Introduction to Information Security

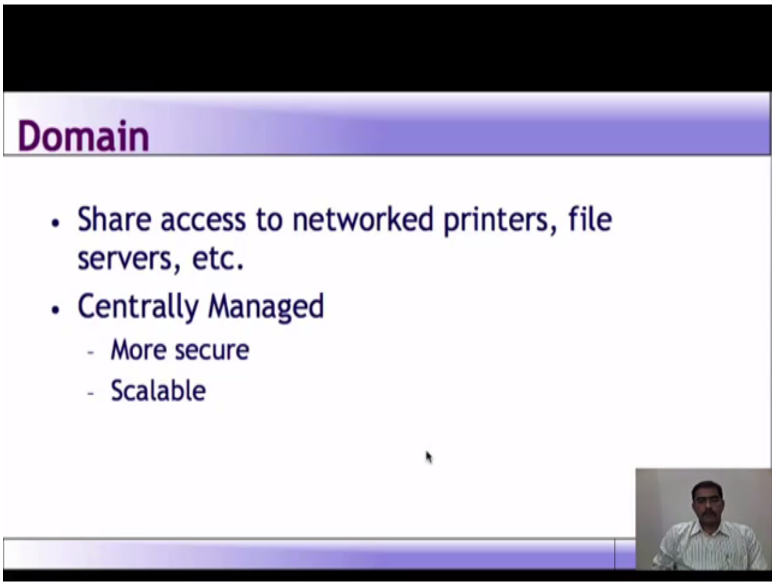
Prof. Dilip H. Ayyar

Department of Computer Science and Engineering

Indian Institute of Technology, Madras

Lecture – 52

(Refer Slide Time: 00:10)

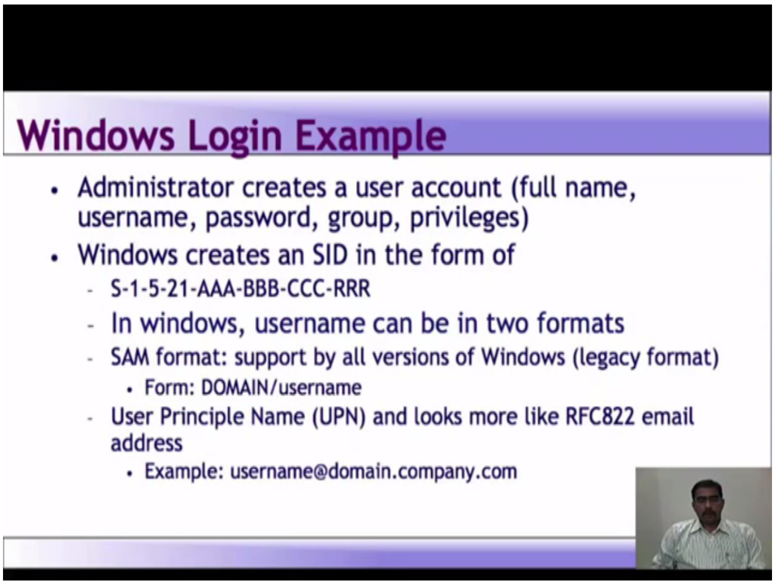


Domain

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

If the users wish they can also logon using local accounts, but local accounts may not have access to the domain resources, such as your network printers, your web server, the email servers and so on. What are the pros and corns of the each scenario? The domain has the major advantage of being centrally managed. So, it is much more secure, if an environment has more than thousand computers and an employee leaves, the user accounts can be disabled centrally rather than thousand individual computers. The only advantage of using local account is that, the computer need not have or does not require the infrastructure to support the domain using AD.

(Refer Slide Time: 01:07)



Windows Login Example

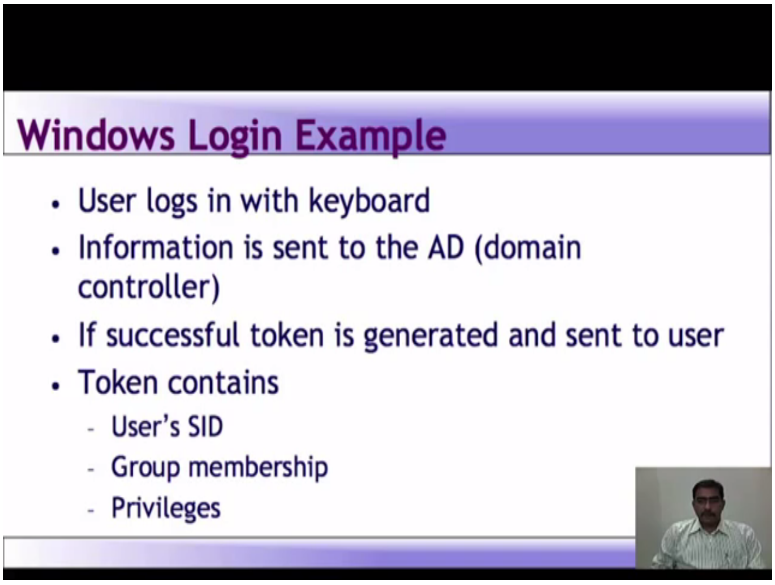
- Administrator creates a user account (full name, username, password, group, privileges)
- Windows creates an SID in the form of
  - S-1-5-21-AAA-BBB-CCC-RRR
  - In windows, username can be in two formats
  - SAM format: support by all versions of Windows (legacy format)
    - Form: DOMAIN/username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

Now, we now the basic elements that make up the core windows at the infrastructure. Let us go over an example when a user logs into a windows system. Before a user log on to the windows network, domain administrator must add the users account into the system. This will include the user name, the account name all this should be unique, the password.

Optionally, the admin can grant group membership and privileges. Now, the administrator creates an user accounts windows creates an SID in the form of S152 1triple A triple B triple C triple R etcetera. In this case S means SID, 1 means SID version number, 5 means the identifier authority, 21 means it is not unique. It just means that there is no guarantee of uniqueness. SID is unique within a domain. Triple A triple B triple C it is a unique number representing the domain and triple R is the relative Id.

It is a unique number that increments by 1 as each SID is unique. In windows username can be of 2 format, one is a SAM format which is supported by all versions of windows or it is also called as a legacy format. The form is domain/username and then there is a user principle name. And it looks more like RFC 8 2 2 email address, example user name at domain.company.com.

(Refer Slide Time: 02:55)

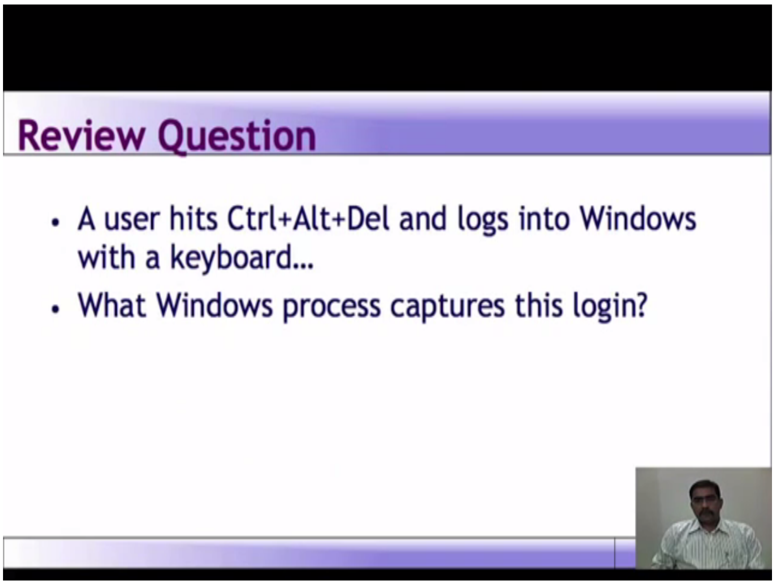


Windows Login Example

- 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

Now, user logs in with the keyboard. Information is sent to the AD or the domain controller. If successful token is generated and sent to the user the token contains users SID, the group membership, the privileges.

