Lecture 2

BASICS OF NETWORKS AND PROTOCOLS - SECURITY ISSUES

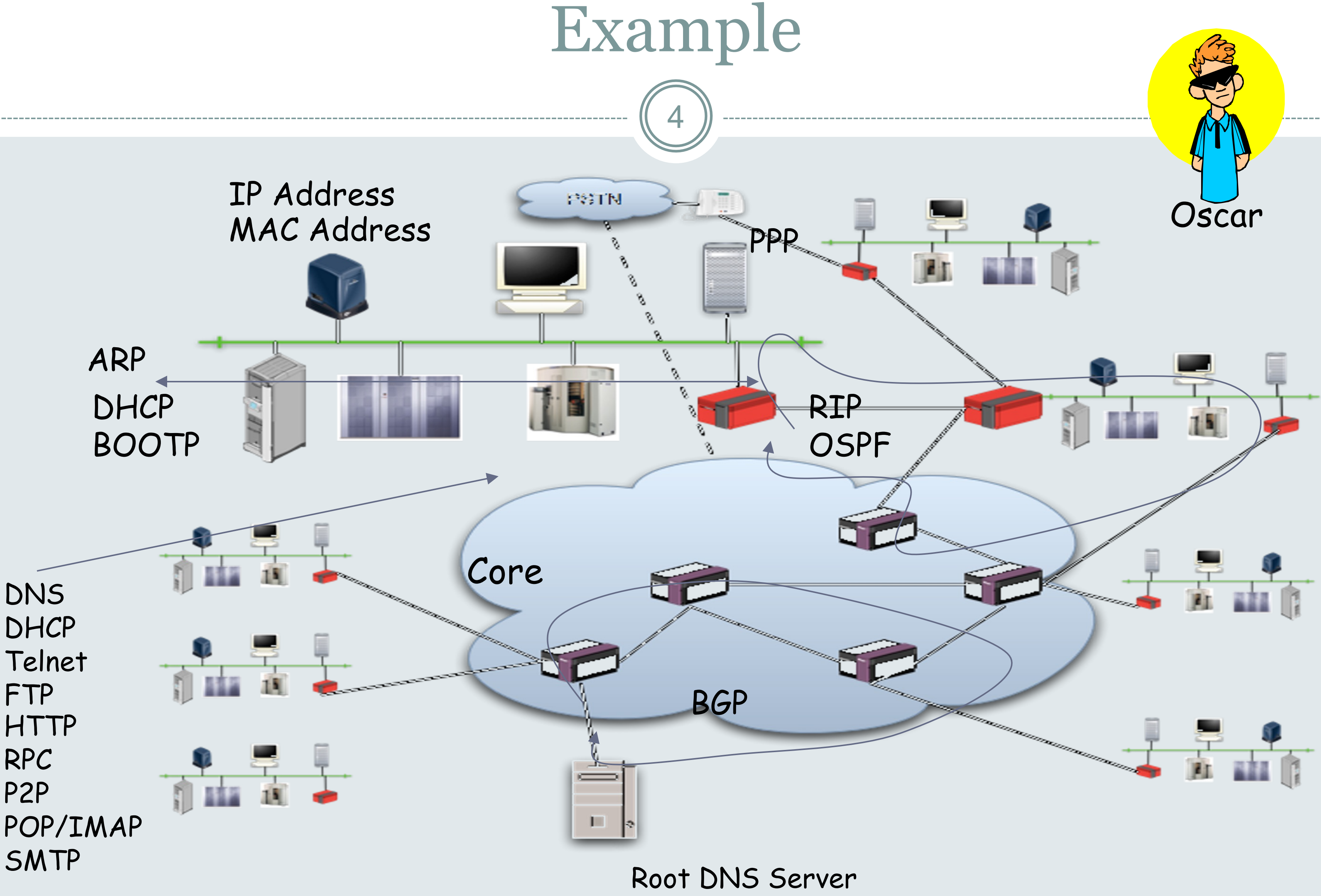
Review

- Network security is very complex
  - Many sources of threats
  - Many types of vulnerabilities
  - Some are not even “network” related, but the network provides access to Oscar
- One successful attack can lead to another!
  - Needs a lot of care and sometimes paranoia

This Class

- Consider some basics of network protocols
  - Understand some vulnerabilities through some examples
- Overview of different attacks
  - Details of a couple of other attacks will be considered next week

Example



Oscar

IP Address
MAC Address

PSTN

PPP

ARP

DHCP
BOOTP

RIP
OSPF

Core

BGP

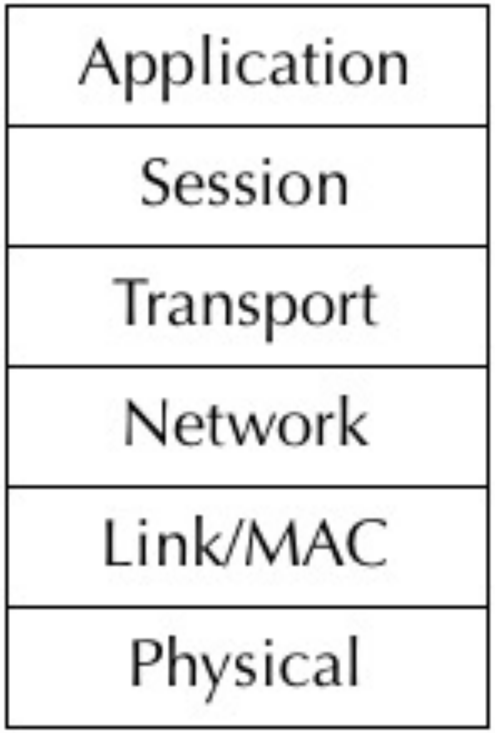
DNS
DHCP
Telnet
FTP
HTTP
RPC
P2P
POP/IMAP
SMTP

Root DNS Server

It is Complex!

