**Научно-исследовательская часть**

1. **. Постановка задачи проектирования**

Задача курсовой работы состоит в программной реализации технологии внедрения DLL библиотеки в целевой процесс «Reflective DLL Injection, которая позволяет улучшить функционал уже существующего программного обеспечения. А также необходимо разработать удобный и понятный интерфейс

1. **. Описание предметной области**

Предметной областью является реализация технологии «Reflective DLL Injection». первооткрывателем этой техники считается Стивен Фьюер. Технология позволяет внедрять из оперативной памяти дополнения в виде dll библиотек, которые в свою очередь улучшают или упрощают использование различных приложений. Задача заключается в загрузке библиотеки сначала в память процесса, а потом во внедрении этой библиотеки в выбранный пользователем процесс. Пользователь, должен уметь находить идентификатор уже запущенного процесса и владеть базовыми знаниями работы с компьютером. Рефлексивное внедрение отличается от обычного тем, что достаточно один раз загрузить dll библиотеку в память и использовать ее, а также не надо хранить dll библиотеку на диске.

Приложение должно иметь интерактивный интерфейс для удобного использования пользователем, а конкретнее:

1. Иметь подсказки в виде всплывающих окон, чтобы пользователь понимал, как использовать приложение;

2. Всплывающее окно об удачном или не удачном внедрении

Реализованная технология позволит пользователям внедрять различные dll библиотеки в приложения для удобства или для расширения функционала уже существующего приложения. Например: компания создает приложение и хочет чтобы другие компании или пользователи создавали дополнения и расширения, а также DLL библиотеки полезны для управления проектами, памятью, распределением ресурсов.

Технология имеет неограниченный функционал, который позволит внедрять любые dll библиотеки в процесс, а также этот метод предлагает отличный уровень скрытности по сравнению с другими методами и широко используется в Metasploit. Основным преимуществом самой загрузки библиотеки является то, что она никаким образом не регистрируется с помощью хост-систему и, как следствие, практически невозможно обнаружить как на уровне системы, так и на уровне процесса.

1. **. Анализ аналогов и прототипов**

Технология используется самыми разными людьми, от профессионалов кибербезопасности до хакеров. Внедрение происходит по этапам:

1. Некоторый исполняемый файл считывает DLL-библиотеку с диска в адресное пространство своего процесса и передает управление на ее экспортируемую функцию ReflectiveLoader.

2. Поскольку теперь библиотека существует в произвольном месте в памяти, ReflectiveLoader вычисляет текущее местоположение самой DLL-библиотеки в памяти. Для этого ReflectiveLoader получает адрес текущей инструкции и, двигаясь в обратном направлении, ищет байты 4D5A, соответствующие MZ-сигнатуре. Это нужно, чтобы дать библиотеке возможность анализировать свои собственные заголовки для дальнейшего запуска.

3. ReflectiveLoader определяет адрес библиотеки kernel32.dll в текущем процессе, после чего анализирует таблицу ее экспорта и находит функции LoadLibrary, GetProcAddress и VirtualAlloc, необходимые для дальнейшей загрузки.

4. ReflectiveLoader выделяет непрерывный участок памяти (VirtualAlloc), где и размещает код DLL-библиотеки (заголовки и секции) в соответствии с виртуальными адресами. Затем ReflectiveLoader обрабатывает свою вновь загруженную таблицу импорта: загружает необходимые библиотеки (LoadLibrary) и импортируемые из них функции (GetProcAddress).

5. ReflectiveLoader вызывает DllEntryPoint внедряемой библиотеки.

Один из аналогов рефлексивной технологии внедрения в целевой процесс считается технология Dll injection, она отличается от рефлексивного внедрения тем, что для загрузки DLL-библиотеки в Windows нужно вызывать функцию LoadLibrary, которая принимает в качестве аргумента путь к библиотеке. Для этого нужно, чтобы библиотека была на диске, с которого и производится загрузка. Данная технология имеет недостаток, в том, что нужно хранить DLL библиотеку на носителе информации, а также многие антивирусы удаляют DLL библиотеки при попытке загрузить ее в целевой процесс. Решение этих недостатков привело к созданию новой технологии, которая будет скрывать изъяны прошлой технологии, Reflective DLL Injection позволяет внедрить код DLL-библиотеки в процесс из памяти. Основное преимущество такого подхода заключается в том, что библиотека не регистрируется в системе. В результате ее практически невозможно обнаружить ни на уровне системы, ни на уровне процесса.

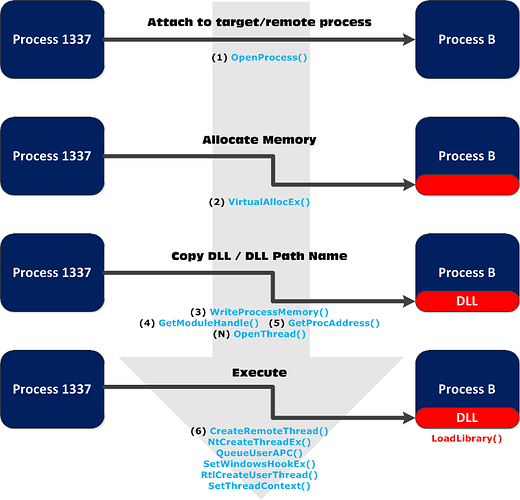


Рис. 1 – Как происходит рефлексивное внедрение в целевой процесс.

