Network Protocols

Security
Securing the Internet is hard!

- Lots and lots of things need to be secured
- Software tends to be buggy and poorly designed
  - There is lots of software distributed everywhere
- Vendors ship bad or poor default configurations
- Each system user is required to be a sysadmin
- Few people are really good at security
- One person's security problem is also another's
Compare the teleco model

- Telco
  - Centralized control
  - Intelligent network
  - Fixed parameters
- Internet
  - Distributed control
  - Intelligent hosts
  - Bursty traffic
Where does security belong?
The end-to-end (e2e) argument

- Functionality at the ends of the communication
  - Internet hosts end up running lots of software
- Peer-to-peer is not a feature, it's built-in
  - You have to purposely turn it off (filter/firewall)
- Securing the ends or close to them is a necessity
  - This is extremely hard to do on a large scale
- We don't know how to implement e2e security yet
  - We use hacks that fit badly with the architecture
Layered defenses

- The belt and suspenders approach
- Place security mechanisms throughout the system
- There may be a layer that attackers can't break
- Multiple layers tend to slow attackers down
- Failure at one layer isn't detrimental to the system
Address resolution protocol (ARP) security

- Layer 2 / layer 3 address mapping mechanism
  - Sending host needs MAC address of IP receiver
  - Sends ARP broadcast containing IP address
  - All MAC stations see it, one with the IP responds

- No authentication mechanisms in ARP
- Any station could masquerade as any IP
- Hosts/have a limited ARP cache/table
  - What happens when it gets full?
Dynamic Host Configuration Protocol (DHCP) attacks

- Temporary, automatic IP address assignment
  - Host requesting IP sends limited broadcast
  - If available, DHCP responds with config info
  - Host generally accepts first response
- DHCP authentication is not widely implemented
- Rogue servers can supply bogus configuration
- Rogue end hosts can use up leased IP addresses
Internet Control Message Protocol (ICMP) security

• Status and error messages for communications
• Some uses include:
  • testing connecting (ping)
  • detecting errors (unreachables)
  • reporting the need for fragmentation (path MTU)
• Some ICMP can induce DoS or information leaks
  • Relatively easy to guard against
Domain Name System (DNS) security

- Most commonly used to map query to a RR
  - For example, what is IP for www.depaul.edu?
- Authentication is not widely supported
- Cache poisoning tricks lead to DoS, interception
- Potential for information leakage
- One of two (other being routing) critical services
  - Requires effort and energy to protect fully
Routing security

- Routers forward towards ultimate destination
  - Critical to protect again DoS, interception
- Can use authentication and filtering liberally
  - Accept only valid routes from other routers
  - Ignore bogon source/destination IPs in packets
- Multicast is a whole 'nother ball game
  - we'll discuss this later, very difficult to secure
Perimeter security

- Define a boundary
- Separate a trusted inside from a untrusted outside
- Typical example is the network-based firewall
Network-based firewalls

- Centralize control of boundary/border crossings
- Limit the type of traffic that can pass
- Generally a network solution to an end problem
- Network inspection of end-app data is difficult
- These often eliminate useful types of traffic
- Often perpetuates neglect of fixing end problems
- JTK: people tend to spend too much energy here
Packet filtering

- On packet-by-packet basis, inspect and filter/pass
- Can inspect:
  - Protocol types (IP, UDP, TCP, ICMP, etc.)
  - Sources and destinations (e.g. IP address)
  - Protocol control fields (e.g. TCP flags)
  - Other custom pattern matches (e.g. “root”)
Stateful inspection

• Keep track of a session between hosts
• Significant security improvement over packet filters
• Often used to limit inbound connections to hosts
• Increases complexity of communication system
  • Complexity is the enemy of reliability
• End host robustness is reduced
  • Apps fail if “state” is corrupted or goes away
The screened subnet
Application layer gateways aka proxy firewalls

- No direct communication across boundary
- Requires lots of state, fate and probably complexity
- Desired protocols/apps must be supported by proxy
An aside: TCP 3-way handshake
Example packet filter: ipchains

Don't want to see packets with private IP addresses

-A input -s 192.168.0.0/255.255.0.0 -d 0/0 -j DENY
-A input -s 172.0.0.0/255.240.0.0 -d 0/0 -j DENY
-A input -s 10.0.0.0/255.0.0.0 -d 0/0 -j DENY

Let SSH, established TCP connections, FTP data, UDP and BOOTP/DHCP in

-A input -s 0/0 -d a.b.c.d/255.255.255.255 22:22 -p 6 -j ACCEPT
-A input -s 0/0 -d a.b.c.d/255.255.255.255 1024:65535 -p 6 ! -y -j ACCEPT
-A input -s 0/0 20:20 -d 0/0 1024:65535 -p 6 -y -j ACCEPT
-A input -s 0/0 -d 0/0 1024:65535 -p 17 -j ACCEPT
-A input -s 0/0 -d 0/0 67:67 -p 17 -j ACCEPT