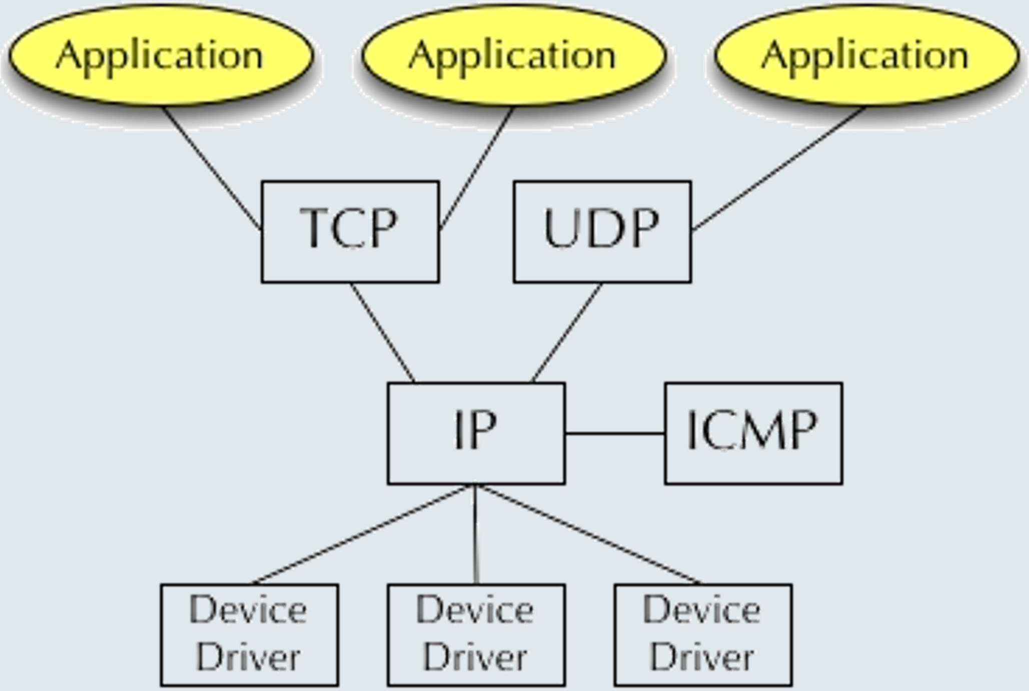
- Many protocols at many layers
  - Link layer - Ethernet & 802.11 are major
  - Network layer and its “helper” protocols
    - IP, ICMP, ARP, DNS, DHCP, …
  - Transport layer - TCP and UDP are major
- Applications
  - HTTP, SMTP, FTP, Telnet, IM, RSS feeds, Other Services, Real, …

Basic Concepts



- Application
- Session
- Transport
- Network
- Link/MAC
- Physical

OSI Model

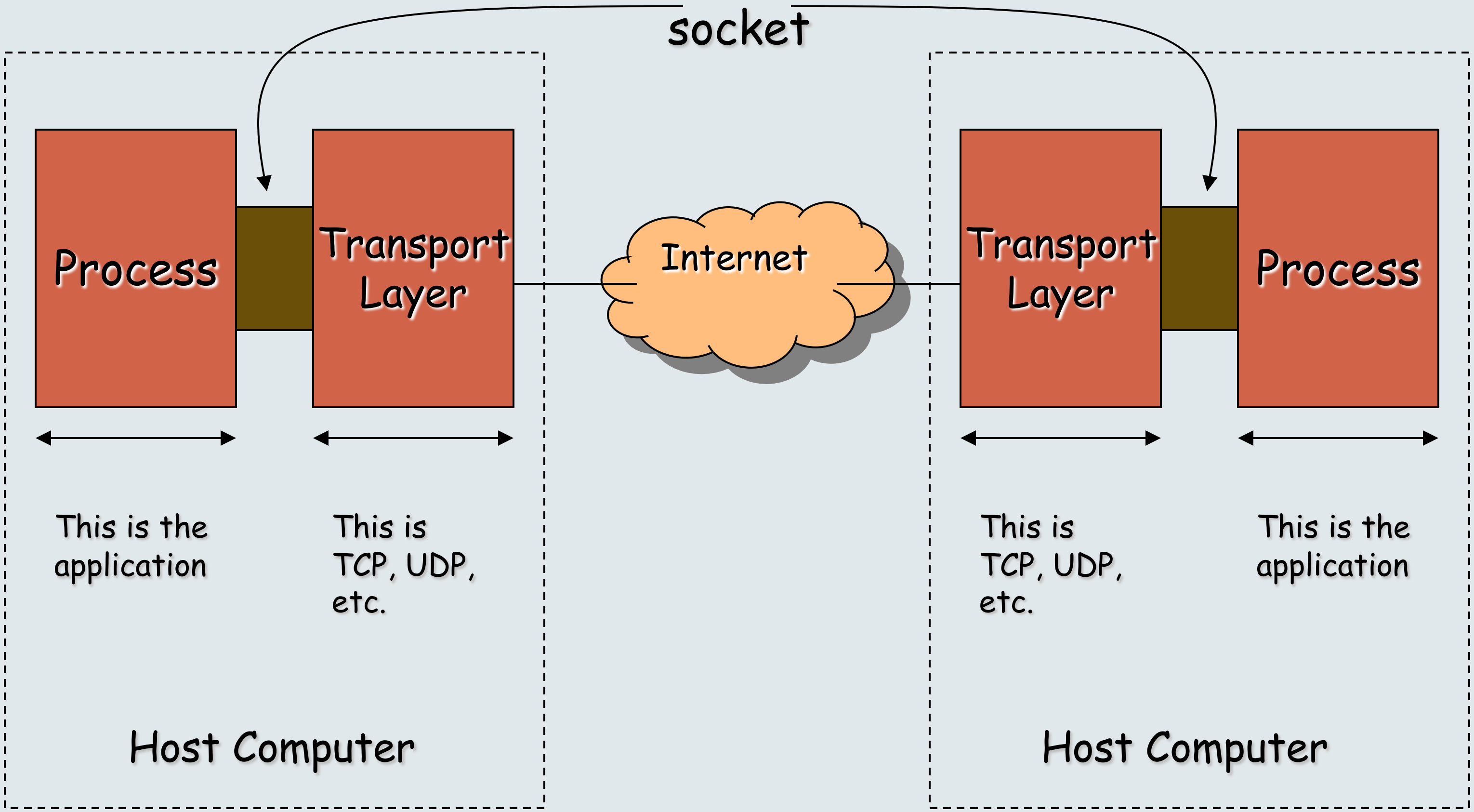


Schematic of TCP/IP Operation

Communications Across a Network

- Communicating processes typically have a client side and a server side
  - Two processes on two different hosts that communicate using *sockets*
  - A socket is like a door through which messages are sent and received
    - Interface between the application process and the transport layer
- Addressing a *process*
  - Globally unique IP address
  - Receive side port number

