Network Security Protocols and Defensive Mechanisms

John Mitchell

Plan for today

- Network protocol security
  - IPSEC
  - BGP instability and S-BGP
  - DNS rebinding and DNSSEC
  - Wireless security – 802.11i/WPA2
- Standard network perimeter defenses
  - Firewall
    - Packet filter (stateless, stateful), Application layer proxies
  - Traffic shaping
  - Intrusion detection
    - Anomaly and misuse detection

© art.com

Dan's lecture last Thursday

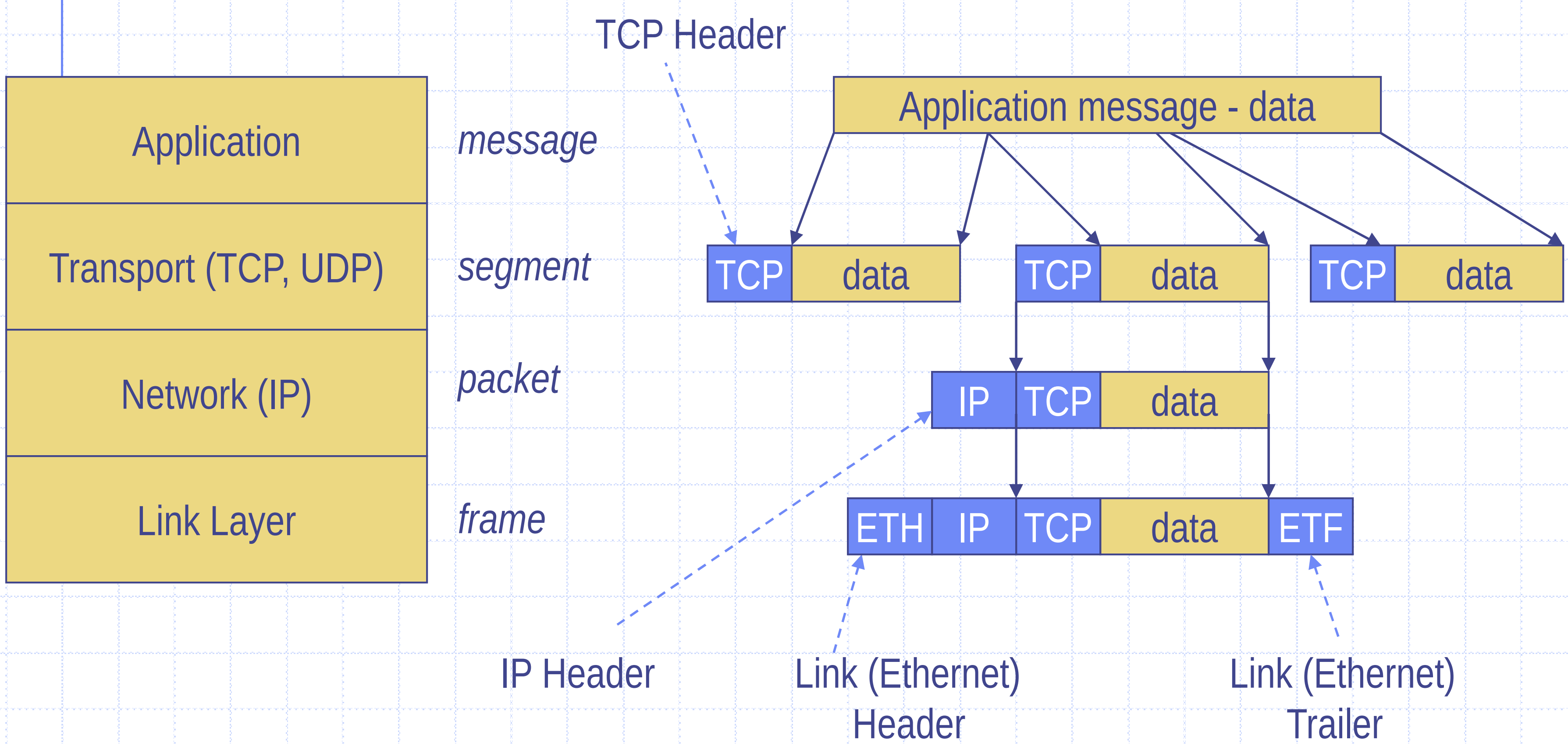
- Basic network protocols
  - IP, TCP, UDP, BGP, DNS
- Problems with them
  - TCP/IP
    - No SRC authentication: can't tell where packet from
    - Packet sniffing
    - Connection spoofing, sequence numbers
  - BGP: advertise bad routes or close good ones
  - DNS: cache poisoning, rebinding

(out of time; cover today)

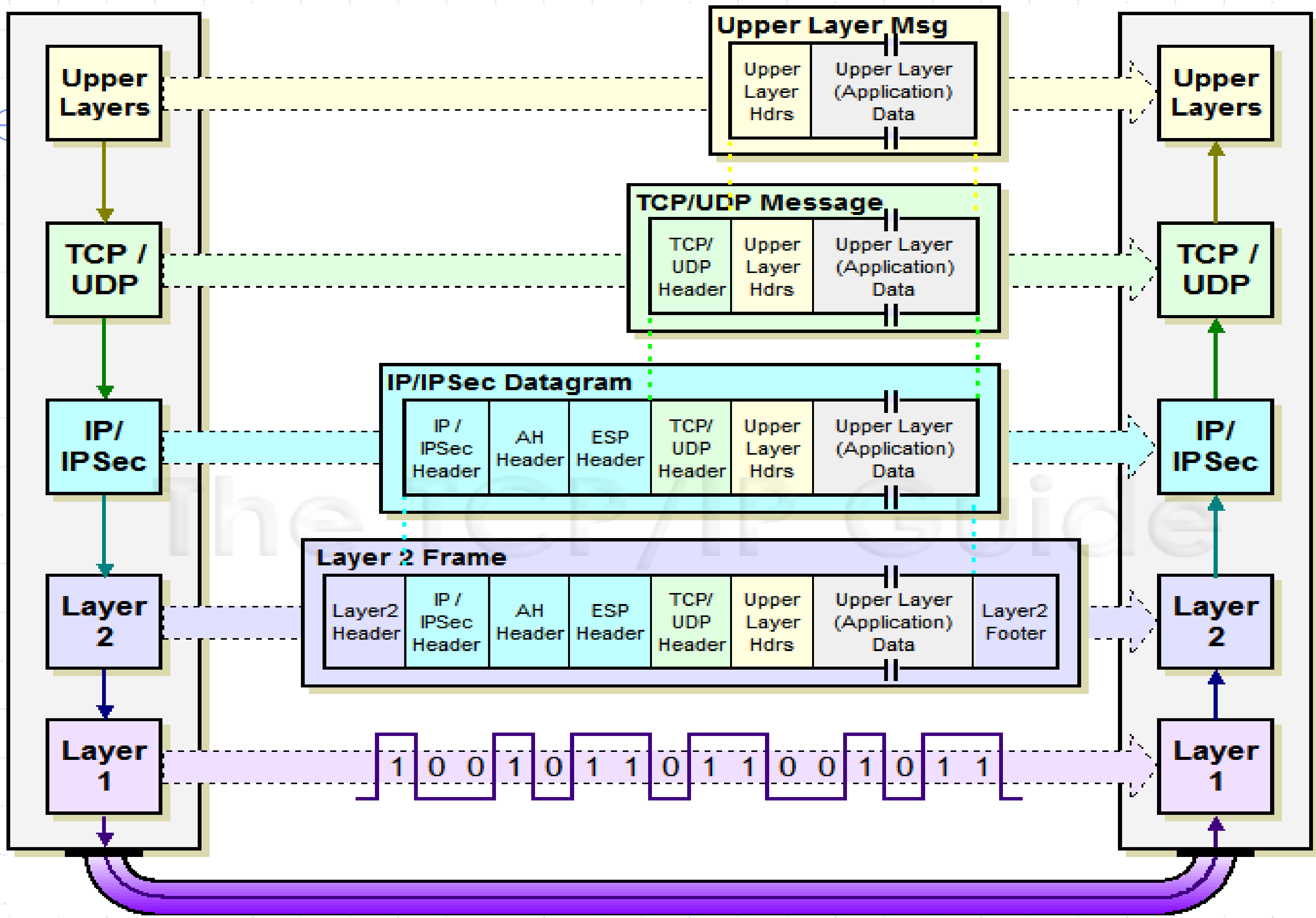
IPSEC

- Security extensions for IPv4 and IPv6
- IP Authentication Header (AH)
  - Authentication and integrity of payload and header
- IP Encapsulating Security Protocol (ESP)
  - Confidentiality of payload
- ESP with optional ICV (integrity check value)
  - Confidentiality, authentication and integrity of payload

