УДК 004.031

5.2.2. Математические, статистические и инструментальные методы в экономике

МЕТОДИКА РЕАЛИЗАЦИИ РАБОТЫ ПРОГРАММНОГО ИНТЕРФЕЙСА МЕТОДОМ ПРЯМОГО ДОСТУПА К УСТРОЙСТВАМ ВВОДА RAW INPUT API

Минина Евгения Александровна
к.э.н., доцент

Кузьмина Эвелина Вячеславовна
к.п.н., доцент

Минин Никита Артемович
студент 3 курса факультета прикладной информатики
Кубанский государственный аграрный университет имени И. Т. Трубилина, Краснодар, Россия

В статье представлена методика реализации программного интерфейса для взаимодействия с устройствами ввода с использованием технологии Raw Input API, встроенной в операционные системы семейства Windows. Основное внимание уделено применению данного метода для создания кейлоггеров — программ, регистрирующих ввод с клавиатуры. Описаны архитектурные принципы и ключевые этапы реализации, включая регистрацию устройств ввода и обработку низкоуровневых сообщений WM_INPUT. Особенностью подхода является использование скрытых окон, предназначенных только для обработки сообщений, что обеспечивает скрытность работы приложения. Рассмотрены практические аспекты, такие как запись данных в лог-файл, обработка специальных клавиш и возможность добавления программы в автозагрузку. Отмечены преимущества Raw Input API, среди которых: прямой доступ к данным устройств, возможность работы с несколькими однотипными устройствами одновременно и независимость от высокоуровневых системных обработчиков. Подчёркивается, что несмотря на потенциальное использование технологии в целях несанкционированного сбора информации, она также может служить для создания безопасных и гибких систем обработки ввода при условии реализации соответствующих мер защиты

Ключевые слова: RAW INPUT API, НИЗКОУРОВНЕВЫЙ ВВОД, ПРОГРАММНЫЙ ИНТЕРФЕЙС, WINDOWS, ОБРАБОТКА СОБЫТИЙ, КЕЙЛОГГЕР, БЕЗОПАСНОСТЬ

http://dx.doi.org/10.21515/1990-4665-213-040

UDC 004.031

5.2.2. Mathematical, statistical and instrumental methods in economics

METHODOLOGY FOR IMPLEMENTING THE OPERATION OF A PROGRAM INTERFACE BY DIRECT ACCESS TO INPUT DEVICES RAW INPUT API

Minina Evgeniia Alexandrovna
Cand.Econ.Sci, associate professor

Kuzmina Evelina Viacheslavovna
Cand.Ped.Sci., associate professor

Minin Nikita Artemovich
srudent
Kuban State Agrarian University, Krasnodar

The article presents a methodology for implementing a software interface for interacting with input devices using the Raw Input API technology built into Windows operating systems. The main focus is on using this method to create keyloggers, programs that register keyboard input. The architectural principles and key implementation steps are described, including the registration of input devices and the processing of low-level WM_INPUT messages. A special feature of the approach is the use of hidden windows designed only for message processing, which ensures the secrecy of the application. Practical aspects such as writing data to a log file, processing special keys and the possibility of adding a program to startup are considered. The advantages of the Raw Input API are noted, including: direct access to device data, the ability to work with several devices of the same type simultaneously, and independence from high-level system handlers. It is emphasized that despite the potential use of technology for unauthorized information collection, it can also serve to create secure and flexible input processing systems, provided appropriate security measures are implemented

Keywords: RAW INPUT API, LOW-LEVEL INPUT, PROGRAMMING INTERFACE, WINDOWS, EVENT HANDLING, KEYLOGGER, SECURITY

Одним из методов реализации кейллогеров является использование интерфейса прямого доступа к устройствам ввода - Raw Input API [1]. Основное его отличие от прочих методов [3, 4], работающих с устройствами ввода, состоит в том, что интерфейс имеет способ прямого взаимодействия с устройствами ввода, не включающий более высокоуровневые средства, давая возможность получать информацию без вмешательства в работу других процессов системы [2].

Особенностями данного подхода можно назвать то, что приложение должно явно запрашивать получение сообщений WM_INPUT, так как по умолчанию этого не происходит [6]. Это делает данный метод более легким для обнаружения, но так как он остается редко используемым, то вероятность его обнаружить не повышается при использовании стандартных антивирусных служб системы.

В статье представлена методика реализации работы программных кейлоггеров, работающих в режиме пользователя. Необходимо отметить, что выявлены отличия от первоначальной модели доступа к информации ввода от устройств в семействе Windows [6]. Изначально модель приложения регистрировала независимые от определенного устройства ввода данные с помощью таких сообщений, как WM_CHAR, WM_MOUSEMOVE и WM_APPCOMMAND. При установлении прямого доступа к устройству, приложение регистрировало устройства, от которых оно получает информацию о вводе. Так, приложение, которое берет данные из устройств ввода напрямую, должно обрабатывать сообщение WM_INPUT.

Рассмотрим архитектурные принципы и этапы, которые проходит приложение и подсистема ввода для совместной работы.

