

CSE 484 / CSE M 584 (Spring 2012)

# Network Security

---

Tadayoshi Kohno

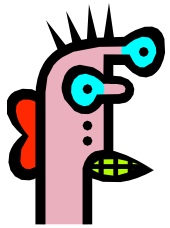
Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

# Goals for Today

---

- ◆ Network security

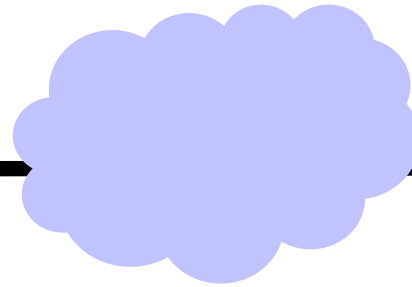
# (Some) Malicious Goals



**User**



**Network  
Admin**



**Intermediate  
ISPs**



**Server**

Launch  
undetectable  
attacks

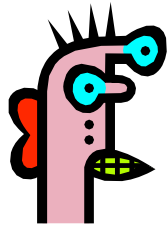
Probe for  
vulnerabilities

Spy on/tamper with traffic

Impersonate servers/users

Identify  
anonymous  
users

# Detecting attacks

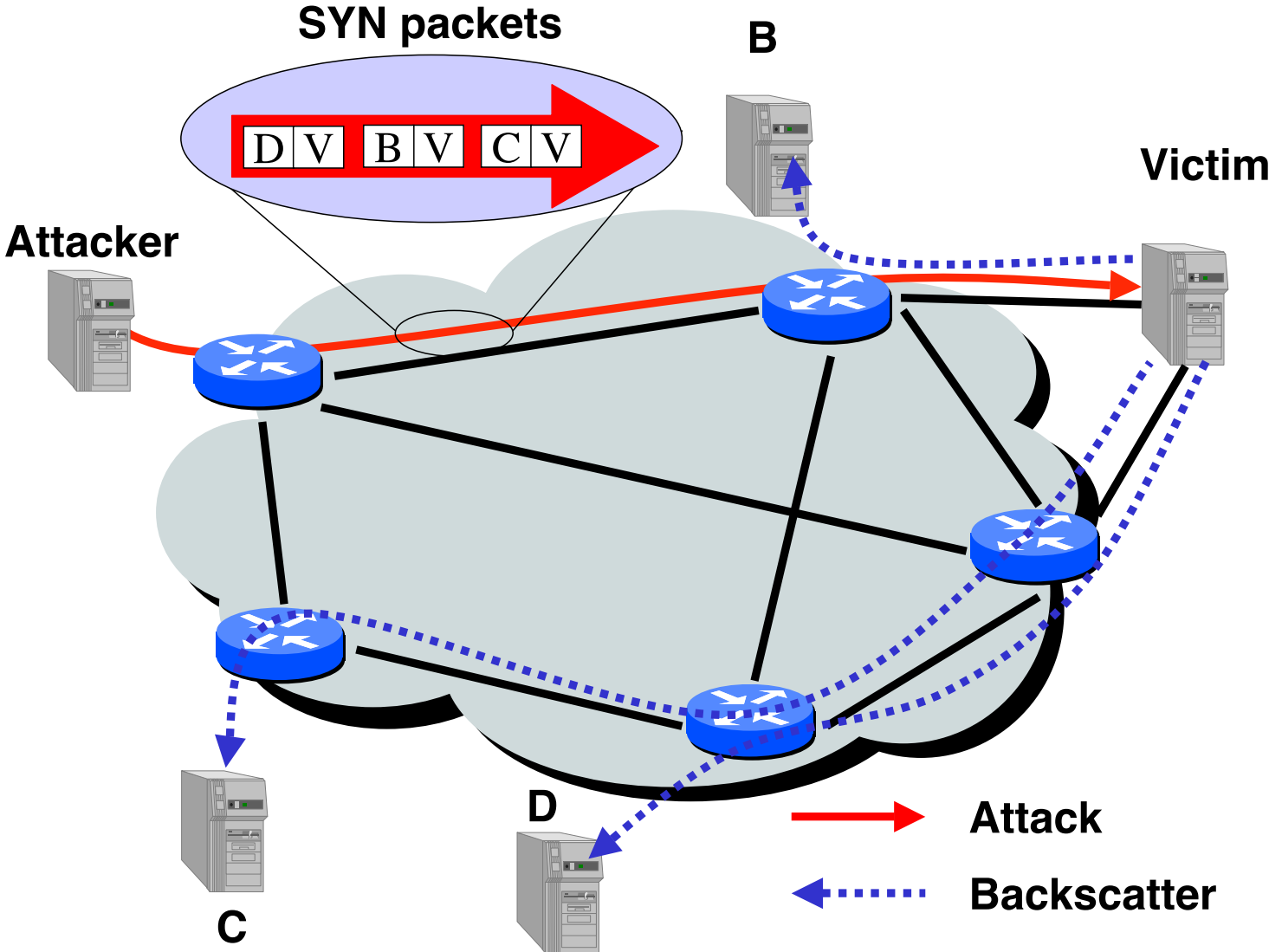


**User**

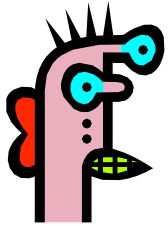
Launch  
undetectable  
attacks

- **Problem:** IP packets contain source IP address
- **Solution:** Spoof IP address

# Inferring DDOS (Moore, Voelker, Savage '01)



# Finding vulnerabilities



**User**

Probe for  
vulnerabilities

- **Many, many tools**
- One example: **Nmap**
  - Many services have known TCP/UDP ports
  - These give away what services you're running

# Nmap example

```
% nmap dsp.cs.washington.edu
```

```
Starting Nmap 5.51 ( http://nmap.org ) at 2011-12-05 14:05 PST
```

```
Nmap scan report for dsp.cs.washington.edu (128.208.4.246)
```

```
Host is up (0.0062s latency).
```

```
Not shown: 996 closed ports
```

```
PORT      STATE SERVICE
```

```
22/tcp    open  ssh
```

```
139/tcp   open  netbios-ssn
```

```
443/tcp   open  https
```

```
445/tcp   open  microsoft-ds
```

```
Nmap done: 1 IP address (1 host up) scanned in 1.36 seconds
```

# Nmap example

```
% nmap aqua.cs.washington.edu
```

```
Starting Nmap 5.51 ( http://nmap.org ) at 2011-12-05 14:06 PST
```

```
Nmap scan report for aqua.cs.washington.edu (128.208.4.187)
```

```
Host is up (0.0022s latency).
```

```
Not shown: 990 filtered ports
```

```
PORT      STATE SERVICE
```

```
80/tcp    open  http
```

```
135/tcp   open  msrpc
```

```
139/tcp   open  netbios-ssn
```

```
445/tcp   open  microsoft-ds
```

```
1025/tcp  open  NFS-or-IIS
```

```
1026/tcp  open  LSA-or-nterm
```

```
1027/tcp  open  IIS
```

```
1028/tcp  open  unknown
```

```
1048/tcp  open  neod2
```

```
3389/tcp  open  ms-term-serv
```

```
Nmap done: 1 IP address (1 host up) scanned in 5.29 seconds
```



