Network Forensics

Enrico RUSSO <enrico.russo@dibris.unige.it>

Outline

- Network Forensics and investigation process
- Network based evidence
- Correlation of different sources
- Collecting network-based evidence

This lecture takes its cue from

European Union Agency For Cybersecurity (ENISA) - Introduction to Network Forensics

Network Forensics

Network forensics is a sub-branch of digital forensics relating to the **monitoring** and **analysis** of **computer network traffic** for the purposes of information gathering, legal evidence, or intrusion detection.

- Shift to **network-centric computing**
  - Computing has evolved from isolated computers to network-centered systems as people rely on networks for various activities
- Importance of **following the cybertrail**
  - Digital investigators must now trace evidence across networks including the public Internet, private networks, and commercial systems to collect relevant digital evidence

Network Forensics: investigation process [1]

The investigation process **identifies communicating hosts** in terms of time, frequency, protocol, application, and data. It tries to answer **the 5W and 1H**[1]

- **Who**: identifying the source
  - determining who is behind an attack or suspicious activity by identifying **originating hosts** or **users**
- **What**: identifying the payload and activities
  - identifying exactly what **data** or **malicious payloads** are being transmitted
- **Where**: determining the destination or target
  - identifying which **users**, **systems**, or **resources** are being targeted or have been compromised

[1] These questions are identical for both digital and analogue forensic cases and are used in several other areas where information gathering and problem solving is required (cfr. the Kipling method)

Network Forensics: investigation process [2]

The investigation process **identifies communicating hosts** in terms of time, frequency, protocol, application, and data. It tries to answer **the 5W and 1H**

- **When**: establishing a timeline
  - defining the **exact timeframe** of events to reconstruct incident progression
- **Why**: understanding motivation and intent
  - determining the attacker's goal (gaining initial access, obtaining credentials or escalating privileges, collecting specific sensitive data, establishing persistence, lateral movement, …)
- **How**: techniques and methods used
  - understanding the attacker's methodology, tactics, and specific exploits

Timeline

- The 5Ws and 1H framework is always relevant and necessary at **every investigative step** (or for **homogeneous groups of steps**)
- Applying this framework systematically ensures completeness, consistency, and analytical rigor, enabling investigators to precisely **reconstruct the timeline** of an attack

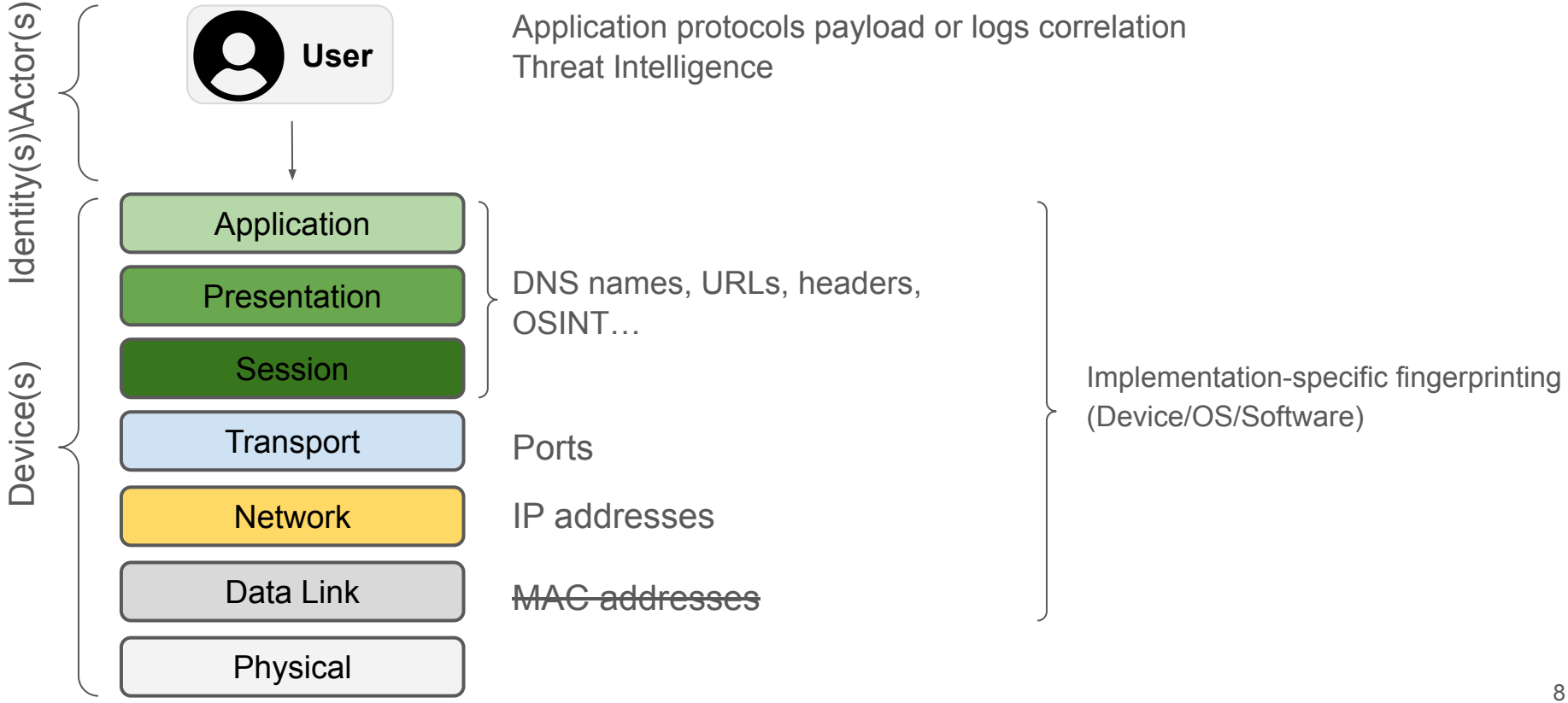
When	Who	What	Where	Why	How
2024-03-28 03:15 AM	IP: 1.2.3.4 User: Unknown	Initial unauthorized access attempt (login)	IP: 10.0.2.20 (Webserver, port 443)	Gaining initial foothold in network	Exploiting known vulnerability (RCE on web application)
2024-03-28 03:20 AM	IP: 10.0.2.20 (compromised Webserver)	Lateral movement attempt via SSH	IP: 10.0.3.15 (Internal DB Server, port 22)	Access to sensitive database data	Exploitation of SSH credentials (private key on 10.0.2.20)
…	…	…	…	…	…