(Refer Slide Time: 03:14)

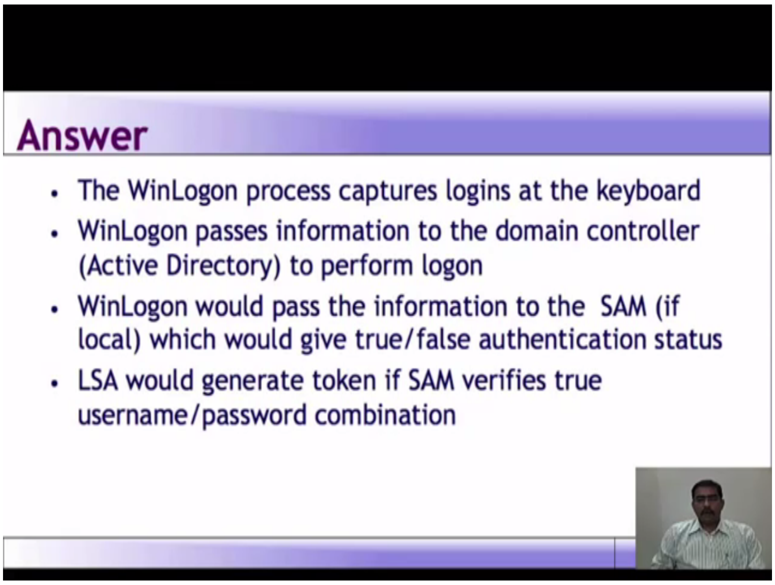


Review Question

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

Now, let us see a quick question.A user hits control plus alt plus del and logs into windows with a keyboard, what windows process captures this logon? That is the question.

(Refer Slide Time: 03:28)

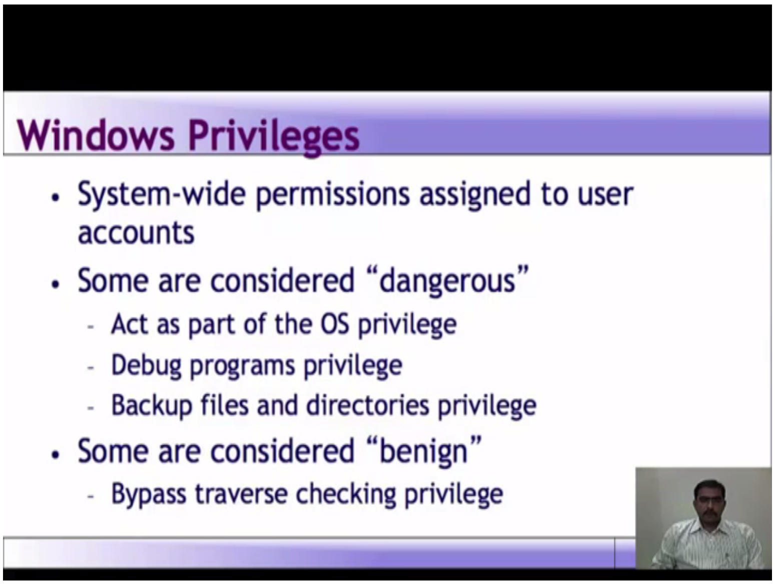


Answer

- 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

The answer is, the WinLogon process captures logins at the keyboard. Now, as we have seen the WinLogon passes information to the domain controller or active directory to perform this logon. The WinLogon would pass this information to the SAM, if it is local which would give true or false authentication status. The local security authority would generate a token if SAM verifies username password combination. Now, let us look at windows privileges.

(Refer Slide Time: 04:00)



Windows Privileges

- System-wide permissions assigned to user accounts
- Some are considered "dangerous"
  - Act as part of the OS privilege
  - Debug programs privilege
  - Backup files and directories privilege
- Some are considered "benign"
  - Bypass traverse checking privilege

System wide permissions assigned to user accounts. Some are considered dangerous that is act as a part of the OS privilege that is considered as dangerous. Debug program privileges is considered as dangerous. Backup files and directory privilege some are considered benign bypass traverse checking privilege of benign means it does not have much impact on that.

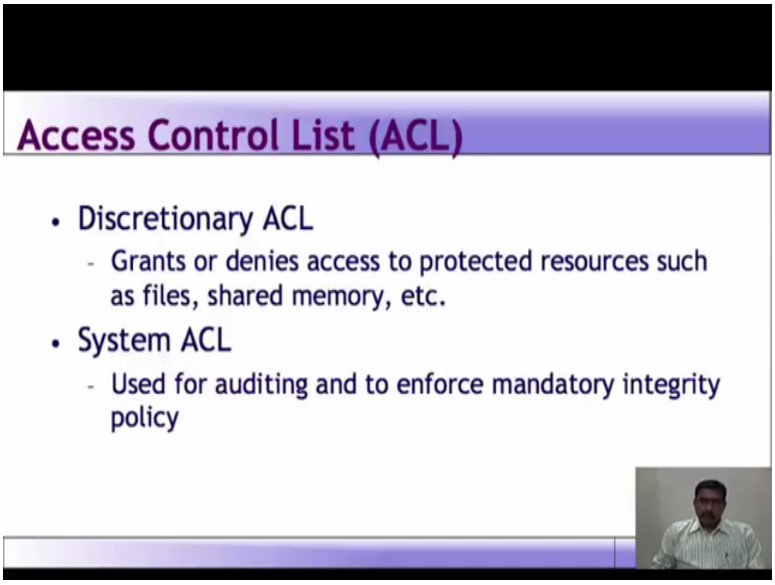
Now, privileges are essentially system wide permissions assigned to user accounts. In other words a privilege is something that you get to do. For example, a windows privilege includes ability to back of the computer, performing backup this because the privilege it bypasses all access checks. So, complete backup is performed. Now, if you take the example of windows vista or windows 7 there are something like over 45 privileges available.

Some privileges are considered very dangerous. Now, some examples of that is act as a part of OS privilege that is it is known as trusted computing based privilege. It allows code run by an account, granted this privilege to act as a part of the most prospective in the operating system, example is the security code privilege. Why it is most dangerous because it is granted only, it grants only the local system account and even admins are granted these privileges.

Now, debug program privilege allows an account to debug any process running in windows because of the nature of debugger is available nowadays. The privilege basically means a user can run any code he or she wants in any running processor. Now, you correlate this to domain 3, you will know what I mean backup end directory privilege. Any process running with this privilege will bypass access control checks.

Some privileges are benign means it is used to traverse directory trees even though the user may not have the permission on the traversed directory. Traverse means travel or browse through. This privilege is assign to all user accounts by default and it is used as a NTFS file system optimization.

(Refer Slide Time: 06:31)



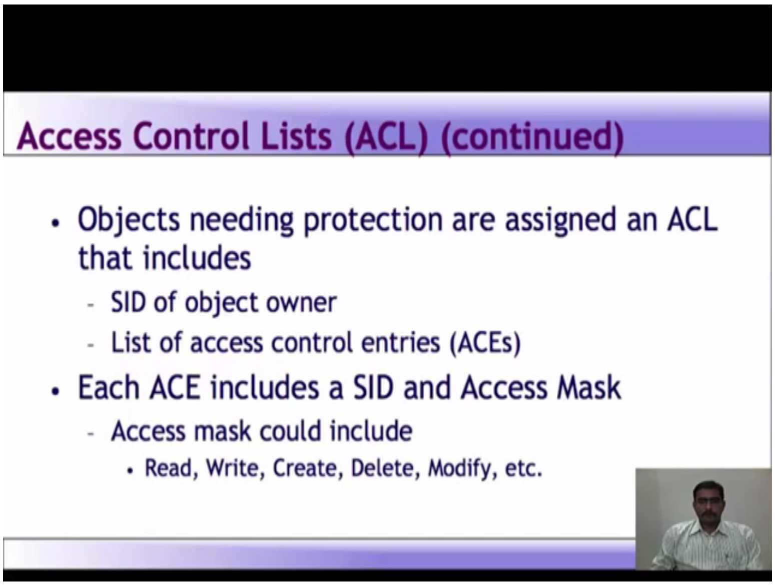
Access Control List (ACL)

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

Then we come to access control list. There are discretionary access control which grants or denies access to protected resources such as files, the shared memory and so on. Then the system access control list which is used for auditing and to enforce the mandatory integrity policy. Now, ACL itself is a list of permission which is 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 that object.