Processes and Sockets



Source: Computer Networking: A top down approach by Kurose and Ross

Ports and Servers

- Client contacts the server initially for all communications
  - Server should react to the initial contact – it keeps listening to the port
    - It has an initial “socket object” to accept connections
    - It creates a new socket dedicated to a particular client after connection
  - The initial socket object is what we loosely call as an “open” port
    - It is really a half-open object
- Popular standard protocols have assigned (fixed) port numbers
  - Clients are aware of these numbers before they place a call

Port Numbers Continued

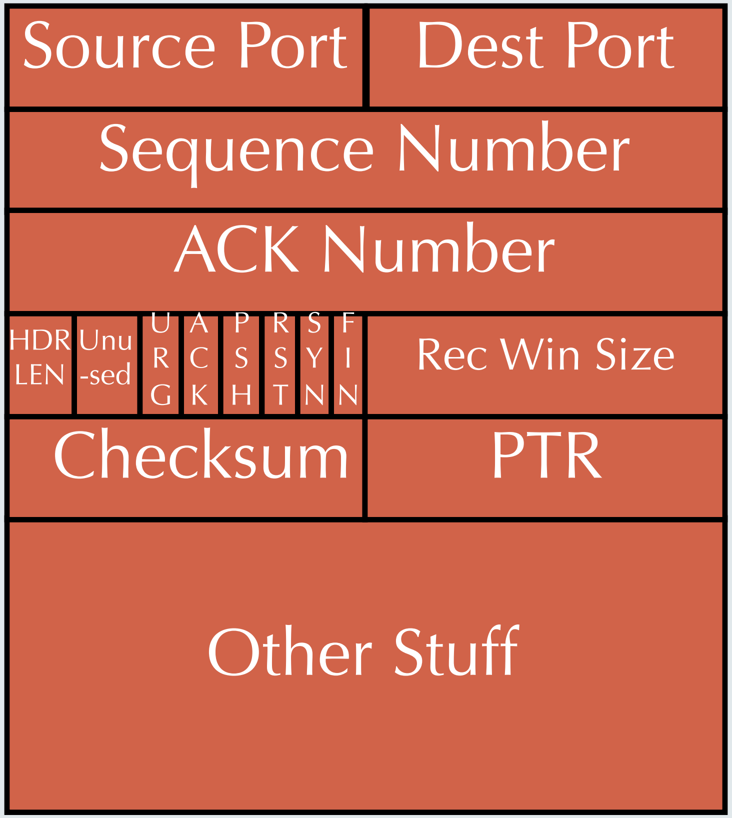
- Port numbers by convention are low numbered
  - Conventions are not always followed
  - In UNIX and UNIX-like OSs, port numbers smaller than 1024 are privileged
    - Only "root" can create these ports
    - Remote systems can trust the authenticity of these ports
- Some standard port numbers
  - Web server (http) – 80, (https) - 443;
  - DNS – 53;
  - Mail server (smtp) – 25; SSH – 22;
  - Telnet server – 23; FTP – 20 and 21;
  - POPv2 - 109, POPv3 - 110, IMAP - 143

TCP Review

- We know TCP as
  - A transport layer protocol that is carried by IP
  - A "packet" of TCP is called a segment and it is identified by a source port and a destination port
  - IP is unreliable - TCP maintains the sequence of packets in the right order and provides for acknowledgment and retransmission of lost packets
  - TCP provides flow control
    - It throttles the flow of packets if the receiver cannot handle the rate at which packets are sent
    - If a packet is dropped because of congestion, TCP will reduce the sending rate by changing the congestion window size
    - It limits the number of segments sent, but yet to be acknowledged

TCP Segment Structure

- There are six flag bits
- ACK - indicates its ACK field is valid
- RST, SYN and FIN are used for connection set up and tear down
- PSH - send data to higher layers right away
- URG - there is some urgent data



Source Port | Dest Port

Sequence Number

ACK Number

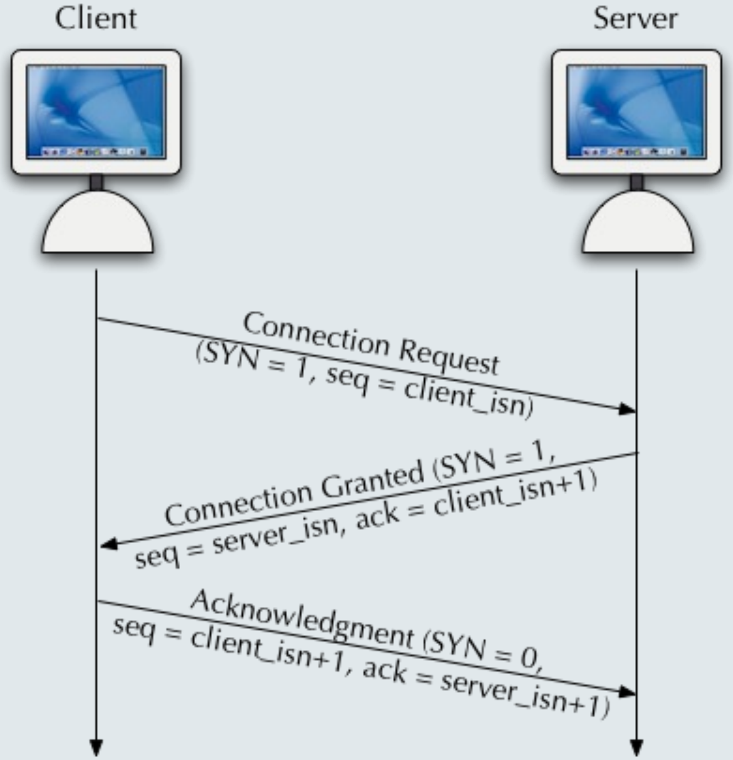
HDR LEN | Unu-sed | URG | ACK | PSH | RST | SYN | FIN | Rec Win Size

