Windows Security Model

Erik Fredericks (fredericks@oakland.edu)

Overview

In-class dealie

General security model
- Refresher on Linux
- Overview of Windows
  - Slides from University of Kansas
    - https://people.eecs.ku.edu/~saiedian/710/Lectures/Readings/00-EECS710-Workshops/12-OS-security-workshop.ppt

In-Class (Powershell) Assignment

Split off into your groups
Let's work on a script for adding each of you to the ActiveDirectory infrastructure

Each team should be added to the following domain:

OU=students,DC=csi3670,DC=local

What you need to do is come up with **pseudocode to:**

1) **Read in user account information from a text file**
2) **Loop over each user**
3) **Add each user to ActiveDirectory**

This will be something you turn in, and if you are able to turn in working code, there will be extra credit.
The working script will be posted after the Moodle deadline

File format:

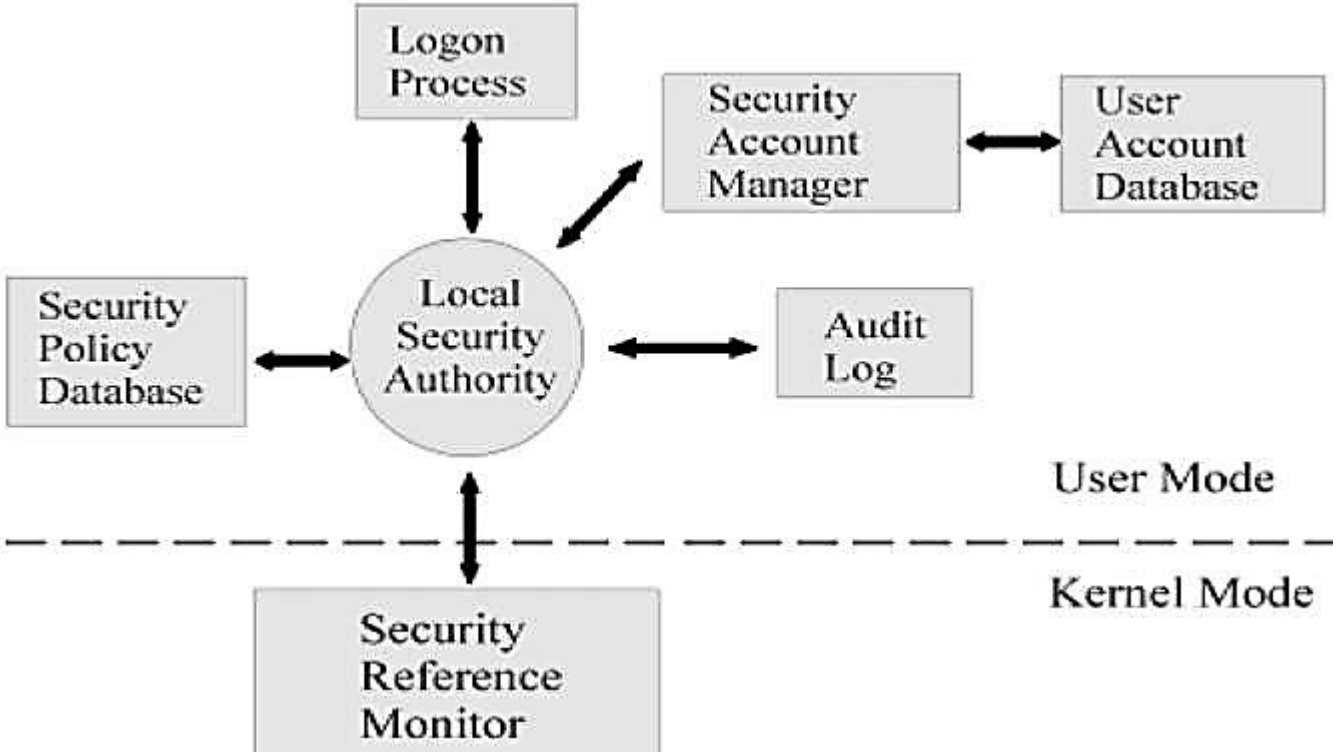
First Name, Last Name, Email
Erik, Fredericks, fredericks@oakl...

Windows Security Architecture

Security Reference Monitor

- Local Security Authority
- Security Account Manager
- Active Directory
- Local vs. Domain Accounts
- Access Control Lists
- Integrity Control
- User Account Controls

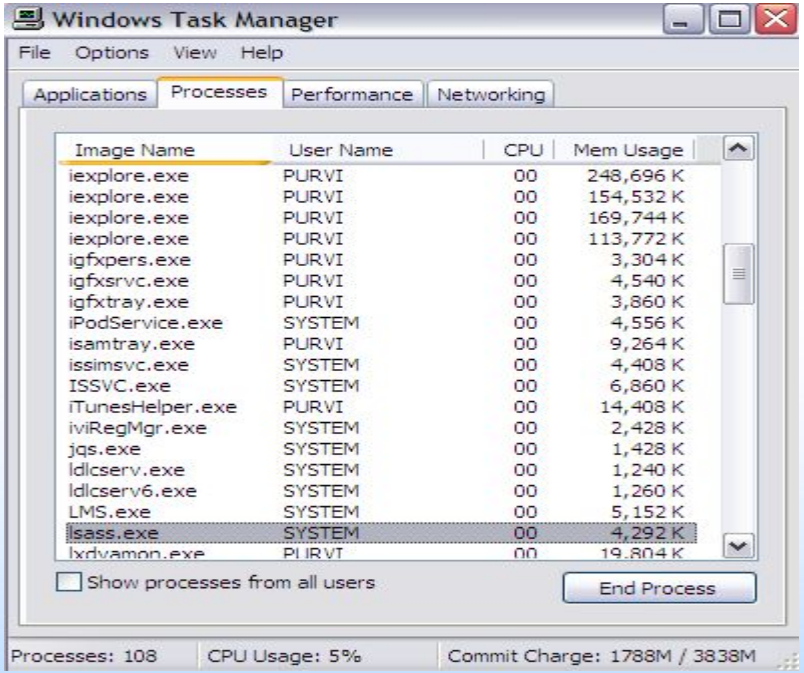
Architecture Overview



Security Reference Monitor (SRM)