# Fingerprinting users



## Server

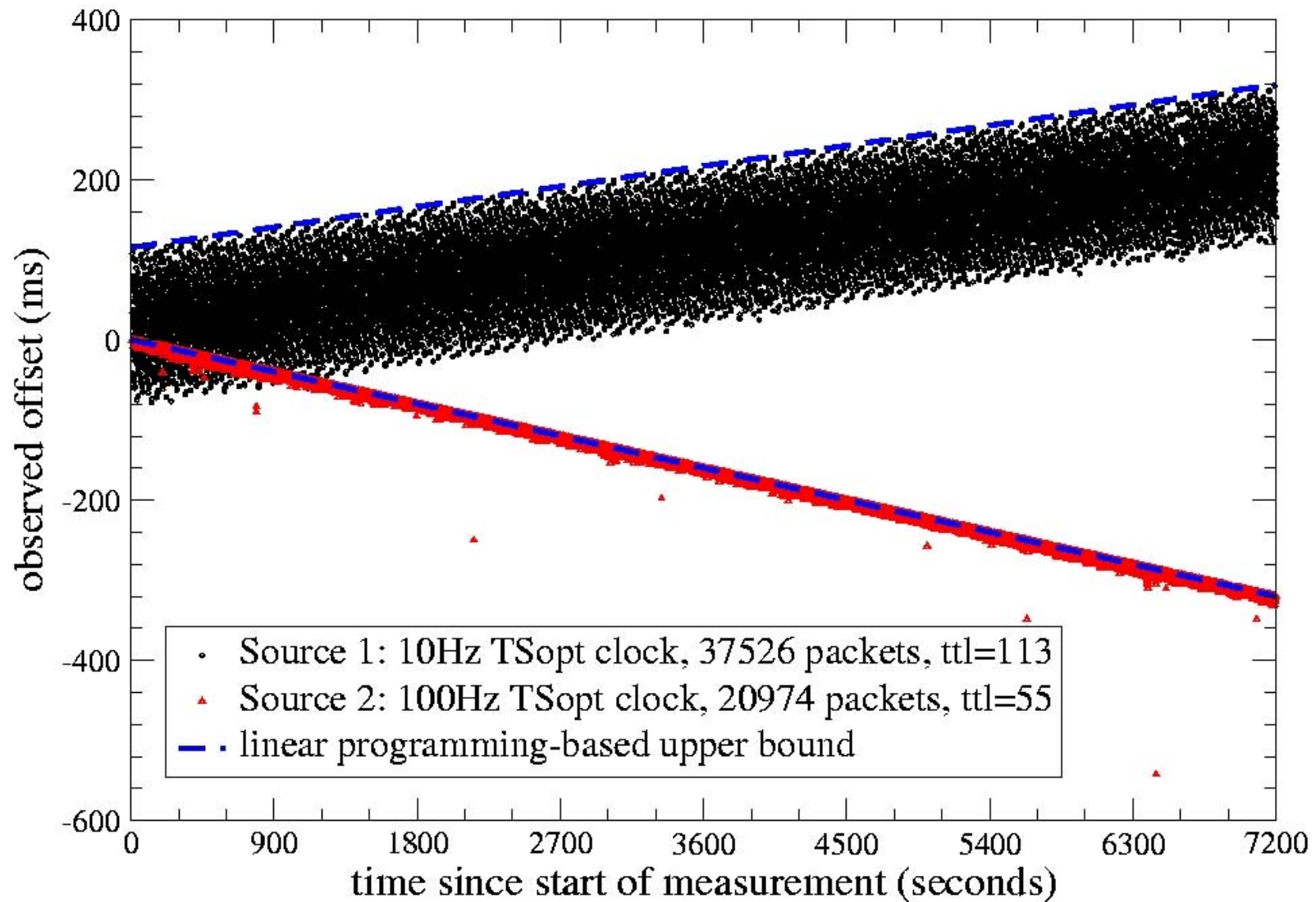
Identify  
anonymous  
users

- **Browser**
- **Clocks**
- More

Browser example

<http://panopticklick.eff.org/>

# Clocks

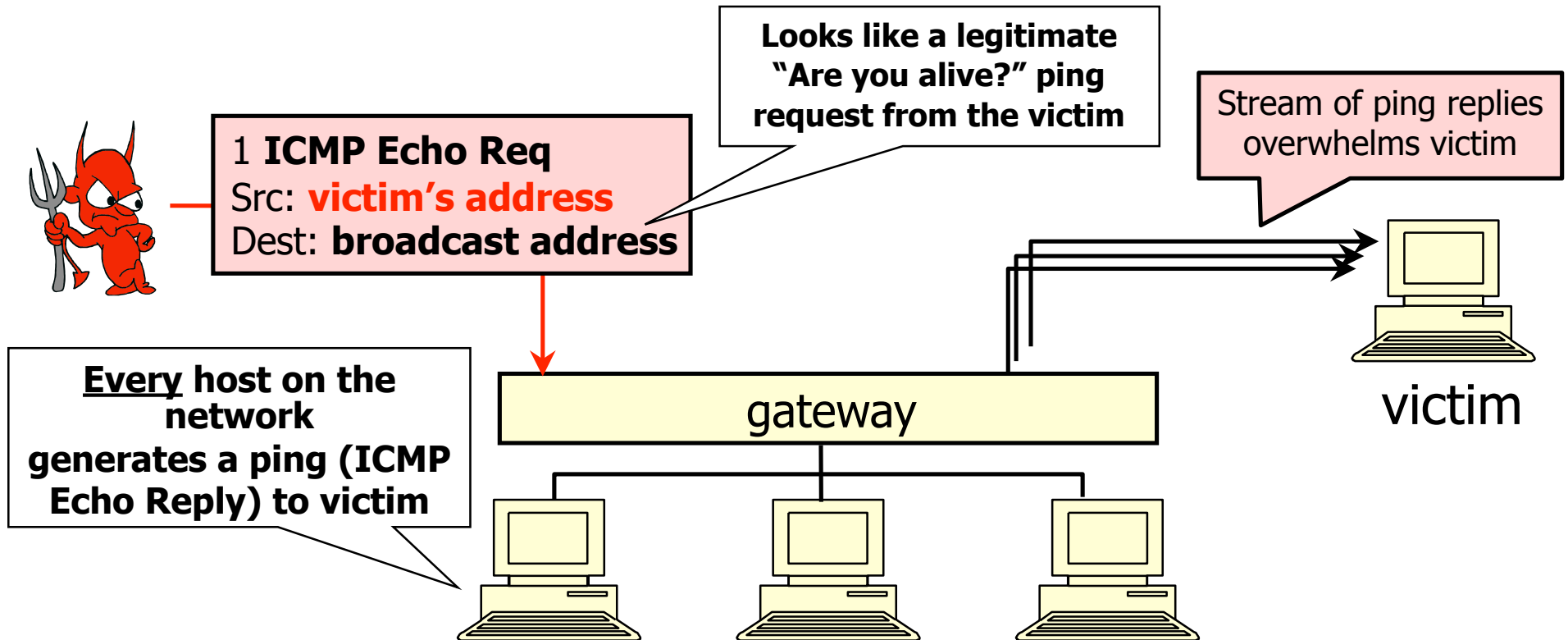


# Security Issues in TCP/UDP

---

- ◆ Network packets pass through/by untrusted hosts
  - Eavesdropping (packet sniffing)
  - Modifications
- ◆ IP addresses are public
  - Smurf attacks
  - Anonymity?
- ◆ TCP connection requires state
  - SYN flooding
- ◆ TCP state is easy to guess
  - TCP spoofing and connection hijacking

