

Security

CS 256/456

Dept. of Computer Science, University of Rochester

The Security Environment

Security goals:

- Authentication
- Data confidentiality
- Data integrity
- System availability

Threats of intruders or adversaries:

- Identity hijacking
- Exposing data
- Tampering with data
- Denial of service attacks

We focus on OS-related security issues.

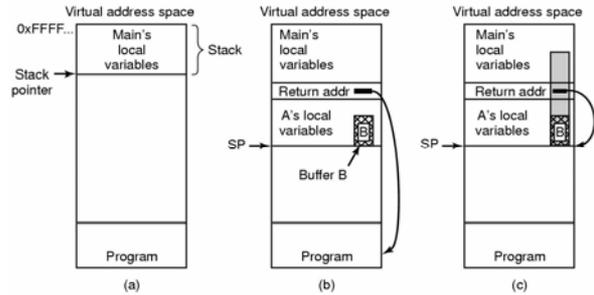
Login Spoofing

- Login spoofing
 - A program running by the attacker displays a login screen (like the real one)
 - After a legitimate user types in username and password, it records those, kills itself, and a real login screen is shown
 - The user thinks she typed in a wrong password and tries again, which works
- Countermeasure?
 - Start each login session with a non-user-catchable key combination "Ctrl-Alt-Delete"

User Authentication

- UNIX user passwords are mapped using a one-way function "e()"; and then stored in a globally readable file "/etc/passwd"
 - Bobbie, e(Dog)
 - Tony, e(6%%TaeFF)
 -
- Attack:
 - used a precomputed common password list
- Countermeasure?
 - salt
 - Bobbie, 4238, e(Dog4238)
 - Tony, 2918, e(6%%TaeFF2918)

Buffer Overflow



- (a) Situation when main program is running
 (b) After program A called
 (c) Buffer overflow shown in gray

Countermeasures:

- boundary checks, non-executable stack/data segment, ...

3/28/2007

CSC 256/456 - Spring 2007

5

Leaking Unnecessary Information

```
LOGIN: ken
PASSWORD: FooBar
SUCCESSFUL LOGIN
```

(a)

```
LOGIN: carol
INVALID LOGIN NAME
LOGIN:
```

(b)

```
LOGIN: carol
PASSWORD: Idunno
INVALID LOGIN
LOGIN:
```

(c)

User authentication:

- (a) A successful login
 (b) Login rejected after name entered
 (c) Login rejected after name and password typed

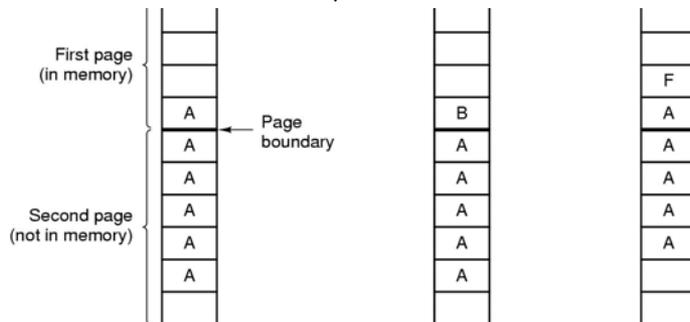
3/28/2007

CSC 256/456 - Spring 2007

6

The TENEX Password Problem

- Files are accessed with passwords. At each access, the password is checked byte-by-byte and an error is returned as soon as a byte is mismatched.



3/28/2007

CSC 256/456 - Spring 2007

7

Information Leaking Through Side Channel

- Side channels
 - performance observations
 - program execution signals (e.g., cache usage, memory bus usage)
- Side channel attack on hyper-threading processors [Percival 2005]
 - OpenSSH running DES encryption on one hyper-thread
 - attacker running on the other hyper-thread
 - attacker and OpenSSH share hardware cache, so attacker can monitor its own cache miss pattern to infer the execution of OpenSSH (and its DES encryption key)

3/28/2007

CSC 256/456 - Spring 2007

8

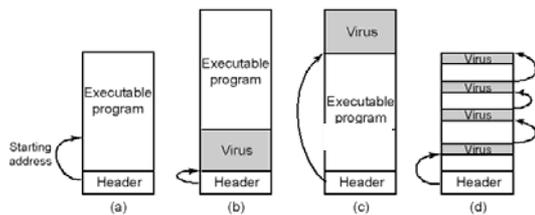
Denial-of-Service Attack

- Attacker attempts to consume all available resources at the host so no resources are left to serve legitimate users
 - attacks often come from network and they are distributed
- TCP flooding:
 - attackers establish many bogus TCP connections
 - host allocates buffer space for each connection
 - host memory being exhausted eventually
- Countermeasure?
 - **discard** flooded requests: throw out good and bad ones
 - **trace back** to source of floods
 - attack requests with spoofed identities
 - sources are most likely an innocent, compromised machines
 - **delayed processing/resource allocation**
 - stateless TCP [Shieh et al., NSDI2005]

Virus

- Virus
 - program can reproduce itself
 - e.g., when invoked, traverse the file system and attach it to randomly selected executables
 - additionally, do harm
 - steal your data
 - temporarily crash the system
 - permanently damage data or hardware
 - denial of service by using all available system resources
- "Good" virus
 - quickly spreading virus
 - difficult to detect
 - hard to get rid of

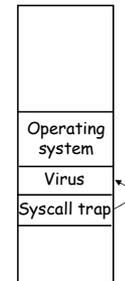
Infecting An Executable (Trojan Horses)



- (a) An executable program
- (b) With a virus at the front
- (c) With the virus at the end
- (d) With a virus spread over free space within program

Memory Resident Viruses

- Virus resides in memory; intercepting system calls
- Where in memory to put the virus?
 - known unused memory in OS kernel
 - make the OS believe the memory that virus uses is "legitimately used"
- How to load virus there in the first place?
 - boot sector viruses
 - device driver viruses



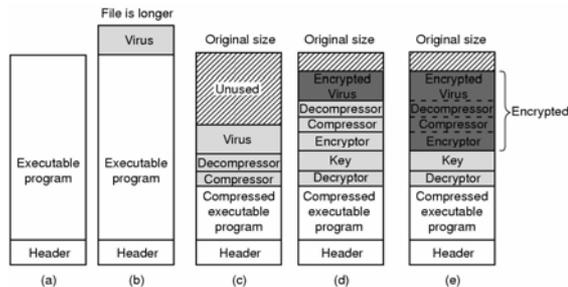
How Viruses Spread

- Try to infect programs on
 - networks: exploiting buffer overflow errors in network server daemons
 - floppy drives
- Attach to innocent looking email
 - when it runs, use mailing list to replicate

Antivirus Techniques

- Size checkers
 - keep a record on the size of disk files and scan them periodically for any size changes
 - apply on readonly executables.
- Signature scanning
 - maintain a database of patterns of common viruses
 - scan disk files for these patterns

Anti-Antivirus Techniques



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

More Antivirus Techniques

- Integrity checkers
 - similar to size checkers, but this time we compute a checksum for file and store them somewhere; we periodically check all files to see whether the checksum still matches
- Behavioral checkers (memory-resident anti-virus program)
 - intercept system calls and detect suspicious activities: overwriting executables, ...

Disclaimer

- Parts of the lecture slides contain original work of Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Andrew S. Tanenbaum, and Gary Nutt. The slides are intended for the sole purpose of instruction of operating systems at the University of Rochester. All copyrighted materials belong to their original owner(s).