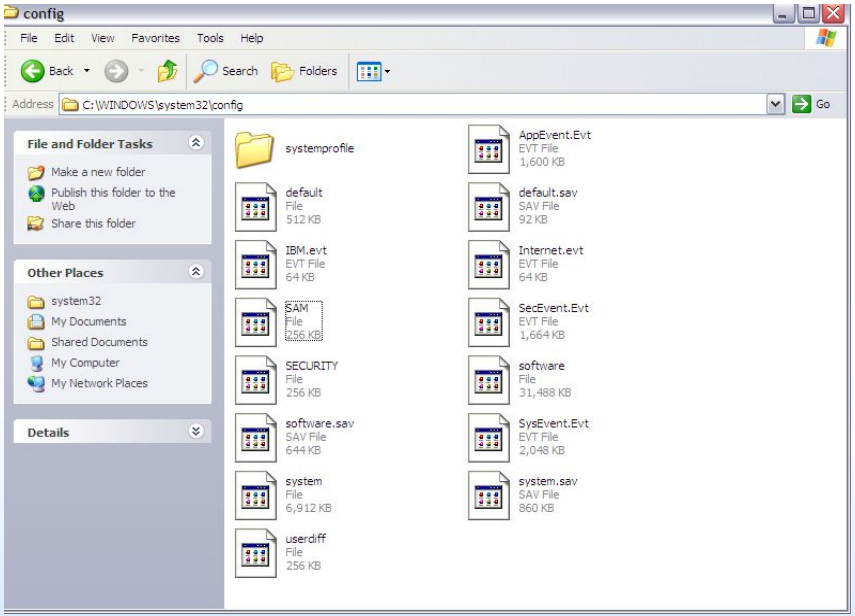
- Kernel Mode Component that
  - Performs Access Checks
  - Generates Audit Log Entries
  - Manipulates User Privileges

Local Security Authority (LSA)



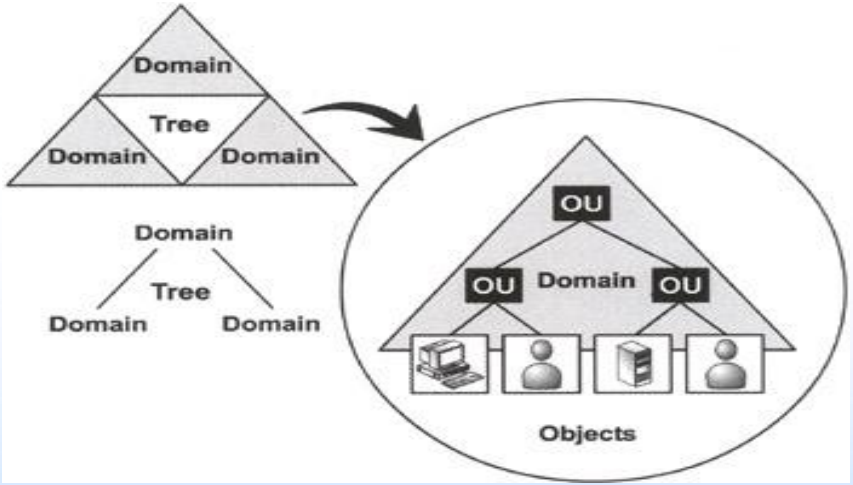
- Responsible for enforcing local security policy
  - Lsass.exe
  - User mode
- Issues security tokens to accounts
- Key component of the logon process

Security Account Manager (SAM)



- A database that stores user accounts and local users and groups security information
- SamSrv.exe

Active Directory



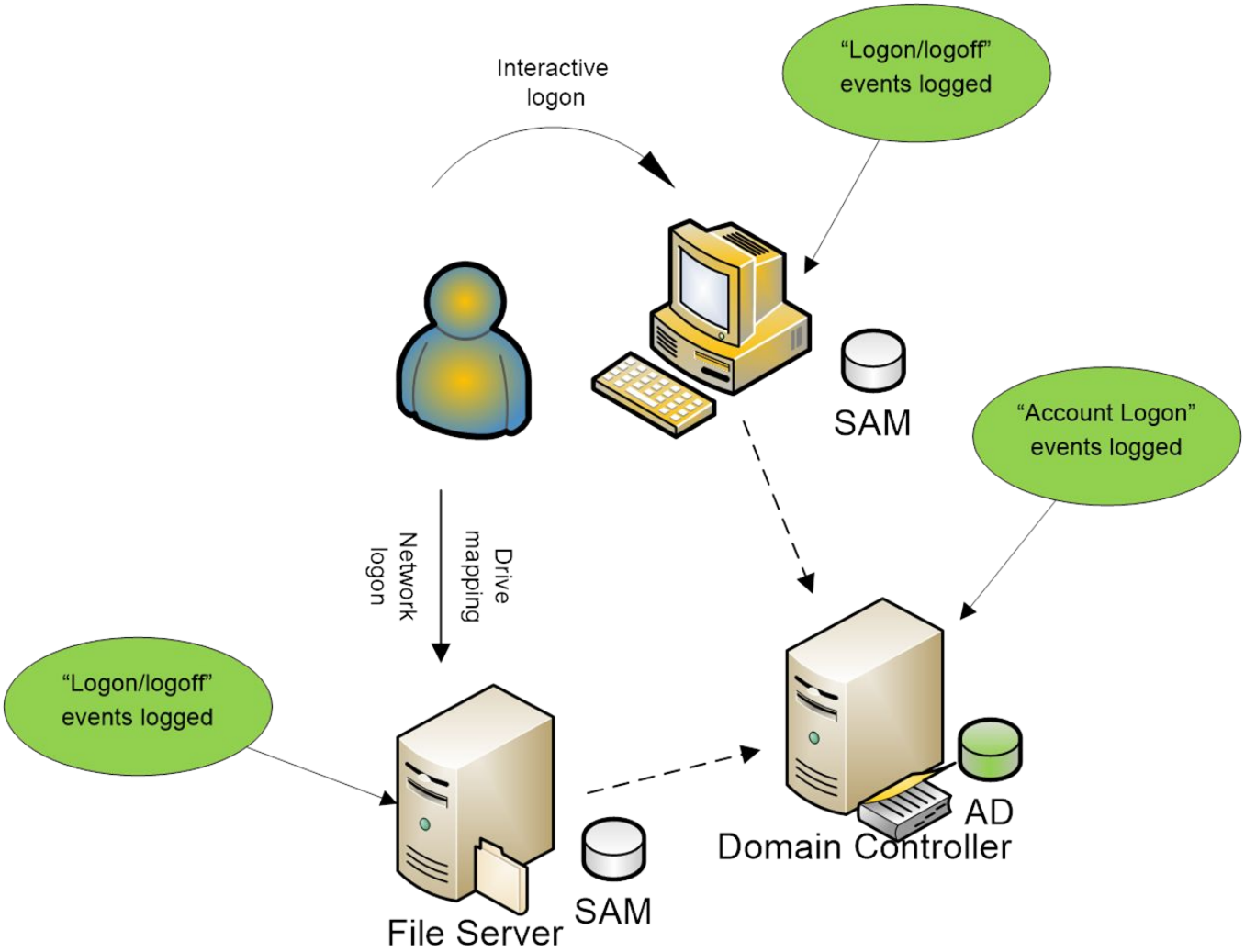
- Directory Service
  - Server-based authentication
  - Centrally managed

