CSC 4103 - Operating Systems
Spring 2007

Lecture - XX
Protection and Security

Tevfik Koşar

Louisiana State University
April 12th, 2007

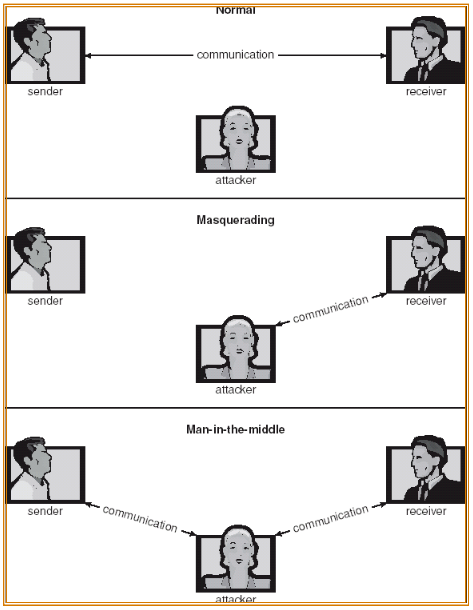
The Security Problem

- Security must consider external environment of the system, and protect the system resources
- Intruders (crackers) attempt to breach security
- **Threat** is potential security violation
- **Attack** is attempt to breach security
- Attack can be accidental or malicious
- Easier to protect against accidental than malicious misuse

Security Violations

- Categories
  - **Breach of confidentiality** (information theft, identity theft)
  - **Breach of integrity** (unauthorized modification of data)
  - **Breach of availability** (unauthorized destruction of data)
  - **Theft of service** (unauthorized use of resources)
  - **Denial of service** (crashing web servers)
- Methods
  - **Masquerading (breach authentication)**
    - Pretending to be somebody else
  - **Replay attack (message modification)**
    - Repeating a valid data transmission (eg. Money transfer)
    - May include message modification
  - **Session hijacking**
    - The act of intercepting an active communication session
  - **Man-in-the-middle attack**
    - Masquerading both sender and receiver by intercepting messages

Standard Security Attacks

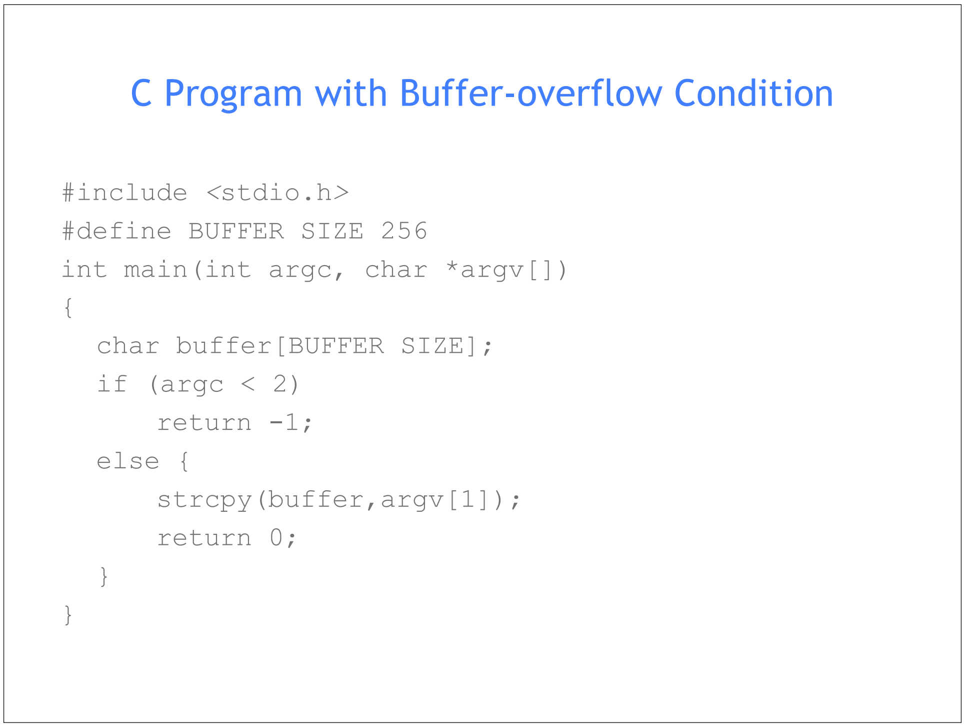


Security Measure Levels

- Security must occur at four levels to be effective:
  - Physical
  - Human
    - Avoid **social engineering, phishing, dumpster diving**
  - Operating System
  - Network
- Security is as week as the weakest chain

