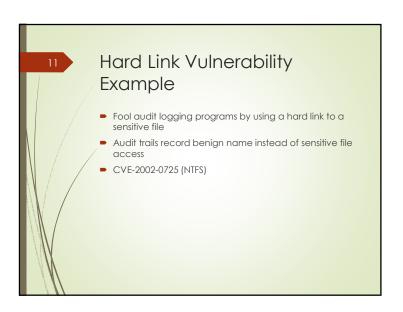
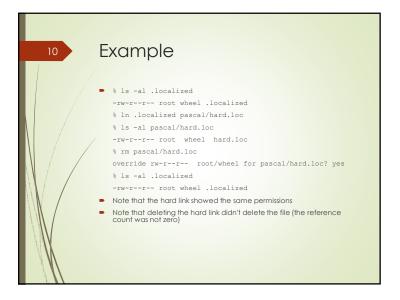
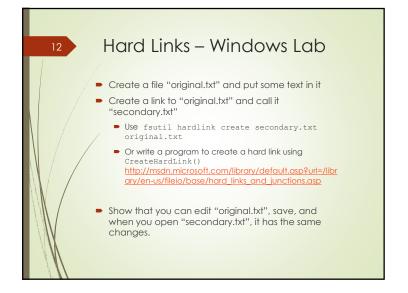
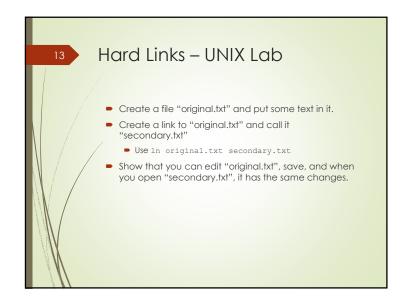


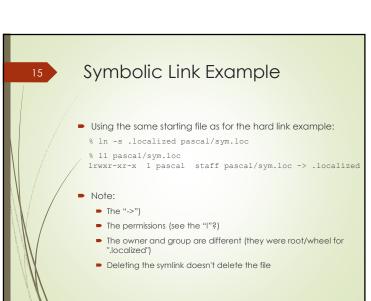
Hard Links Indistinguishable from original entry (peer) May not refer to directories or span file systems Created link is subject to the same, normal file access permissions. Deleting a hard link doesn't delete the file unless all references to the file have been deleted A reference is either a hard link or an open file descriptor In Windows, a hard link exists at the NTFS file system level and is not supported by FAT32 Note: Different from a Windows shortcut