WinLogon & NetLogon

- WinLogon – keyboard requests
- NetLogon – network requests

Local vs. Domain Accounts

- Local Accounts for computers not hooked up to a network
- Networked computers can be:
  - Workgroup joined
  - Domain joined


Interactive
logon
"Logon/logoff"
events logged
SAM
"Account Logon"
events logged
Network
logon
Drive
mapping
"Logon/logoff"
events logged
AD
Domain Controller
SAM
File Server

Workgroup Joined

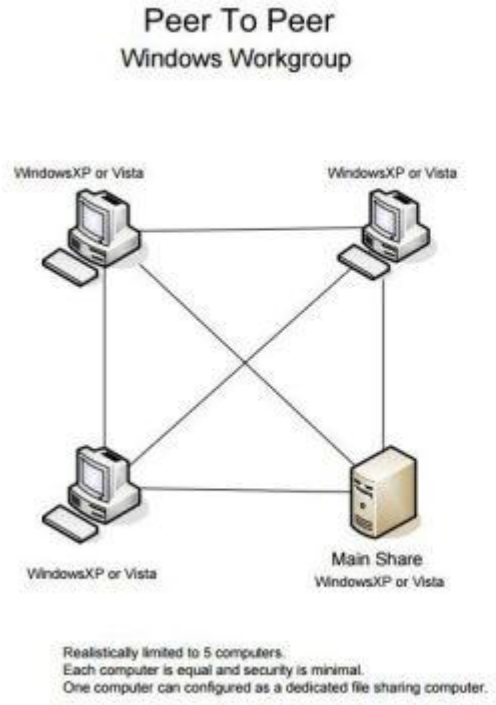
- A collection of computers connected together
- Only local accounts in SAM can be used
- No infrastructure to support AD

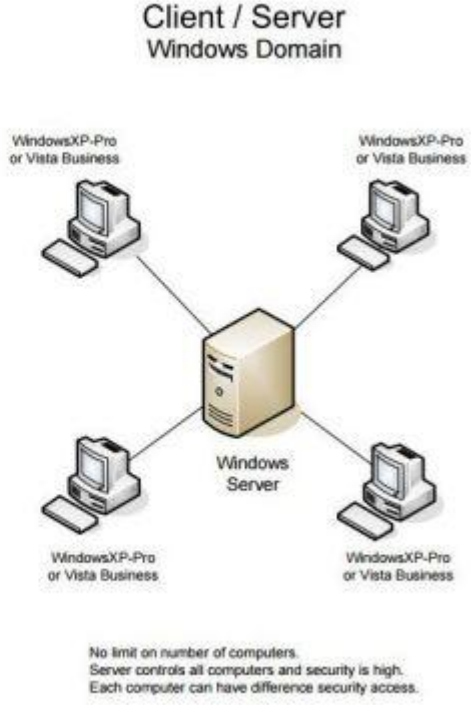
Domain Joined

- Share access to networked printers, file servers, etc.
- Centrally Managed
  - More secure
  - Scalable

Differences

P2P vs. Client/Server





Windows Login Example

- Administrator creates a user account (full name, username, password, group, privileges)
- Windows creates an SID in the form of
  - S-1-5-21-AAA-BBB-CCC-RRR
- In windows, username can be in two formats
  - **SAM format**: support by all versions of Windows (legacy format)
    - Form: DOMAIN\username
  - **User Principle Name** (UPN) and looks more like RFC822 email address
    - Example: username@domain.company.com

Windows Login Example

- User logs in with keyboard
- Information is sent to the AD (domain controller)
- If successful token is generated and sent to user
- Token contains
  - User's SID
  - Group membership
  - Privileges

Review Question

- A user hits Ctrl+Alt+Del and logs into Windows with a keyboard…
- What Windows process captures this login?

Answer

- The WinLogon process captures logins at the keyboard
- WinLogon passes information to the domain controller (Active Directory) to perform logon
- WinLogon would pass the information to the SAM (if local) which would give true/false authentication status
- LSA would generate token if SAM verifies true username/password combination

Windows Privileges

- System-wide permissions assigned to user accounts
- Some are considered “dangerous”
  - Act as part of the OS privilege
  - Debug programs privilege
  - Backup files and directories privilege

Access Control List (ACL)

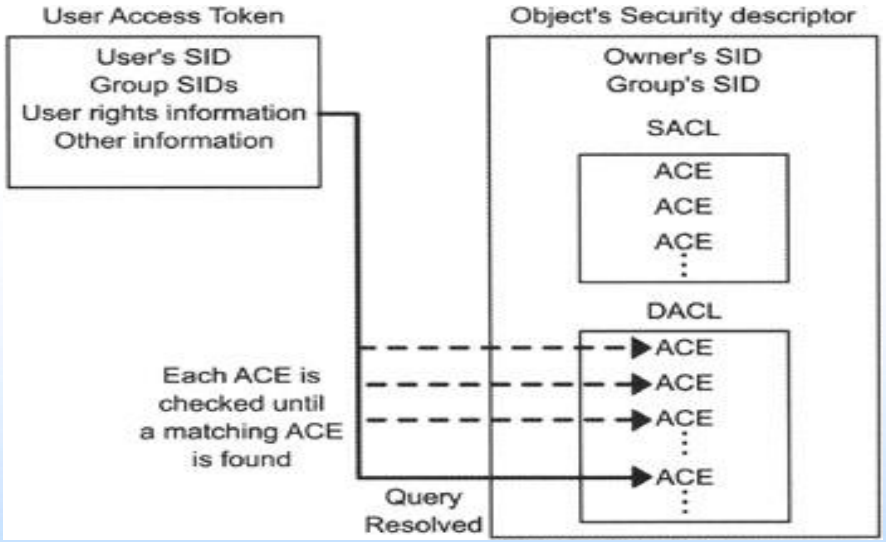
- Discretionary ACL
  - Grants or denies access to protected resources such as files, shared memory, etc.
- System ACL
  - Used for auditing and to enforce mandatory integrity policy (Vista)

Access Control Lists (ACL) (continued)

- Objects needing protection are assigned an ACL that includes
  - SID of object owner
  - List of access control entries (ACEs)
- Each ACE includes a SID and Access Mask
  - Access mask could include
    - Read, Write, Create, Delete, Modify, etc.

