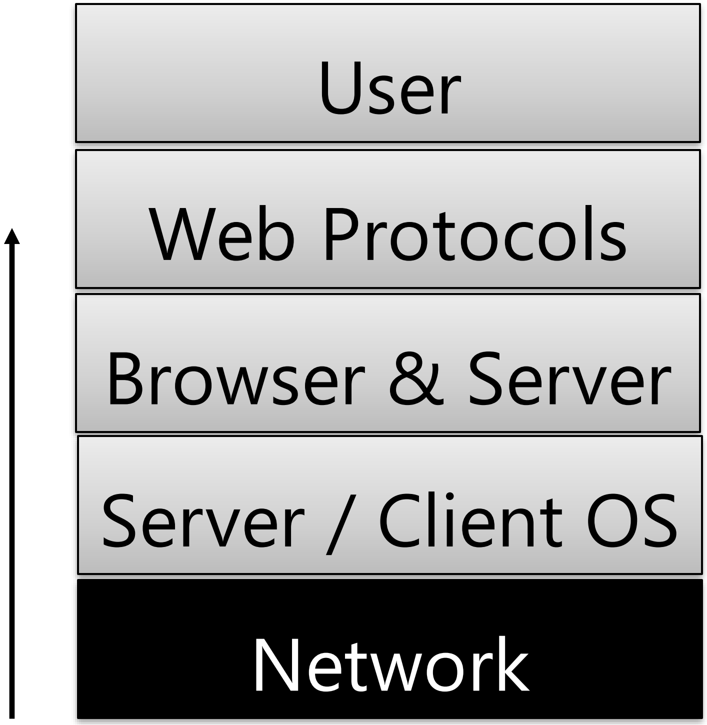
Network Attacks & Defenses

Prateek Saxena

CS3235 – Computer Security

Threat Model : Network Attacker

- A Threat Model defines:
  - Desired Security Property / Goal
  - Attacker Capabilities
  - Assumptions about the setup
- **This Module**:
  - Stack of Threat Models



User

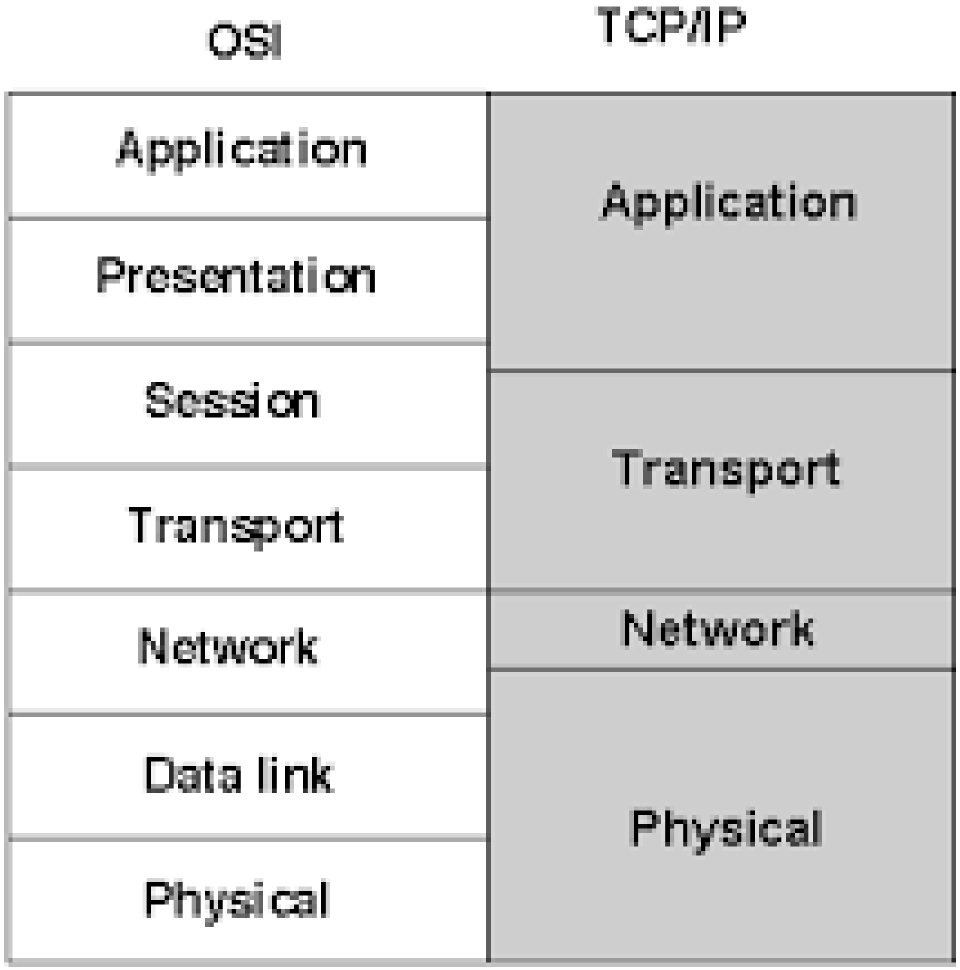
Web Protocols

Browser & Server

Server / Client OS

Network

Today:
The Network Stack



OSI	TCP/IP
Application	Application
Presentation	
Session	Transport
Transport	
Network	Network
Data link	Physical
Physical	

Networking 101:
How The Internet Works

Internet Works

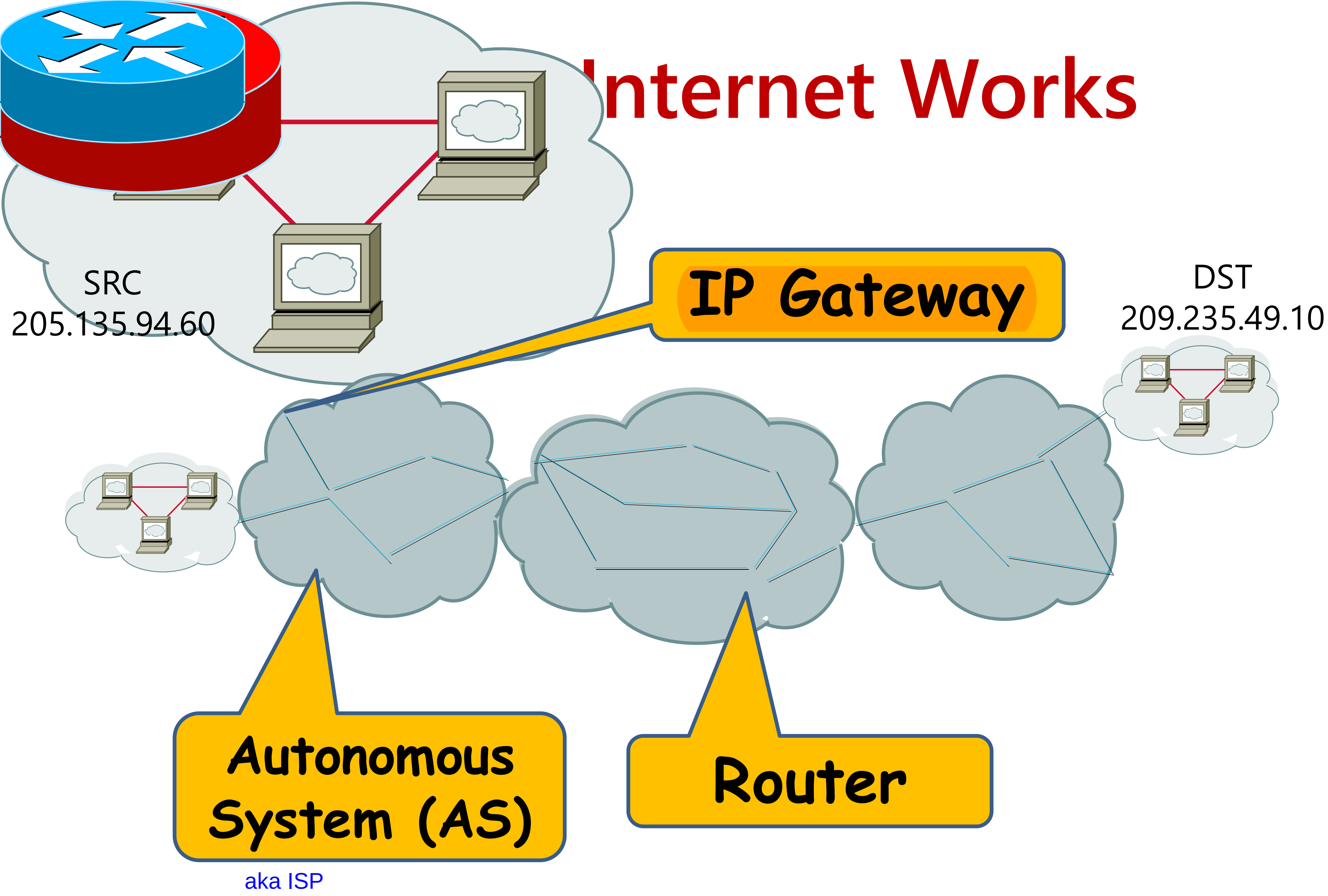
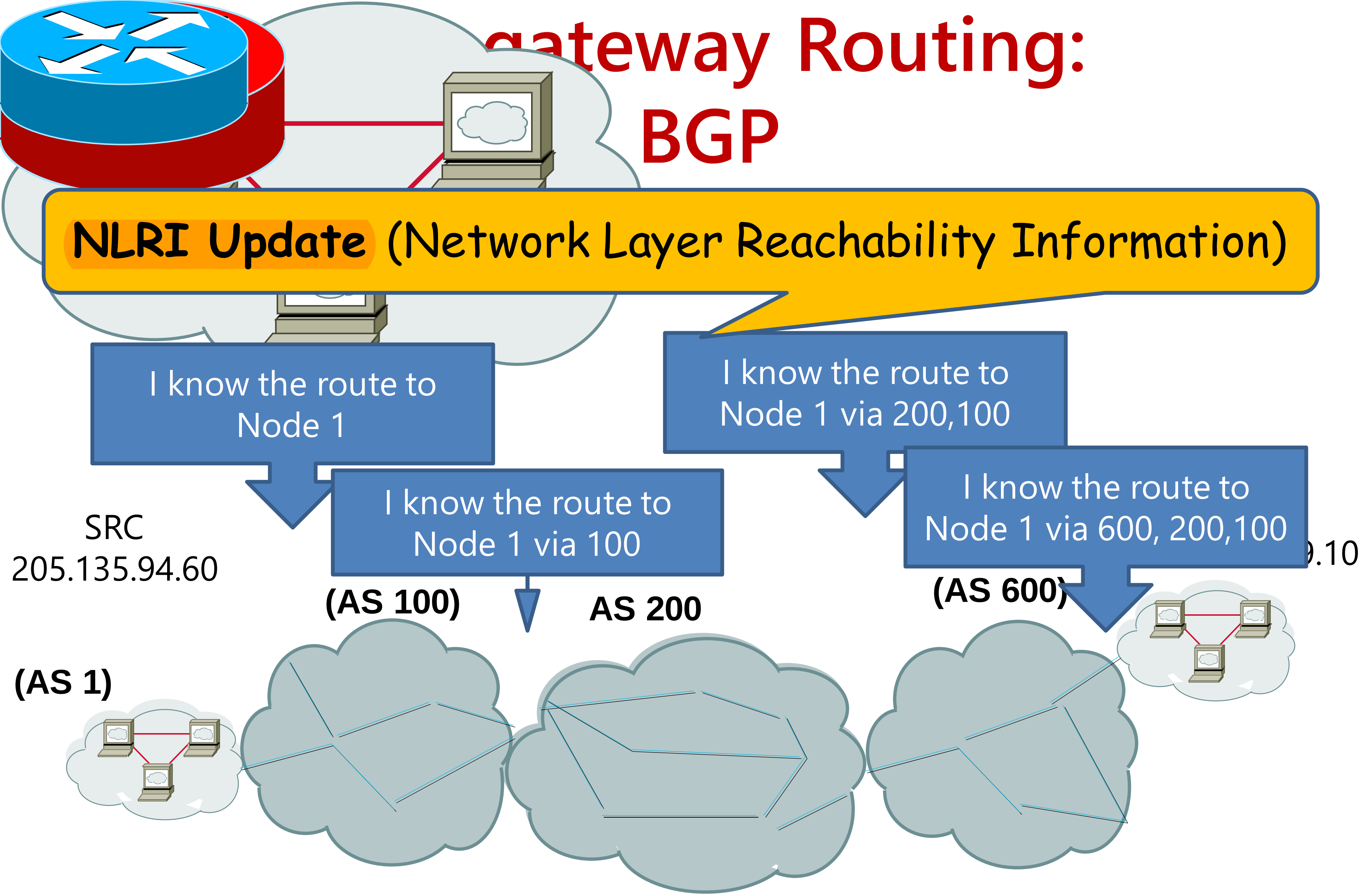


