

## Introduction to Information Security

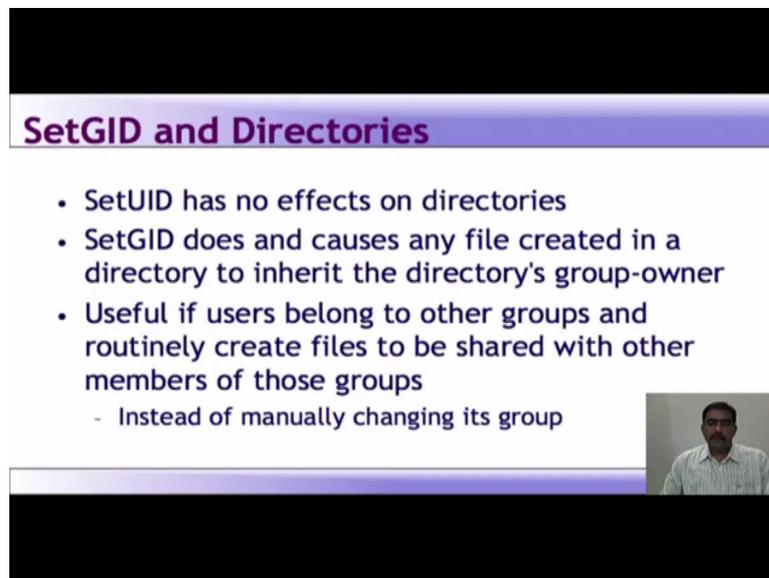
Prof. Dilip H. Ayyar

Department of Computer Science and Engineering

Indian Institute of Technology, Madras

### Lecture - 53

(Refer Time Slide: 00:10)



**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



Setgid and directories, the setuid has no effect on the directories, setgid does and causes any file created in a directory, to inherit the directory's group-owner. It is useful, if users belong to other groups, and routinely create files to be shared with other members of the group, instead of manually changing its group. So, that is the difference. So, that is the correlation, between setgid and directories.

(Refer Time Slide: 00:43)

## 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



Then, we have discussed before, about kernel space and user space kernel. Space refers to memory used by Linux kernel, and the loadable modules, like you have device drivers. The user space refers to memory, used by all other processors. Since, kernel enforces Linux dac end security, it is extremely critical to isolate the kernel from user space. For this reason, kernel space never is never swapped to the disk, only root may load and unload kernel modules.

(Refer Time Slide: 01:25)

## 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



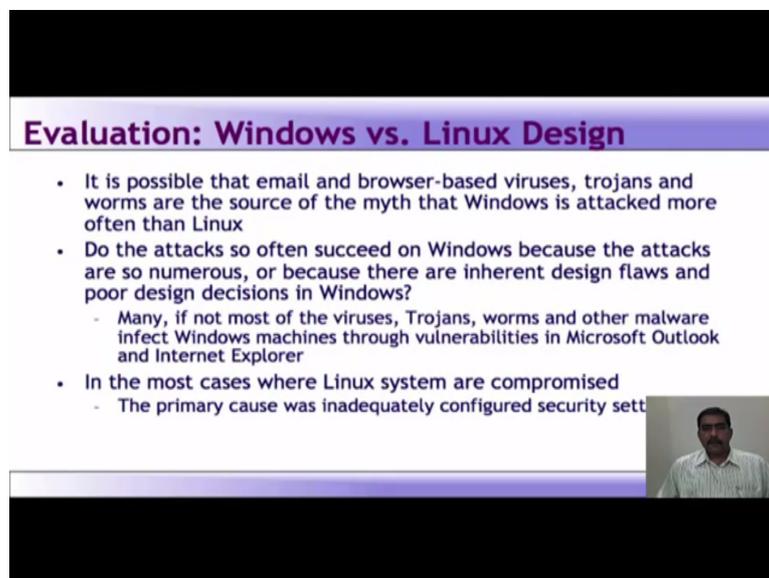
Mandatory access control, now discretionary access control module, we have seen the owner of a given system object, can set whatever access permission on that resource, he or she likes, right. So, that is dac security model. The user, who creates a file on the mac system, mac is

mandatory access control 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, the only thing is that the super user account is used for maintaining the global security policy. Day to day system administration is performed using accounts, that lack or do not have the authority to change the global security policy. Now, as a result, it is 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 the application.

So, the mac systems have been very hard to manage. Now, furthermore the more restrictive, the security controls are on a given system, it will be less convenient, for the users to operate on. So, there should be a proper balance between security, and usability. So, now the mac or mandatory access controls impose a global security policy, we have already discussed that. Novell's SUSE Linux has got AppArmor, Red Hat Enterprise has SELinux. Now, pure SELinux for high sensitivity, high availability that is what SELinux means.

