

# Lab 9

# Three Goals

---

- **Implementing** a simple traceroute
  - Get familiar with creating packets
- **Analyzing** TLS Traffic
- **Identifying** Traffic *after* an incident

# traceroute

---

```
./traceroute <IP_ADDRESS>
```

Need to set both IP and ICMP headers

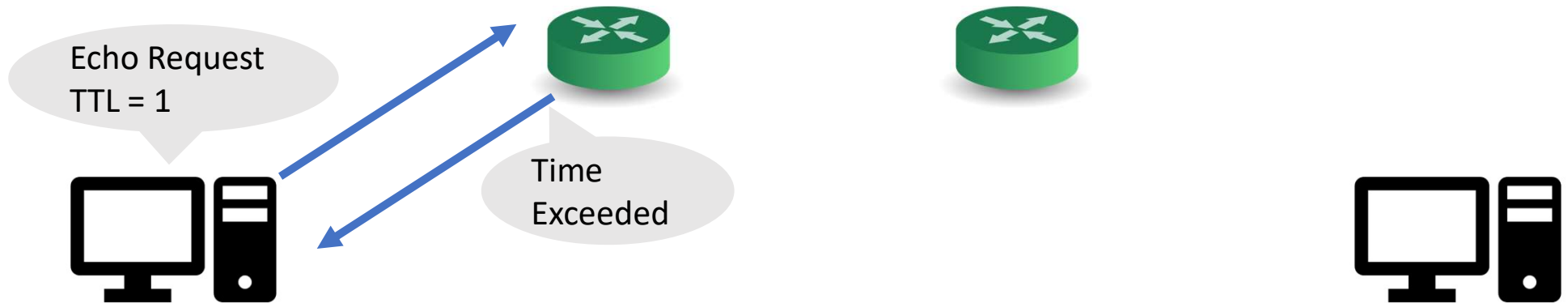
- Main IP fields
  - Dst IP address
  - TTL
- Main ICMP types:
  - Request
  - Reply
  - Time Exceeded

Internet Control Message Protocol (ICMP)					
Offsets	Octet	0	1	2	3
Octet	Bit	0–7	8–15	16–23	24–31
0	0	Type	Code	Checksum	
4+	32+	Variable			

# traceroute

---

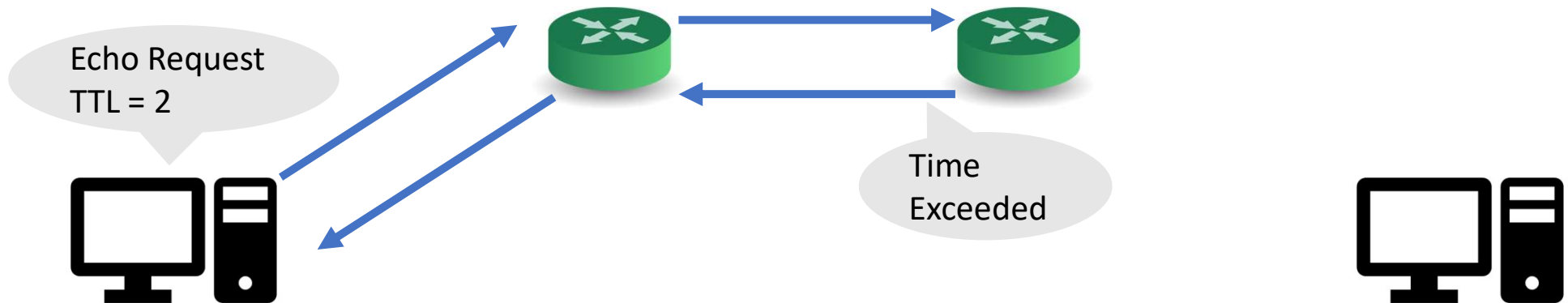
Build a path of routers from source to destination. How?



