Networking MP Checkpoint 2

Zane Ma

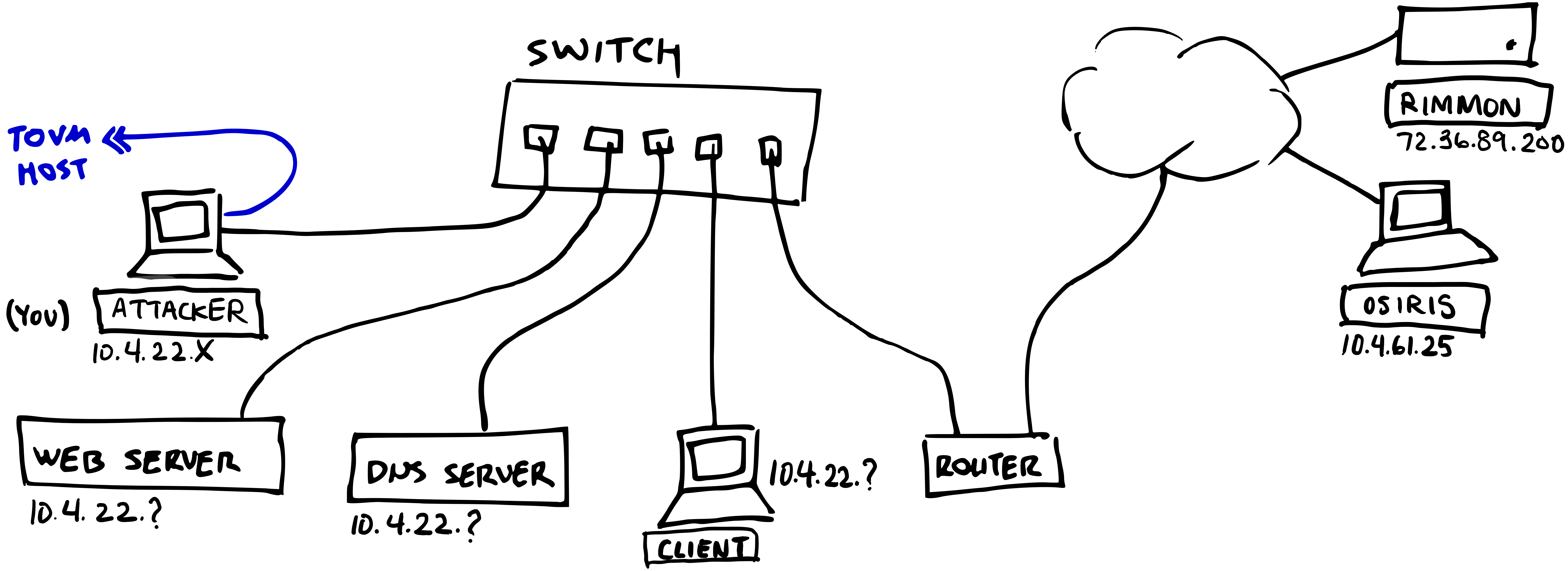
University of Illinois

CS 461 / ECE 422 - Spring 2020

Educational Objectives

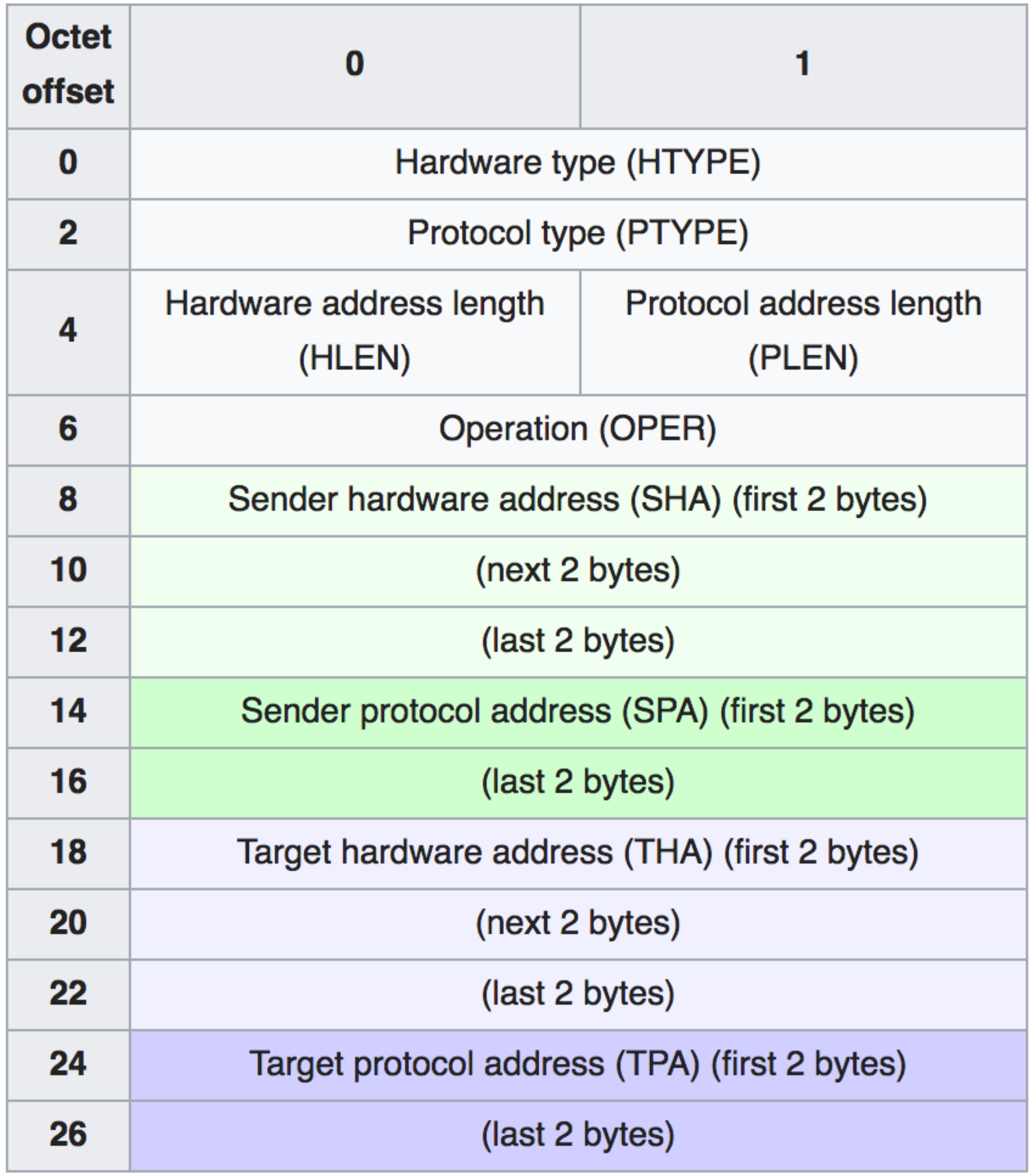
- Review ARP packets and protocol in Wireshark
- Examine local ARP cache
- Understand challenges of performing man-in-the-middle on UDP/DNS and TCP/HTTP
- Describe Mitnick attack and MP variation
- Demonstrate working examples for each checkpoint in Wireshark

MP4 Network Setup



- How to map IP to MAC address?

