What is a Network or Packet sniffer?

- A device or software to Inspect & Collect Network Traffic
- General network diagnostics
  - Wireshark
  - Microsoft Network Monitor 3.4
  - TCPDump
  - Commview
- Special purpose
  - Sniff passwords: Cain, Ettercap, Dsniff
  - IDS: Snort
  - Network forensics: NetworkMiner, Ettercap, P0f, Satori

Many use libpcap/WinPcap libraries
Data Acquisition

 Acquisition mean to take a forensically sound copy of digital evidence acquired at the scene

- Forensically sound
  - chain of evidence
  - digital signatures
  - Access controls.

- Complete copy
  - This can be difficult sometimes due to speed of networks.
  - Might extend beyond the scope of the warrant or authority - be careful.
  - Privacy considerations
Network Inspection Challenges

Inspection & Collection is a trade-off

Too much data
  - Difficult to manage the volumes obtained
  - Privacy and legal constraints
  - What kind of data?

Too little data
  - Might miss critical information
  - Exculpatory evidence

Level of access
  - Might not be possible to acquire data on networks outside your control.
Network Topology Challenges
Basic acquisition

Record all traffic seen at the network interface

- Most popular tool is WireShark or Etherial
  - Uses libpcap - a library which abstracts away the raw sockets facility
  - Note that libpcap is a thin wrapper around the kernel's capture facility
  - Most Unix like kernels can do packet matching in the kernel
  - Better tools operate in the Kernel to keep up
    - Gulp
    - Ruminate
**IPv4 Header**

Version

Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

Header Length

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

IHL (Header Length)

Total Length

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Type of Service (TOS)

IP Protocol ID. Including (but not limited to):
- 1 ICMP
- 2 IGMP
- 6 TCP
- 9 IGRP
- 17 UDP
- 47 GRE
- 50 ESP
- 51 AH
- 57 SKIP
- 68 EIGRP
- 89 OSPF
- 115 L2TP

Identification

IP Flags

IP Flags

<table>
<thead>
<tr>
<th>x</th>
<th>D</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0x80 reserved (evil bit)</td>
<td>D</td>
</tr>
</tbody>
</table>

Fragment Offset

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

Header Checksum

Checksum of entire IP header

Time To Live (TTL)

Source Address

Protocol

Destination Address

RFC 791

Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Copyright 2008 - Matt Baxter - mjb@fatpipe.org - www.fatpipe.org/~mjb/Drawings/
Network Card

- **Normal**
  - Only frames destined for the NIC’s MAC address, and broadcasts, are passed up the network stack

- **Promiscuous mode**
  - Lets you see traffic in your collision domain, even if it’s not destined for your MAC address
  - Some wireless card don’t support it

- **Monitor mode (RFMON)**
  - Allows raw viewing of 802.11 frames
  - Generally you have to use *nix (some exceptions)
  - `ifconfig wlan0 down`
  - `iwconfig wlan0 mode monitor`
  - `ifconfig wlan0 up`
  - Kismet
PCAP File format

Very simple file format - A file header, then a series of records one for each packet:

```c
struct pcap_file_header {
    uint32_t magic;
    uint16_t version_major;
    uint16_t version_minor;
    uint32_t thiszone; /* gmt to local correction */
    uint32_t sigfigs;  /* accuracy of timestamps */
    uint32_t snaplen; /* max length saved portion of each pkt */
    uint32_t linktype; /* data link type (LINKTYPE_*) */
}
```
PCAP File format

struct pcap_pkthdr {
    uint32_t ts_sec;     /* time stamp */
    uint32_t ts_usec;
    uint32_t caplen;    /* length of portion present */
    uint32_t len;       /* length of this packet (off wire) */
    char *data;
}

- Big endian or Little endian
- No inherent file size limitation (but some tools restrict to 2Gb due to internal 32 bit addressing).
- No internal addressing - not possible to seek to a given packet.
- No timezone specified (what is wrong with this zone?)
How do we actually acquire?

We need to acquire all Data transmitted on the wire:

Hubs: A hub is a network device which broadcasts all packets on all of its ports
  o Very easy to acquire:
    ▪ Just connect a machine to one of the ports
    ▪ Set your NIC into promiscuous mode
    ▪ Start Capturing
  o Hubs are very inefficient - every packet is visible on all ports
    ▪ Throughput is shared between all devices
  o Very hard to come by and expensive
Managed Switches

We can enable spanning on some of the ports

- This means the switch will repeat the traffic on those ports from other ports
- Better efficiency than a hub
- Pretty standard way of acquisition
- We need a managed switch with the appropriate level of access.
Dedicated acquisition device

