Computer Security: Principles and Practice

Chapter 23 - Linux Security
Chapter 24 - Windows and Windows Vista Security

Windows vs. Linux Security

EECS 710

Professor: Hossein Saiedian

Presented by Purvi Patel

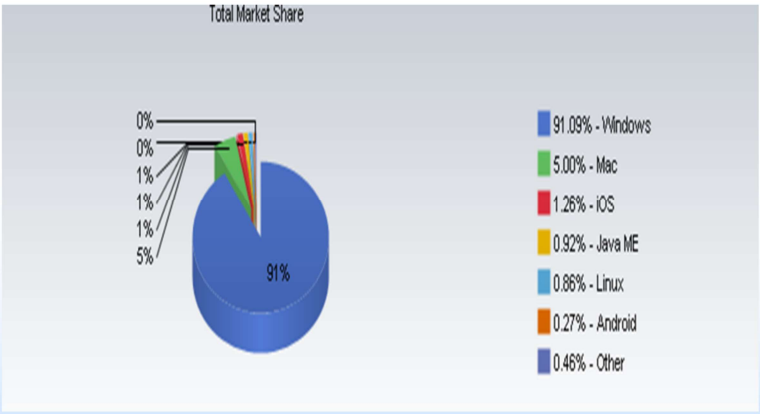
KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

1

Agenda

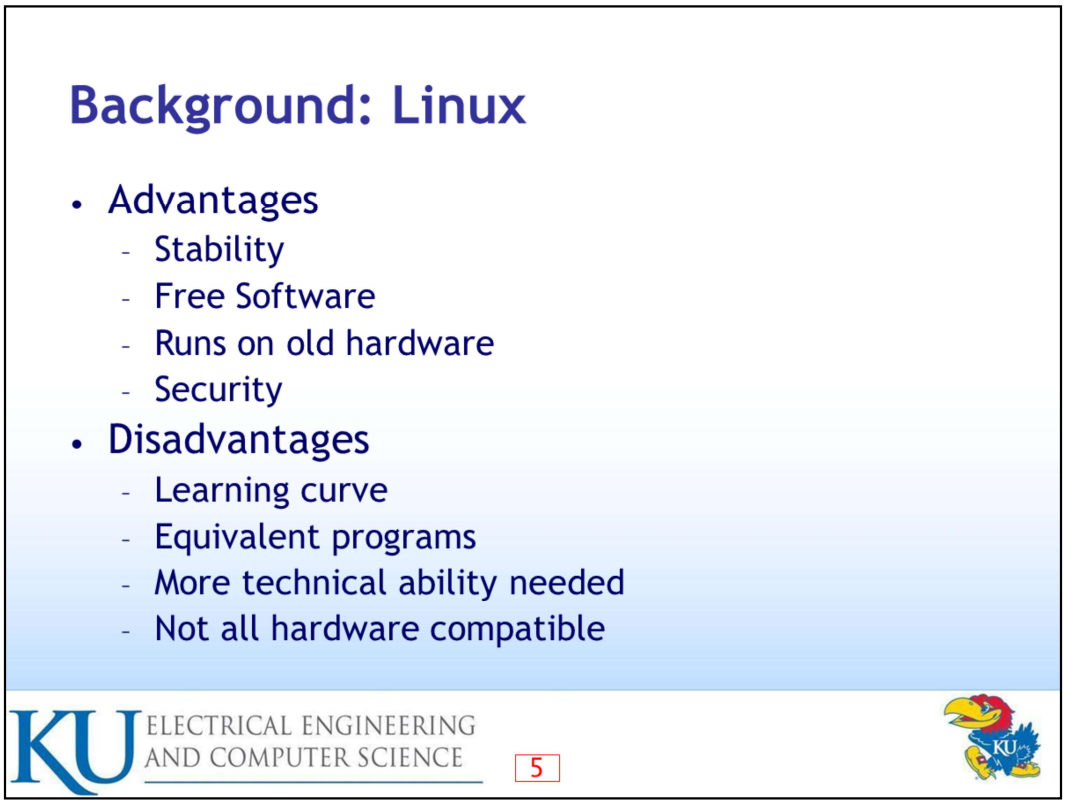
- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for “Microsoft” and “Linux” Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

Background: Operating System Market Share (October 2010)



Background: Windows

- Advantages
  - User friendly
  - Enhancements can help millions of users
  - Defects found quickly because of widespread use
- Disadvantages
  - Security defects can leave millions vulnerable
  - Non-technical user-base
  - Industry dominance leaves MS handcuffed - any move to expand capabilities seen as anticompetitive



Advantages:

The majority of Linux variants are available for free or at a much lower price than Microsoft Windows.

Linux is and has always been a very secure Operating System. Although it still can be attacked when compared to Windows, it much more secure.

The majority of Linux variants and versions are notoriously reliable and can often run for months and years without needing to be rebooted

Disadvantages:

Although the majority Linux variants have improved dramatically in ease of use, Windows is still much easier to use for new computer users

Linux has a large variety of available software programs, utilities, and games. However, Windows has a much larger selection of available software.

Required someone who knows Linux really well. Alternately, you could hire someone who has experience with Linux. A good Linux administrator needs to be on hand as you start to migrate your systems over. This is a disadvantage financially, at least in the beginning.

Some of the latest and greatest hardware that is being produced is not compatible with Linux. One thing you can do is before your purchase, ask if the hardware vendor has support for Linux. Some manufacturers do write their own Linux drivers and distribute them with your purchase, making it very easy to integrate with your existing system.

Agenda

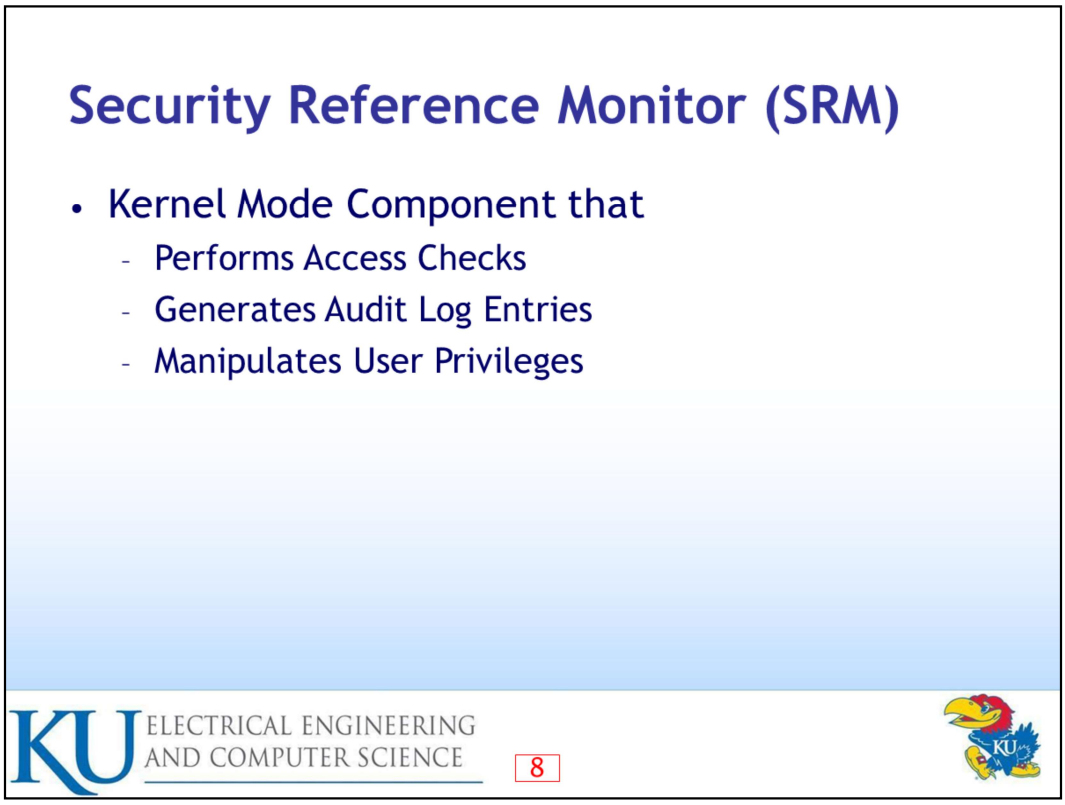
- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for “Microsoft” and “Linux” Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

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

Anyone who wants to understand windows security must have knowledge of the basic fundamental security blocks of security system.

There are many components in windows that make up the fundamental security infrastructure.



Its kernel mode component that performs access checks, generates audit log entities, and manipulates user privileges

Simply put it checks for proper authorization before granting access to objects

Object manager is client of SRM: It asks SRM if the process has the proper rights to execute a certain type of action on an object

Uses Access Control Lists to do this, which we will cover later in this presentation

Implement auditing function to keep track of attempts to access an object

Implements DoD C2 level security: Department of Defense: C2 Level Security:

The following list includes some of the most important requirements of C2-level security, as defined by the U.S. Department of Defense:

•It must be possible to control access to a resource by granting or denying access to individual users or named groups of users.

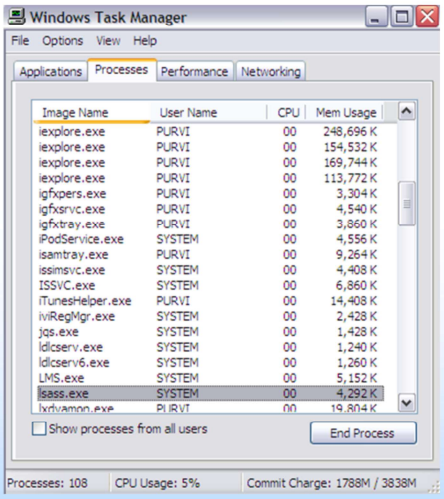
•Memory must be protected so that its contents cannot be read after a *process* frees it. Similarly, a secure file system, such as NTFS, must protect deleted files from being read.

•Users must identify themselves in a unique manner, such as by password, when they log on. All auditable actions must identify the user performing the action.

•System administrators must be able to audit security-related events. However, access to the security-related events audit data must be limited to authorized administrators.

•The system must be protected from external interference or tampering, such as modification of the running system or of system files stored on disk.

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

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

9

Resides in user-mode process named lsass.exe and is responsible for enforcing local security policy in windows.

Responsible for validating users for both local and remote logon

Issues security tokens to accounts as they log on to the system

Security policy includes

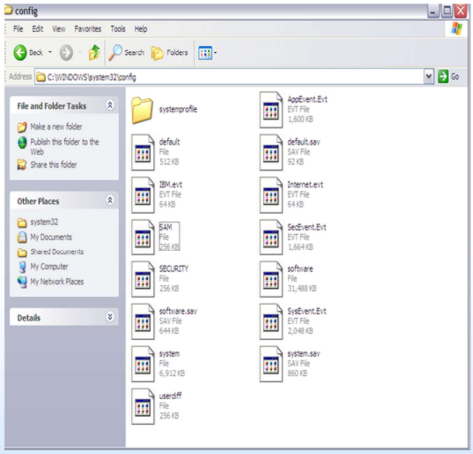
Password policy (complexity rules …expiration times etc.)

Auditing policy and privilege settings

During the local (interactive) logon to a machine, a person enters their name and password to the logon dialog. This information is passed to the LSA, which then calls the appropriate authentication package. The password is sent in a nonreversible secret key format using a one-way hash function. The LSA then queries the SAM database for the user's account information. If the key provided matches the one in the SAM, the SAM returns the users SID and the SIDs of any groups the user belongs to. The LSA then uses these SIDs to generate the security access token.

Was successfully exploited in the Sasser worm (by an 18 year old German student) :2004

Security Account Manager (SAM)



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

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 10

Database that stores user accounts and relevant security information about local users and local groups

Accounts Manager (SAM) is a registry file

Stores users' passwords in a hashed format

When a user logs on to a computer using a local account, the SAM process (Samsrv) takes the logon information and performs a lookup against the SAM database, which resides in the windows system32/config directory(Something similar in UNIX, think etc/password). If credential match, then the user can log on to the system, assuming there are no other factors preventing logon, such as logon time restrictions or privilege issues.

Note that SAM does not perform the logon; that is the job of the LSA.

The SAM file is binary rather than text and passwords are stored using the MD4 hash algorithms.

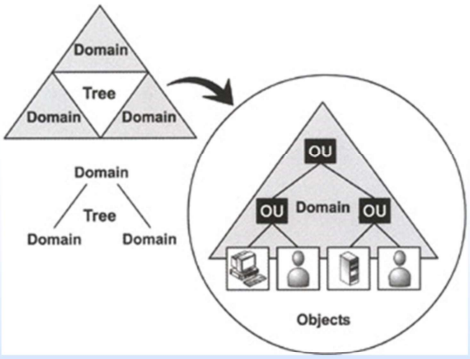
On windows vista, the SAM stores password information using a password-based key derivation function (PBKCS).

In an attempt to improve the security of the SAM database against offline software cracking, Microsoft introduced the SYSKEY function in Windows NT 4.0. When SYSKEY is enabled, the on-disk copy of the SAM file is partially encrypted, so that the password hash values for all local accounts stored in the SAM are encrypted with a key (usually also referred to as the "SYSKEY").

In the case of online attacks, it is not possible to simply copy the SAM file to another location. The SAM file cannot be moved or copied while Windows is running, since the Windows kernel obtains and keeps an exclusive filesystem lock on the SAM file, and will not release that lock until the operating system has shut down or a blue screen exception has been thrown. However, the in-memory copy of the contents of the SAM can be dumped using various techniques, making the password hashes available for offline brute-force attack.

Active Directory

- Directory Service
  - Server-based authentication
  - Centrally managed



KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 11

A directory service used to store information about the network resources across a domain and also centralizes the network.

Server based authentication

Centrally managed

Its microsofts's LDAP directory included with window server 2000 and later.

All client versions of windows, including windows XP and windows Vista, can communicate with AD to perform security operations including account logon

Windows client will authenticate using AD when the user logs on to the computer using a domain account rather than a local account.

Like the SAM scenario, the user's credential information sent securely across the network, verified by AD, and then if the information is correct, the user can Logon.

WinLogon & NetLogon

- WinLogon – keyboard requests
- NetLogon – network requests

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 12

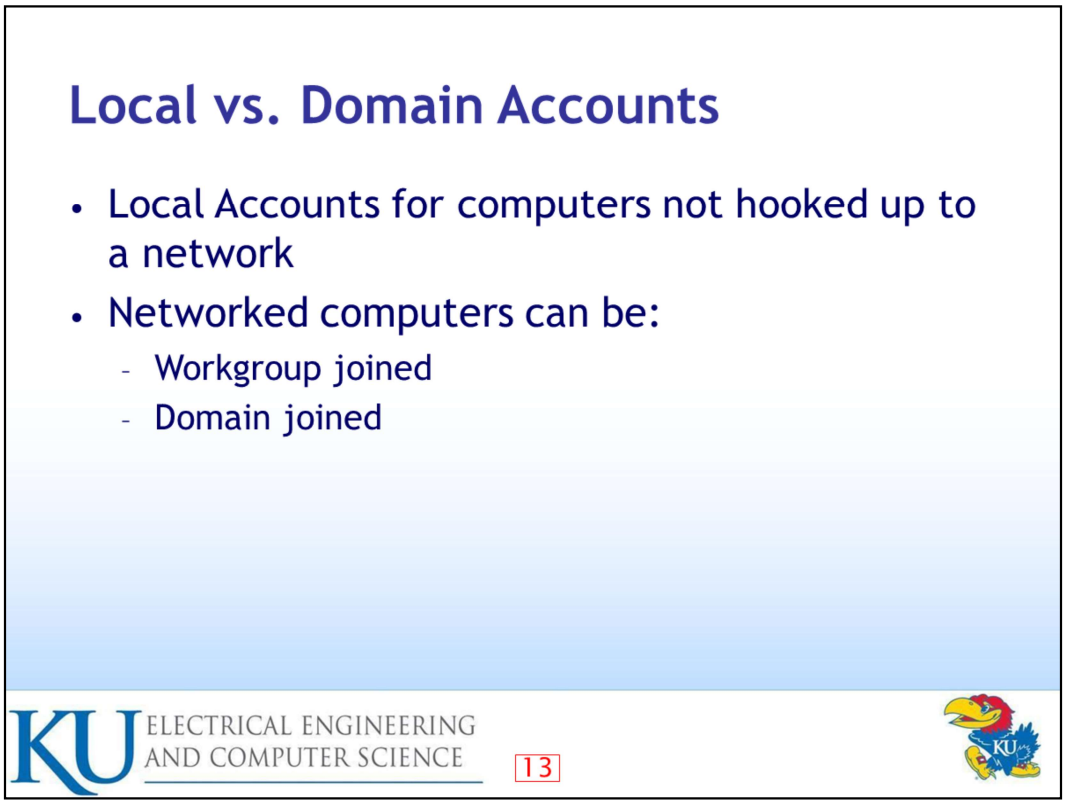
WinLogon handles local logons at the keyboard and NetLogon handles logons across the network

Winlogon is the component of Microsoft Windows operating systems that is responsible for handling the secure attention sequence, loading the user profile on logon, and optionally locking the computer when a screensaver is running (requiring another authentication step). The actual obtainment and verification of user credentials is left to other components.

Winlogon is a common target for several threats that could modify its function and memory usage. Increased memory usage for this process might indicate that it has been "hijacked". In Windows Vista and later operating systems, Winlogon's roles and responsibilities have changed significantly.