(Refer Time Slide: 03:20)



**Evaluation: Windows vs. Linux Design**

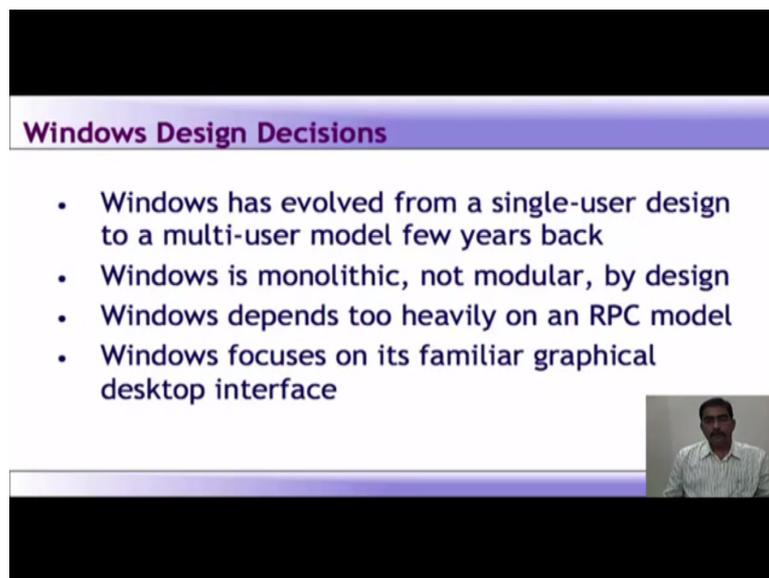
- It is possible that email and browser-based viruses, trojans and worms are the source of the myth that Windows is attacked more often than Linux
- Do the attacks so often succeed on Windows because the attacks are so numerous, or because there are inherent design flaws and poor design decisions in 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

Video inset showing a man speaking.

Now, let us look at the evaluation between windows and Linux designs. Now, it is possible that email, and browser based viruses, Trojans, and worms are a source of myth, that windows attack 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 decision, in windows.

Many if not, most of the viruses, trojans, worms, and other malware infect windows machine, through vulnerabilities in Microsoft outlook, and internet explorer application. In most cases, where the Linux systems are compromised the primary cause of was inadequately, configured security settings.

(Refer Time Slide: 04:13)



**Windows 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

A small video inset in the bottom right corner of the slide shows a man with glasses and a light-colored shirt speaking.

Windows has evolved from a single user design, to a multi-user module few years back. Windows is monolithic, means it is not modular by design. Windows depends too heavily on the rpc module, we will see, what is rpc, windows focuses on its familiar graphical desktop user interface.

(Refer Time Slide: 04:35)

**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

Now, what does it mean, that it has evolved from a single user design to a multi user model few years back. Windows has long been hampered, by its origin as a single user system. So, when it was windows a few years back, say it is excellent for a desktop. Windows was originally designed, to allow both users and applications, free access to the entire system, which means, anybody could tamper with critical system and put on an attack file.

XP was the first version, of the windows to reflect a serious effort to isolate, the users from the system, so that the users each have their own private files, and limited system privileges. This caused many legacy window applications to fail. So, what the solution was windows xp includes a compatibility mode, a mode that allows program to operate as if, they were running the original insecure single user design. Xp represented progress, but even xp could not be justifiably referred to as a true multiuser system.

(Refer Time Slide: 05:45)

## 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



Now, again when we spoke about monolithic by design, and it is not modular, but what does it mean, monolithic design means one where most features are integrated into a single unit. So, Microsoft successfully makes competing products irrelevant, or it makes the competing products irrelevant, by integrating more and more of the services they provide, into the operating system.

But, this approach creates a monster of inextricably interdependent services. Interdependencies side effect is also there, every flaw in the piece of the system is exposed through all of the service, and applications that depend on it. Then, it is unstable by nature, when you design a system, which has too many interdependencies you introduce numerous risks, when you change one piece of the system. So, it affects the entire system itself. Thus monolithic system tends to make security vulnerabilities more critical, than they need to be.

(Refer Time Slide: 06:47)

**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 itself



They depend very heavily, on the rpc module, rpc stands for remote procedure call. Simply put an rpc is what happens, when one program sends a message over the network to tell another program, to do something. Rpc's are potential security risks because, they are designed to let other computers, somewhere on the network, tell your computer what to do. So, unfortunately windows users cannot disable rpc, because windows depends on rpc, even if you computer is not connected to the network rpc is required for windows. The most common way to exploit an rpc related vulnerability, is to attack that use the rpc not rpc itself.