Can build a transparent acquisition device via a bridge

```
#brctrl addbr capture
#brctrl capture addif eth0
#brctrl capture addif eth1
#ifconfig eth0 0.0.0.0
#ifconfig eth1 0.0.0.0
#ifconfig capture 1.1.1.1 netmask 255.255.255.255 up
#tcpdump -w capture.pcap -s0 -i capture
```
Dedicated acquisition device
Packet Analysis
Packet Analysis

SSL/TLS Protocol Layers:

- **Application Layer**
  - HTTP
  - FTP
  - Telnet
  - Other

- **Handshake Layer**
  - Handshake
  - Change Cipher Spec
  - Alert

- **Record Layer**
  - Record

- **Transport Layer**
  - TCP/IP
Tools: Wireshark

- Probably the most commonly used tool for inspection of packets.
- Has support for a huge number of protocols
- Unfortunately it doesn’t scale to the volumes encountered in forensic applications
  - Stores all packets in memory and rescans them each time a new query is issued

This is generally the best way to get a quick overview
The TCP protocol is stream based
- This means that a continuous stream of data is broken up into discrete packets which can be sent out of order or retransmitted
- Packet order is dictated by TCP sequence numbers
- Streams are differentiated by means of IP tuples:

  (Source IP, Source Port, Destination IP, Destination Port)

Stream reassembly is the process of joining up all the discrete packets, reordering them, removing retransmissions and producing a continuous stream.
Other points

HTTP is a complex protocol

• Its possible for the same IP address to host multiple sites.
• An IP address is not necessarily representative of the owner of the site.
• Sometimes its possible to see the HTML pages in the traffic, other times special encoding is required.
Custom code - Scapy

Sometime you need to write some custom code to look for something specific: malware communicating with a proprietary protocol, collecting statistics etc.

- Scapy is a useful python library for parsing lots of protocols
- Not very fast
- Very portable
import scapy

c = {}
for packet in scapy.PcapReader("full_dump.pcap"): try:
    if packet.flags == 2:
        dest = "%s:%s" % (packet.dst, packet.dport)
        connections[dest] = c.get(dest, 0) + 1
except AttributeError: pass

for k, v in c.items():
    print "%s = %s" % (k, v)
Time is critical for forensic Analysis

- Forensic evidence can be discredited with incorrect time information.
- Accurate timing information allows us to estimate things like bandwidth, sequence of events etc.
- Can corroborate with other information, such as building access times, CCTV monitoring etc.
- Time jitter and clock skew are fairly unique fingerprints of a specific machine.
Timezones

Pcap files always store time in UTC time.

Many tools convert this time to local time when displaying the data

- Be very careful here:
  - Local time to UTC conversion is not constant with time
  - Many tools (e.g. tcpdump) assume its constant which may give wrong results
  - It's best to always report results in the correct timezone
Where can we get time information

- PCAP file itself contains packet capture time
- TCP Timestamp option
- HTTP Date header
- Cookies can reveal a client's timestamp
Traffic Analysis

- Sometimes network traffic is encrypted
- Although we can not see the content of the data we can sometimes infer some information about it by looking at:
  - Traffic volumes
  - Timing
  - Response times
ARP Poisoning

- On the local subnet, IPs are translated to MAC addresses using ARP (Address resolution Protocol)
- ARP queries are sent and listened for, and a table of IPs to MACs is built (arp -a)
- Pulling off a MITM (Man In The Middle) attack
- If you MITM a connection, you can proxy it and sometime get around encryption
  - SSL
  - RDP
  - WPA
Passive OS Fingerprinting

- RFCs are implemented differently by different vendors
  - Different window sizes
  - Different TTL
  - Different responses to probes
  - Different DHCP requests
- Tools like P0f, Ettercap and Satori do passive OS finger printing
- NetworkMiner combines them all!! 😊
NetworkMiner Demo
Useful Links

Articles:

- Intro to Sniffers

- Cain RDP (Remote Desktop Protocol) Sniffer Parser

- Caffeinated Computer Crackers: Coffee and Confidential Computer Communications

- The Basics of Arpspoofing/Arppoisoning

- Fun with Ettercap filters
Useful Links

Protection:

- SSH Dynamic Port Forwarding

- An Introduction to Tor

- Encrypting VoIP Traffic With Zfone To Protect Against Wiretapping

- Finding Promiscuous Sniffers and ARP Poisoners on your Network with Ettercap

- DecaffeinatID: A Very Simple IDS / Log Watching App / ARPWatch For Windows
Useful Links

Tools:

- Wireshark
  http://www.wireshark.org/
- Ettercap
  http://ettercap.sourceforge.net/
- Cain
  http://www.oxid.it/cain.html
- NetworkMiner
  http://networkminer.wiki.sourceforge.net/NetworkMiner
- Firesheep
  http://codebutler.github.com/firesheep/
- Backtrack Linux
  http://www.backtrack-linux.org/downloads/