Figure from Boneh et al.

gateway Routing: BGP



NLRI Update (Network Layer Reachability Information)

I know the route to Node 1

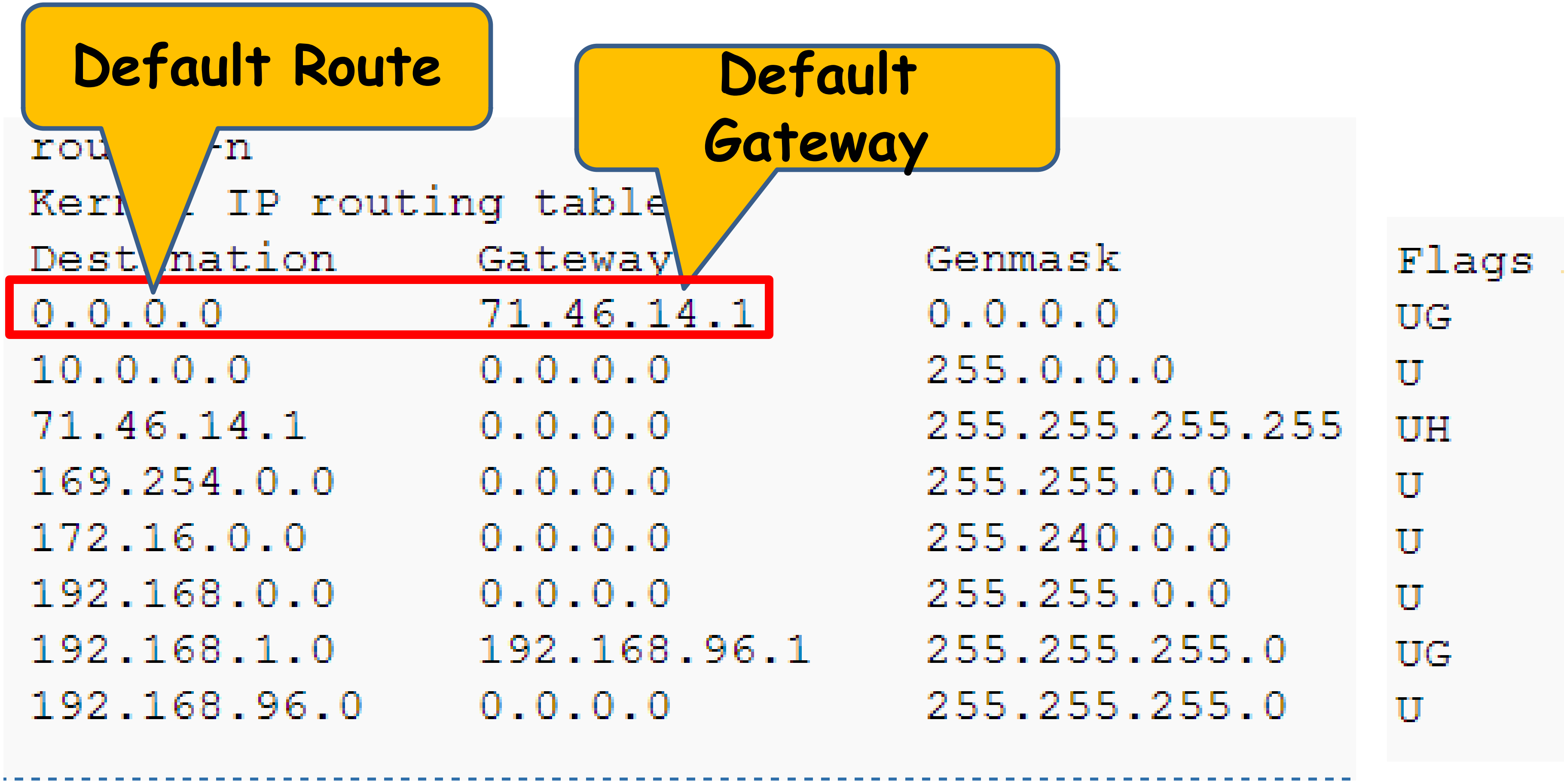
I know the route to Node 1 via 100

I know the route to Node 1 via 200,100

I know the route to Node 1 via 600, 200,100

Figure from Boneh et. al.

How The Internet Routing Works: IP Protocol



Default Route

Default Gateway

```
route -n
Kernel IP routing table
Destination      Gateway         Genmask            Flags
0.0.0.0          71.46.14.1      0.0.0.0            UG
10.0.0.0         0.0.0.0         255.0.0.0          U
71.46.14.1       0.0.0.0         255.255.255.255    UH
169.254.0.0      0.0.0.0         255.255.0.0        U
172.16.0.0       0.0.0.0         255.240.0.0        U
192.168.0.0      0.0.0.0         255.255.0.0        U
192.168.1.0      192.168.96.1    255.255.255.0      UG
192.168.96.0     0.0.0.0         255.255.255.0      U
```

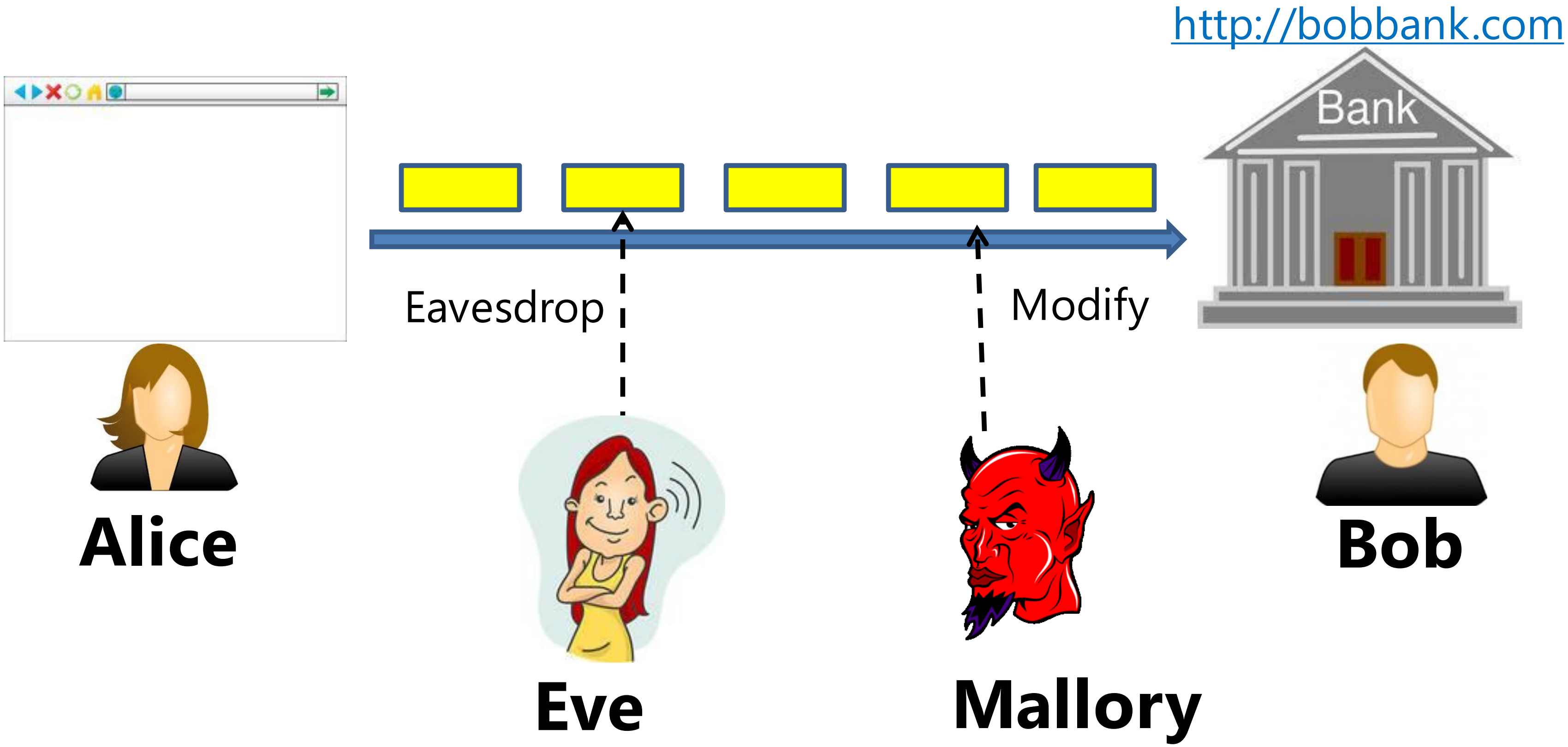
Figure from Wikipedia

How The Internet Works: UDP & TCP

- Unreliable Data delivery over IP: **UDP**
- "Reliable" Data Delivery: **TCP**
  - Connection-oriented, ordered packets