Recall packet formats and layers

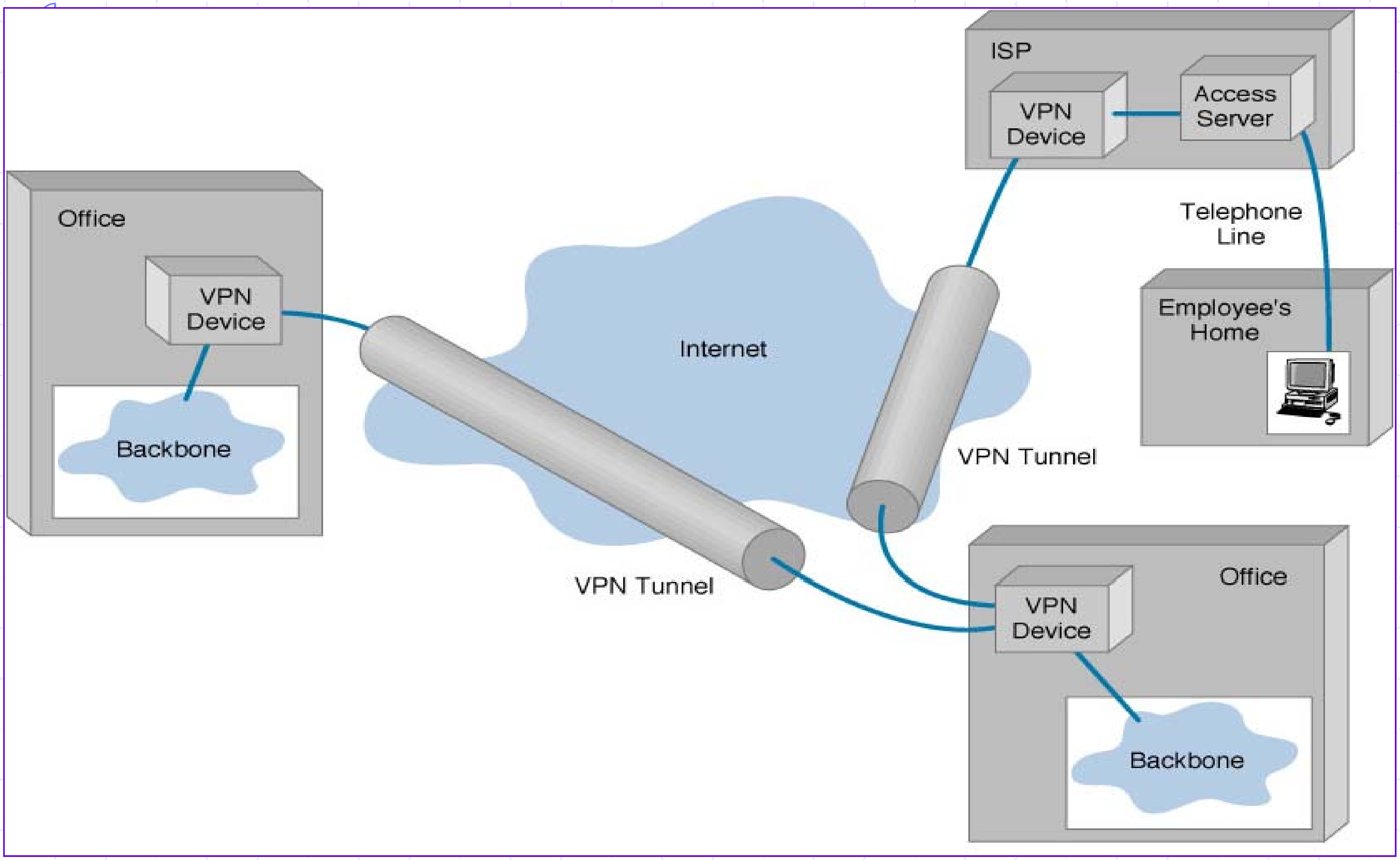


IPSec Transport Mode: IPSEC instead of IP header

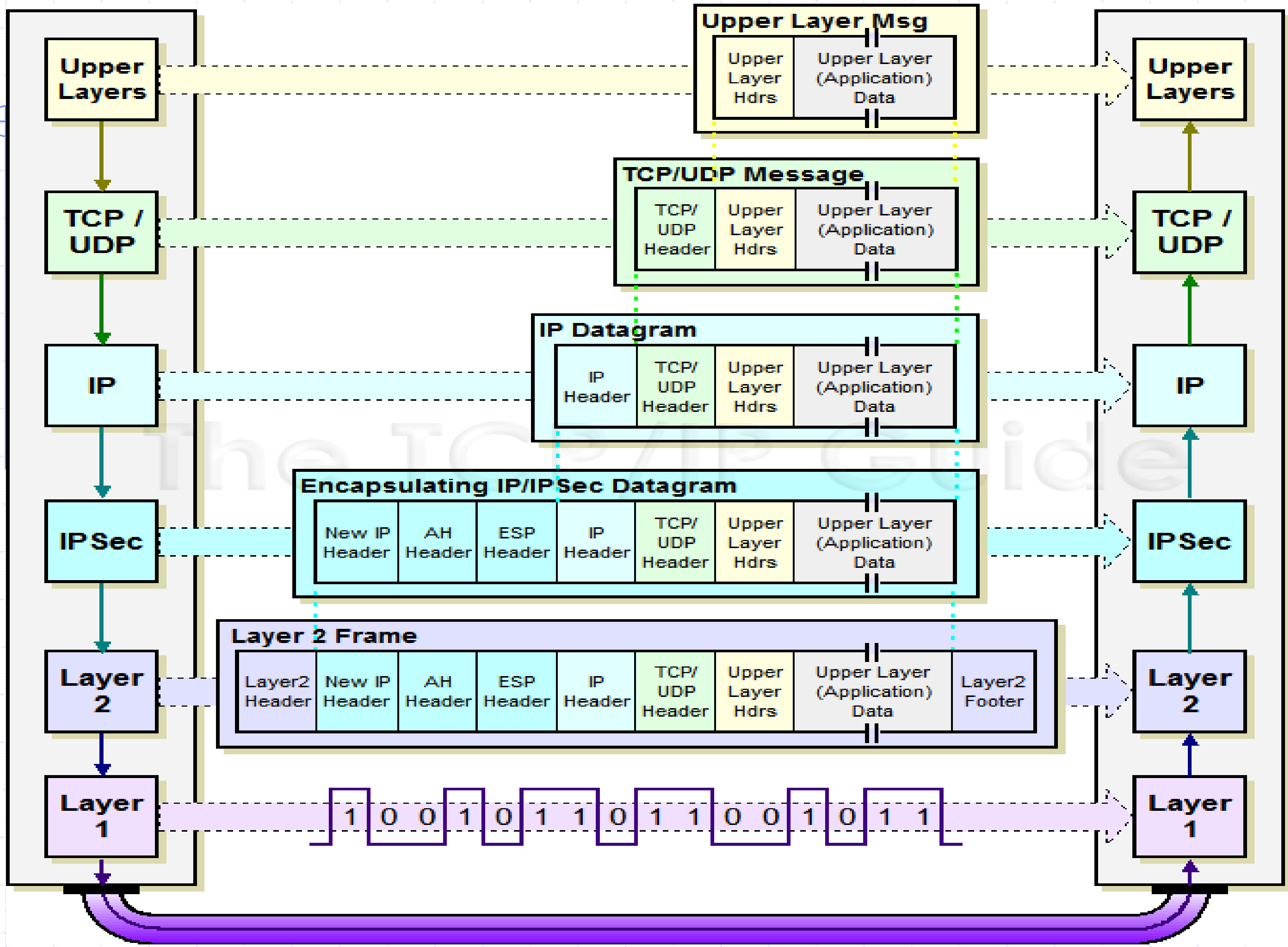


http://www.tcpipguide.com/free/t_IPSecModesTransportandTunnel.htm

IPSEC Tunnel Mode



IPSec Tunnel Mode: IPSEC header + IP header

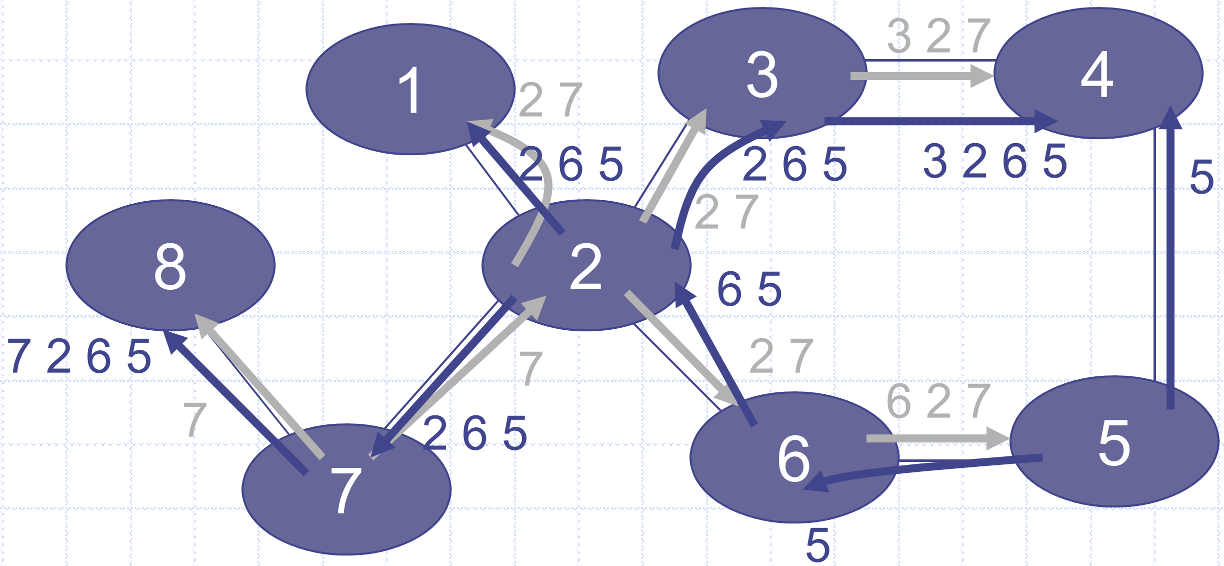


VPN

- Three different modes of use:
  - Remote access client connections
  - LAN-to-LAN internetworking
  - Controlled access within an intranet
- Several different protocols
  - PPTP – Point-to-point tunneling protocol
  - L2TP – Layer-2 tunneling protocol
  - IPsec (Layer-3: network layer)

Data layer

BGP example



- Transit: 2 provides transit for 7
- Algorithm seems to work OK in practice
  - BGP is does not respond well to frequent node outages

Figure: D. Wetherall

BGP Security Issues

- BGP is the basis for all inter-ISP routing
- Benign configuration errors affect about 1% of all routing table entries at any time
- The current system is highly vulnerable to human errors, and a wide range of malicious attacks
  - links
  - routers
  - management stations
- MD5 MAC is rarely used, perhaps due to lack of automated key management, and it addresses only one class of attacks

Slide: Steve Kent

S-BGP Design Overview

- IPsec: secure point-to-point router communication
- Public Key Infrastructure: authorization framework for all S-BGP entities
- Attestations: digitally-signed authorizations
  - Address: authorization to advertise specified address blocks
  - Route: Validation of UPDATEs based on a new path attribute, using PKI certificates and attestations
- Repositories for distribution of certificates, CRLs, and address attestations
- Tools for ISPs to manage address attestations, process certificates & CRLs, etc.

Slide: Steve Kent

Address Attestation

- Indicates that the final AS listed in the UPDATE is authorized by the owner of those address blocks to advertise the address blocks in the UPDATE
- Includes identification of:
  - owner's certificate
  - AS to be advertising the address blocks
  - address blocks
  - expiration date
- Digitally signed by owner of the address blocks, traceable up to the IANA via certificate chain
- Used to protect BGP from erroneous UPDATEs (authenticated but misbehaving or misconfigured BGP speakers)

Route Attestation

- Indicates that the speaker or its AS authorizes the listener's AS to use the route in the UPDATE
- Includes identification of:
  - AS's or BGP speaker's certificate issued by owner of the AS
  - the address blocks and the list of ASes in the UPDATE
  - the neighbor
  - expiration date
- Digitally signed by owner of the AS (or BGP speaker) distributing the UPDATE, traceable to the IANA ...
- Used to protect BGP from erroneous UPDATEs (authenticated but misbehaving or misconfigured BGP speakers)

Validating a Route

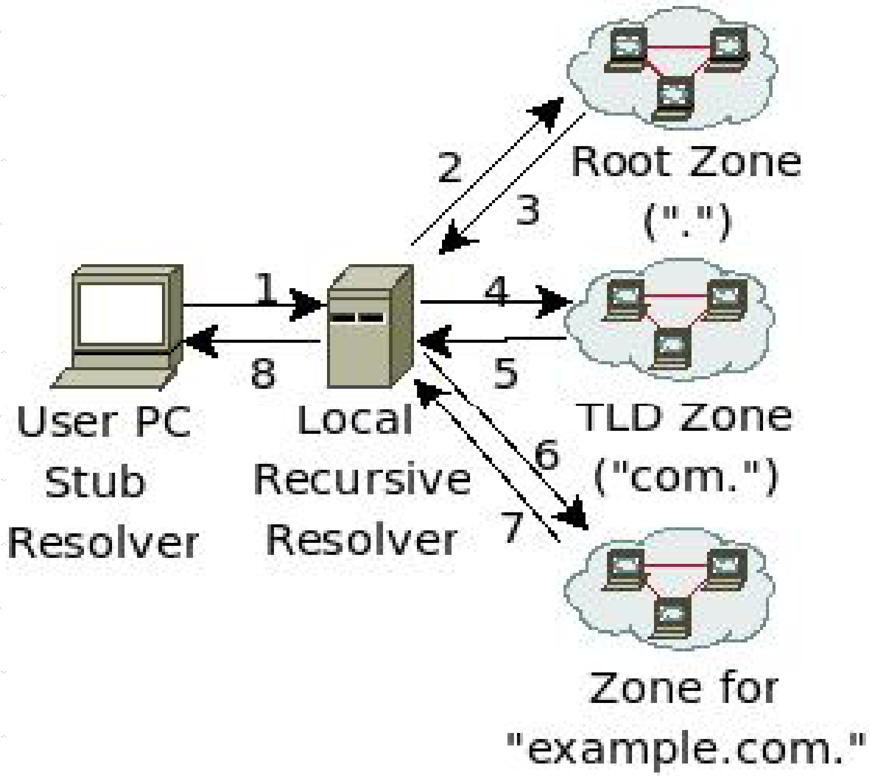
- To validate a route from AS_n, AS_{n+1} needs:
  - address attestation from each organization owning an address block(s) in the NLRI
  - address allocation certificate from each organization owning address blocks in the NLRI
  - route attestation from every AS along the path (AS_1 to AS_n), where the route attestation for AS_k specifies the NLRI and the path up to that point (AS_1 through AS_{k+1})
  - certificate for each AS or router along path (AS_1 to AS_n) to check signatures on the route attestations
  - and, of course, all the relevant CRLs must have been checked

Slide: Kent et al.

Recall: DNS Lookup

Query: "www.example.com A?"

Reply	Resource Records in Reply
3	"com. NS a.gtld.net" "a.gtld.net A 192.5.6.30"
5	"example.com. NS a.iana.net" "a.iana.net A 192.0.34.43"
7	"www.example.com A 1.2.3.4"
8	"www.example.com A 1.2.3.4"



Local recursive resolver caches these for TTL specified by RR

DNS is Insecure

- Packets over UDP, < 512 bytes
- 16-bit TXID, UDP Src port only "security"
- Resolver accepts packet if above match
- Packet from whom? Was it manipulated?

- Cache poisoning
  - Attacker forges record at resolver
  - Forged record cached, attacks future lookups
  - Kaminsky (BH USA08)
    - Attacks delegations with "birthday problem"

DNSSEC Goal

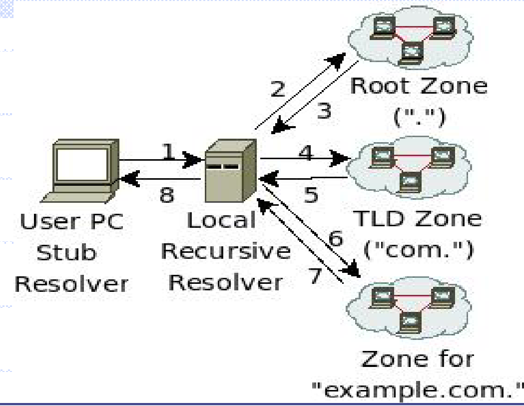
"The Domain Name System (DNS) security extensions provide origin authentication and integrity assurance services for DNS data, including mechanisms for authenticated denial of existence of DNS data."

-RFC 4033

DNSSEC

- Basically no change to packet format
  - Object security of DNS data, not channel security
- New Resource Records (RRs)
  - RRSIG : signature of RR by private zone key
  - DNSKEY : public zone key
  - DS : crypto digest of child zone key
  - NSEC / NSEC3 :authenticated denial of existence
- Lookup referral chain (unsigned)
- Origin attestation chain (PKI) (signed)
  - Start at pre-configured trust anchors
    - DS/DNSKEY of zone (should include root)
  - DS → DNSKEY → DS forms a link

DNSSEC Lookup



Query: "www.example.com A?"