(Refer Time Slide: 07:45)

**Focuses on its Familiar Graphical Desktop Interface**

- Microsoft considers its familiar Windows interface as the number one benefit for using Windows Server XXXX
  - Quote from the Microsoft web site, *"With its familiar Windows interface, Windows Server XXXX3 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 XXXX 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

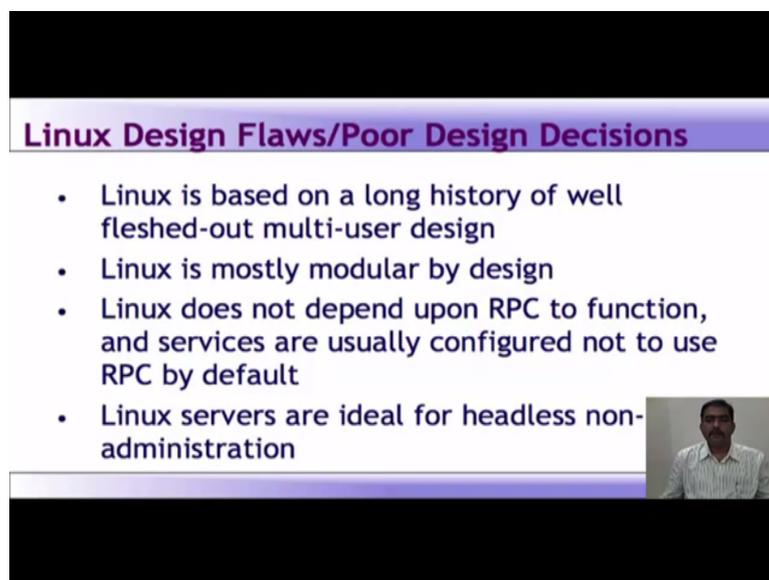


The focuses, windows focuses on its familiar graphical interface, Microsoft considers its familiar windows interface, as the number one benefit of using a windows server, whatever

version. Now, there's support from Microsoft website, with its familiar windows interface, windows server whatever version it is easy to use. New streamline result simplify, the setup of specific server roles, and routine server management tasking.

Now, by advocating this type of usage, Microsoft invites, administrators to work with the windows server at the server level itself, logged in with the local administrative privileges. This makes windows administrator, most vulnerable to security risks, because using vulnerable programs such as ie, exposes the server to security risks.

(Refer Time Slide: 08:34)



**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-administration

A small video inset in the bottom right corner of the slide shows a man with glasses and a light-colored shirt speaking.

When you come to Linux design flaws, or poor design decisions, Linux is based on a long history of well flushed out multiuser design. Linux is modular by design, Linux does not depend on rpc to function, and services are usually configures not to use rpc, by default. Linux servers are ideal for headless non-local administration, what is what does that mean.

(Refer Time Slide: 09:05)

## 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 basically have a history, of being a single user system. Therefore, it has been designed from ground up, to isolate the user from applications, the files, and the directories, that affect the entire operating system. Each user is given a user directory, where all of the, user data files and configuration files are stored. When a user runs an application, such as a word processor, the word processor runs with a restricted privilege of the user.

Now, what is important here is, Linux provides almost all capabilities, such as rendering of jpeg images, as modular jpeg images. As a result, when a word processor renders a jpeg image, that jpeg rendering functioning will run with the same restricted privileges, as the word processor itself. If there is a flaw in the jpeg rendering routine, a malicious hacker can only exploit this flaw, to gain the same privilege as the user, thus preventing potential damage.

That is the benefit of modular design, and it follows more closely the spherical energy, of an ideally designed operating system. Given these 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 the virus.

(Refer Time Slide: 10:40)

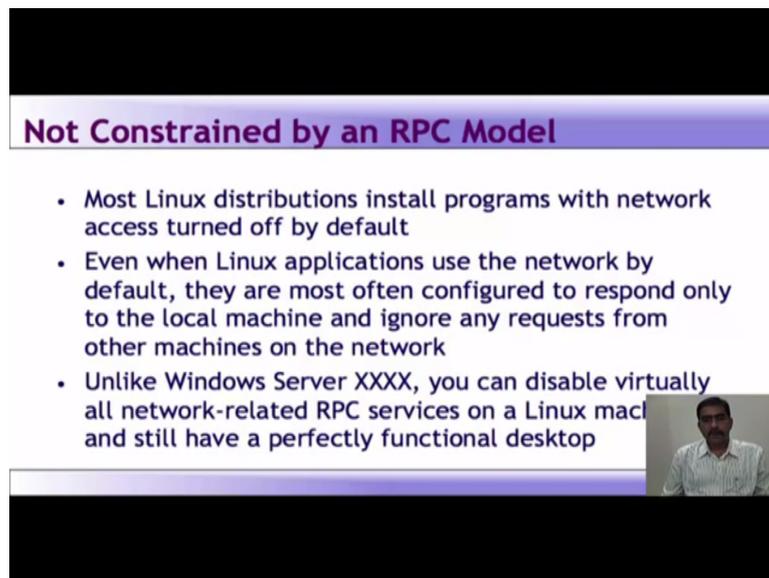
## Modular by Design, not Monolithic

- 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 to the applications. Kernel is again, we emphasize the code or bridge of Linux. Not everything in Linux, is also modular. There are two most popular graphical desktops, gnome and kde, they 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. And advanced, or adverse impact of this monolithic approach, is minimized by the fact that Linux kernel is designed, to be as minimal a part of the system as possible. Now, there is Linux follows, a so called philosophy all most to the 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, now useful here is as specified by the end user. It is feature that does not have access, to the vulnerable parts of the unit system it is not constrained by an rpc model.