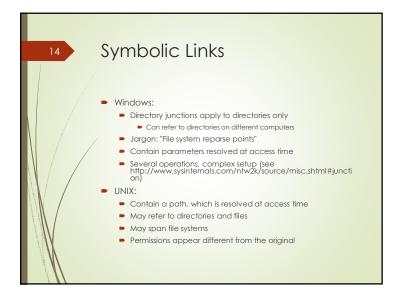


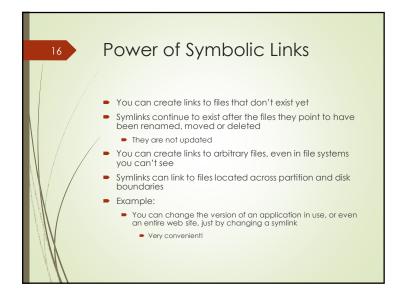




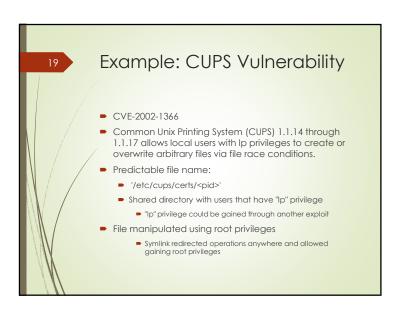


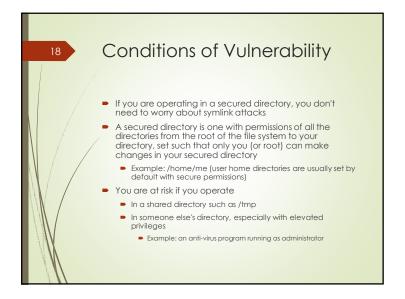






Pasic Attack Trick a process (with higher privileges) to operate on another file than the one it thinks it is. Example: Create the link "temp -> /etc/password" A privileged process executes truncate("temp".0) The "truncate" call follows symlinks changes the length of the file "temp" to 0 But truncated /etc/password instead! Note that the contents are deleted, not the file Can be used for write or read operations Or deletion if the symlink is in the path and not the end point







Best Defenses (Both Windows and UNIX)

If you are operating:
In someone else's directory, relinquish elevated privileges

If you are root (or administrator), set your effective user ID to that of the directory's owner for file operations in that directory

assuming that the directory is secured for that user; otherwise, you may endanger that user's files

If you are not root, you may be at risk of attacks against files you own elsewhere

don't operate on files in other user's directories

In a shared directory such as /tmp, consider using instead

A temporary directory inside your home directory

A secured directory for root or administrator temporary files

Processive deletion utility: "rd"

Scenario:

Attacker makes a link from c:\temp\tempdir to c:\windows\system32 or any other sensitive directory

Administrator does "rd /s c:\temp"

Sensitive directory is erased!

(Example from Howard and Leblanc 2003)

Note: Unlike the Windows "rd" utility, the UNIX command "rm -rf" does not traverse symlinked directories when deleting

Symlink File Write Example:
Netscape 6.01

The Netscape installer (running as root) creates a file in /tmp using a predictable name

Attack: create a number of symlinks to cover the probable names, all pointing to a file to be deleted

The installer will delete the target!

CVE-2001-1066

Which installers are you using that might have similar vulnerabilities?

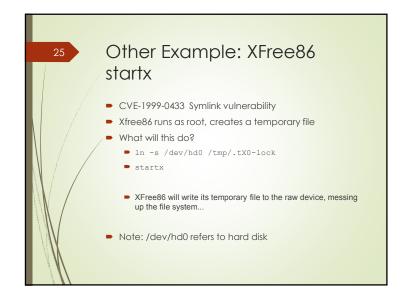
Destion

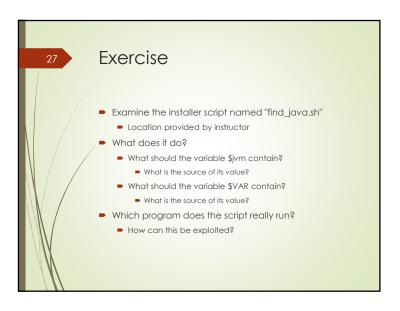
Let's say you have a setuid program "myprogram" that uses a temporary file, /tmp/mytemp

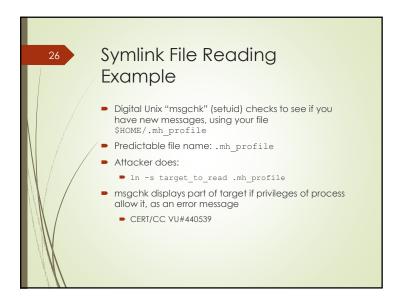
What happens when attacker does this?

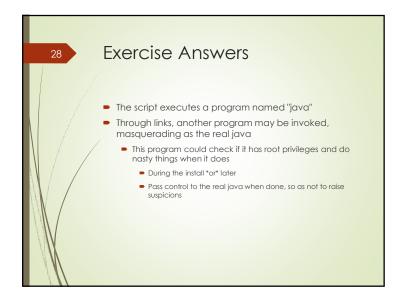
In -s /somedir/somefile /tmp/mytemp
/usr/bin/myprogram &

a) Nothing
b) There's a symlink pointing to "/somedir/somefile", and "myprogram" runs
c) "myprogram" may perform operations on "/somedir/somefile" instead of the intended temporary file









##