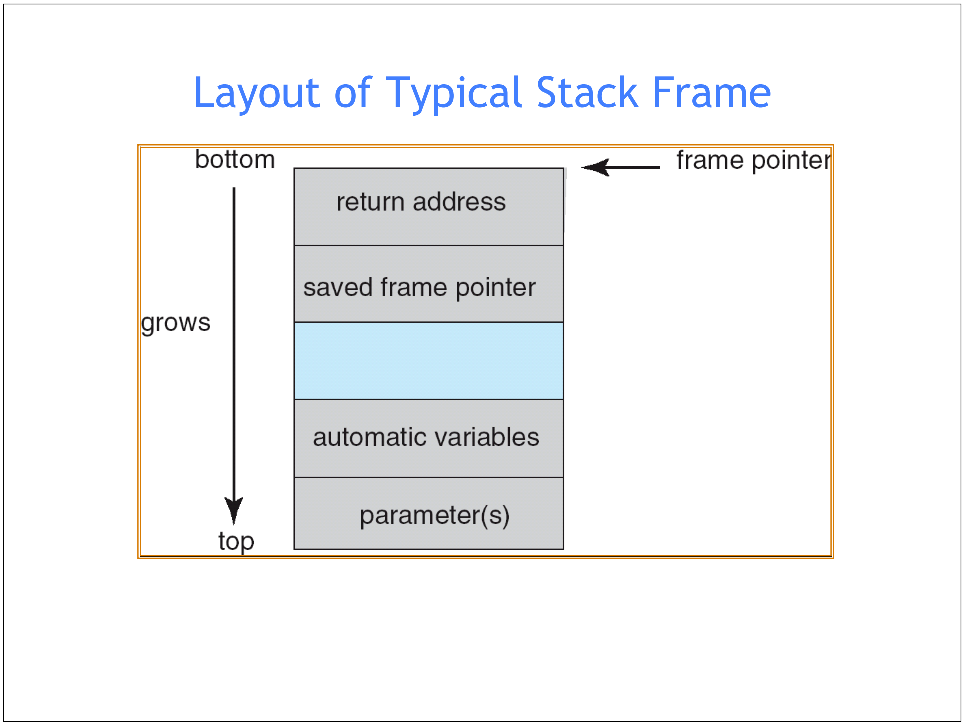
Program Threats

- Trojan Horse
  - Code segment that misuses its environment
  - Exploits mechanisms for allowing programs written by users to be executed by other users
  - **Spyware, pop-up browser windows, covert channels**
- Trap Door
  - Specific user identifier or password that circumvents normal security procedures
  - Could be included in a compiler
- Logic Bomb
  - Program that initiates a security incident under certain circumstances
- Stack and Buffer Overflow
  - Exploits a bug in a program (overflow either the stack or memory buffers)



C Program with Buffer-overflow Condition

```
#include <stdio.h>
#define BUFFER SIZE 256
int main(int argc, char *argv[])
{
  char buffer[BUFFER SIZE];
  if (argc < 2)
      return -1;
  else {
      strcpy(buffer,argv[1]);
      return 0;
  }
}
```

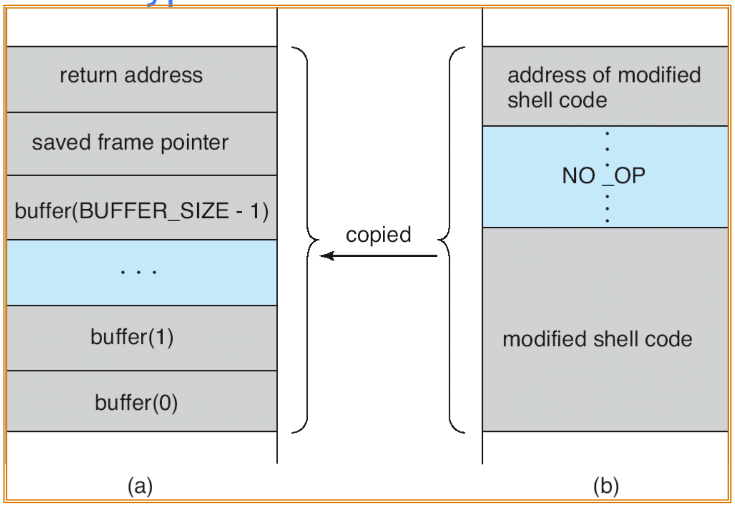


Layout of Typical Stack Frame

Modified Shell Code

```
#include <stdio.h>
int main(int argc, char *argv[])
{
  execvp(‘‘\bin\sh’’,‘‘\bin\sh’’, NULL);
  return 0;
}
```

Hypothetical Stack Frame



Before attack

After attack

Program Threats (Cont.)