Report (a possible structure)

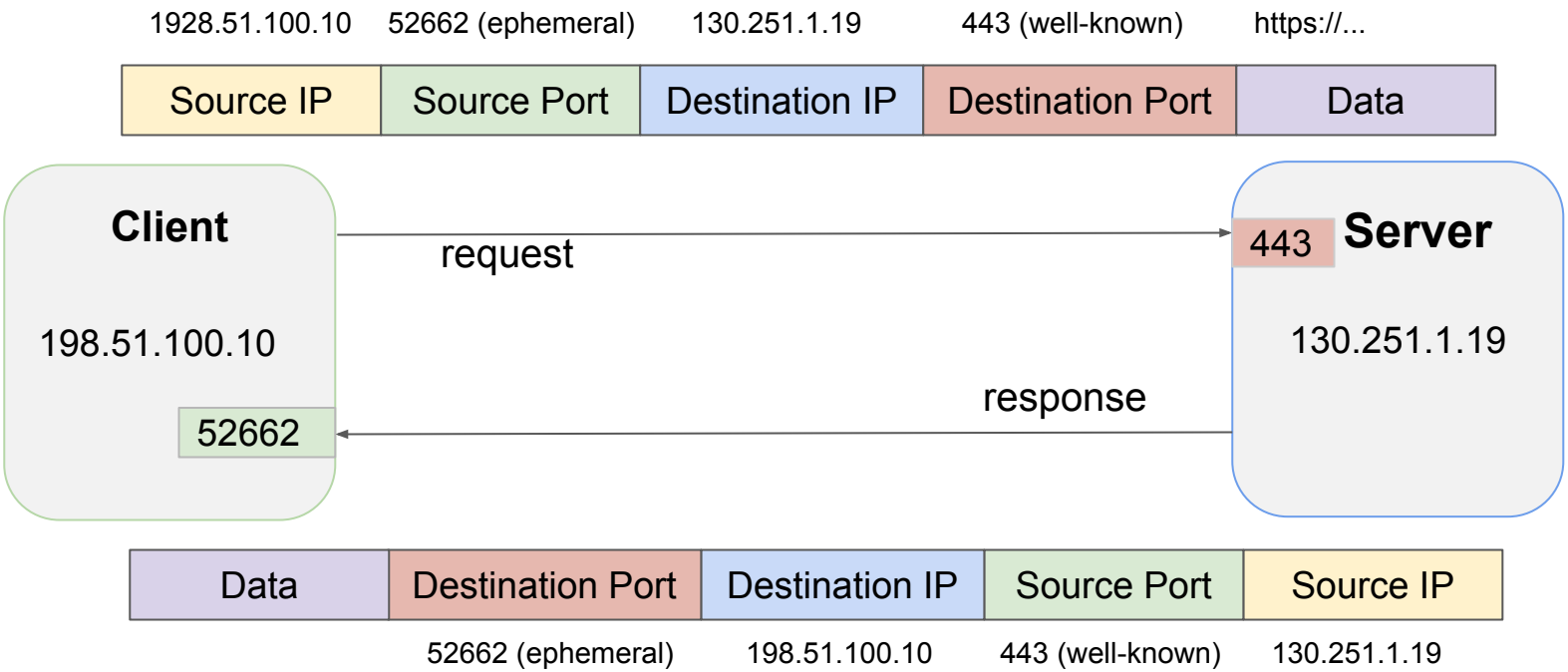
There is no single “official” format for network forensics reports

- An overview of the **investigation objectives**
- A **recap** of events
- What **digital forensics tools** (and methodologies) have been used
- Detailed **timeline**
  - Clearly identify each artifact and include its **cryptographic hash**
- **Recommendations** and **mitigations**
  - Suggest improvements to security processes based on lessons learned from the incident

Who(/Where): OSI Layers



Who/Where: the client-server model (example)



Layer 4 addressing: ports

- Layer 4 is in charge of the **process-to-process** communication. Transmitter and receiver are identified using **ports**
  - 16-bit unsigned integer (0-65535, 0 reserved) ***conventionally*** divided into:
    - **Well-known ports** (0-1023): used by **system processes** that provide widely used types of network services (requires **superuser privileges**)
    - **Registered ports** (1024-49151): **assigned** by a central authority (the Internet Assigned Numbers Authority, IANA) for specific services
    - **Ephemeral ports** (49152–65535): contains **dynamic** or **private ports** that cannot be registered with IANA