Access Control Example

- User opens text file



Integrity Control

- New to Vista: a low-level change to Windows that isolates different objects on a trust-based scale
- Controlled by a new OS component called Windows Integrity Control (WIC)
- Integrity levels trounce permissions
  - Example: malware no longer runs in the privilege level of the logged-on user, as it does in XP
  - It runs in the integrity level of the object that spawned it
- Makes process isolation and other Vista security measures possible

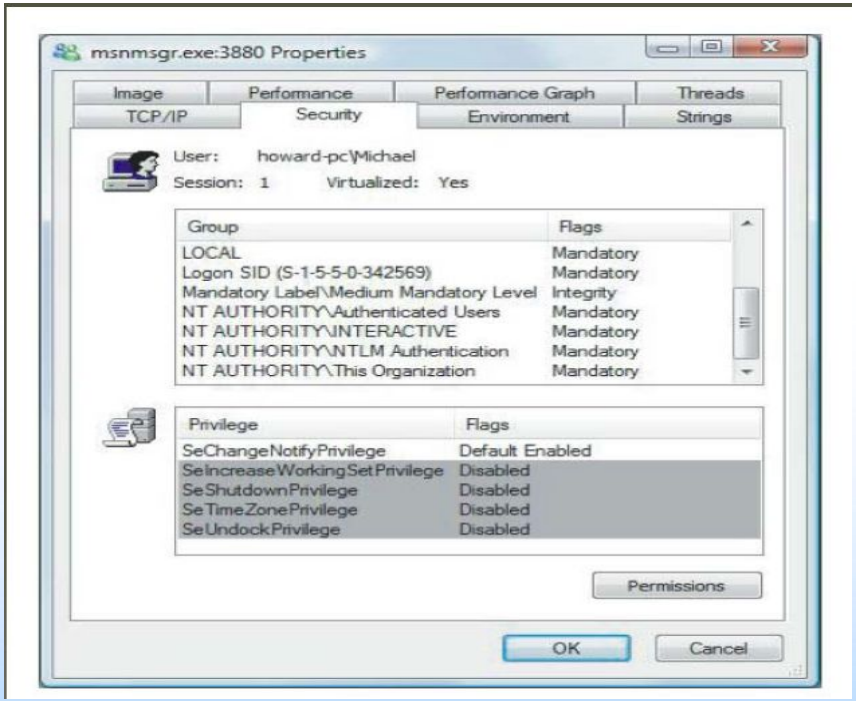
Six Integrity Levels

Object and Principals are labeled

- Untrusted
- Low
- Medium
- High
- System
- Installer

MAC: Vista Integrity Control

- Having Integrity Control in Windows Vista
  - Limits operations changing an object's state



User Account Controls

- A new feature in Windows Vista designed to help prevent unauthorized changes to your computer
- UAC is similar to security features in UNIX-like operating systems
- Perhaps the most reviled and misunderstood feature ever added to Windows

User Account Controls (continued)

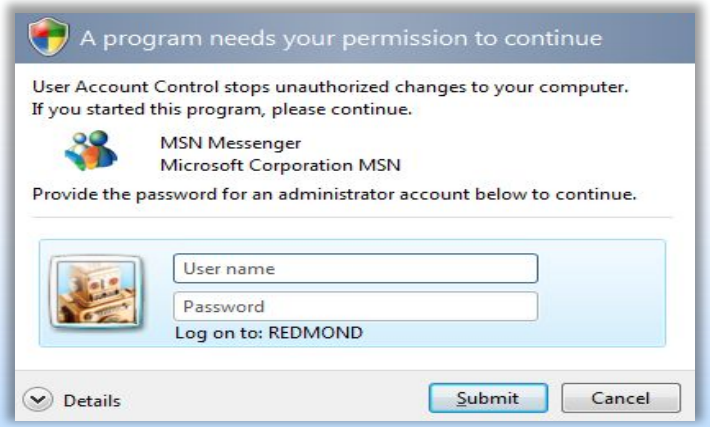
- How it works: When your consent is required to complete a task, UAC will prompt you with a dialog box
- Tasks that will trigger a UAC prompt include anything that will affect the integrity or security of the underlying system
  - This is a surprisingly long list of tasks
- UAC works slightly differently with standard user and administrator-class accounts

UAC Consent UI: Type 1

- **Prompt:** Windows needs your permission to continue
- **Why you see this:** You attempt to change a potentially dangerous system setting, such as a running a Control Panel

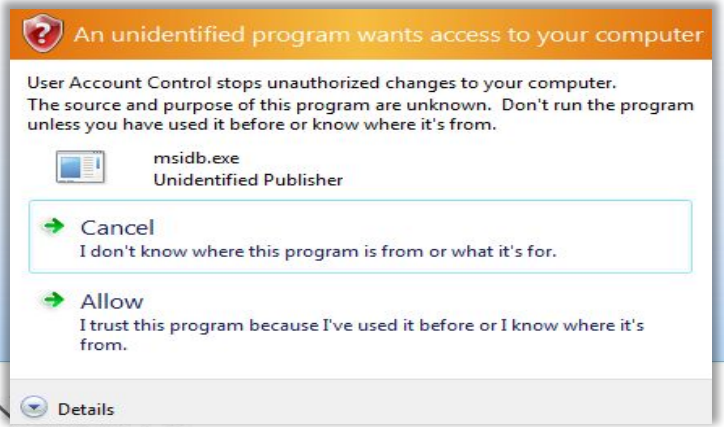
UAC Consent UI: Type 2

- **Prompt:** A program needs your permission to continue
- **Why you see this:** An external application with a valid digital signature is attempting to run with admin privileges