Основные этапы реализации работы программного интерфейса методом прямого доступа к устройствам ввода Raw Input API включают в себя:

1) регистрацию устройства;
2) обработку сообщения входных данных с устройств ввода.

Приложение, использующее модель прямого доступа к устройствам ввода должно иметь окно, чтобы получать сообщения WM_INPUT. Однако, кейлоггер должен быть незаметной программой и видимое окно для него непозволительная роскошь. В операционной системе Windows есть такой тип окон, как окна только для сообщений (message-only windows). Такое окно не отображается на экране компьютера и не появляется в списке окон. Однако приложение, имеющее такое окно способно получать сообщения, как и любое другое оконное приложение Windows.

Выполнение программы начинается, как и положено, с функции WinMain() либо, как в нашем случае, для приложений Unicode с wWinMain(). Весь код этой функции приводить бессмысленно, он стандартный, за исключением создания окна для сообщений.

Далее применена функция оконной процедуры, а особенно участки, отвечающие за обработку сообщений WM_CREATE и WM_INPUT. Прототип её стандартен.

Далее идет код, выполняющийся при создании окна: определяем переменные и структуры, необходимые в этом блоке кода.

Следующий шаг определения пути к файлу, куда будут протоколироваться нажатия клавиш. Здесь, вызывается функция get_default_log_path(), которая определяет путь для сохранения лога на текущей машине. Лог сохраняется в папке пользователя по пути %ApplicationData%\Roaming\klog.txt.

Функция die() вызывается в случае критической ошибки. В качестве первого параметра она принимает текст сообщения об ошибке, в качестве второго – код который программа возвращает при завершении. Её реализация тривиальна и приводиться в исследовании не будет.

Далее, определяется текущее время и дата, которые записываются в лог для обозначения начала сессии записи. Запись в файл-протокол производится с помощью функции write_log(), в качестве первого параметра она принимает путь к файлу, в качестве второго указатель на блок данных, в качестве третьего – размер данных.

Затем приложение регистрируется для получения сообщений WM_INPUT о вводе с клавиатуры. Таким образом, описание блока кода, отвечающего за обработку сообщения WM_CREATE, закончено.

Код функции записи в лог-файл write_log() каждый раз открывает файл и записывает данные в конец.

Теперь требует характеристик блок, отвечающий за обработку сообщения WM_INPUT. Именно здесь происходит обработка и получение данных от клавиатуры. Для начала, описаны переменные case WM_INPUT, используемые в этом блоке.

Результатом будет получение данных о вводе и размере данных от Raw Input. Далее необходимо выделить память для данных Raw Input/ Теперь собственно получим данные о вводе от устройства. Необходимо сохранить себе интересующие нас данные и освободить выделенную память: ri_data = (PRAWINPUT) pRawInputData.

Далее сохраняем себе скан-код и виртуальный код клавиш, конвертируем виртуальный код клавиши в ASCII, освобождаем данные о вводе.

Теперь обработаем полученные данные. И сохраним данные о нажатии клавиши при необходимости.

Обработаем некоторые клавиши (Tab, Enter, Backspace) особым образом, чтобы они были видны в логе:

Обработаем остальные клавиши, при этом будем учитывать текущую раскладку клавиатуры и состояние нажатия кнопок и получим информацию о нажатых клавишах. Определим текущую раскладку для

активного окна. Сконвертируем виртуальный код клавиши в символ юникода согласно раскладке и информации о нажатых клавишах, запишем символ в лог. Обработка сообщения о поступлении ввода закончена.

Можно также реализовать функцию для добавления программы в автозагрузку. Её вызов можно добавить в блок кода, где обрабатывается сообщение WM_INPUT. Функция пытается добавить исполняемый файл программы в автозапуск, указывается путь в реестре к ветке автозагрузки, получаем хэндл на ветку реестра HKCU. Таким образом, создается новая запись, указывается в качестве значения записи путь к исполняемому файлу.

Этого вполне достаточно для кейлоггера. Можно реализовать также отправку файла-протокола по электронной почте, копирование исполняемого файла программы в скрытое хранилище и таким образом превратить кейлоггер в троянскую программу. Однако этого делать не рекомендуется, реализация подобного функционала в программе оставлена на усмотрение пользователя.

В процессе исследования определены особо значимые плюсы программного интерфейса прямого доступа к устройствам ввода [5]: приложение не должно получать доступ или обнаруживать устройство; приложение получает информацию напрямую от устройства и обрабатывает полученные данные как ему угодно; приложение может различить конкретное устройство ввода, даже если оно получает ввод от нескольких устройств одного типа; например, обработать по-разному две одновременно подключенные компьютерные мыши; приложение контролирует поступление данных от конкретного типа устройств или конкретного устройства; новые устройства ввода могут без проблем использоваться сразу после своего появления, не дожидаясь пока ввод от них будет стандартизирован в сообщениях операционной системы.

Метод реализации взаимодействия с устройствами ввода и использованием модели Raw Input API демонстрирует высокую эффективность при разработке приложений прямого доступа к данным ввода. Низкоуровневый характер и интерфейса обеспечивает независимость от состояния подсистемы Windows и предоставляет возможность обработки ввода с нескольких устройств одновременно.

