

УДК 004.453

Использование испытательного стенда на основе средств виртуализации для проверки эффективности средств обнаружения вторжений

Строганов И.С., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»*

*Научный руководитель: Алешин В.А., к. т. н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана*

v.a.matveev@bmstu.ru

Структура испытательного стенда следующая (рис.1).

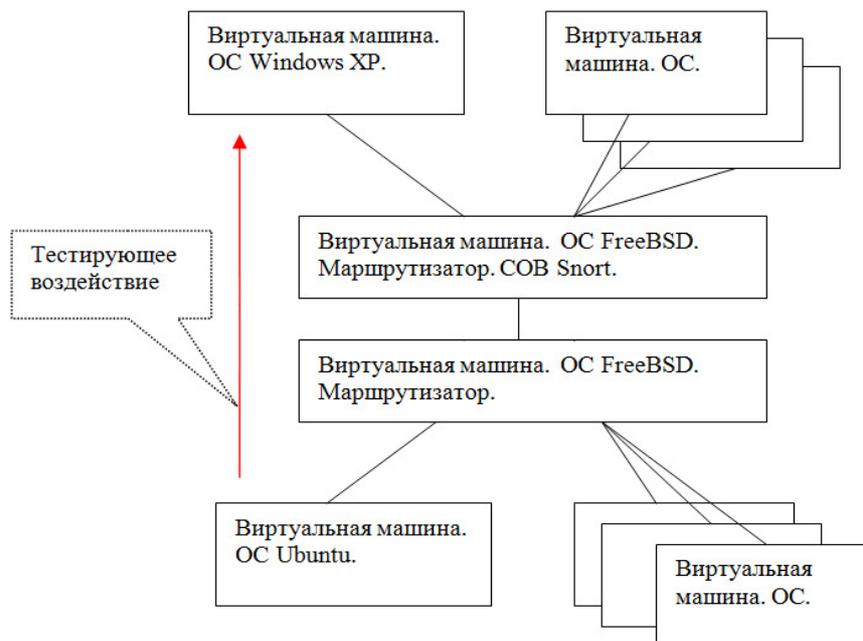


Рис.1. Структура испытательного стенда

Принцип работы.

Атака имитируется в клиент-серверном приложении, работающем по протоколу TCP IP (порт 3333), при передаче от клиента к серверу исполняемого файла *hello_world.exe*, который сохраняется и автоматически запускается на стороне сервера.

Атака происходит из узла с установленной Ubuntu (атакующая сеть) в узел с установленной Windows XP (защищаемая сеть) через два маршрутизатора с FreeBSD, на одном из которых функционирует IDS Snort в режиме обнаружения вторжений.

Приложение состоит из передатчика файла для Ubuntu (*Sender*) и приемника файла для Windows XP (*Receiver.exe*).

Приемник при запуске переходит в режим ожидания подключения.

Запуск передатчика осуществляется командой:

```
./Sender <путь к файлу> [<IP адрес получателя>]
```

При отсутствии второго аргумента командной строки используется по умолчанию IP адрес 127.0.0.1.

Запуск Snort в режиме обнаружения вторжений:

```
snort -i <прослушиваемый сетевой интерфейс> -l <директория для хранения лог файлов> -c <путь к файлу конфигурации>
```

Правила Snort записываются в файл конфигурации в следующем виде:

```
<действие> <протокол> <IP отправителя> <PORT отправителя> -> <IP получателя> <PORT получателя> (sid:<идентификатор правила>; content:"<искомое содержимое>"; msg:"<сообщение об атаке>")
```

Результаты тестирования.

Передается файл *hello_world.exe* с узла 192.168.2.5 (Ubuntu) на узел 192.168.0.5 (Windows XP).

Передатчик запускается командой:

```
./Sender hello_world.exe 192.168.0.5
```

Приемник на стороне сервера сохраняет полученный файл под именем *recv.exe* и запускает его на выполнение (рис.2).

```
C:\Users\Admin\Desktop>Receiver.exe
File size: 57344 bytes
Hello World!
C:\Users\Admin\Desktop>
```

Рис.2. Запуск исполняемого файла на стороне сервера

Исходные коды на языке С.