Checksum | PTR

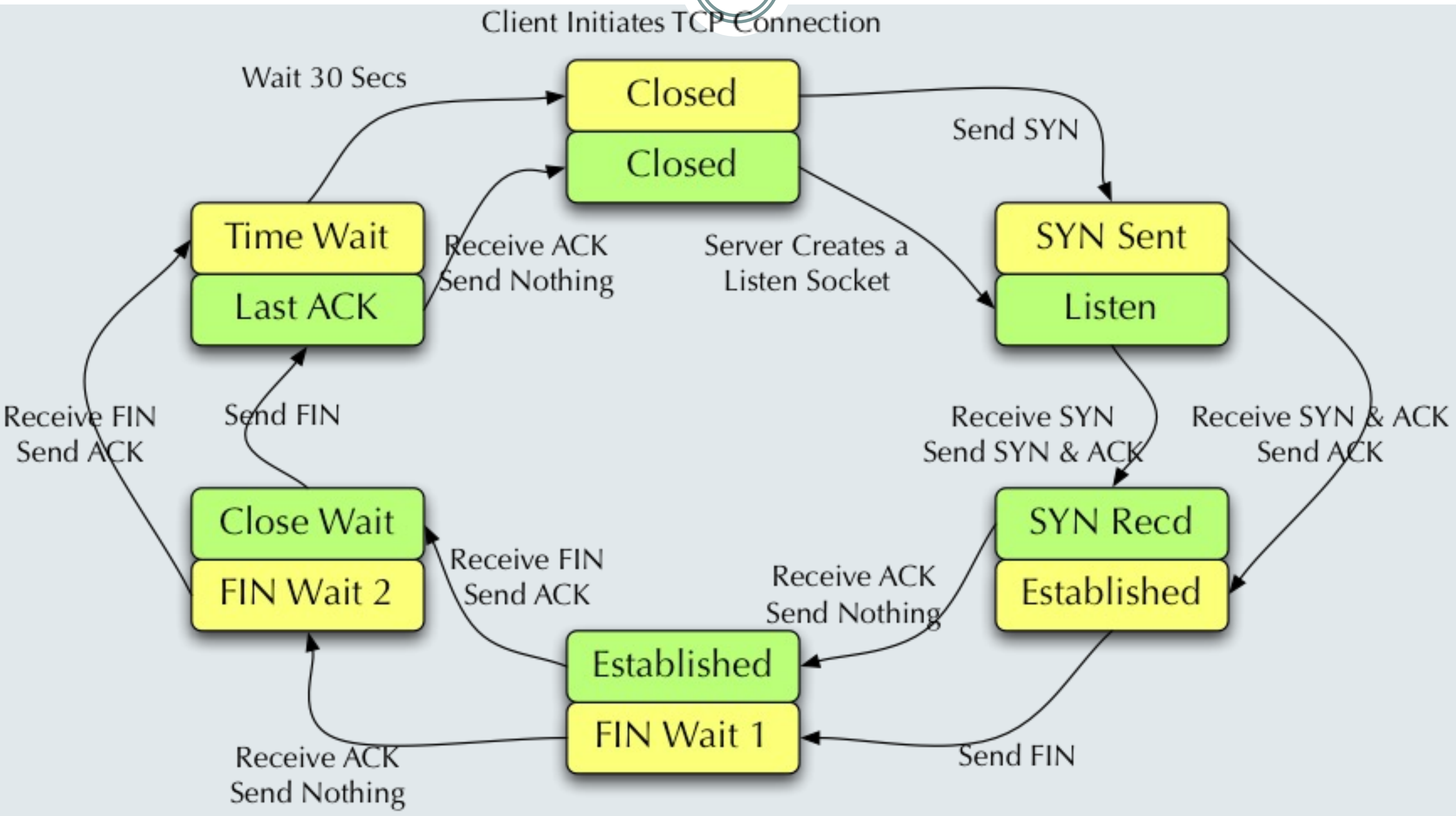
Other Stuff

TCP Connection Management

- Client wants to initiate connection to server
  - It sends a special TCP segment to the server with the SYN bit set to 1
  - The initial sequence number is say `client_isn`
  - This is called a SYN segment
- Server receives the SYN segment
  - It allocates buffers and variables to the connection and replies
  - Reply has SYN = 1, acknowledgment number = `client_isn` +1
  - Sequence number is `server_isn`
  - This is called a SYNACK segment
- Connection is completed


Client
Server
Connection Request
(SYN = 1, seq = client_isn)
Connection Granted (SYN = 1,
seq = server_isn, ack = client_isn+1)
Acknowledgment (SYN = 0,
seq = client_isn+1, ack = server_isn+1)

TCP States - Client and Server



Client Initiates TCP Connection

Closed

Closed

Wait 30 Secs

Send SYN

Time Wait

Last ACK

Receive ACK
Send Nothing

Server Creates a
Listen Socket

SYN Sent

Listen

Receive FIN
Send ACK

Send FIN

Receive SYN
Send SYN & ACK

Receive SYN & ACK
Send ACK

Close Wait

FIN Wait 2

Receive FIN
Send ACK

Receive ACK
Send Nothing

SYN Recd

Established

Established

FIN Wait 1

Receive ACK
Send Nothing

Send FIN

Connection Termination

- The graceful method to terminate the connection is to use the FIN field followed by ACK
  - In this case, either the client or the server will first send a TCP segment with the FIN bit set
  - The receiving host will ACK the FIN
  - This process closes *half* the connection - it has to be repeated by the receiving host
- The abrupt method of closing the TCP connection is for either the client or the server to send an RST (reset) segment
  - This aborts the TCP connection and no further communications take place between the hosts