Address Resolution Protocol



Octet offset	0	1
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

- HTYPE/PTYPE = Layer 2/3 protocol
- OPER = Request (1) or Reply (2)
- SHA/SPA = Sender Layer 2 address/ Sender Layer 3 address
- THA/TPA = Target Layer 2 address/ Target Layer 3 address
- *What headers would ARP packet have? Layer 3? Layer 2?*

Address Resolution Protocol

- Scapy + Wireshark Demo of ARP request + arp cache

Address Resolution Protocol

- Scapy + Wireshark Demo of ARP request + arp cache

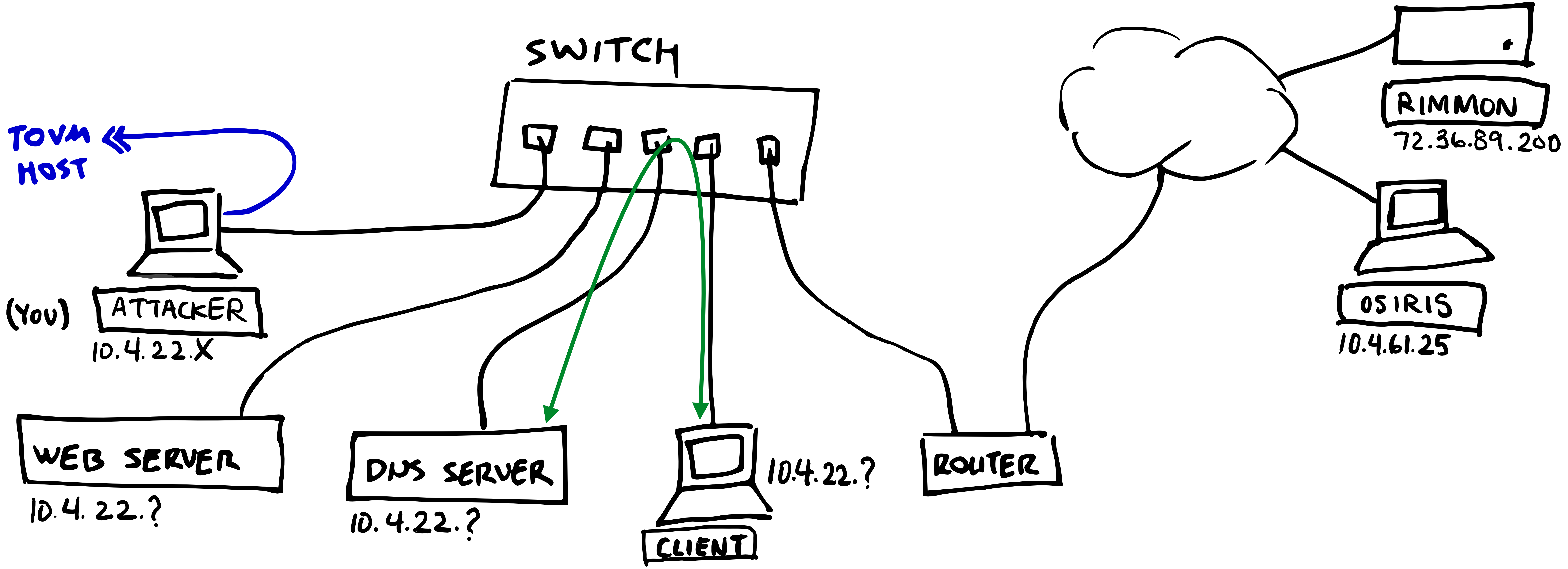
- Any security? How to poison?

Address Resolution Protocol

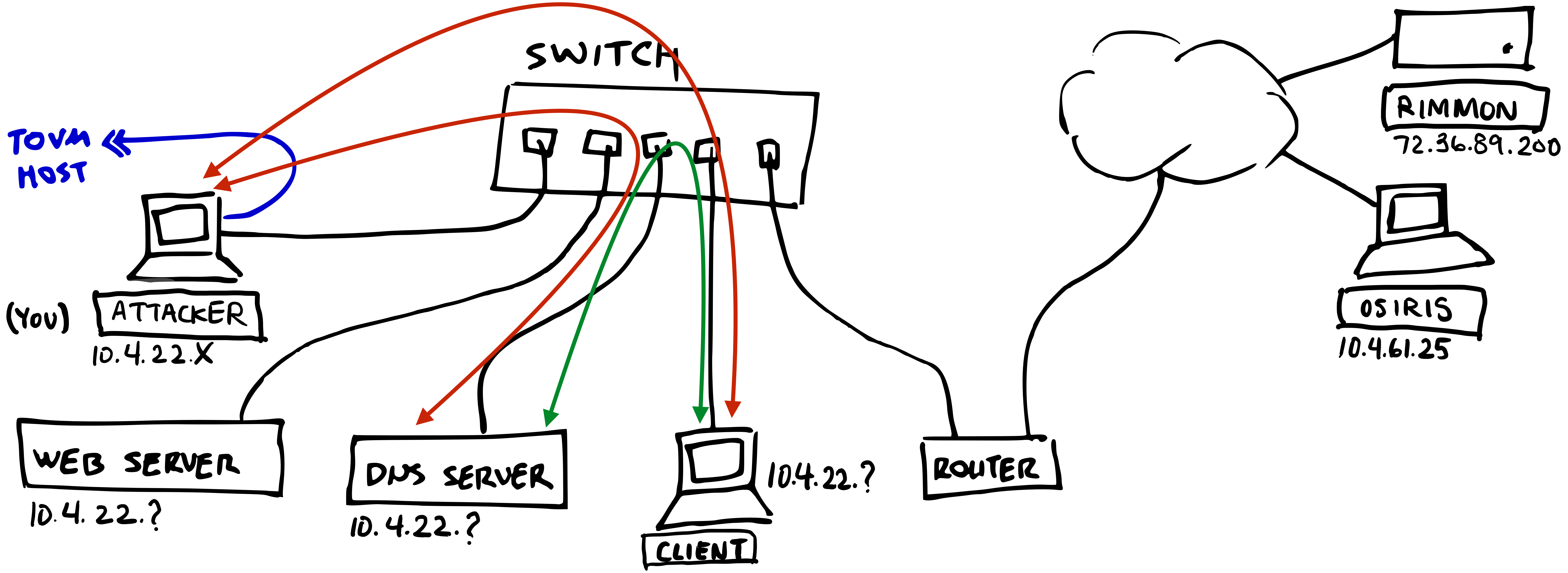
- Scapy + Wireshark Demo of ARP request + arp cache

- Any security? How to poison?
  - passive: wait for request, flood response
  - active: gratuitous ARP