UAC Consent UI: Type 3

- **Prompt:** An unidentified program wants access to your computer
- **Why you see this:** in external application without a valid digital signature is trying to run



UAC: What's really happening

- Administrator accounts now logon with a mixed token
- Half of this mixed token is a standard user token: this is what is typically used to determine your memberships and privileges
- The other half, the administrator token, is invoked only when required: you can do so manually (run as) or automatically (certain tasks in Vista are tagged as requiring an admin token)

Mandatory Access Controls

- Linux uses a DAC security model
- Mandatory Access Controls (MAC) imposes a global security policy on all users
  - Users may not set controls weaker than policy
  - Normal admin done with accounts without authority to change the global security policy
  - But MAC systems have been hard to manage
- Novell's SuSE Linux has AppArmor
- RedHat Enterprise Linux has SELinux
- "pure" SELinux for high-sensitivity, high-security

Evaluation: Windows vs. Linux Design

- It is possible that email and browser-based viruses, trojans and worms are the source of the myth that Windows is attacked more often than Linux
- Do the attacks so often succeed on Windows because the attacks are so numerous, or because there are inherent design flaws and poor design decisions in Windows?
  - Many, if not most of the viruses, trojans, worms and other malware infect Windows machines through vulnerabilities in Microsoft Outlook and Internet Explorer
- In the most cases where Linux system are compromised
  - The primary cause was inadequately configured security settings

Windows Design Flaws/Poor Design Decisions

- Windows has evolved from a single-user design to a multi-user model few years back
- Windows is monolithic, not modular, by design
- Windows depends too heavily on an RPC model
- Windows focuses on its familiar graphical desktop interface

Evolved from Single-User Design to a multi-user model few years back

- Windows has long been hampered by its origin as a single-user system
  - Windows was originally designed to allow both users and applications free access to the entire system, which means anyone could tamper with a critical system program or file
- Windows XP was the first version of Windows to reflect a serious effort to isolate users from the system, so that users each have their own private files and limited system privileges

Monolithic by Design, not Modular

- Monolithic Design: one where most features are integrated into a single unit
  - Microsoft successfully makes competing products irrelevant by integrating more and more of the services they provide into its operating system
  - But this approach creates a monster of inextricably interdependent services
- Interdependencies side effects:
  - Every flaw in a piece of that system is exposed through all of the services and applications that depend on that piece of the system
  - Unstable by nature: when you design a system that has too many interdependencies, you introduce numerous risks when you change one piece of the system
-


What is my purpose?
You install Chrome
Oh, my god

Depends Heavily on an RPC Model

- RPC stands for Remote Procedure Call
- Simply put, an RPC is what happens when one program sends a message over a network to tell another program to do something
- RPCs are potential security risks because they are designed to let other computers somewhere on a network to tell your computer what to do

Focuses on its Familiar Graphical Desktop Interface

- Microsoft considers its familiar Windows interface as the number one benefit for using Windows Server 2003
  - Quote from the Microsoft web site, *"With its familiar Windows interface, Windows Server 2003 is easy to use. New streamlined wizards simplify the setup of specific server roles and routine server management tasks..."*
- By advocating this type of usage, Microsoft invites administrators to work with Windows Server 2003 at the server itself, logged in with Administrator privileges

Linux Design Flaws/Poor Design Decisions

- Linux is based on a long history of well fleshed-out multi-user design
- Linux is mostly modular by design
- Linux does not depend upon RPC to function, and services are usually configured not to use RPC by default
- Linux servers are ideal for headless non-local administration

Based on Multi-User Design

- Linux does not have a history of being a single-user system
  - Designed from the ground-up to isolate users from applications, files and directories that affect the entire operating system
  -
- Each user is given a user directory where all of the user's data files and configuration files are stored
  - When a user runs an application, such as a word processor, that word processor runs with the restricted privileges of the user

Modular by Design, not Monolithic

- Linux is for the most part a modularly designed operating system
  - From the kernel (the core “brains” of Linux) to the applications
- Not everything in Linux is modular
  - The two most popular graphical desktops: KDE and GNOME, are somewhat monolithic by design

Not Constrained by an RPC Model

- Most Linux distributions install programs with network access turned off by default
- Even when Linux applications use the network by default, they are most often configured to respond only to the local machine and ignore any requests from other machines on the network
- Unlike Windows Server 2003, you can disable virtually all network-related RPC services on a Linux machine and still have a perfectly functional desktop


Dawn of
The Second Day
-48 Hours Remain-

Windows Vulnerabilities

- Windows like all other OS has security bugs
  - Bugs have been exploited to compromise customer accounts
- Multiple versions of Windows
  - Each with substantial user-base
- Attackers are now (organized) criminals highly motivated by money
- Microsoft Security Bulletin Summaries and Webcasts provides latest vulnerabilities list and relative security updates (and status)

Windows Vulnerabilities Example

- Vulnerabilities in Windows Kernel Could Allow Elevation of Privilege Microsoft Security Bulletin MS10-021, April 2010)
  - Most severe of these vulnerabilities could allow elevation of privilege if an attacker logged on locally and ran a specially crafted application
  - An attacker must have valid logon credentials and be able to log on locally to exploit this vulnerability
    - The vulnerability could not be exploited remotely or by anonymous users

Windows Vulnerabilities Example

- Vulnerabilities in Windows Kernel Could Allow Elevation of Privilege Microsoft Security Bulletin MS10-021, April 2010) (continued)
  - Security update resolves several privately reported vulnerabilities in Microsoft Windows
    - Rated Important for all supported editions of Microsoft Windows 2000, Windows XP, Windows Server 2003, and the original release version of Windows Vista
    - Rated Moderate for all supported versions of Windows Vista Service Pack 1 and Windows Vista Service Pack 2, Windows Server 2008, Windows 7, and Windows Server 2008 R2
    - Most likely result in a denial of service condition