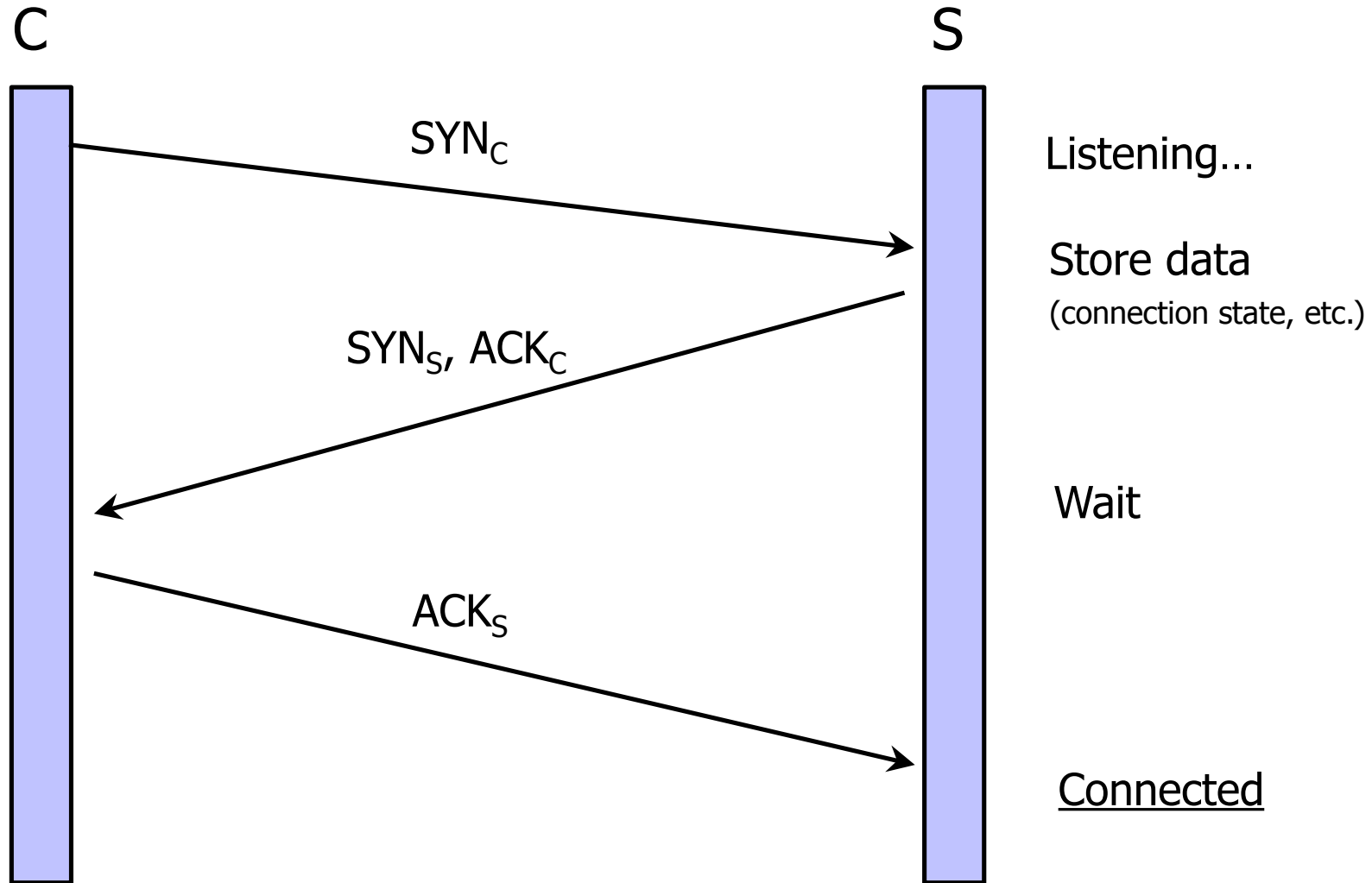
# Smurf Attack



