хранящихся и отображаемых в кодировке UTF-16, каждая строка оканчивается NULL-символом, а весь список оканчивается вторым NULL-символом;

– REG_QWORD — значение в виде четверичного слова, 64-битное целое число.

Доступ к иерархии ключей регистра осуществляется из корневого ключа (ветви). Всего существует 7 предустановленных ветвей:

– HKEY_LOCAL_MACHINE или HKLM — содержит информацию и параметры о конфигурации данного компьютера, такую как драйверы устройств, аппаратное оборудование и протоколы сети. Данная ветвь не хранится в виде файла на диске, а управляется исключительно памятью ядра в целях обозначить остальные подключи данной ветви и собирается каждый раз во время запуска компьютера;

– HKEY_CURRENT_CONFIG или HKCC — содержит информацию о профиле оборудования, используемом локальным компьютером при запуске системы. Информация не хранится на диске, но собирается во время запуска системы;

– HKEY_CLASSES_ROOT или HKCR — является ссылкой на HKLM\Software\Classes и HKCU\Software\Classes. Если определённое значение содержится в обоих подразделах, то в приоритет ставится значение из второго подраздела. Содержит информацию о зарегистрированных приложениях, такую как ассоциации файлов и уникальные идентификаторы OLE, которые являются ссылками для связывания и внедрения объектов в другие документы или объекты;

– HKEY_CURRENT_USER или HKCU — является ссылкой на один из профилей HKU. Содержит настройки текущего пользователя, находящегося в системе.

– HKEY_USERS или HKU — содержит информацию о всех пользователях компьютера;

– HKEY_PERFORMANCE_DATA (только в Windows NT, но невидим в редакторе реестра Windows) — данная ветвь предоставляет информацию о времени выполнения в данные производительности. Является самой уязвимой веткой из-за того, что её можно изменять удалённо при помощи протокола RPC, при этом можно делать с компьютером что угодно: от безобидной перезагрузки до кражи паролей и логинов, а также другой конфиденциальной информации;

– HKEY_DYN_DATA (только в Windows 9x) — представляет собой динамические данные о состоянии устройств, установленных на компьютере. Данные этого раздела формируются и изменяются операционной системой в процессе загрузки и в виде файлов не сохраняются.

Из-за того, что каждая ветвь имеет своё назначение, подключи данных ветвей также имеют разные функции. Ключи и подключи были исследованы, отфильтрованы и рассортированы в три главные группы: программное обеспечение, аппаратное обеспечение и сеть. Ключи и их функции указаны в таблице 1. [2, с. 126-127]

Таблица 1 — Ключи и их функции

Ключ	Описание
Программное обеспечение	
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Program Path	Установка приложений
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall	Удаление приложений
HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Profile List	Профили SID
HKLM\SOFTWARE\MICROSOFT\WindowsNT\CurrentVersion\SystemRestore	Точки восстановления
HKLM\SOFTWARE\Classes	Регистрация классов и ассоциация файлов
HKCU\Software\Classes	Настройки для текущего пользователя
HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Comdlg32	Наиболее используемые файлы
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU	Список наиболее используемых файлов
HKCU\SOFTWARE\Microsoft\Search Assistant\ACMru	Недавний поиск
HKLM\Software\Microsoft\Command Processor	Автоматический запуск
HKCU\Software\Microsoft\Protected Storage System Provider	Защищённое хранилище Windows
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces	Интерфейсы протокола TCP/IP
HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon	Последние вошедшие пользователи

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32	Последние посещённые наиболее используемые файлы
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU	Открытие и сохранение недавних файлов
HKCU\Software\Microsoft\Windows\SearchAssistant\ACMru	Файлы и слова, которые использовались при поиске
Аппаратное обеспечение	
HKLM\SYSTEM\CurrentControlSet\HardwareProfile\XXXX	Текущие настройки аппаратного обеспечения
HKLM\SYSTEM\MountedDevices	Установленные устройства
HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR	Вся информация по USB-устройствам
Сеть	
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\GUID	IP-адреса и порты
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MapNetworkDriveMRU	Карта сетевых устройств

Таким образом, мы выделили наиболее потенциально уязвимые ключи реестра, которые могут быть подвержены атаке.

Для того, чтобы обеспечить безопасность таких ключей в реестре, они имеют ассоциированные с ними списки доступа, которые описывают какие пользователи или группа пользователей имеют доступ/ограничение прав к реестру. Всего набор прав реестра состоит из 10 пунктов, которые можно отдельно настраивать для пользователя или группы пользователей:

- запрос значения — право на просмотр значения ключа;
- задание значений — право на создание новых значений ключа;
- создание подключей — право создавать подключи;
- подсчёт подключей — право на подсчёт ключей;
- уведомление — право запрашивать уведомления об изменении для ключей и подключей реестра;

– создание ссылок — право на создание ссылок, зарезервировано операционной системой;

– удаление — право удалять ключи;

– изменение списков доступа — право изменять привилегии пользователей в списках доступа;

– изменение владельца — право изменять владельца ключа;

– чтение списков доступа — право на просмотр прав пользователей в списке доступа.