When a computer is joined to a domain, the Netlogon service allows users and services to authenticate to the domain through a secure channel. If your computer is not part of a domain, you do not need this service, but since it is only run manually, there is no benefit to disabling it

> Maintains a secure channel between this computer and the domain controller for authenticating users and services. If this service is stopped, the computer may not authenticate users and services and the domain controller cannot register DNS records. If this service is disabled, any services that explicitly depend on it will fail to start



Networked windows computer can be one of two configuration either domain joined or workgroup

Difference between a workgroup and a domain is simply where accounts are authenticated

The workgroup has no domain controllers; authentication is performed on each computer, and a domain authenticates accounts at domain controllers running AD

Workgroup Joined

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

Wokgroup: when computer is workgroup, only local accounts can be used, held in the SAM

Notion of workgroup smipy a collection of computers connected to one another using a network; but rather using central database of accounts in AD, the machines use only local accounts

Domain Joined

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

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 15

Domain joined: user can gain access to the computer using domain accounts, which are centrally managed in active directory

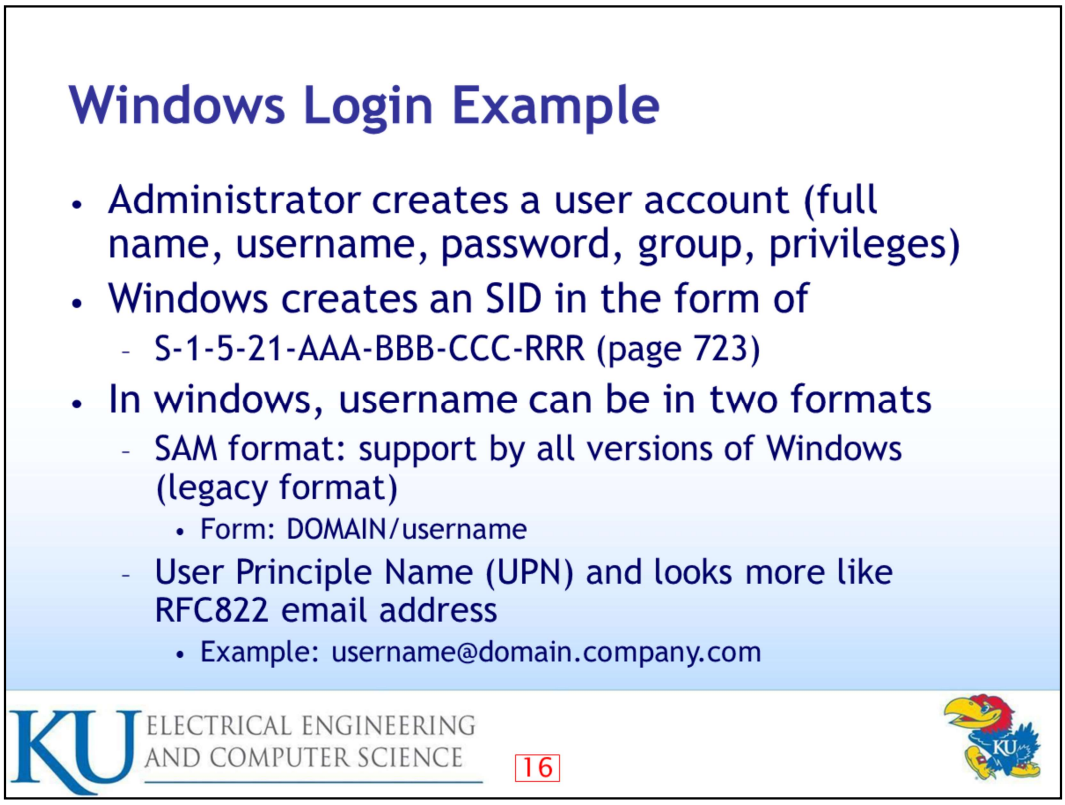
If they wish also can log on using local accounts but local accounts may not have access to domain resources such as networked printers, web servers, email servers and so on

Pros and Cons to each scenario

Domain has major advantage of being centrally managed; as such is much more secure

If environment has 1000 computers and an employee leaves, the user's account can be disabled centrally rather than on 1000 individual computer

Only advantage of using local accounts is that a computer does not need the infrastructure required to support a domain using AD



now we know the basic elements that makes up the core windows security infrastructure. Lets go over example when user logs on to a windows system

before a user log on to windows network, domain administrator must add user's account information to the system; this will include username, account name(must be unique), password

optionally the admin can grant group membership and privileges

Windows creates an SID in the form of

S-1-5-21-AAA-BBB-CCC-RRR

S : means SID

1: SID version number

5: identifier authority

21: means not unique , which just means there is no guarantee of uniqueness, a SID is unique within a domain

AAA-BBB-CCC: unique number representing domain

RRR: called relative ID (RID) ; its unique number that increments by 1 as each SID unique

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

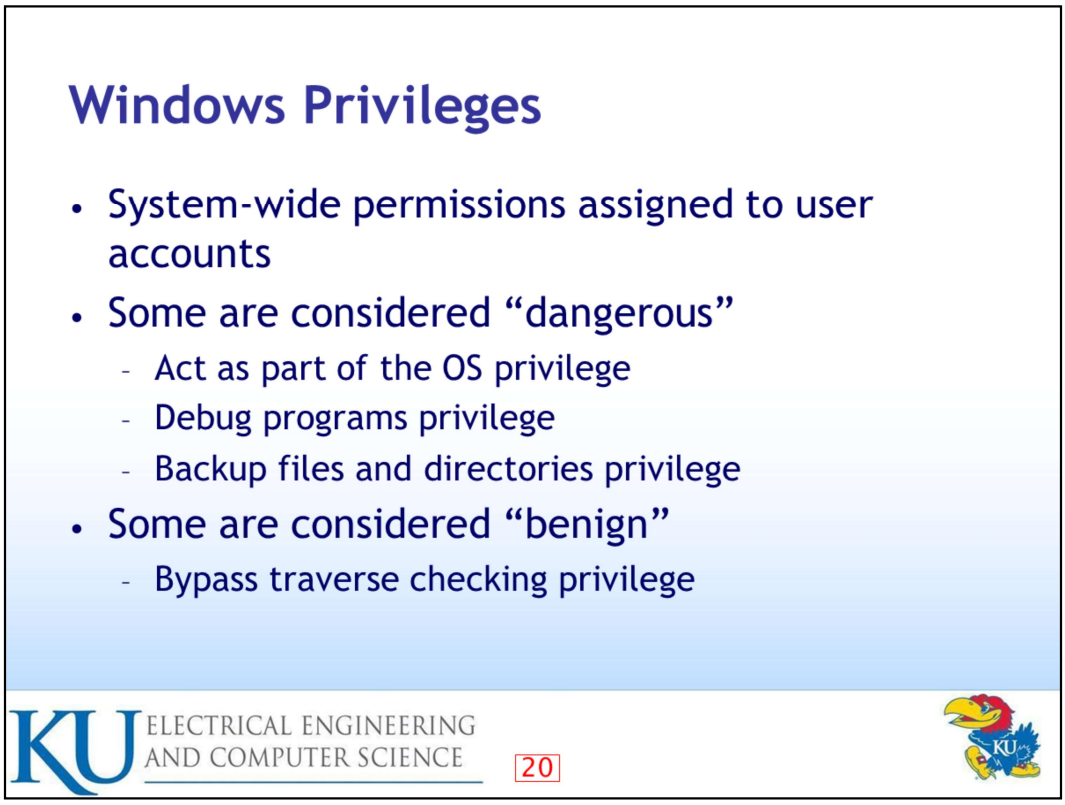
- User logs in with keyboard
- Information is sent to the AD (domain controller)
- If successful token is generated and sent to user
- Token contains
  - User's SID
  - Group membership
  - Privileges

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



Privileges are essentially system wide permissions assigned to user accounts

In other words, A privilege is something that you "get to do" . For example: windows privileges includes ability to backup the computer. Performing backup is privileged because it by pass all access checks so a complete backup is performed.

There are over 45 privileges in Windows Vista.

Some privileges are considered 'dangerous" such as

Act as part of the OS privilege: known as trusted computing base (TCB) privilege

It allows code run by the an account granted this privilege to act as part of the most trusted code in operating system: security code

Most dangerous, is granted only the local system account, even admin are granted this privilege

Debug programs privilege:

Allows an account to debug any process running in windows

Because of nature of debuggers, this privilege basically means a user can run any code he or she wants in any running process

Backup files and directories privilege

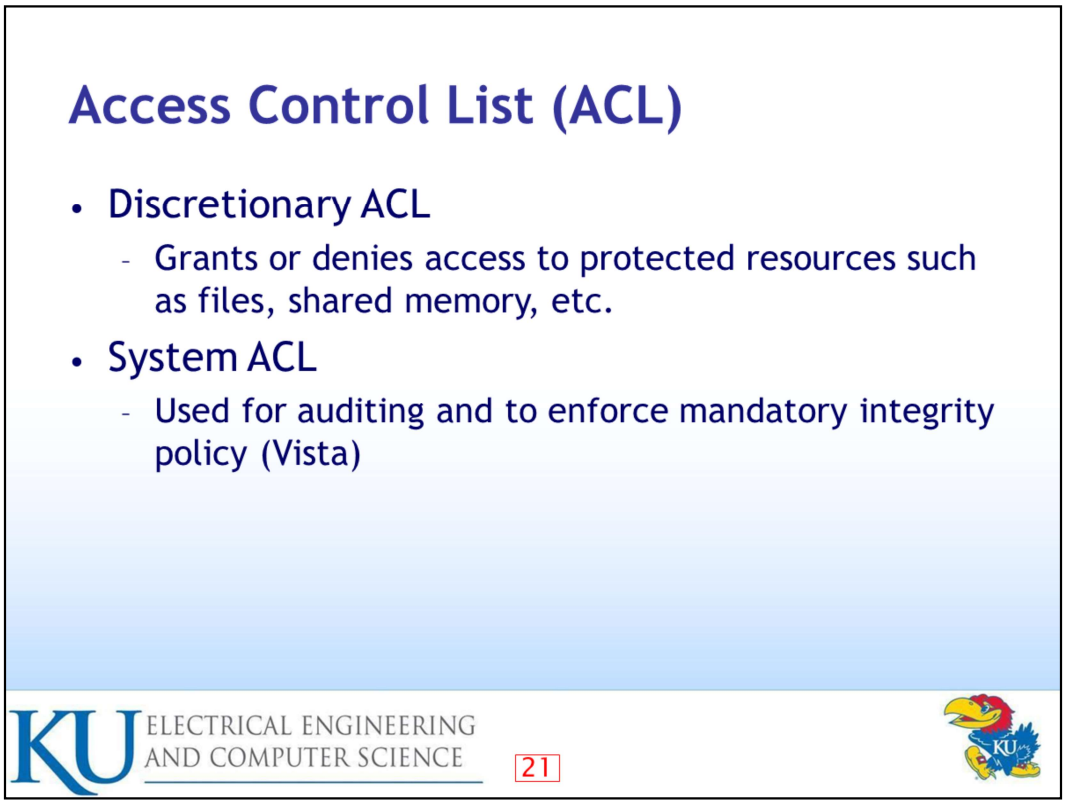
Any process running with this privilege will by pass all access control list(ACL) checks

Some privileges are generally deemed "benign"

Bypass traverse checking

Used to traverse directory trees even though the user may not have permission on the traversed directory.

This privilege is assigned to all user accounts by default and is used as NTFS file system optimization



An ACL is a list of permissions attached to an object

The list specifies who or what is allowed to access the object and what operations are allowed to be performed on the object.

Windows has two forms of access control list

Discretionary ACL (DACL):

Grants and denies access to protected resources in windows such as files, shared memory

System ACL (SACL)

Used for auditing and in windows vista used to enforce mandatory integrity policy

Objects that requires protection are assigned a DACL (and possible SACL), which includes SID of object owner as well as a list of access control entries (ACEs)

Each ACE includes a SID and an access mask

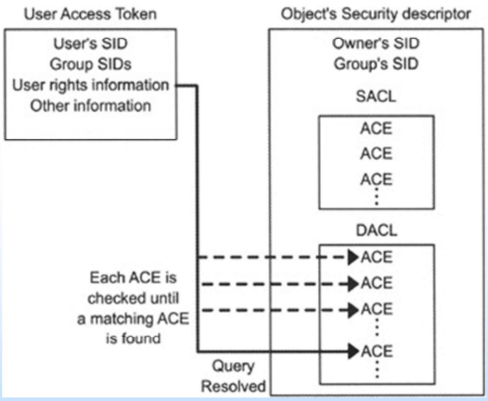
Access mask could include an ability to read, write, create, delete , modify and so on

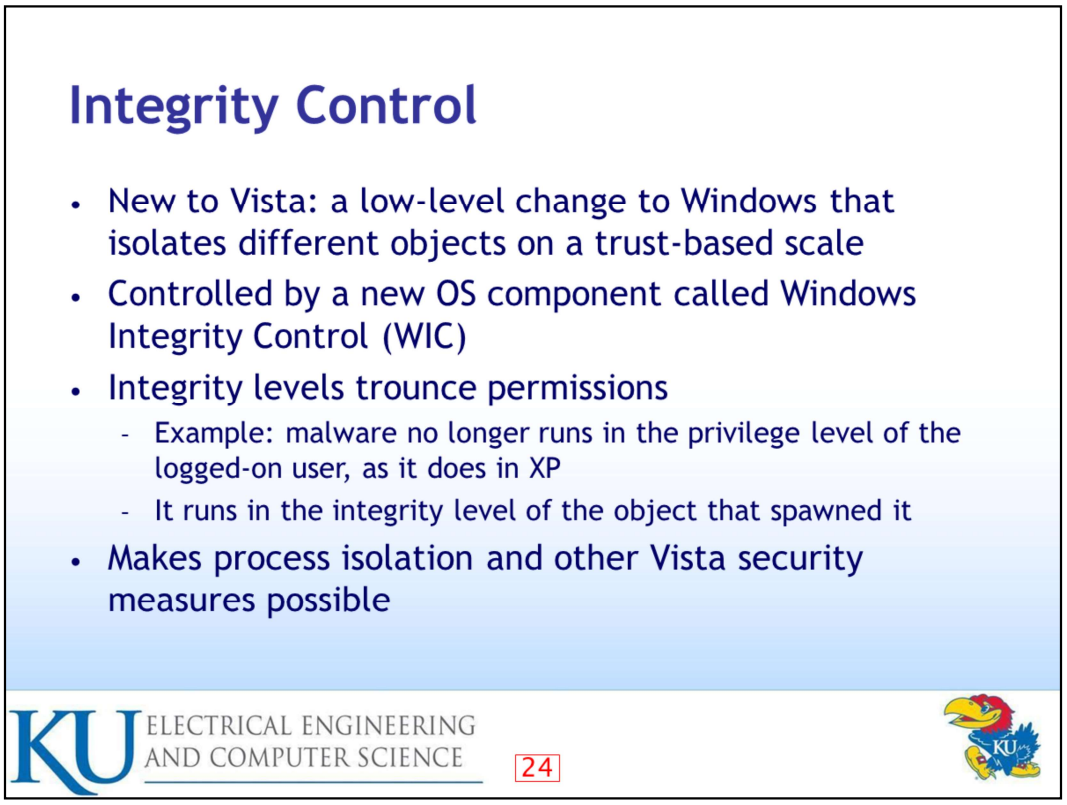
Access Control Lists (ACL) (continued)

- 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





Windows Integrity Control (WIC) capabilities in Windows Vista by examining how it protects objects such as files and folders on Vista computers, the different levels of protection offered

WIC is intended to protect a system from malware and user error by helping to establish different levels of trust on objects.

The purpose of WIC is to protect objects, whether they are files, printers, named pipes, registry keys, and so on from attacks, malware or even innocent user error. The concept of WIC is based on establishing the trustworthiness of the various objects and controlling the interactions between objects based on their integrity, or level of trustworthiness.

The primary objective of WIC is to ensure that only objects with an integrity level equal to or greater than the target object are allowed to interact with it. Essentially, if an object is less trustworthy, it is prohibited from acting on, or interacting with more trustworthy objects

Six Integrity Levels

Object and Principals are labeled

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

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 25

In order to police the interactions between objects, Windows must first determine the trustworthiness, or integrity level of each object. WIC assigns one of the following six integrity levels to each object:

Untrusted. Rarely seen; for anonymous logons.

Low. Internet features (including IE 7 and the Temporary Internet Files folder).

Medium. The default integrity level. Used for Standard User accounts and most files.

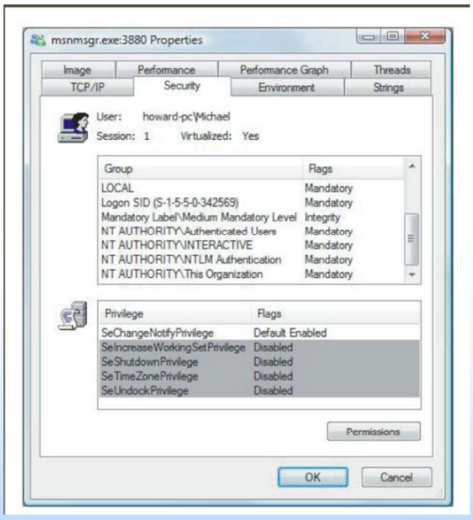
High. Administrator accounts.

System. Most of the kernel and system services.

Installer. Why? Because installers need to have a higher integrity level than other objects in order to ensure uninstall works properly.

MAC: Vista Integrity Control

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



KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 26

One of my favorite new security features in Windows Vista is Mandatory Integrity Control (MIC).

While discretionary access control lists (DACLs) are useful, they have some limitations. They do little to safeguard system stability and they can't stop malicious software from tricking users into executing it.

MIC adds the notion of trustworthiness evaluation into the operating system. Subjects with low degrees of trustworthiness can't change data of higher degrees; subjects with high degrees of trustworthiness can't be forced to rely on data of lower degrees

Whiteboard: When a Modify (e.g. Write) operation occurs first check that the Subjects integrity level dominates the Objects integrity level.

> I find these levels interesting. Levels are mandatory, and are always assigned by the OS itself or an administrator.

Look how different elements of the system are trusted in different ways.