Reply	RRs in DNS Reply	Added by DNSSEC
3	"com. NS a.gtld.net" "a.gtld.net A 192.5.6.30"	"com. DS" "RRSIG(DS) by ."
5	"example.com. NS a.iana.net" "a.iana.net A 192.0.34.43"	"com. DNSKEY" "RRSIG(DNSKEY) by com." "example.com. DS" "RRSIG(DS) by com."
7	"www.example.com A 1.2.3.4"	"example.com DNSKEY" "RRSIG(DNSKEY) by example.com." "RRSIG(A) by example.com."
8	"www.example.com A 1.2.3.4"	Last Hop?

Authenticated Denial-of-Existence

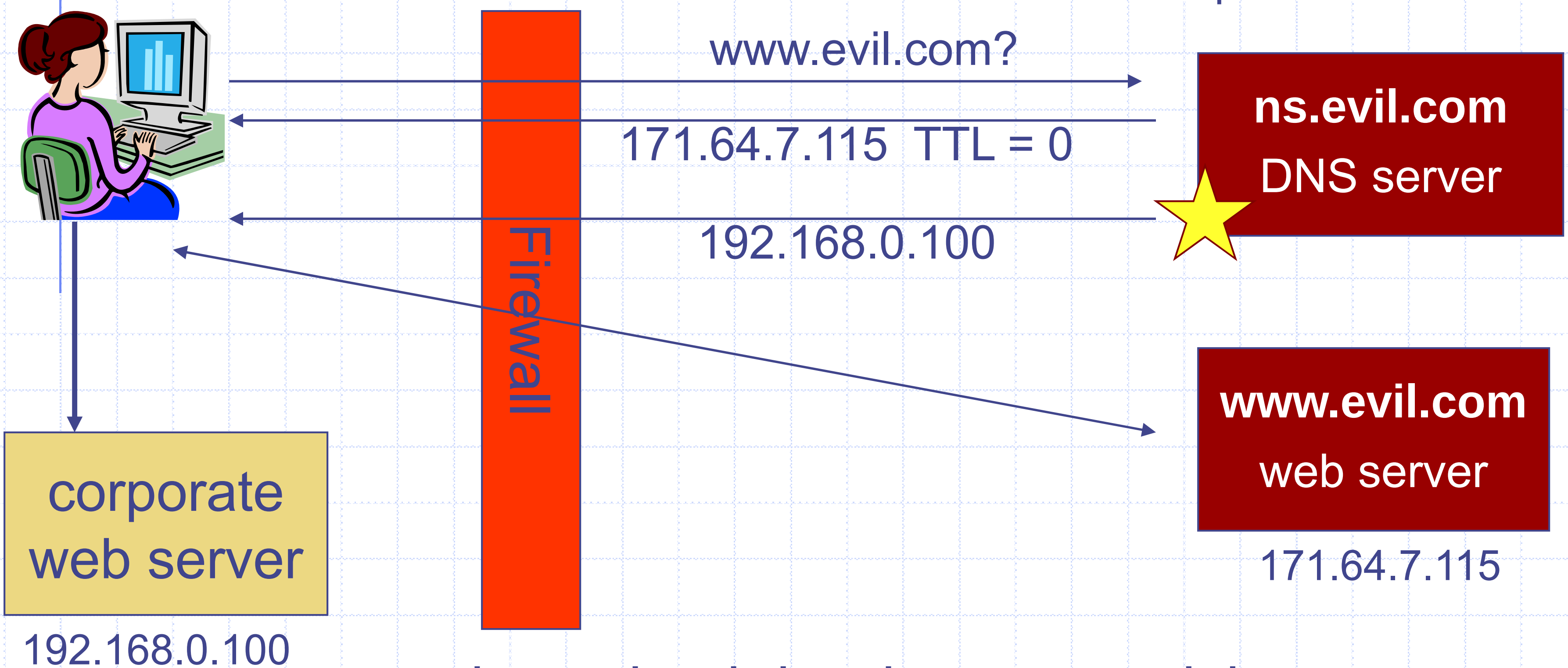
- Most DNS lookups result in denial-of-existence
- Understood mandate of offline-technique
- NSEC (Next SECure)
  - Lists all extant RRs associated with an owner name
  - Points to next owner name with extant RR
  - Easy zone enumeration
- NSEC3
  - Hashes owner names
    - Public salt to prevent pre-computed dictionaries
  - NSEC3 chain in hashed order
  - Opt-out bit for TLDs to support incremental adoption
    - For TLD type zones to support incremental adoption
    - Non-DNSSEC children not in NSEC3 chain

[DWF'96, R'01]

DNS Rebinding Attack

<iframe src="**http://www.evil.com**">

DNSSEC cannot stop this attack

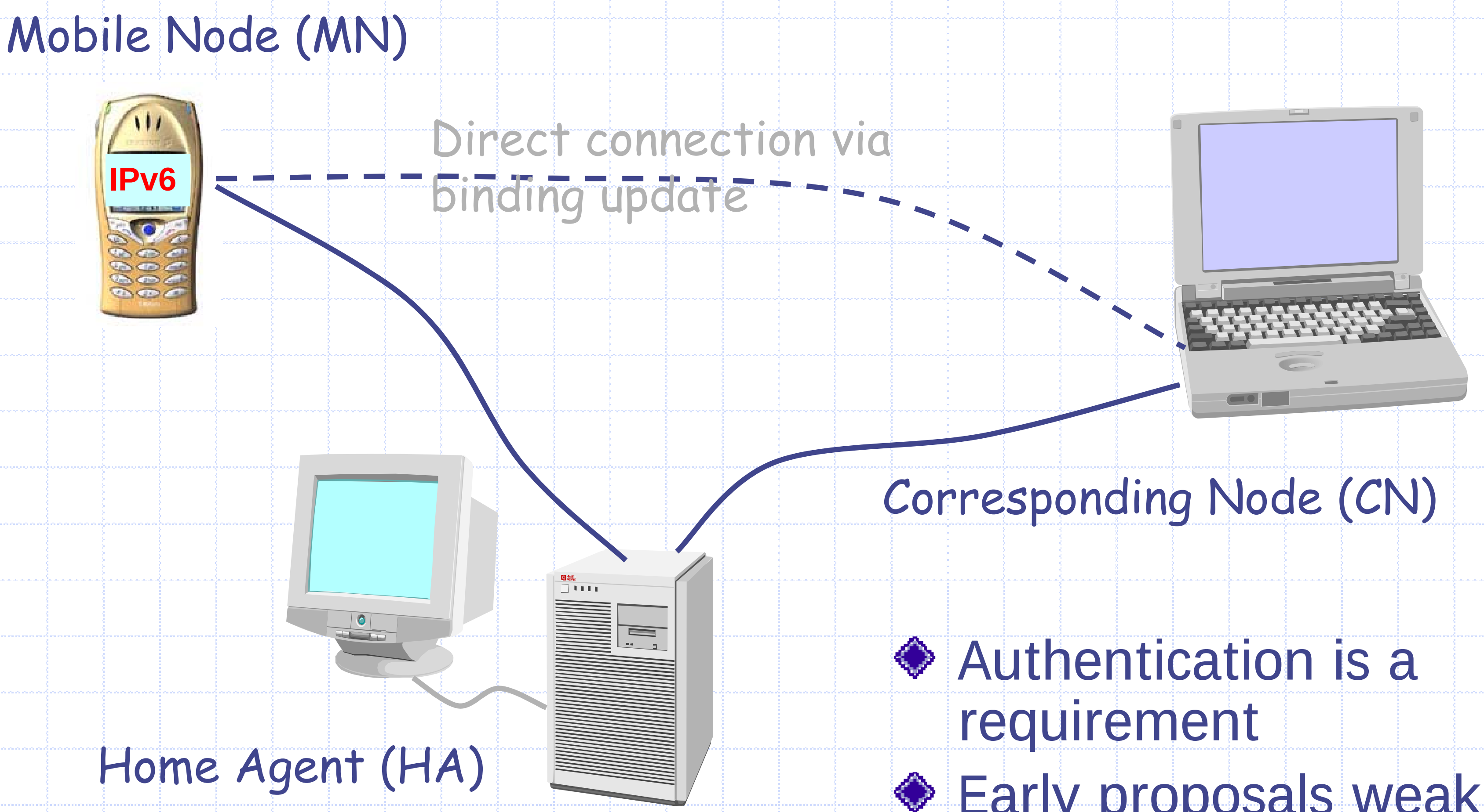


Read permitted: it's the "same origin"

DNS Rebinding Defenses

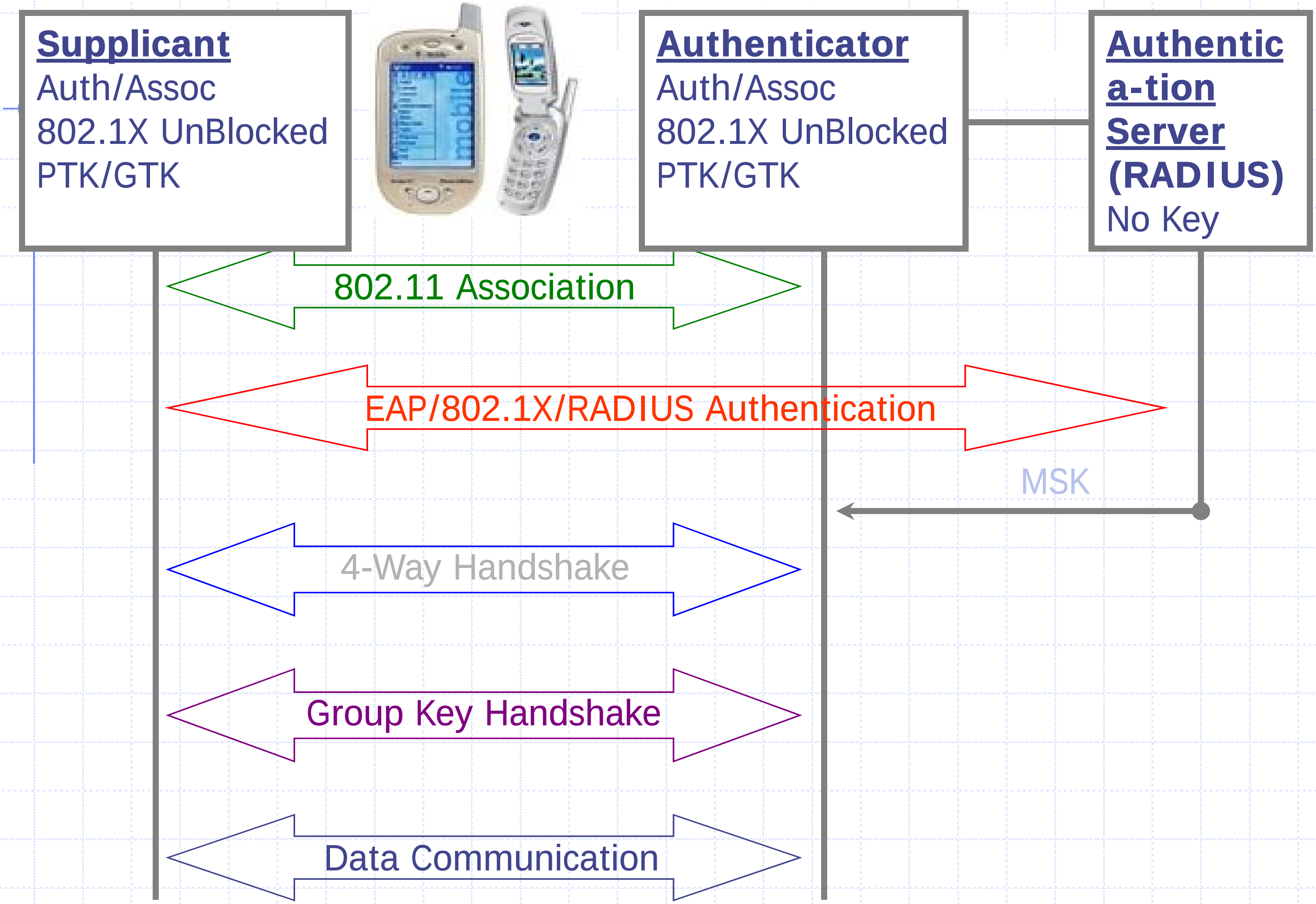
- Browser mitigation: DNS Pinning
  - Refuse to switch to a new IP
  - Interacts poorly with proxies, VPN, dynamic DNS, ...
  - Not consistently implemented in any browser
- Server-side defenses
  - Check Host header for unrecognized domains
  - Authenticate users with something other than IP
- Firewall defenses
  - External names can’t resolve to internal addresses
  - Protects browsers inside the organization