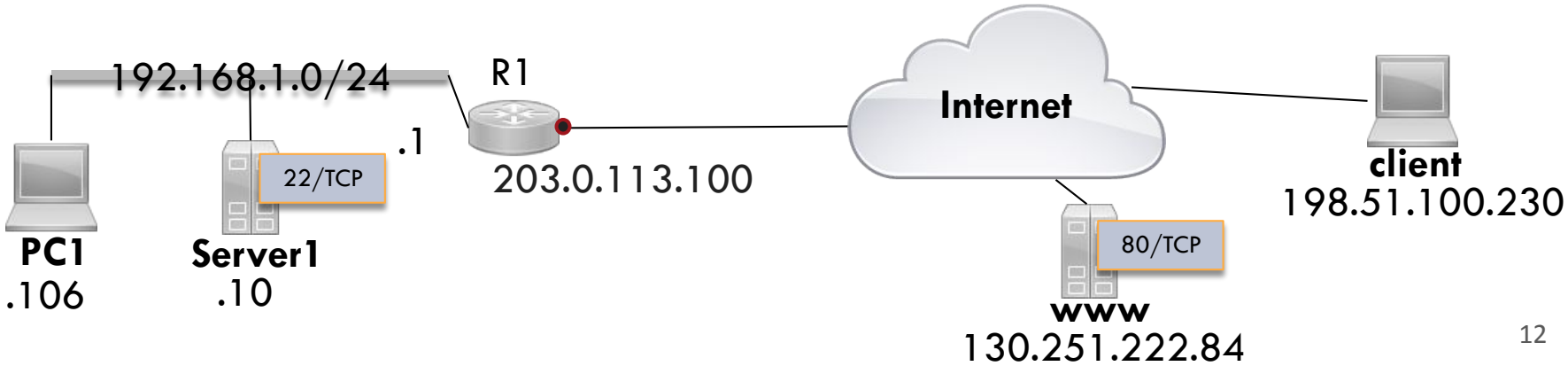
see `/etc/services`

Layer 4 addressing: ports

- The use of well-known and registered ports allows the requesting process to easily **locate** the corresponding **server application processes** on other hosts
  - For example, a web browser knows that the web server process listens on port 80/TCP
- Despite these agreements, **any service can listen on any port**
  - For example, a web server process can listen on port 8080/TCP instead of the well-known one
- Identifying services on ports:
  - Banner grabbing (e.g., nmap -sV, netcat)
  - Application payload inspection (service-specific protocol signatures)

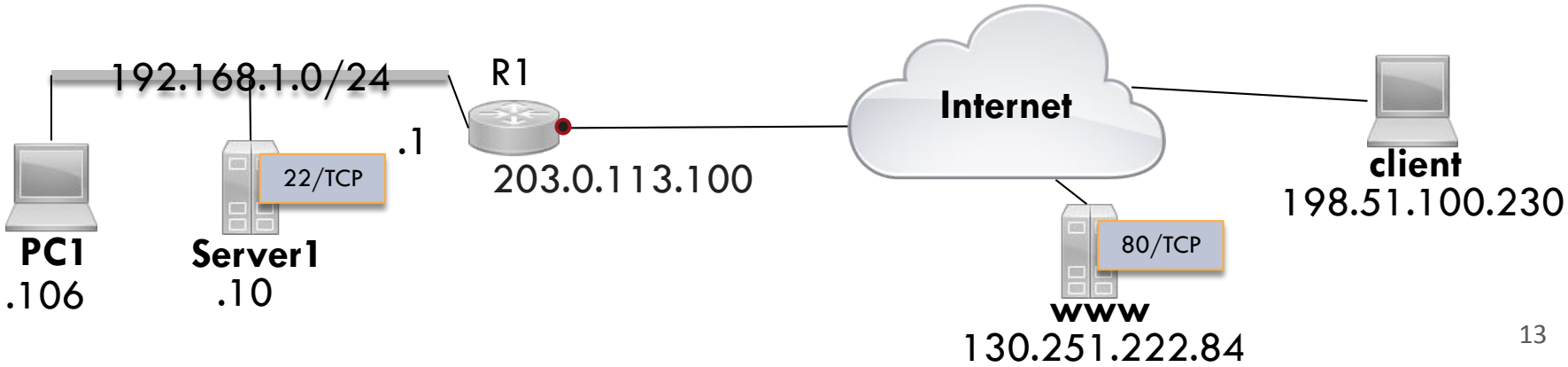
Who/Where: Network Address Translation (NAT)

- Network Address Translation (NAT) generally involves **rewriting the source** and/or destination addresses of IP packets as they pass through a router or firewall
- NAT affects answering Who/Where questions
  - 192.168.1.0/24 is a **private network** and it is **not routable** on the Internet



Source NAT and Masquerade

- Masquerade is a **source NAT** rule, i.e., it is related to the source address of a packet
- The popular usage of NAT Masquerade is to **translate a private address** range to a **single public IP address**



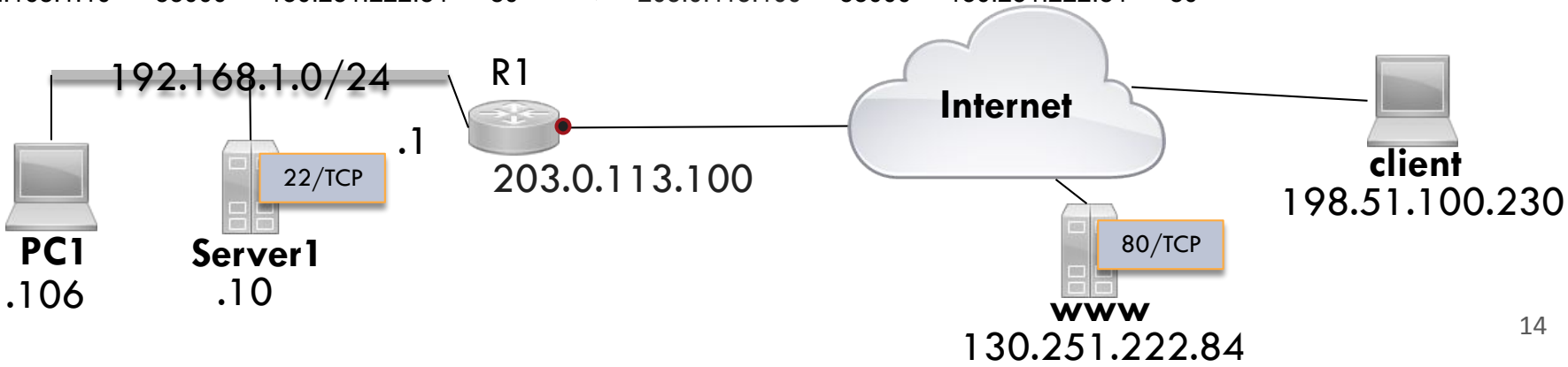
Source NAT and Masquerade (example request)