Internet below standard user. [Smart]

Installer? Image an attack found a way to create a file or process at a level above "High." [Even the administrator wouldn't be able to delete it]

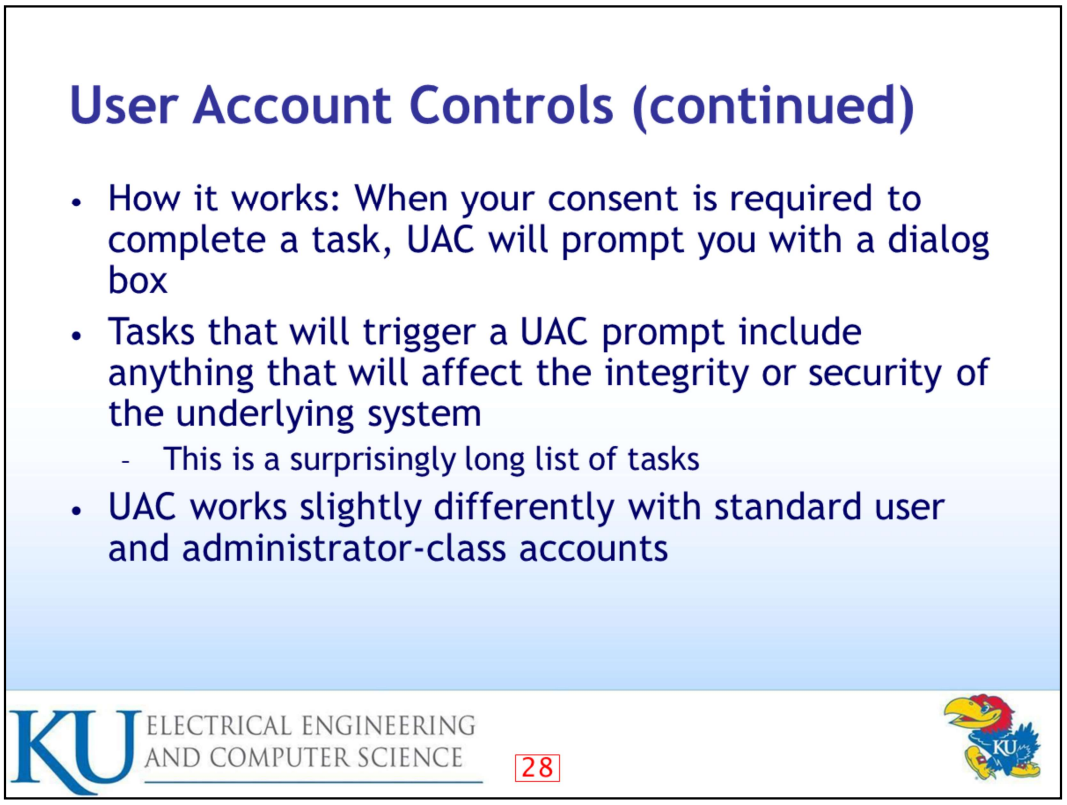
User Account Controls

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

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 27

This feature is so well-known that even Apple makes fun of it in their "I'm a PC, I'm a Mac ads." But here's some irony and perhaps hypocrisy for you: Apple's Mac OS X includes a feature just like User Account Control and always had. It's just good security.

Put another way: UAC tries to help you from making stupid mistakes. It also tries to prevent malware from acting on your behalf.

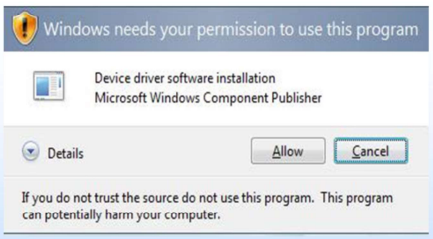


How it works: When your consent is required to complete a task, UAC will prompt you with a dialog box.

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 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

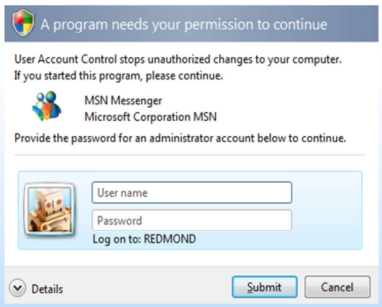


This dialog occurs when you try to change a Windows system component.

In the example shown, the user is an admin-class user, so they only have to click “Allow” (or type CTRL+A) to move along. If the user were a standard user, you’d see the same dialog, but with a space for entering the name and password of an admin-level account.

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



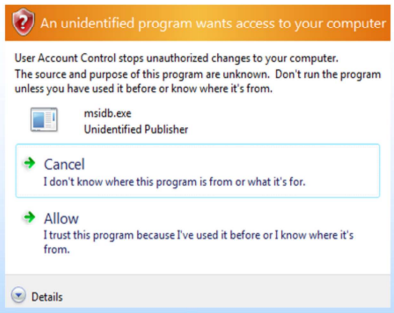
KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 30

This one is identical to the previous example except that an external application is attempting to perform a task that requires ***a higher integrity level***.

In this example, I am showing the standard user experience, where a user name and password is required to continue. Had this been an admin account, you'd only see the Allow/Cancel buttons. Note that the applications is "trusted" in the sense that Vista can confirm its digital signature. But it still requires elevation, so UAC snaps to attention.

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



KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 31

My favorite UAC dialog, because it's so colorful and large: This happens when an unknown application--e.g. virtually any third party application you download from the Internet today--tries to run on Vista.

Again, this is the admin experience, with Allow/Cancel options. But notice how different this dialog is. Microsoft is *really* trying to get your attention this time. That's because Vista has no idea whether this application is safe.

Do you see the problem with these dialogs? Aside from being annoying, which they are, the fear is that users will simply get used to them popping up and will mindlessly click Allow all the time.

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)

Review Question

- When a Modify (e.g. Write) operation occurs first check that the subjects integrity level dominates the objects integrity level
- This is most like a property of which?
  - Bell-LaPadula model
  - Biba Model
  - Chinese Wall Model

Answer: Biba Model

- Simple Integrity Rule
  - A subject can modify an object only if the integrity level of the subject dominates the integrity level of the object
    - » I(S)>=I(O)

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

Linux Security Model

- Overview of Linux Security Model
- File System Security (DAC)
- Users and Groups
- File and Directory Permissions
- Kernel Space vs. User Space

Overview of Linux Security Model

- Since Linux Torvalds created in 1991, it has been evolved into one of the world's most popular and versatile operating system
- Free, open-sourced and available in a wide variety of "distributions"
- Traditional security model
  - People or processes with "root" privileges can do anything
  - Other accounts can do much less
- Goal of hackers – to gain root privilege
- Linux can be run robust and secure
  - Many system Admin fail to use the security features
  - Add-on tools like sudo and Tripwire available
- Crux of the problem – Discretionary Access Control

Traditional security model

People or processes with "root" privileges can do anything

Other accounts can do much less

Thus, from attacker perspective, the challenge is cracking a linux system therefore boils down to grating root privileges

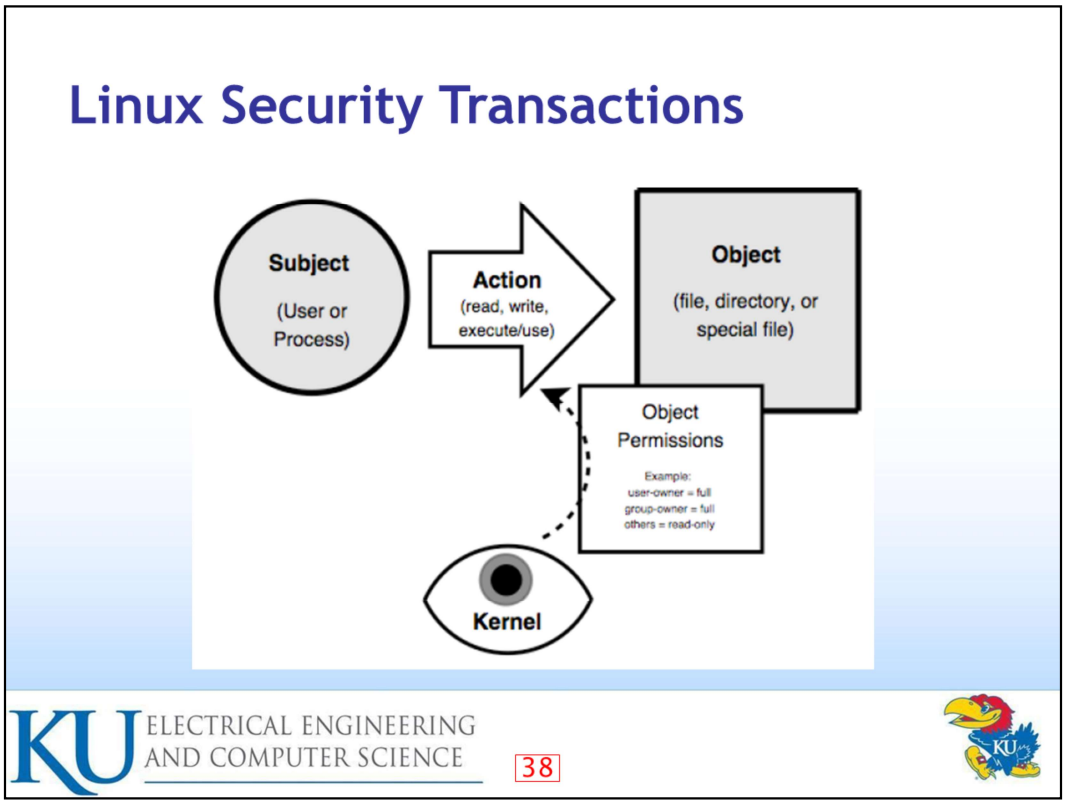
Once root privileges granted, attackers can erase or edit logs; hide their processes, files and directories; and basically redefine the reality of the system as experienced by its admins and users.

Linux security is a game of "root takes all"

Main cause of exposure: many linux admins fail to take advantage of the security features available to them

People can and do run robust, secure linux systems by making careful native linux security contols, plus selected add-on tools such as sudo or Tripleware

However crux of the problem of linux secutiry is linux security model on discretionary access control (DAC)



In the linux DAC system there users, each belongs to one or more group

There are objects: files and directories

Users read, write, and execute these objects, based on the object's permissions of which each object has three sets: user-owner, group-owner, and other

These permissions are enforced by the linux kernel, the brain of operating system

Basic transaction: subject attempts some action against some object

Whoever owns an object can set or change its permissions

Real weakness: root account has ability to both takes ownership and change the permissions s of all objects in the system, this provide way for attackers to hijack those privileges

Lets take closer look at how the linux DAC implementation actually works

File System Security

- In Linux everything is a file
- I/O to devices is via a "special" file
  - Example: /dev/cdrom points to /dev/hdb which is a special file
- Have other special files like named pipes
  - A conduit between processes / programs
- Since almost everything a file – security very important

Linux treats everything as file

Documents, pictures even executable programs are very easy to conceptualize as files

Special files such as named pipes, act as I/O "conduits", allowing one process or program to pass data to another.

One common example of a named pipe on linux systems is /dev/urandom: when programs reads this file /dev/urandom return random characters from kernal's random number generators

Users and Groups

- Users and Groups are not files
- Users
  - Someone or something capable of using files
  - Can be human or process
  - e.g. lpd (Linux Printer Daemon) runs as user lp
- Groups
  - List of user accounts
  - User's main group membership specified in /etc/passwd
  - User can be added to additional group by editing /etc/group
  - Command line -> useradd, usermod, and userdel

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 40

There are two things on a Unix system that aren't represented by files: user accounts and group accounts which for short we can call users and groups

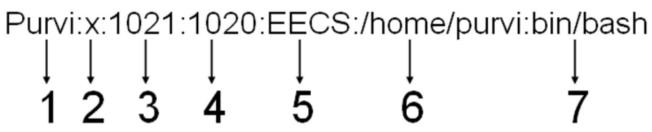
User account represents some or something capable of using files…ex users, processes

> Standard linux user accounts "lp" for example, is used by the line printer demon(lpd): lpd program runs as the user lp

A group account is simply a list of user accounts. Each user accounts is defined with a main group membership, but may in fact belongs to as many groups as you went or need it to

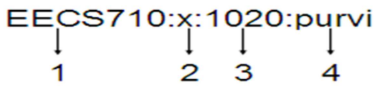
> User's main group memebership is specified in the user account's entry in /etc/password; you can add that user to additional groups by editing /etc/group and adding the username to the end of the entry for each group the user needs to belong to or via the usermod command

Understanding: /etc/password



1. username: Used when user logs in. It should be between 1 and 32 characters in length
2. password: An x character indicates that encrypted password is stored in /etc/shadow file
3. user ID (UID): Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts
   - UID 100-999 are reserved by system for administrative and system accounts/groups
4. group ID (GID): The primary group ID (stored in /etc/group file)
5. user ID Info: The comment field
   - Allows you to add extra information about the users such as user's full name, phone # etc
   - This field used by finger command
6. home directory: The absolute path to the directory the user will be in when they log in
   - If this directory does not exists then users directory becomes /
7. command/shell: The absolute path of a command or shell (/bin/bash)
   - Typically, this is a shell. Please note that it does not have to be a shell.

Understanding of /etc/group



1. group_name: Name of group
2. password: Generally password not used, hence it is empty/blank. It can store encrypted password. Useful to implement privileged groups
3. group ID (GID): Group ID must be assigned to every user
4. group List: List of user names of users who are members of the group. The user names must be separated by commas

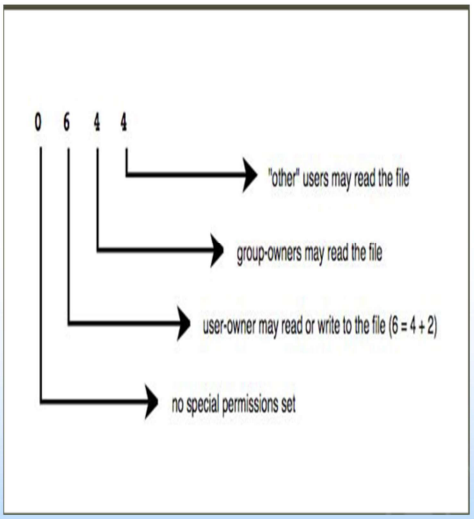
File Permissions

- Files have two owners: a user & a group
  - Each with its own set of permissions
  - With a third set of permissions for other
- Permissions are to read/write/execute in order user/group/other
  - rw-rw -r-- 1 maestro user 35414 Mar 25 01:38 baton.txt
- Permission can be changed using chmod command

Numeric File Permissions

- Read (r) = 4
- Write (w) = 2
- Execute (x) = 1
- Example from textbook:

  drwxr-x--- 8 biff drummers
  288 Mar 25 01:38
  extreme_casseroles



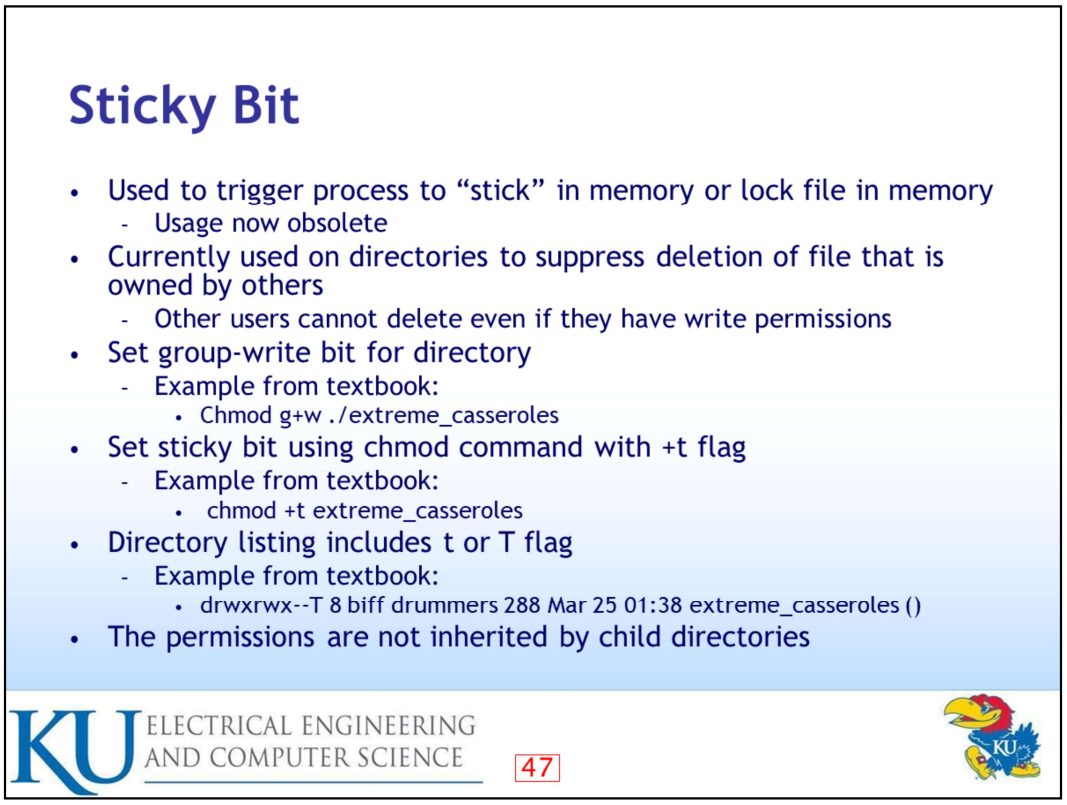
Directory Permissions

- Permissions on folder slightly works different
  - read = list contents
  - write = create or delete files in directory
  - execute = use anything in or change working directory to this directory
- Example from textbook:

```
$ chmod g+rx extreme_casseroles
$ ls -l extreme_casseroles
  drwxr-x--- 8 biff drummers 288 Mar 25 01:38
  extreme_casseroles
```


Difference between File and Directory Permissions

Access Type	File	Directory
Read	If the file contents can be read	If the directory listing can be obtained
Write	If user or process can write to the file (change its contents)	If user or process can change directory contents somehow: create new or delete existing files in the directory or rename files.
Execute	If the file can be executed	If user or process can access the directory, that is, go to it (make it to be the current working directory)



Drummer friend Biff wants to all his fellow drummers not only to read this recipes, but also to add their own….he needs set the "group-write" bit for this directory.

There's only one problem with this: "write" permissions include both the ability to create new files in this directory, but also to delete them. What's to stop one of his drummer pals from deleting other people's recipes? The 'sticky bit"

In linux, When you set sticky bit on a directory, it limits user ability to delete things in that directory, to delete a given file in the directory you must either own that file or own that directory.

