**Аналіз атаки MitM**

Атака "людина посередині" відбувається, коли зловмисник вставляє себе між двома сторонами, які комунікують (наприклад, між клієнтом і сервером або між пристроєм і маршрутизатором). У локальній мережі це часто досягається через:

1. **ARP-спуфінг**: Зловмисник підробляє ARP-повідомлення, щоб асоціювати свою MAC-адресу з IP-адресою шлюза або іншої жертви.
2. **DNS-спуфінг**: Підміна відповідей DNS для перенаправлення трафіку.
3. **Перехоплення трафіку**: Зловмисник аналізує або модифікує пакети даних.

Для виявлення такої атаки сніферу потрібно:

* Знати IP-адресу шлюза (щоб перевірити, чи не перенаправляється трафік).
* Аналізувати тип трафіку (наприклад, ARP, TCP, UDP).
* Сканувати мережу для визначення всіх активних IP-адрес і виявлення підозрілих пристроїв.

**Алгоритм роботи сніфера**

1. **Отримати IP шлюза**: Використати системні команди або бібліотеки Python.
2. **Сканувати мережу**: Використати ARP-запити для пошуку активних IP-адрес.
3. **Аналізувати трафік**: Перехоплювати пакети й перевіряти підозрілу активність (наприклад, кілька відповідей ARP для однієї IP-адреси).
4. **Визначити тип трафіку**: Використати бібліотеку scapy для аналізу пакетів.

**from** scapy**.all** **import** ARP**,** Ether**,** srp**,** sniff**,** conf

**import** socket

**import** struct

**import** os

**import** platform

**import** ctypes

# Функція для перевірки прав адміністратора

**def** is\_admin**():**

**if** platform**.**system**()** **==** "Windows"**:**

**try:**

**return** ctypes**.**windll**.**shell32**.**IsUserAnAdmin**()** **!=** 0

**except:**

**return** **False**

**elif** platform**.**system**()** **==** "Linux" **or** platform**.**system**()** **==** "Darwin"**:** # macOS

**return** os**.**getuid**()** **==** 0 # os.getuid() замінює os.geteuid()

**else:**

**return** **False**

# Функція для отримання IP шлюза

**def** get\_gateway\_ip**():**

**try:**

**if** platform**.**system**()** **==** "Windows"**:**

gateway **=** **[**l **for** l **in** os**.**popen**(**'netstat -rn | find "0.0.0.0"'**)][**0**].**split**()[**2**]**

**else:**

gateway **=** **[**l **for** l **in** os**.**popen**(**'ip route | grep default'**)][**0**].**split**()[**2**]**

**return** gateway

**except** **Exception** **as** e**:**

**print(**f"Не вдалося отримати IP шлюза: {e}"**)**

**return** **None**

# Функція для сканування мережі

**def** scan\_network**(**ip\_range**):**

**print(**f"Сканування мережі: {ip\_range}"**)**

arp **=** ARP**(**pdst**=**ip\_range**)**

ether **=** Ether**(**dst**=**"ff:ff:ff:ff:ff:ff"**)** # Broadcast MAC

packet **=** ether **/** arp

result **=** srp**(**packet**,** timeout**=**3**,** verbose**=**0**)[**0**]**

devices **=** **[]**

**for** sent**,** received **in** result**:**

devices**.**append**({**'ip'**:** received**.**psrc**,** 'mac'**:** received**.**hwsrc**})**

**return** devices

# Функція для аналізу пакетів (виявлення MitM)

**def** analyze\_packet**(**packet**):**

**if** packet**.**haslayer**(**ARP**):**

arp **=** packet**[**ARP**]**

**print(**f"ARP пакет: {arp**.**op} | Джерело: {arp**.**psrc} ({arp**.**hwsrc}) | Ціль: {arp**.**pdst}"**)**

**if** arp**.**op **==** 2**:** # ARP-відповідь

check\_arp\_spoofing**(**arp**)**

# Логіка для перевірки ARP-спуфінгу

arp\_table **=** **{}**

**def** check\_arp\_spoofing**(**arp**):**

ip **=** arp**.**psrc

mac **=** arp**.**hwsrc

**if** ip **in** arp\_table**:**

**if** arp\_table**[**ip**]** **!=** mac**:**

**print(**f"Попередження: Можливий ARP-спуфінг! IP {ip} має нову MAC-адресу: {mac} (було: {arp\_table**[**ip**]**})"**)**

**else:**

arp\_table**[**ip**]** **=** mac

# Основна функція

def main():

# Отримати IP шлюза

gateway\_ip = get\_gateway\_ip()

if not gateway\_ip:

return

print(f"IP шлюза: {gateway\_ip}")

# Визначити діапазон мережі

ip\_base = ".".join(gateway\_ip.split(".")[:-1]) + ".0/24"

# Сканувати мережу

devices = scan\_network(ip\_base)

print("Знайдені пристрої в мережі:")

for device in devices:

print(f"IP: {device['ip']}, MAC: {device['mac']}")

# Почати перехоплення трафіку

print("Починаю аналіз трафіку для виявлення MitM...")

sniff(prn=analyze\_packet, filter="arp", store=0)

if \_\_name\_\_ == "\_\_main\_\_":

# Перевірка прав адміністратора

if not is\_admin():

print("Запустіть скрипт від імені адміністратора!")

if platform.system() == "Windows":

print("Спосіб: Клацніть правою кнопкою миші -> 'Запустити від імені адміністратора'")

else:

print("Спосіб: sudo python script.py")

else:

main()