Now, when you say discretionary access control, it is basically granting or denying access to files, shared memory etcetera. Then there is system access control list or you know discretionary access control which is system access control which is used for auditing and in windows it is used to enforce mandatory integrity policy. Now, objects that required the protection or assigned DACL, DACL is discretionary access control which includes the SID of the object owner as well as the list of access control entries access control entries ACE. Now, each ACE includes SID and an access mask. What is an access mask? Access mask could include an ability to read, to write, to create, delete. modify etcetera. So, those are the access masks.

(Refer Slide Time: 08:12)

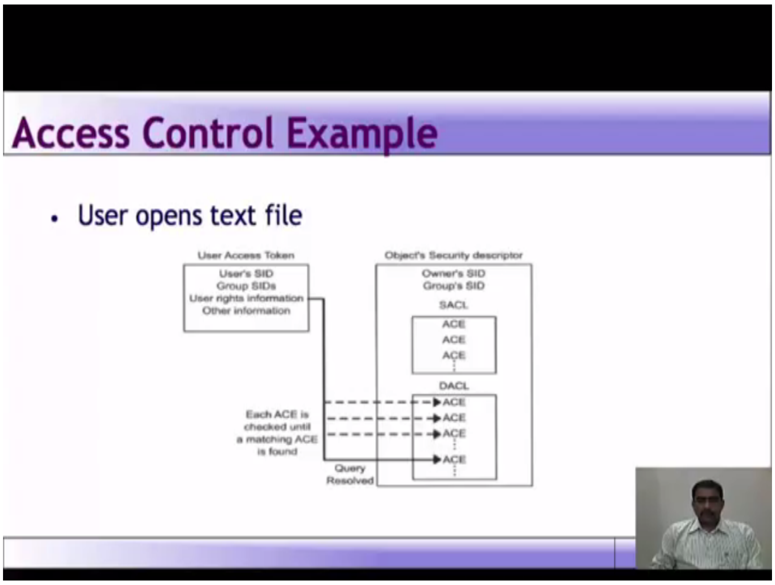


Access Control Lists (ACL) (continued)

- 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.

Objects needing protections are assigned an access controller, right. So, which includes SID of the object owner, the list of access control entries and each ACE should include or includes SID and an access mask. Access mask has we have seen can include read write create delete modify and so on. Now, let us see an example user opens a text file

(Refer Slide Time: 08:43)



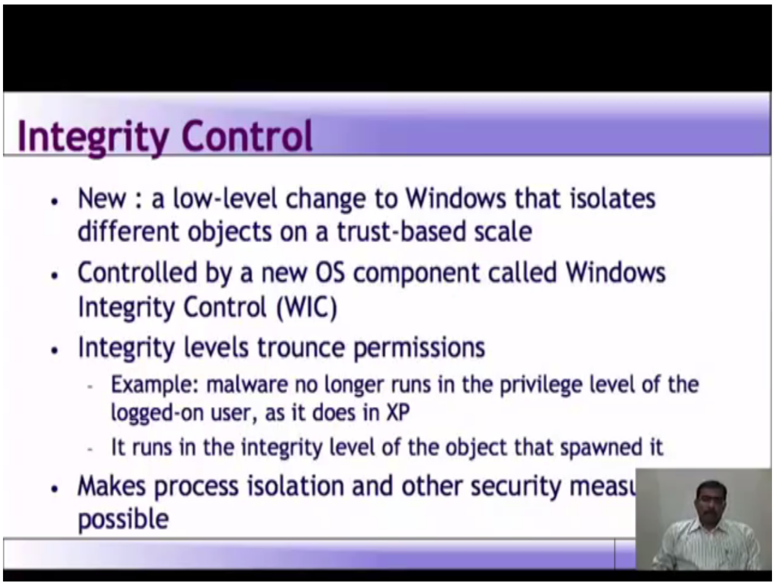
Access Control Example

- User opens text file

Now, user access token as given users SID, group SID, user right information and other information is there. Now, each access or ACE is checked until a matching ACE is found. So, that particular query is resolved, then the object security descriptor says the owner's SID, the group's SID, the SACL whether it is a discretionary access control or a SACL. So, this is basically how it works.

Now, user opens a text file, the system checks to which user whether that user is valid, whether the group SID is, whether it belongs to that particular group, what are the rights on that particular object and other information associated with this. If everything matches then the access is given.

(Refer Slide Time: 09:42)



Integrity Control

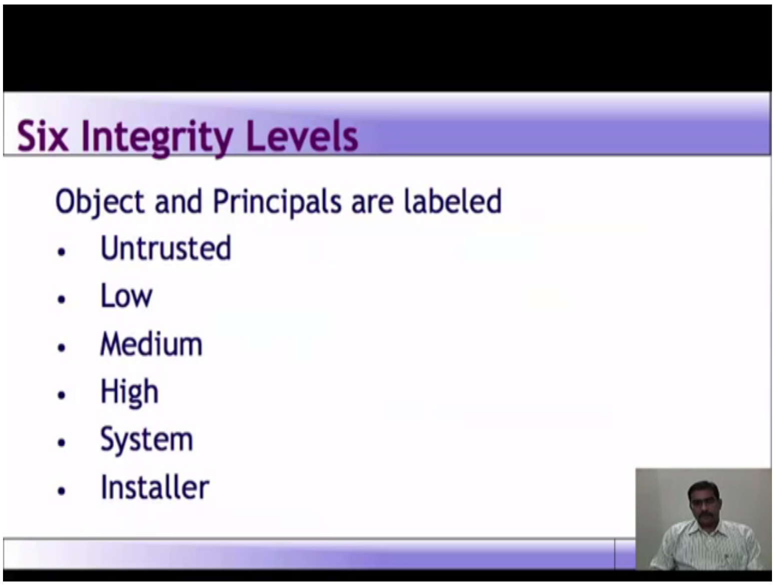
- New : a low-level change to Windows that isolates different objects on a trust-based scale
- Controlled by a new OS component called Windows Integrity Control (WIC)
- Integrity levels trounce permissions
  - Example: malware no longer runs in the privilege level of the logged-on user, as it does in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

Then we come to integrity control. It is a new control right from windows vista, it is the low level change to windows that isolates different objects on a trust based scale. Now, in windows vista there is something called a windows integrity control capability. What it does is, it examines how the OS protects the object such as files and folders on the computers. The different levels of protections that are offered and WIC or integrity control windows integrity control 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 object whether they are files printers, named files registery trees it is and so on from different kinds of attack, from malware or innocent user error or unknowing user errors. The concept of WIC is based on establishing the trust worthiness of various objects and controlling the interaction between the objects based on their integrity or level or trust worthiness. The primary objective of the WIC is to ensure that only objects with integrity level equal to or greater than the target objects allowed to interact with it.