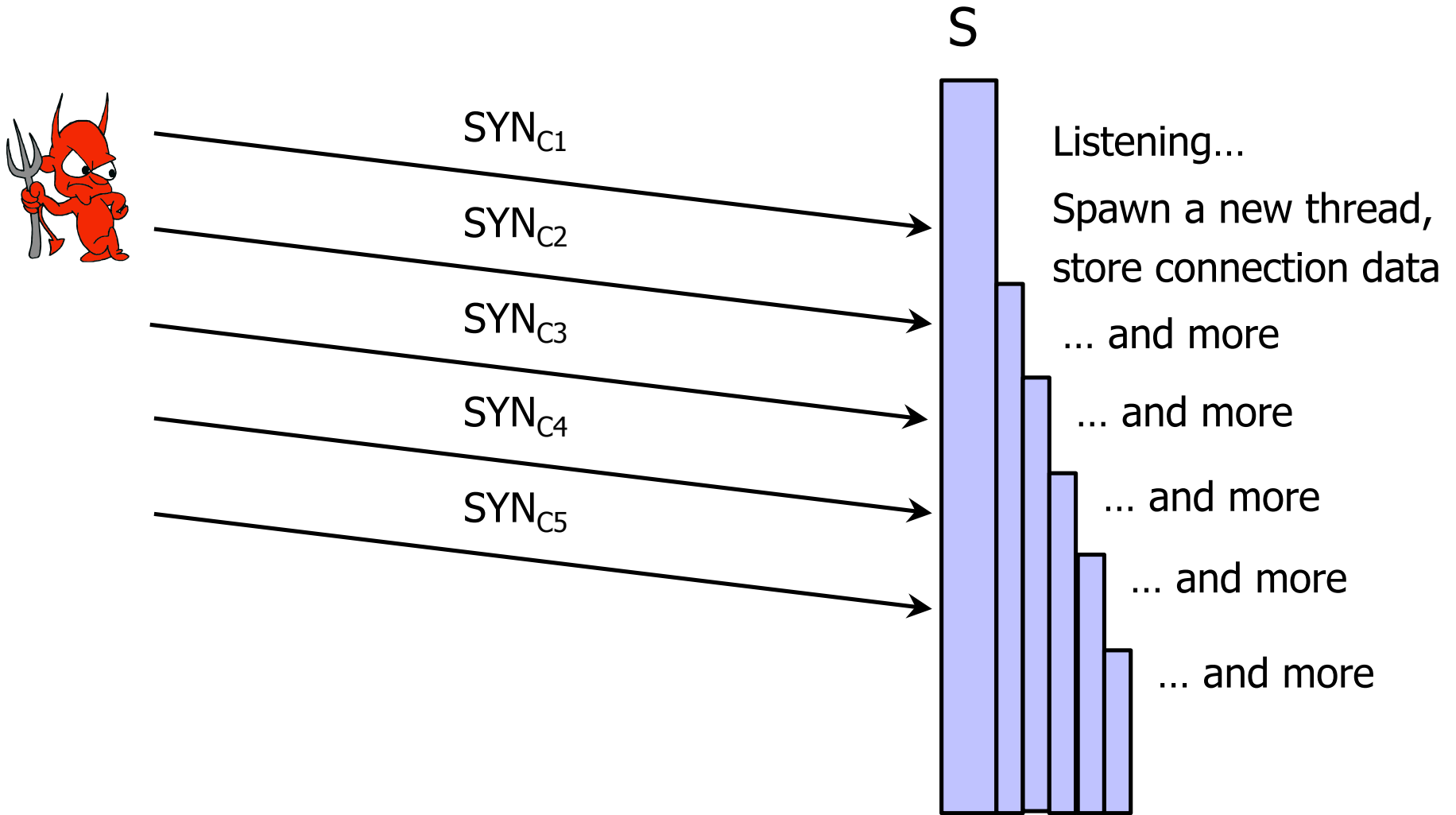
Solution: reject external packets to broadcast addresses