Network Attacks

Network Attacker

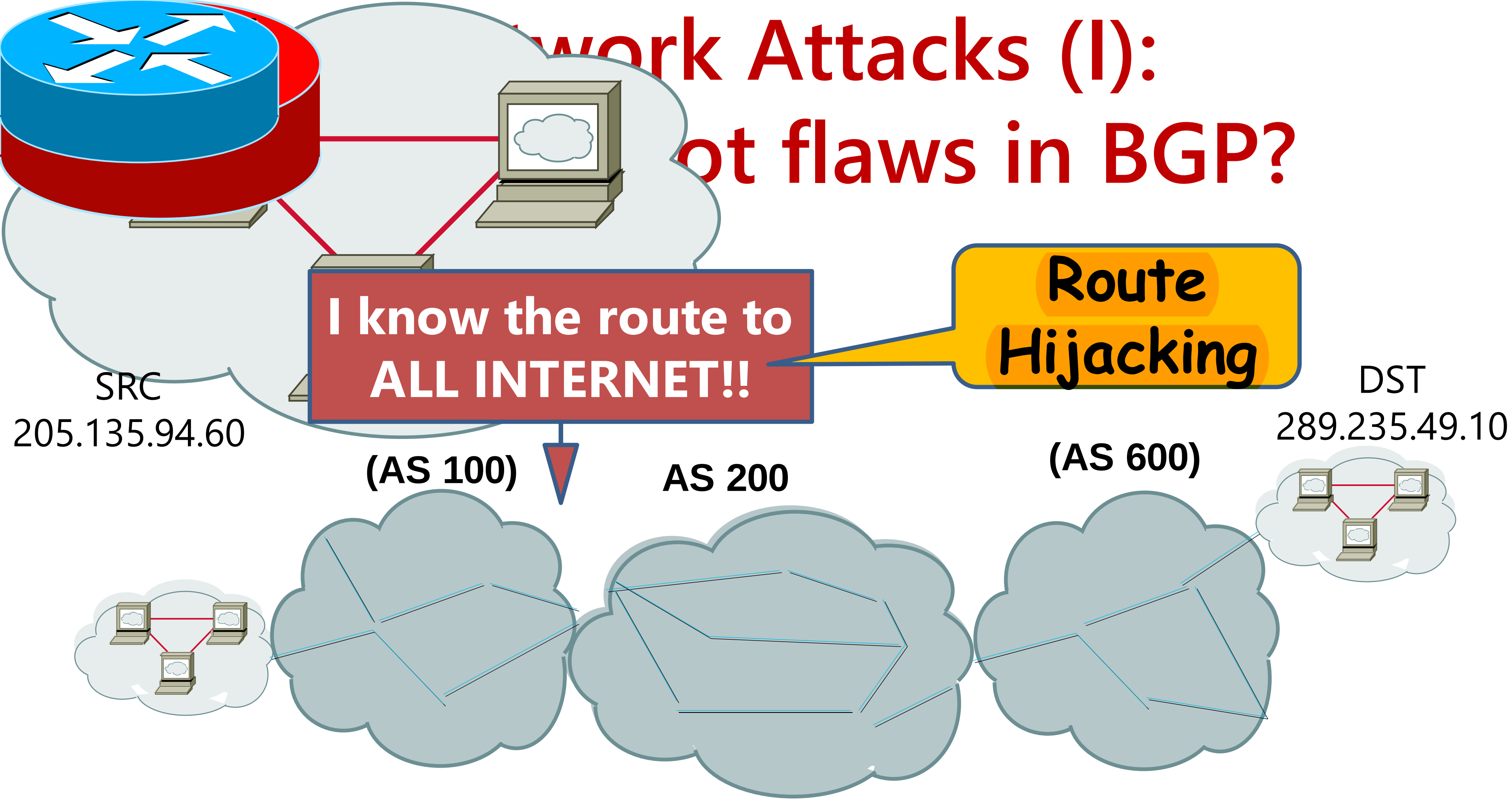


attackers do not need to be on the path of the network

Intercept Traffic between Alice and ANY website!

Network Attacks:
BGP

work Attacks (I): ot flaws in BGP?

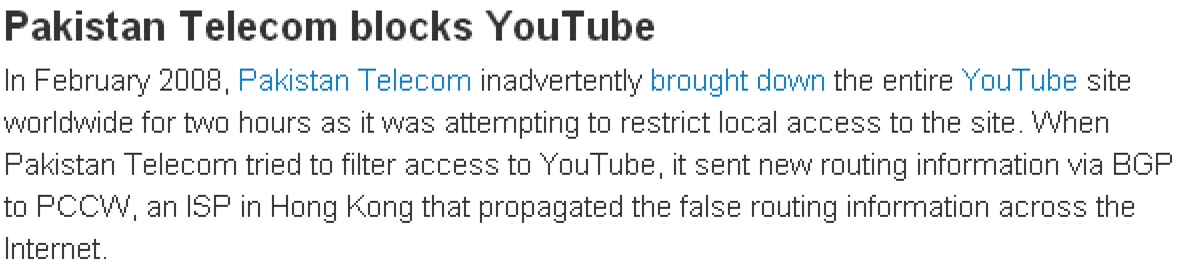


Swamp a BGP link
& force traffic via AS200

Re-advertise
Withdrawn routes

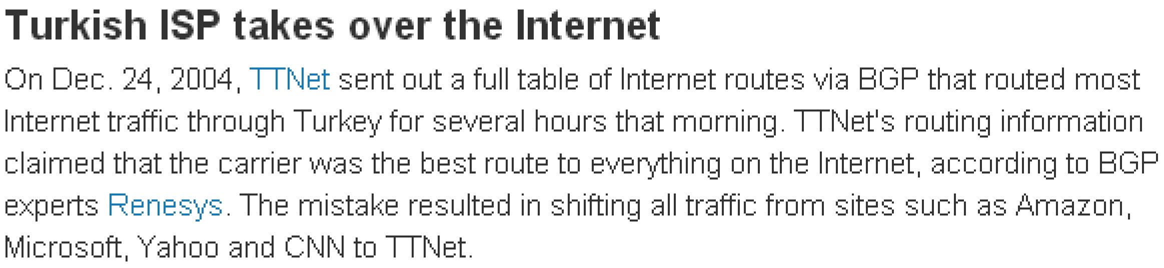
Figure from Boneh et. al.

Route Hijacking BGP Exploits in the Wild



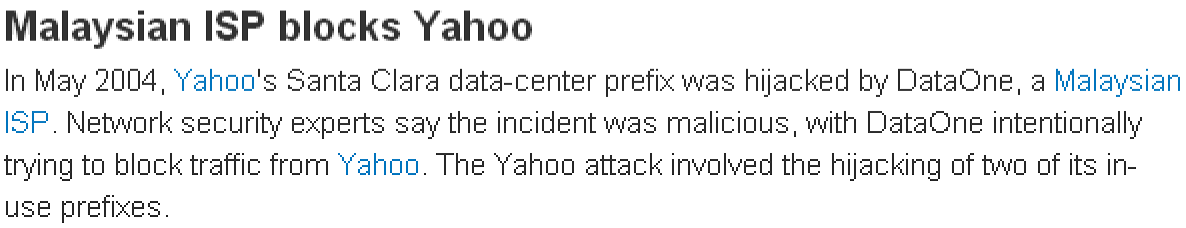
Pakistan Telecom blocks YouTube

In February 2008, Pakistan Telecom inadvertently brought down the entire YouTube site worldwide for two hours as it was attempting to restrict local access to the site. When Pakistan Telecom tried to filter access to YouTube, it sent new routing information via BGP to PCCW, an ISP in Hong Kong that propagated the false routing information across the Internet.



Turkish ISP takes over the Internet

On Dec. 24, 2004, TTNet sent out a full table of Internet routes via BGP that routed most Internet traffic through Turkey for several hours that morning. TTNet's routing information claimed that the carrier was the best route to everything on the Internet, according to BGP experts Renesys. The mistake resulted in shifting all traffic from sites such as Amazon, Microsoft, Yahoo and CNN to TTNet.

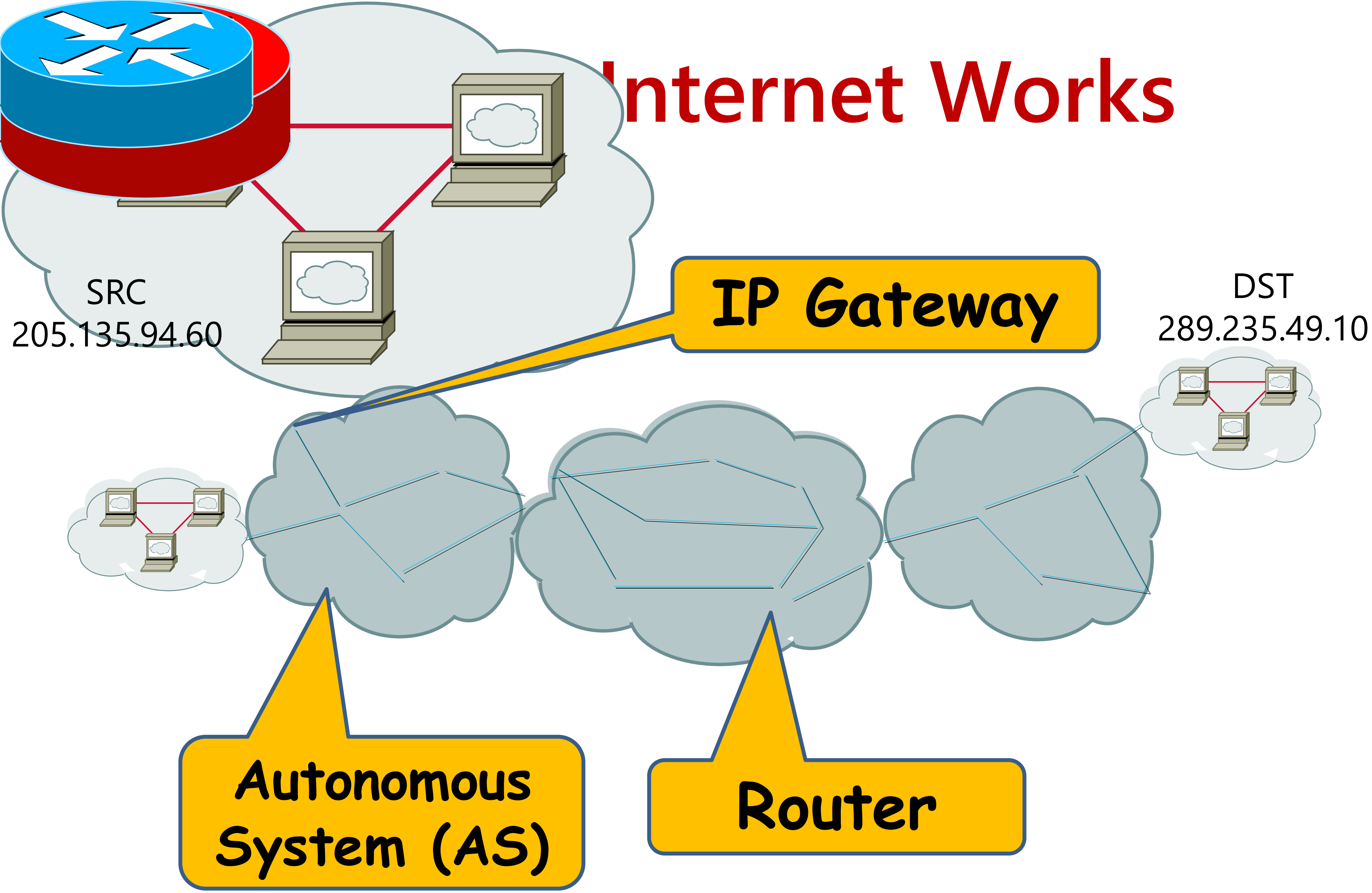


Malaysian ISP blocks Yahoo

In May 2004, Yahoo's Santa Clara data-center prefix was hijacked by DataOne, a Malaysian ISP. Network security experts say the incident was malicious, with DataOne intentionally trying to block traffic from Yahoo. The Yahoo attack involved the hijacking of two of its in-use prefixes.

Network Attacks:
IP

Internet Works



SRC
205.135.94.60

IP Gateway

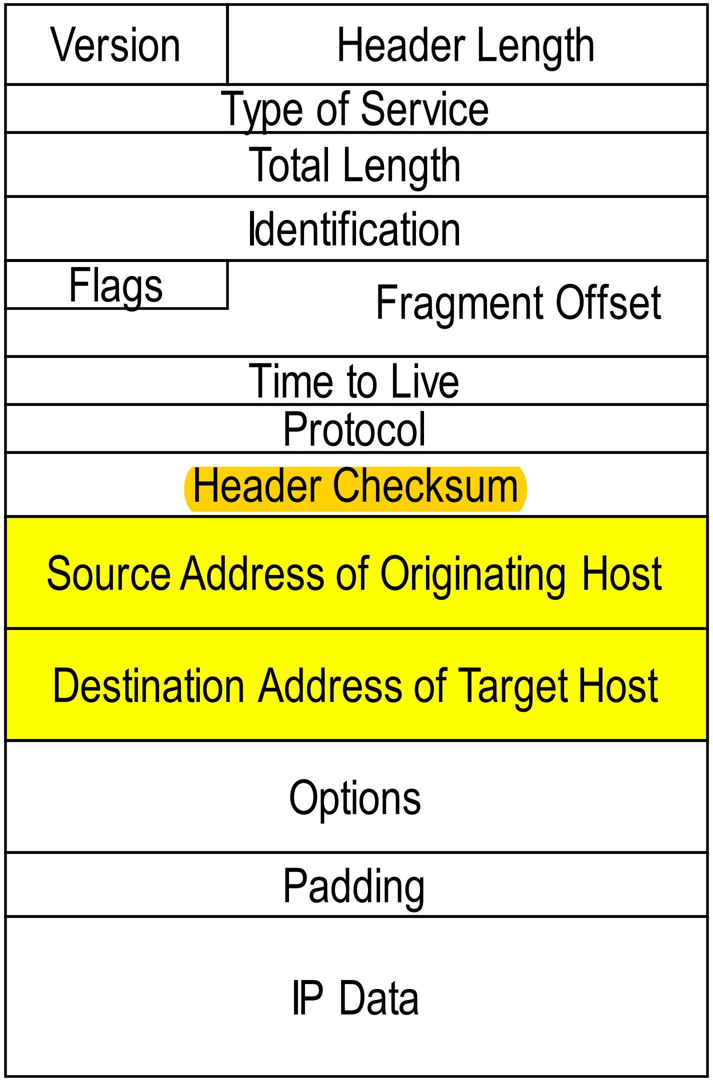
DST
289.235.49.10

Autonomous System (AS)

Router

Figure from Boneh et al.