Sequence Numbers in TCP

- Sequence and acknowledgment numbers are very important in TCP for reliable data transfer
- The sequence number of a TCP segment tells the receiver how many bytes of data has been sent
  - Example: the first TCP segment carries 1000 bytes of data and the sequence number is 235, the next TCP segment will have a sequence number 1235
- The acknowledgment number tells the recipient what is the next expected byte number
  - Example: the server receives 1000 bytes from the TCP segment with sequence number 235 - it has received bytes numbered 235 through 1234. So its sets the ack number to be 1235

ICMP

- Internet Control Message Protocol - ICMP is supposedly a very low-key protocol to answer simple requests
  - It sits below the transport layer and above the IP layer of the protocol stack
  - No port numbers of any kind - but it has types and codes in the first two bytes of the header
  - No concept of client or server - effects are mostly internal to the recipient host
  - No guarantees of delivery
- Hosts need not be listening to ICMP messages
- ICMP messages can be broadcast to hosts
- Can be a source of information leaks - e.g. host is unreachable

ICMP Codes and Types

- ICMP contains first 8 bytes of IP header that caused the ICMP response
- Ping transmits ICMP (8,0) and receives ICMP (0,0)
- Traceroute uses ICMP
  - Sends an ICMP with TTL = 1, 2, 3, 4 … to destination
  - Each router along the path detects the TTL as expired and responds with an ICMP (11,0) allowing traceroute to determine the route

Type	Code	Remark
0	0	Echo reply (ping)
3	0	Destination Network Unreachable
3	1	Destination Host Unreachable
3	3	Destination Port Unreachable
8	0	Echo request
9	0	Router advertisement
11	0	TTL Expired
12	0	IP Header Bad

Legitimate ICMP Activity

- Routers deliver “host unreachable” message
  - Common when hosts are shut down for maintenance or otherwise
  - Can be used in reconnaissance information
- Port unreachable
  - ICMP can be used to check if a UDP port is open
  - TCP ports reply with a RST/ACK flags
- Routers sometime inform you that ICMP traffic is blocked!
- Router redirect messages
  - Informs host of a more optimum router
- Need to fragment packets because MTU is exceeded
- TTL expired (time exceeded in transit)

DNS

- Domain Name System
  - Maps host names to IP addresses and vice versa
    - A tree for forward queries – What is the IP address of www.kmutnb.ac.th?
    - A tree for inverse queries – What is the host name of 136.142.116.28?
  - Common implementation is *bind*
- DNS stores so-called resource records (RRs)
  - Can reveal a lot of information about hosts and addresses

DNS Vs Typical Client-Server

- Typical client-server interaction
  - Client request connection to server
  - Server responds - handshakes take place
  - Session is initiated with interaction only between the two entities
- DNS is a bit different
  - Client issues a DNS query to the server
  - Server accepts query - may contact other DNS servers
  - Upon obtaining the information, it returns it to the client

DNS Details

- Many protocols employ DNS to translate user supplied names to IP addresses
  - DNS has to be called by http, ftp, smtp etc.
  - DNS can add delay to the communications process
- DNS is an application level protocol, but is typically not used directly by the user
- DNS queries and responses are on port 53 using UDP
  - TCP is used for zone transfers

Other DNS Services

- In addition to address mapping, DNS provides
  - Host Aliasing (e.g. www.kmutnb.ac.th can have two aliases – kmutnb.ac.th and web.kmutnb.ac.th)
  - Mail Server Aliasing (e.g. phongsakk@kmutnb.ac.th has to go to mail.kmutnb.ac.th)
  - Load Distribution (e.g. many sites use replicated web servers each running on a different end-system host)
    - DNS responds with the entire set of hosts, but rotates the order periodically

Resource Records

- Resource records (RRs) store the hostname to IP address mapping
- Each RR has four fields
  - [Name, value, type, TTL]
  - Many different types
  - TTL specifies how long the RR is valid