Как и с остальными охраняемыми объектами в операционной системе, права доступа для ключей могут быть настроены отдельно, либо могут быть унаследованы от родительских ключей.

Также операционная система Windows имеет специальную функцию — защиту ресурсов, которая не позволяет администраторам и самой системе изменять значения необходимых ключей для поддержания целостности системы от вирусов и случайных модификаций. Ключи реестра имеют внешнюю защиту, которая не позволяет удалять или изменять их по иным причинам. Они содержат символы NULL, которые не могут быть удалены или изменены при помощи стандартных редакторов реестра, а требуют специальную программу для удаления, такую как RegDelNull.

Реестр – главная база данных в Windows для хранения настроек конфигурации. Он хранит огромное количество сложных, незадокументированных и незащищённых данных о конфигурации, которые очень важны для системы и которые окажут влияние на стабильность Windows, если пропадут или будут неправильно настроены. Такая особенность реестра Windows сделала его самым уязвимым компонентом всей операционной системы. Пользователи с правами администратора имеют практически полный контроль для редактирования ключей реестра, а так как большинство людей за своими домашними компьютерами являются администраторами системы, они становятся самыми лакомыми жертвами злоумышленников. Нечаянно запущенная вирусная программа с правами администратора может привести к модификации реестра, что окажет неблагоприятное влияние на стабильность системы. Если настройки реестра были изменены нарушителем, например на включение автозапуска USB-накопителей или других программ в определённое время, то безопасность может быть нарушена. Как бы то ни было, системный

администратор не может просто закрыть реестр. Следовательно, наиболее важно настроить списки доступа к реестру. Кроме того, все значения контроля доступа и различные параметры, находящиеся в реестре, также привлекают хакера в систему. Он может изменить значения так, чтобы определённая программа, например браузер, вела себя по-другому. Обеспечение охраны администраторского аккаунта, однако, будет намного более сложно реализовать, нежели просто установить хороший и безопасный пароль. Особенности, баги и небезопасные значения настроек конфигурации по умолчанию будут постоянно оставлять множество дыр в безопасности, которые нарушитель может использовать в своих целях, чтобы захватить контроль над системой. Очень важно обеспечивать безопасность реестра Windows, чтобы сделать всю систему безопасней и предотвратить последствия от возможных ошибок, а, если они случатся, то выявить ошибку в операционной системе будет намного легче. [3, с. 841-842]

Для редактирования реестра может использоваться множество программ, однако в большинстве случаев используют предустановленную на операционную систему программу regedit.exe. Она предоставляет следующие базовые функции:

– быстрый импорт и экспорт всего реестра или его частей при помощи файлов реестра;

– подключение сетевых реестров компьютеров в локальной сети для удалённой работы с ними;

– изменение, создание и удаление ключей, а также принадлежащих им различных параметров;

– настройка списков доступа для каждого пользователя или группы пользователей к отдельным ключам;

– сравнительно быстрый поиск необходимых ключей и их значений по всему реестру;

Также необходимо периодически делать резервные копии реестра Windows, чтобы в случае его редактирования нарушителем или неисправности можно было бы его восстановить в кратчайшие сроки. Если необходимы только жизненно важные для функционирования системы ключи, то можно отдельно сделать копию такого ключа и его подключей.

Разработка программа для выявления и удаления USB-устройств

В качестве задания производственной практики в Центрально-Черноземном региональном учебно-научном центре по проблемам информационной безопасности была поставлена задача разработать программу нахождения серийных номеров всех USB-накопителей, когда-либо подключённых к системе, дату и время их подключения, а также возможность удаления упоминаний серийных номеров из всех веток реестра.

Для выполнения данной задачи я разработал следующий алгоритм работы программы:

- для каждого подключа в ключе HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR необходимо открыть каждый дочерний ключ подключа и сохранить его название в программе, оно будет являться серийным номером USB-накопителя. Также необходимо сохранить значения ClassGUID и ContainerID;
- отформатировать полученные номера: убрать такие знаки, как “&”, “*”, “_” и другие, так как они могут добавляться ошибочно, удалить повторяющиеся значения;
- предложить пользователю выбор USB-накопителя для просмотра его информации и удаления данных о нём;
- показать всю информацию пользователю;
- при нажатии кнопки удаления, просканировать весь реестр на наличие серийных номеров, значений ClassGUID и ContainerID данного накопителя, удалить эти значения из реестра.

Таким образом, передо мной встал выбор среды разработки и языка программирования для работы с реестром. Я сразу выбрал своей средой разработки Visual Studio 2019 Community Edition из-за её удобности, скорости, опыту работы, а также бесплатности.