Can you Identify IP Protocol Flaws?



Version

Header Length

Type of Service

Total Length

Identification

Flags

Fragment Offset

Time to Live

Protocol

Header Checksum

Source Address of Originating Host

Destination Address of Target Host

Options

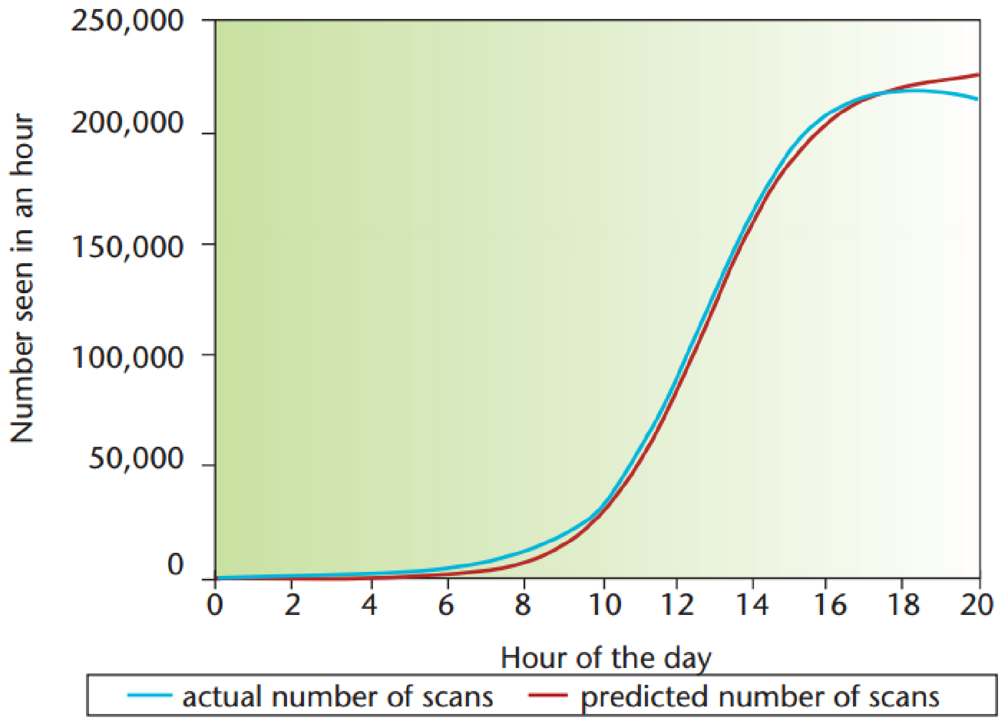
Padding

IP Data

IP Packet Format

- Confidentiality Attacks?
  - Packet sniffing
- Integrity Attacks
  - IP Data pollution
  - Source IP forgery
    - Useful for DDoS
    - Anonymous Infection (e.g. Slammer worm)

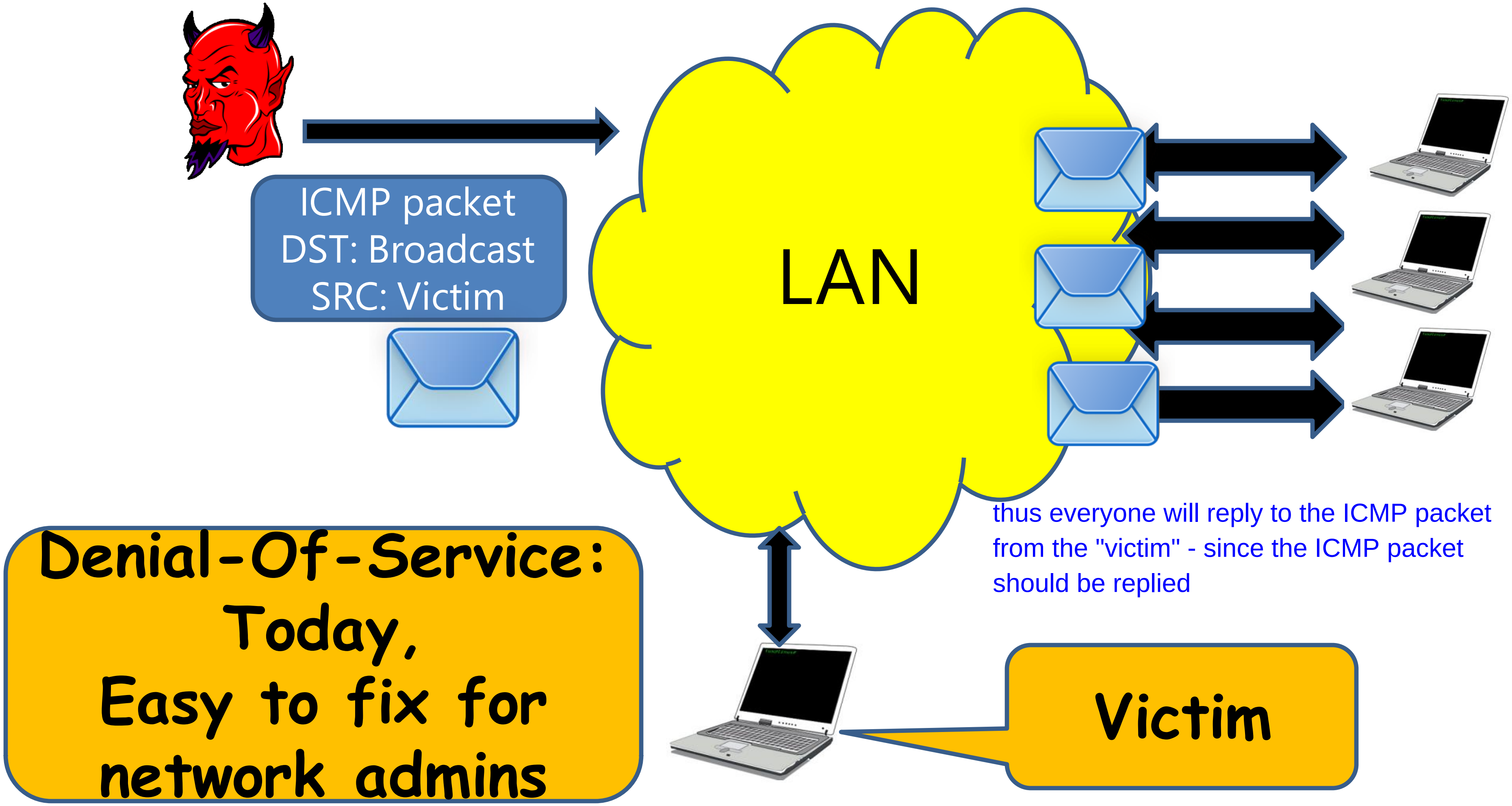
Example Src IP forgery attacks: Slammer Worm



"Traceless" or Anonymous Infection:
SRC field random

Inside the Slammer Worm, IEEE S&P 2003

Example Src IP forgery attacks: Smurf Attacks



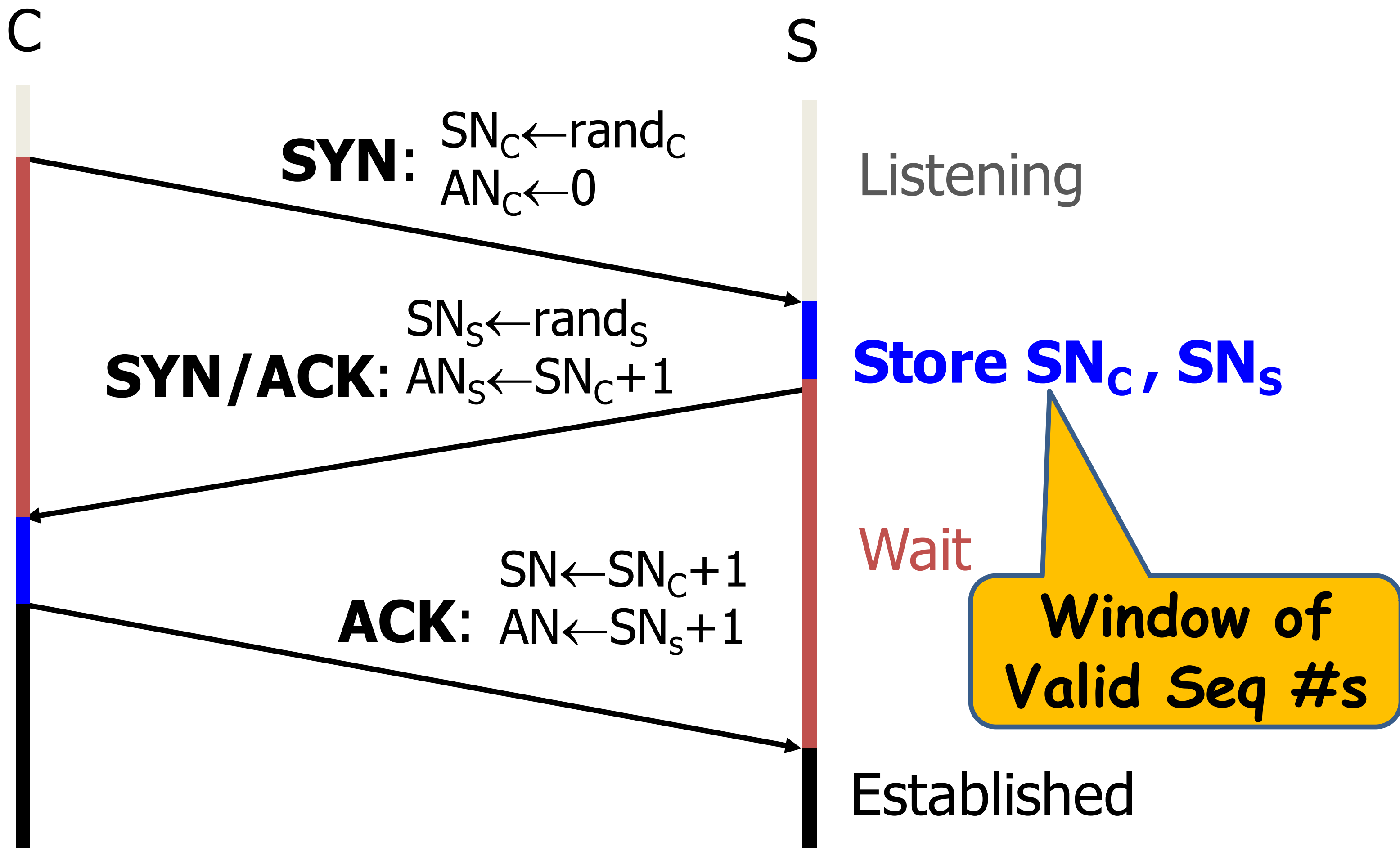
ICMP packet
DST: Broadcast
SRC: Victim

thus everyone will reply to the ICMP packet from the "victim" - since the ICMP packet should be replied