To issue sticky bit, issue chomd comand with +t flag

After issuing sticky bit command, not the "T" at the end of permissions string, "T" denotes that the directory is not "other-executable" but has the stick bit set.

A lowercase "t" would denote that directory is othe-executable and has the stick bit set

SetUID and SetGID

- SetUI bit: means program "run with same privileges as" owner
  - No matter who executes it
- SetGID bit: means "run with same privileges as" a member of group that owns it
  - Again regardless of who executes it
- *Are very dangerous* if set of any file owned by if root or other privileged account or group
  - Only used on executable files, not shell scripts
  - The command "sudo" is much better tool for delegating root's authority
- Note that the linux kernel ignores the setid and setgid bits on shell scripts; these bits only work on binary (compiled) executables

Most dangerous permissions bit in the unix world" setuid and setgid

SetGID and Directories

- SetUID has no effects on directories
- SetGID does and causes any file created in a directory to inherit the directory's group-owner
- Useful if users belong to other groups and routinely create files to be shared with other members of those groups
  - Instead of manually changing its group

Kernel Space and User Space

- Kernel space: refers to memory used by the Linux kernel and its loadable modules (e.g device drivers)
- User space: refers to memory used by all other processes
- Since kernel enforces Linux DAC and security, its extremely critical to isolate kernel from user space
  - For this reason, kernel space never swapped to disk
  - Only root may load and unload kernel modules

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

DAC security model, owner of given system object can set whatever access permissions on that resource he/she likes

User who creates a file on a MAC system generally may not set access controls on that file that are weaker than the controls dictated by the system security policy

In MAC based system, only thing the superuser account is used for is maintaining the global security policy. Day-to-day system administration is performed using accounts that lack the authority to change the global security policy. As a result, its impossible to compromise the entire system by attacking any one process.

To create an effective global security policy requires detailed knowledge of the intended behavior of application on the system. Thus MAC systems have been hard to manage. Furthermore the more restrictive the security controls are on a given system, less convenient that system becomes for its users to use.

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

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
  - This caused many legacy Windows applications to fail
  - Solution: Windows XP includes a compatibility mode - a mode that allows programs to operate as if they were running in the original insecure single-user design
- Windows XP represented progress, but even Windows XP could not be justifiably referred to as a true multi-user system

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. It also means viruses, Trojans and other malware could tamper with any critical system program or file, because Windows did not isolate users or applications from these sensitive areas of the operating system.

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. This caused many legacy Windows applications to fail, because they were used to being able to access and modify programs and files that only an administrator should be able to access. That's why Windows XP includes a compatibility mode - a mode that allows programs to operate as if they were running in the original insecure single-user design. This is also why each new version of Windows threatens to break applications that ran on previous versions. As Microsoft is forced to hack Windows into behaving more like a multi-usersystem, the new restrictions break applications that are used to working without those restraints.

Windows XP represented progress, but even Windows XP could not be justifiably referred to as a true multi-user system.

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
- Thus, Monolithic system tends to make security vulnerabilities more critical than they need to be

monolithic system is one where most features are integrated into a single unit.

Another direct result of deliberate design decisions such as its monolithic design (integrating too many features into the core of the operating system)

Microsoft made the Netscape browser irrelevant by integrating Internet Explorer so tightly into its operating system that it is almost impossible not to use IE. Like it or not, you invoke Internet Explorerwhen you use the Windows help system, Outlook, and many other Microsoft and third-party applications.

best business interest of Microsoft but this approach made impossible to escape from independent services

Interdependencies like these have two unfortunate cascading side effects.

First, in a monolithic system, 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

Example: When Microsoft integrated Internet Explorer into the operating system, Microsoft created a system where any flaw in Internet Explorer could expose your Windows desktop to risks that go far beyond what you do with your browser

A single flaw in Internet Explorer is therefore exposed in countless other applications, many of which may use Internet Explorer in a way that is not obvious to the user, giving the user a false sense of security.

Finally, a monolithic system is 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

The Windows XP service pack 2 already has a growing history of causing existing third-party applications to fail.

This is the natural consequence of a monolithic system - any changes to one part of the machine affect the whole machine, and all of the applications that depend on the machine.

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
  - Unfortunately, Windows users cannot disable RPC because Windows depends upon it, even if your computer is not connected to a network
- The most common way to exploit an RPC-related vulnerability is to attack the service that uses RPC, not RPC itself

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. For example, one program can use an RPC to tell another program to calculate the average cost of tea in China and return the answer. The reason it's called a *remote* procedure call is because it doesn't matter if the other program is running on the same machine, another machine in the next cube, or somewhere on the Internet.

RPCs are potential security risks because they are designed to let other computers somewhere on a network to tell your computer what to do. Whenever someone discovers a flaw in an RPC-enabled program, there is the potential for someone with a network-connected computer to exploit the flaw in order to tell your computer what to do. Unfortunately, Windows users cannot disable RPC because Windows depends upon it, even if your computer is not connected to a network. Many Windows services are simply designed that way. In some cases, you can block an RPC port at your firewall, but Windows often depends so heavily on RPC mechanisms for basic functions that this is not always possible.

The most common way to exploit an RPC-related vulnerability is to attack the service that uses RPC, not RPC itself

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
  - This makes the Windows administrator most vulnerable to security flaws, because using vulnerable programs such as Internet Explorer expose the server to security risks

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
  - Therefore it has been 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

Linux does not have a history of being a single-user system. Therefore it has been 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. It can only write to the user's own home directory. It cannot write to a system file or even to another user's directory unless the administrator explicitly gives the user permission to do so.

Even more important, Linux provides almost all capabilities, such as the rendering of JPEG images, as modular libraries. As a result, when a word processor renders JPEG images, the JPEG rendering functions will run with the same restricted privileges as the word processor itself. If there is a flaw in the JPEG rendering routines, a malicious hacker can only exploit this flaw to gain the same privileges as the user, thus limiting the potential damage. This is the benefit of a modular system, and it follows more closely the spherical analogy of an ideally designed operating system (see the section *Windows is Monolithic by Design, not Modular*).

Given the default restrictions in the modular nature of Linux; it is nearly impossible to send an email to a Linux user that will infect the entire machine with a virus.

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
- The Linux kernel supports modular drivers, but it is essentially a monolithic kernel where services in the kernel are interdependent
  - Any adverse impact of this monolithic approach is minimized by the fact that the Linux kernel is designed to be as minimal a part of the system as possible

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; at least enough so that an update to one part of GNOME or KDE can potentially break other parts of GNOME or KDE

The Linux kernel supports modular drivers, but it is essentially a monolithic kernel where services in the kernel are interdependent. Any adverse impact of this monolithic approach is minimized by the fact that the Linux kernel is designed to be as minimal a part of the system as possible.

Linux follows the following philosophy almost to a point of fanaticism: "Whenever a task can be done outside the kernel, it must be done outside the kernel." This means that almost every useful feature in Linux ("useful" as perceived by an end user) is a feature that does not have access to the vulnerable parts of a Linux system.

Not Constrained by an RPC Model

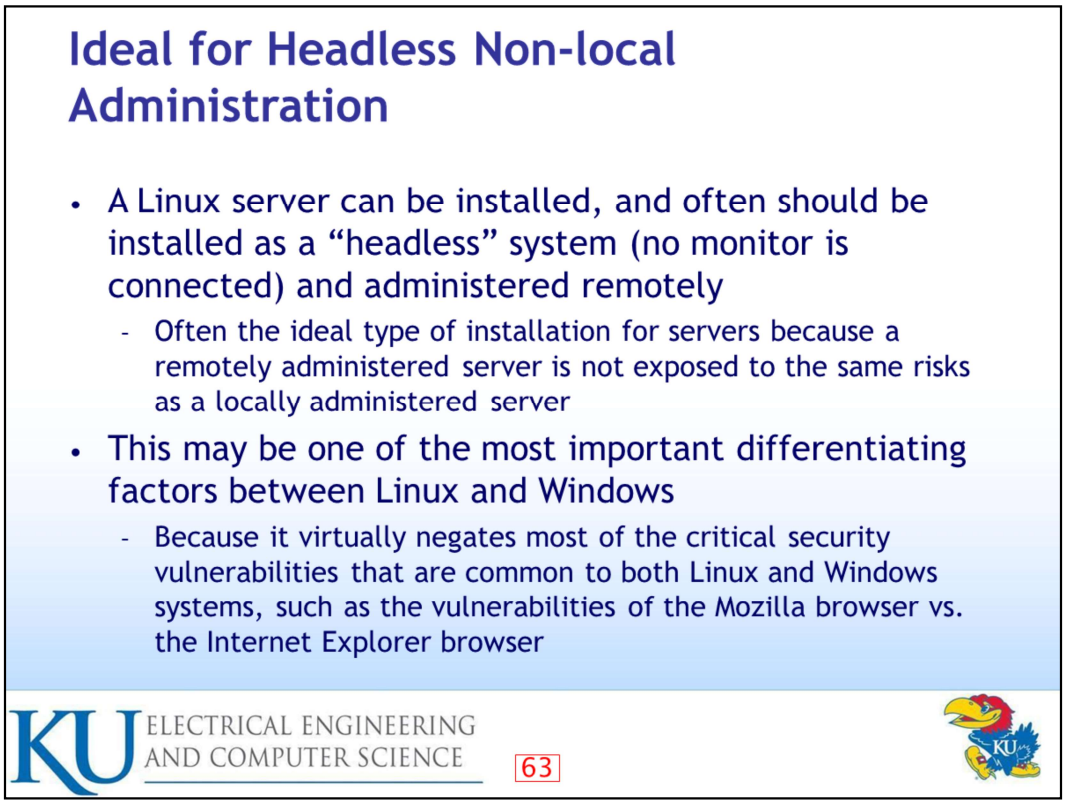
- Most Linux distributions install programs with network access turned off by default
- Even when Linux applications use the network by default, they are most often configured to respond only to the local machine and ignore any requests from other machines on the network
- Unlike Windows Server 2003,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

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 62

Most Linux distributions install programs with network access turned off by default. For example, the MySQL SQL database server is usually installed such that it does not listen to the network for instructions. If you build a web site using Apache and MySQL on the same server machine, then Apache will interact with MySQL without MySQL having to listen to the network

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.



A Linux server can be installed, and often should be installed as a "headless" system (no monitor is connected) and administered remotely. This is often the ideal type of installation for servers because a remotely administered server is not exposed to the same risks as a locally administered server.

For example, you can log into your desktop computer as a normal user with restricted privileges and administer the Linux server through a browser-based administration interface. Even the most critical browser-based security vulnerability affects only your local user-level account on the desktop, leaving the server untouched by the security hole.

This may be one of the most important differentiating factors between Linux and Windows, because it virtually negates most of the critical security vulnerabilities that are common to both Linux and Windows systems, such as the vulnerabilities of the Mozilla browser vs. the Internet Explorer browser

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

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

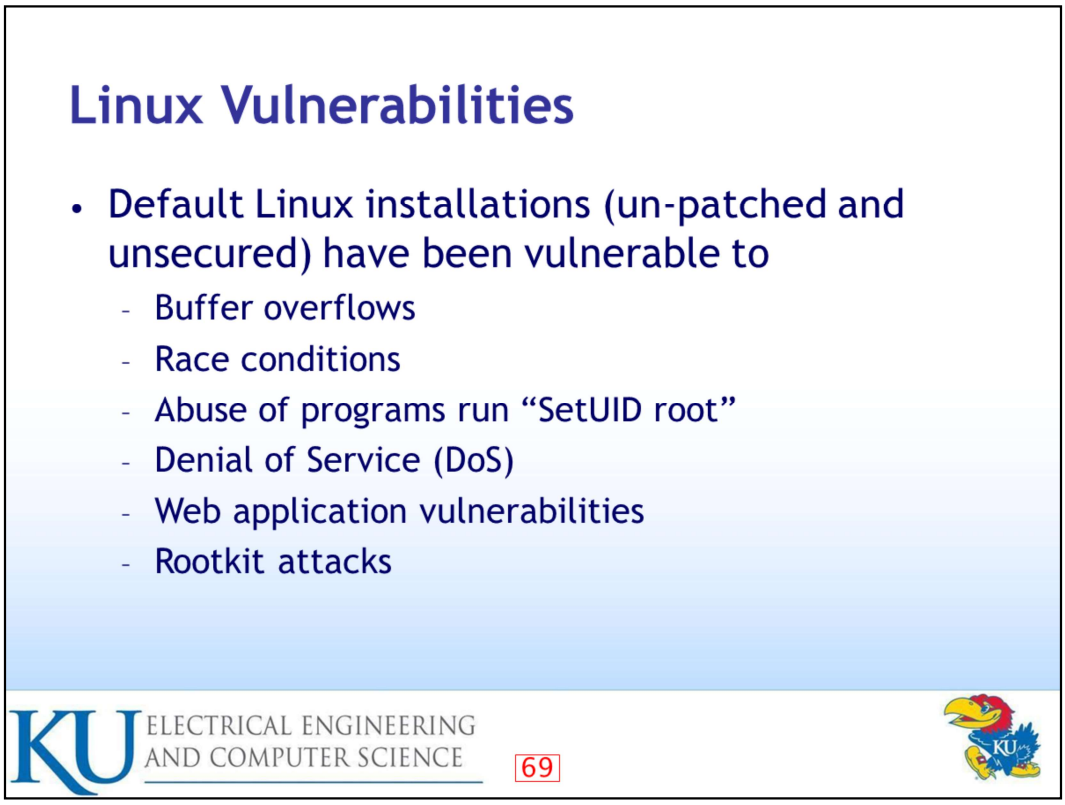
- Vulnerabilities in Windows Kernel Could Allow Elevation of Privilege Microsoft Security Bulletin MS10-021, April 2010)
  - Most severe of these vulnerabilities could allow elevation of privilege if an attacker logged on locally and ran a specially crafted application
  - An attacker must have valid 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

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion



We will discuss on next few slides weakness of linux systems

Some of the most common and far-reaching vulnerabilities in default linux installations

Buffer overflows

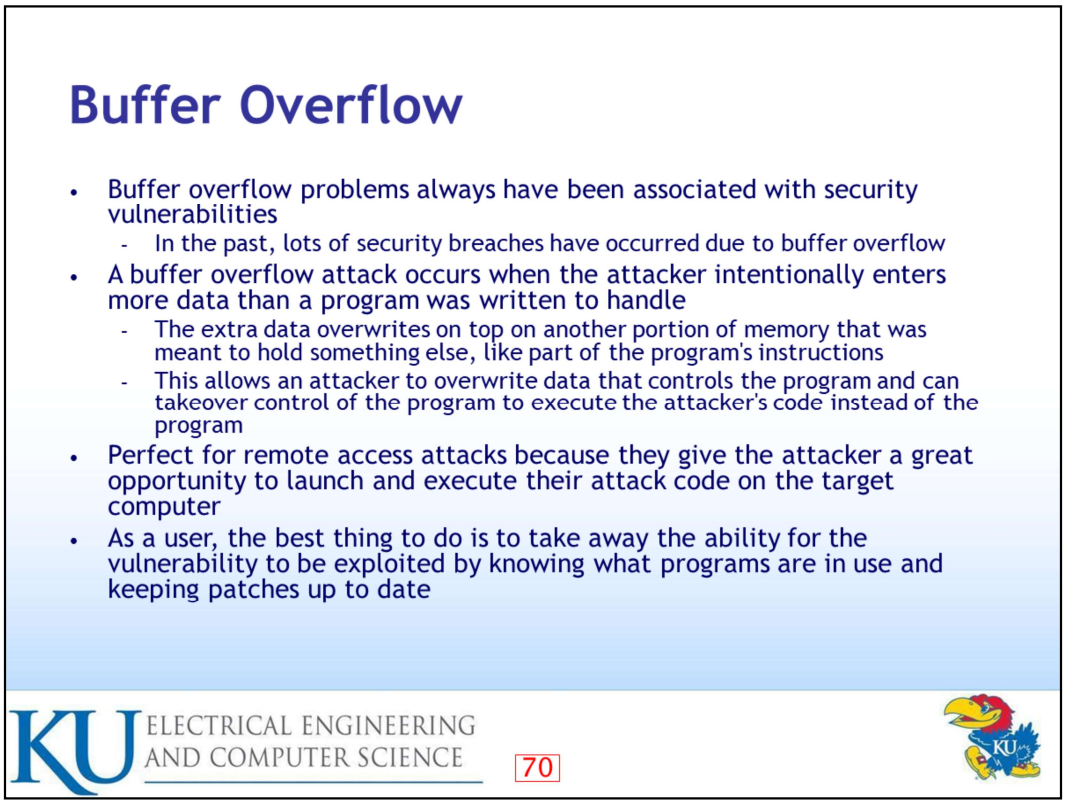
Race conditions

Abuse of programs run "SetUID root"

Denial of Service (DoS)

Web application vulnerabilities

Rootkit attacks



We talk about buffer overflow, security vulnerabilities are associated with it

What is buffer overflow?