(Refer Time Slide: 11:59)

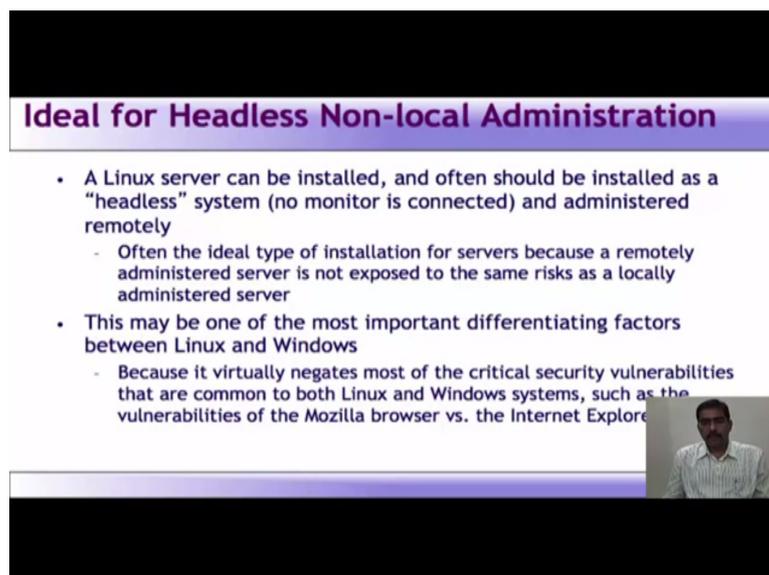


**Not Constrained by an RPC Model**

- 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 XXXX, you can disable virtually all network-related RPC services on a Linux machine and still have a perfectly functional desktop

So, most Linux distribution installed programs are with network access, turned off by default. So, even when Linux applications use the network, they are most often configured to respond only to the local machine, and ignore any request from other machines on the network. Unlike windows server, you can disable virtually all network related rpc service, on the Linux machine, and still perfectly have a good functional desktop.

(Refer Time Slide: 12:31)



**Ideal for Headless Non-local Administration**

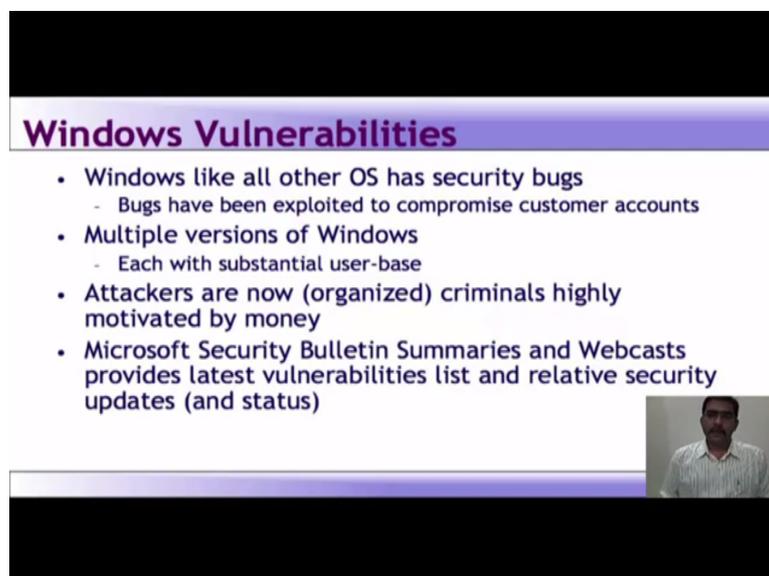
- A Linux server can be installed, and often should be installed as a “headless” system (no monitor is connected) and administered remotely
  - Often the ideal type of installation for servers because a remotely administered server is not exposed to the same risks as a locally administered server
- This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

It is ideal for headless non-local administration, now a Linux server can be installed and often should be installed as a headless system, which is no monitor connected, no key board

connected, no mouse connected and administered remotely. Often, the ideal type of installations for servers, because a remotely administered server is not exposed to the same risk, as locally administered server.

