Motivation

- Internet currently used for important services
  - financial transactions, medical records

- Could be used in the future for critical services
  - 911, surgical operations, energy system control, transportation system control

- Networks more open than ever before
  - global, ubiquitous Internet, wireless

- Malicious Users
  - selfish users: want more network resources than you
  - malicious users: would hurt you even if it doesn’t get them more network resources
Network Security Problems

- Host Compromise
  - attacker gains control of a host

- Denial-of-Service
  - attacker prevents legitimate users from gaining service

- Attack can be both
  - e.g., host compromise that provides resources for denial-of-service

- Other forms of attack
  - less common today because these two are so easy
Other Forms of Security

- Prevent malicious users from
  - reading transmitted data (privacy)
  - pretending to be someone else (authentication)
  - doing something without permission (authorization)
  - modifying transmitted data (integrity)
  - claiming they did not send a message (nonrepudiation)

- Detect
  - a compromise by a malicious user (intrusion detection)
Host Compromise

- One of earliest major Internet security incidents
  - Internet Worm (1988): compromised almost every BSD-derived machine on Internet

- Today: estimated that a single worm could compromise 10M hosts in < 15 min

- Attacker gains control of a host
  - reads data
  - erases data
  - compromizes another host
  - launches denial-of-service attack on another host
Definitions

- **Trojan**
  - relies on user interaction to activate
  - usually relies on user exploitation
- **Worm**
  - replicates itself
  - usually relies on stack smash attack
- **Virus**
  - worm that attaches itself to another program
Host Compromise: Stack Smash

- typical code has many bugs because those bugs are not triggered by common input
- network code is vulnerable because it accepts input from the network
- network code that runs with high privileges (i.e., as root) is especially dangerous
  - e.g., web server
Example

- what is wrong here:

```c
// Copy a variable length user name from a packet
#define MAXNAMELEN 64
char username[MAXNAMELEN];
int offset = OFFSET_USERNAME;
int name_len;

name_len = packet[offset];
memcpy(&username, packet[offset + 1], name_len);
```
Effect of Stack Smash

- Write into part of the stack or heap
  - write arbitrary code to part of memory
  - cause program execution to jump to arbitrary code

- Stack Smashing Worm
  - probes host for vulnerable software
  - sends bogus input
  - attacker can do anything that the privileges of the buggy program allows
    - launches copy of itself on compromised host
  - rinse, repeat at exponential rate
  - 10M hosts in < 15 minutes
Hall of Shame

- Software that have had many stack smash bugs:
  - BIND (most popular DNS server)
  - RPC (Remote Procedure Call, used for NFS)
    - NFS (Network File System), widely used at UCB
  - sendmail (most popular UNIX mail delivery software)
  - IIS (Windows web server)
  - SNMP (Simple Network Management Protocol, used to manage routers and other network devices)
Solution

- Don’t write buggy software
  - it’s not like people try to write buggy software
- Type-safe Languages
  - unrestricted memory access of C/C++ contributes to problem
  - use Java, Perl, or Python instead
- OS architecture
  - compartmentalize programs better, so one compromise doesn’t compromise the entire system
  - e.g., DNS server doesn’t need total system access
  - e.g., web server probably doesn’t need to complete write access
- Firewalls
Firewalls

- Gateway machine that blocks out certain data, e.g.,
  - any external packets not for port 80
  - any external packets with an internal IP address
    - ingress filtering
  - any email with an attachment

- Properties
  - easier to deploy firewall than secure all internal hosts
  - doesn’t prevent user exploitation
  - tradeoff between availability of services (firewall passes more ports on more machines) and security
    - if firewall is too restrictive, users will find way around it, thus compromising security
    - e.g., have all services use port 80
Host Compromise: User Exploitation

- Some security architectures rely on the user to decide if a potentially dangerous action should be taken, e.g.,
  - run code downloaded from the Internet
    - “Do you accept content from Microsoft?”
  - run code attached to email
    - “subject: You’ve got to see this!”
  - allow a macro in a data file to be run
    - “Here is the latest version of the document.”
User Exploitation

- Users are not good at making this decision
  - Which of the following is the real name Microsoft uses when you download code from them?
    - Microsoft
    - Microsoft, Inc.
    - Microsoft Corporation

- Typical email attack
  - Attacker sends email to some initial victims
  - Reading the email / running its attachment / viewing its attachment opens the hole
  - Worm/trojan/virus mails itself to everyone in address book
Solutions

- OS architecture
- Don’t ask the users questions which they don’t know how to answer anyway
- Separate code and data
  - viewing data should not launch attack
- Be very careful about installing new software
Denial of Service

- Huge problem in current Internet
  - Yahoo!, Amazon, eBay, CNN, Microsoft attacked in 2001
  - 12,000 attacks on 2,000 organizations in 3 weeks
  - some more that 600,000 packets/second
    - more than 192Mb/s
  - almost all attacks launched from compromised hosts

- General Form
  - prevent legitimate users from gaining service by overloading or crashing a server
  - e.g., spam, SYN attack
SYN Attack

- Compromised hosts send TCP SYN packets to target
  - sent at max rate with random spoofed source address
    - spoofing: use a different source IP address than own
    - random spoofing allows one host to pretend to be many

- Victim receives many SYN packets
  - sends SYN+ACK back to spoofed IP addresses
  - holds some memory until 3-way handshake completes
    - usually never, so victim times out after long period (e.g., 2 minutes)
Affect on Victim

- buggy implementations allow unfinished connections to eat all memory, leading to crash
- better implementations limit the number of unfinished connections
  - once limit reached, new SYNs are dropped
- victim’s network connection also saturated
- affect on victim’s users
  - users can’t access the targeted service on the victim because the unfinished connection queue is full
  - users can’t access the other services in victim’s network because connection is saturated
Other Denial-of-Service Attacks

- SYN attack is simple
- more sophisticated attacks possible
  - attack DNS, BGP
  - reflection
  - cause one non-compromised host to attack another
  - e.g., host A sends DNS request with source B to server C. C sends reply to B.
Dealing with Attack

- distinguish attack from flash crowd
- prevent damage
  - distinguish attack traffic from legitimate traffic
  - rate limit attack traffic
- stop attack
  - identify attacking machines
  - shutdown attacking machines
  - usually done manually, requires cooperation of ISPs, other users
- identify attacker
  - very difficult, except
  - usually brags/gloats about attack on IRC
  - also done manually, requires cooperation of ISPs, other users
Incomplete Solutions

- Quality of Service
  - Fair queueing, Integrated Services, Differentiated Services, RSVP
  - prevent a user from sending at 10Mb/s and hurting a user sending at 1Mb/s
  - does not prevent 10 users from sending at 1Mb/s and hurting a user sending a 1Mb/s
Identifying Attacking Machines

- Defeat spoofed source addresses
- Does not stop or slow attack
- Egress filtering
  - a domain’s border router drop outgoing packets which do not have a valid source address for that domain
  - if universal, could abolish spoofing (why isn’t it universal?)
- IP Traceback
  - routers probabilistically tag packets with an identifier
  - destination can infer path to true source after receiving enough packets
Summary

- Network security is possibly the Internet’s biggest problem
  - preventing Internet from expanding into critical applications
- Host Compromise
  - poorly written software
  - Solutions: better OS security architecture, type-safe languages, firewalls
- Denial-of-Service
  - no clear solution