- PC1 and Server1 accessing www (request)

SNAT table (dynamic)

203.0.113.100	52000,80	192.168.1.106
203.0.113.100	53000,80	192.168.1.10

SRCIP	SRCPORT	DSTIP	DSTPORT		SRCIP	SRCPORT	DSTIP	DSTPORT
192.168.1.106	52000	130.251.222.84	80	R1 →	203.0.113.100	52000	130.251.222.84	80
192.168.1.10	53000	130.251.222.84	80	R1 →	203.0.113.100	53000	130.251.222.84	80



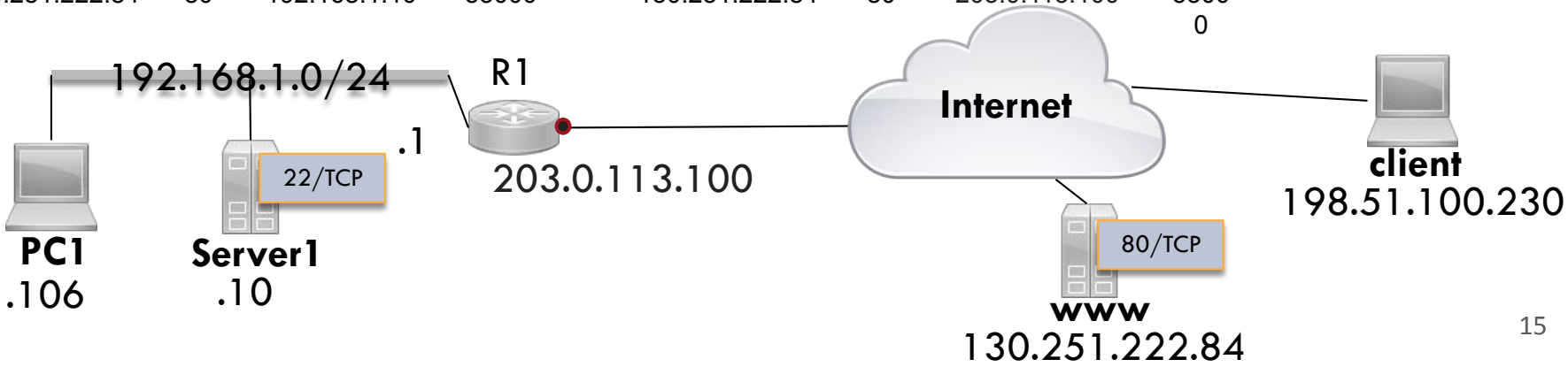
Source NAT and Masquerade (example response)

- PC1 and Server1 accessing www (response)

SNAT table (dynamic)

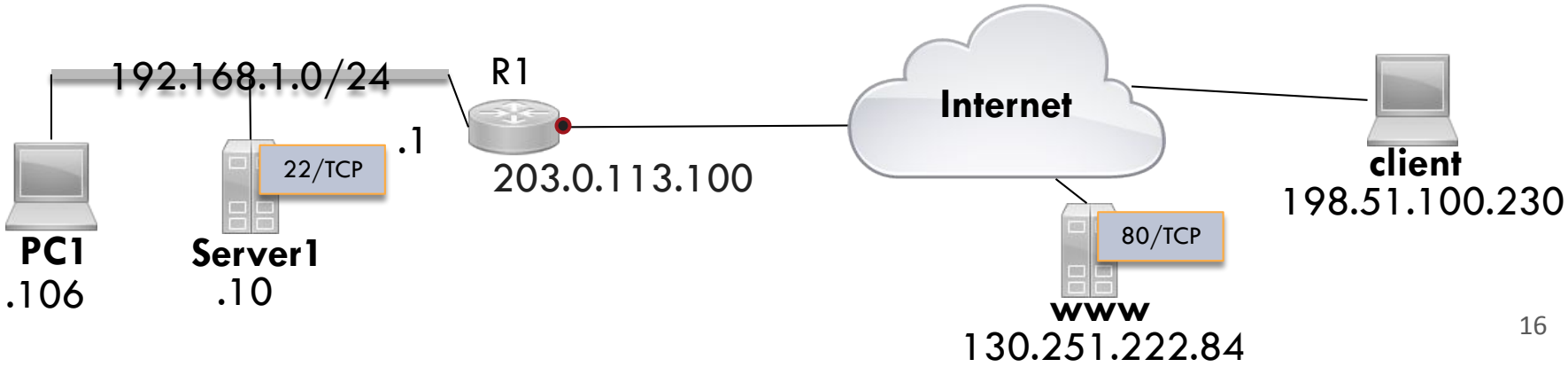
203.0.113.100	52000,80	192.168.1.106
203.0.113.100	53000,80	192.168.1.10

SRCIP	SRCPORT	DSTIP	DSTPORT		SRCIP	SRCPORT	DSTIP	DSTPORT
130.251.222.84	80	192.168.1.106	52000	R1 ←	130.251.222.84	80	203.0.113.100	52000
130.251.222.84	80	192.168.1.10	53000	R1 ←	130.251.222.84	80	203.0.113.100	53000