Mobile IPv6 Architecture



- Authentication is a requirement
- Early proposals weak

802.11i Protocol



Announcements

- Homework 2 will be out by Thurs
  - Due one week from Thursday

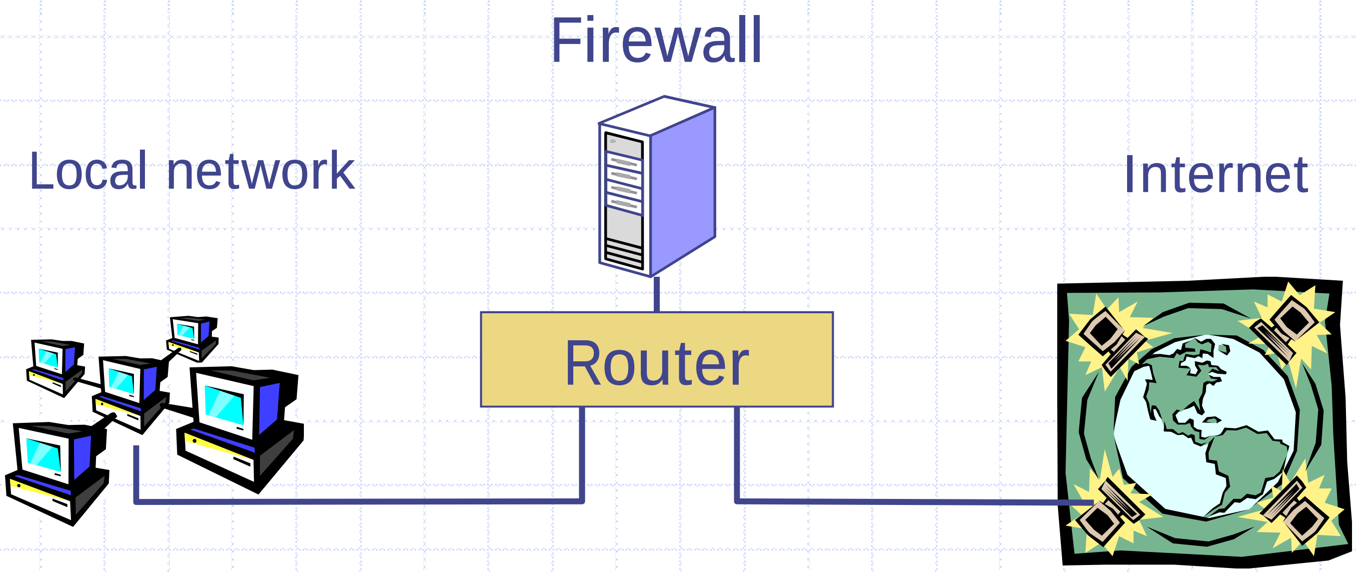
Perimeter and Internal Defenses

- Commonly deployed defenses
  - Perimeter defenses – Firewall, IDS
    - Protect local area network and hosts
    - Keep external threats from internal network
  - Internal defenses – Virus scanning
    - Protect hosts from threats that get through the perimeter defenses
  - Extend the "perimeter" – VPN

Rest of this lecture

Basic Firewall Concept

- Separate local area net from internet

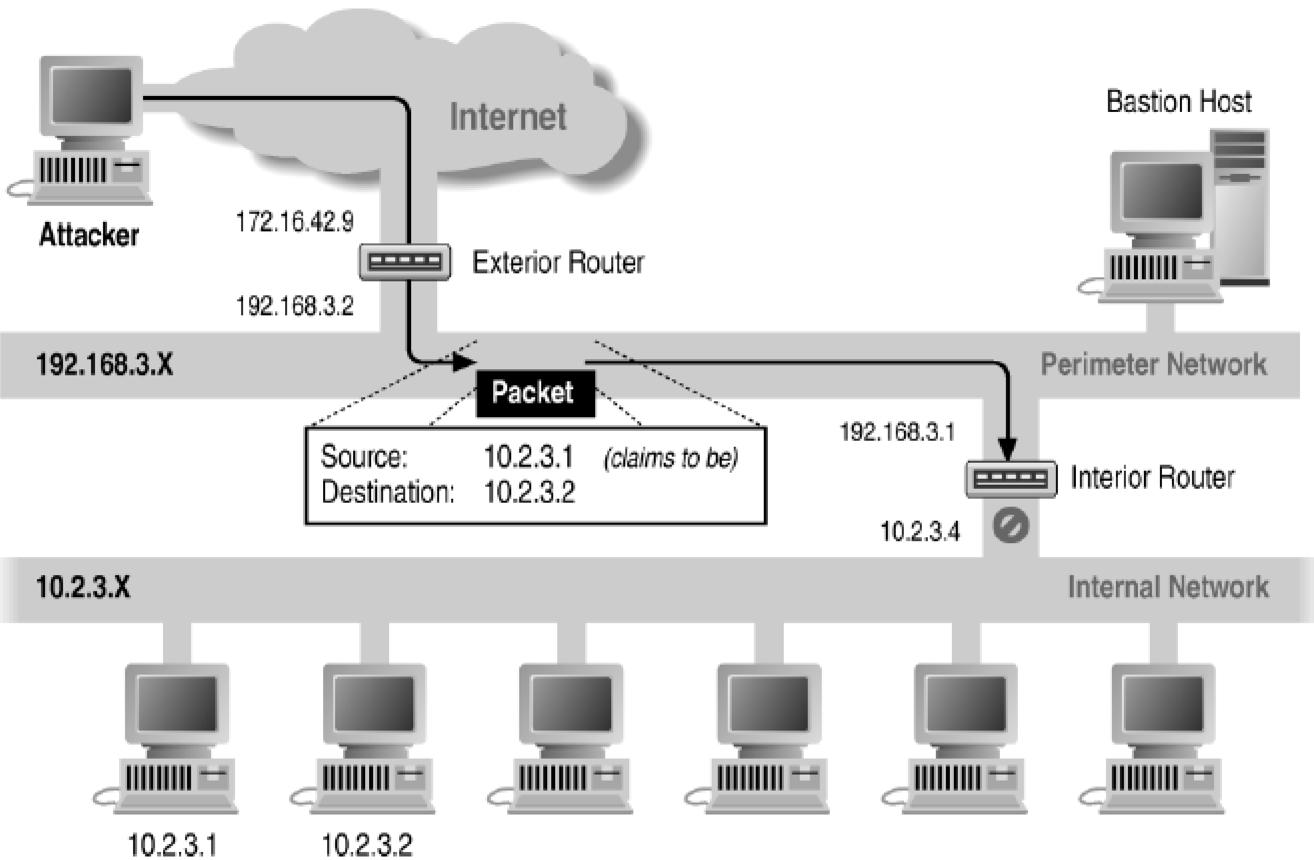


All packets between LAN and internet routed through firewall

Packet Filtering

- Uses transport-layer information only
  - IP Source Address, Destination Address
  - Protocol (TCP, UDP, ICMP, etc)
  - TCP or UDP source & destination ports
  - TCP Flags (SYN, ACK, FIN, RST, PSH, etc)
  - ICMP message type
- Examples
  - DNS uses port 53
    - Block incoming port 53 packets except known trusted servers
- Issues
  - Stateful filtering
  - Encapsulation: address translation, other complications
  - Fragmentation

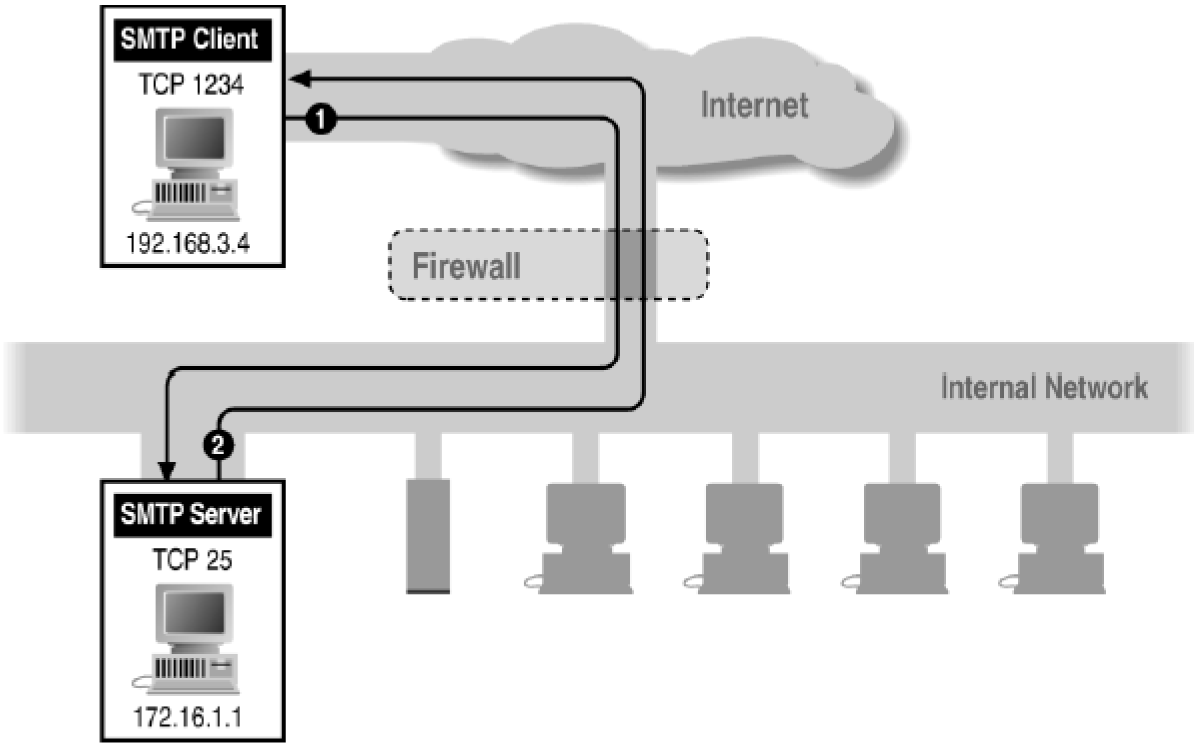
Source/Destination Address Forgery



More about networking: port numbering

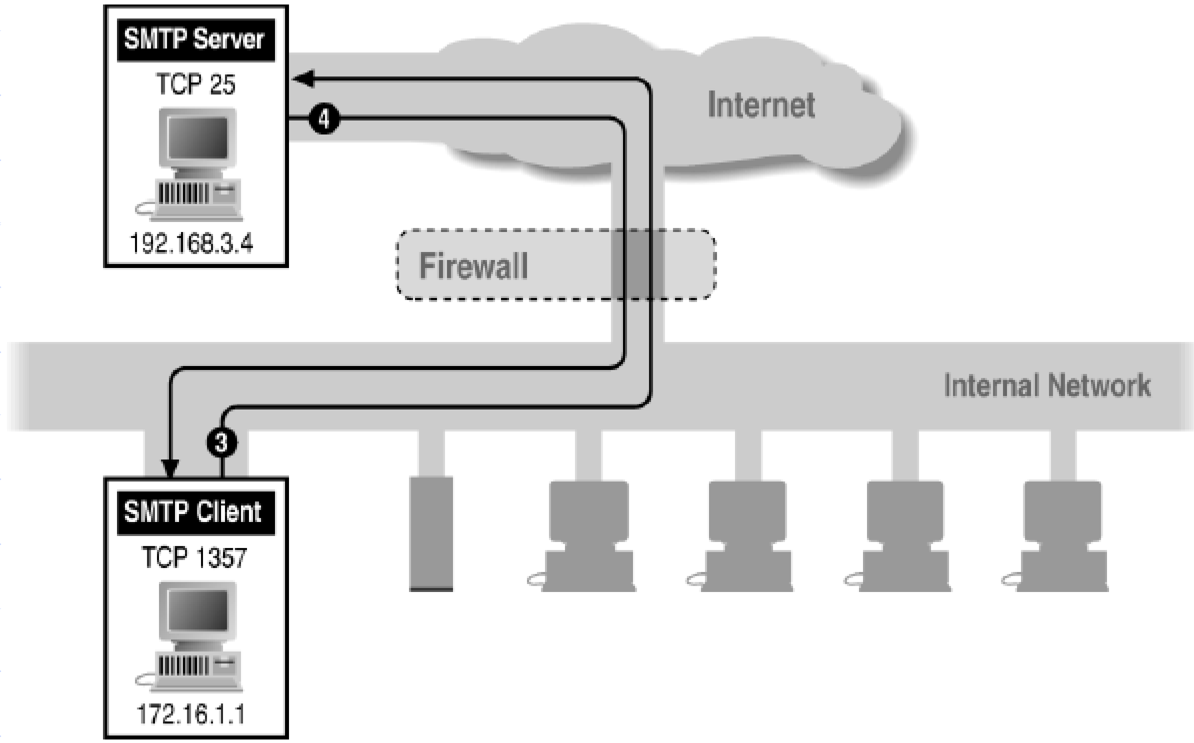
- TCP connection
  - Server port uses number less than 1024
  - Client port uses number between 1024 and 16383
- Permanent assignment
  - Ports <1024 assigned permanently
    - 20,21 for FTP 23 for Telnet
    - 25 for server SMTP 80 for HTTP
- Variable use
  - Ports >1024 must be available for client to make connection
  - Limitation for stateless packet filtering
    - If client wants port 2048, firewall must allow incoming traffic
  - Better: stateful filtering knows outgoing requests
    - Only allow incoming traffic on high port to a machine that has initiated an outgoing request on low port