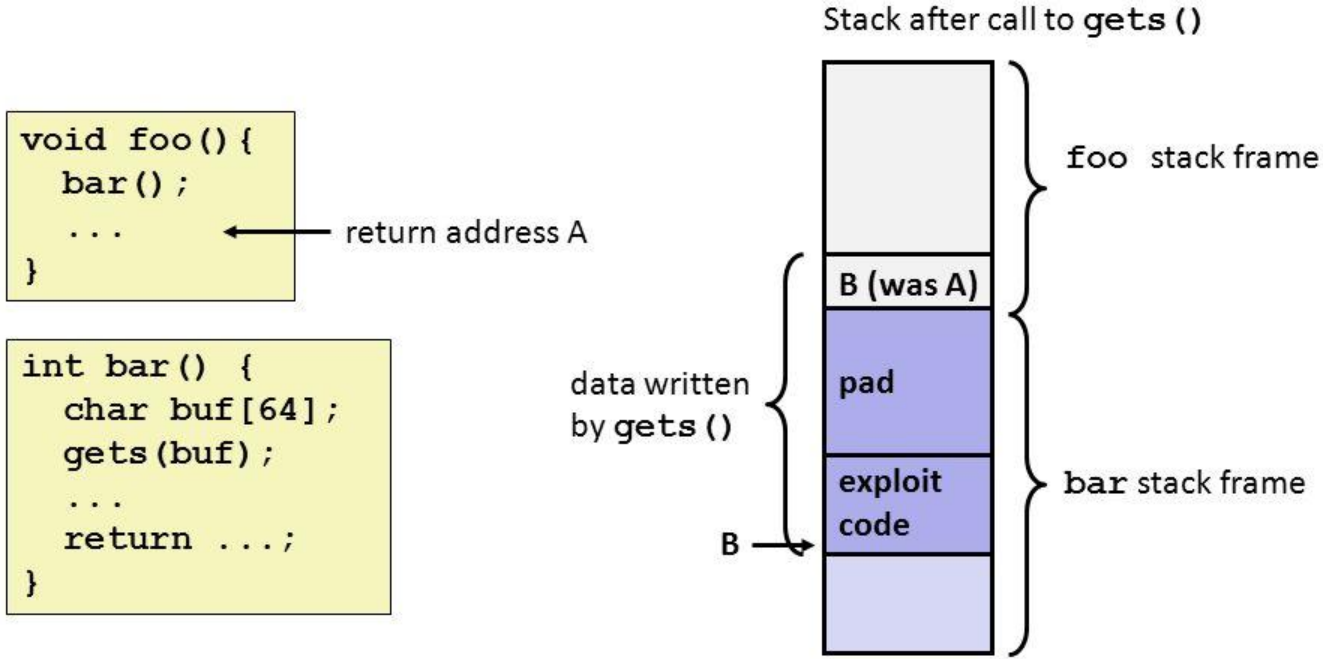
Linux Vulnerabilities

- Default Linux installations (un-patched and unsecured) have been vulnerable to
  - Buffer overflows
  - Race conditions
  - Abuse of programs run “SetUID root”
  - Denial of Service (DoS)
  - Web application vulnerabilities
  - Rootkit attacks

Buffer Overflow

Remember this?

Malicious Use of Buffer Overflow



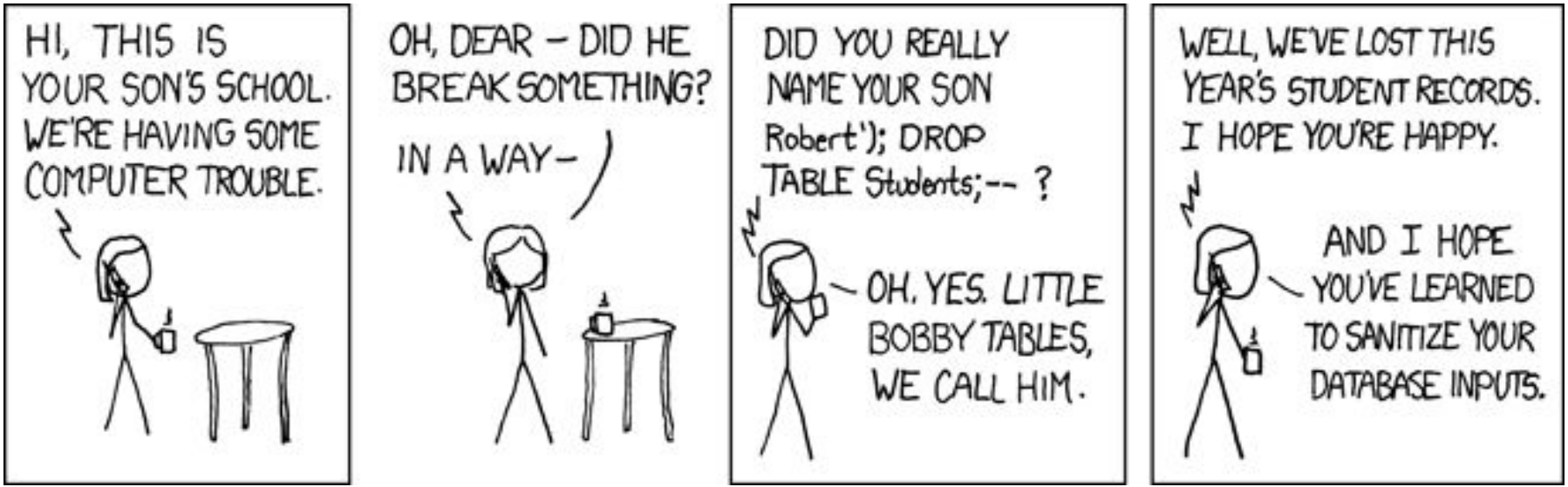
- Input string contains byte representation of executable code
- Stack frame must be big enough to hold exploit code
- Overwrite return address with address of buffer (need to know B)
- When `bar()` executes `ret`, will jump to exploit code (instead of A)

SetUID Root Vulnerabilities

- SetUID root program is a root-owned program
  - Runs as root no matter who executes it
- Unprivileged users can gain access to unauthorized privileged resources
- Must be very carefully programmed
- SetUID root programs necessary
  - Example: to change password
- Distributions now do not ship with unnecessary SetUID root programs
- System attackers still scan for them

Web Application Vulnerabilities

- Very broad category of vulnerabilities
- When written in scripting languages
  - Not as prone to classic buffer overflows
  - Can suffer from poor input-handling, XSS, SQL code injection etc.
- Linux distributions ship with few “enabled-by-default” web applications
  - Example: default CGI scripts included with Apache Web server


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Rootkit Attacks

- If successfully installed before detection, it is very difficult to find and remove
- Originally began as collections of hacked commands
  - Hiding attacker's files, directories, processes
- Now use loadable kernel modules (LKMs)
  - Intercepts system calls in kernel-space
  - Hides attacker from user
- Even LKMs not completely invisible
  - May be able to detect with chkrootkit
  - Generally have to wipe and rebuild system