Essentially, if the object is less trust worthy it is prohibited from hacking or interacting with more trustworthy for the object. Now, this is something which is similar to your access control model or confidentiality models like the Lapadula model and there is some Biba model integrity. So, you can actually read about different confidentiality models integrity models, clock Wilson is an integrity model. So, here in this particular case in windows, the primary object of the integrity control is to ensure that the objects with the integrity level equal to or greater than the target object are allowed to interact with it.

(Refer Slide Time: 12:24)



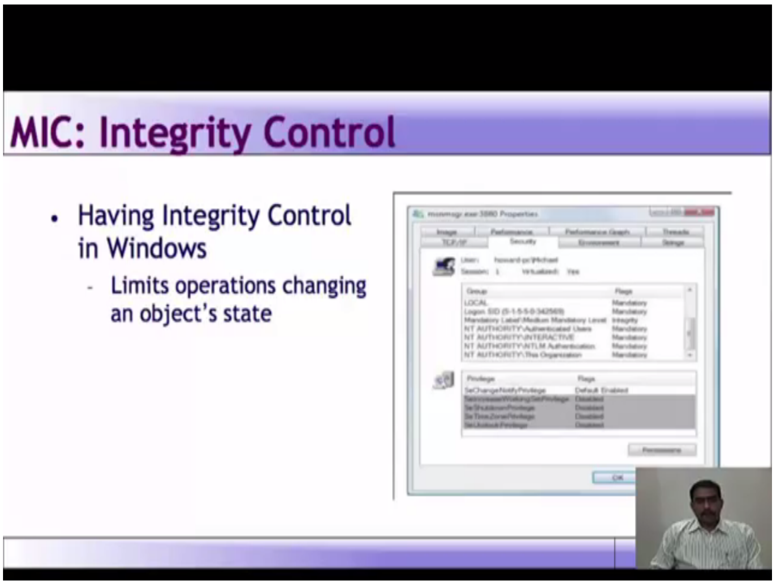
Six Integrity Levels

Object and Principals are labeled

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

There are 6 integrity levels which are labeled, they are untrusted means it is rarely seen. For example, anonymous logs on logons low or internet features that is IE 7 and the temporary internet file folders let medium which is the default integrity level. It is used for standard user accounts and most files. High is for administrator accounts, system is for the kernel and system services, installer, but why is something called installer. Because installer needs to have a higher integrity level then other objects in order to ensure that the uninstall works properly. So, that is why there is an integrity level call installer also.

(Refer Slide Time: 13:15)



MIC: Integrity Control

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

msnmsgr.exe:3880 Properties

User: howard-pc\Michael
Session: 1 Virtualized: Yes

Group	Flags
LOCAL	Mandatory
Logon SID (S-1-5-5-0-342569)	Mandatory
Mandatory Label\Medium Mandatory Level	Integrity
NT AUTHORITY\Authenticated Users	Mandatory
NT AUTHORITY\INTERACTIVE	Mandatory
NT AUTHORITY\NTLM Authentication	Mandatory
NT AUTHORITY\This Organization	Mandatory

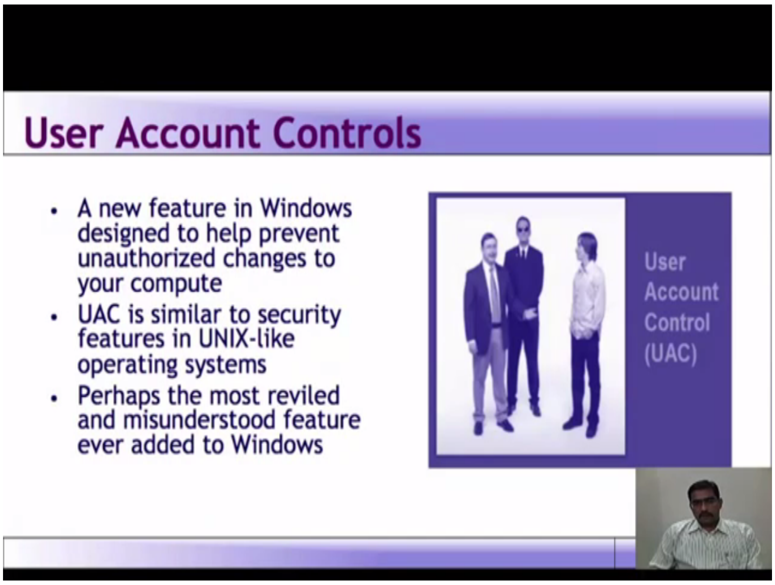
Privilege	Flags
SeChangeNotifyPrivilege	Default Enabled
SeIncreaseWorkingSetPrivilege	Disabled
SeShutdownPrivilege	Disabled
SeTimeZonePrivilege	Disabled
SeUndockPrivilege	Disabled

Now, mandatory integrity control. Now, why discretionary access control is which is DACL are useful, they also have some limitation. They do very little to safe guard the system stability and they cannot or they cannot stop the malicious software from tricking users into executing it. Now, MIC adds notion of trust worthiness, evaluations into operating system. Subjects with low degrees trustworthiness cannot change data of higher degrees subjects with higher degrees of trust worthiness cannot be force to rely on data of lower degrees.

So, that is what MIC does, it adds the notion of trust worthiness, evaluation into operating system. Now, when a modified or a write operation occurs, check the security integrity or check that the subjects integrity level dominates the objects integrity level. Now, these levels are very interesting, levels are mandatory and they are always assigned by the OS itself or by the administrator. Now, we have to look at how different elements of this system or trusted in different ways.

Now, internet below standard user, it is a very smart move installed. Now, if you image an attack to create of the file or process at a level above high. Even the administrators will not be able to delete it. So, that is a very dangerous, these things can happen even that this kind of security levels.

(Refer Slide Time: 15:13)

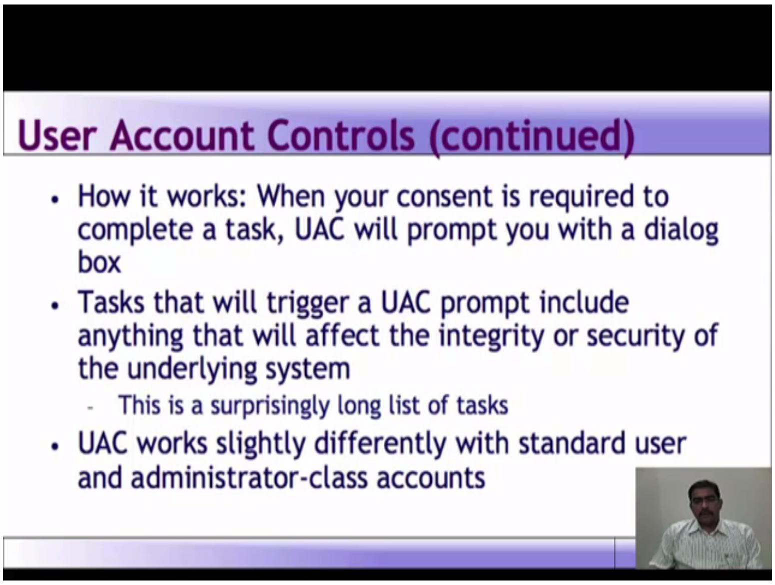


User Account Controls

- A new feature in Windows 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

User access control or user account controls. This is a new feature in windows after vista design to prevent help prevent unauthorized changes to the computer. UAC similar to security feature an Unix like operating system. Perhaps the most reviled and misunderstood feature ever added to windows. Now, this feature is so well known that even apple makes fun of it in their I am their PC add, I am a PC, I am a MAC add , but there is some ironing perhaps hypocricy . Apple's mac OS X includes the feature just like user access control and always add. It is just good security. If you put it in other way UAC tries to help you from making stupid nature. It also tries to prevent malware from acting on your behalf.