This may be one of the most important differentiating factors, between the Linux, and windows, because it virtually negates most of the critical security vulnerabilities, that are common to both Linux, and windows system, such as vulnerabilities of the Mozilla browser, versus the internet explorer browser. So, you cannot run a Mozilla browser on a server. Let us take a look at windows vulnerabilities.

(Refer Time Slide: 13:25)



**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)

Video inset showing a man speaking.

Some of them at least, windows like any other os, has security bugs. Bugs have been exploited to compromise customer accounts. There are multiple versions of windows, and each of the versions with a substantial user base, and attackers are now more organized, or they are organized criminals. They are highly motivated by money. Microsoft security bulletin summarize, and webcasts provide latest vulnerabilities list and relative security updates and status.

(Refer Time Slide: 14:03)

## 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



Now, some example we will look at what kind of vulnerabilities are coming, vulnerabilities in windows kernel could allow elevation of privileges, that is in 2010, what this particular vulnerability does. Most severe of these vulnerabilities could allow elevation of privileges, if an attacker logged on locally, and ran a specifically crafted application.

An attacker must have a valid log on credential, and should be able to log on locally to exploit this vulnerability. This vulnerability could not be exploited remotely, or by anonymous users. Now, this is where the fundamental difference, between windows and Linux come in. The system administrators in a windows system log on locally, use it like a desktop, install untrusted applications, introduce unknowingly viruses, trojans or other malware into the system. So, this is exactly an example of why an headless system is required.

(Refer Time Slide: 15:15)

## 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



Another Vulnerability in windows kernel, could allow elevation of privileges security updates, resolves several privately imported vulnerabilities in Microsoft windows. Now, here, what basically happens is the denial of service condition, and this particular vulnerability affects, almost all versions of windows that is, vista service pack 2, server 2008, windows 7, Microsoft windows, release , r2. So, again most likely it will result in a denial of service condition.

(Refer Time Slide: 15:48)

## Linux Vulnerabilities

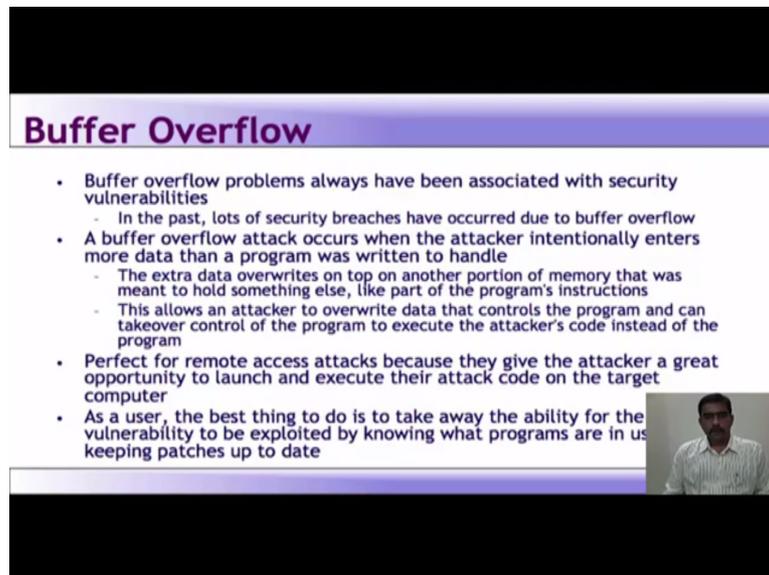
- Default Linux installations (un-patched and unsecured) have been vulnerable to
  - Buffer overflows
  - Race conditions
  - Abuse of programs run "SetUID root"
  - Denial of Service (DoS)
  - Web application vulnerabilities
  - Rootkit attacks



Linux vulnerabilities, now default Linux installation that is un-patched, and unsecured systems have been vulnerable to buffer overflows, race conditions, abuse of programs that is run setuid and a run setuid, denial of service, web application vulnerabilities. Why web

application vulnerabilities have come here is, because most of the web servers which are running the big ones, run on some version of Linux, then rootkit attacks, we will look into all of those, first let us look into buffer overflow.

(Refer Time Slide: 16:21)



**Buffer Overflow**

- Buffer overflow problems always have been associated with security vulnerabilities
  - In the past, lots of security breaches have occurred due to buffer overflow
- A buffer overflow attack occurs when the attacker intentionally enters more data than a program was written to handle
  - The extra data overwrites on top on another portion of memory that was meant to hold something else, like part of the program's instructions
  - This allows an attacker to overwrite data that controls the program and can takeover control of the program to execute the attacker's code instead of the program
- Perfect for remote access attacks because they give the attacker a great opportunity to launch and execute their attack code on the target computer
- As a user, the best thing to do is to take away the ability for the vulnerability to be exploited by knowing what programs are in use and keeping patches up to date