Port forwarding

- Port forwarding is a **destination NAT** rule, i.e., it is related to the destination address of a packet
- Maps **external IP addresses** and **ports** to **Internal IP addresses** and **ports** allowing access to internal services from the Internet



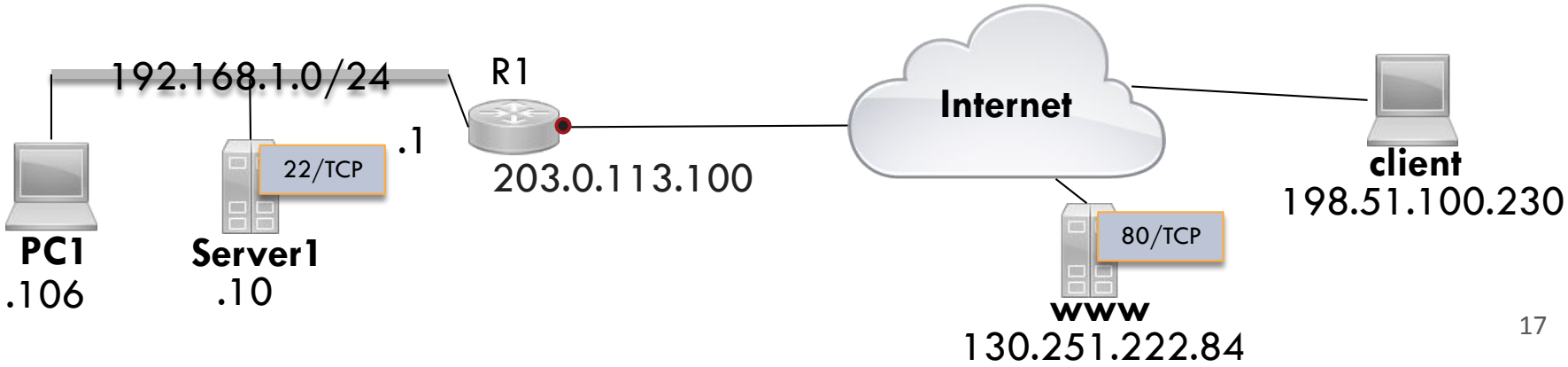
Port forwarding (example request)

- Client connecting to Server1 (request)

DNAT table (static)

Public IP address	Ext. port	Private IP address	Int. Port
203.0.113.100	2222	192.168.1.10	22

SRCIP	SRCPORT	DSTIP	DSTPORT		R1	SRCIP	SRCPORT	DSTIP	DSTPORT
198.51.100.230	54000	192.168.1.10	22	←		198.51.100.230	54000	203.0.113.100	2222



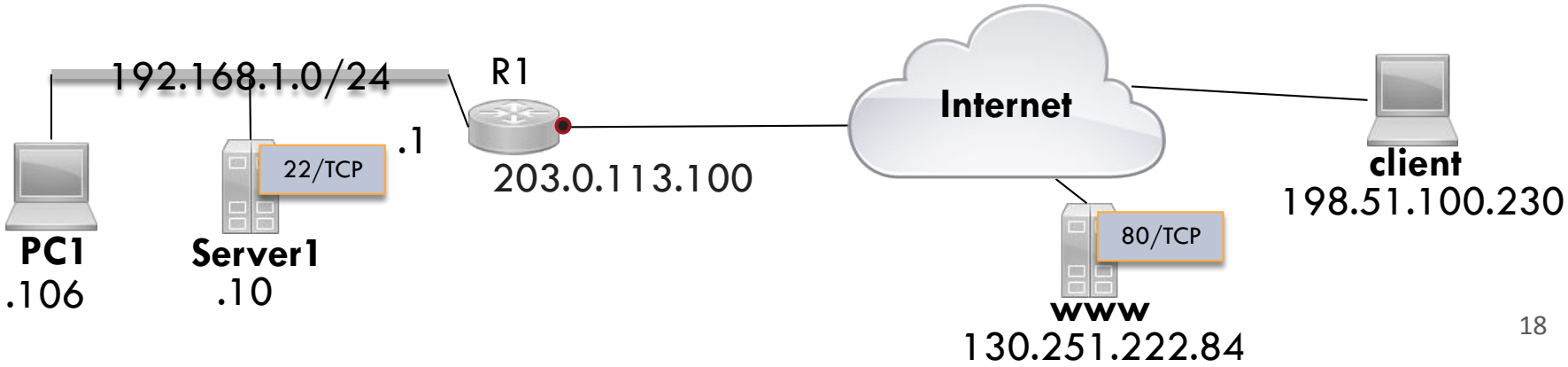
Port forwarding (example response)

- Client connecting to Server1 (response)

DNAT table (static)

Public IP address	Ext. port	Private IP address	Int. Port
203.0.113.100	22	192.168.1.10	22

SRCIP	SRCPORT	DSTIP	DSTPORT		SRCIP	SRCPORT	DSTIP	DSTPORT
192.168.1.10	22	198.51.100.230	54000	R1 →	203.0.113.100	2222	198.51.100.230	54000



Identifying internal hosts behind NAT

- NAT device translation logs (NAT table)
- correlate ephemeral ports between pre-NAT and post-NAT sessions to identify the original internal host
- application protocols:
  - FTP (File Transfer Protocol) → PORT command includes the client's IP and port for data connection
  - SIP (Session Initiation Protocol) → `Via:` and `Contact:` headers may contain the internal IP address of the user agent
  - ICMP (Error messages) → Error payload includes the original IP header
  - HTTP / Custom APIs → Sometimes apps send their local IP in headers or structured payloads (e.g., JSON payload `"client_ip": "192.168.1.10"`)
  - …

Network-based evidence: full content data

Full content data refers to every **single piece of information** that is **transmitted over a network or networks**, **without any filtering** or **modification**.