MITM Packet Flow



MITM Packet Flow

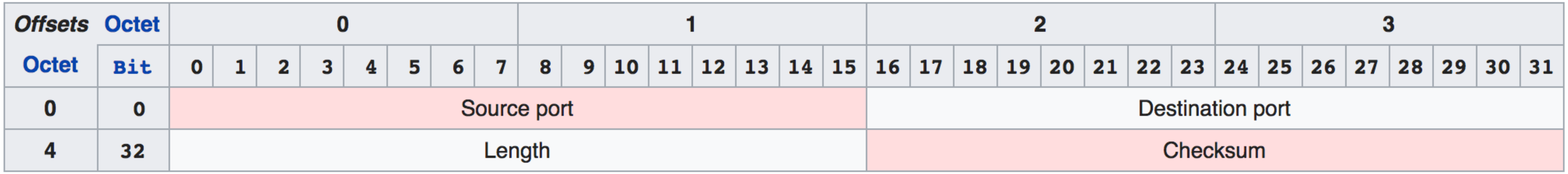


Passive Interception

- Demo passive interception

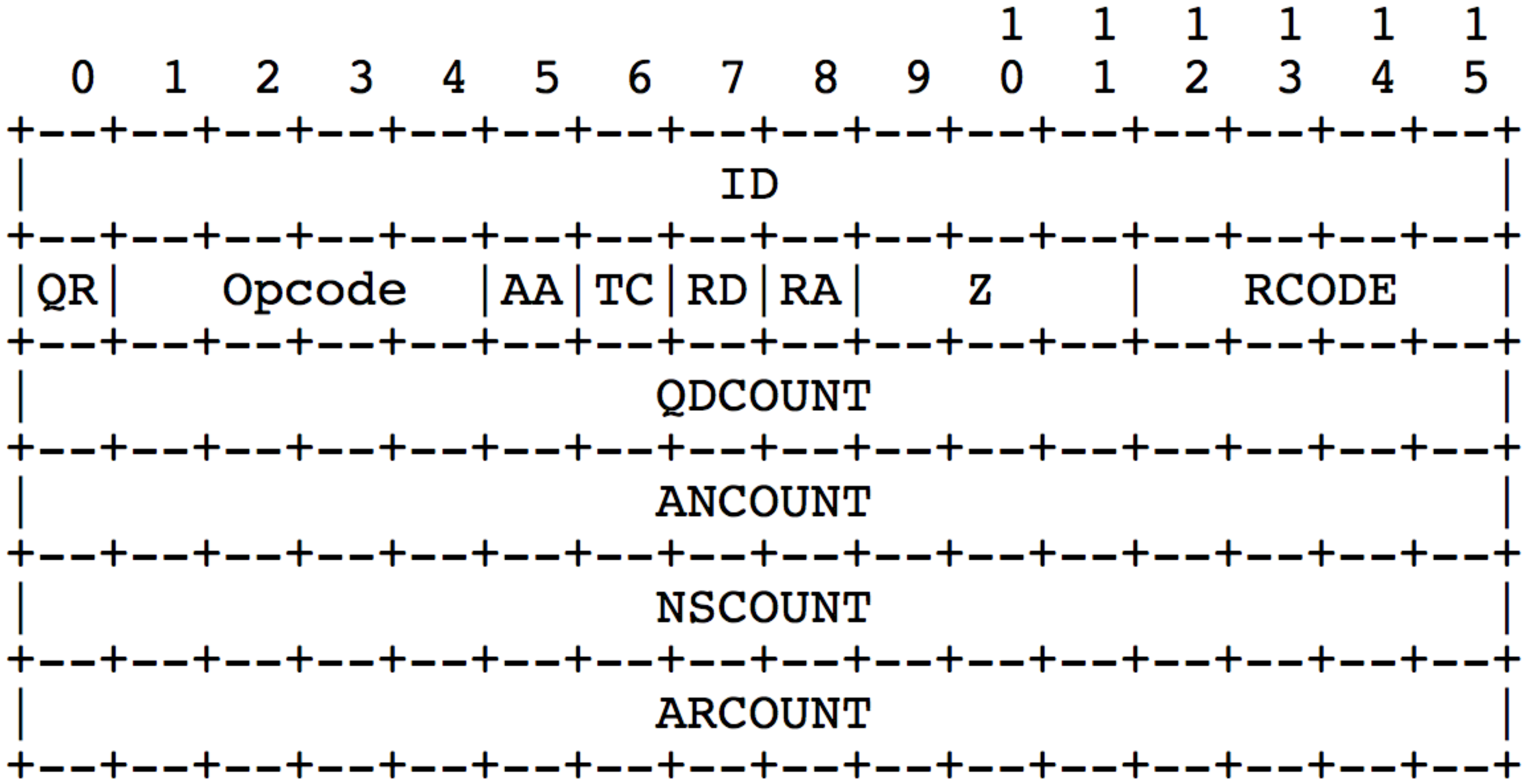
UDP/DNS Interception

UDP Header



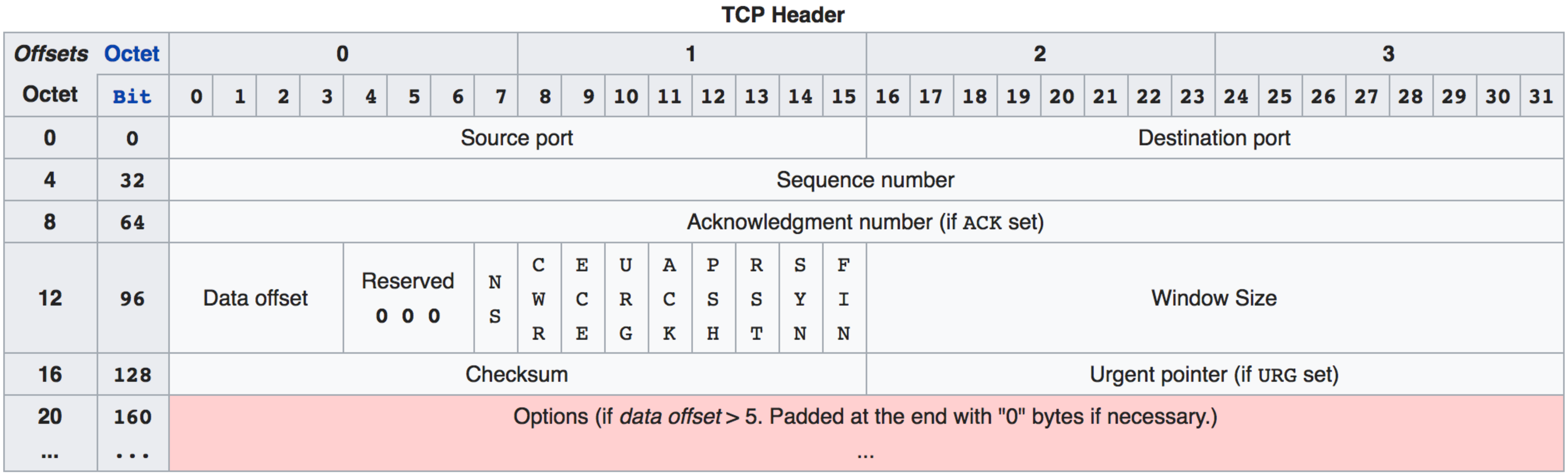
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