Filtering Example: Inbound SMTP



Can block external request to internal server based on port number

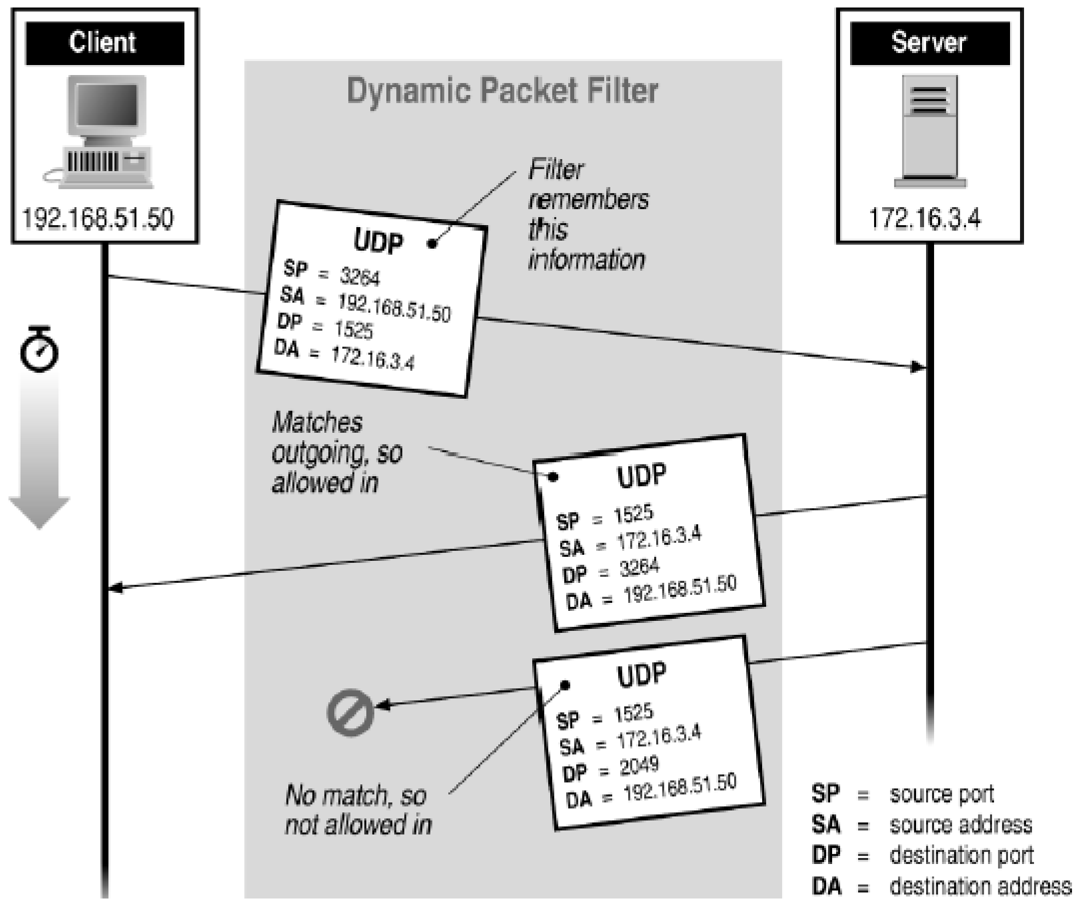
Filtering Example: Outbound SMTP



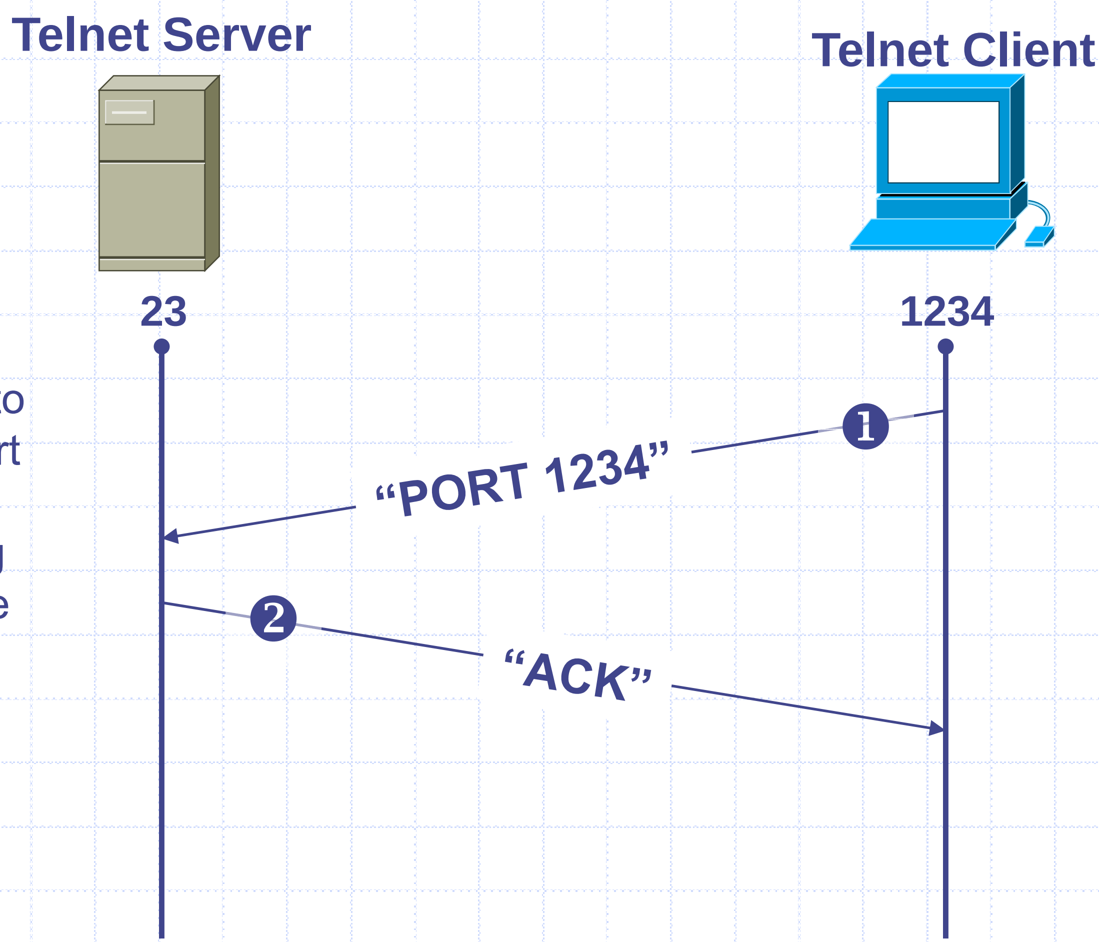
Known low port out, arbitrary high port in

If firewall blocks incoming port 1357 traffic then connection fails

Stateful or Dynamic Packet Filtering



Telnet



❶ Client opens channel to server; tells server its port number. The ACK bit is not set while establishing the connection but will be set on the remaining packets