# TCP Handshake

---



# SYN Flooding Attack



# SYN Flooding Explained

---

- ◆ Attacker sends many connection requests with spoofed source addresses
- ◆ Victim allocates resources for each request
  - Connection state maintained until timeout
  - Fixed bound on half-open connections
- ◆ Once resources exhausted, requests from legitimate clients are denied
- ◆ This is a classic **denial of service (DoS)** attack
  - Common pattern: it costs nothing to TCP initiator to send a connection request, but TCP responder must allocate state for each request (**asymmetry!**)



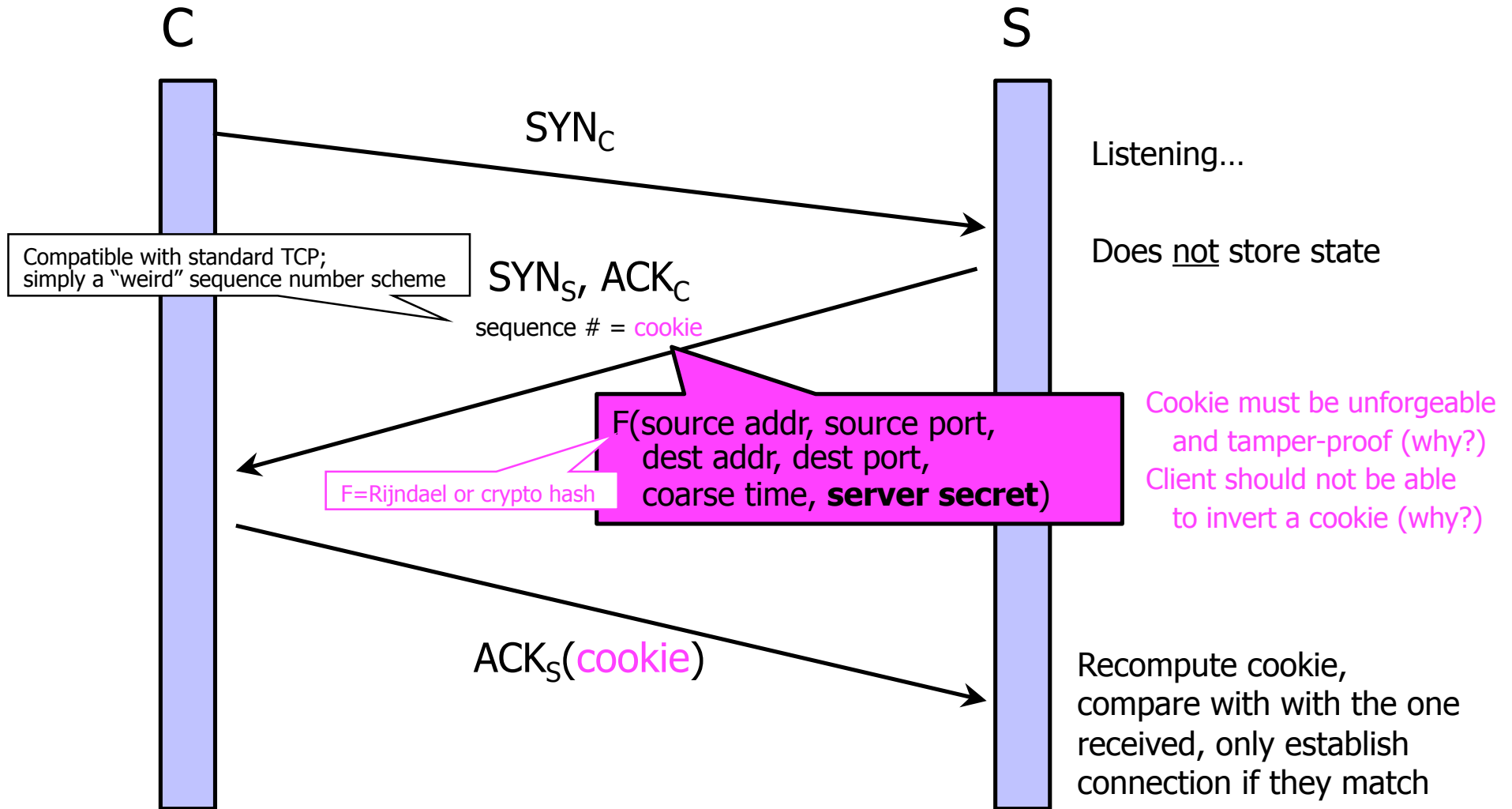
# Preventing Denial of Service

---

- ◆ DoS is caused by asymmetric state allocation
  - If responder opens a state for each connection attempt, attacker can initiate thousands of connections from bogus or forged IP addresses
- ◆ **Cookies** ensure that the responder is stateless until initiator produced at least 2 messages