DNS header



```
                                    1  1  1  1  1  1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                      ID                       |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |QR|   Opcode  |AA|TC|RD|RA|   Z    |   RCODE   |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                    QDCOUNT                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                    ANCOUNT                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                    NSCOUNT                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                    ARCOUNT                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

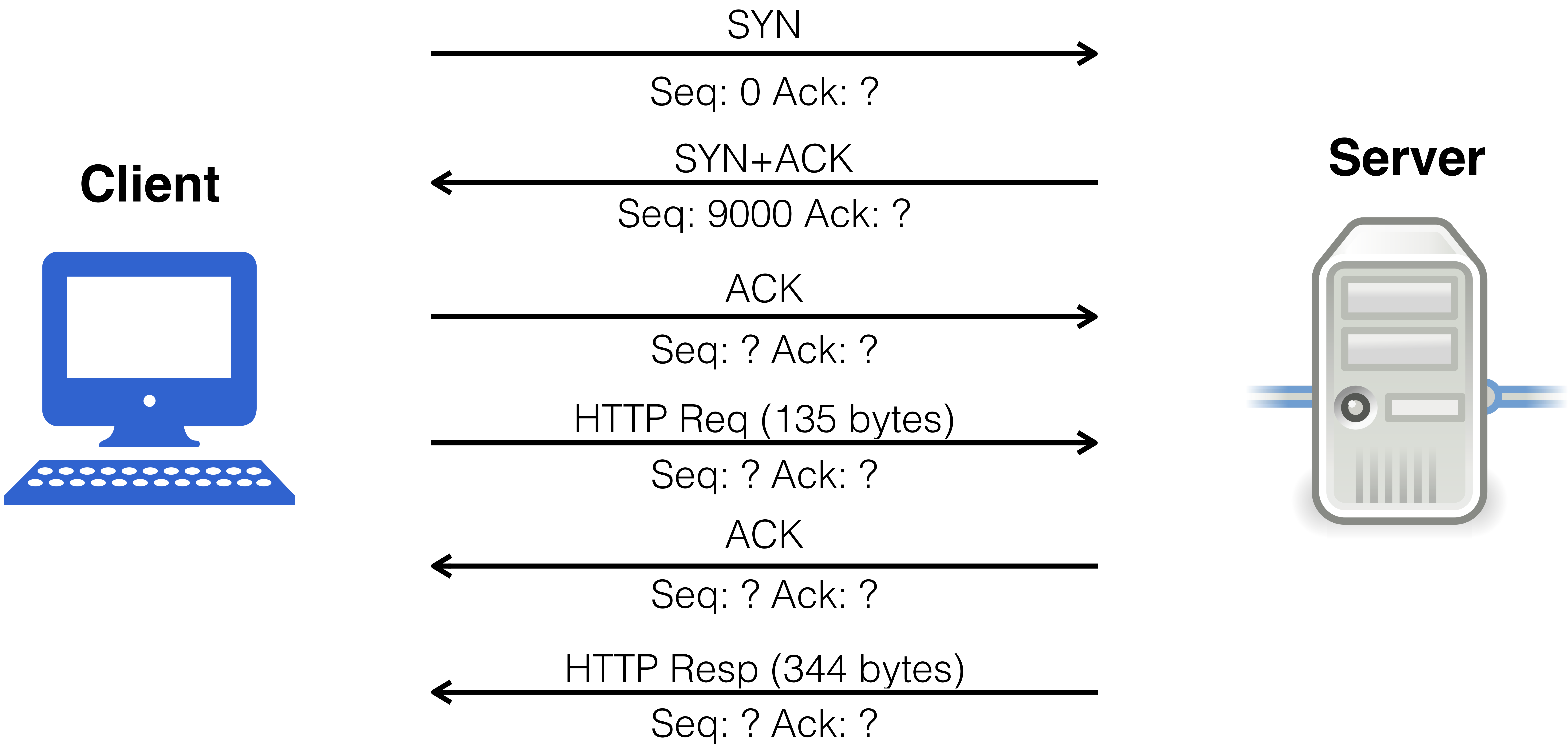
TCP Interception



TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			N S	C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																															

TCP Seq/Ack Numbers



Client

Server

SYN

Seq: 0 Ack: ?

SYN+ACK

Seq: 9000 Ack: ?

ACK

Seq: ? Ack: ?

HTTP Req (135 bytes)

Seq: ? Ack: ?

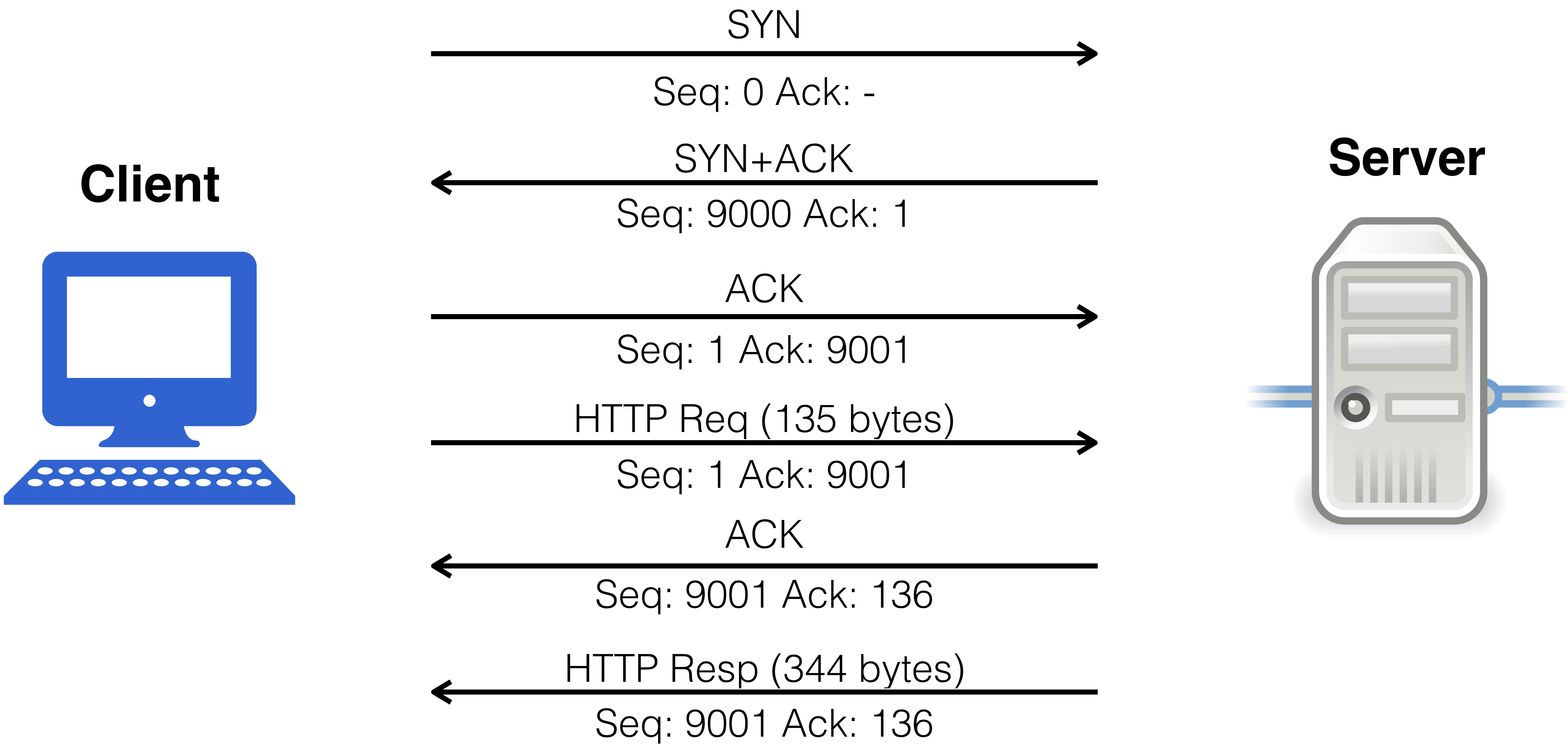
ACK

Seq: ? Ack: ?

HTTP Resp (344 bytes)

Seq: ? Ack: ?

TCP Seq/Ack Numbers



Client

Server

SYN

Seq: 0 Ack: -

SYN+ACK

Seq: 9000 Ack: 1

ACK

Seq: 1 Ack: 9001

HTTP Req (135 bytes)

Seq: 1 Ack: 9001

ACK

Seq: 9001 Ack: 136

HTTP Resp (344 bytes)

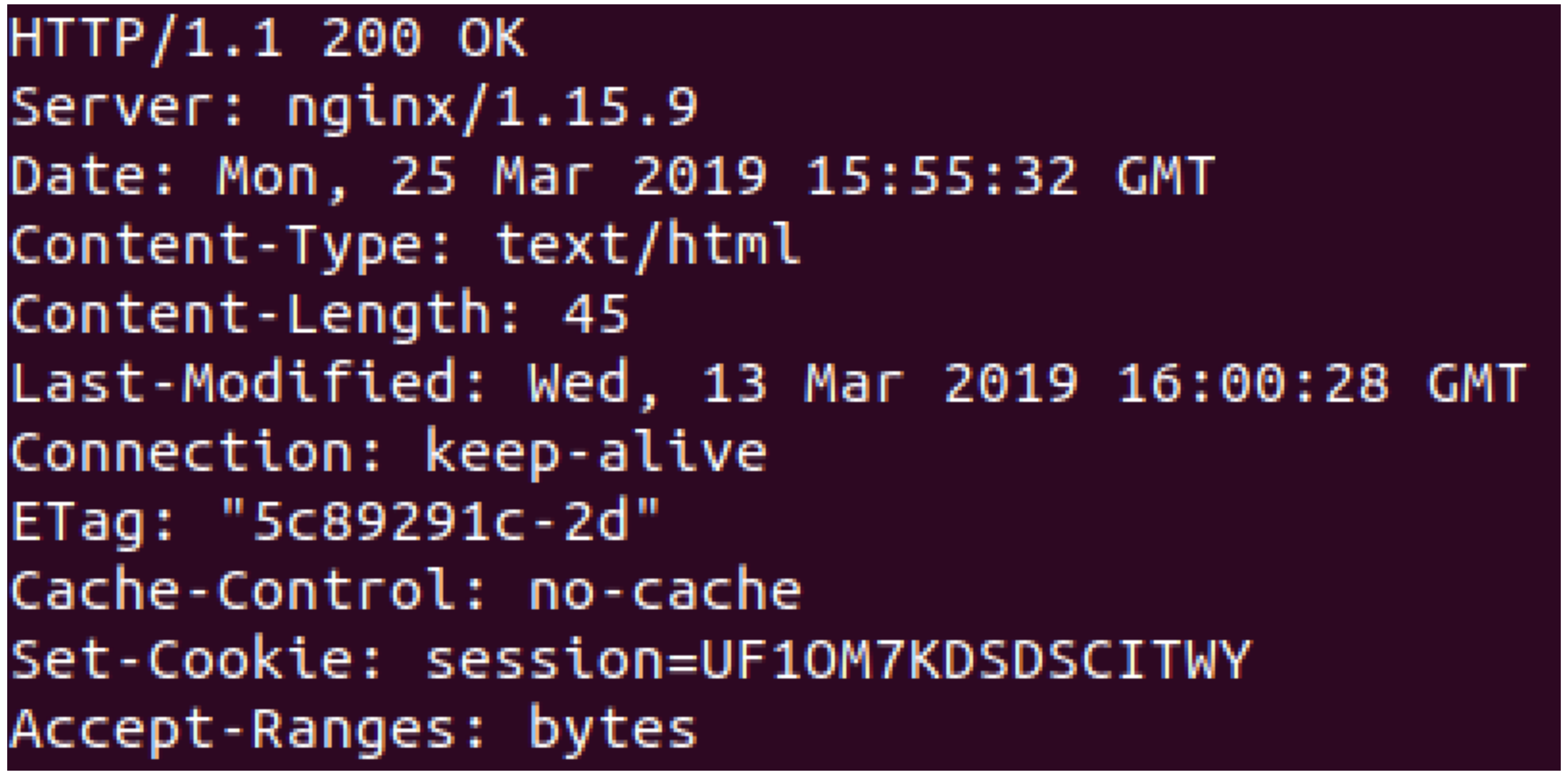
Seq: 9001 Ack: 136

TCP Seq/Ack Numbers

- Demo sequence numbers in Wireshark observing HTTP traffic
- Demo absolute sequence numbers in Wireshark

HTTP Interception

HTTP header

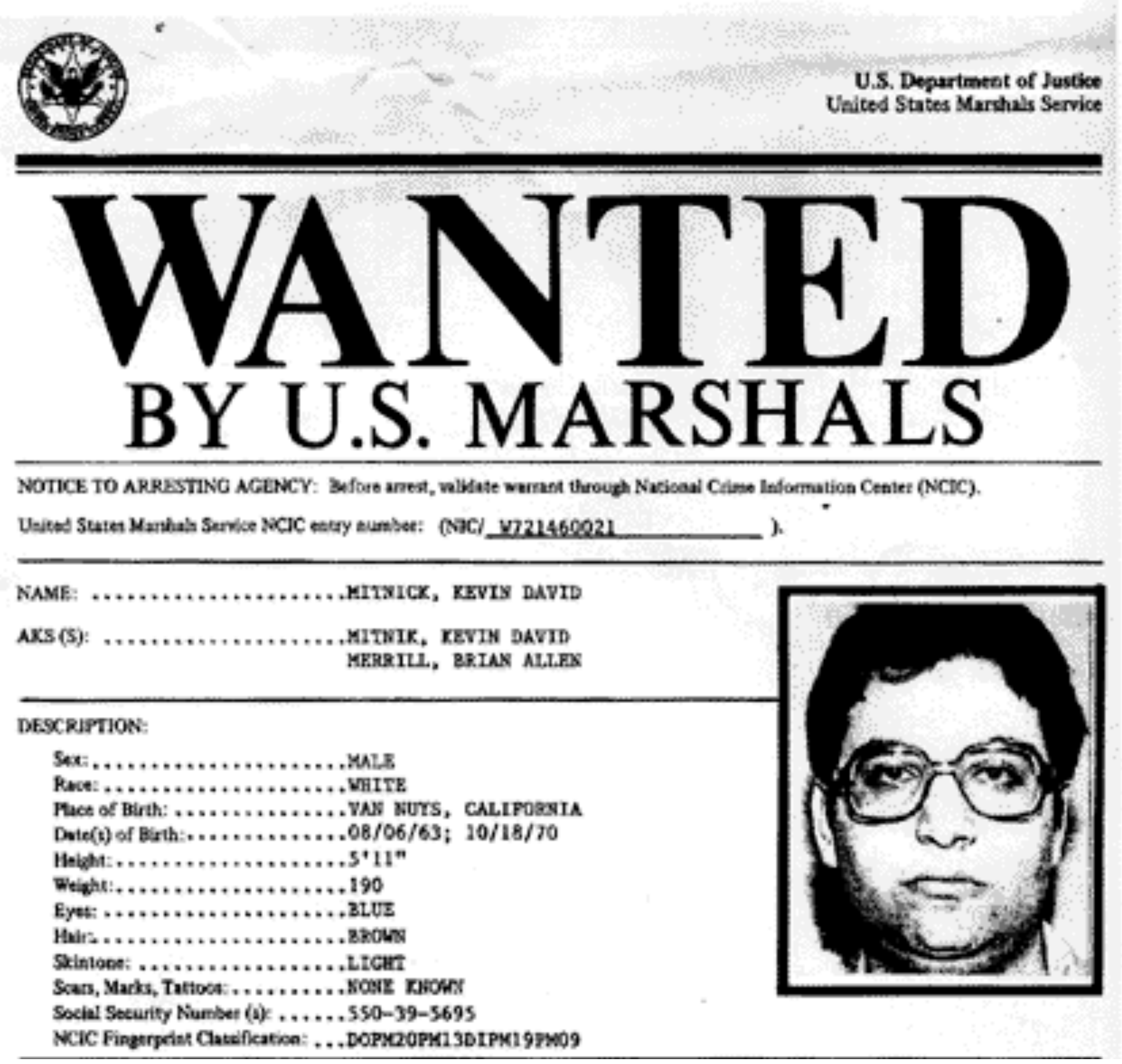