Однако к выбору языка программирования нужно было подойти с особыми требованиями:

- из-за того, что реестр очень большой, необходима скорость работы языка с операционной системой;
- изучение языка и его библиотеки для работы с реестром не должно занять много времени;

– язык должен поддерживать объектно-ориентированное программирование.

В виду вышеперечисленных требований пал выбор на три потенциально возможных языка: C++, C# и Python. Несмотря на высокую скорость работы с реестром, изучение библиотеки C++ для работы с ним заняло бы много времени, а Python не поддерживал перегрузок функций и динамических переменных, а также был самым медленным из всех возможных вариантов, поэтому C# стал самым предпочтительным вариантом из-за удобного и понятного синтаксиса библиотеки System.Win32, которая содержала в себе работу с компонентами операционной системы Windows, в том числе и работу с реестром. Для изучения языка программирования C# я воспользовался книгой «Программирование. Базовый курс C#» [4]

Реализация алгоритма выявления и удаления USB-устройств на языке C#.

Главное окно программы представлено на рисунке 1.

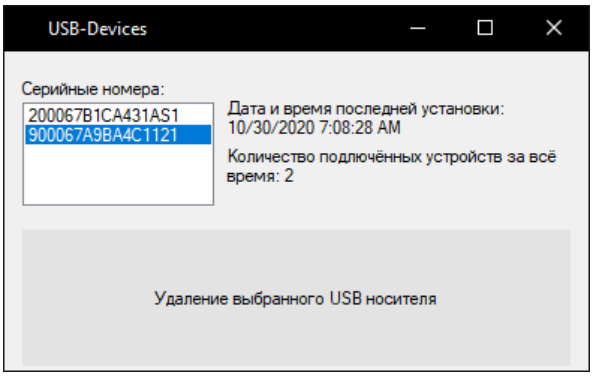

Рисунок 1 — Окно программы

Интерфейс программы простой и состоит из двух активных элементов: кнопки и списка. При нажатии на элемент в списке о нём появится информация справа в виде даты и времени. При нажатии на кнопку программа удалит все упоминания о данном USB-накопителе. Рассмотрим обработчик события при нажатии на кнопку:

```
private void button_Click(object sender, EventArgs e)
{
   controller = new ServiceController();
   if (controller.ServiceName != @"USB_Devices")
   {
```

```
            ManagedInstallerClass.InstallHelper(new string[] { "./USB-
Devices_service.exe" });
        }
        controller.ServiceName = @"USB_Devices";
        string[] args = { listBox1.GetItemText(listBox1.SelectedItem)};
        controller.Start(args);
        if (!controller.Status.Equals(ServiceControllerStatus.Stopped))
        {
           controller.Stop();
        }
        Delete();
    listBox1.Items.Remove(listBox1.SelectedItem);
        MessageBox.Show("Выбранный USB-накопитель удалён!");}
```

В данном фрагменте кода вызывается специальной контроллер, который управляет службами в Windows. Он запускает специальную службу и передаёт ей аргумент, в качестве которого выступает выбранный серийный номер в списке. После того, как служба обработала запрос, она закрывается и высвечивается уведомление об успешном удалении серийного номера из реестра.

ЗАКЛЮЧЕНИЕ

Реестр Windows является очень важной частью операционной системы. Он позволяет быстро получить необходимую информацию, редактировать те элементы, которые не получится редактировать обычными способами. Разнообразие типов данных, которые может принять значение ключа, позволяет работать реестру как с программным, так и с аппаратным обеспечением. Удобный и быстрый импорт и экспорт реестра позволяет быстро переносить настройки с компьютера на компьютер, а также делать резервные копии необходимых ключей.

Очень важно, чтобы целостность, конфиденциальность и доступность реестра никогда не нарушались. Данная работа может помочь понять структуру реестра, его функционирование, а также выявить самые уязвимые ключи реестра Windows и способ обеспечения их безопасности.

Результаты проекта могут быть предложены работающим в сфере информационной безопасности, системным администраторам, а также рядовым пользователям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. StatCounter. — [Электронный ресурс].— Режим доступа: https://statcounter.com (дата обращения: 05.11.2020).
2. M.H.N.M. Nasir, N.H. Hassan and S.S.M. Fauzi, 2008. Protecting Windows Registry Directory and Hence Increasing the Security Level of the Windows Operating System. Information Technology Journal, 7. С. 840-849.
3. International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012 DOI: 10.5121/ijnsa.2012.4209, 121 c ,FORENSIC ANALYSIS OF WINDOWS REGISTRY AGAINST INTRUSION. С. 124-127.
4. Подбельский, В. В. Программирование. Базовый курс C# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // ЭБС Юрайт [сайт]. — Режим доступа: https://urait.ru/bcode/450868 (дата обращения: 5.11.2020).