- is captured and stored in its entirety, including all the traffic that passes through the network, often referred to as **packet captures** or **PCAP**
- includes **not only the payload or data portion** of the network packets, but also the **header information**, **metadata**, **timestamps**, and **any other data** associated with the network communication

Network-based evidence: session data

Session data consists of **aggregated traffic metadata** and usually refers to the **conversation** between **two network entities**

- grouped together into **flows** and/or groups of network packets related to one another
- are able to inform the investigator about questions such as **who talked to whom** (who/where), **when**, **for how long**, etc. without looking at any contents of the conversation(s) at all.

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2020-04-22 12:19:58.824	0.004	TCP	44.30.248.239:443	->	44.244.6.114:54044	...AP...	16	141	204984	35250	410.0 M	1453	1
2020-04-22 12:22:37.845	0.004	TCP	175.68.86.47:80	->	44.244.6.114:53717	...AP...	128	133	184649	33250	369.3 M	1388	1
2020-04-22 12:20:37.844	0.004	TCP	175.68.86.47:80	->	44.244.6.114:53717	...AP...	128	133	184649	33250	369.3 M	1388	1
2020-04-22 12:24:37.845	0.004	TCP	175.68.86.47:80	->	44.244.6.114:53717	...AP...	128	132	184609	33000	369.2 M	1398	1

Cisco Netflow: https://en.wikipedia.org/wiki/NetFlow

Network-based evidence: alert data

Specific systems, typically Network Intrusion Detections System (NIDS), trigger alerts based on

- **signature-based** detection: looking for specific patterns, such as byte sequences
- **anomaly-based** detection: studies the normal behaviour of the monitored system and then looks out for any difference in it

Mainstream open source solutions are **SNORT** (https://www.snort.org/) and **Suricata** (https://suricata.io/)


```
{
    "timestamp": "2019-01-02T03:50:11.315110",
    "flow_id": 52710912,
    "in_iface": "eth0",
    "event_type": "alert",
    "src_ip": "8.42.77.171",
    "src_port": 49678,
    "dest_ip": "138.68.3.71",
    "dest_port": 22,
    "proto": "TCP",
    "alert": {
      "action": "allowed",
      "gid": 1,
      "signature_id": 2001219,
      "rev": 19,
      "signature": "ET SCAN Potential SSH Scan",
      "category": "Attempted Information Leak",
      "severity": 2
    }
```

Network-based evidence: statistical data

Statistical data provide the analyst with network-related aspects such as

- the number of bytes contained in a packet trace
- start and end times of network conversations
- number of services and protocols being used
- most active network nodes
- least active network nodes
- outliers in network usage
- average packet size, average packet rate
- …

It can therefore also act as a useful source for anomaly detection.

Correlation of different sources [1]

Network-based evidences can be correlated with other sources for enabling forensics investigations and root cause analyses:

- network nodes (e.g., attack computer, intermediate computers, or victim computer): logs, files, …
  - journalctl -u sshd (ssh server)

    Apr 15 13:45:29 server1 sshd[3859975]: Accepted publickey for **user** from **10.187.10.221** port **60149** ssh2: RSA SHA256:lGl07jSE1e>

  - tail -f /var/log/squid.log (proxy server)

    1681571063.731 171297 **93.51.10.270** TCP_TUNNEL/200 6889 CONNECT **www.google.com:443 enricorusso** HIER_DIRECT/172.217.4.196 -

  - tail -f /var/log/mail.log (mail server)

    Jul 4 19:47:58 mammon postfix/smtpd[4936]: 07A1753F: client=**c-69-181-123-456.hsd1.ca.comcast.net**[**69.181.123.456**], sasl_method=PLAIN, sasl_username=**mgorven**

Correlation of different sources [3]

- Internetworking devices (e.g., router, access point, or VPN concentrators): logs and buffers

```
Log Buffer (4096 bytes):

05:32:23: NAT: s=10.10.50.4->172.16.11.70, d=172.16.11.7 [70]

05:32:23: NAT*: s=172.16.11.7, d=172.16.11.70->10.10.50.4 [70]

05:32:25: NAT*: s=10.10.50.4->172.16.11.70, d=172.16.11.7 [71]

05:32:25: NAT*: s=172.16.11.7, d=172.16.11.70->10.10.50.4 [71]

05:32:27: NAT*: s=10.10.50.4->172.16.11.70, d=172.16.11.7 [72]
```

Who: IP and domain OSINT

Investigate suspicious IP addresses and domains using publicly available sources.

Use cases:

- Check if an IP or domain has been reported for malicious activity
- Identify infrastructure related to known threat actors
- Gather context (geolocation, ISP, reverse DNS, historical records)

VirusTotal
- Multi-engine scanner for IPs, domains, files
- Shows passive DNS, related indicators, threat labels

AbuseIPDB
- Community-powered IP reputation
- View abuse reports, threat categories, and risk scores

Shodan
- Search engine for internet-exposed devices
- Discover open ports, banners, services running on an IP

What/How: TTPs

Tactics, Techniques, and Procedures (TTPs) describe the behavioral patterns of threat actors.

The MITRE ATT&CK framework provides a structured knowledge base of real-world TTPs observed across threat campaigns.

- **Tactic (What)**: the attacker's goal or objective at a certain stage (e.g., Credential Access, Lateral Movement)
- **Technique/Procedure (How)**: the method or implementation used to achieve that goal (e.g., Brute Force - T1110, Exploitation of Remote Services - T1210)

How: CWE and CVE

- **Common Weakness Enumeration** (CWE) represents a general weakness (e.g., improper input validation)
  - a community-developed knowledge base of common software and hardware weaknesses, maintained by MITRE
  - If you only identify a CWE, you still understand what **kind of mistake** allowed the attack
- **Common Vulnerabilities and Exposures** (CVE) identifies a specific vulnerability in a real product (e.g., a buffer overflow in OpenSSL version X.Y)
  - If you know the CVE, you know exactly **which hole** the attacker used

Searching for CVEs

- NVD (National Vulnerability Database)
- MITRE CVE site
- Exploit DB
- [Google/GitHub search + product/version or artifacts]

Collecting network-based evidence

The task of acquiring network evidence can be divided into **active** and **passive** acquisition.