```
HTTP/1.1 200 OK
Server: nginx/1.15.9
Date: Mon, 25 Mar 2019 15:55:32 GMT
Content-Type: text/html
Content-Length: 45
Last-Modified: Wed, 13 Mar 2019 16:00:28 GMT
Connection: keep-alive
ETag: "5c89291c-2d"
Cache-Control: no-cache
Set-Cookie: session=UF1OM7KDSDSCITWY
Accept-Ranges: bytes
```

- What header fields need to change for injection?
- What if injection occurs in separate packet?
- What if HTTP data exceeds one TCP packet? How large is a TCP packet?

Mitnick Xmas Day Attack

- 12/25/1994 attack on San Diego Supercomputer Center
  - Arrested Feb 1995, spent five years in prison, eight months solitary confinement
- Elaborate, multi-step off path TCP hijacking attack



U.S. Department of Justice
United States Marshals Service

WANTED
BY U.S. MARSHALS

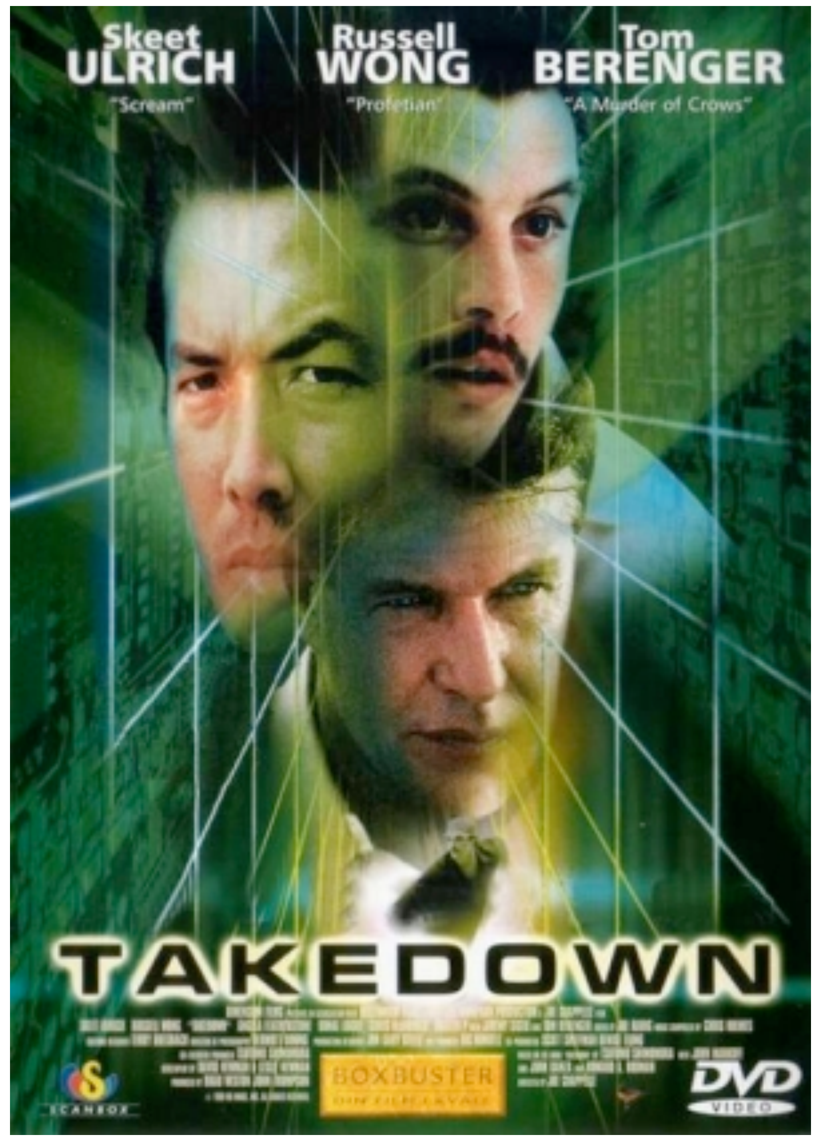