Name Servers

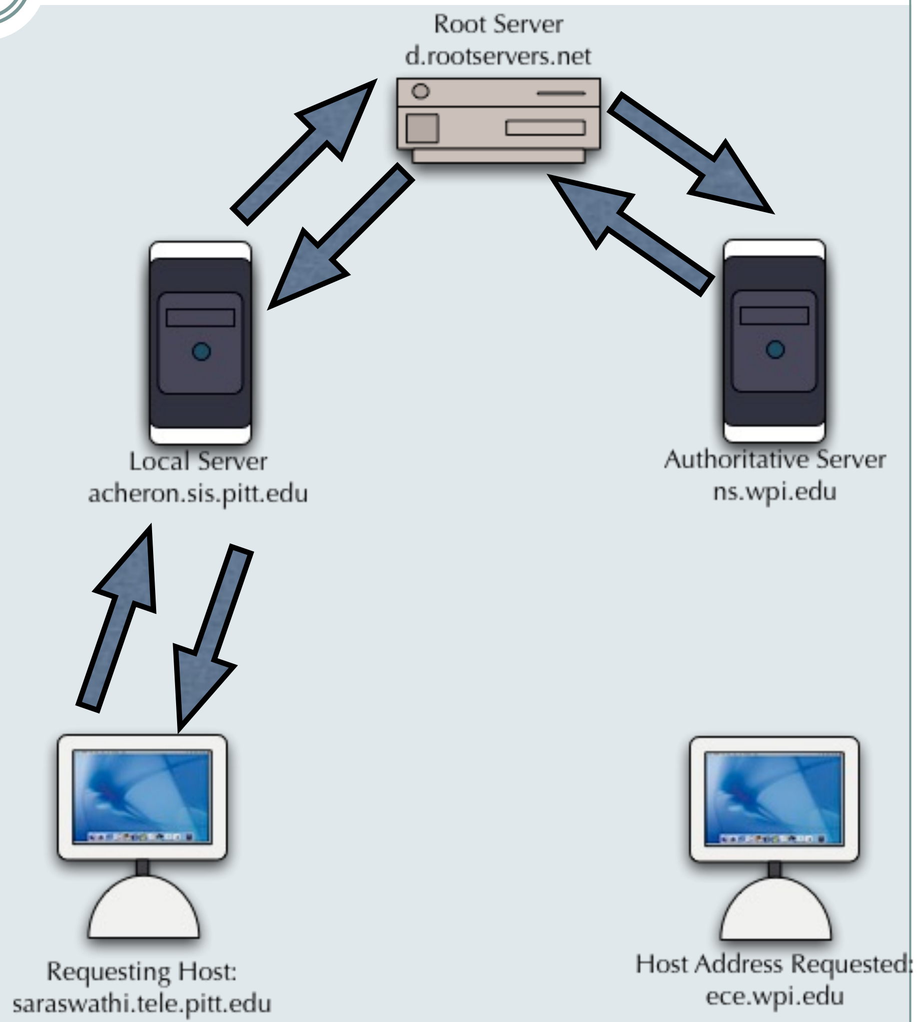
- Local Name Servers
  - Each ISP has its own name servers - all local machines contact the local name server first
  - Local translations are fast, simple and easy to implement
- Root Name Servers
  - Countable numbers worldwide (13)
  - Local servers contact the root server if they cannot resolve a name
- Authoritative Name Servers
  - Root servers direct local servers to an authoritative name server that has the information related to a host
  - Maintain authoritative data for a zone

Zone Transfers

- Zone
  - Name spaces are divided into zones based on separating “periods” in the name
  - Example: kmutnb.ac.th is a zone
- Each zone maintains primary and secondary name servers
  - Secondary servers periodically poll primary servers to obtain zone data
  - If data has changed, a zone transfer is initiated that downloads the entire database

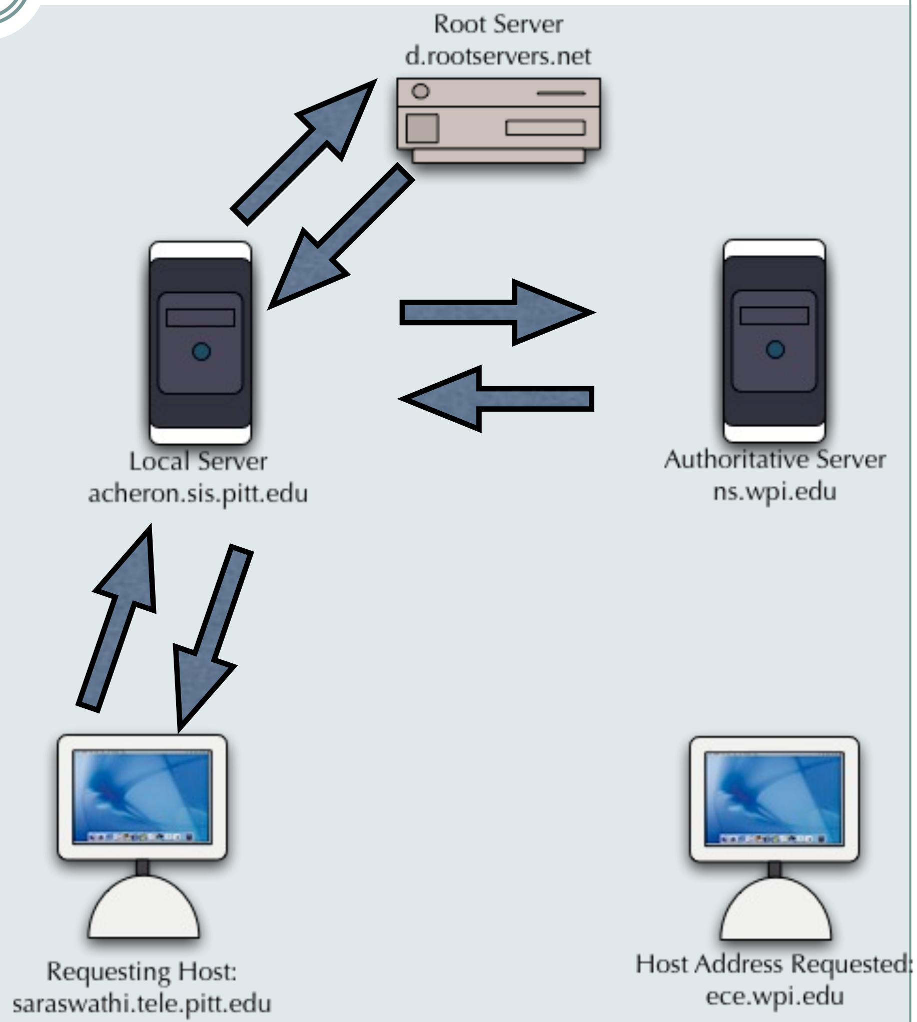
Recursive Queries

- Local server does not know the IP address of host
  - It contacts the root server
  - The root server also does not know the IP address
    - It contacts an authoritative name server that returns the address
  - Root server returns the address to local server
- Local server forwards the IP address to requesting host
- Intermediate servers may also be used



Iterative Queries

- If any server does not know the IP address, it may send the address of the next server in the list to the requesting host
- The requesting host makes direct request to the new name server
- Typically most requests are recursive, except when made to a root server
  - Query chains are a mix of iterative and recursive queries



Inverse Lookup

- Inverse look-ups are performed in a slightly different way by DNS servers
- Example: Lookup 136.142.116.28
  - The query resolves 28.116.142.136.in-addr.arpa
  - Similarities between forward and inverse look-up
    - The top-level domain "arpa" has exactly one sub-domain "in-addr"
    - The host address (say 28) comes first just like forward lookups
- Inverse trees are often not current and could lead to potential security problems