- **passive acquisition**: refers to gathering data **without emitting data** at OSI Layer 2 or above, such as capturing or sniffing network traffic.
- **active acquisition**: involves **interacting with systems** on the network, such as sending queries or logging to a central host, SIEM, or management station, and may also include scanning network ports to determine system status

To **preserve** as much of the evidence as possible, acquisition should not **change the packets**, send out **additional packets** or **alter the network configuration**.

Passive acquisition

Network forensic investigators can passively acquire network traffic by

- intercepting it as it is sent **across cables**
- through **network equipment** such as (hubs) and switches
- through the **air**

Inline network taps

OSI **layer 1** devices which can be inserted inline between two physically connected network devices.

- it forwards the packets and create **physical duplicates** that are sent to one or more monitoring ports.
- four ports
  - two connected inline to **facilitate normal traffic**
  - two sniffing ports, which **mirror that traffic** (one for each direction)
- insertion of an inline network tap typically causes a **brief disruption**, since the cable must be separated to connect the network tap inline
- they are commonly designed to require **no power** for passively passing packets

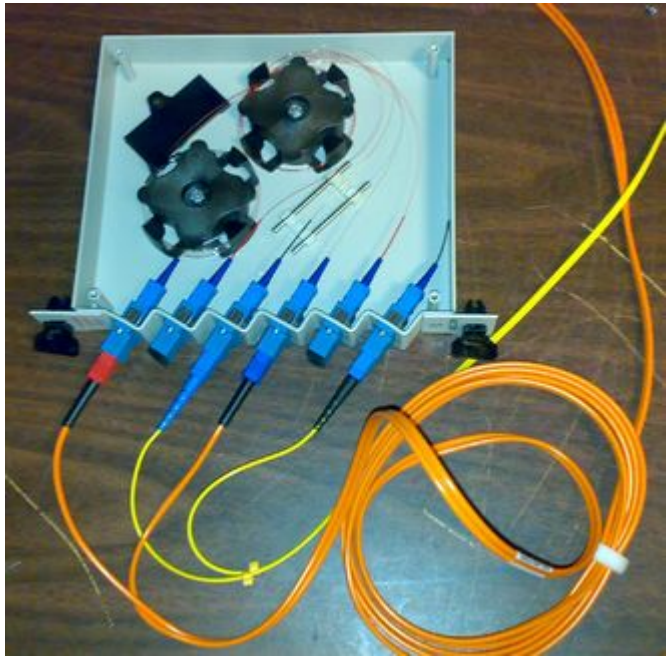
Inline network taps: Gigabit Ethernet

Gigabit copper links **cannot be passively tapped** because, in 1000BASE-T links, all copper pairs are used for transmitting and receiving data simultaneously.

- the stations on each end of the cable are able to **distinguish** incoming voltages from outgoing because **they know what they've sent**
- an **intermediary** on the wire (the tap) cannot make that distinction, so the voltages observed by a 3rd party are just **undecipherable noise**
- 1000BASE-T taps are never passive devices: they're more like a **small switch** with mirror functions enabled

Fiber Optic network taps

- Fiber tapping uses a network tap method that **extracts signal** from an **optical fiber** without breaking the connection
- It uses a **beam splitter**: an optical device that **splits a beam** of light into a transmitted and a reflected beam
- Tapping gigabit copper links is not possible without a **temporary disruption** and **signal attenuation**
- More complex taps may require a **power supply** for signal amplification



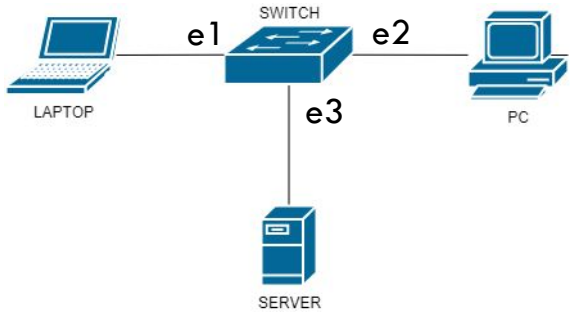
Acquisition: bridges and switches

Devices providing interconnectivity at Layer2 are called *(Transparent) Bridges or Switches*

- analyze all frames received, find the **destination MAC address**, and **forward** to the appropriate port
- to determine where to forward the traffic uses a special table (**MAC address table**)

A basic switched network

- A switch device provides connection to a number of common devices
- Let's assume that all of the devices are powered on but have not sent any traffic
- In this case, the MAC address table of the switch would be empty

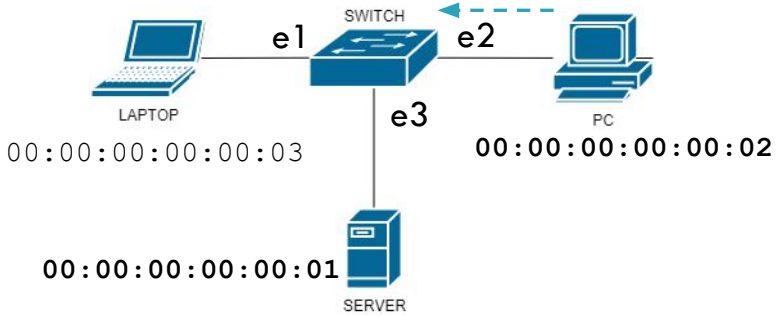


MAC address table (switch)