Evaluation: Windows Vs. Linux Vulnerabilities

- The United States Computer Emergency Readiness Team (CERT) uses its own set of metrics to evaluate the severity of any given security flaw
- https://www.kb.cert.org/vuls/
  - View by CVSS score

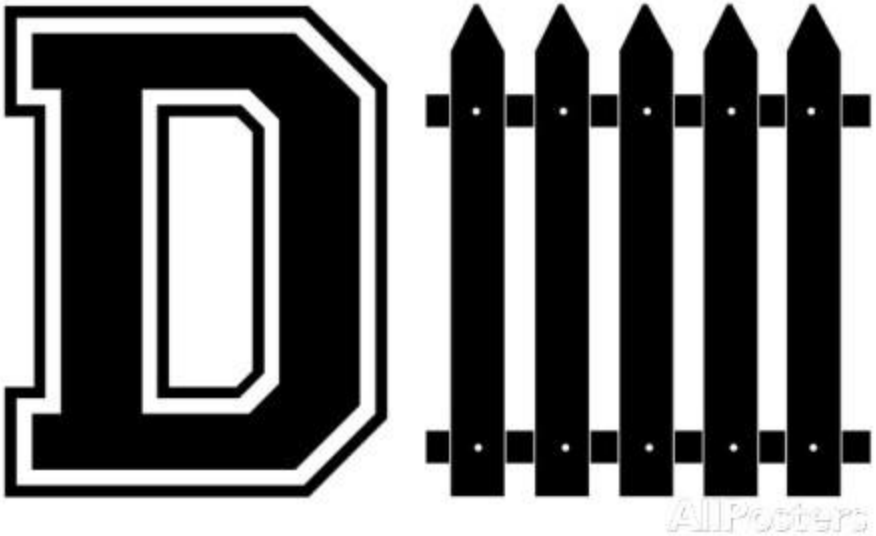
584653
113765

Table 14: Qualitative severity rating scale

Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

System Hardening

- Shoring up defenses, reducing exposed functionality, disabling less-used features
  - Called attack surface reduction
  - 80/20 rule of functionality
  - Not always achievable
    - Strip mobile code support on servers
- Servers easier to harden
  - Used for specific and controlled purposes
  - Administrative users with better skills than workstation users


D
AllPosters

Windows Defenses

- Microsoft Security Development Lifecycle
  - Net effect approx. 50% reduction in security bugs
  - Vista used SDL start to finish
- Categorize Security Defenses
  - Account defenses
  - Network defenses
  - Buffer over-run defenses
  - Browser defenses

Account Defenses

- Least Privilege
  - Operate with just enough privileges for task
- Another defense is to strip privileges from an account soon after an application start
- Windows Vista reserves default with UAC
  - Users prompted to perform privileged operations

Network Defenses

- Need more than account/user defenses
- Vulnerable to network attacks
- IPSec and IPv6 with authentication packets available in Vista
- Built-in software firewall
  - Block inbound connection of specific ports
  - Block outbound connections

Browser Defenses

- Browser is key point of attack
  - Via script code, graphics, helper objects, add-ons, cookies
- Added defenses in IE7
  - ActiveX disabled by default
  - Protected mode

Cryptographic Services

- Encrypting File System (EFS)
  - Files and directories encrypted/decrypted transparently
  - Generates random key, protected by DPAPI
- Bitlocker Drive Encryption
  - Encrypts entire volume with AES
  - Key either USB or TPM 1.2 compatible chip
- Data Protection API (DPAPI)
  - Manages encryption key maintenance
  - Keys derived from user's password

DPAPI

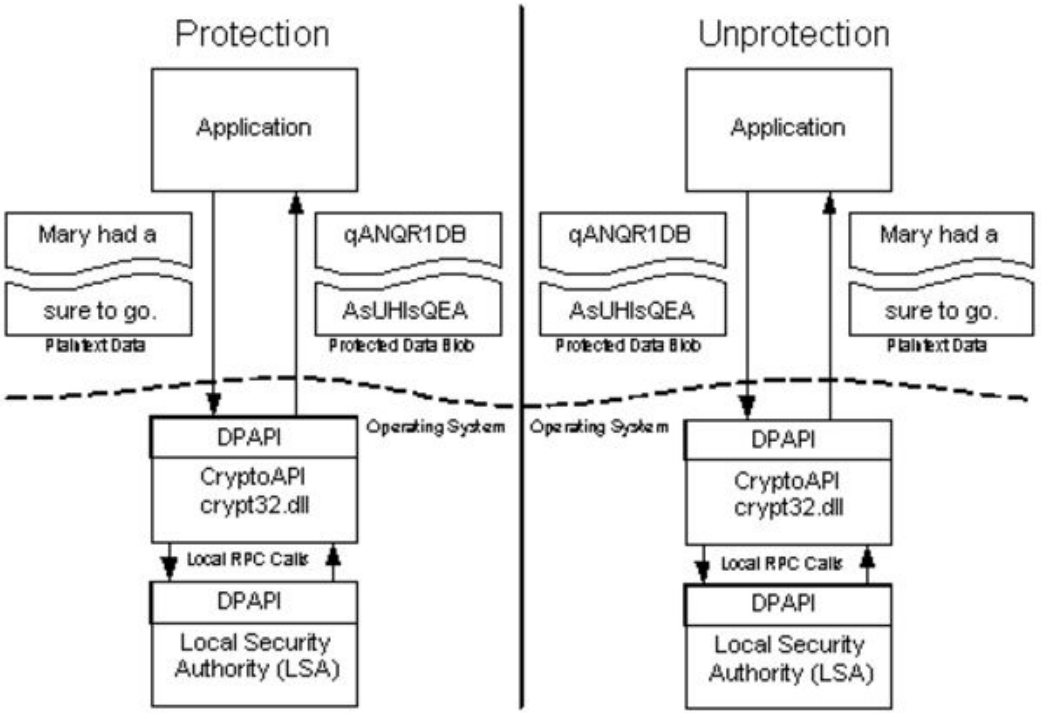


Figure 1. DPAPI is a simple API

https://msdn.microsoft.com/en-us/library/ms995355.aspx

Relevant: https://www.youtube.com/watch?v=Ow7M3nFN1ho

Also relevant: https://www.youtube.com/watch?v=ss47ffaqA_o

And one more: https://www.youtube.com/watch?v=-rQPdWwv3k8

Linux System Hardening

- Can be done at system and application levels
- Generalized steps to Linux System Hardening
  - Preliminary Planning
  - Physical System Security
  - Operating System Installation
  - Securing Local File Systems
  - Configuring and Disabling Services
  - Securing the root account
  - User Authentication and User Account Attributes
  - Securing Remote Authentication
  - Setup Ongoing System Monitoring
  - Backups