Buffer overflow occurs when attackers intentionally enters more data than a program was written to handle

As a user, the best thing to do is to take away the ability for the vulnerability to be exploited by know what programs are in use and keeping patches up to date

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

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 71

As we discuss earlier when “setuid” permissions bit is set program will run with the privileges of the user that owns it, rather than those of the process or user executing it

SetUID root program is a root-owned program---Runs as root no matter who executes it

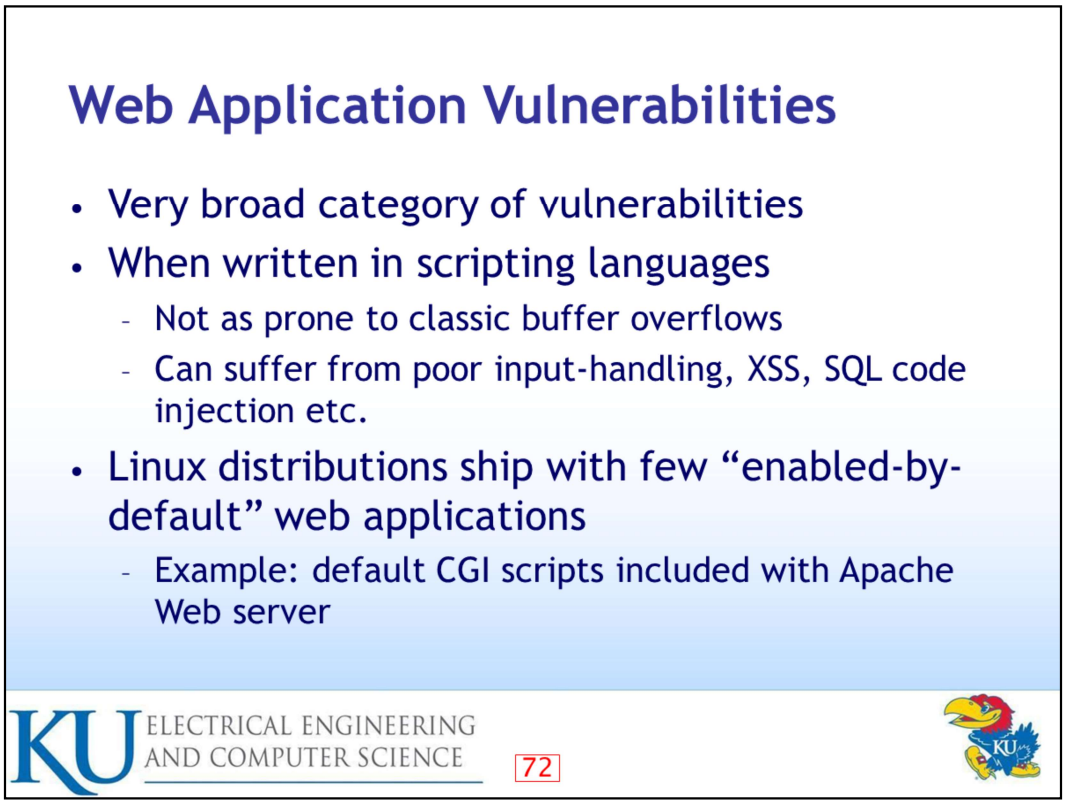
If setuid root program can be exploited or abused in some way (for example via buffer overflow or race condition) then Unprivileged users can gain access to unauthorized privileged resources

So running setuid root is necessary for programs that need to be run by unprivileged users yset must provide such users with access to privileged functions. For example changing their password, which requires changes to protected system files.

Thus such program required must programmed very carefully

Due to history of abuse against setuid root programs, major linux distributions no longer ship with unneccessary setuid-root programs

But System attackers still scan for them

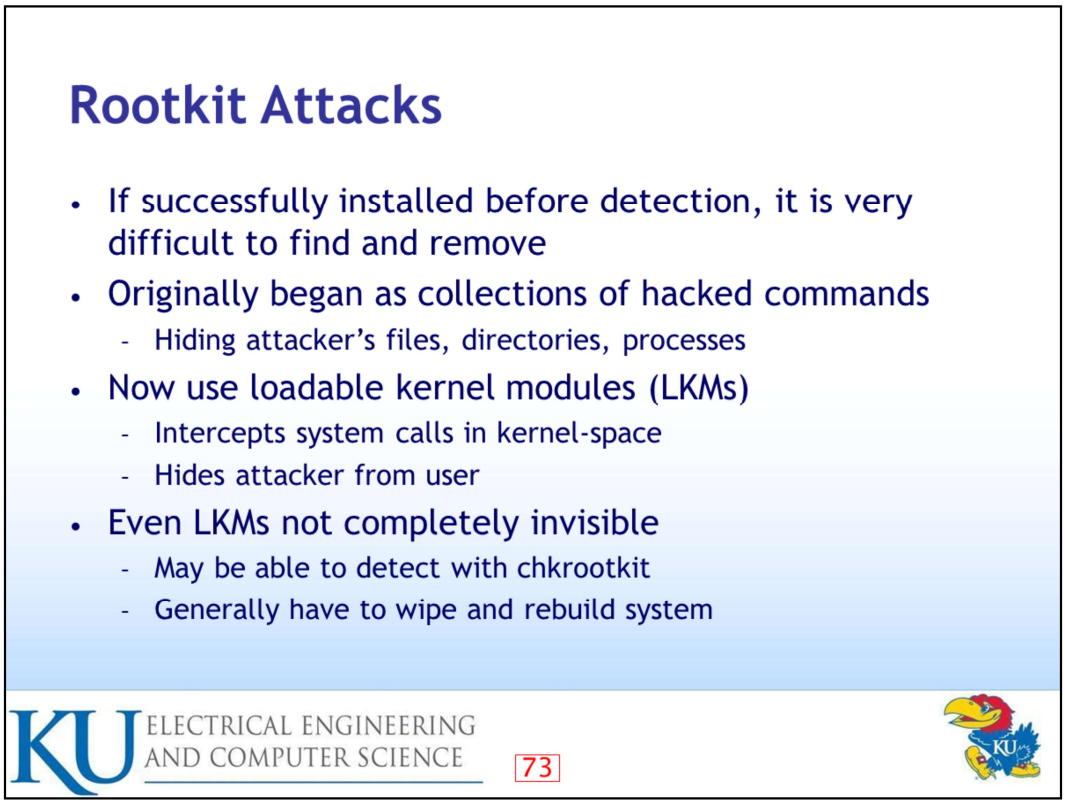


This is very broad category of vulnerabilitie

As we know web applications written in scripting languages such as PHP, Perl and Java, thus it may not be as prone to classic buffer overflows (thanks to the additional layers of abstraction presented by those languages)

However it can suffer from poor input handling, including cross-site scriptiing, SQL code injection

Now days, Linux distributions ship with few "enabled-by-default" web applications…for example default CGI scripts included with Apache Web server



This attack, which allows an attackers to cover her tracks, typically occurs after root compromise

Rootkits began as collections of "hacked replacements" for common unix commands (like ls) that behaved like legitimate commands they replaced, except for hiding an attacker's file, directories or processes. For example, if an attacker was able to replace a compromised linux system ls command with a rookkit version of ls, then anyone executing ls command to view files and directories would see everything except that attacker's files and directories

Since the advent of loadable kernel modules (LKMs), rootkits have more frequently taken the form of LKMs. An LKM rootkit covers the tracks of attackers in kernal space and intercept system calls pertaining to any user's attempts to view the intruder's resources.

Luckily, LKM rootkit is not completes invisible

Many traditional LKM rootkits are detectable with scrip chkrootkit, available at www.chkrootkit.org.

However attacker gets far enough, we have to wipe and rebuild system

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

Means Of Evaluating Metrics

- The severity of security vulnerabilities, derived from the following metrics:
  - Damage potential (how much damage is possible?)
  - Exploitation potential (how easy is it to exploit?)
  - Exposure potential (what kind of access is necessary to exploit the vulnerability?)
- Overall severity risk
  - Given the above three factors, the overall severity risks range from minimal to catastrophic
  - Example: Microsoft often ranks a security flaw as *Critical* for all Windows operating systems *except* Windows Server 2003, in which case it is ranked at the lower value, *Important*
    - The reason given for this difference is that Windows Server 2003 has different default settings than other versions of Windows

Example: Microsoft Security Bulletin MS08-067 – Critical

Vulnerability Information

⊟ **Severity Ratings and Vulnerability Identifiers**

Vulnerability Severity Rating and Maximum Security Impact by Affected Software		
Affected Software	**Server Service Vulnerability - CVE-2008-4250**	**Aggregate Severity Rating**
Microsoft Windows 2000 Service Pack 4	**Critical** Remote Code Execution	**Critical**
Windows XP Service Pack 2 and Windows XP Service Pack 3	**Critical** Remote Code Execution	**Critical**
Windows XP Professional x64 Edition and Windows XP Professional x64 Edition Service Pack 2	**Critical** Remote Code Execution	**Critical**
Windows Server 2003 Service Pack 1 and Windows Server 2003 Service Pack 2	**Critical** Remote Code Execution	**Critical**
Windows Server 2003 x64 Edition and Windows Server 2003 x64 Edition Service Pack 2	**Critical** Remote Code Execution	**Critical**
Windows Server 2003 with SP1 for Itanium-based Systems and Windows Server 2003 with SP2 for Itanium-based Systems	**Critical** Remote Code Execution	**Critical**
Windows Vista and Windows Vista Service Pack 1	**Important** Remote Code Execution	**Important**
Windows Vista x64 Edition and Windows Vista x64 Edition Service Pack 1	**Important** Remote Code Execution	**Important**
Windows Server 2008 for 32-bit Systems*	**Important** Remote Code Execution	**Important**
Windows Server 2008 for x64-based Systems*	**Important** Remote Code Execution	**Important**
Windows Server 2008 for Itanium-based Systems	**Important** Remote Code Execution	**Important**

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

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
- I query CERT vulnerabilities notes database for "Windows" and "Linux" keywords to examine metrics for the 40 most recent reported vulnerabilities
- A number between 0 and 180 expresses the final metric, where the number 180 represents the most serious vulnerability
- The ranking is not linear
  - In other words, a vulnerability ranked 100 is not twice as serious as a vulnerability ranked at 50
- CERT considers any vulnerability with a score of 40 or higher to be serious enough to be a candidate for a special CERT Advisory and US-CERT technical alert

CERT: Query Result for Keyword "Microsoft"

US-CERT
UNITED STATES COMPUTER EMERGENCY READINESS TEAM

Search Results

Metric	ID	Date Public	Name
78	VU#117394	03/17/2003	Buffer Overflow in Core Microsoft Windows DLL
47.04	VU#309739	08/12/2008	Microsoft Color Management System (MSCMS) module remote code execution
45.56	VU#567620	11/11/2003	Microsoft Windows Workstation service vulnerable to buffer overflow when sent specially crafted network message
45.24	VU#274496	10/12/2004	Microsoft Excel parameter validation error
41.04	VU#492515	01/14/2010	Microsoft Internet Explorer HTML object memory corruption vulnerability
37.93	VU#390044	06/13/2006	Microsoft JScript memory corruption vulnerability
37.46	VU#222044	06/03/2003	Microsoft Windows Media Player fails to properly launch URLs based on Dynamic HTML (DHTML) behaviors
28.43	VU#238064	12/12/2006	Microsoft Remote Installation Service Writable Path Vulnerability
27	VU#641460	04/11/2006	Microsoft Windows fails to properly handle COM objects
25.65	VU#511577	02/13/2007	Microsoft Malware Protection Engine fails to properly process a specially crafted PDF File
25.51	VU#252146	01/10/2006	Microsoft Outlook and Microsoft Exchange TNEF decoding buffer overflow
25.24	VU#543907	05/13/2008	Microsoft Office fails to properly handle specially crafted Rich Text Format files
24.3	VU#155563	04/08/2008	Microsoft Office Project vulnerable to remote code execution via specially crafted Project file
23.72	VU#378160	12/14/2004	Microsoft Windows Internet Naming Service (WINS) contains a buffer overflow
23.28	VU#287067	10/04/2001	Microsoft PowerPoint and Excel fail to properly detect macros thereby automatically executing malicious code via crafted document (MS01-050)
22.57	VU#810772	11/14/2006	Microsoft Agent fails to properly handle specially crafted .ACF files
22.27	VU#303452	05/09/2006	Microsoft Exchange fails to properly handle vCal and iCal properties
22.03	VU#271860	01/09/2007	Microsoft Outlook fails to properly parse Office Saved Searches (.oss) files
19.23	VU#176556	10/10/2006	Microsoft Office fails to properly parse malformed records
19.23	VU#234900	10/10/2006	Microsoft Office fails to properly parse malformed strings

CERT: Query Result for Keyword "Microsoft" (continued)

19.23	VU#234900	10/10/2006 Microsoft Office fails to properly parse malformed strings
19.23	VU#807780	10/10/2006 Microsoft Office fails to properly parse malformed Smart Tags
17.95	VU#534276	10/10/2006 Microsoft Office fails to properly parse malformed chart records
16.4	VU#901584	12/12/2006 Microsoft Windows SNMP Memory Corruption Vulnerability
12.65	VU#484814	05/11/2004 Microsoft Help and Support Center (HCP) fails to properly validate HCP URLs
12.48	VU#918652	07/23/2003 Microsoft SQL Server becomes unresponsive when large packet is sent to specific named pipe
10.6	VU#139150	01/13/2004 Microsoft Data Access Components (MDAC) contains buffer overflow
8.85	VU#279323	07/24/2002 Microsoft SQL Server contains buffer overflows in several Database Consistency Checkers
7.89	VU#411516	08/08/2006 Microsoft Windows kernel fails to properly manage exception handling
7.48	VU#869640	07/13/2004 Microsoft Outlook Express fails to properly validate malformed e-mail headers
6.37	VU#739844	02/14/2006 Microsoft Windows Korean Input Method Editor vulnerability
4.72	VU#581603	07/24/2001 Microsoft Services for UNIX Network File System (NFS) server is vulnerable to denial of service via memory leak
4.09	VU#617436	01/09/2007 Microsoft Outlook vulnerable to DoS via a malformed email message
3.09	VU#866305	05/08/2007 Microsoft Cryptographic API Component Object Model Certificates ActiveX control contains a remote code execution vulnerability
2.25	VU#447569	04/09/2003 Microsoft Windows Virtual Machine (VM) ByteCode Verifier fails to properly check Java applets for malicious code
1.68	VU#963628	02/14/2006 Microsoft PowerPoint may disclose information in the Temporary Internet Files Folder
0.9	VU#449438	09/14/2004 Microsoft Office WordPerfect 5.x Converter contains a buffer overflow vulnerability
0.6	VU#899713	08/27/2002 Microsoft Word and Excel documents allow local file reading by via embedded fields
0.39	VU#348953	07/10/2007 Microsoft Windows Active Directory fails to properly validate client sent LDAP requests
0.37	VU#768440	09/11/2007 Microsoft Windows Services for UNIX privilege escalation vulnerability
0	VU#159484	08/08/2006 Microsoft Visual Basic for Applications buffer overflow

CERT: Query Result for Keyword “Linux”

US-CERT
UNITED STATES COMPUTER EMERGENCY READINESS TEAM

Search Results

Metric	ID	Date Public	Name
26.52	VU#981222	02/18/2004	Linux kernel mremap(2) system call does not properly check return value from do_munmap() function
25.98	VU#176888	03/26/2001	Linux kernel contains race condition via ptrace/procfs/execve
23.62	VU#301156	12/01/2003	Linux kernel do_brk() function contains integer overflow
20.84	VU#362983	10/19/2010	Linux kernel RDS protocol vulnerability
20.05	VU#960877	11/04/2000	Red Hat linux restore uses insecure environment variables allowing root compromise
18.87	VU#153653	10/31/2000	Linux dump uses environment variables insecurely, allowing for root compromise
15.82	VU#134025	02/07/2003	kernel-utils sets insecure permissions on "uml_net" utility
14.42	VU#415734	03/10/2004	F-Secure Anti-Virus for Linux fails to properly detect Sober.D virus
14.25	VU#628849	03/17/2003	ptrace contains vulnerability allowing for local root compromise
13.53	VU#490620	01/05/2004	Linux kernel do_mremap() call creates virtual memory area of 0 bytes in length
13.38	VU#258564	07/14/2003	Linux NFS utils package "rpc.mountd" contains off-by-one buffer overflow in xlog() function
12.57	VU#361181	09/26/2005	Helix Player format string vulnerability
11.81	VU#973654	06/14/2004	Linux kernel fails to properly handle floating point signals generated by "fsave" and "frstor"
10.96	VU#405955	07/29/2002	util-linux package vulnerable to privilege escalation when "ptmptmp" file is not removed properly when using "chfn" utility
10.79	VU#399883	07/26/2001	Linux groff utility pic contains format string vulnerability
10.54	VU#653160	09/14/2004	Mozilla Linux installer does not properly set file permissions
9.21	VU#698640	02/08/2001	Linux kernel does not properly validate user input via sysctl for negative value
8.9	VU#971179	06/27/2001	UUCP package contains multiple buffer overflows via long string of characters sent as command line argument
8.77	VU#685461	03/27/2005	Linux kernel Bluetooth support fails to properly bounds check "protocol" variable
7.03	VU#527736	04/11/2001	mkpasswd uses weak random number generator

CERT: Query Result for Keyword "Linux" (continued)