Victim

Denial-Of-Service:
Today,
Easy to fix for network admins

Network Attacks:
TCP



C

S

SYN: $SN_C \leftarrow rand_C$
$AN_C \leftarrow 0$

Listening

$SN_S \leftarrow rand_S$
SYN/ACK: $AN_S \leftarrow SN_C+1$

Store SN_C , SN_S

Wait

Window of Valid Seq #s

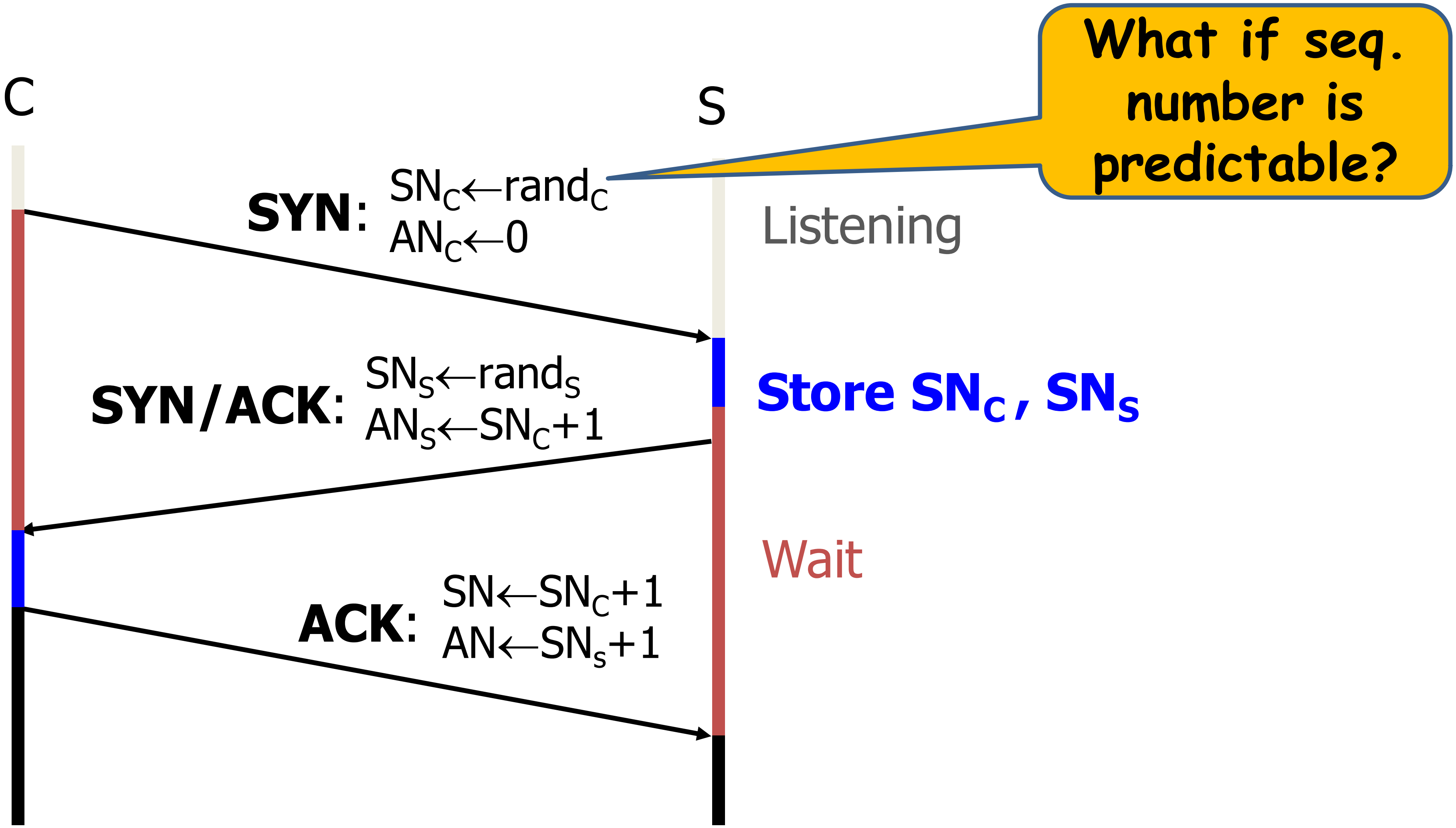
$SN \leftarrow SN_C+1$
ACK: $AN \leftarrow SN_s+1$

Established

TCP Handshake

Figure from Boneh et. al.

A Classical Attack on TCP



What if seq. number is predictable?

C

S

SYN: $SN_C \leftarrow rand_C$ $AN_C \leftarrow 0$

Listening

SYN/ACK: $SN_S \leftarrow rand_S$ $AN_S \leftarrow SN_C+1$

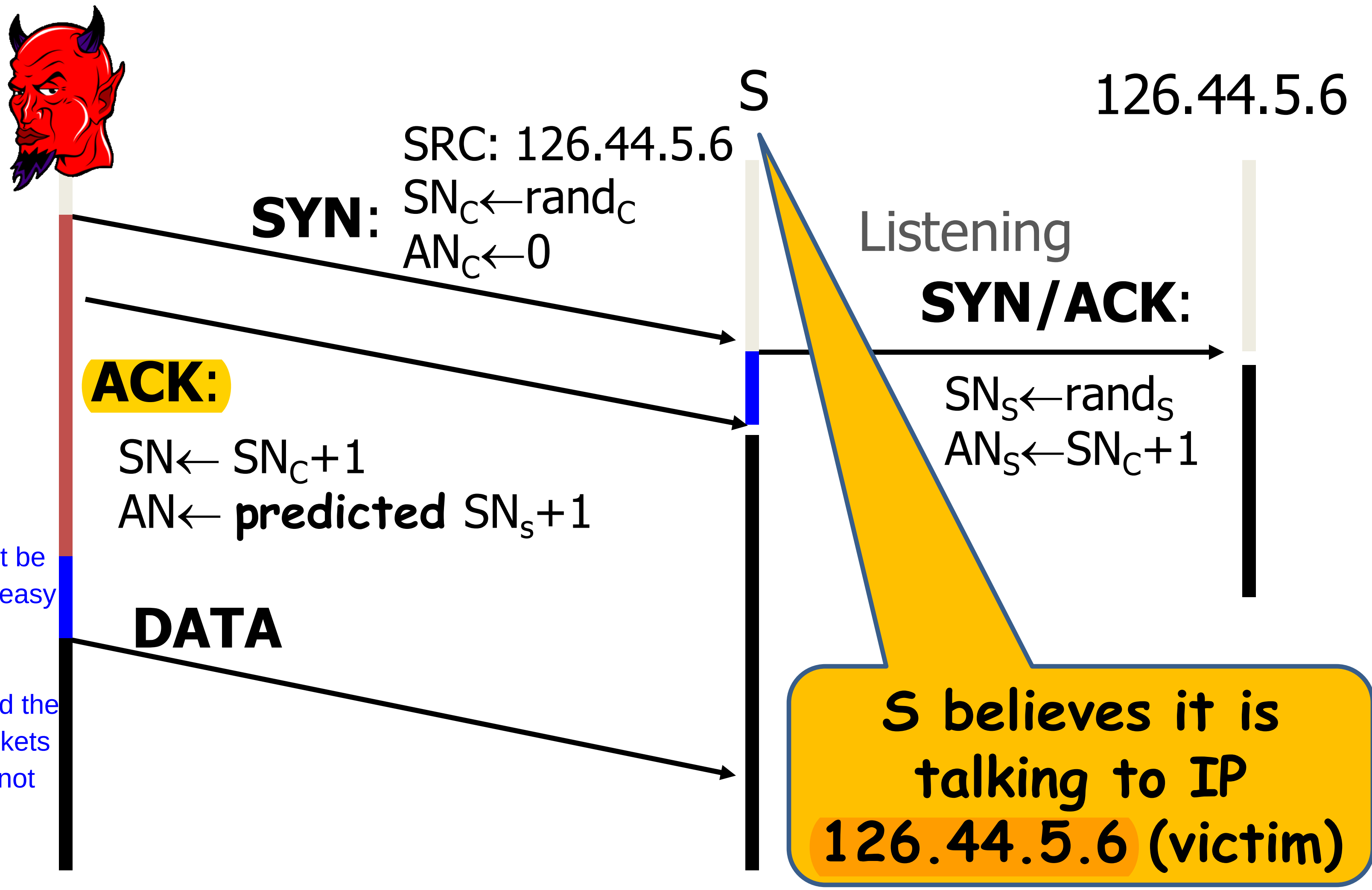
Store SN_C , SN_S

Wait

ACK: $SN \leftarrow SN_C+1$ $AN \leftarrow SN_s+1$

Classical Attacks on TCP:

Sequence Number Prediction



S

126.44.5.6

SYN: SRC: 126.44.5.6
$SN_C \leftarrow rand_C$
$AN_C \leftarrow 0$

Listening

SYN/ACK:
$SN_S \leftarrow rand_S$
$AN_S \leftarrow SN_C+1$

ACK:
$SN \leftarrow SN_C+1$
$AN \leftarrow$ **predicted** SN_s+1

DATA

S believes it is talking to IP 126.44.5.6 (victim)

last time the AN might be only 16 bits and thus easy to brute force

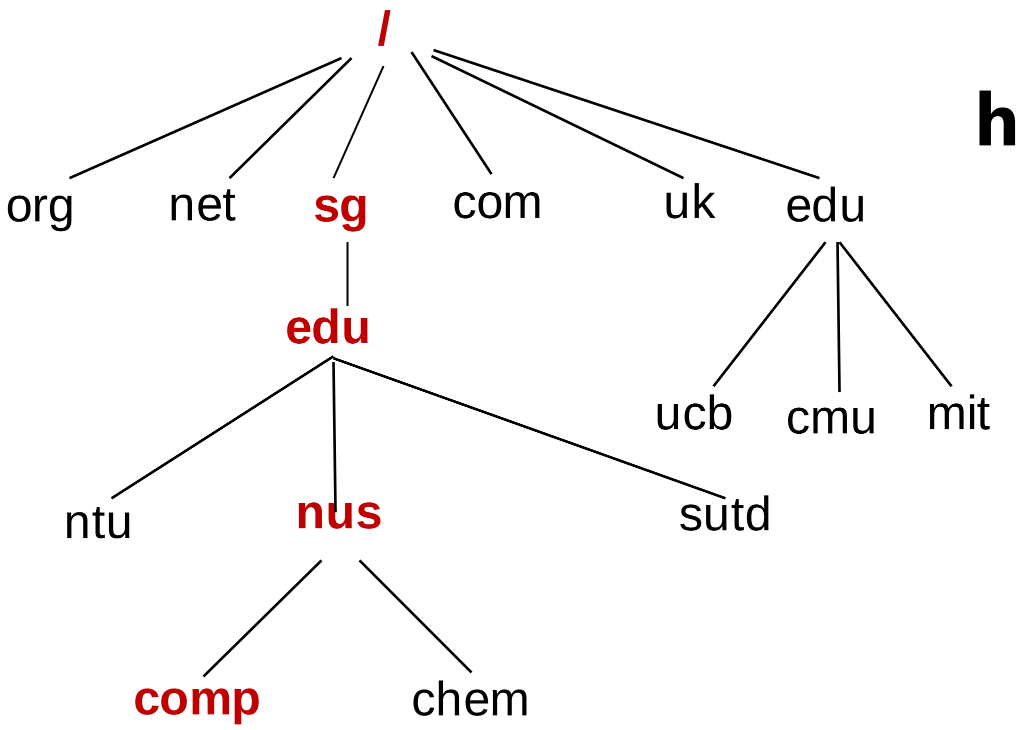
- now only 32 bits, and the speed of sending packets now might make this not secured

Predictable SNs -> Break IP-based Authentication!