❷ Server acknowledges

Stateful filtering can use this pattern to identify legitimate sessions

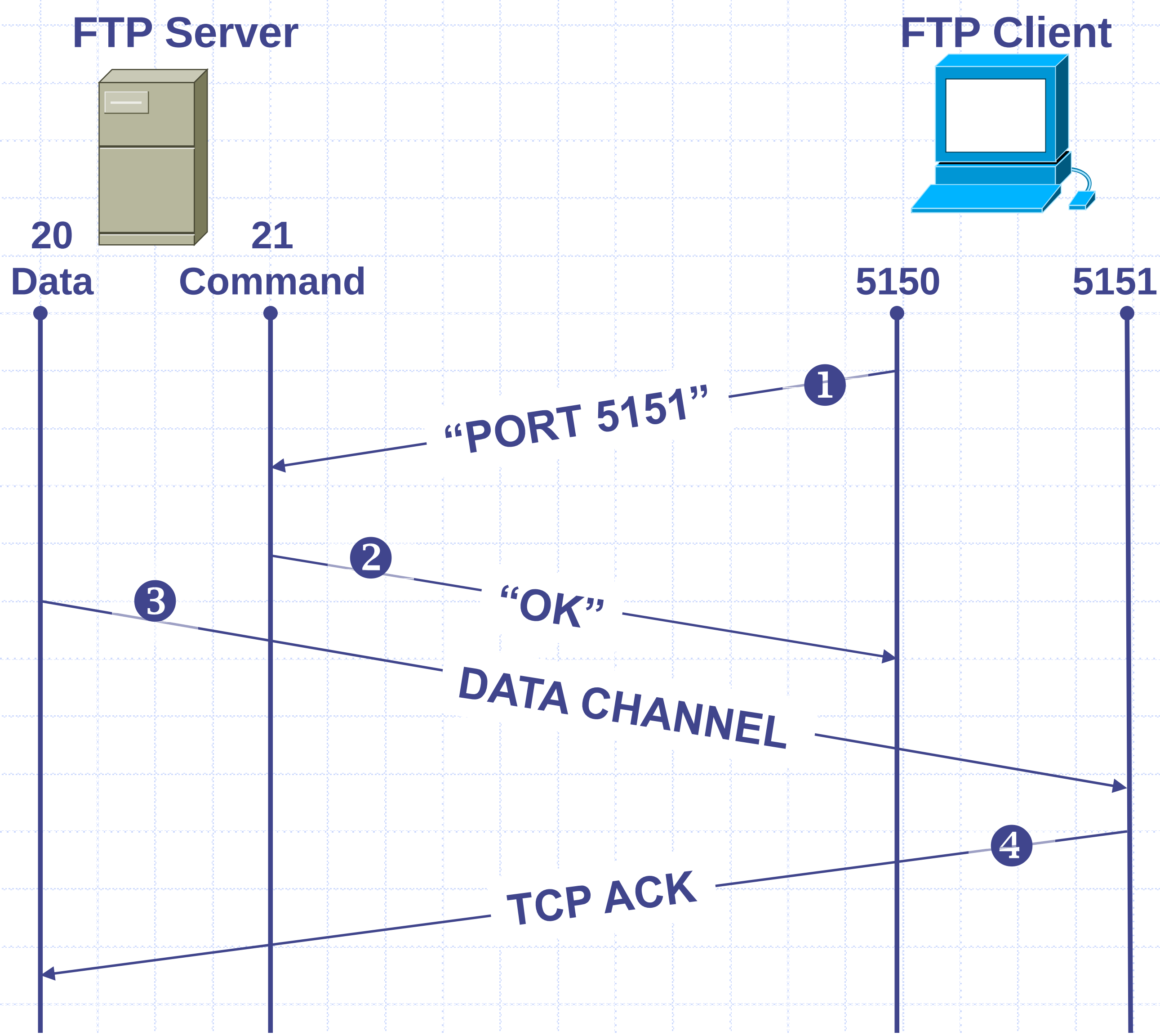
FTP

❶ Client opens command channel to server; tells server second port number

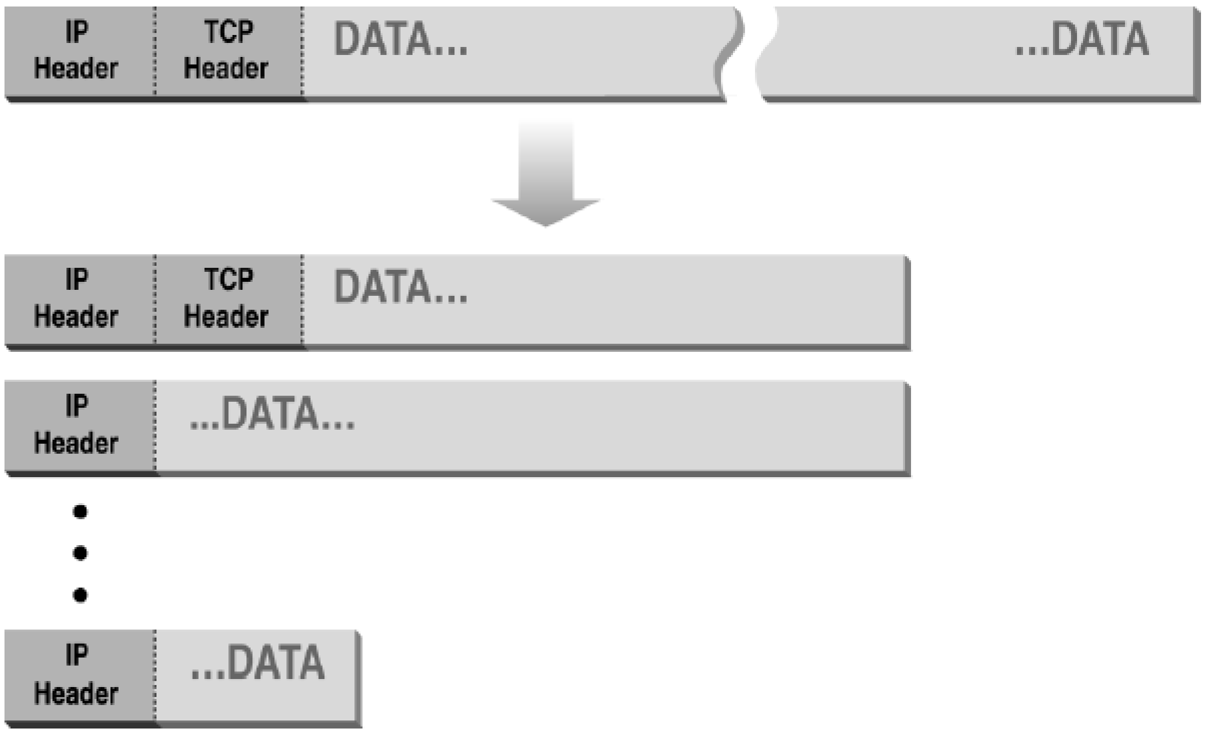
❷ Server acknowledges

❸ Server opens data channel to client's second port

❹ Client acknowledges

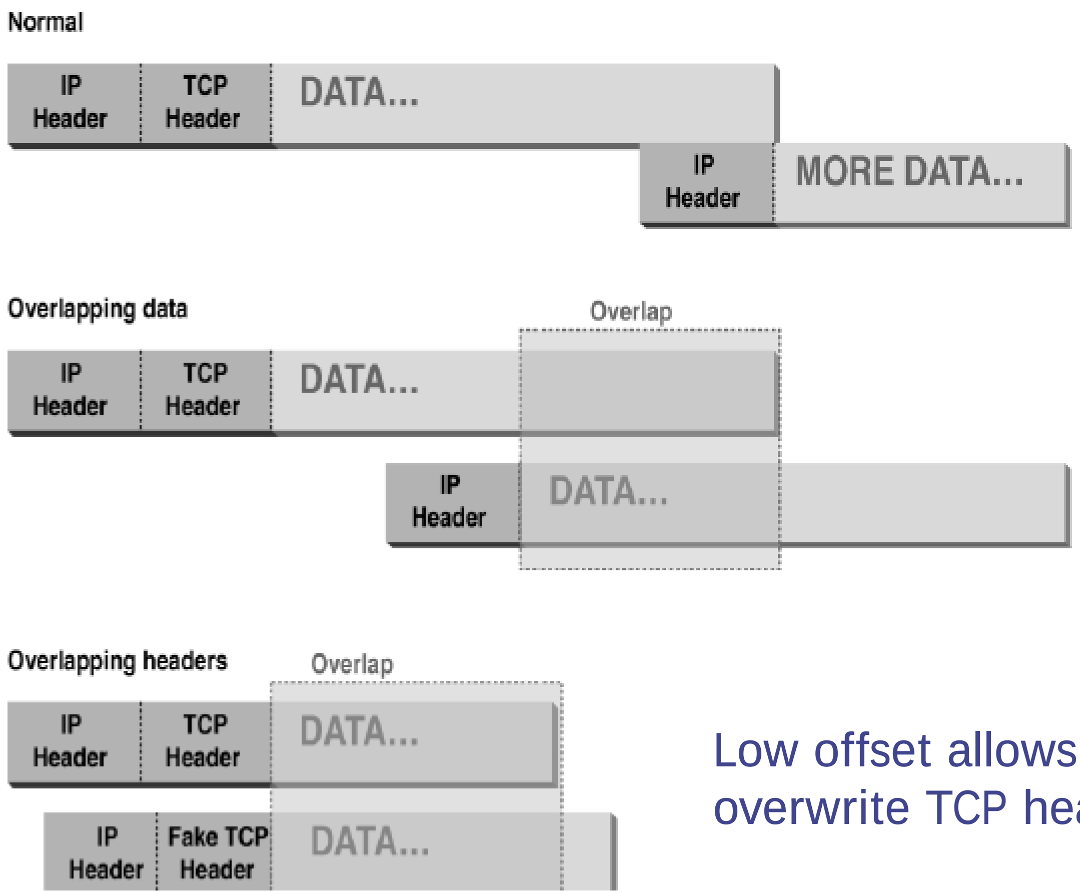


Normal IP Fragmentation



Flags and offset inside IP header indicate packet fragmentation

Abnormal Fragmentation



Low offset allows second packet to overwrite TCP header at receiving host

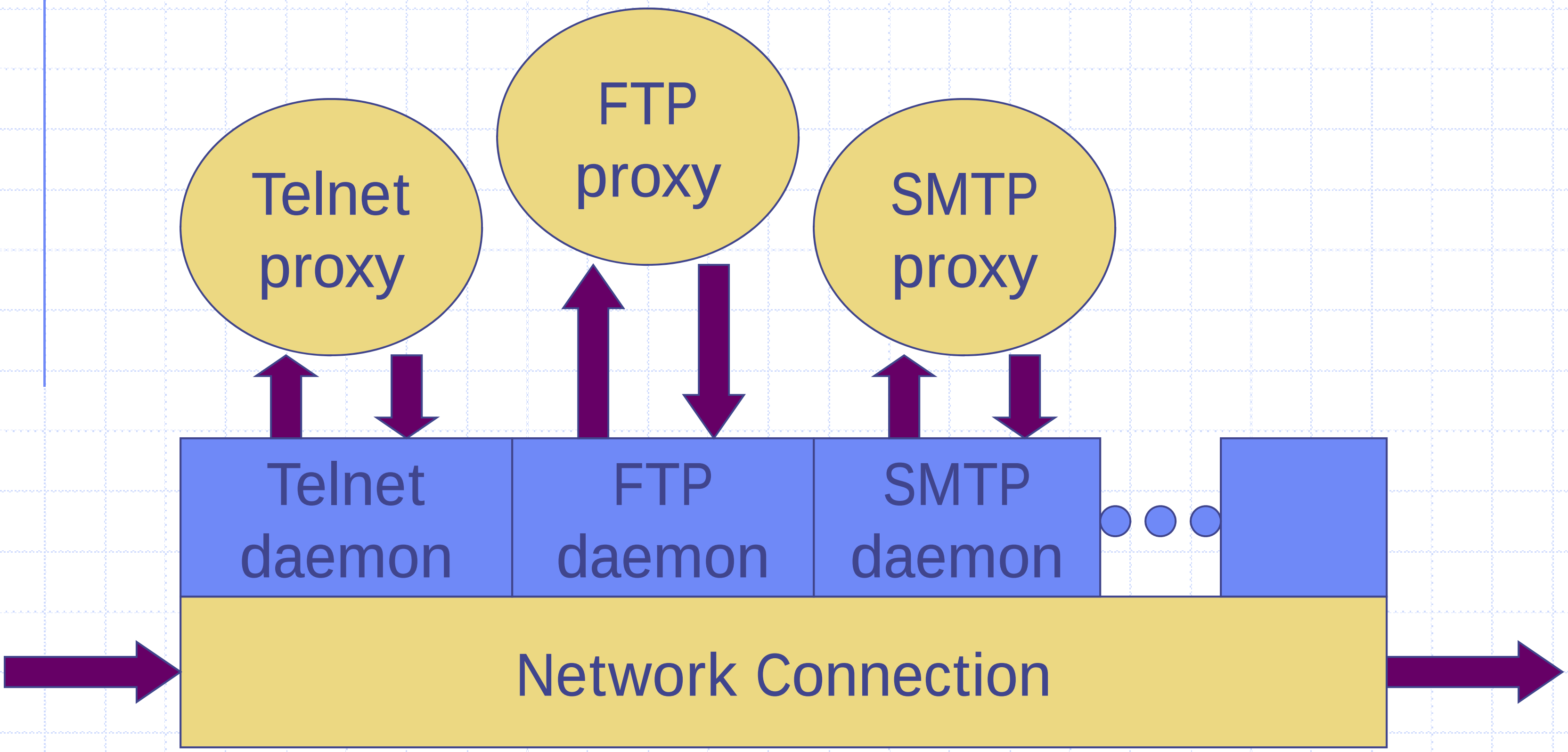
Packet Fragmentation Attack

- Firewall configuration
  - TCP port 23 is blocked but SMTP port 25 is allowed
- First packet
  - Fragmentation Offset = 0.
  - DF bit = 0 : "May Fragment"
  - MF bit = 1 : "More Fragments"
  - Destination Port = 25. TCP port 25 is allowed, so firewall allows packet
- Second packet
  - Fragmentation Offset = 1: second packet overwrites all but first 8 bits of the first packet
  - DF bit = 0 : "May Fragment"
  - MF bit = 0 : "Last Fragment."
  - Destination Port = 23. Normally be blocked, but sneaks by!
- What happens
  - Firewall ignores second packet “TCP header” because it is fragment of first
  - At host, packet reassembled and received at port 23

Proxying Firewall

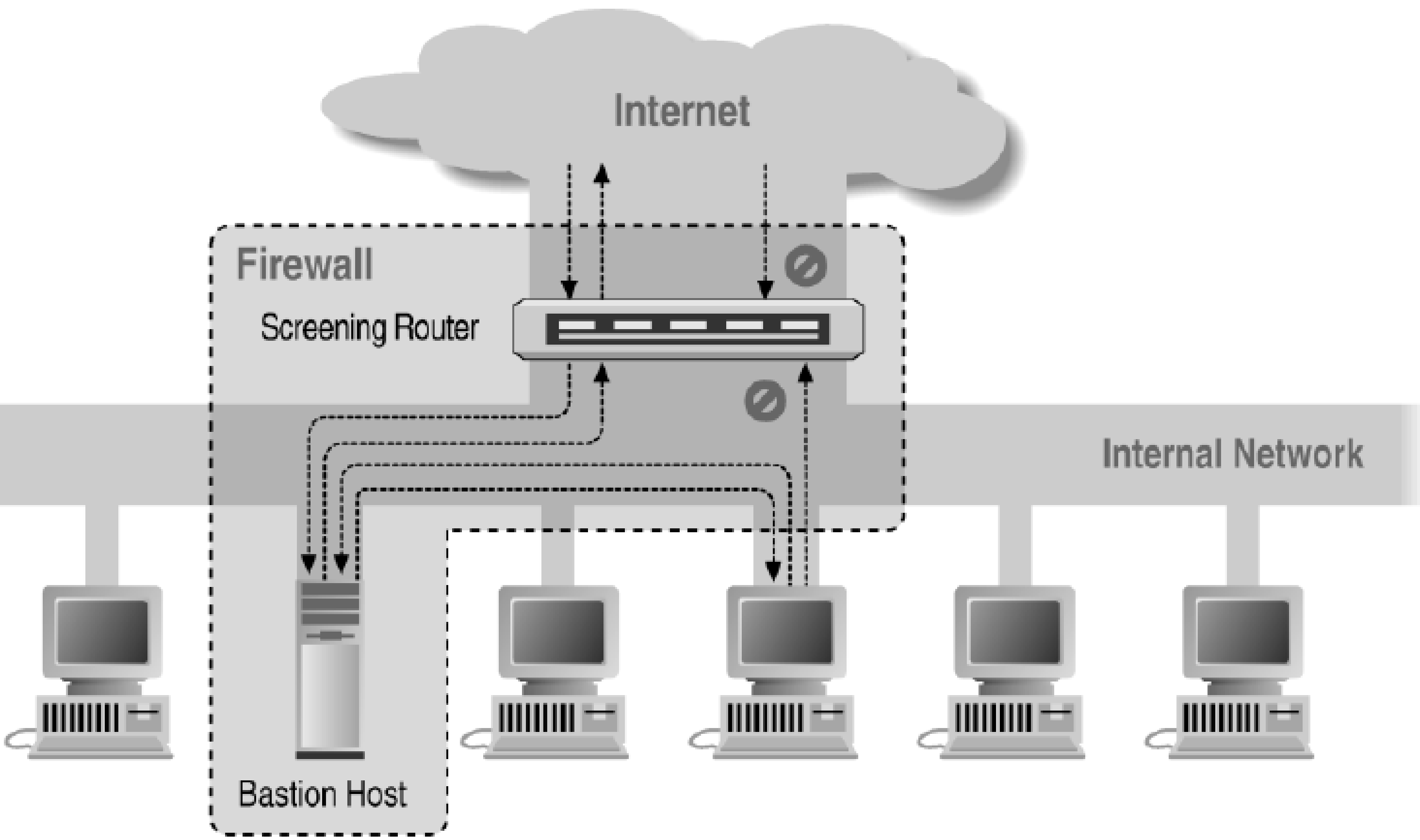
- Application-level proxies
  - Tailored to http, ftp, smtp, etc.
  - Some protocols easier to proxy than others
- Policy embedded in proxy programs
  - Proxies filter incoming, outgoing packets
  - Reconstruct application-layer messages
  - Can filter specific application-layer commands, etc.
    - Example: only allow specific ftp commands
    - Other examples: ?
- Several network locations – see next slides

Firewall with application proxies

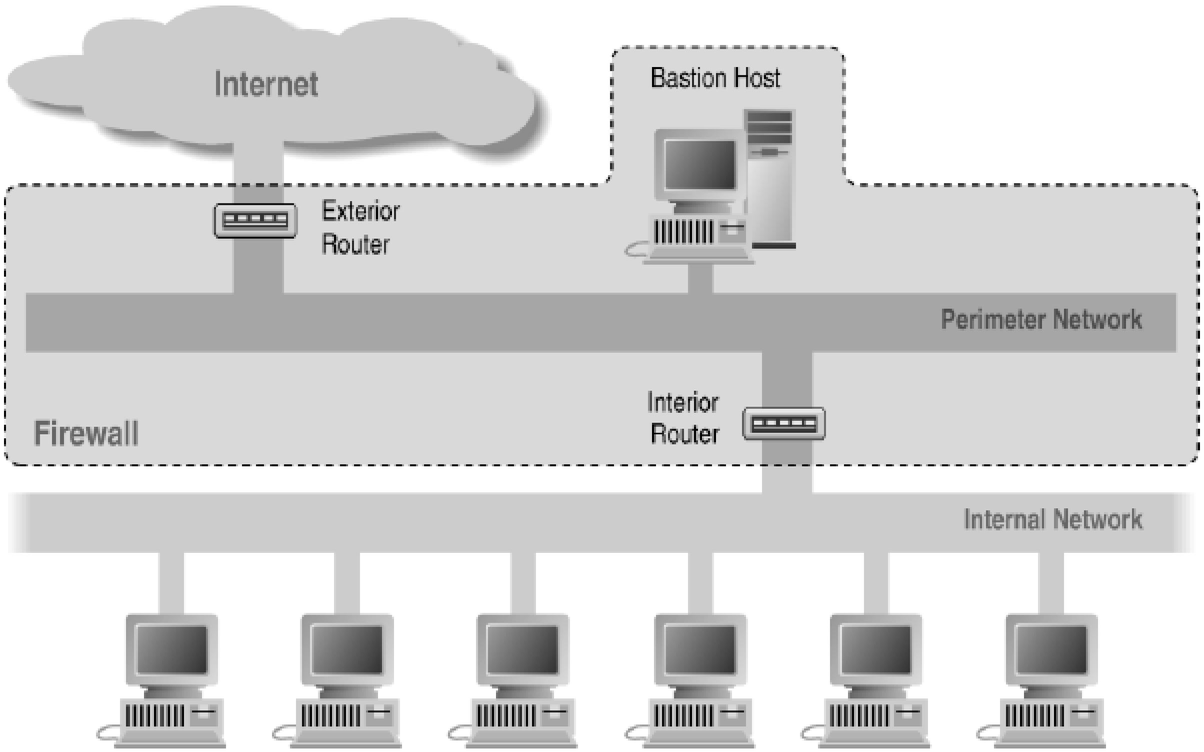


Daemon spawns proxy when communication detected ...