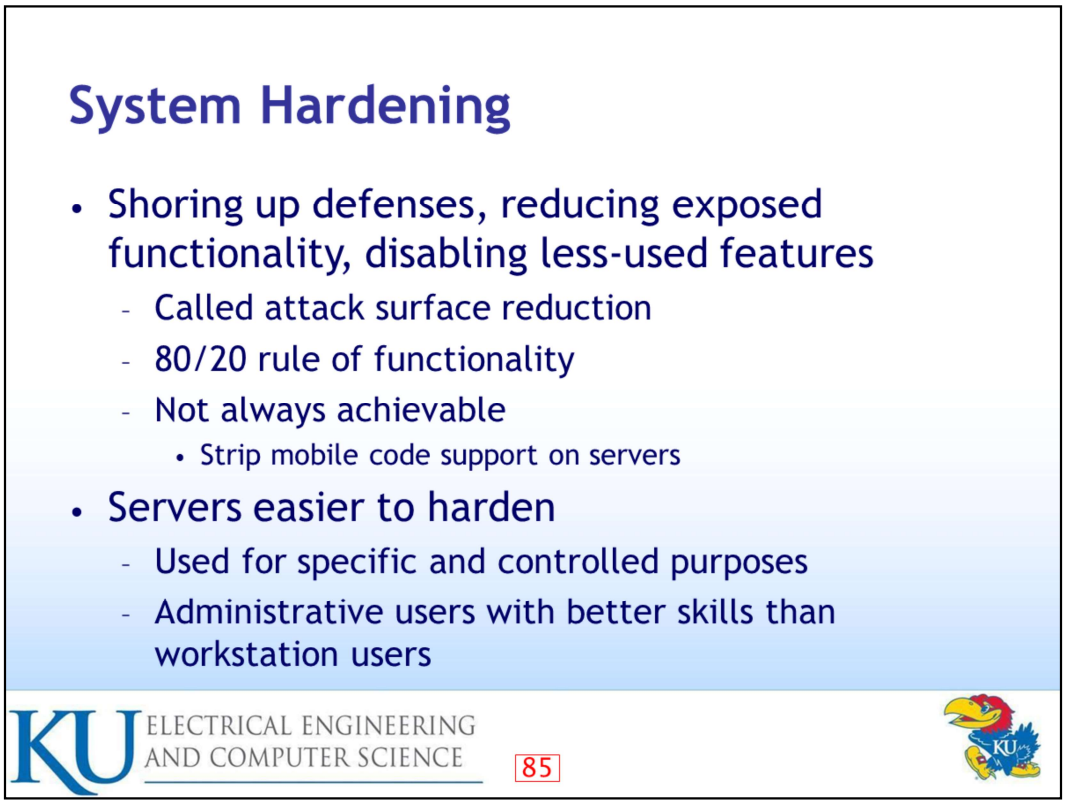
7.03	VU#527736	04/11/2001 mkpasswd uses weak random number generator
5.73	VU#230307	02/25/2002 Linux kernel netfilter IRC DCC helper module creates overly permissive firewall rules
4.3	VU#995038	12/23/2004 Debian Linux Netkit telnetd-ssl contains a format string vulnerability
3.71	VU#920689	03/12/2007 Linux Kernel vulnerable to DoS via the ipv6_getsockopt_sticky() function
3.15	VU#898480	11/20/2001 MandrakeSoft Mandrake Linux Apache default configuration sample programs disclose server information
3.03	VU#25701	07/27/2000 Linux gpm daemon allows arbitrary file removal
3.03	VU#35842	07/03/2000 man 'makewhatis' insecurely uses /tmp
2.95	VU#337238	01/16/2004 Red Hat Enterprise Linux kernel-2.4.21 does not perform adequate checking of eflags when in 32-bit ptrace emulation mod
2.69	VU#681569	05/23/2006 Linux Kernel may fail to properly handle SNMP packets
2.65	VU#426456	01/10/2001 gpm creates temporary files insecurely
2.64	VU#24140	03/27/2000 Linux kernel IP Masquerading "destination loose" (DLOOSE) configuration passes arbitrary UDP traffic
1.82	VU#249579	02/10/2001 klogd does not adequately handle NULL byte when parsing text using LogLine( )
1.39	VU#801526	02/03/2004 util-linux login program discloses sensitive information
1.36	VU#471084	06/09/2003 Linux kernel IP stack incorrectly calculates size of an ICMP citation for ICMP errors
0.48	VU#981134	08/25/2004 Linux kernel USB drivers do not initialize kernel memory properly
0.21	VU#913704	11/20/2001 MandrakeSoft Mandrake Linux Apache default configuration enables directory indexing
0.21	VU#927256	11/20/2001 MandrakeSoft Mandrake Linux Apache default configuration enables Perl ProxyPass server on 8200/tcp
0.18	VU#455323	06/17/2002 Mandrake Security may make unexpected system modifications
0.06	VU#300368	08/29/2006 X.Org fails to check for setuid failure on Linux systems
0	VU#493966	02/12/2004 Libxml2 URI parsing errors in nanohttp and nanoftp
0	VU#717844	07/12/2006 Linux kernel fails to properly handle malformed SCTP packets

CERT: Evaluation of Query Results for "Microsoft" and "Linux"

- CERT web search capabilities do not produce perfectly desirable results in terms of granularity or longevity
  - Especially True for Linux
    - The "Linux" search results include a number of Oracle security vulnerabilities that are common to Linux, UNIX, and Windows
  - In Top 40 CERT results for "Microsoft",
    - Top entry containing the severity metric of 78
    - 5 entries have a severity rating of 40 or greater
  - In Top 40 CERT results for Linux
    - Top entry containing the severity metric of 26.52
    - None other entry have a severity rating 27 or greater
- Note that CERT results only reflect how Windows security flaws tend to be far more frequently severe than those of Linux
  - These results cannot be expected to mirror our own analysis of recent vulnerability patches

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion



Process of hardening is process of shoring up defenses, reducing the amount of functionality exposed to untrusted users and disabling less-used features.

In microsoft it called Attack surface reduction.

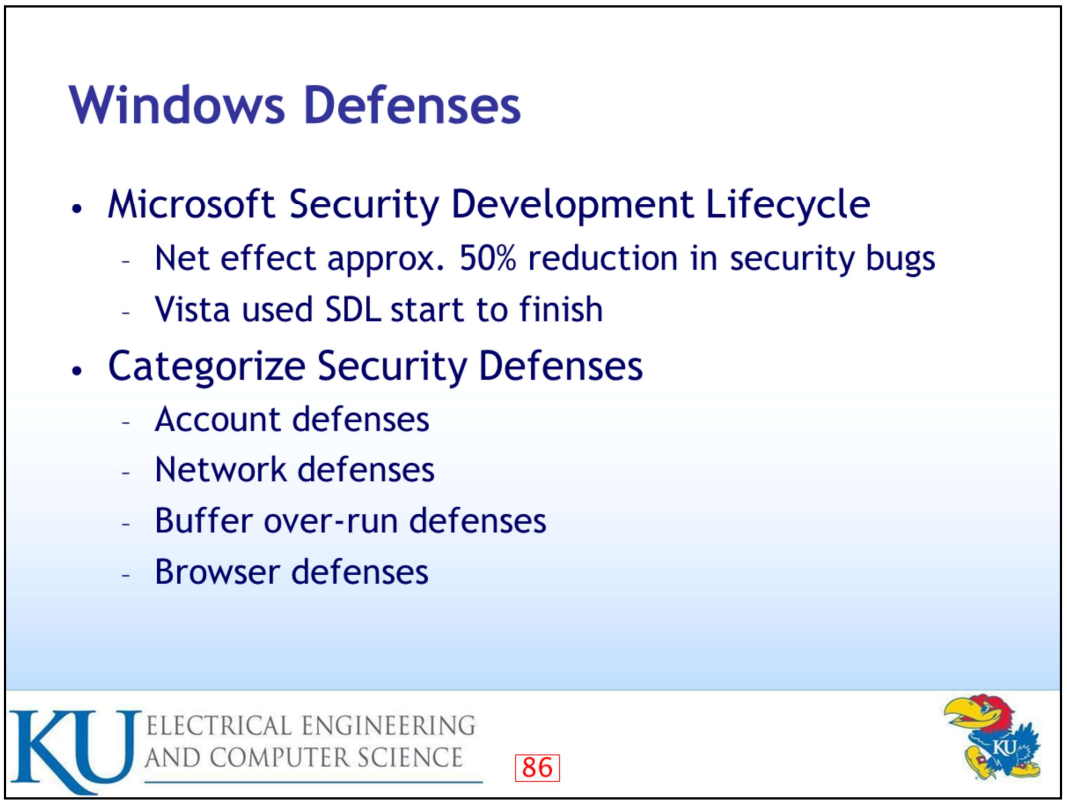
The concept is simple: apply 80/20 rule to features. If the feature is not used by 80% of population, then feature should be disabled by default.

While this was goal it was not always achievable, as disabling too many features makes product unusable for nontechnical users.

Servers easier to harden because

Used for specific and controlled purposes

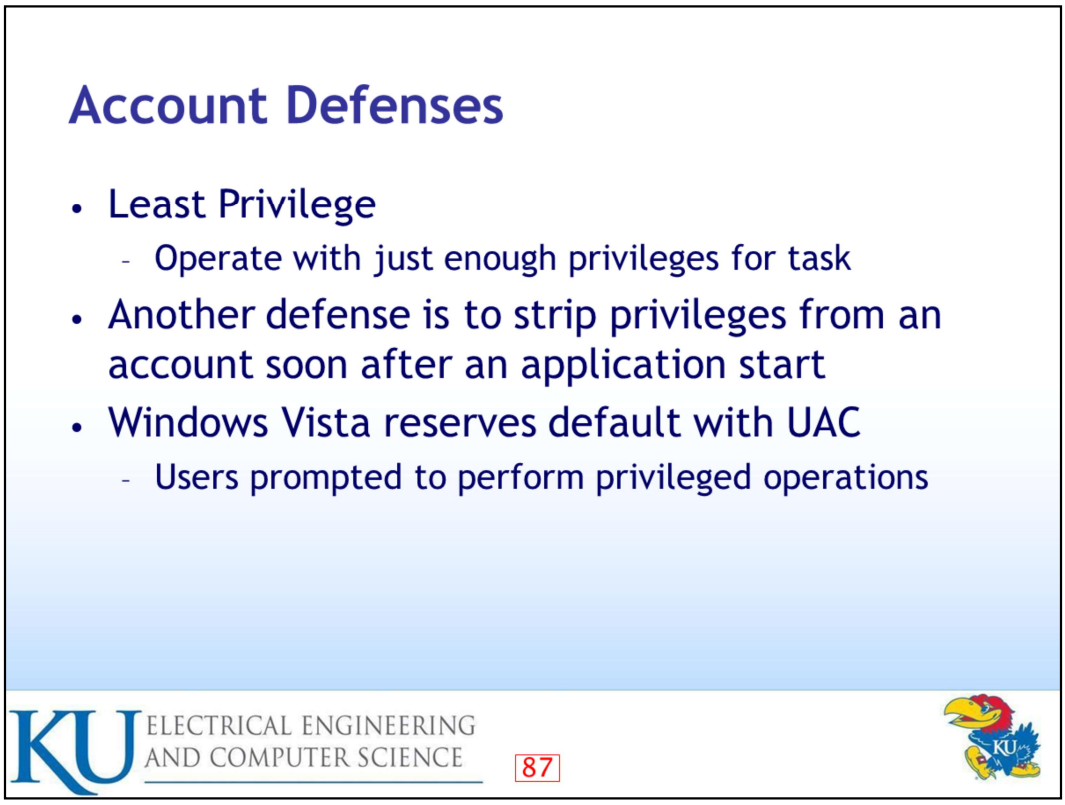
Administrative users with better skills than workstation users



After 2001, microsoft decide to change its software development process to better accommodate secure design, coding, testing and maintenance requirements, with one goal in mind: reduce the number of vulnerabilities in all microsoft products.

This process improvement is called the security development lifecycle.

The core SDL requirements includes mandatory security education, secure design requirements, attack surface analysis and reduction…etc.



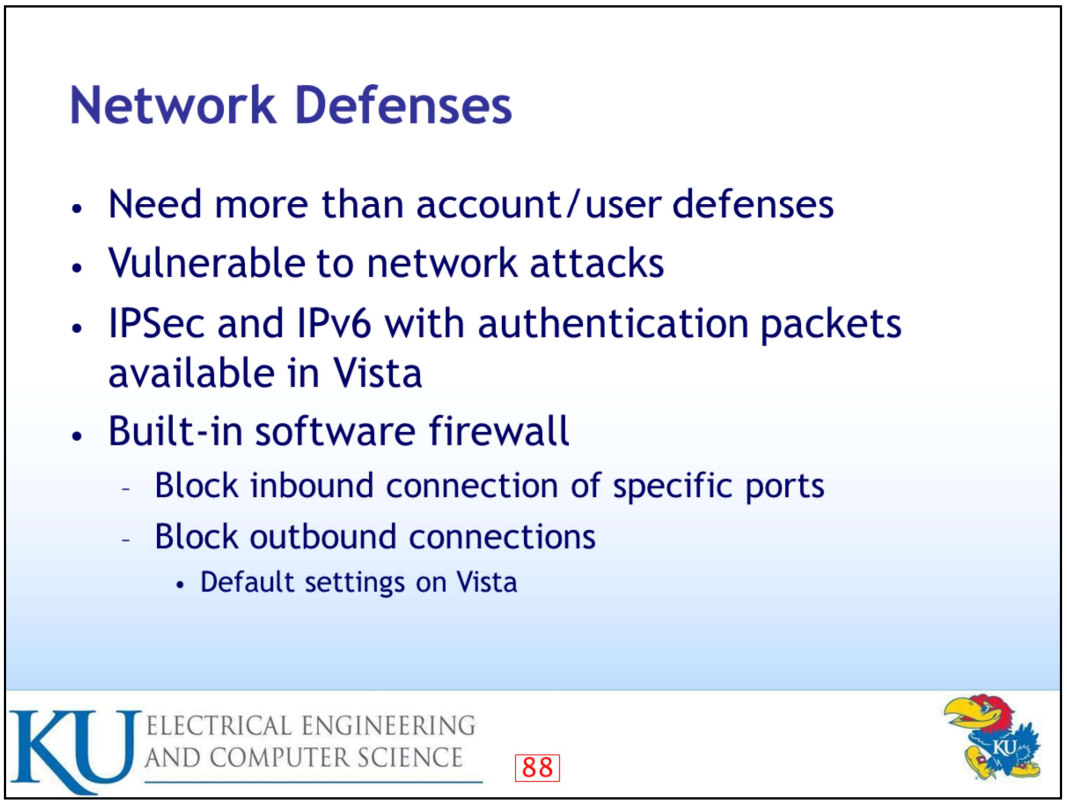
The principle of least privilege dictates that users should operate with just enough privilage to get the tasks done, and no more.

Historically, windows xp users operated by default as members of the local administrators group, this was simply done for application compatibility reasons. In some cases(if applications run on windows 95,98 may not run on windows xp without administrator privilege) a windows xp user running as a “standard user” could run into some errors.

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



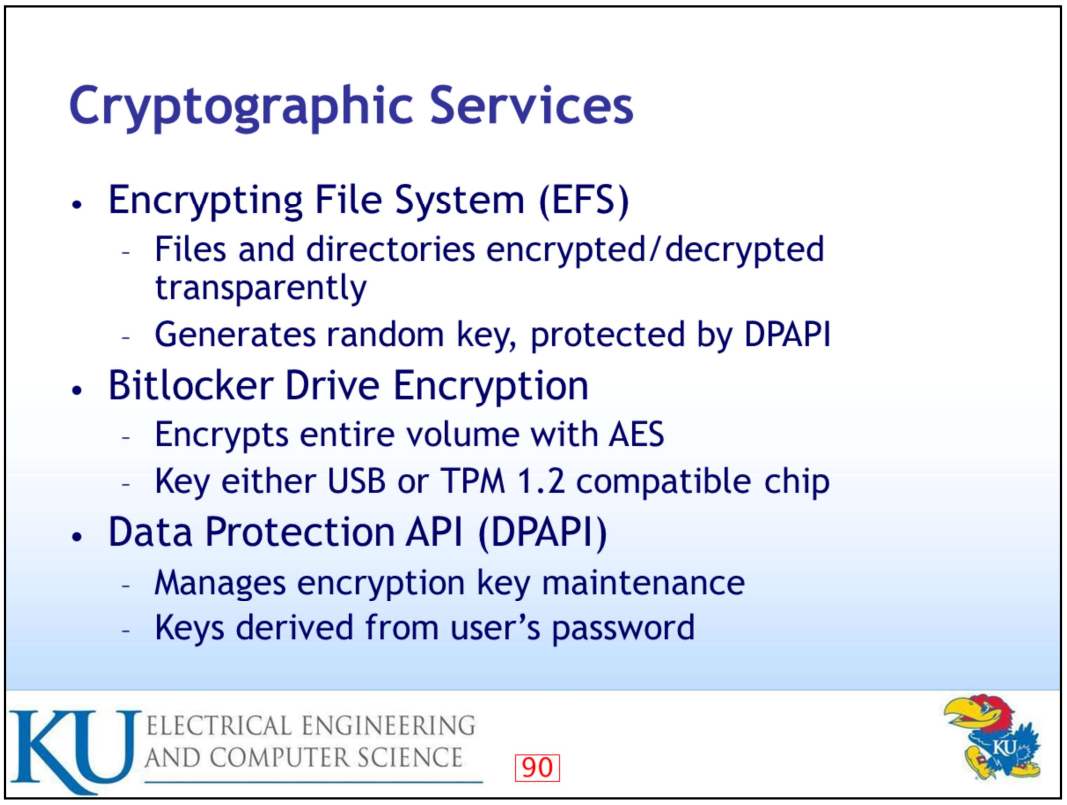
There is one big problem with defenses that focus on user and user accounts: they do nothing to protect computers from low-level network attacks. So we need network defences.

Windows offers many network defenses, most notably native IPSec and IPv6, and bi-directional firewall.

See book for Detail discussion

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



Windows includes full-fledged cryptographic defenses such as EFS, Bitlocker, data protection APIs

ETS allows files and directories to be encrypted and decrypted transparently for authorized users.

At a very high level, EFS works by generating a random encryption key and storing that key, encrypted using the user's encryption key. This key is protected by data protection api in windows and the key used by DPAPI is derived form the user's password.

Thus , Data protection API allows users to encrypt and decrypt data transparently.

Bitlocker encrypts the entire volume with using AES, and the encryption key is stored either on a USB drive or within a Trusted platform module (TPM) chip on the computer motherboard.