Исходный код программы-передатчика для Ubuntu:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

#define BUFFERSIZE 1024

int sendall(int sock, char *szBuffer, int iLength);

int main(int argc, char *argv[])
{
    if(argc!=2 && argc!=3)
    {
        printf("Usage: ./Sender <path to file> [<IP address>]\n");
        return 0;
    }
    int iCounter=0;
    char szBuffer[BUFFERSIZE];
    FILE *fl;
    int sock;
    struct sockaddr_in addr;
    if(!(fl=fopen(argv[1], "rb")))
    {
        printf("Error: file opening\n");
        return 0;
    }
    sock=socket(AF_INET, SOCK_STREAM, 0);
    if(sock<0)
    {
        printf("Error: socket creation\n");
        return 0;
    }
    addr.sin_family=AF_INET;
    addr.sin_port=htons(3333);
    if(argc==2)
        addr.sin_addr.s_addr=htonl(INADDR_LOOPBACK);
    else
        addr.sin_addr.s_addr=inet_addr(argv[2]);
    if(connect(sock, (struct sockaddr *)&addr, sizeof(addr))<0)
    {
        printf("Error: connection failed\n");
        return 0;
    }
    while(1)
```

```

    {
        if(feof(fl))
        {
            sendall(sock, szBuffer, iCounter-1);
            break;
        }
        if(iCounter>=BUFFERSIZE)
        {
            sendall(sock, szBuffer, iCounter);
            iCounter=0;
        }
        szBuffer[iCounter++]=fgetc(fl);
    }
    close(sock);
    fclose(fl);
}

int sendall(int sock, char *szBuffer, int iLength)
{
    int iBytesTotalSent=0, iBytesSent;
    while(iBytesTotalSent<iLength)
    {
        iBytesSent=send(sock, szBuffer+iBytesTotalSent, iLength-iBytesTotalSent,
0);
        if(iBytesSent==-1)
            break;
        iBytesTotalSent+=iBytesSent;
    }
    return (iBytesSent==-1 ? -1 : iBytesTotalSent);
}

```

Исходный код программы-приемника для Windows XP:

```

#include <winsock2.h>
#include <stdio.h>
#include <string.h>

#pragma comment(lib, "wsock32.lib")

#define BUFFERSIZE 1024

int main()
{
    WSADATA wsadata;
    if(WSAStartup(0x101, &wsadata))
    {
        printf("Error: initialization winsock\n");
        return 0;
    }
    SOCKET sock_listen, sock_accept;
    struct sockaddr_in addr;

```

```

char szBuffer[BUFFERSIZE];
int iBytesRead, iFileSize;
FILE *fl;
sock_listen=socket(AF_INET, SOCK_STREAM, 0);
if(sock_listen<0)
{
    printf("Error: socket creation\n");
    return 0;
}
addr.sin_family=AF_INET;
addr.sin_port=htons(3333);
addr.sin_addr.s_addr=htonl(INADDR_ANY);
if(bind(sock_listen, (struct sockaddr *)&addr, sizeof(addr))<0)
{
    printf("Error: socket binding\n");
    return 0;
}
listen(sock_listen, 1);
sock_accept=accept(sock_listen, 0, 0);
if(sock_accept<0)
{
    printf("Error: acception\n");
    return 0;
}
if(!(fl=fopen("recv.exe", "wb")))
{
    printf("Error: file opening\n");
    return 0;
}
while(1)
{
    iBytesRead=recv(sock_accept, szBuffer, BUFFERSIZE, 0);
    if(iBytesRead<=0)
        break;
    for(int i=0; i<iBytesRead; i++)
        fputc(szBuffer[i], fl);
}
iFileSize=ftell(fl);
fclose(fl);
closesocket(sock_accept);
closesocket(sock_listen);
WSACleanup();
printf("File size: %d bytes\n", iFileSize);
system("recv.exe");
}

```

Краткие выводы.

Выполнена атака (тестирование Snort) с помощью передачи файла с вредоносной программой, которая была обнаружена.

Данный механизм можно использовать для проверки функционирования других средств обнаружения вторжений.

Список литературы

1. Настройки испытательного стенда для проведения оценки эффективности функционирования средств защиты информации от НСД на основе средств виртуализации. В.А. Алешин, И.С. Строганов. Кафедра ИУ8. Студенческая весна 2013.
2. Lincoln Laboratory ID Evaluation Website, MIT, URL. <http://www.ll.mit.edu/IST/ideval/index.html> (дата обращения: 15.05.2013г.).
3. Lippmann, Richard et al. Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation. Proceedings of DARPA Information Survivability Conference & Exposition (DISCEX), Hilton Head, South Carolina, 25-27 January 2000. Los Alamitos, CA: IEEE Computer Society, 1999: Vol. 2, 12-26.