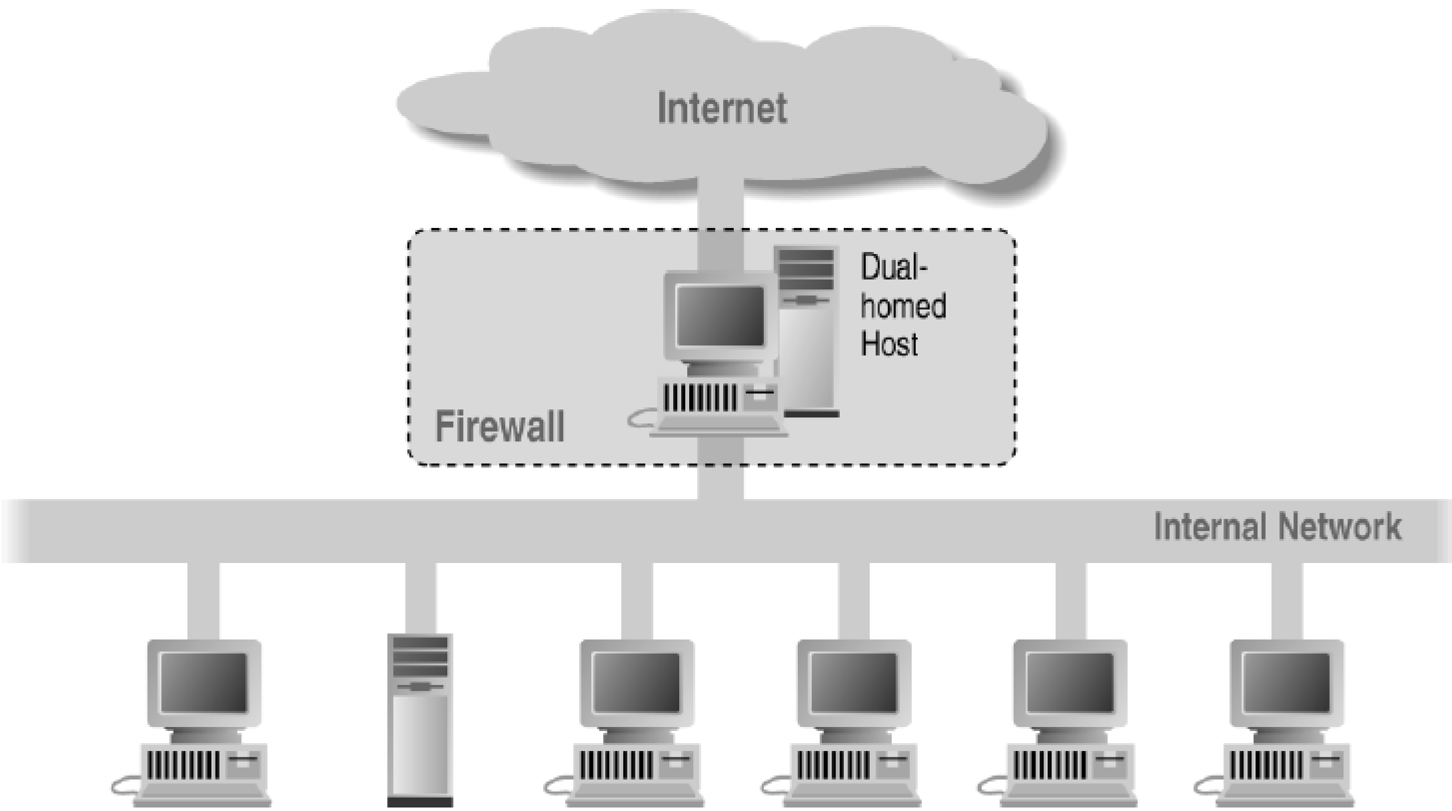
Screened Host Architecture



Screened Subnet Using Two Routers



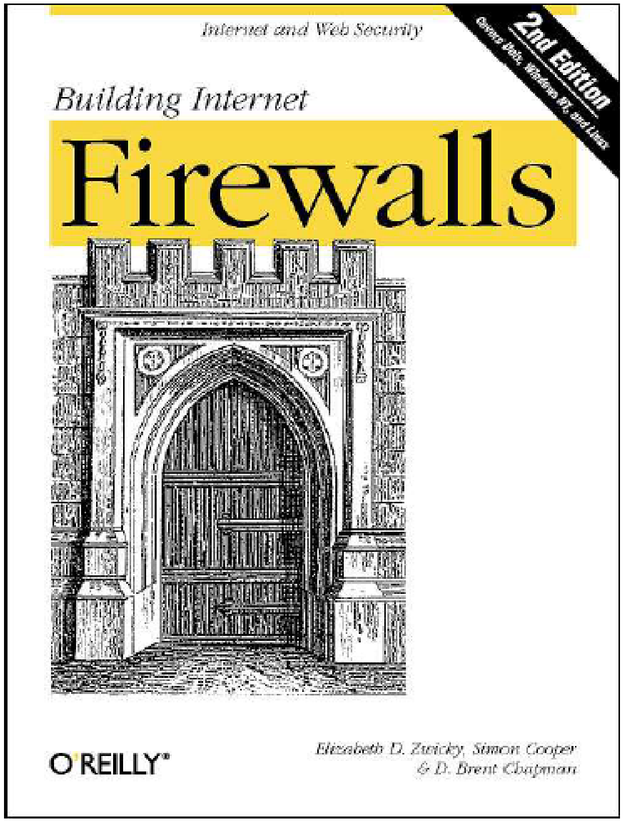
Dual Homed Host Architecture

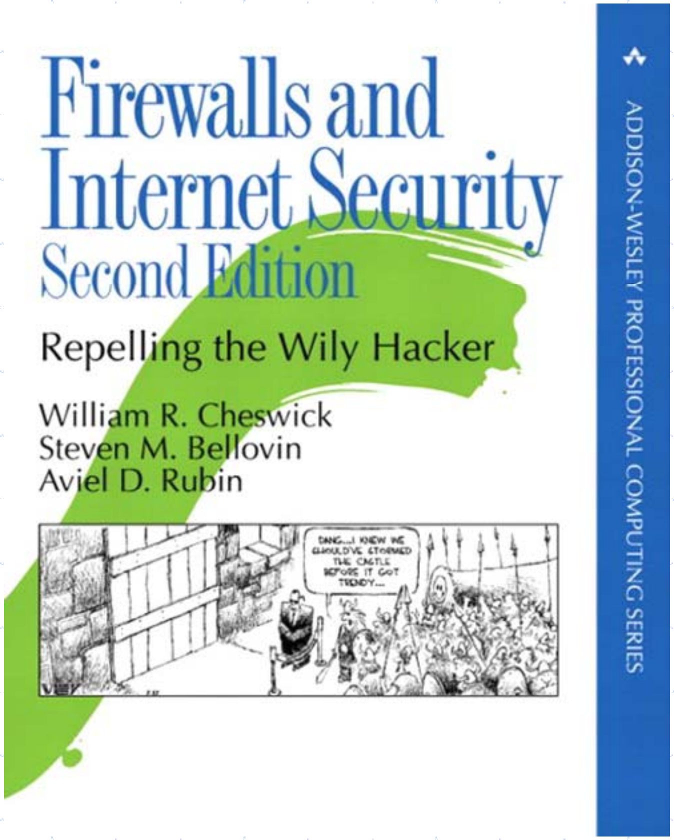


Application-level proxies

- Enforce policy for specific protocols
  - E.g., Virus scanning for SMTP
    - Need to understand MIME, encoding, Zip archives
  - Flexible approach, but may introduce network delays
- “Batch” protocols are natural to proxy
  - SMTP (E-Mail) NNTP (Net news)
  - DNS (Domain Name System) NTP (Network Time Protocol
- Must protect host running protocol stack
  - Disable all non-required services; keep it simple
  - Install/modify services you want
  - Run security audit to establish baseline
  - Be prepared for the system to be compromised

References





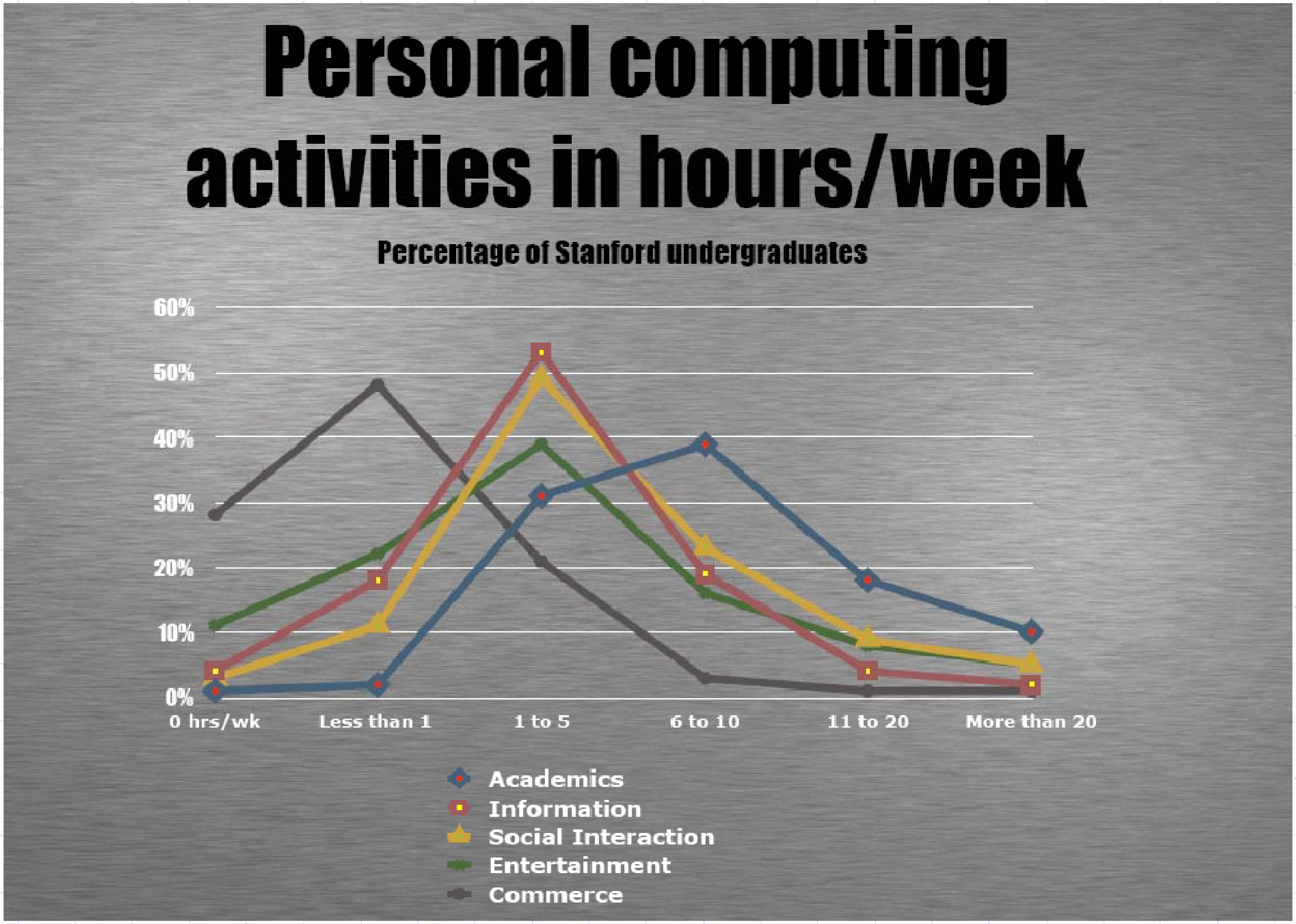
Elizabeth D. Zwicky
Simon Cooper
D. Brent Chapman