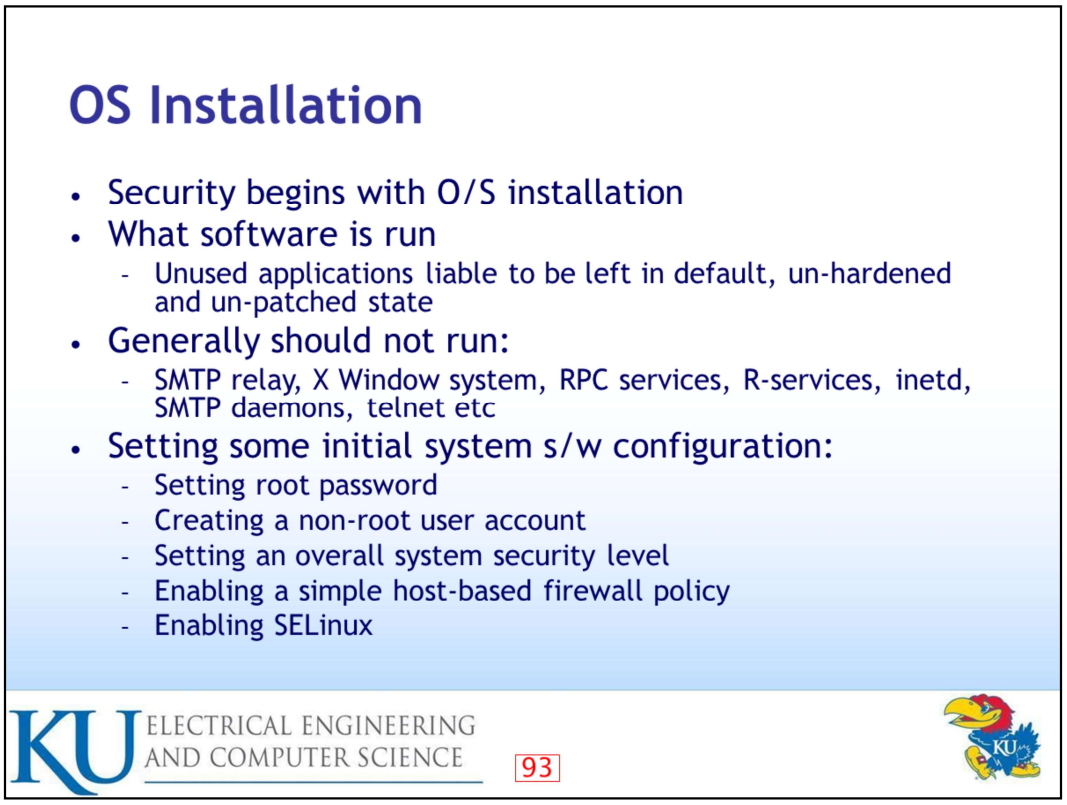
When booting a system that requires the USB device, the device must be present so the keys can be read by the computer, after which Bitlocker decrypts the hard drive on the fly, with no perceptible performance degradation.

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

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



Linux system security begins at operating system installation time: one of the most critical, system impacting decisions a system admin makes is what software will run on the system.

What software you should not run?

SMTP relay, X Window system, RPC services, R-services, inetd, SMTP daemons, telnet etc

Additionally, Linux installation utilities also perform varying amounts of initial system and software configuration such as

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

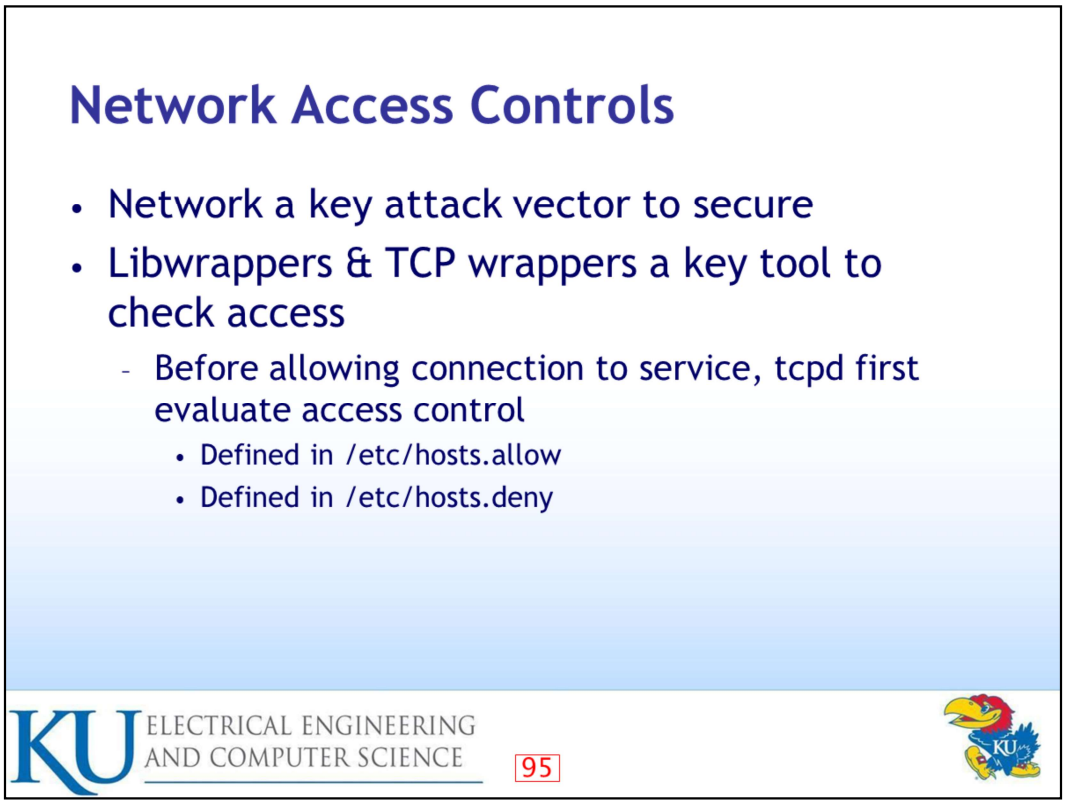
KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 94

Carefully selecting what gets installed on linux system is an important but also they must also kept up to date with security patches and configure securely

Patch rat-race : there will always be software vulnerabilities that attackers are able to exploit for some period of time before vendor issue patches for them.

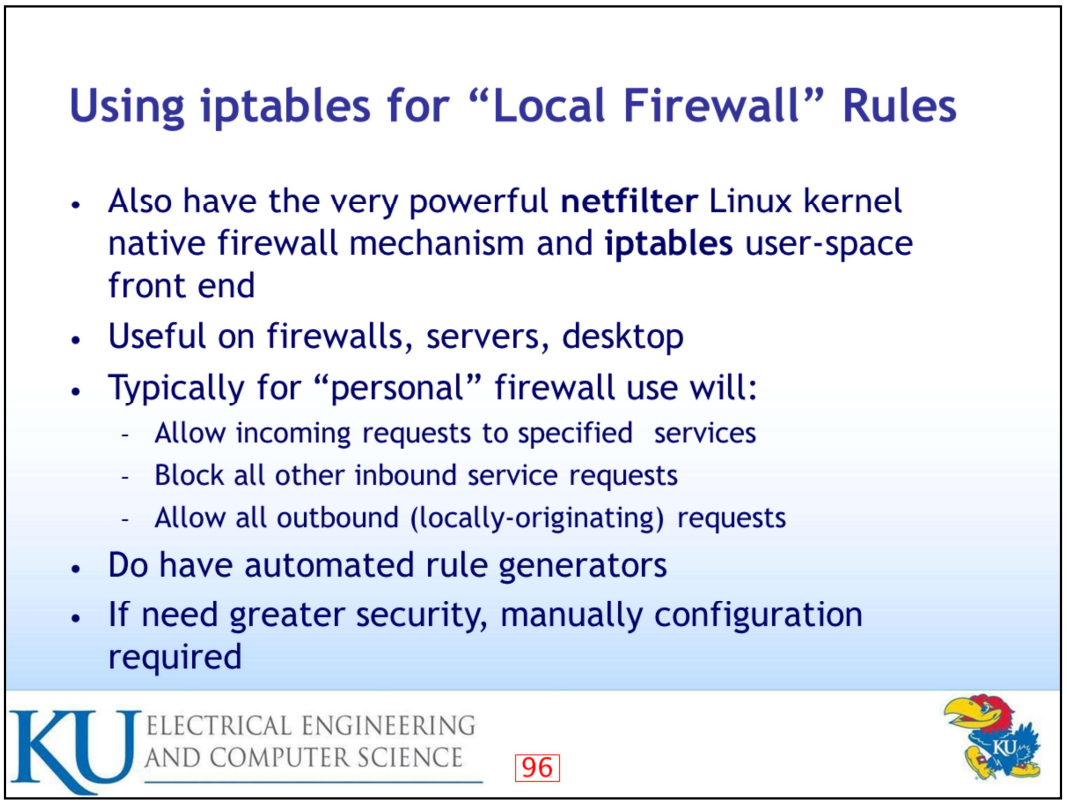
Good news is modern linux distributions includes tools for automatically download and install security updates

Should not run automatic updates on change-controlled systems without testing because it may introduce instability



One of the most important attack vectors in linux threats is the network

Network-level access control (that is, controls that restrict access to local resources based on ip addresses of the networked system attempting to access them) are there for import tool in linux security



Linux kernel's native firewall mechanism, netfilter is powerful tool because netfilter is commonly referred to by the name of its user-space frontend, iptables.

Useful on firewalls, servers, desktop

Typically for "personal" firewall use will:

Allow incoming requests to specified services

Block all other inbound service requests

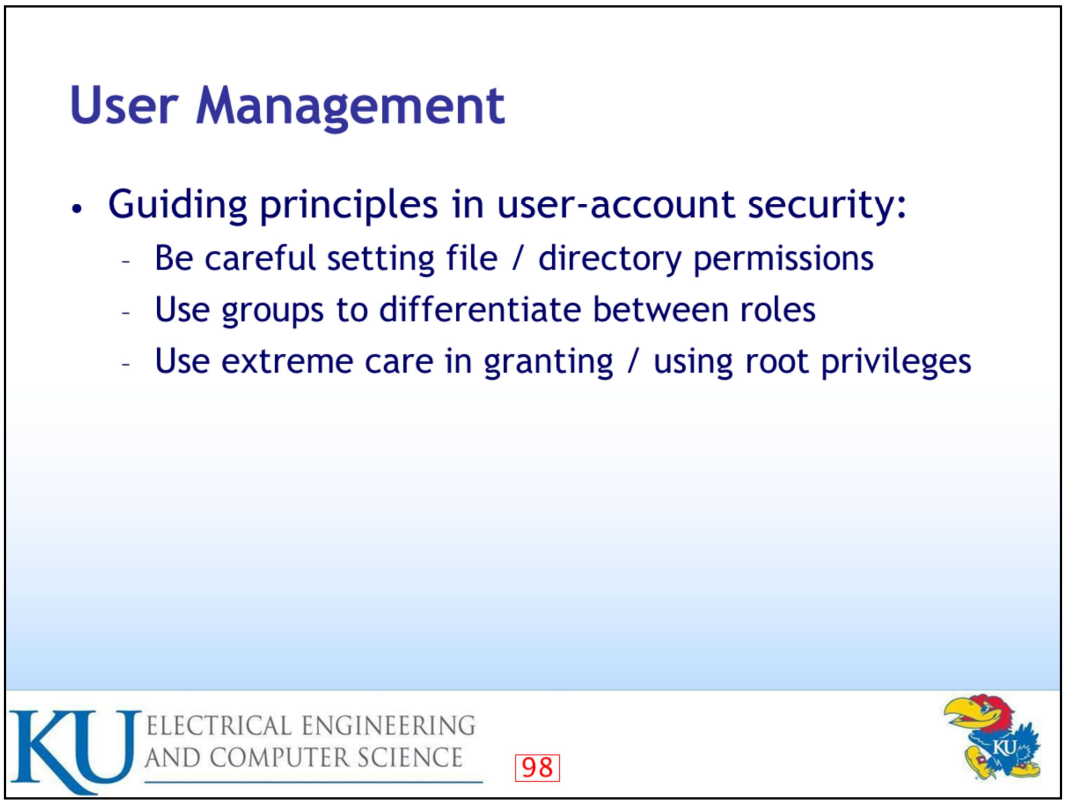
Allow all outbound (locally-originating) requests

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 users privileges
- Non-privileged user account
  - Less scope of being exploited
- Growing Linux popularity means growing exploits
- Hence antivirus software will be more important
  - Various commercial and free Linux A/V



These guidelines user need to follow for security

Be careful setting file / directory permissions

Use groups to differentiate between roles

Use extreme care in granting / using root privileges

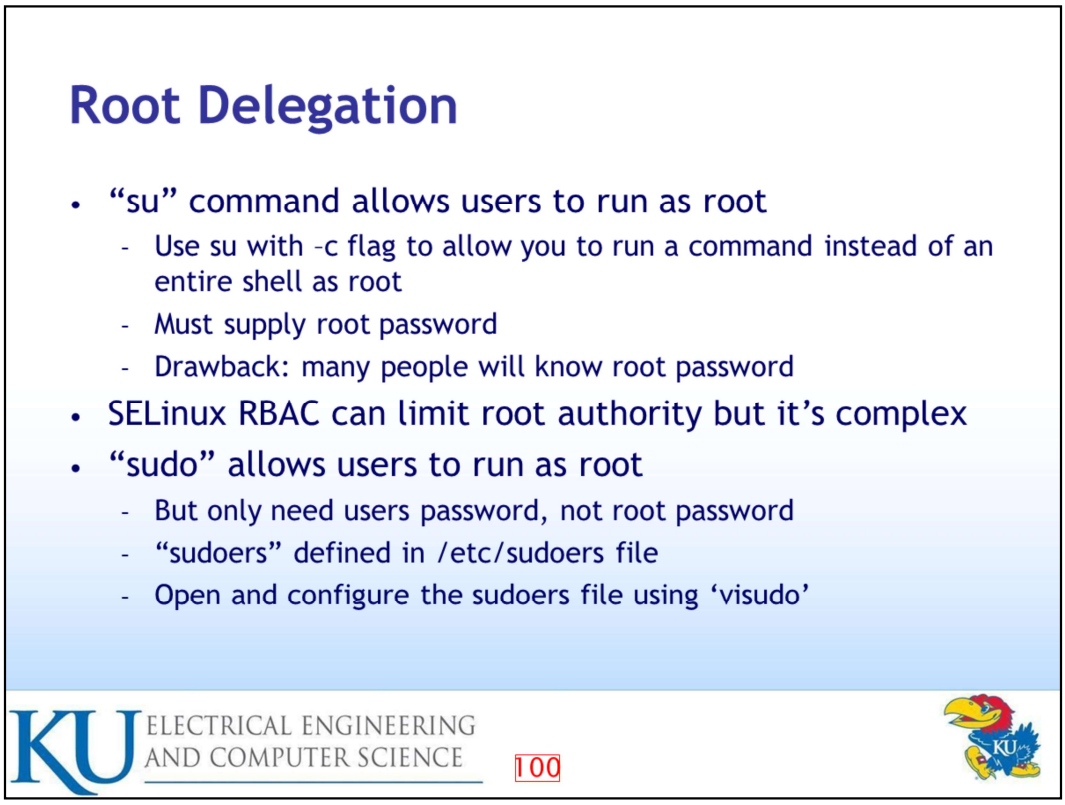
Password Aging

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

KU ELECTRICAL ENGINEERING AND COMPUTER SCIENCE 99

Password againg is ste globally in the files /etc/login.defs and /etc/default/useradd but these settings are only applied when new user accounts are created

To modify password lifetime for existing account use the ‘change’ command



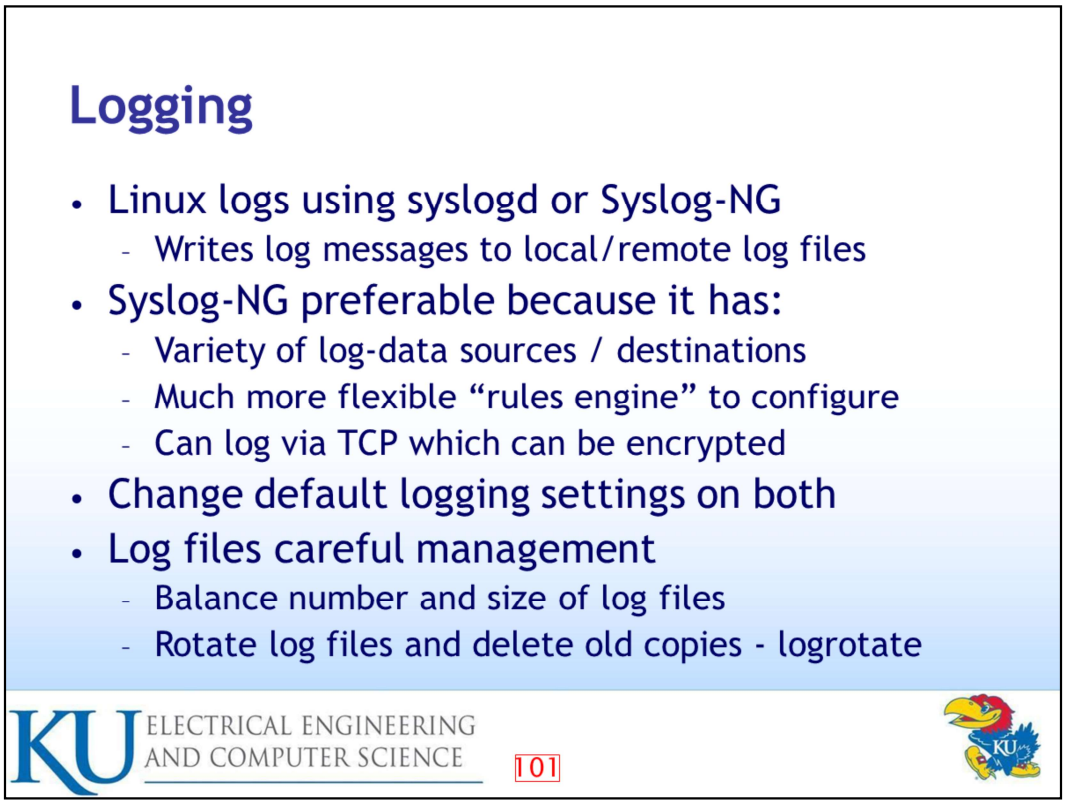
"su" command allows users to run as root

Su with –c flag allows you to specify a single command to run as root rather than an entire shell session.

However, this require you to enter the root password, so required root password sharing. So only good for small number of people.

SELinux RBAC can limit root authority but it's complex

Middle ground solution is sudo command. Sudo is configured via the file/etc/sudoers, but you should not edit this file directly. Rather you should used command visudo, which opens a special vi session.



Logging isn't a proactive control because logs only tell you about bad things that have already happened.