Drop any packets that don't have our source IP and log those attempts

-A output -s 140.192.0.1/255.255.255.255 -d 0/0 -j DENY -l
Example packet filter: cisco ACL

Block private IP addresses

access-list 100 deny ip 192.168.0.0 0.0.255.255 any
access-list 100 deny ip 172.0.0.0 0.15.255.255 any
access-list 100 deny ip 10.0.0.0 0.255.255.255 any

Block source port of 111 from going anywhere

access-list 100 deny tcp any eq sunrpc any
access-list 100 deny udp any eq sunrpc any

Allow DNS and TELNET (log it) to 1.2.3.4, deny everything else

access-list 100 permit tcp any host 1.2.3.4 eq domain
access-list 100 permit tcp any host 1.2.3.5 eq telnet log
access-list 100 deny ip any any
Example packet filter: ipf

Allow SSH in

pass in quick on fxp0 proto tcp from any to any port=22 flags S keep state

Block bogus addresses

block in quick on fxp0 from any to 10.0.0.0/8
block in quick on fxp0 from any to 172.16.0.0/12
block in quick on fxp0 from any to 192.168.0.0/16

Allow outbound ICMP

pass out quick on fxp0 proto icmp from any to any keep state
How to defeat a firewall

- Disguise packets to pass firewall rules
- DoS attack firewall (make it inoperable)
- Compromise the firewall
- Get hosts/users inside to do something dumb
- Go around
Intrusion detection systems

- Examine packet-by-packet, stateful or anomalies
- Inspect, report and possibly respond to intrusions
- Difficult to minimize false positives/negatives
- Can often result in information overload
- Useful where firewalls cannot be deployed
How defeat an IDS

- Fragment packets
- Use encryption or uncommon data encoding
- Go fast and/or DoS the IDS
- Inject background noise
- Tunnel protocols and applications
- Compromise the IDS
- Go around
Honeypots

- Closely monitored system that welcomes attacks
- Useful tool to study attacks and threats
- There is some inherent liability and risk involved
Encryption

- Try to make something readable, unreadable
- Generally requires complicated math algorithms
- Encryption strength relies on cipher and key length
- Plain text -> cipher text -> plain text
- Safekeeping of the decryption keys is... key
  - Public versus private keys
  - How to do key exchange securely?
  - Key escrow, recovery and trusted third parties
Shared secrets
aka symmetric encryption

• Each communicating party shares the secret key
• The secret key can be used to encrypt/decrypt
• Safekeeping the key gets harder as users increase
• How do trusted parties learn the key?
• Example:
  • Ciphertext: 7,23,4-52,32,6
  • Key: Book=Ulysses:Page,Line,Word
Public key cryptography

- Everyone has a 2-key pair, one private, one public
  - The key pair are mathematically related
  - Should be difficult to deduce one from the other
- Public key can be widely published, used to encrypt
- Private key decrypts public key encrypted message
- Owner of the key pair, must safeguard private key
Cryptography illustrated
Virtual private networks

- Using encryption, protects data between endpoints
- Used to help secure and insecure public network
- IPSec protocols are typically used
- Often used to make ends appear on a trusted net
- Usually only guards against network eavesdropping
How to defeat VPNs
Kerberos

- Network-based authentication/authorization service
- Also used to encrypt network traffic
- Time limited ticket granting system used
- Centralized server for management and control
- Applications and protocols must support kerberos
Network address translation

- A solution designed for an address space problem
- Converts internal info to something used externally
  - IP addresses (NAT)
  - Port addresses (PAT)
- Significant complexity, state and fate issues
- Often applied as a security solution - wrongly IMHO
- NAT really sucks!
Investigating your target

• Network/host probes
  • ping, traceroute, nmap, nbtstat
• Publicly available information
  • News reports, DNS, search engines, data leaks
Authentication

- Password sniffing and capture
- Password cracking and brute force attacks
- Strong encryption should be used
- If possible authenticate in both directions
- Poor authentication protocols by default:
  - HTTP, TELNET, FTP, SMTP, POP3
- Better protocols to be using:
  - SSH, SSL, kerberos
Weak validation of input

- Software errors taken advantage of by user input
- Usually in the form of overflows or format strings
  - strcpy(d-variable, s-variable)
  - snprintf() and printf() %<format> trickery
- Programs often run as root/administrator
- Overflow data contains low level instructions
  - Generally not good
Denial of service

- Prevents or impairs standard service
- Source is commonly spoofed
- Extremely difficult problem to solve
Basic SMURF attack
Basic DDoS attack
SYN flooding and session hijack