  - Responder's state (IP addresses and ports of the connection) is stored in a cookie and sent to initiator
  - After initiator responds, cookie is regenerated and compared with the cookie returned by the initiator

# SYN Cookies

[Bernstein and Schenk]



More info: <http://cr.yp.to/syncookies.html>

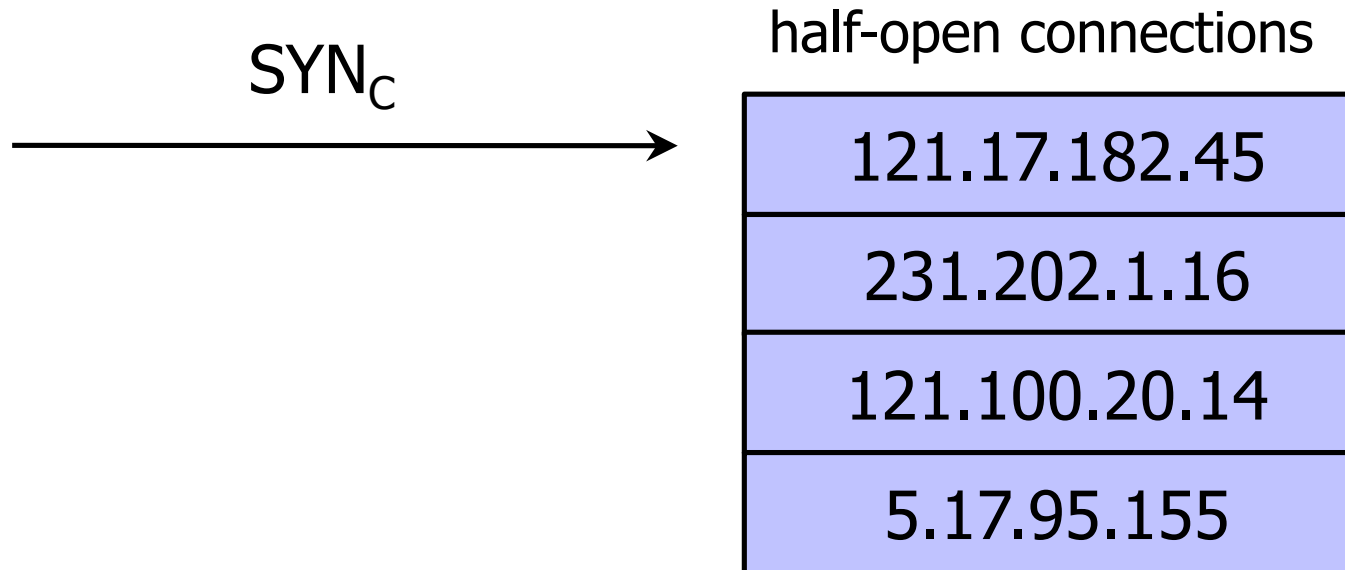
# Anti-Spoofing Cookies: Basic Pattern

---

- ◆ Client sends request (message #1) to server
- ◆ Typical protocol:
  - Server sets up connection, responds with message #2
  - Client may complete session or not (potential DoS)
- ◆ Cookie version:
  - Server responds with hashed connection data instead of message #2
  - Client confirms by returning hashed data
    - If source IP address is bogus, attacker can't confirm
  - Need an extra step to send postponed message #2, except in TCP (SYN-ACK already there)