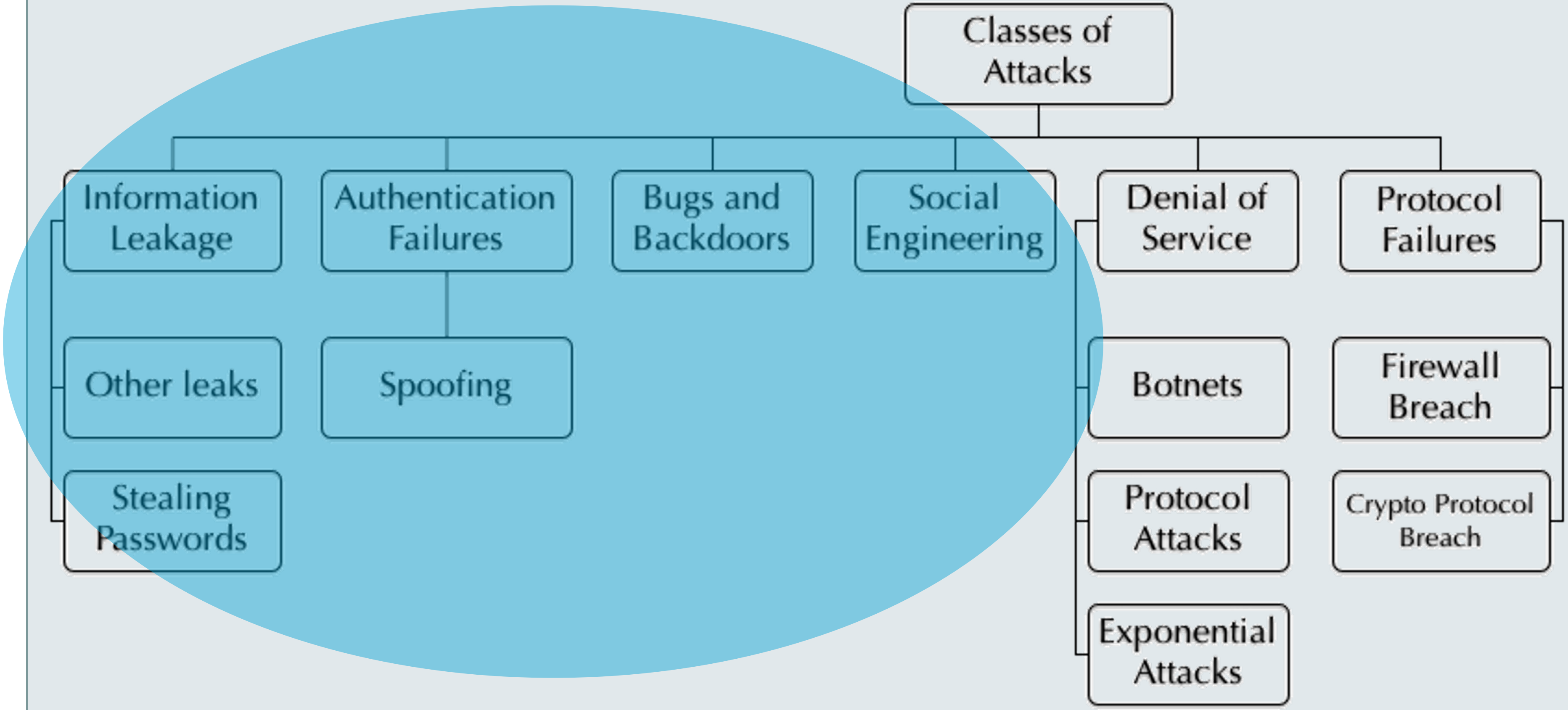
DNS Software

- Berkeley Internet Name Domain (BIND)
  - Most common implementation named
  - Many versions exist (latest is 9.3.y)
- ATLAS
  - Advanced Translation Look-up And Signaling
  - Verisign’s proprietary DNS software
- Microsoft has its own DNS software since Win2K
- djbdns - Free DNS software
  - See www.tinydns.org

The Security Breach Process

- Phases
  - Reconnaissance
  - Exploitation
  - Reinforcement
  - Consolidation
  - Pillage
- Network and protocol complexity and weakness aids this process

Classes of Attacks



Process and Attacks - 1

- Reconnaissance
  - Makes extensive use of “Information Leakage”
  - Passive and active leakage is possible
    - Passive reconnaissance is hard to detect
    - Example: Google search
- Active reconnaissance could appear “normal”
  - Use protocols the way they are supposed to be used
- Other reconnaissance tactics are blatant
  - Port scanning, directed broadcast and so on

Process and Attacks - 2

- Exploitation, Reinforcement and Consolidation
  - Make use of stolen passwords, authentication failures, social engineering, and bugs and backdoors
- A combination of techniques can be used
  - Bugs are hard to prevent
  - Procedures and training can help prevent social engineering attacks
  - Security measures can prevent passwords from being stolen and authentication failures

Process and Attacks - 3

- Pillage
  - Good example is Denial of Service Attacks
  - We consider this later

Stealing Passwords

PACKETS FLOWING ON THE NETWORK

FILES STORED ON HOSTS ACCESSIBLE THROUGH THE NETWORK

Stealing Passwords

- Special case of information leakage
  - Larger impact on security
  - If passwords are stolen, Oscar can do a lot more than just get information
    - Oscar can move from being an outsider to a legitimate user to administrator in steps - reinforcement and consolidation
  - Harder to detect attacks because Oscar looks like a legitimate user
- Many ways in which passwords can be stolen

How passwords can be stolen

- Password in cleartext
  - Password and login are sent in cleartext by some protocols
  - Several cracker tools exist to sniff packets and get passwords
- Dictionary attacks
  - Access to the hashed password file (information leak)
  - Users typically choose a small subset of passwords – not one random password of 2^{80} choices
  - Faster to break using current technology
- Other attacks
  - Crafted Javascripts can fool users into revealing passwords
  - Other social engineering attacks