MAC address	Port

A basic switched network

- PC wants to send traffic to SERVER that has MAC address `00:00:00:00:00:01`
  - Creates a frame containing `00:00:00:00:00:02` as the source address and `00:00:00:00:00:01` as the destination address
  - Sends it off toward the switch

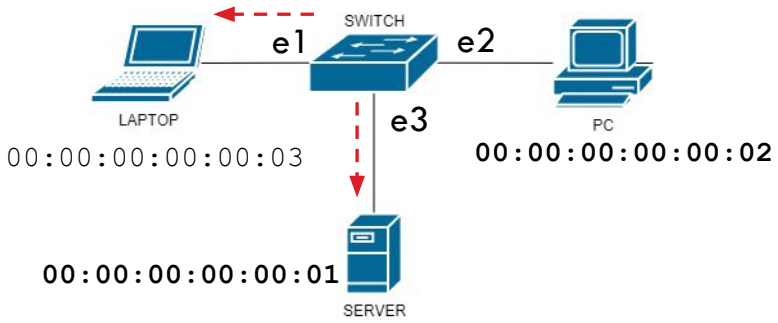


MAC address table (switch)

MAC address	Port

A basic switched network

- The switch receives the traffic
  - Creates a new entry in its MAC address table for PC MAC address (PC → e2)
  - Performs a lookup on its MAC address table to determine whether it knows which port to send the traffic to
  - Since no matching entries exist in the switch's tables, it would **flood** the frame out all of its interfaces except the receiving port (**broadcast**)

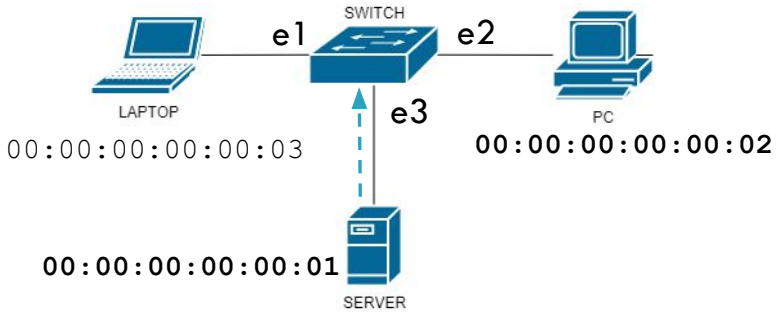


MAC address table (switch)

MAC address	Port
00:00:00:00:00:02	e2

A basic switched network

- The broadcast forwards the frame also to the target server
- (Assuming that the server wants to respond to PC) It sends a new frame back toward the switch containing `00:00:00:00:00:01` as the source address and `00:00:00:00:00:02` as the destination address
- The switch would receive the frame and create a new entry in its MAC address table for the Server MAC address (Server → e3)

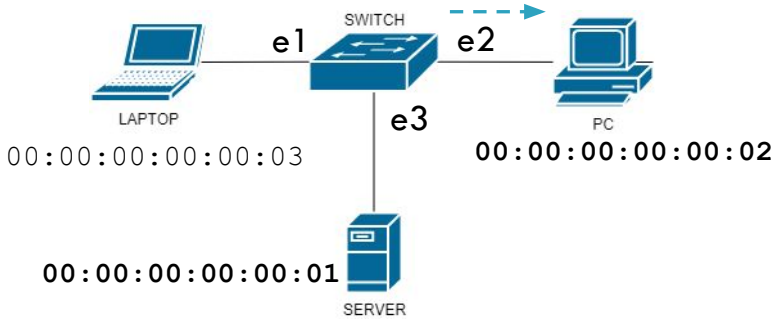


MAC address table (switch)

MAC address	Port
00:00:00:00:00:02	e2
00:00:00:00:00:01	e3

A basic switched network

- Switch performs a lookup of its MAC address table to determine whether it knows which port to send the server frame to
- In this case, it does, so it sends the return traffic out only its e2 port (PC), without flooding



MAC address table (switch)

MAC address	Port
00:00:00:00:00:02	e2
00:00:00:00:00:01	e3

Passive acquisition: port mirroring

Switches can often be configured to **replicate traffic** from one or more ports to some other port for aggregation and analysis

- the most vendor-neutral term for this is **port mirroring**
- investigators will need **administrative access** to the switch's operating system to configure port mirroring
- a **monitoring station** needs to be connected to the mirroring port to capture the traffic

WLAN: passive acquisition

To capture WLAN traffic, investigators need an 802.11[1] wireless card capable of running in **Monitor mode** (many WLAN cards do not support this mode)

- packets are captured **without associating** to an access point
- traffic from (and to) **all access points and stations in radio range** will be captured, independent of SSID
- in monitor mode the packets are captured in **802.11 datalink format**

[1] the set of standards that define communication for wireless LANs

WLAN: active acquisition

Active acquisition requires running the wireless adapter in Promiscuous mode (we have a shared medium)

- packets are captured **after associating with an access point** and the system will be listening to all packets, even those not addressed to it
- “all packets” in promiscuous mode means packets from all stations being **associated with the AP**
- in promiscuous packets are presented in Ethernet (**802.3**)

WLAN: encryption

Typically in a WLAN the traffic is encrypted

- If the WLAN use a single shared key (PSK) for all stations (WPA Personal)
  - the 4-way handshake is the process of exchanging 4 messages between an access point (authenticator) and the client device (supplicant) to generate some encryption keys derived from PSK which can be used to encrypt actual data sent over Wireless medium
  - PSK can be brute forced from a traffic capture (offline with a passive acquisition) containing a 4-way handshake (EAPOL frames)
- If the WLAN use personal credential with an authentication server (WPA Enterprise):
  - only online brute force is possible