Речь к курсовой работе

**1 слайд**

Была поставлена задача о разработке защищенного клиент-серверного приложения, основной целую которого является обмен текстовыми сообщениями между пользователями.

**2 слайд**

Исходным языком разработки является С++. Система состоит из двух частей: сервера и клиента. После запуска обе эти части ждут ввода лицензионного ключа. Если процесс аутентификации проходит успешно, то приложение запускается, иначе программа закрывается.

**3 слайд**

Система защиты приложения: два пароля на клиенте, один на сервере. Таймеры на ввод паролей. Избегание системных вызовов. *(Пример - Можно на примере создания файлов (CreateFile), записи в файл(WriteFile) и удаления файлов(DeleteFile)).* Шифрование пароля, разнос по коду участков ввода пароля и его проверки. Добавление левых функций, циклов и т.д.

**4 слайд**

Как уже упоминалось, для работы сервера нужен пароль. Он вводится с клавиатуры в саму консоль. Имеется таймер на 10 секунд.

Для работы клиента необходимо ввести 2 пароля. Один записывается в файл, другой пишется в консоль. На ввод пароля в консоль также имеется таймер. Пароли хранятся на сервере в виде хешей.

**5 слайд**

При неверном вводе пароля на сервере или истечении по времени, консоль автоматически закрывается. Для запуска необходима библиотека AddingsLib.dll. Если попытаться запустить клиент без запущенного сервера, то программа выдаст соответствующую ошибку.

**6 слайд**

На сервере отображаются подключившиеся пользователи. Также видны их сообщения. Как только заходит новый пользователь, другие пользователи, которые уже были на сервере, получат сообщение о новом подключении.

**7 слайд**

Клиент ожидает ввод двух паролей. Первый пароль записывается в файл. То, что ввёл пользователь преобразуются в символы ASCII кода. Т.е. наш пароль состоит из цифр. Потом идёт посимвольное сложение этих символов. Таким образом формируется пароль. Для запуска необходима библиотека AddingsLib.dll.

В клиентской части приложения также реализована защита от пустых полей.

**8 слайд**

В приложении используются собственная библиотека. Она находится в зашифрованном виде. При запуске программы она дешифруется. Она содержит в себе важные функции, такие как подсчёт хеша, вывод системных сообщений и отправку сообщений всем пользователям.

**9 слайд**

Также в данной работе была поставлена задача о проведении аудита безопасности программного продукта студента группы ИБ-116 Куликова И.М . Программный продукт (сервер/клиент) дизассемблировался, отлаживался и редактировался с помощью программ, IDA Pro 7.0., OllyDbg v1.1, Hexplorer v2.6.

**10 слайд**

Поиск символов, которые могут помочь в обнаружении функции аутентификации. Поиск производился по запросам «password», «pass», «key», «hash» и т.д. Была найдена функция проверки пароля.

Была обнаружена защита от приложений, которые помогают взламывать приложение (Ида и ОлиДебагер).

По всему приложению (и серверу и клиенту были разбросаны таймеры).

**11 слайд**

Далее видно использование библиотеки ultra.dll. Блок программы, где она используется возвращает значения true или false.

Также была выявлена защита целостности приложения.

После запуска, на диске С была найдена папка.

При запуске программы, в ней создаётся библиотека “ultra.dll”. В ней хранятся два числа. Попробуем изменить программу, чтобы убедиться в этом.

Да, это именно оно. Оно имеет два числа. Первое число – хеш исходной программы. Второе – хеш запущенного приложения. Если числа отличаются, то приложение не работает. Чтобы обойти эту защиту, мы создадим там файл “ultra.dll”. Запишем в него два одинаковых числа, чтобы ошибок не было. Но при запуске приложения, файл сам перезаписывается.

**12 слайд**

Была проведён анализ функции проверка пароля на сервере. Переход “jz”. Он является условным. Т.е. для того, чтобы программа принимала любой пароль, нужно заменить этот условным переход “jz” на безусловный “jmp”. Я пытался изменить этот адрес в программе Hexplorer, программа переставала работать. Мне не позволяет это сделать защита целостности.

Таким же методом была проверена функция проверки пароля и на клиенте.

**13 слайд**

В ходе выполнения этой курсовой работы было написано многопоточное клиент - серверное приложение для обмена текстовыми сообщениями. Была реализована защита от статического и динамического анализа.

Изменение структуры программы желаемого результата не дал, но зато мы смогли получены сведения о защите программного продукта «Tahc-Chat»:

* В защите от статический анализа программа имеет (шифрование пароля, разнос ввода и проверки пароля по коду, наличию левых функций, связанных с использованием памяти).
* Динамический анализ защищается (таймер, защита от дебадинга).
* Защите от патчинга реализуется (библиотекой “ultra.dll”, подсчитывающая хеш приложения).