(Refer Slide Time: 16:19)

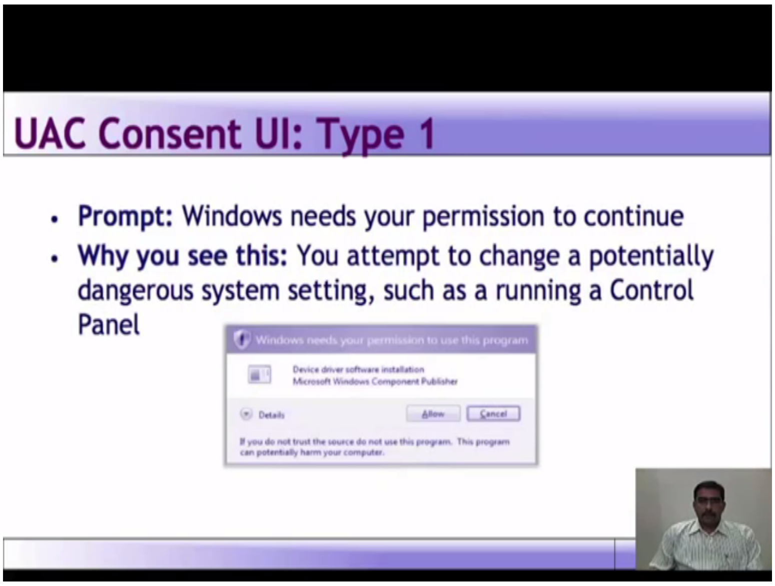


User Account Controls (continued)

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

How it works? When your consent is required to complete a task the 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 itself. There is surprisingly a very long list of tasks which are for a UAC.

(Refer Slide Time: 16:44)



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

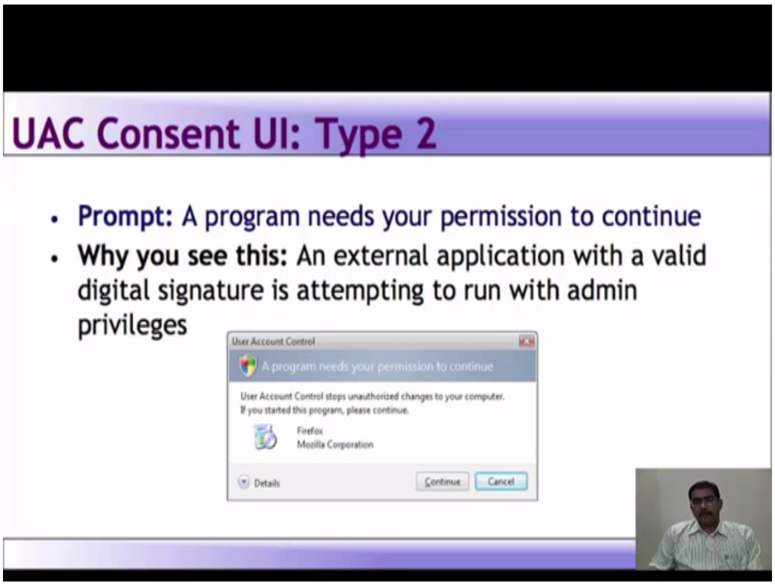
Windows needs your permission to use this program

Device driver software installation
Microsoft Windows Component Publisher

If you do not trust the source do not use this program. This program can potentially harm your computer.

Now, UAC consent user interfere type one. This you would have seen a lot of in this windows. The prompt windows needs your permission to continue, that is the prompt that is the dialog box. Why do you see this, your attempt to change a potentially dangerous system setting such as a running a control panel. Now, if in this example if you see the user is admin user. So, they only have to click allow or type control A to move a long, if the user where standard user you would see the same dialog, but with the space for entering the name and password of an admin level account.

(Refer Slide Time: 17:32)



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

User Account Control

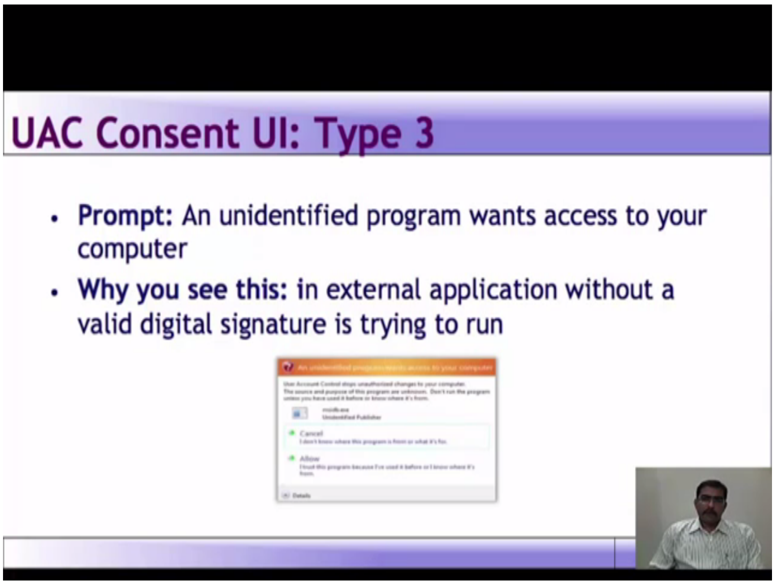
A program needs your permission to continue

User Account Control stops unauthorized changes to your computer.
If you started this program, please continue.

Firefox
Mozilla Corporation

Now, this type 2 is identical to the previous example except that an external application is attempting to perform tasks that is for higher integrity level. In this example, I am showing the standard using experience, where a user name and password is required to continue. Had this been an admin account you would only see allow or cancel one. Note that this application is trusted, in the sense that windows can confirm its digital signature, but it still requires privileges. So, UAC snaps to attention.

(Refer Slide Time: 18:10)



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

An unidentified program wants access to your computer

User Account Control stops unauthorized changes to your computer.
The source and purpose of this program are unknown. Don't run the program unless you have used it before or know where it's from.

[illegible]
Unidentified Publisher

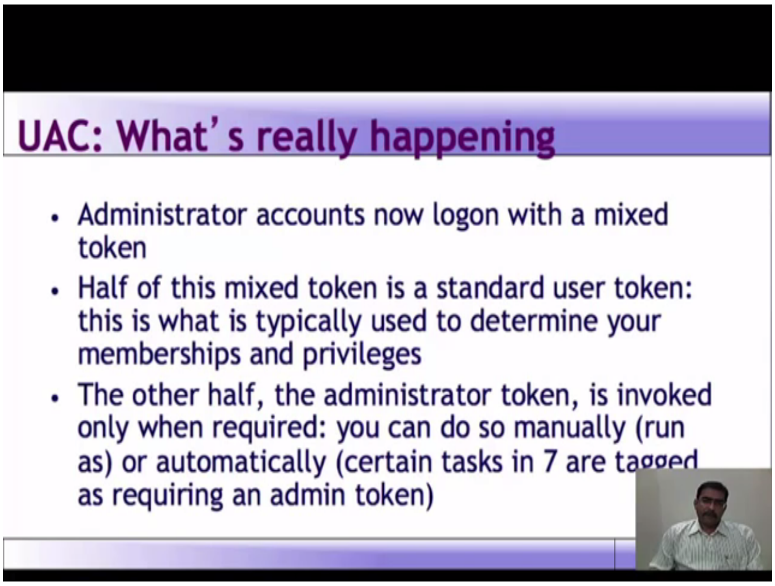
Cancel
I don't know where this program is from or what it's for.

Allow
I trust this program because I've used it before or I know where it's from.

Now, type 3 the prompt is an unidentified program wants access to your computer. Where do you see this? In external application without a valid digital signature is trying to run. It is very colorful, this particular name is very large. This happens when an unknown application. for example, virtually any third party application you can download from the internet that tries to run on windows. Again this is an admin experience where you have allow and cancel option, but if you notice how different this dialog is Microsoft is really trying to get your attention this time.