1. **. Перечень задач подлежащих решению в процессе разработки**
2. реализация технологии Reflective DLL injection с помощью WinAPI на языке С++.
3. Реализация DLL библиотеки, с вызовом MessageBox.
4. Изучение функций WinAPI для успешной реализации технологии.
5. Запись заголовков DLL в память.
6. Тестирование и проверка разработанной программы на виртуальной машине.
7. Вывод всплывающих окон с информацией о внедрении
8. **. Обоснование выбора инструментов и платформы для разработки**

Для выполнения курсовой работы были выбраны:

1. Операционная система Windows 10.

В операционной системе Windows реализована объектно-ориентированная идеология. Базовый объект системы – окно, поведение которого определяется методом, называемым функцией окна. Графический образ окна на экране дисплея – прямоугольная рабочая область. Независимо от своего типа любой объект Windows идентифицируется описателем или дескриптором (handle). Все взаимоотношения программного кода с объектом осуществляются только через его дескриптор. Интерфейс прикладного программирования (API – Application Programming Interface) представляет собой совокупность 32-битных функций (Win32 API), которые предназначены для создания приложений (программ), работающих под управлением Microsoft Windows. Функции объявлены в заголовочных файлах. Главный из них − файл windows.h, в котором содержатся ссылки на другие заголовочные файлы.

API — это аббревиатура названия Application Programming Interface (интерфейс прикладного программирования). API представляет собой совокупность функций и инструментов, позволяющих программисту создавать приложения (программы), работающие в некоторой среде.

Win32 API — это набор функций для создания программ, работающих под управлением Microsoft Windows 98, Windows NT или Windows 2000. Все функции этого набора являются 32 битными, что отражено в названии интерфейса

Причины выбора Операционной системы Windows 10:

1. Простота эксплуатации. Даже неподготовленному человеку легко разобраться в принципах работы операционной системы, ведь все интуитивно понятно, также система не требует знаний в программировании.

2. Большое количество программ сторонних компаний выпускают именно под данную операционную систему. Есть как платные, так и бесплатные версии.

3. Большая часть выпускаемого оборудования для ПК выпускается с поддержкой ОС Windows. Веб-камеры, сканеры, принтеры, игровые манипуляторы и так далее. Все драйвера, в первую очередь, выходят под данную ОС, и лишь по прошествии большого промежутка времени выходят под другие операционные системы.

4. Большая распространённость. Данная операционная система, по состоянию на апрель 2020 года, заняла второе место в общей мировой статистике использования ОС.

5. Различные формы и виды оформления. Кроме стандартных вариантов ОС позволяет сторонним программам вносить изменения во внешний вид рабочего стола, папок, заставок и так далее.

2. Среда разработки Microsoft Visual Studio 2019

Microsoft Visual Studio 2019— это набор инструментов разработки, основанных на использовании компонентов, и других технологий для создания мощных, производительных приложений. Кроме того, среда Visual Studio оптимизирована для совместного проектирования, разработки и развертывания корпоративных решений. Visual Studio предоставляет средства для проектирования, разработки b отладки.

Среда разработки Visual Studio представляет собой полный набор средств разработки для создания веб-приложений ASP.NET, XML (веб-службы), настольных приложений и мобильных приложений. Visual Basic, Visual C# и Visual C++ используют единую интегрированную среду разработки (IDE), которая позволяет совместно использовать средства и упрощает создание решений на базе нескольких языков. Кроме того, в этих языках используются функциональные возможности платформы .NET Framework, которая позволяет получить доступ к ключевым технологиям, упрощающим разработку веб-приложений ASP и XML (веб-службы).Можно создать обычные приложения Microsoft Windows и приложения с архитектурой "клиент-сервер" с помощью использования конструкторов в Visual Studio.

Почему именно эта версия продукта, потому что в нем есть такие удобные нововведения как:

1. Visual Studio IntelliCode повышает эффективность разработки программного обеспечения с помощью искусственного интеллекта (ИИ). Для создания рекомендаций IntelliCode анализирует 2000 проектов с открытым кодом на GitHub

2. Рефакторинг В C++ есть много новых удобных возможностей рефакторинга, которые помогают упорядочить код. Они отображаются как предложения со значком лампочки и включают такие действия, как перемещение элементов в интерфейс или базовый класс, настройку пространств имен в соответствии со структурой папок.

3. Язык программирования C++

C++ — компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком C — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Синтаксис C++ унаследован от языка C. Одним из принципов разработки было сохранение совместимости с C. Тем не менее C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C. Одними из отличительных особенностей и очень важных преимуществ C++ является:

1. Поддерживаются различные стили и технологии программирования, включая традиционное директивное программирование, ООП, обобщённое программирование, метапрограммирование (шаблоны, макросы).

2. Предсказуемое выполнение программ является важным достоинством для построения систем реального времени. Весь код, неявно генерируемый компилятором для реализации языковых возможностей (например, при преобразовании переменной к другому типу), определён в стандарте. Также строго определены места программы, в которых этот код выполняется. Это даёт возможность замерять или рассчитывать время реакции программы на внешнее событие.

3. Автоматический вызов деструкторов объектов при их уничтожении, причём в порядке, обратном вызову конструкторов. Это упрощает (достаточно объявить переменную) и делает более надёжным освобождение ресурсов (память, файлы, семафоры и т. п.), а также позволяет гарантированно выполнять переходы состояний программы, не обязательно связанные с освобождением ресурсов (например, запись в журнал).

4. Пользовательские функции-операторы позволяют кратко и ёмко записывать выражения над пользовательскими типами в естественной алгебраической форме.