# Another Defense: Random Deletion

---



- ◆ If SYN queue is full, delete random entry
  - Legitimate connections have a chance to complete
  - Fake addresses will be eventually deleted
- ◆ Easy to implement

# “Ping of Death”

---

- ◆ If an old Windows machine received an ICMP packet with a payload longer than 64K, machine would crash or reboot
  - Programming error in older versions of Windows
  - Packets of this length are illegal, so programmers of Windows code did not account for them
- ◆ Recall “security theme” of this course - every line of code might be the target of an adversary

Solution: patch OS, filter out ICMP packets

# Intrusion Detection Systems

---

- ◆ Advantage: can recognize new attacks and new versions of old attacks
- ◆ Disadvantages
  - High false positive rate
  - Must be trained on known good data
    - Training is hard because network traffic is very diverse
  - Definition of “normal” constantly evolves
    - What’s the difference between a **flash crowd** and a **denial of service** attack?

# Intrusion Detection Problems

---

- ◆ Lack of training data with real attacks
  - But lots of “normal” network traffic, system call data
- ◆ Data drift
  - Statistical methods detect changes in behavior
  - Attacker can attack gradually and incrementally
- ◆ Main characteristics not well understood
  - By many measures, attack may be within bounds of “normal” range of activities
- ◆ False identifications are very costly
  - Sysadm will spend many hours examining evidence

# Intrusion Detection Errors

---

- ◆ **False negatives:** attack is not detected
  - Big problem in signature-based misuse detection
- ◆ **False positives:** harmless behavior is classified as an attack
  - Big problem in statistical anomaly detection
- ◆ Both types of IDS suffer from both error types
- ◆ Which is a bigger problem?
  - Attacks are fairly rare events



# Base-Rate Fallacy

---

- ◆ 1% of traffic is SYN floods; IDS accuracy is 90%
  - IDS classifies a SYN flood as attack with prob. 90%, classifies a valid connection as attack with prob. 10%
- ◆ What is the probability that a connection flagged by IDS as a SYN flood is actually valid traffic?

# Conditional Probability

---

- ◆ Suppose two events A and B occur with probability  $\Pr(A)$  and  $\Pr(B)$ , respectively
- ◆ Let  $\Pr(AB)$  be probability that both A and B occur
- ◆ What is the **conditional probability** that A occurs assuming B has occurred?

$$\Pr(A \mid B) = \frac{\Pr(AB)}{\Pr(B)}$$

# Bayes' Theorem

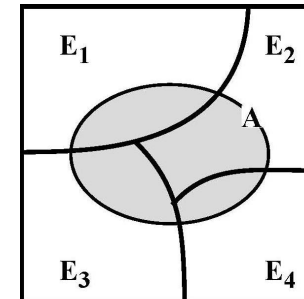
---

- ◆ Suppose mutually exclusive events  $E_1, \dots, E_n$  together cover the entire set of possibilities
- ◆ Then probability of any event  $A$  occurring is

$$\Pr(A) = \sum_{1 \leq i \leq n} \Pr(A | E_i) \cdot \Pr(E_i)$$

– Intuition: since  $E_1, \dots, E_n$  cover entire

probability space, whenever  $A$  occurs,  
some event  $E_i$  must have occurred



- ◆ Can rewrite this formula as

$$\Pr(E_i | A) = \frac{\Pr(A | E_i) \cdot \Pr(E_i)}{\Pr(A)}$$

# Base-Rate Fallacy

---

- ◆ 1% of traffic is SYN floods; IDS accuracy is 90%
  - IDS classifies a SYN flood as attack with prob. 90%, classifies a valid connection as attack with prob. 10%
- ◆ What is the probability that a connection flagged by IDS as a SYN flood is actually valid traffic?

$$\begin{aligned} \Pr(\text{valid} \mid \text{alarm}) &= \frac{\Pr(\text{alarm} \mid \text{valid}) \cdot \Pr(\text{valid})}{\Pr(\text{alarm})} \\ &= \frac{\Pr(\text{alarm} \mid \text{valid}) \cdot \Pr(\text{valid})}{\Pr(\text{alarm} \mid \text{valid}) \cdot \Pr(\text{valid}) + \Pr(\text{alarm} \mid \text{SYN flood}) \cdot \Pr(\text{SYN flood})} \\ &= \frac{0.10 \cdot 0.99}{0.10 \cdot 0.99 + 0.90 \cdot 0.01} = 92\% \text{ chance raised alarm} \\ &\quad \text{is false!!!} \end{aligned}$$