NOTICE TO ARRESTING AGENCY: Before arrest, validate warrant through National Crime Information Center (NCIC).

United States Marshals Service NCIC entry number: (NIC/ W721460021).

NAME:	MITNICK, KEVIN DAVID
AKS (S):	MITNIK, KEVIN DAVID MERRILL, BRIAN ALLEN

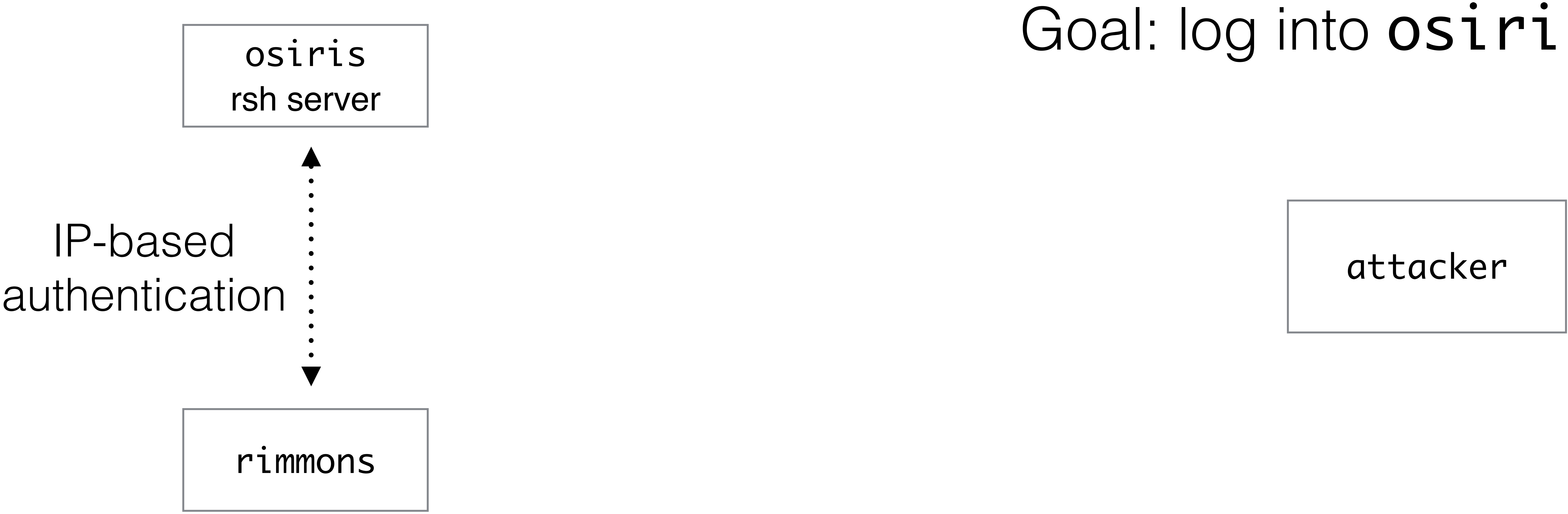
DESCRIPTION:

Sex:	MALE
Race:	WHITE
Place of Birth:	VAN NUYS, CALIFORNIA
Date(s) of Birth:	08/06/63; 10/18/70
Height:	5'11"
Weight:	190
Eyes:	BLUE
Hair:	BROWN
Skintone:	LIGHT
Scars, Marks, Tattoos:	NONE KNOWN
Social Security Number (s):	550-39-5695
NCIC Fingerprint Classification:	DOPM20PM13DIPM19PM09



Movie trailer: https://uofi.app.box.com/file/431669292713

Mitnick Xmas Day Attack

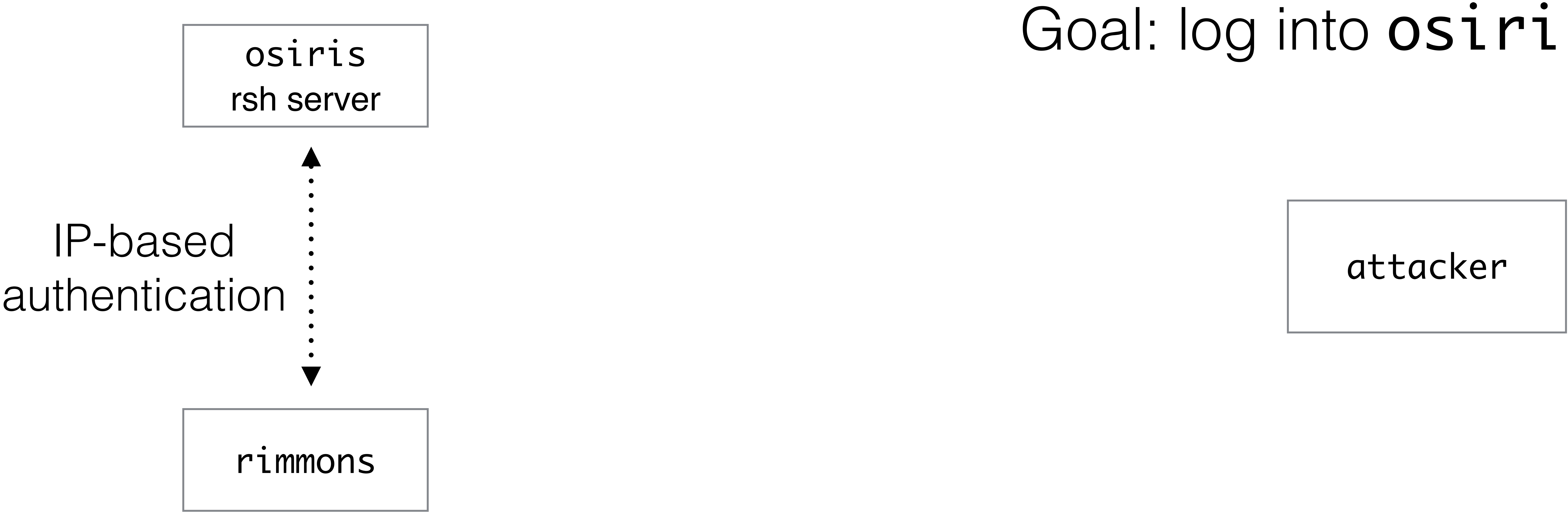


Goal: log into `osiris`

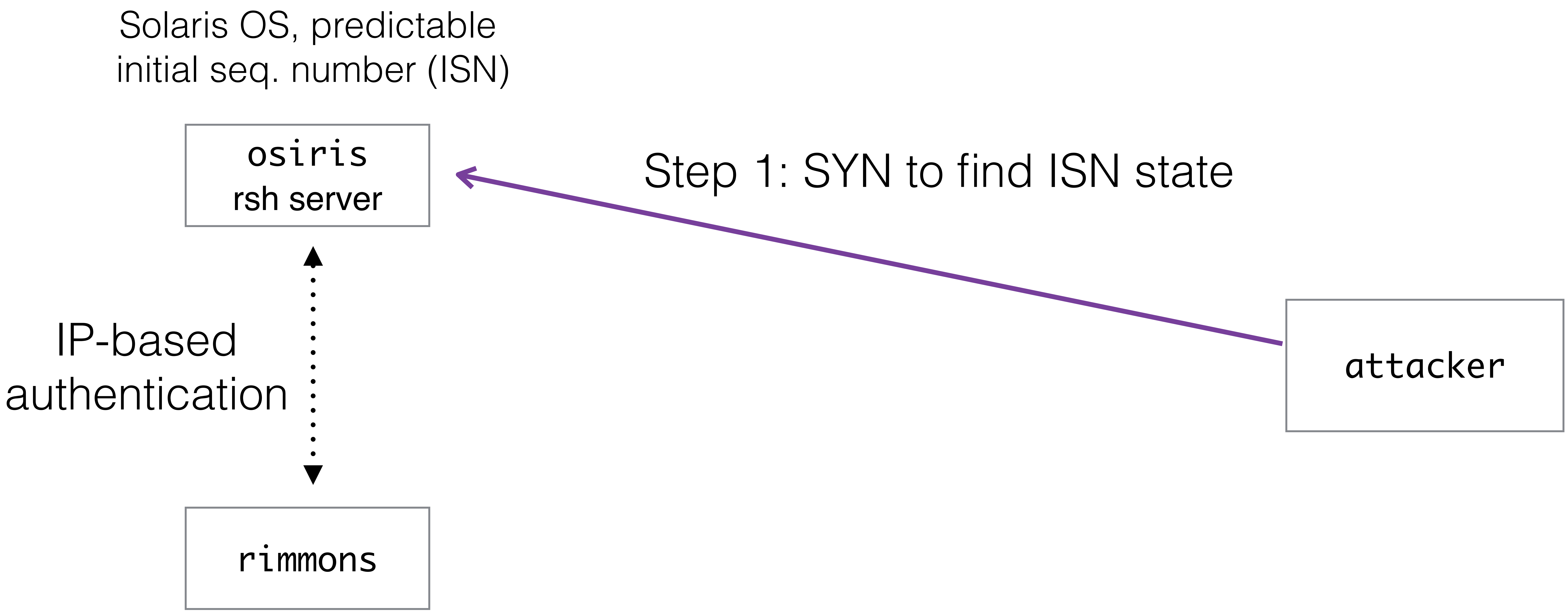
Mitnick Xmas Day Attack

Solaris OS, predictable initial seq. number (ISN)