Just kidding one more

https://www.youtube.com/watch?v=j2wazCmNYaw

OS-Level Security Tools and Techniques

- OS Installation: Software Selection and Initial Setup
- Patch Management
- Network-Level Access Controls
- Using iptables for “Local Firewall” Rules
- Antivirus Software
- User Management
- Password ageing
- Root Delegation
- Logging

OS Installation

- Security begins with O/S installation
- What software is run
  - Unused applications liable to be left in default, un-hardened and un-patched state
- Generally should not run:
  - SMTP relay, X Window system, RPC services, inetd, SMTP daemons, telnet etc
- Setting some initial system s/w configuration:
  - Setting root password
  - Creating a non-root user account
  - Setting an overall system security level
  - Enabling a simple host-based firewall policy
  - Enabling SELinux

Patch Management

- Installed server applications must be:
  - Configured securely
  - Kept up to date with security patches
- Patching can never win “patch rat-race”
- Have tools to automatically download and Install security updates
  - Example: up2date, YaST, apt-get
  - Should not run automatic updates on change-controlled systems without testing

Network Access Controls

- Network a key attack vector to secure
- Libwrappers & TCP wrappers a key tool to check access
  - Before allowing connection to service, tcpd first evaluate access control
    - Defined in /etc/hosts.allow
    - Defined in /etc/hosts.deny

Using iptables for “Local Firewall” Rules

- Also have the very powerful **netfilter** Linux kernel native firewall mechanism and **iptables** user-space front end
- Useful on firewalls, servers, desktop
- Typically for “personal” firewall use will:
  - Allow incoming requests to specified services
  - Block all other inbound service requests
  - Allow all outbound (locally-originating) requests
- Do have automated rule generators
- If need greater security, manually configuration required

Antivirus Software

- Historically Linux not as vulnerable to viruses
  - Windows targeted more due to popularity
- Prompt patching of security holes more effective for worms
- Viruses abuse user's privileges
- Non-privileged user account
  - Less scope of being exploited
- Growing Linux popularity means growing exploits

User Management

- Guiding principles in user-account security:
  - Be careful setting file / directory permissions
  - Use groups to differentiate between roles
  - Use extreme care in granting / using root privileges

Password Aging

- Maximum and minimum lifetime for user passwords
  - Globally changed in /etc/login.defs
  - To change password settings for existing users
    - command line -> change

Root Delegation

- "su" command allows users to run as root
  - Use su with –c flag to allow you to run a command instead of an entire shell as root
  - Must supply root password
  - Drawback: many people will know root password
- SELinux RBAC can limit root authority but it's complex
- "sudo" allows users to run as root
  - But only need users password, not root password

Logging

- Linux logs using syslogd or Syslog-NG
  - Writes log messages to local/remote log files
- Syslog-NG preferable because it has:
  - Variety of log-data sources / destinations
  - Much more flexible “rules engine” to configure
  - Can log via TCP which can be encrypted
- Logrotate

Application Security (Hardening)

- A large topic
- Many security features are implemented in
- Similar ways across different applications
- Sub-topics
  - Running as unprivileged user/group
  - Running in chroot jail
  - Modularity
  - Encryption
  - Logging

Running As Unprivileged User/Group

- Every process “runs as” some user
- Extremely important user is not root
  - Since any bug can compromise entire system
- May need root privileges, e.g. bind port
  - Have root parent perform privileged function
  - But main service from unprivileged child
- User/group used should be dedicated
  - Easier to identify source of log messages

Running in “chroot” Jail

- “chroot” confines a process to a subset of /
  - Maps a virtual “/” to some other directory
  - Directories outside the chroot jail aren’t visible or reachable at all
  - Contains effects of compromised daemon
- Complex to configure and troubleshoot

Modularity

- Applications running as a single, large, multipurpose process can be:
  - More difficult to run as an unprivileged user
  - Harder to locate / fix security bugs in source
  - Harder to disable unnecessary functionality
- Hence modularity a highly prized feature
  - Providing a much smaller attack surface
- cf. postfix vs sendmail, Apache modules

Encryption

- Sending logins & passwords or application data over networks in clear text exposes them to various network eavesdropping attacks
- Hence many network applications now support encryption to protect such data
  - SSL and TLS protocols in OpenSSL library used
- May need own X.509 certificates to use
  - Can generate/sign using openssl command
  - May use commercial/own/free CA

Logging

- Applications can usually be configured to log to any level of detail (debug to none)
- Centralized logging using (syslog) can be used for consistency
- Must ensure there is some form of logging management as discussed before like rotating

References

- Stallings, W., Brown, L., *Computer Security: Principles and Practice*, Prentice Hall, NJ,2008
- Toxen, B., *Real World Linux Security*, Prentice Hall, NJ, 2002
- Howard, M. LeBlanc, D., *Writing Secure Code for Windows Vista*, Microsoft Press, WA,2006
- Ahmad, D., The Contemporary Software Security Landscape, *IEEE Security and Privacy*, vol. 5, no. 3, 2007, pp. 75-77
- Xinyue, S., Stinson, M., Lee, R.,Albee, P., An Approach to Analyzing the Windows and Linux Security Models, *IEEE Conference on Computer and Information Science (2006)*, July, pp. 56-62
- Xinyue, S., Stinson, M., Lee, R., Albee, P., A Qualitative Analysis of Privilege Escalation, *IEEE International Conference Proceedings on Information Reuse and Integration (2006)*, pp. 363-368

References (continued)

- Unix System Hardening Checklist, Accessed Dec 8,2008, http://www.linux-mag.com/downloads/2002-10/guru/harden_list.htm
- http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8
- http://www.kb.cert.org/vuls/bymetric?searchview&query=microsoft&searchorder=4&count=40
- http://www.kb.cert.org/vuls/bymetric?searchview&query=linux&searchorder=4&count=40
- http://www.microsoft.com/technet/security/bulletin/ms10-021.mspx
- http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx
- http://people.eecs.ku.edu/~saiedian/Teaching/Fa10/710/Readings/linux-security.pdf