But effective logging helps ensure that the event of a system failure, sysystm admins can more quickly and accurately identify cause therefore effectively focus on recovery and remediation efforts

Linux logs using syslogd or Syslog-NG

Syslog-NG preferable because it has:

Variety of log-data sources / destinations

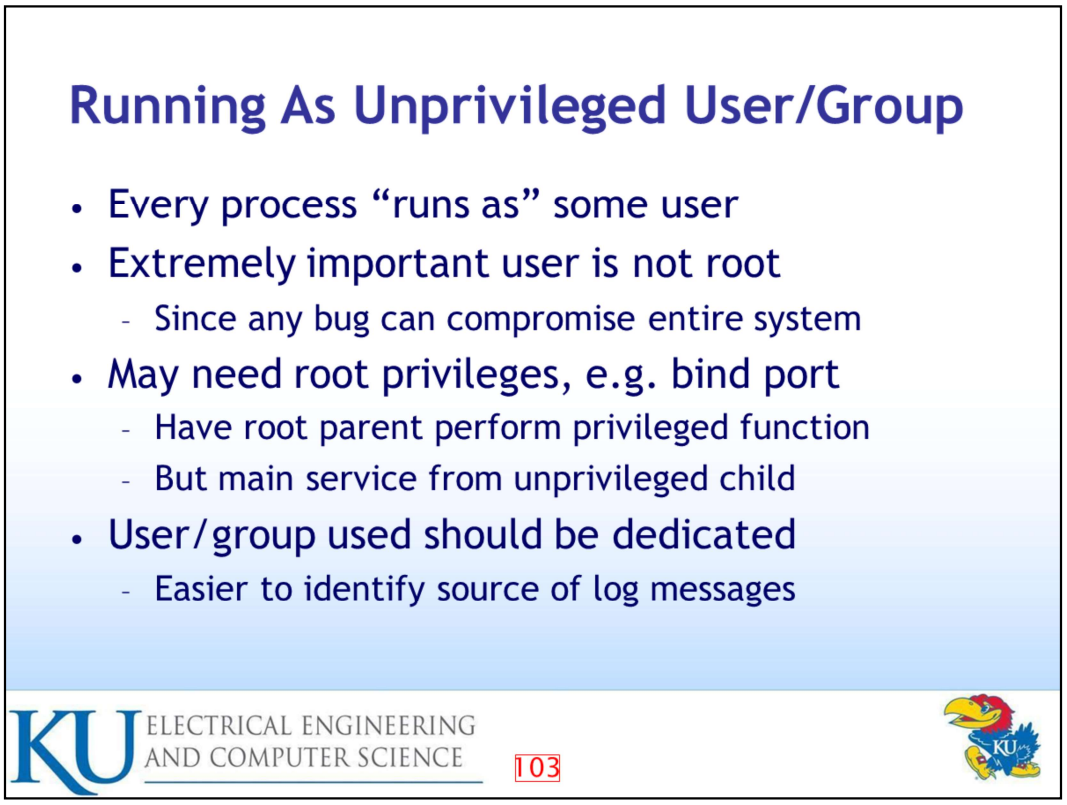
Much more flexible "rules engine" to configure

Can log via TCP which can be encrypted

Log files require careful management

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



Every process “runs as” some user

Extremely important user is not root

Since any bug can compromise entire system

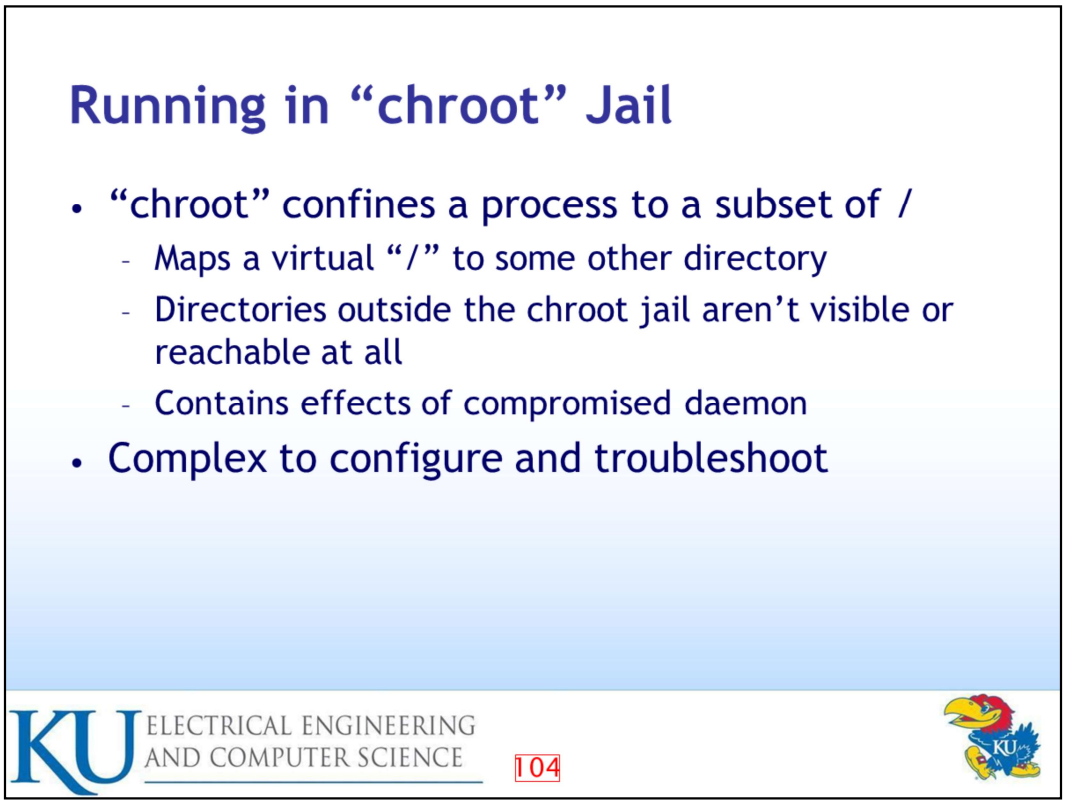
May need root privileges, e.g. bind port

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



If an FTP daemon serves files from a particular directory says, /srv/ftp/public, there shouldn’t be any reason for that daemon to have access to the rest of the file system

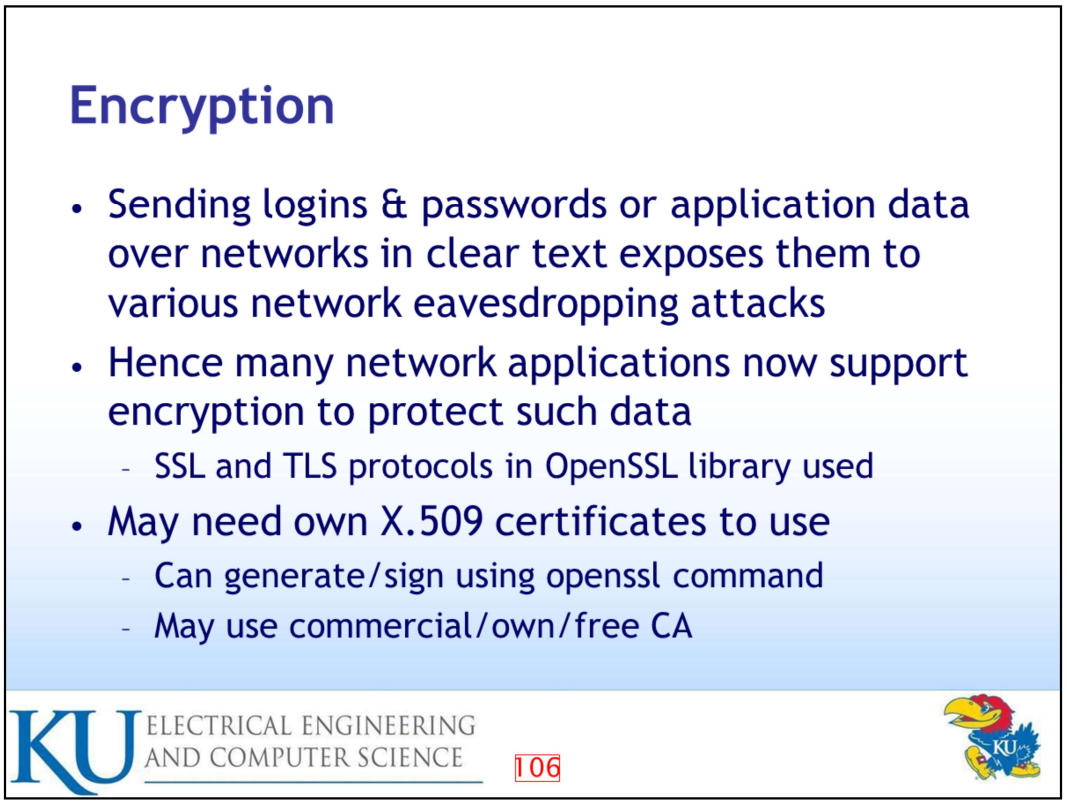
Chroot system call confines a process to some subset of /, that is, it maps a virtual “/” to some other directory. We call this directory to which we restrict the daemon a chroot jail.

To the “chrooted” daemon, everyting in the chroot jail appears to actually be in /. Things in directories outside the chroot jail aren’t visible or reachable at all.

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



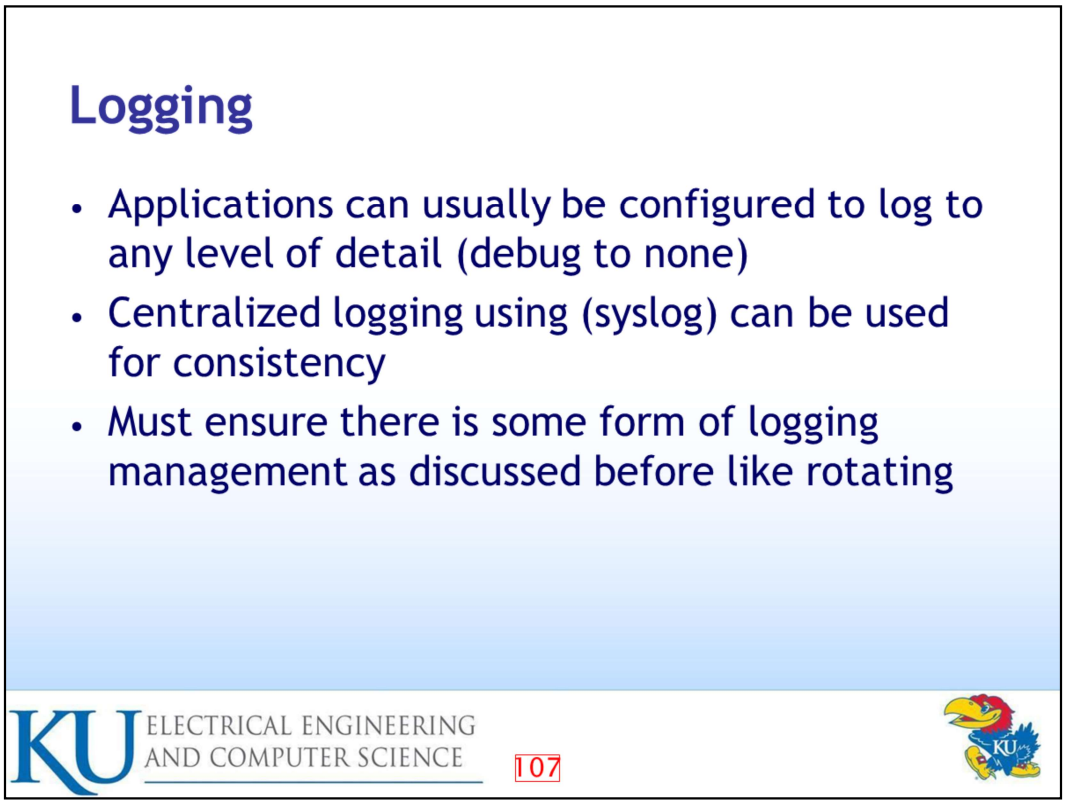
Sending logins & passwords or application data over networks in clear text exposes them to various network eavesdropping attacks

Hence many network applications now support encryption to protect such data SSL and TLS protocols in OpenSSL library used that require to use X.509 digital certificates.

Thus, need own X.509 certificates to use

can generate/sign using openssl command

May use commercial/own/free CA



Most applications can configured to log to whatever level of detail you want, ranging from “debugging” (maximum details) 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

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

OS Security Capabilities: Linux vs. Windows

- A qualitative assessment of operating system security is subjective and your "mileage may vary" based on present and past experience
- A true comparison between Windows and Linux on the values of the inherent security of each operating system is hard to obtain, and the matter is extremely contentious among both security professionals and computer hobbyists

OS Security Capabilities: Linux vs. Windows (continued)

- Because there are many more Windows systems in the world, there are simply more targets available for attack
  - Windows a richer and more attractive target for malware developers
  - Windows monoculture: because Windows systems are all tightly binary-compatible, a single successful attack can affect a large fraction of them
- The security differences between Windows and Linux is heavily debated and the security track record of both operating systems has proven Linux has fewer serious vulnerabilities
  - Also Linux derives its security from the underlying Unix design philosophy

OS Security Capabilities: Linux vs. Windows (continued)

	Windows	Linux
Malware	• As of 2009, well over 2 million malware programs target Windows • Botnets: networks of infected computers controlled by malicious persons - with more than one million computers have been witnessed • Once malicious software is present on a Windows-based system, it can sometimes be incredibly difficult to locate and remove • As such, users are advised to install and run anti-malware programs • In the event of rootkit infection, users may have to resort to reformatting the system's hard disk and re-installing Windows	•As of 2006, more than 800 pieces of Linux malware had been discovered •Some malware has propagated through the Internet. • However, in practice, reports of bonafide malware presence on Linux-based systems are extremely rare •Aniware Tools like ClamAV and Panda Security's DesktopSecure for Linux do exist

OS Security Capabilities: Linux vs. Windows (continued)

Open Vs. Closed	•Claims its platform is more secure because of a comprehensive approach to security using the Security Development Lifecycle •However, because Windows is closed-source, only Microsoft-employed programmers can fix bugs •Recent Windows versions have some security vulnerabilities detected	•Claims its platform is more secure because all of its code is reviewed by so many people that bugs are detected •Anyone with programming experience is free to fix bugs and submit them for inclusion in future releases and updates •However such an approach has indeed produced several vulnerabilities, although this is a rare case

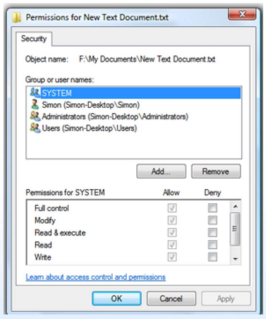
OS Security Capabilities: Linux vs. Windows (continued)

Response Speed	•There are claims that closed source offers a faster and more effective response to security issues •However, critical bug fixes are released only once a month after extensive programming and testing, and certain bugs have been known to go un-patched for months or even years	•Bugs can be fixed and rolled out within a day of being reported (often within hours), though usually it takes a few weeks before the patch is available on all distributions

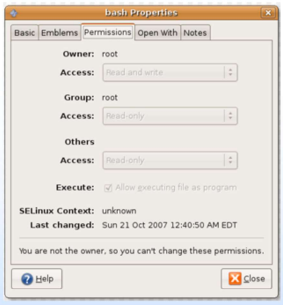
OS Security Capabilities: Linux vs. Windows (continued)

User Accounts	•In Windows Vista, all logged-in sessions run with standard user permissions, preventing malicious programs from gaining total control of the system •Processes that require administrator privileges can be run using the User Account Control framework. •For standard users, this presents a credentials dialogue that requires the password of a member of the administrators group (who are listed) •For users who are already logged in an administrator, only confirmation is necessary •The first user account created during the setup process is automatically a member of the administrators group •The majority of users did not change to an account type with fewer rights, meaning that, in Windows versions prior to the introduction of UAC, malicious programs would have full control over the system •However the security of the User Account Control is not guaranteed to prevent malicious programs and users from unlimited access in Windows	•Users typically run as limited accounts, having created both administrator (root) and at least one user account during installation •In most Linux distributions, there are commands (su, sudo) that will temporarily grant elevated permissions to processes that need it •In practice, this can be very dangerous, as any error can lead to severe damage to the system •New frameworks such as PolicyKit seek to rectify this problem • However, as of Feb. 2009, PolicyKit is not in widespread use.

OS Security Capabilities: Linux vs. Windows (continued)

Windows: File System Permissions	•The DOS based Windows ME, Windows 98, Windows 95, and previous versions of non-NT Windows only operated on the FAT file system and did not support file system permissions •Windows NT and subsequent NT-based versions of Windows use NTFS-based Access Control Lists to administer permissions, using tokens •On Windows XP and prior versions, most home users still ran all of their software with Administrator accounts, as this is the default setup upon installation •The existence of software that would not run under limited accounts and the cumbersome "Run As..." mechanism forced many users to use administrative accounts •However, few organizations have taken advantage of the richness of the Token based system of NTFS which can be applied to almost all NT operating system objects	File system permissions on a Windows Vista system. 

OS Security Capabilities: Linux vs. Windows (continued)

Linux: File System Permissions	•Linux has a traditional Unix-like "user, group, other" approach to file system permissions at a minimum •This approach is extended by Access Control Lists on some file systems •There are some specific to Linux frameworks such as AppArmor and SELinux which add even finer-grained controls over which users and programs can access certain resources or perform certain operations •Some distributions use them out of the box	File system permissions on an Ubuntu Linux system running GNOME 

Agenda

- Background
- Windows Security Architecture
- Linux Security Model
- Evaluation: Windows vs. Linux Design
- Windows Vulnerabilities
- Linux Vulnerabilities
- Means of Evaluating Metrics
- Evaluation: Windows vs. Linux Vulnerabilities
  - CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords
- System Hardening
  - Windows Defenses
  - Linux System Hardening
    - OS-Level
    - Application-Level
- OS Security Capabilities: Windows vs. Linux
- Conclusion

Conclusion

- Security is a perennial concern for IT administrators
- Managers need a framework to evaluate operating system security
- The challenge in evaluating Windows and Linux on any criteria is that there is not a single version of each operating system
- Users need to keep in mind that there are philosophical differences in the design of Linux and Windows
- The ability to make either operating system more secure varies depending on architectural design
  - The Windows operating system is designed to support applications by moving more functionality into the operating system, and by more deeply integrating applications into the Windows kernel
  - Linux differs from Windows in providing a clear separation between kernel space and user space

Questions ?

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