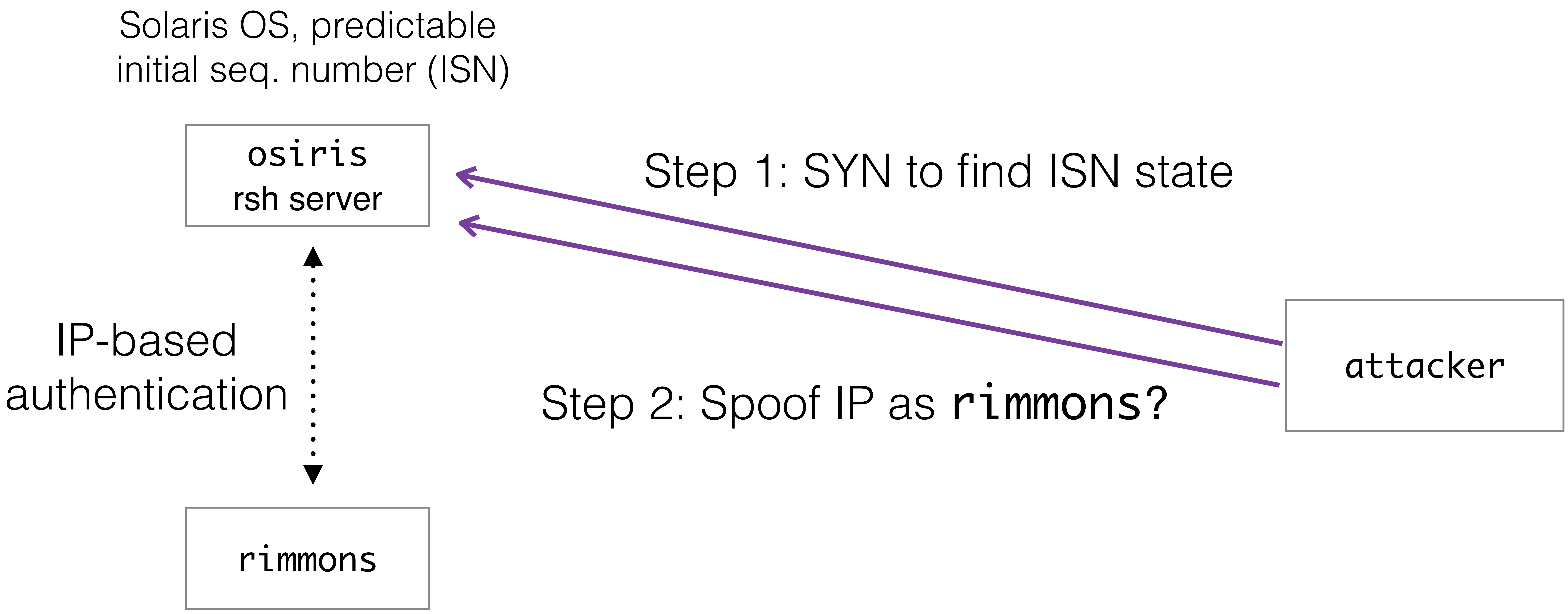
Goal: log into `osiris`



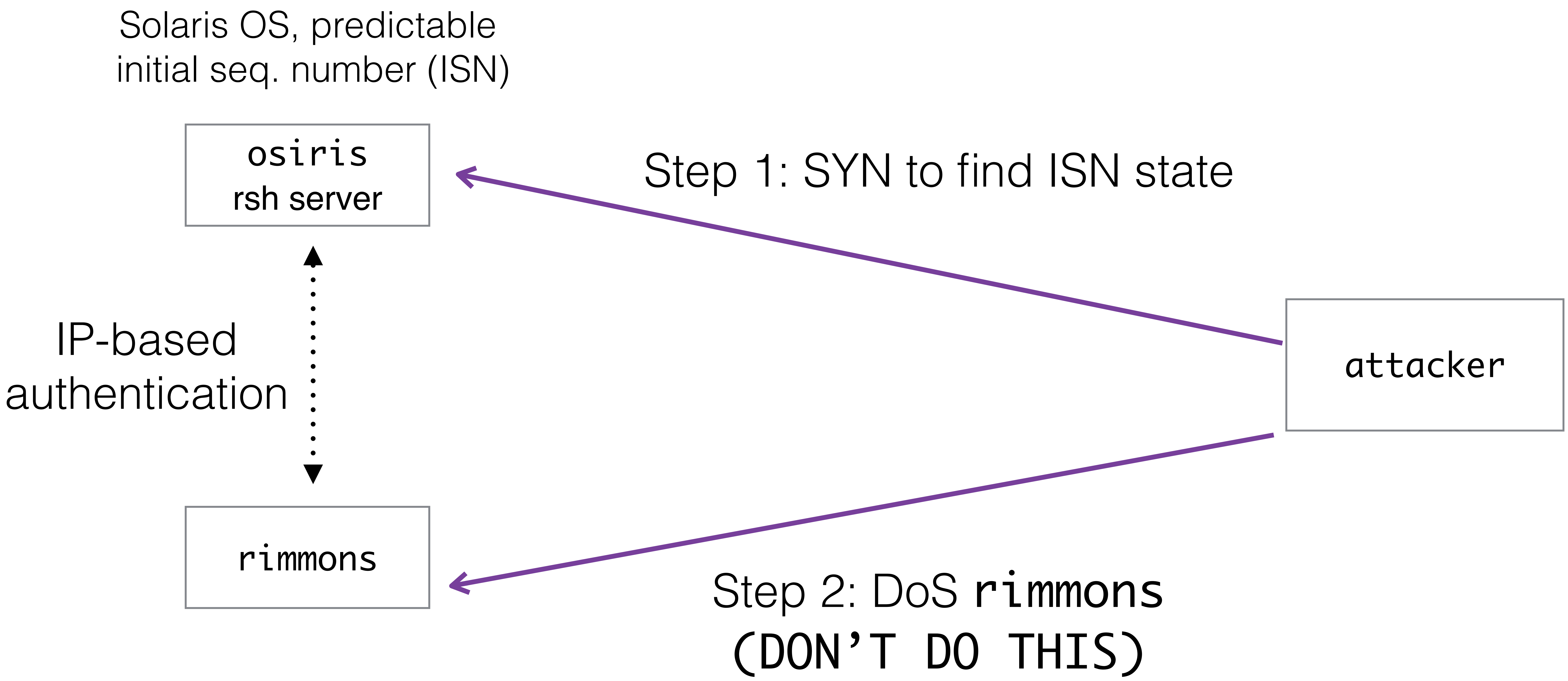
Mitnick Xmas Day Attack


Solaris OS, predictable initial seq. number (ISN)
osiris
rsh server
Step 1: SYN to find ISN state
IP-based authentication
attacker
rimmons

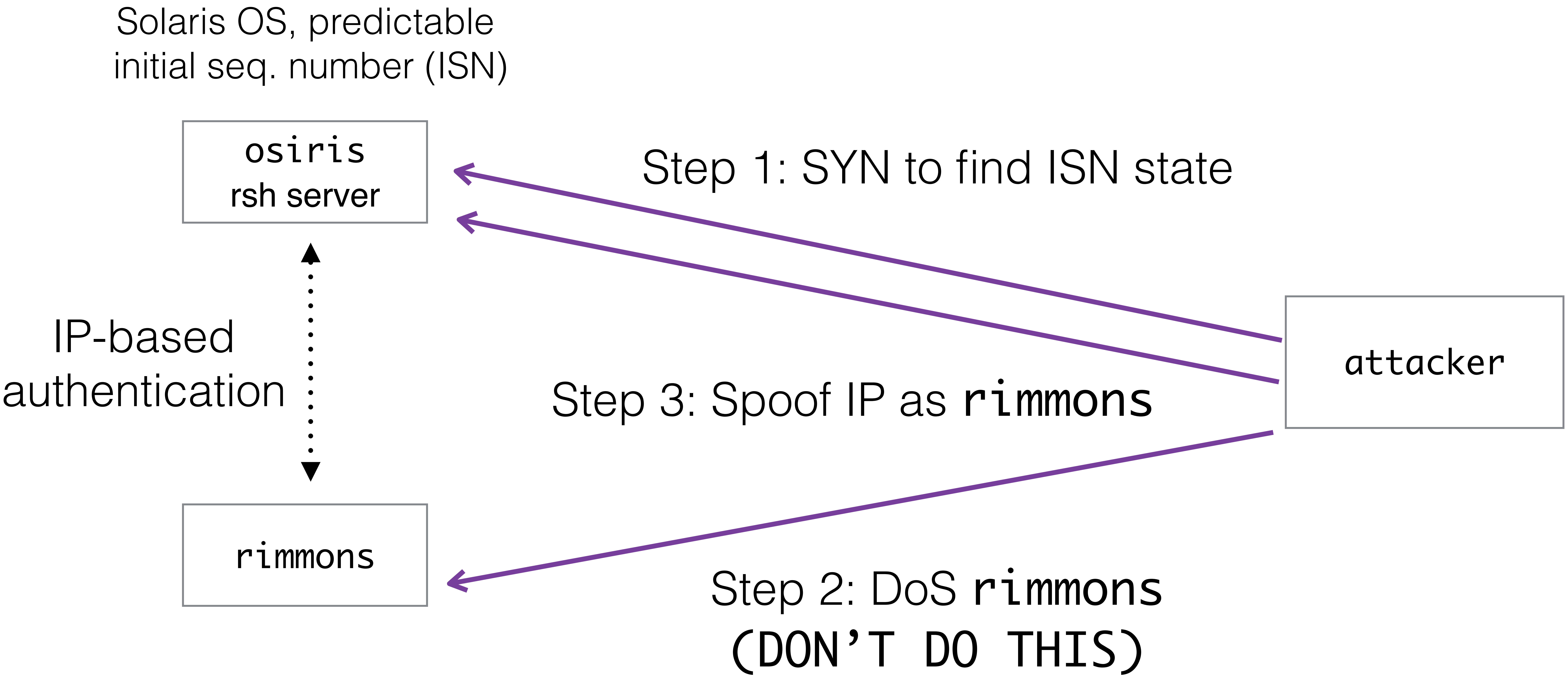
Mitnick Xmas Day Attack


Solaris OS, predictable
initial seq. number (ISN)
osiris
rsh server
IP-based
authentication
rimmons
Step 1: SYN to find ISN state
Step 2: Spoof IP as **rimmons?**
attacker

Mitnick Xmas Day Attack



Mitnick Xmas Day Attack



Solaris OS, predictable initial seq. number (ISN)

Step 1: SYN to find ISN state

Step 3: Spoof IP as `rimmons`

Step 2: DoS `rimmons`

`(DON’T DO THIS)`

Mitnick Demo

- How to determine ISN?
  - Deductive - read the source code link in MP handout
  - Inductive - measure it and observe the pattern