- Viruses
  - Code fragment embedded in legitimate program
  - Very specific to CPU architecture, operating system, applications
  - Usually borne via email or as a macro
    - Visual Basic Macro to reformat hard drive

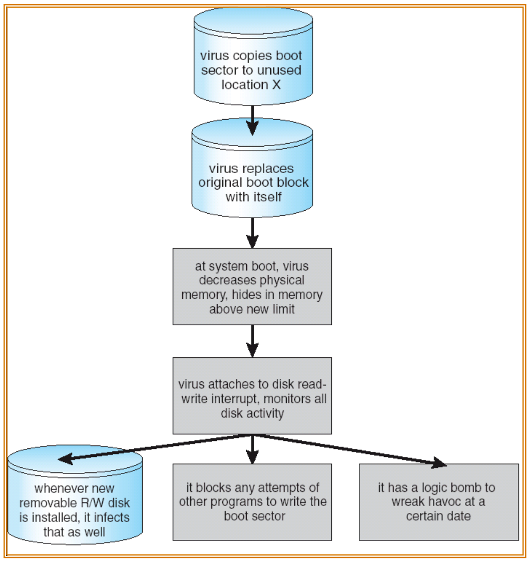
```
Sub AutoOpen()
Dim oFS
  Set oFS = CreateObject(''Scripting.FileSystemObject'')
  vs = Shell(''c:command.com /k format         c:'',vbHide)
End Sub
```

Program Threats (Cont.)

- **Virus dropper** inserts virus onto the system
- Many categories of viruses, literally many thousands of viruses:
  - File (appends itself to a file, changes start pointer, returns to original code)
  - Boot (writes to the boot sector, gets exec before OS)
  - Macro (runs as soon as document containing macro is opened)
  - Source code (modifies existing source codes to spread)
  - Polymorphic (changes each time to prevent detection)
  - Encrypted (first decrypts, then executes)
  - Stealth (modify parts of the system to prevent detection, eg read system call)
  - Tunneling (installs itself as interrupt handler or device driver)
  - Multipartite (can infect multiple pars of the system, eg. Memory, bootsector, files)
  - Armored (hidden and compressed virus files)

  - Browser virus, keystroke logger ..etc

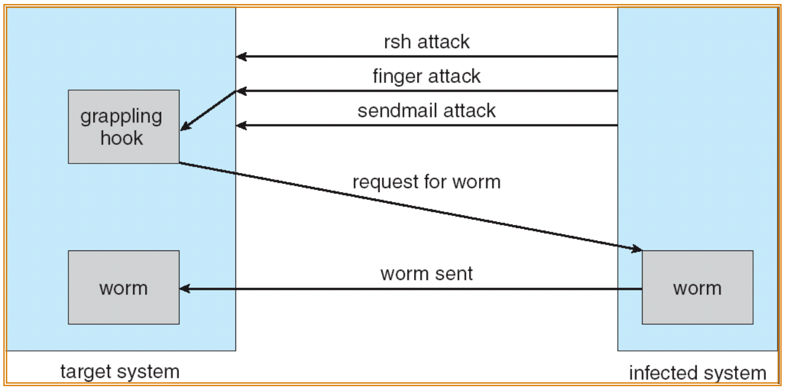
A Boot-sector Computer Virus


virus copies boot sector to unused location X
virus replaces original boot block with itself
at system boot, virus decreases physical memory, hides in memory above new limit
virus attaches to disk read-write interrupt, monitors all disk activity
whenever new removable R/W disk is installed, it infects that as well
it blocks any attempts of other programs to write the boot sector
it has a logic bomb to wreak havoc at a certain date

System and Network Threats

- Worms – use **spawn** mechanism; standalone program
- Internet worm *(Robert Morris, 1998, Cornell)*
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs
  - **Grappling hook** program uploaded main worm program
- Port scanning
  - Automated attempt to connect to a range of ports on one or a range of IP addresses
- Denial of Service
  - Overload the targeted computer preventing it from doing any useful work
  - Distributed denial-of-service (**DDOS**) come from multiple sites at once