That is because windows itself does not have any idea whether this application is safe. Now, do you see the problem with these dialogue. Aside from being annoying which they are, the fear is that users simply will get used to the popping up and will mindlessly click allow each and every time.

(Refer Slide Time: 19:16)

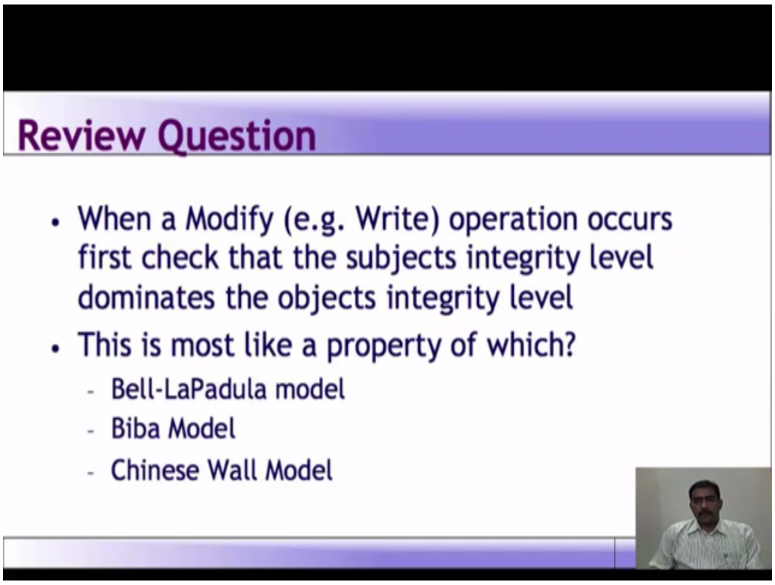


UAC: What's really happening

- 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 7 are tagged as requiring an admin token)

But in UAC what is really happening? Administrator accounts now logged on with the mixed token, half of this mixed token is a 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 that is run as administrator or automatically certain tasks in 7 are tagged as required an admin token. So, that is what actually happens in a UAC.

(Refer Slide Time: 19:54)

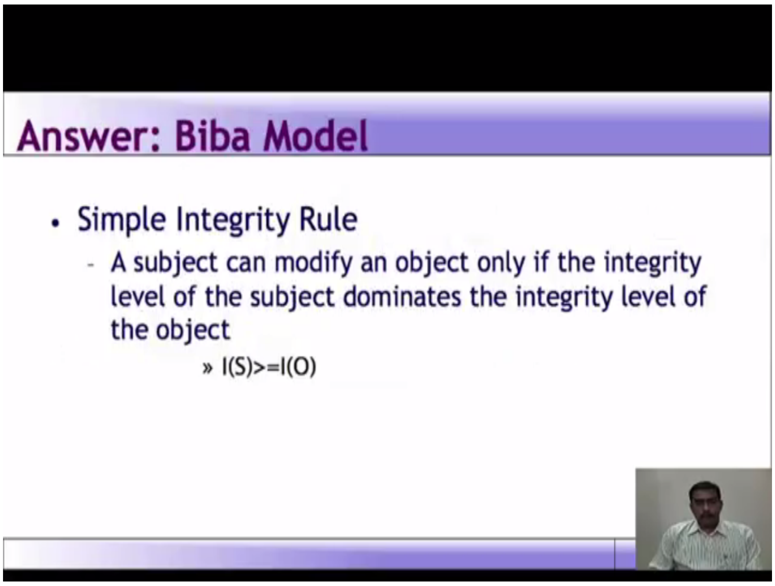


Review Question

- 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

When a modify or a write operation occurs, the first check that the subjects integrity level dominates the objects integrity level. This is most like a property of which it will be Bell Lapadula model Biba is a integrate model. Then there is Chinese model, Chinese wall model let us check the answer.

(Refer Slide Time: 20:15)

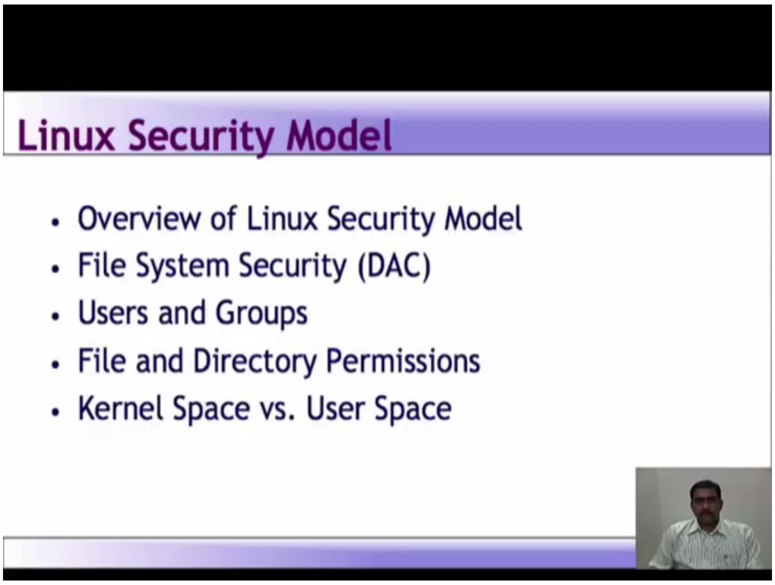


Answer: Biba Model

- 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)

Answer is simple integrity rule Biba model. A subject can modify an only if the integrity level of the subject dominates the integrity level of the object.

(Refer Slide Time: 20:30)

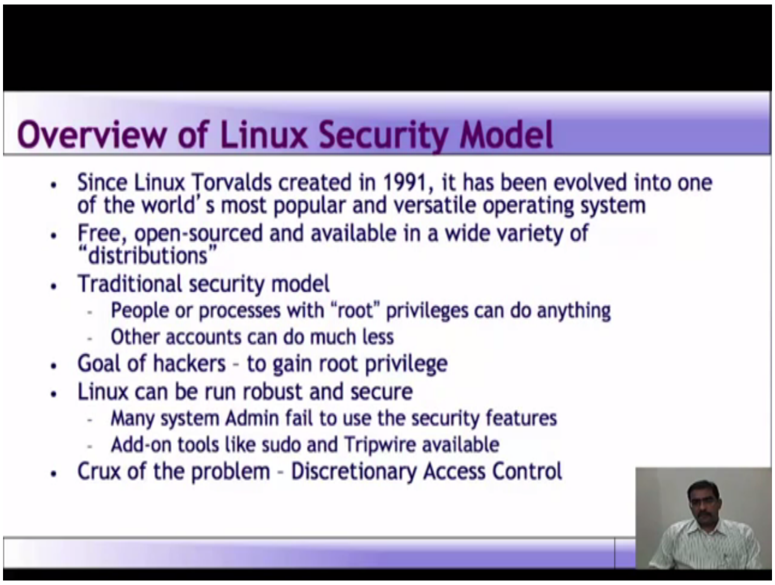


Linux Security Model

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

Now, let us look at Linux security model. We will see an overview of the Linux security model, the file security the users and groups, file and directory permission, kernel space verses users space.