- First found in 1985, in BSD Unix Systems [Morris'85]
- IP-based authentication is common
  - (and weak)!
  - E.g. /etc/hosts, application layer gateways
- Personal Firewalls
- IP-based filtering on gateways/firewalls
- Custom User identification logic
- Apache

Network Attacks:
DNS

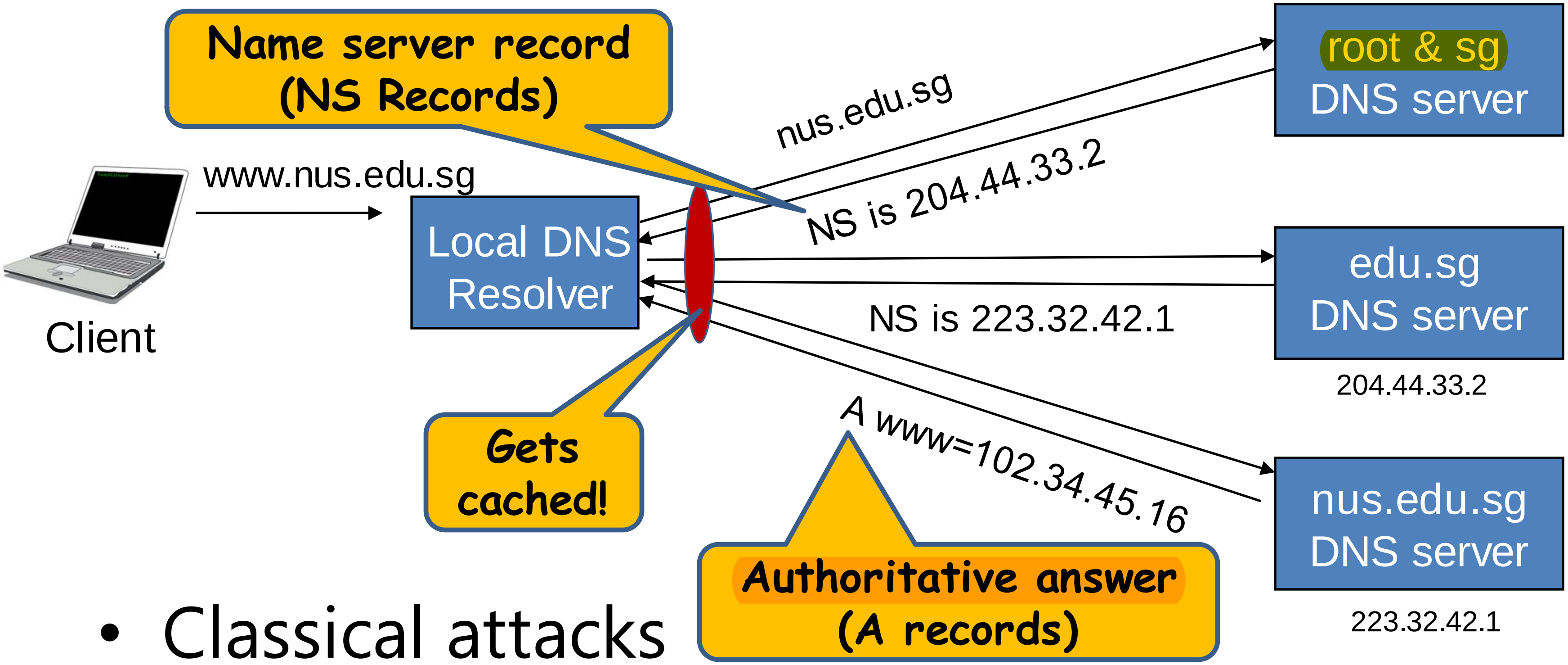
Domain Names to IP addresses: DNS



http://comp.nus.edu.sg

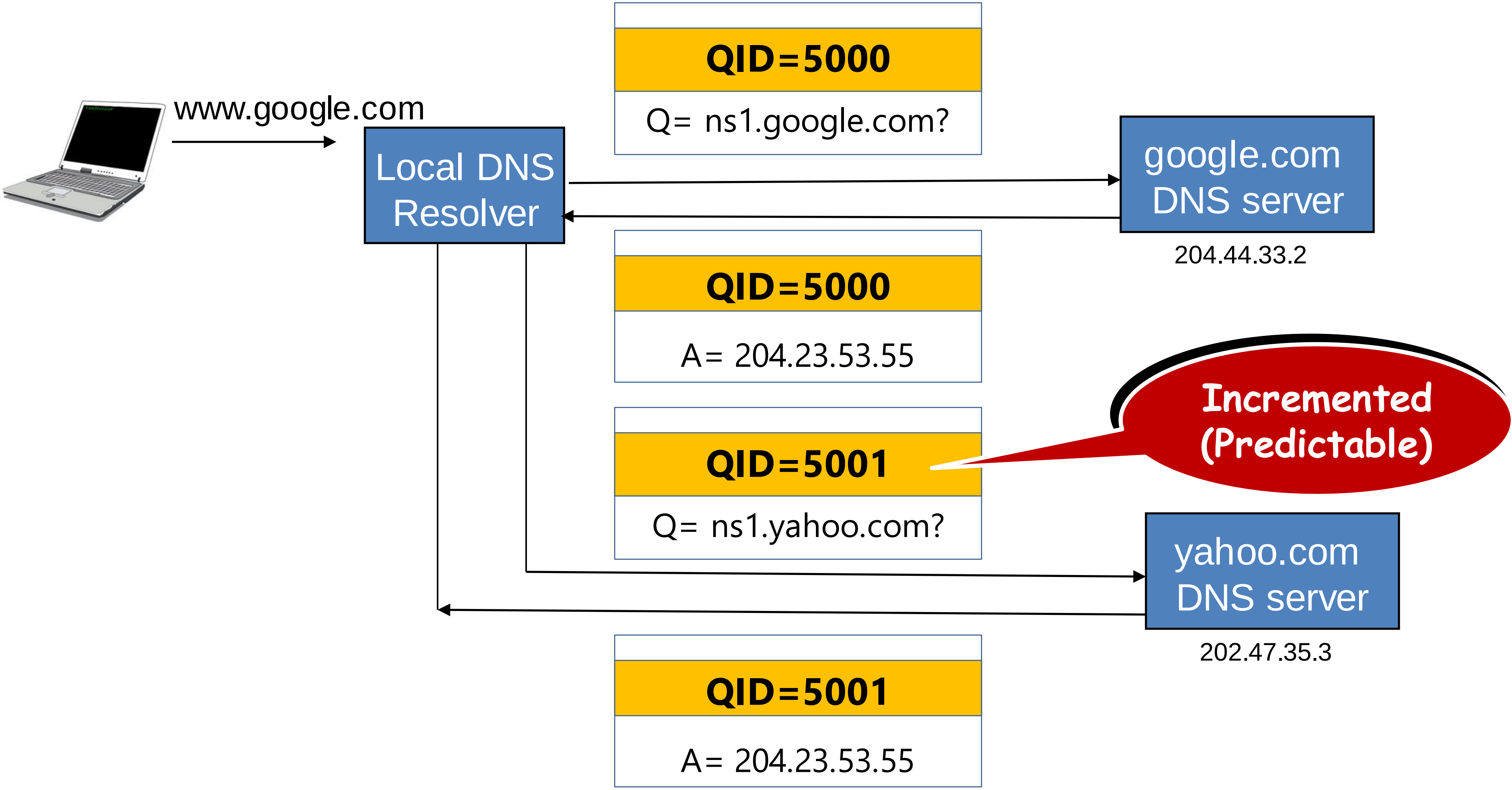
205.135.94.60

How DNS Works!



- Classical attacks
  - Modify in-transit DNS responses
  - Find software vulnerabilities in DNS software
- A more advanced (and easy) attack!
- Let’s see how you can own *.google.com !

DNS Resolution:
A Bit of Implementation Detail



www.google.com

Local DNS Resolver

QID=5000

Q= ns1.google.com?

google.com DNS server

204.44.33.2

QID=5000

A= 204.23.53.55

Incremented (Predictable)

QID=5001

Q= ns1.yahoo.com?

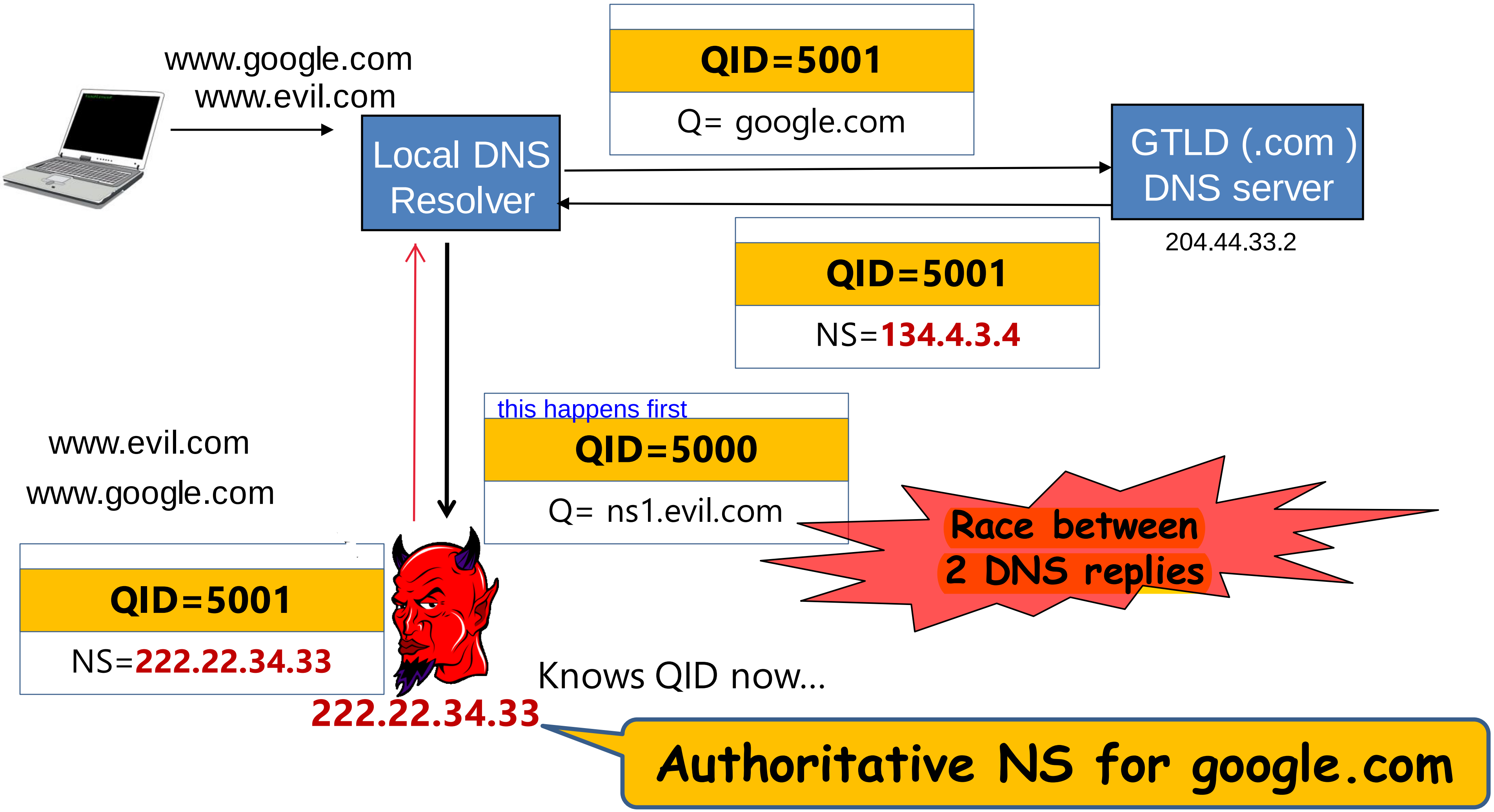
yahoo.com DNS server

202.47.35.3

QID=5001

A= 204.23.53.55

DNS Cache Poisoning [Kaminsky'08]