В статье проанализированы принципы построения программ для захвата данных с клавиатуры. Основы Raw Input API дают понимание как программное и аппаратное обеспечение работают вместе; это может быть использовано для создания систем, которые не позволяют в них проникать без разрешения. Необходимо отметить, что в статье мы осознанно обошли правовые и моральные аспекты применения такого рода программ. Обладая подобными знаниями во избежание проблем, связанных с информационной безопасностью, важно понимать и не выставлять их напоказ широкой аудитории.

Подобного рода программное обеспечение может быть использовано, как в целях создания более удобных инструментов работы с устройствами ввода, так и в целях скрытого считывания данных. Raw Input API следует использовать в качестве универсального инструмента для создания безопасных и гибких способов обработки ввода при соответствующей реализации защиты данных [6] в системе.

Литература

1. Баранов А. И. Программные интерфейсы для работы с устройствами ввода: Учеб. пособие / А. И. Баранов. – Москва: Техника, 2018. – 320 с.

2. Дежинин П. Н., Хорошилов А. В., Тележников В. Ю. Формирование методологии разработки безопасного системного программного обеспечения на примере операционных систем // Тр. ИСП РАН. 2021. Т. 33. № 5. С. 25-40.

3. Караев А. В. Актуальность и особенности внедрения ИТ-сервисов с применением облачных технологий / А. В. Караев, Д. О. Емельянов, Т. П. Барановская // Информационное общество: современное состояние и перспективы развития : сборник материалов XIII международного форума, Краснодар, 13–18 июля 2020 года. –

Краснодар: Кубанский государственный аграрный университет имени И.Т. Трубилина, 2020. – С. 387-390.

4. Кузьмина, Э. В. Повышение эффективности деятельности предприятия на основе приложения для визуализации бизнес-стратегии / Э. В. Кузьмина, Е. А. Минина // Финансовый менеджмент. – 2023. – № 6-2. – С. 101-111. – DOI 10.25806/fm6-22023101-111.

5. Минин, Н. А. Программный интерфейс методом прямого доступа к устройствам ввода Raw input API / Н. А. Минин, Е. А. Минина // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2024. – № 201. – С. 420-428.

6. Петров А. А. Анализ основных технических характеристик систем активной защиты в сетях общего пользования / А. А. Петров, Е. А. Минина // Современная экономика: проблемы и решения. – 2022. – № 10(154). – С. 34-46.

7. Mathematical modeling of corporate network tolerance troubleshooting methods / A. A. Petrov, D. N. Savinskaya, E. A. Minina, L. K. Dunskaya // Modern Economics: Problems and Solutions. – 2020. – No. 12(132). – P. 35-45.

References

1. Baranov A. I. Programmny`e interfejsy` dlya raboty` s ustrojstvami vvoda: Ucheb. posobie / A. I. Baranov. – Moskva: Texnika, 2018. – 320 s.

2. Dezhinin P. N., Xoroshilov A. V., Telezhnikov V. Yu. Formirovanie metodologii razrabotki bezopasnogo sistemnogo programmnogo obespecheniya na primere operacionny`x sistem // Tr. ISP RAN. 2021. T. 33. № 5. S. 25-40.

3. Karaev A. V. Aktual`nost` i osobennosti vnedreniya IT-servisov s primeneniem oblachny`x texnologij / A. V. Karaev, D. O. Emel`yanov, T. P. Baranovskaya // Informacionnoe obshhestvo: sovremennoe sostoyanie i perspektivy` razvitiya : sbornik materialov XIII mezhdunarodnogo foruma, Krasnodar, 13–18 iyulya 2020 goda. – Krasnodar: Kubanskij gosudarstvenny`j agrarny`j universitet imeni I.T. Trubilina, 2020. – S. 387-390.

4. Kuz`mina, E`. V. Povy`shenie e`ffektivnosti deyatel`nosti predpriyatiya na osnove prilozheniya dlya vizualizacii biznes-strategii / E`. V. Kuz`mina, E. A. Minina // Finansovy`j menedzhment. – 2023. – № 6-2. – S. 101-111. – DOI 10.25806/fm6-22023101-111.

5. Minin, N. A. Programmny`j interfejs metodom pryamogo dostupa k ustrojstvam vvoda Raw input API / N. A. Minin, E. A. Minina // Politematicheskij setevoj e`lektronny`j nauchny`j zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta. – 2024. – № 201. – S. 420-428.

6. Petrov A. A. Analiz osnovny`x texnicheskix xarakteristik sistem aktivnoj zashhity` v setyax obshhego pol`zovaniya / A. A. Petrov, E. A. Minina // Sovremennaya e`konomika: problemy` i resheniya. – 2022. – № 10(154). – S. 34-46.

7. Mathematical modeling of corporate network tolerance troubleshooting methods / A. A. Petrov, D. N. Savinskaya, E. A. Minina, L. K. Dunskaya // Modern Economics: Problems and Solutions. – 2020. – No. 12(132). – P. 35-45.