(Refer Slide Time: 20:51)



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

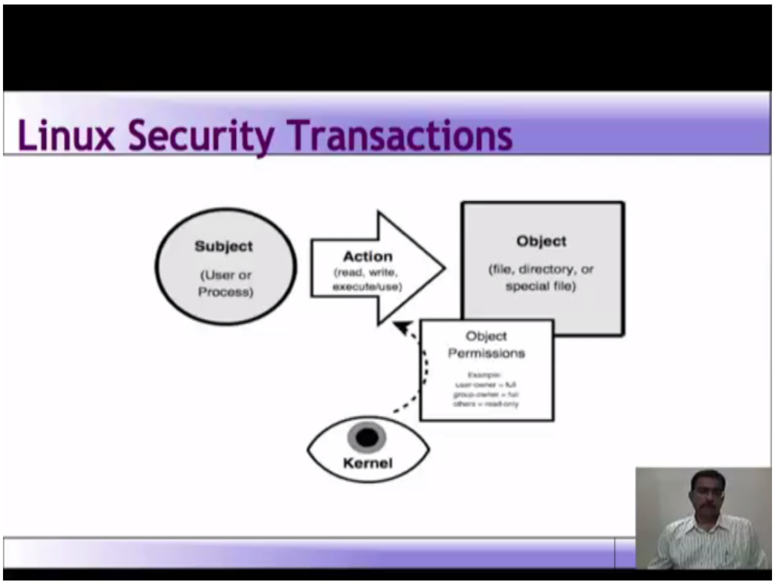
Now, Linux Torvalds created Linux in1991. Since Linux Torvalds created Linux in1991 it has been evolved into or it has evolved into one of the world's most popular and versatile operating system. It is free, it is open source and it is available in wide variety of distributions. Distribution we already discussed Ubuntu is one Fedora is one Mandrake there are live cities also like an optics then there is Suse, open Suse. So, there are so many flavors of this, there is dbm. It is a traditionally very secure model or traditional security model.

People or process with root privileges can do anything. So, underline principle of Linux is anybody having the root can do anything on Linux. Other accounts can do much less based on what rights have been assigned. Now, what is the goal of the hackers in a Linux hack. It is to a gain root privilege. Linux is a very robust and very secure. Many system administrator fail to use the security features just like how we discussed in windows there are several features available or built to Linux, but unfortunately the system administrators fail to see the security features.

Now, there are also add on to like sudo and tripwire available. Now, the crux of the problem here is the discretionary access control. Now, what all this means is that when you look at it from the attackers perspectives, the challenge of cracking the Linux system boils down to getting the privileges. Once root privilege is available then attackers can erase or edit logs. They can hide the processes they can delete or hide files and directories and basically they can redefined the reality of the system itself as experienced by its admin and the users.

The Linux security is basically a game of root takes all. So, if root is compromise then you lost a system. What are the main causes of explosion? Many admins or Linux admins fail to understand or take advantage of the security features which are available in Linux. People can and do run robust secure Linux system while making carefully native Linux security controls plus selected add on tools like tripwire or sudo, but the problem again remians with the discretionary access control.

(Refer Slide Time: 24:02)

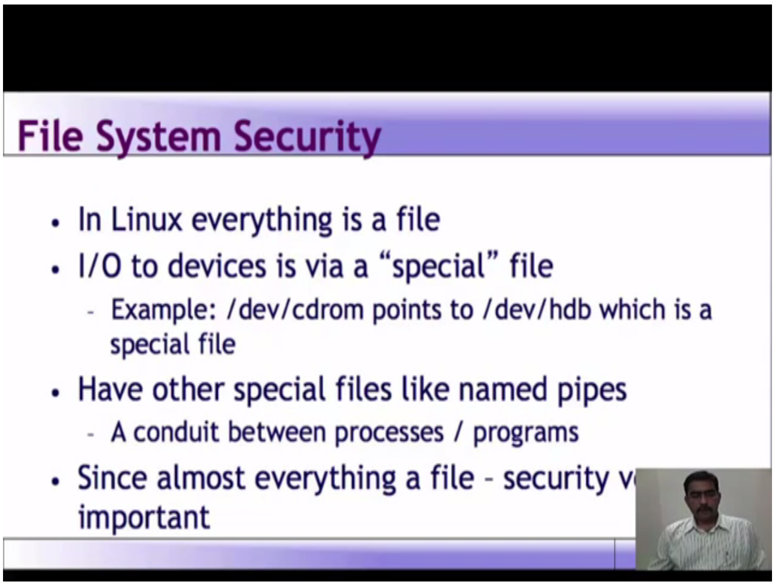


Linux Security Transactions

Now, how a security transaction happens? In the Linux discretionary access control system there are uses, each belongs to one or more groups. There are objects which are are your files and directories or special flies, then users read write and executes these objects based on the objects permission for which each object has 3 sets, which is the user owner, the group owner and other. These permissions are enforced by the Linux kernel. The Linux kernel is the brain of the operating system. What are the basic transaction are happen subjects, subject attempt some action against some object.

Whoever owns and object can set or change its permission, but the real weakness here is the root account has the ability to do both take ownership and change permission of all the objects in the system. So, this provides a way for attackers to hijack those privileges. What are the privileges? The root privileges, again we come back to the same root takes it all. Let us actually take a look or closer look how the DSC in Linux works.

(Refer Slide Time: 25:26)

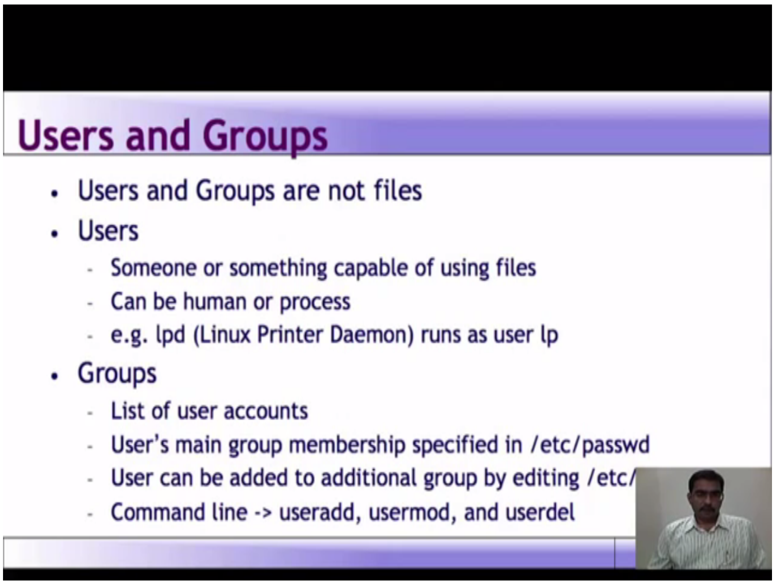


File System Security

- 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 v
important

In Linux everything is a file. So, Linux treats everything as a file, the documents and the pictures, even executable programs are very easy to conceptualize as files. Special files such as named pipes, they act as an input output conduit allowing one process or program to pass data to another processor programs. One common example of a named pipe for a Linux system is, if you go to /dev/u_random where when a program read this file it returns a random characters from kernels random number generator. Now, since almost everything is a file, security is very important in a Linux system.

(Refer Slide Time: 26:22)



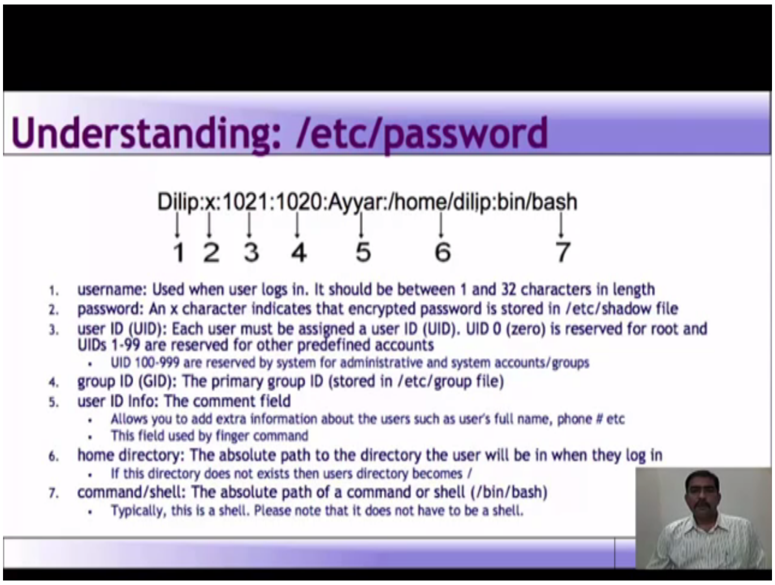
Users and Groups

- 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/
  - Command line -> useradd, usermod, and userdel

Users and groups are not files, users are someone or something capable of using files. It can be human or it can be a process. Example lpd Linux printer daemon runs as user lp, groups list of user accounts. User's main group membership is specified in /etc/password. Users can be added to additional group by editing /etc group. Then using command line we can execute command like adding the user,user add, modify the user delete user. Now, there are two things that are represented or on a Unix system that are not represented by files user accounts and group memberships.