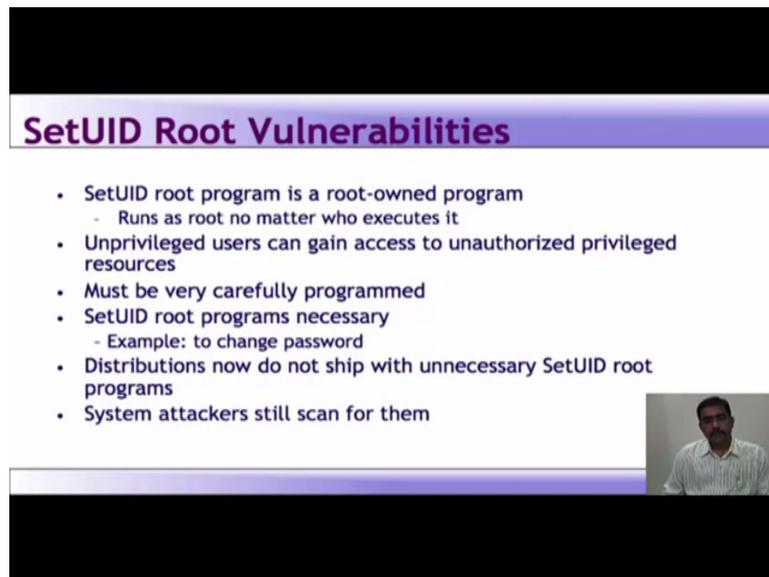


What is a buffer overflow, buffer overflow occurs when attackers intentionally enters more data, than a program is able to handle, that is what a buffer overflow means. As a user, the best thing to do is, to take out the ability for the vulnerability to be exploited, by what is known as patching the system properly, and to know what programs are in use. So, buffer overflow have always been associated with, security vulnerabilities. In the past, lot of security breaches, have occurred due to buffer overflow.

Now, it occurs when the attacker intentionally enters more data than a program is able to handle, we have seen that, the extra data overrides on top of another portion of the memory, that was meant to hold something else, like a part of the program's instruction. This allows the attacker to overwrite data, that controls the program, and can take over control of the program, to execute the attacker's code, instead of the original program.

It is perfect for remote access attacks, because they give the attacker, a great opportunity to launch, and execute their attack code on the target computer. Now, as user the best thing to do is, to take away the ability, for the vulnerability to be exploited, by knowing what program is, are in use, and by keeping the patches up to date, we have already discussed this.

(Refer Time Slide: 18:01)



**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



We will look at setuid root vulnerabilities, as we discussed earlier, when setuid permissions bit is set, the programs will run with the privileges of the user, that owns it rather than, those of the process or user, that is executing it. So, setuid root program is a root-owned program, runs as root, no matter who runs it, or who executes it. If setuid root program can be exploited or abused in any way.

For example, if it can be exploited via a buffer overflow, or a race condition, then the unprivileged user can gain access to unauthorized privileged resources. So, running setuid is necessary for program, which needs to be run by unprivileged users. Yet, it must provide such users, with access to privilege functions, but after changing their password, which requires changes to the protected system file. So, such programs required must be programmed very carefully, due to the history of abuse, again setuid root programs major Linux distributions, no longer ship with the unnecessary setuid root, but system attackers can still scan for that.

(Refer Time Slide: 19:25)

## Web Application Vulnerabilities

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



Then is the web application vulnerabilities, it is a very broad category of vulnerabilities, as we know web applications are written in scripting languages, such as php, perl, java. So, it may not be prone to classic buffer overflow, but it can suffer from poor input handling, including xss, sql code injection, and Linux distribution ship with few enabled by default web applications, like default cgi script, included with the apache web server.

(Refer Time Slide: 20:02)

## Rootkit Attacks

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

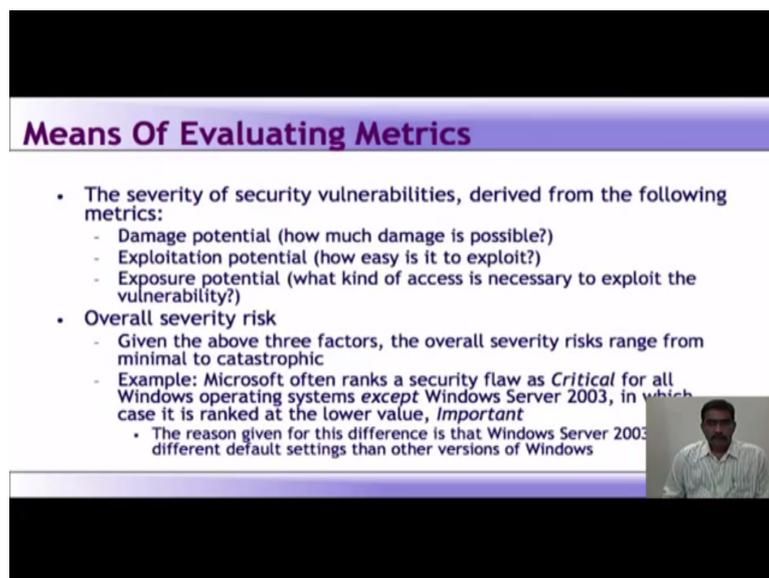


Then comes the rootkit attacks, this is a kind of attack which allows an attacker to cover his or her tracks typically after a root compromise. So, once you get root, then you try to erase your track from the system, so it allows the attacker to tower his or her track which occurs