Passwords in Cleartext

- Several protocols transmit passwords in cleartext
  - Telnet
  - POP (older versions)
  - Basic authentication performed by web servers
  - SNMPv1
- Tools exist that can sniff these packets and recover passwords
  - Trivial to use and requires no knowledge of networking, protocols or programming

Dictionary Attacks

- How can Oscar get access to a hashed password?
  - Revealed /etc/passwd files
    - TFTP, SMB, NIS
    - Compromised hosts
      - Keys of ssh may also be attacked this way
  - Hashed password on the physical medium
    - POPv3
    - Digest authentication by web servers
- Password guessing and dictionary attacks
  - Given access to a password file (encrypted), Oscar tests each password to see if there is a match
  - Easy to do since the hash function is known
  - To improve the probability of success, Oscar tries common words, proper names, lowercase strings etc. – dictionary attacks
  - He can use information obtained through leakage to improve his attack!

Authentication Failures

SPOOFING ATTACKS

Authentication Failure

- Definition of sorts
  - Mechanisms to verify that the source of a request or command is legitimate fail to stop Oscar
- Common examples
  - SMTP
    - You trust that the e-mail originated from the person whose e-mail address shows up in the *From* field
    - You cannot really trust this…
  - The "r" commands
    - Commands like rlogin, rsh, etc. depend on the source address of the requesting host + assertion of username as verification of legitimacy
- Cryptographic authentication protocols are a *must*, but typically not used

IP Spoofing

- IP addresses
  - In IPv4, the address source address and destination address are both 32 bits long
  - The 32 bits are divided into two parts
    - Network portion and Host portion
  - Today people use *classless inter-domain routing* (CIDR)
    - Example: 136.142.116.28/24 means the first 24 bits are the network field
- IP address as authenticators
  - A lot of services and tools use the IP source address for authentication
  - If the IP source address is valid, trust the packet and the request!
- **You cannot rely on the validity of the source IP address except under very controlled circumstances**

ARP Spoofing

- ARP = Address Resolution Protocol
  - The link/MAC layer does not understand IP addresses
  - The NIC can only recognize MAC addresses
  - ARP is used to map the MAC address to the IP address
  - ARP packets are broadcast packets (on a LAN for example)
- If Oscar can write to the local network he can
  - Emit false ARP queries or replies
- Impact
  - Oscar can divert traffic to himself and modify data before sending it to the destination
- Notes
  - Hacker tools exist to do ARP spoofing
  - In IPv6, a “neighbor discovery” or ND protocol is used instead of ARP and can create more serious problems if spoofed

TCP and UDP

- Cannot trust privileged port numbers from TCP
  - In UNIX and UNIX-like OSs, port numbers below 1024 are “privileged”
    - Only “root” can access these numbers
  - This is meaningless for other OSs
    - Also easy to spoof the port numbers in specially crafted packets
  - We consider attacks on TCP in more detail next week
- UDP sequence numbers can be easily spoofed
  - Since there are no handshakes with UDP, it is easier to spoof UDP
  - UDP carries several services (like DNS) and can be dangerous

DNS and Authentication Failures

- DNS reverse lookup is used to authenticate the "r" commands
  - If Oscar controls the reverse lookup tree by some chance, he can falsify it
  - Inverse record will contain the name of a machine that your machine trusts
- Newer DNS lookups perform the lookup both ways to prevent such attacks
  - Cross-checking is done by the *gethostbyaddr*
  - If anomalies are detected, they should be logged

DHCP/BOOTP

- DHCP = Dynamic Host Configuration Protocol
  - Used to assign IP addresses
  - Supply information about name servers, gateways, etc.
  - Client sends a UDP broadcast request
  - Server replies with information
  - Can interface with name servers to enable mapping names to IP addresses
- Can supply a lot of information
- Logs are important for forensics
- Used only on local networks
  - Needs to know the MAC address of client
  - Reduces risks, but spoofed messages can divert traffic
  - Easier to spoof ARP and achieve the same objectives

Cookies etc.

- HTTP is stateless
  - Each HTTP request and response are treated in isolation
  - Hard for web servers to determine their state with the client they are serving
- Cookies
  - Maintain state information for servers
  - Sometimes hidden input fields or special fields in URLs are used to maintain state
  - Some web servers rely on cookies for authentication
- Cookies can be easily spoofed
  - Users can change cookies
  - Server can encrypt cookies but it is subject to other kinds of attacks (like?)
- Canned shopping carts…

Other authentication failures

- RPC and RPCBind
  - Easy to spoof userid, groupid, machine name, etc.
  - RPCbind
    - More dangerous since you can ask RPCbind to issue an indirect call to a service
  - Solutions
    - Use kerberized version of RPC
- NTP – Network Time Protocol
  - Some authentication tokens are designed to expire after a “lifetime”
    - Example: Kerberos
  - Spoofed to allow replay of authentication tokens

Bugs and Backdoors

- *Buffer overflows* are the biggest problem in creating bugs and backdoors
  - Example: finger daemon and the Internet Worm
- Protocols that have seen many bugs
  - Sendmail
  - RPC
  - NFS
  - FTP
- Another common problem is misconfiguration
  - FTP daemon
  - Anonymous FTP sites
  - Example: Java FTP client

Bugs and Backdoors - II

- Other non-obvious ways of exploiting bugs and backdoors
  - HTTP returned documents
    - May request a specific program to process them
  - Spyware, Adware, Foistware
    - No patching
  - ActiveX
    - If the code is signed, it can be trusted!
  - Browsers that allow weak ciphersuites
- Poorly written server scripts
  - Provide entry points for Oscar to insert malicious code

Social Engineering

- Read Kevin Mitnick, “The Art of Deception”
- E-mails, URLs and Javascripts
- Phishing and Pharming
- Ignorance and naiveté
- Carelessness
- FIS suggests near-paranoid behavior