# traceroute

---

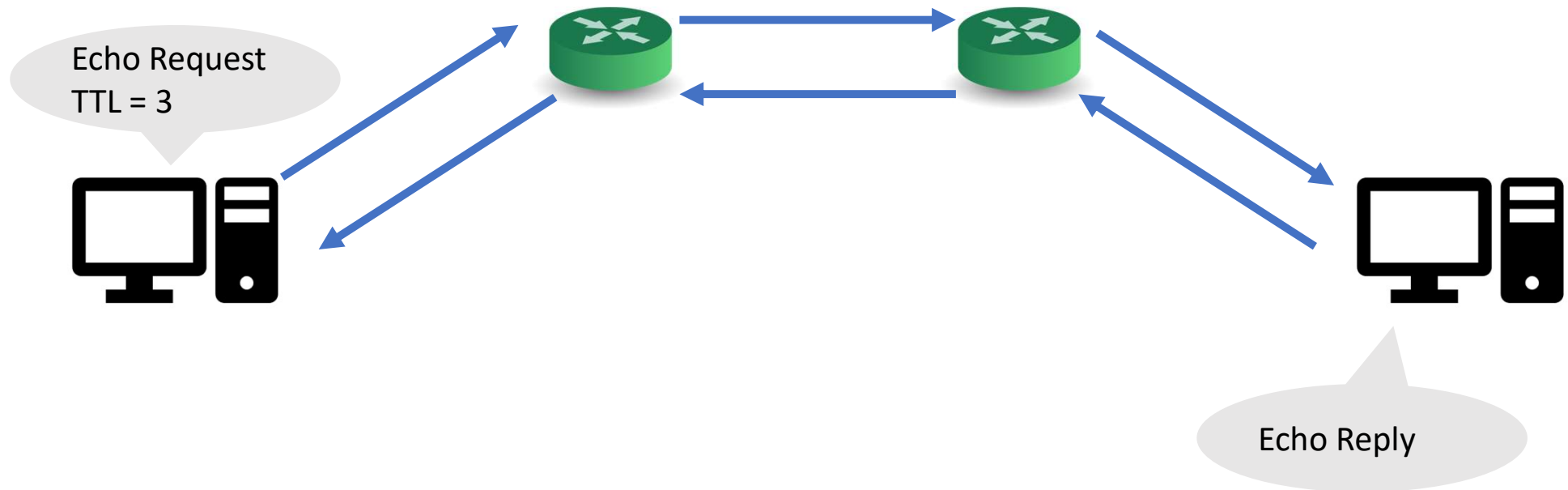
Build a path of routers from source to destination. How?



# traceroute

---

Build a path of routers from source to destination. How?



# scapy APIs

---

- Rich library (useful for spoofing, analysis, tooling, etc.)
- The scapy.all module:
- Functions:
  - sr: send and receive multiple pkts
  - sr1: send pkts and receive the first one!
- Classes:
  - IP and ICMP: corresponding protocol headers
- Basic library usage:  
<https://scapy.readthedocs.io/en/latest/usage.html>

# scapy APIs: Example

---

```
from scapy.all import IP, TCP, sr1
# Alternative: from scapy.all import *
# May collide with your classes/functions

pkt = IP(dst='1.1.1.1')/TCP(dport=80)/'PAYLOAD'
rep = sr1(pkt)
rep.summary()
```

1. **Construct** a TCP packet:
  - IP and TCP headers
  - Payload
2. **Send** the packet using `sr1()`
3. **Print** a summary of the reply packet (if any)



# Analyzing TLS Traffic

---

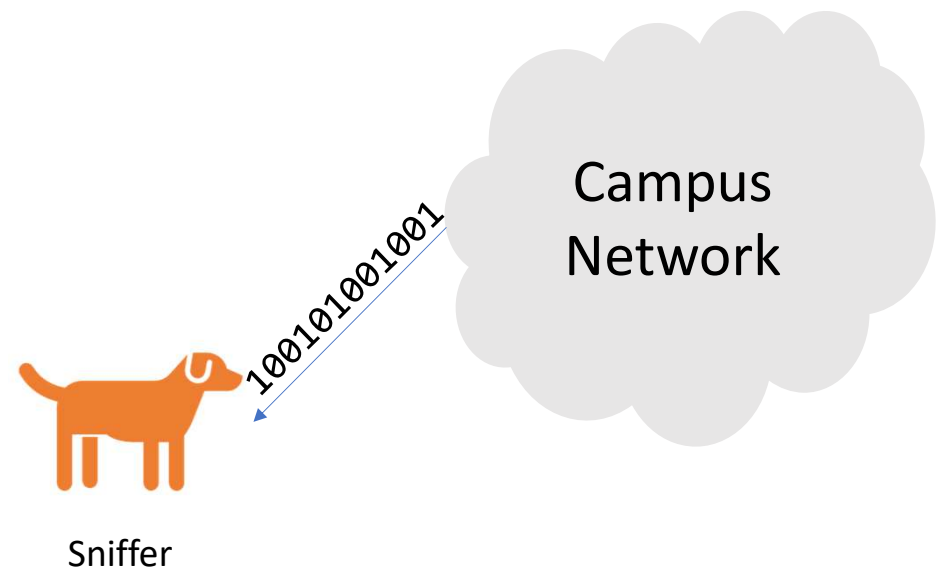
- Eavesdropping HTTPS traffic
- Analyzing HTTPS traffic in detail
  - TLS version?
  - Public encryption scheme?
  - Signature scheme?
  - Hash?
  - etc



# Identifying Traffic

---

- A given harassment scenario
- You need to:
  - analyze the traffic
  - find the harasser
  - provide enough evidence



# Questions?

---