www.google.com
www.evil.com

Local DNS Resolver

QID=5001
Q= google.com

GTLD (.com) DNS server
204.44.33.2

QID=5001
NS=**134.4.3.4**

this happens first
QID=5000
Q= ns1.evil.com

Race between 2 DNS replies

www.evil.com
www.google.com

QID=5001
NS=**222.22.34.33**

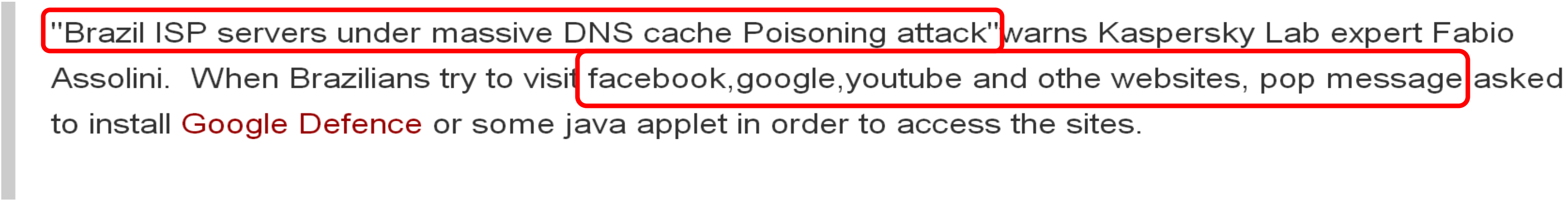
222.22.34.33

Knows QID now…

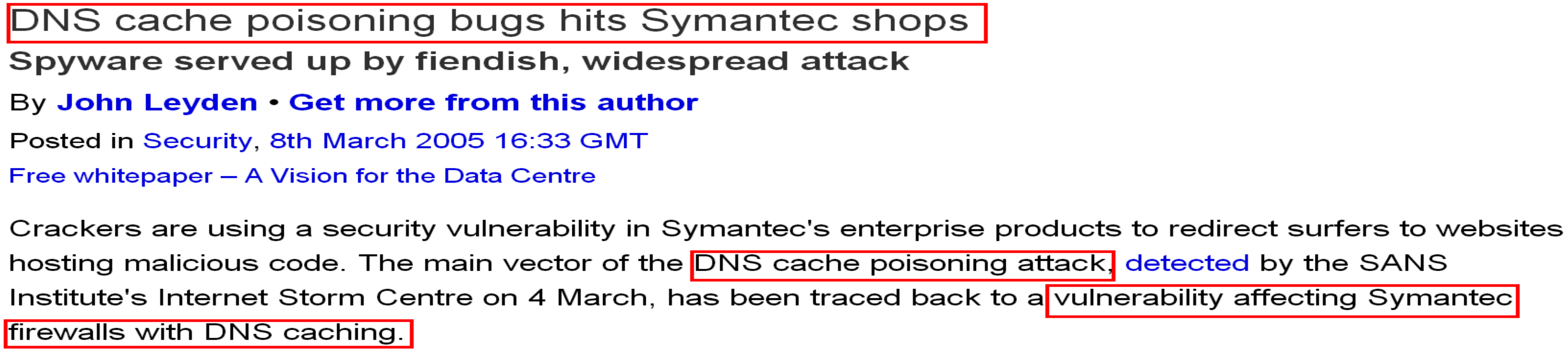
Authoritative NS for google.com

DNS Attacks (I): DNS Cache Poisoning

- Is an example of “DNS pharming” attacks
- DNS pharming and poisoning been actively used:



"Brazil ISP servers under massive DNS cache Poisoning attack" warns Kaspersky Lab expert Fabio Assolini. When Brazilians try to visit facebook,google,youtube and othe websites, pop message asked to install Google Defence or some java applet in order to access the sites.



DNS cache poisoning bugs hits Symantec shops

Spyware served up by fiendish, widespread attack

By **John Leyden • Get more from this author**

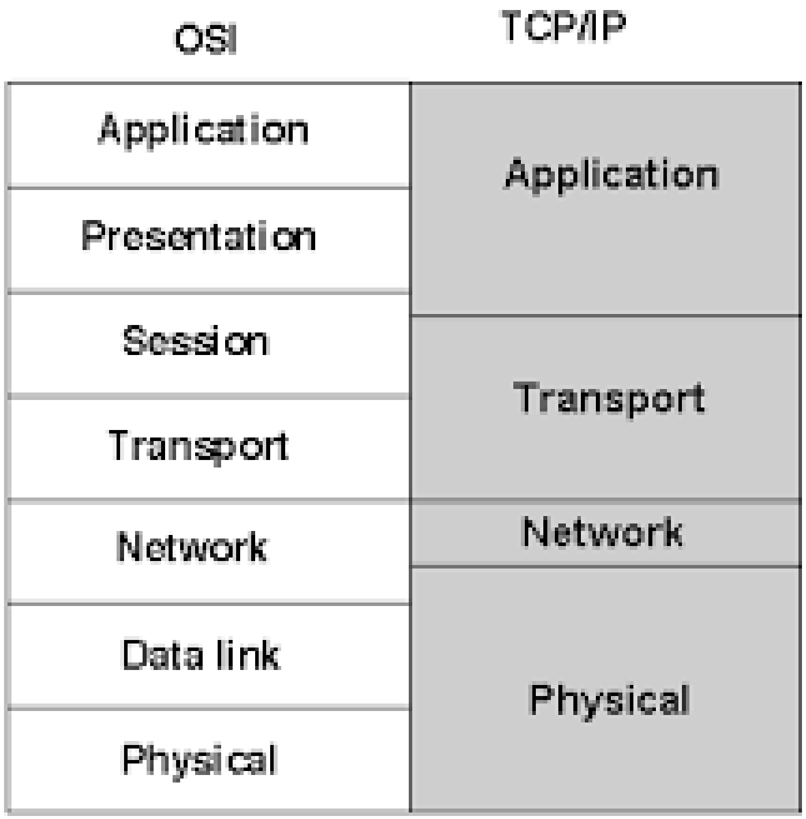
Posted in Security, 8th March 2005 16:33 GMT

Free whitepaper – A Vision for the Data Centre

Crackers are using a security vulnerability in Symantec's enterprise products to redirect surfers to websites hosting malicious code. The main vector of the DNS cache poisoning attack, detected by the SANS Institute's Internet Storm Centre on 4 March, has been traced back to a vulnerability affecting Symantec firewalls with DNS caching.

Summary of Network Attacks

- **The Internet Architecture**
  - BGP
  - ARP
  - IP
  - TCP
  - DNS



… Was Not Designed With Security In Mind!

Additional References

- [Optional] Detailed References:
  - A Study of Prefix Hijacking and Interception in the Internet (Ballani et. al., SIGCOMM 2007)
  - Security Problems in the TCP/IP Protocol Suite (by S.M. Bellovin, SIGCOMM CCR 1989)
  - The Hitchhiker's Guide to DNS Cache Poisoning (Son and Shmatikov, SecureComm 2010)

Break!

Network-level Defense: Firewalls

Firewalls

- Firewalls are tools that control the flow of traffic going between networks.
  - Sitting at border between networks
  - Looking at services, addresses, data, and etc. of traffic
  - Deciding whether a packet should be **allowed or dropped** based on a firewall **policy**

- Network firewalls operate at the TCP / IP Level

- Application-layer firewalls operate to higher layers
  - Network Intrustion Detections systems (NIDS) and Intrusion Prevention systems (IPS) work on same principle

Basic Firewalls: Stateless Packet Filters

- Applies rules to packets in/out of firewall
  - Based on information in packet header like src/dest IP addr & port, IP protocol, interface
- Typically a list of rules of matches on fields
  - if match rule says if allow or deny packets
- Two default policies:
  - Deny - prohibit unless explicitly permitted
    - more conservative, controlled, visible to users
  - Allow - permit unless explicitly prohibited
    - easier to manage/use but less secure

- Good Design Principle: **Default fail-close** policy
  - Don’t open a service to public unless it is necessary

safer to know what is good to allow than what is bad to block - since most of the time we would not know what are all the attacks

Stateless Packet Filter Rules

- Firewall uses a list of filter rules to decide what to do with a packet
- For a packet, firewall apply the rules starting from the top of the list, and execute the operation specified by the first matching rule
- Sample rule format:

Action	Src Addr	Dst Addr	Protocol	Src Port	Dst Port	Ctrl-Bit

Stateless Packet Filter Rules: Example

- A company only allows connections to port 80 (HTTP) of external hosts
  - Internal address: 1.2.3.*

Action	Src Addr	Dst Addr	Protocol	Src Port	Dst Port	Ctrl-Bit
Allow	1.2.3.*	*	TCP	*	80	*
Allow	*	1.2.3.*	TCP	80	>1023	ACK
Deny	*	*	*	*	*	*

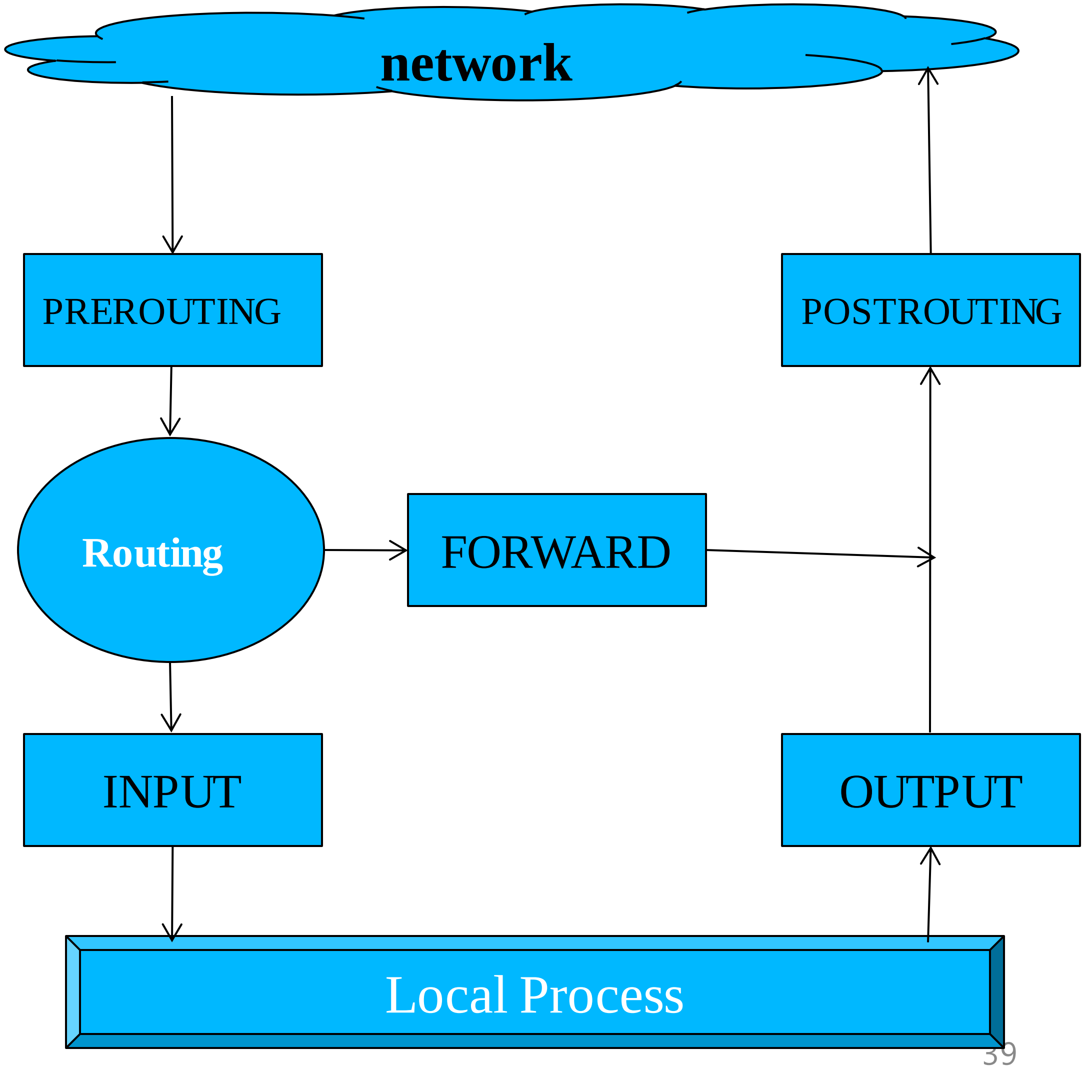
Reference: How to block SQL Slammer Worm with firewall signatures