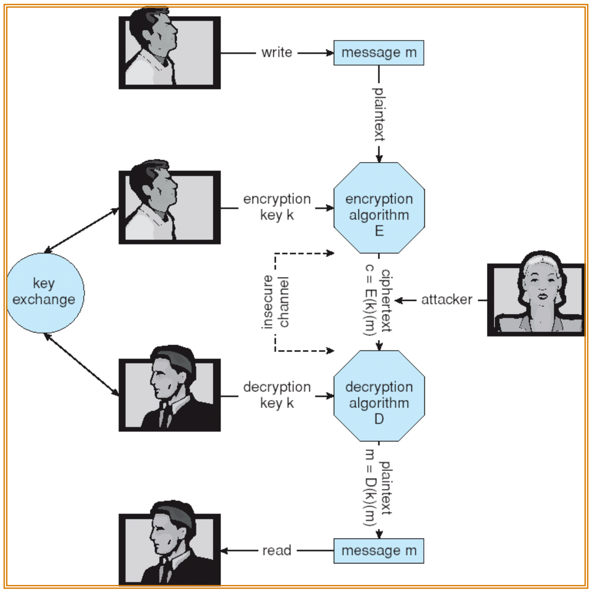
The Morris Internet Worm



Cryptography as a Security Tool

- Broadest security tool available
  - Source and destination of messages cannot be trusted without cryptography
  - Means to constrain potential senders (*sources*) and / or receivers (*destinations*) of *messages*
- Based on secrets (**keys**)

Secure Communication over Insecure Medium



Encryption

- Encryption algorithm consists of
  - Set of K keys
  - Set of M Messages
  - Set of C ciphertexts (encrypted messages)
  - A function $E : K \to (M \to C)$. That is, for each $k \in K$, $E(k)$ is a function for generating ciphertexts from messages.
  - A function $D : K \to (C \to M)$. That is, for each $k \in K$, $D(k)$ is a function for generating messages from ciphertexts.
- An encryption algorithm must provide this essential property: Given a ciphertext $c \in C$, a computer can compute m such that $E(k)(m) = c$ only if it possesses $D(k)$.
  - Thus, a computer holding $D(k)$ can decrypt ciphertexts to the plaintexts used to produce them, but a computer not holding $D(k)$ cannot decrypt ciphertexts.
  - Since ciphertexts are generally exposed (for example, sent on the network), it is important that it be infeasible to derive $D(k)$ from the ciphertexts

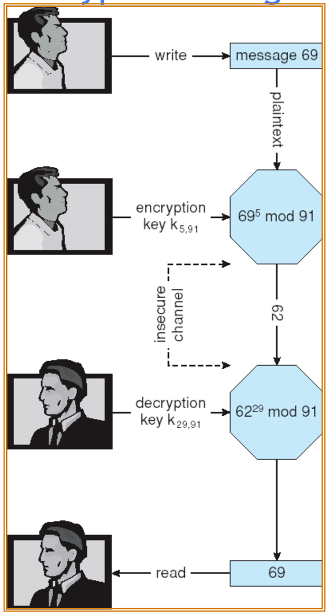
Symmetric Encryption

- Same key used to encrypt and decrypt
  - *E*(*k*) can be derived from *D*(*k*), and vice versa
- **DES** is most commonly used symmetric block-encryption algorithm (created by US Govt)
  - Encrypts a block of data at a time (64 bit messages, with 56 bit key)
- **Triple-DES** considered more secure (repeat DES three times with three different keys)
- Advanced Encryption Standard (**AES**) replaces DES
  - Key length upto 256 bits, working on 128 bit blocks
- **Twofish, RC4, RC5 ..** other symmetric algorithms
- **RC4** is most common symmetric stream cipher (works on bits, not blocks), but known to have vulnerabilities
  - Encrypts/decrypts a stream of bytes (i.e wireless transmission, web browsers)
  - Key is a input to psuedo-random-bit generator
    - Generates an infinite **keystream**

Asymmetric Encryption

- Encryption and decryption keys are different
- Public-key encryption based on each user having two keys:
  - public key – published key used to encrypt data
  - private key – key known only to individual user used to decrypt data
- Must be an encryption scheme that can be made public without making it easy to figure out the decryption scheme
  - Most common is RSA *(Rivest, Shamir, Adleman)* block cipher

Encryption and Decryption using RSA Asymmetric



write

message 69

plaintext

encryption key $k_{5,91}$

69^5 mod 91

insecure channel

62

decryption key $k_{29,91}$

62^{29} mod 91

read

69

Any Questions?

Reading Assignment

- Read chapter 14 and 15 from Silberschatz.

Acknowledgements

- “Operating Systems Concepts” book and supplementary material by Silberschatz, Galvin and Gagne.