after a root compromise. Now, rootkits began as a collection of hacked replacements, for some common missed commands, like ls that behaved like legitimate command.

That they have replaced except for hiding an attacker's file directly are possible for example, if an attacker were able to compromise or replace a Linux system ls command, with a rootkit version of ls, then anyone executing the ls command to give files, and directory would see everything except that attacker's files and directories. Now, since the advent of lkm's loadable kernel modules rootkits, have more frequently taken the form of lkm's, so an lkm rootkit covers the track of an attacker in a kernel space. And intercepts system calls pertaining to any user's attempt, to view the intruder's resources. Luckily, lkm rootkits are not completely invisible. Many traditional lkm rootkits are detectable with script chk rootkit. It is available at [www.chk.rootkit.org](http://www.chk.rootkit.org). However, the attacker can get far enough if at all, they compromise the system then we will have to wipe, and rebuild the entire system.

(Refer Time Slide: 22:17)



**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

Now, what are the means of evaluating metrics between windows and Linux, the severity of security vulnerabilities, are derived from the following metrics. And what is the damage potential, how much damage is possible on the system, exploitation potential, how easy is it to exploit, exposure potential, what kind of access is necessary to exploit the vulnerability.

Then the overall severity, given the above 3 factor, that is damage potential, exploitation potential, and exposure potential, the overall risk rate from, minimal to catastrophic. Now, if you take an example, windows or Microsoft often runs a security flaw as critical, for all

windows operating system except the server 2003 in which case, it is ran at a lower value which is important. The reason given for this difference is that, 2003 has a different default setting, than other versions of windows.

(Refer Time Slide: 23:21)

**Example: Microsoft Security Bulletin MS08-067 - Critical**

Vulnerability Information

ii Severity Ratings and Vulnerability Identifiers

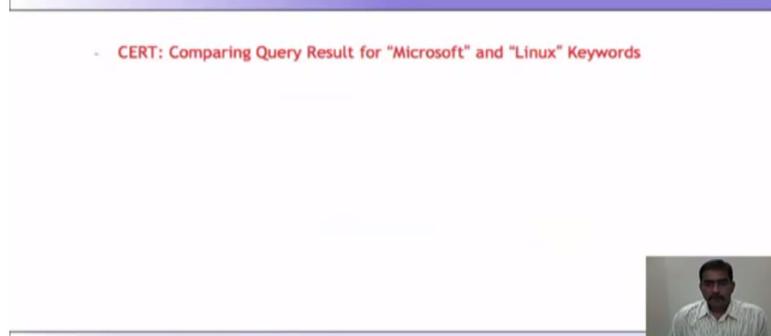
Affected Software	Remote Access Vulnerability - CVE-2008-4250	Aggregate Severity Rating
Microsoft Windows 2000 Service Pack 4	Critical Remote Code Execution	Critical
Windows XP Service Pack 3 and Windows XP Service Pack 2	Critical Remote Code Execution	Critical
Windows XP Professional x64 Edition and 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 SP2 (SP2) for Itanium-based Systems and Windows Server 2003 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



This is an example, of a Microsoft security bulletin, where the vulnerabilities are listed. You can see this critical remote code execution, critical remote code execution, important remote code execution, we have etc. The aggregate severity rate rating, critical important, so this security bulletin looks like this.

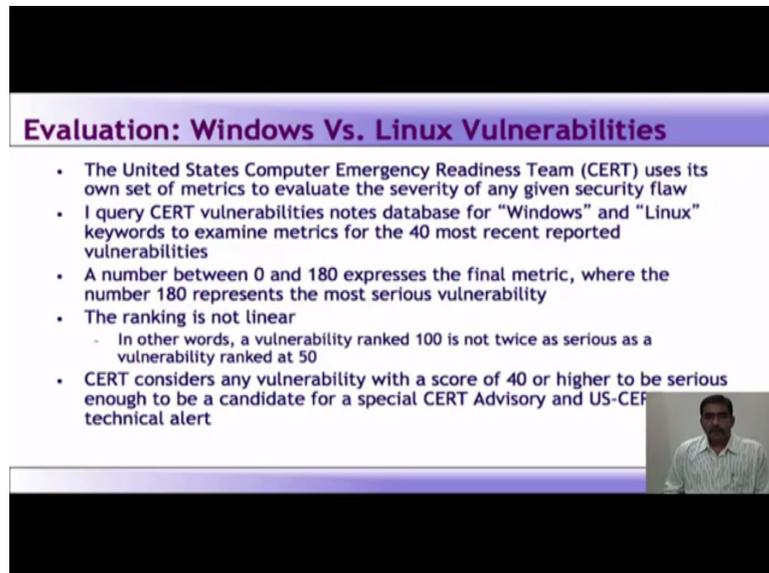
(Refer Time Slide: 23:45)