Types of Firewalls / NIDS / IPS

- Traditional / Stateless Packet Filters
  - Applying rules to packets in/out of firewall
  - Based on information in packet header
- Stateful Packet Filters
  - Maintaining a state table of all active connections
  - Filtering packets based on connection states
- Proxy-based or Application Firewalls
  - Understanding application logic
  - Acting as a relay of application-level traffic

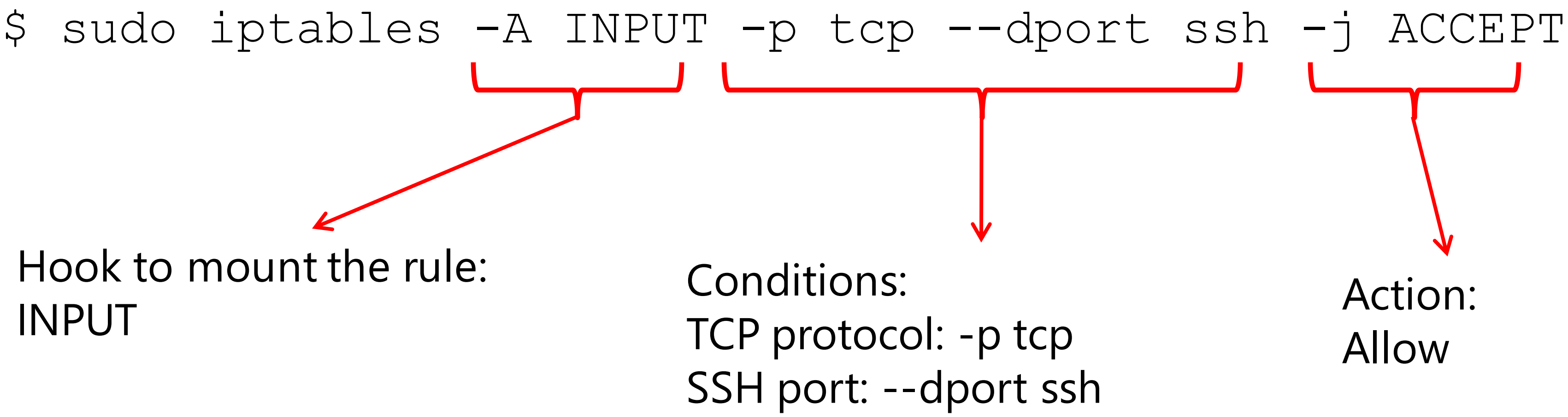
Linux Firewalls: Netfilter Framework

- Netfilter is the packet filtering framework in Linux kernel
- Five hooks to decide the fate of a packet:
  - Prerouting
  - Postrouting
  - Forward
  - Input
  - Output



The iptables Utility

- The Linux program to maintain firewall rules in the Netfilter framework
- Example: allowing ssh connections to this computer



```
$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

Hook to mount the rule:
INPUT

Conditions:
TCP protocol: -p tcp
SSH port: --dport ssh

Action:
Allow

Firewalls & IDS / IPS: Threat Model

- Goal:
  - Stop attacker's packet from reaching the end application
- Adversary Capability:
  - Adversary can send malicious network packets
  - Adversary is outside the network perimeter
- Assumptions
  - The network perimeter is correctly defined
  - The firewall is uncompromised (not buggy)
  - The firewall sees the same data as end application
  - Defender's policy can tell bad from good traffic by inspecting packet content

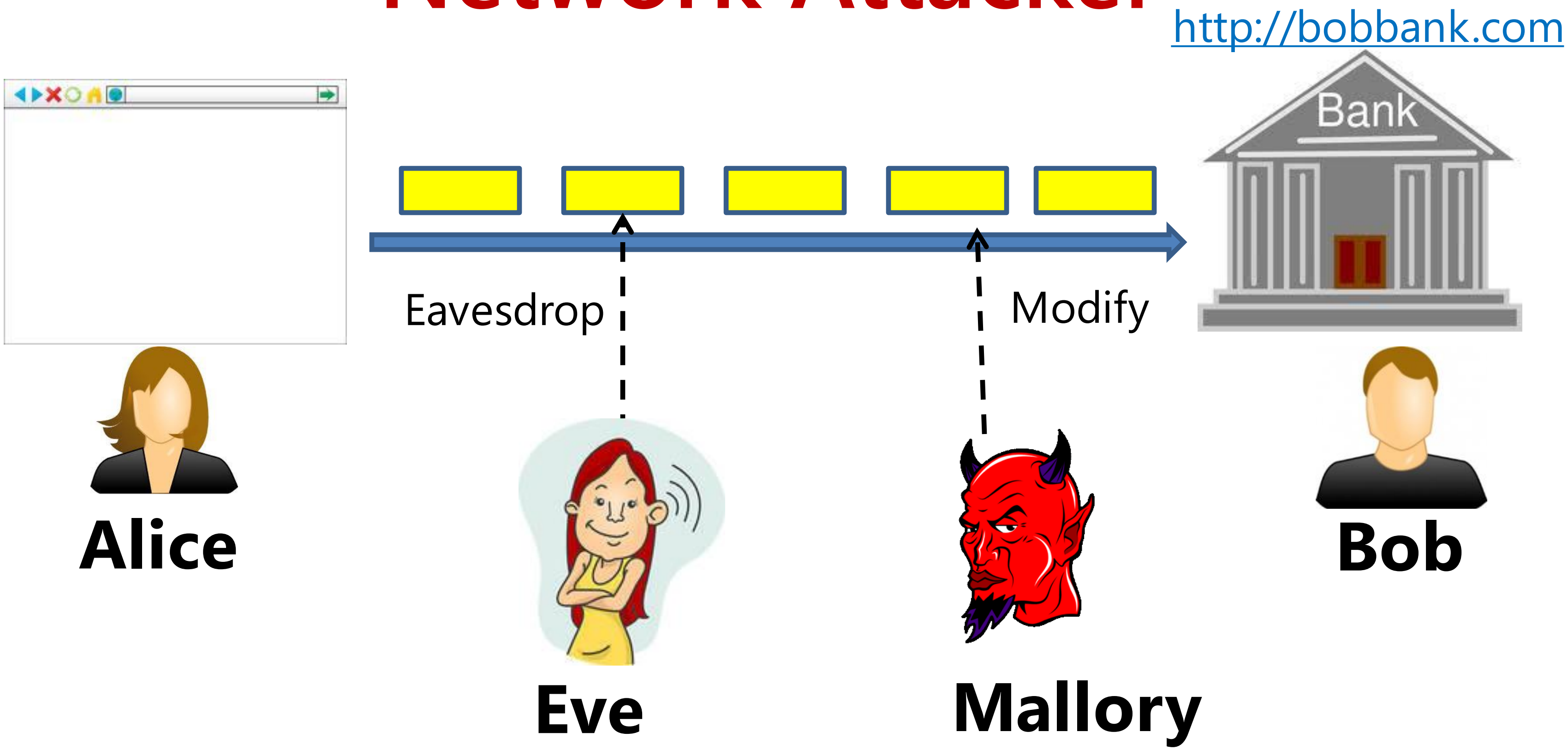
Firewalls & IDS / IPS: Weaknesses of Threat Model

- Weak adversary in the model
  - The defender needs to knows **specific attack patterns** and sets the policy accordingly
  - Easy for attackers to evade those specific attack patterns

- It is easy to violate the assumptions!
  - Physically Compromise the firewall
  - Semantic Gap between firewall vs. the end application
    - Difficult to ascertain which service is targeted from inspecting network flows only
    - Firewall network code may differ from host OS / app code
    - Many attack (raw byte) patterns for the same exploit!
  - Thwarted completely by encrypted traffic (e.g. HTTPS)
  - Is there a n/w perimeter? "Bring your own device" problem
    - Carrying data on a smartphone from outside the network

Cryptographic Secure Channels:
Threat Model

Definition: Network Attacker

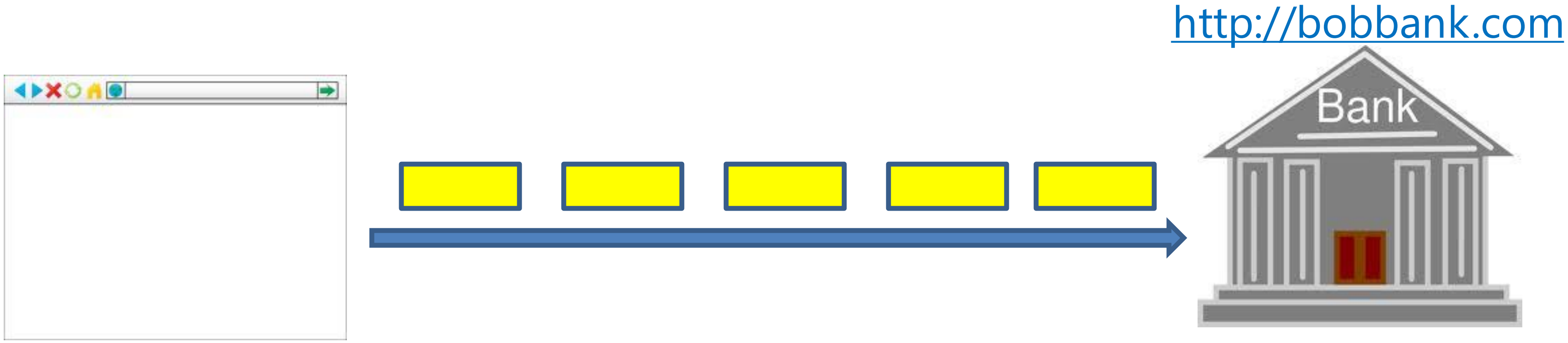


Intercept ALL Traffic between Alice and Bob!

- Eve is assumed to only eavesdrop on traffic
- Mallory can listen and tamper with traffic

Not an attack-specific defn.

Definition:
Properties of Secure Channel



A Secure Channel is a data communication protocol established **between 2 programs** which preserves data:

- **Confidentiality**
- **Integrity**
- **Authentication**

against a computationally-bounded "network attacker" [Dolev-Yao-1983]

* Note that availability is not a goal. So, denial-of-service attacks are permitted by the threat model.

* Integrity is also referred to as "authenticity" sometimes. Not to be confused with "authentication"

Cryptographic Secure Channels:

To be continued next week…

Summary

- Network Layer Attacks: BGP, IP, TCP,DNS,…
- Basic Defense: Network Firewalls
- Segway to Secure Channels

Thanks!

See you next week…