William R Cheswick
Steven M Bellovin
Aviel D Rubin

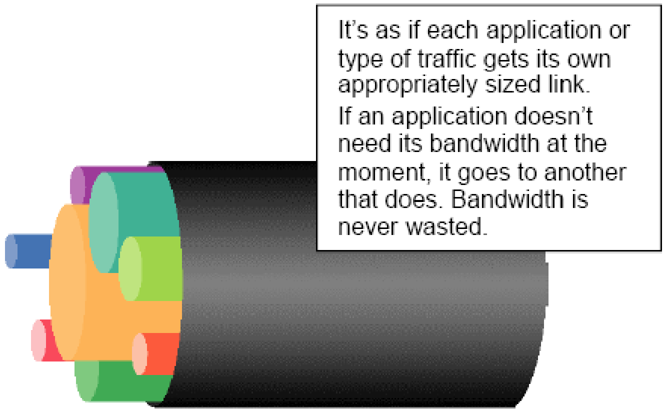
Traffic Shaping

- Traditional firewall
  - Allow traffic or not
- Traffic shaping
  - Limit certain kinds of traffic
  - Can differentiate by host addr, protocol, etc
  - Multi-Protocol Label Switching (MPLS)
    - Label traffic flows at the edge of the network and let core routers identify the required class of service

Stanford computer use



PacketShaper Controls

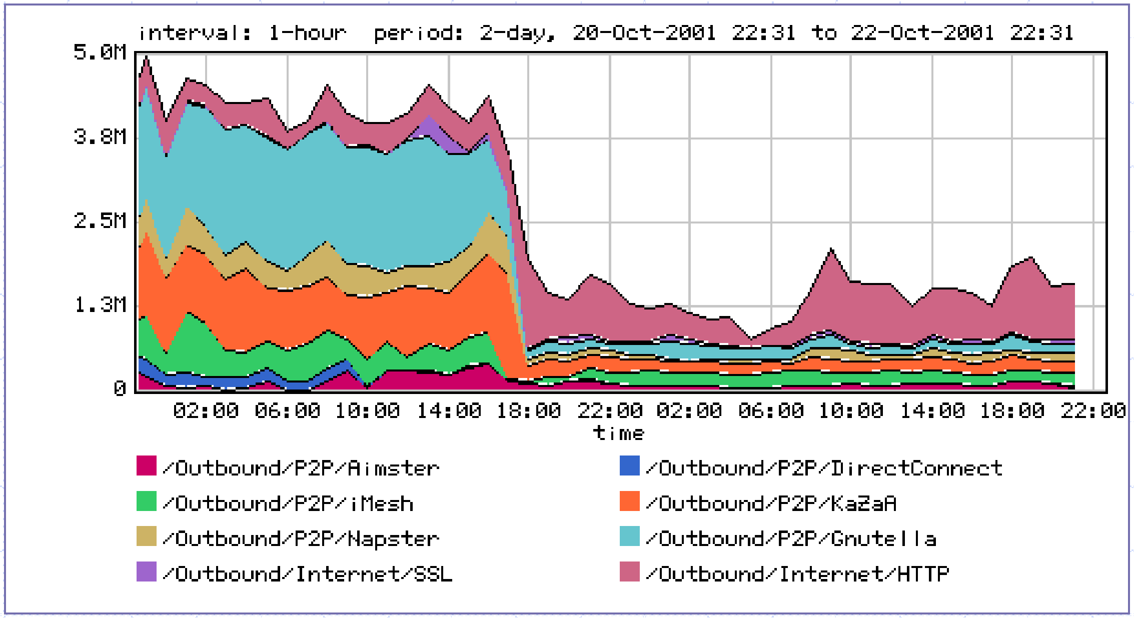


A partition:

- Creates a virtual pipe within a link for each traffic class
- Provides a min, max bandwidth
- Enables efficient bandwidth use

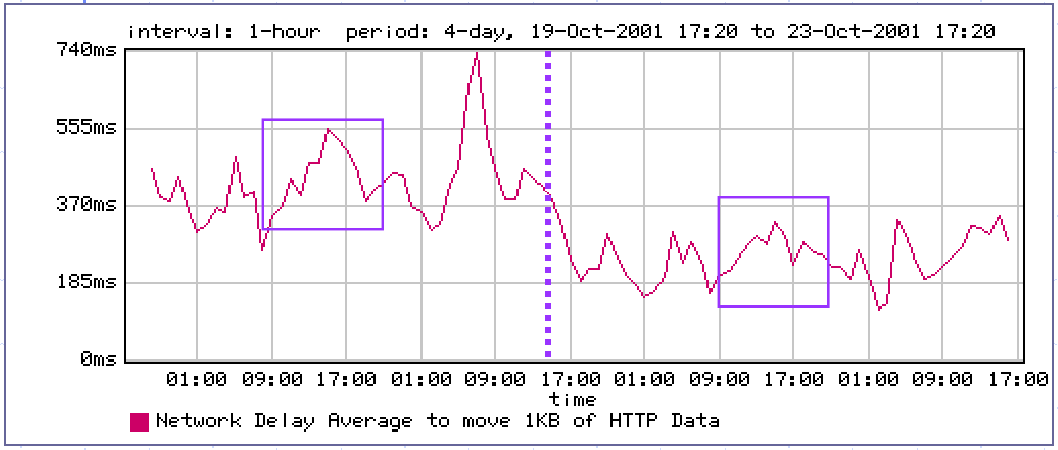
Rate shaped P2P capped at 300kbps ➢

Rate shaped HTTP/SSL to give better performance ➢



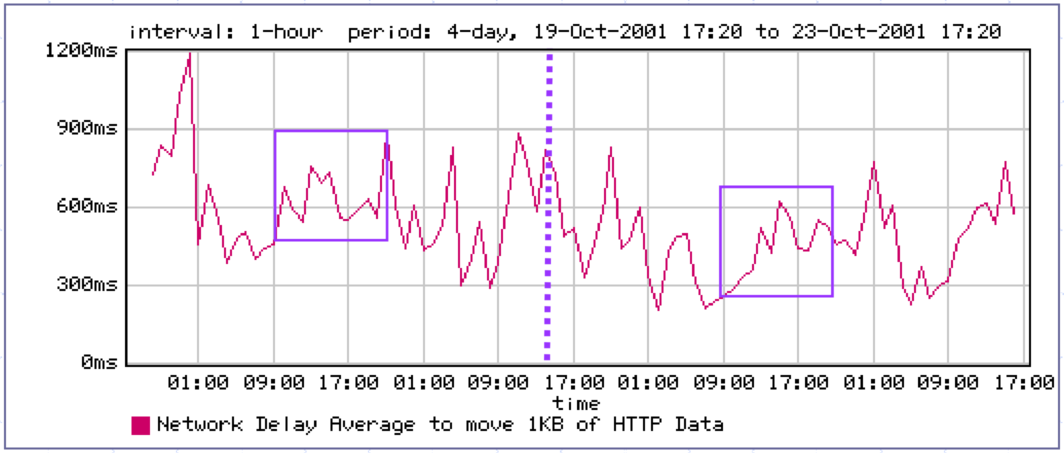
PacketShaper report: HTTP

Outside Web Server Normalized Network Response Times



No Shaping **Shaping**

Inside Web Server Normalized Network Response Times



No Shaping **Shaping**

Host and network intrusion detection

- Intrusion prevention
  - Network firewall
    - Restrict flow of packets
  - System security
    - Find buffer overflow vulnerabilities and remove them!
- Intrusion detection
  - Discover system modifications
    - Tripwire
  - Look for attack in progress
    - Network traffic patterns
    - System calls, other system events

Tripwire

- Outline of standard attack
  - Gain user access to system
  - Gain root access
  - Replace system binaries to set up backdoor
  - Use backdoor for future activities
- Tripwire detection point: system binaries
  - Compute hash of key system binaries
  - Compare current hash to hash stored earlier
  - Report problem if hash is different
  - Store reference hash codes on read-only medium

Is Tripwire too late?

- Typical attack on server
  - Gain access
  - Install backdoor
    - This can be in memory, not on disk!!
  - Use it
- Tripwire
  - Is a good idea
  - Wont catch attacks that don't change system files
  - Detects a compromise that *has happened*

Remember: Defense in depth

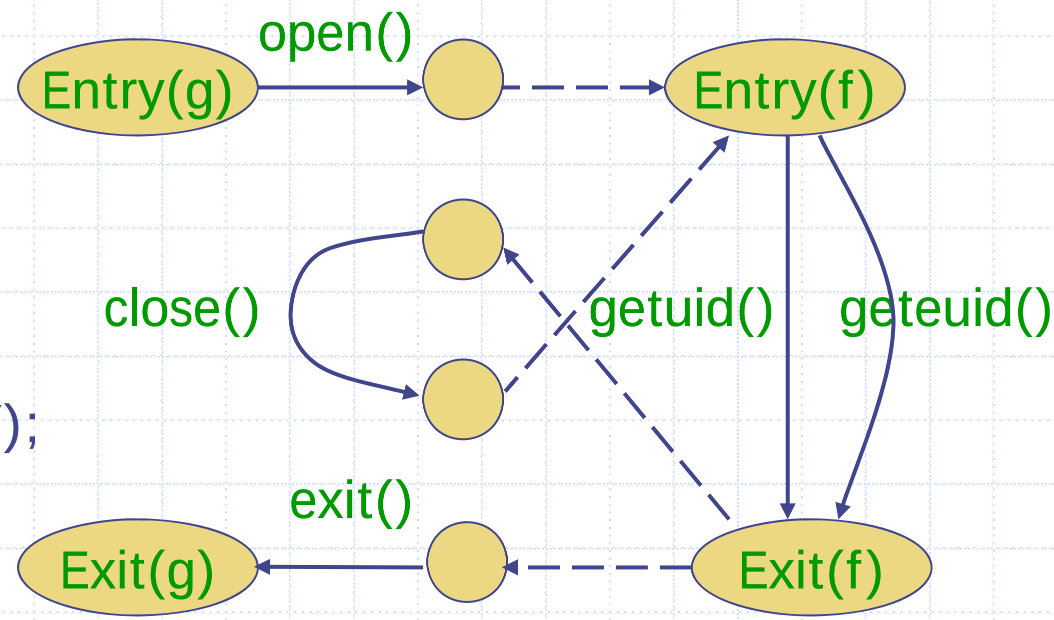
Detect modified binary in memory?

- Can use system-call monitoring techniques
- For example [Wagner, Dean IEEE S&P '01]
  - Build automaton of expected system calls
    - Can be done automatically from source code
  - Monitor system calls from each program
  - Catch violation

Results so far: lots better than not using source code!

Example code and automaton

```
f(int x) {
   x ? getuid() : geteuid();
   x++
}
g() {
   fd = open("foo", O_RDONLY);
   f(0); close(fd); f(1);
   exit(0);
}
```



If code behavior is inconsistent with automaton, something is wrong

General intrusion detection

http://www.snort.org/

- Many intrusion detection systems
  - Close to 100 systems with current web pages
  - Network-based, host-based, or combination
- Two basic models
  - Misuse detection model
    - Maintain data on known attacks
    - Look for activity with corresponding signatures
  - Anomaly detection model
    - Try to figure out what is “normal”
    - Report anomalous behavior
- Fundamental problem: too many false alarms

Anomaly Detection

- Basic idea
  - Monitor network traffic, system calls
  - Compute statistical properties
  - Report errors if statistics outside established range
- Example – IDES (Denning, SRI)
  - For each user, store daily count of certain activities
    - E.g., Fraction of hours spent reading email
  - Maintain list of counts for several days
  - Report anomaly if count is outside weighted norm

Big problem: most unpredictable user is the most important

[Hofmeyr, Somayaji, Forrest]

Anomaly – sys call sequences

- Build traces during normal run of program
  - Example program behavior (sys calls)

    open read write open mmap write fchmod close
  - Sample traces stored in file (4-call sequences)

    open read write open

    read write open mmap

    write open mmap write

    open mmap write fchmod

    mmap write fchmod close
  - Report anomaly if following sequence observed

    open read read open mmap write fchmod close

Compute # of mismatches to get mismatch rate

Difficulties in intrusion detection

- Lack of training data
  - Lots of “normal” network, system call data
  - Little data containing realistic attacks, anomalies
- 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
  - Sys Admin spend many hours examining evidence

Summary

- Network protocol security
  - IPSEC
  - BGP instability and S-BGP
  - DNSSEC, DNS rebinding
  - Wireless security – 802.11i/WPA2
- Standard network perimeter defenses
  - Firewall
    - Packet filter (stateless, stateful), Application layer proxies
  - Traffic shaping
  - Intrusion detection
    - Anomaly and misuse detection