**CERT: Comparing Query Result for "Microsoft" and "Linux" Keywords**




Now, computer emergency response team CERT, comparing query results for microsoft, and Linux keywords.

(Refer Time Slide: 23:53)



**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



So, when you are evaluating windows versus Linux vulnerabilities, the us computer emergency readiness team uses its own matrix, to evaluate the severity of any given flaw. Now, I query cert vulnerability 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.

This ranking is not linear, in other words, a vulnerability ranked 100 is not twice as serious, as vulnerability ranked 50. So, you get the point there. CERT considers any vulnerability with a score of 40 or higher, to be serious to be candidate for special cert advisory, and us cert technical alert.

(Refer Time Slide: 24:50)

## CERT: Query Result for Keyword "Microsoft"

US-CERT  
UNITED STATES COMPUTER EMERGENCY READINESS TEAM

Search Results

Metric	ID	Date	Title	Name
78	VU0117204	03-17-2003	Buffer Overflow in Core Microsoft Windows DLL	
47.04	VU0209720	08-12-2008	Microsoft Code Management System (CMSM) module remote code execution	
45.96	VU0107620	11-11-2003	Microsoft Windows Workstation service vulnerable to buffer overflow when user specially crafted network message	
45.24	VU0274890	10-12-2004	Microsoft Excel parameter validation error	
41.04	VU0109211	01-14-2010	Microsoft Internet Explorer HTML object summary corruption vulnerability	
37.03	VU0109044	08-13-2008	Microsoft Flight simulator corruption vulnerability	
37.06	VU0242041	08-03-2003	Microsoft Windows Media Player fails to properly launch URLs based on Domain HTML (DHHTML) behaviors	
28.43	VU0243094	12-12-2007	Microsoft Remote Installation Services Windows Path Vulnerability	
27	VU0841480	04-11-2008	Microsoft Windows fails to properly handle COM objects	
25.65	VU0212277	02-13-2007	Microsoft Malware Protection Engine fails to properly generate a specially crafted PDF file	
25.81	VU0234146	01-10-2008	Microsoft Outlook and Microsoft Exchange TSPF decoding buffer overflow	
25.24	VU0214997	08-13-2008	Microsoft Office fails to properly handle specially crafted Rich Text Format files	
24.7	VU0125581	04-08-2008	Microsoft Office Project vulnerable to remote code execution via specially crafted Project file	
23.72	VU0275180	12-14-2003	Microsoft Windows Internet Naming Service (WINS) contains a buffer overflow	
23.18	VU0417081	10-04-2001	Microsoft PowerPoint and Excel fail to properly detect security threats; automatically executing malicious code via crafted documents (MS01-030)	
22.87	VU0103772	11-14-2008	Microsoft Agent fails to properly handle specially crafted .ACF files	
22.87	VU0201416	05-09-2008	Microsoft Exchange fails to properly handle cURL and cURL properties	
22.03	VU0271880	01-08-2007	Microsoft Outlook fails to properly parse Office Search Results (.oss) files	
19.13	VU0126780	10-10-2008	Microsoft Office fails to properly parse embedded records	
18.23	VU0214990	10-10-2008	Microsoft Office fails to correctly parse embedded records	



Now, when you do a query result for microsoft, it lists out all the potential vulnerabilities I have taken from the last, from the oldest to the new ones. So, I have taken a pretty old one, for the presentation here. Similarly, when I do it for Linux, I get a similar thing.

(Refer Time Slide: 25:10)

## 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



Cert web search capabilities do not provide, or produce perfectly desirable results, in terms of granularity, or longevity. It is especially true for Linux, because a Linux cert result includes a number of oracle security vulnerabilities, that are common to Linux, unix and windows. So, in top 40 cert result for windows, the top entry containing the severity metric of 78, there were five entries, which had a severity rating of 40 or greater and in top 40 cert results for Linux, the top entry containing the severity metrics of 26.52.

No other entry has a severity rating of 27 or greater. So, you can see the difference. Note that the cert results only reflect how windows security flaws, tend to be far more frequently severe, than those Linux. These results cannot be expected to mirror our own analysis of frequent or recent, vulnerability patches.