We commonly call it users and groups. In a user account it represents some or something capable of using files, example a user or processor. Standard Linux user accounts this lp for example, is used by line printer daemon. A group account is simply a list of user accounts. So, it is simple to understand. So, user is something which is capable of using files and a group account is simply a list of user accounts. Now, that is also specified by the main group memberships and you can use command line like useradd, usermod and userdel etcetera for executing commands.

(Refer Slide Time: 28:11)



Understanding: /etc/password

```
Dilip:x:1021:1020:Ayyar:/home/dilip:bin/bash
  ↓  ↓   ↓    ↓    ↓      ↓           ↓
  1  2   3    4    5      6           7
```

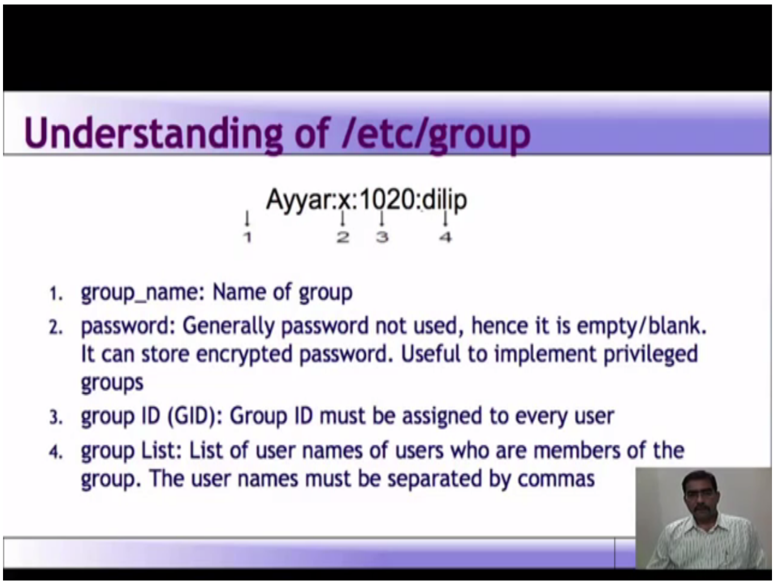
1. username: Used when user logs in. It should be between 1 and 32 characters in length
2. password: An x character indicates that encrypted password is stored in /etc/shadow file
3. user ID (UID): Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved 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.

Now, in the etc password file the structure will be like this. Now, if you see Dilip is the username, it is used when the user logs in, it should be 1 and 32 characters in length. The password marked as an x, x character indicates the encrypted password is stored in /etc/shadow file. The third one is the uid, userid. So, each user must be assigned a user id, uid 0 is reserved for root and uid1 to 99 are reserved for other pre-defined accounts.

Then uid100 to 999 served for system for administrative and system accounts or groups, gid is the group id,that is the primary group id stored in /etc/group file. The user id info the fifth one is the comment field; allows you to add extra information about the users, such as users full name phone number excreta. This field can be used by finger command. Then 6th as the home directory which is absolute path to the directory the user will be in when they log in.

If the directory does not exists then users directory becomes / means root. Then the 7th is the command or shell the absolute path of a command or shell /bin/bash. Typically, this is a shell, but you have to understand need not be a shell for representation purposes we have given it like this.

(Refer Slide Time: 30:06)



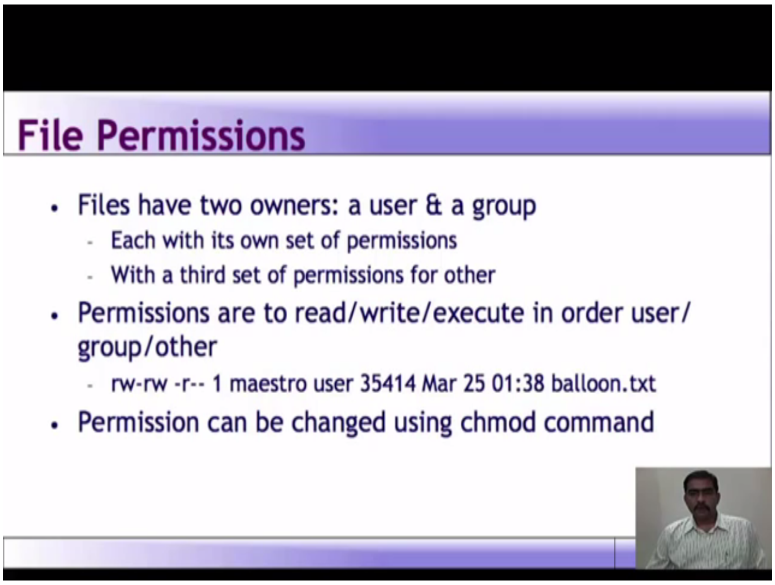
Understanding of /etc/group

Ayyar:x:1020:dilip

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

Now, let us understand what is /etc/group. In this the first one is a group name, the second one is a password again generally password is not used. Hence, it is empty or blank, it can store encrypted password. It is useful to implement privileged groups, the third one is the GID. Group id is assigned to every user and 4th is the group list, list of user names of users who are members of the group. The user names must be separated by commas.

(Refer Slide Time: 30:44)



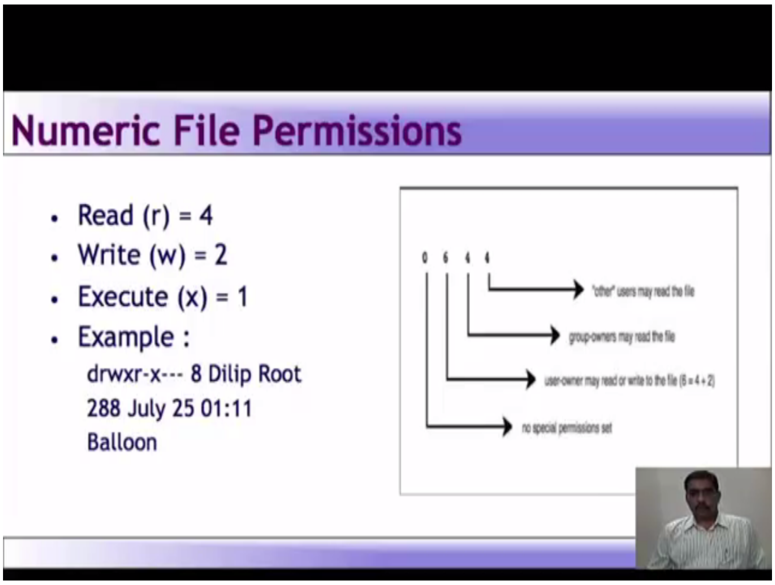
File Permissions

- 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 balloon.txt
- Permission can be changed using chmod command

Then you have file permission. Files have 2 owners a user and a group. Each with its own set of permissions and it can also have the third set of permissions for other. So, now for the user this is the permission for the group this is the permission, for others this is the permission. Take that for creating document in Linux. You are the owner of that. So, you have your read write execute permission for the group, you may give read permission and execute permission.

For others no access will be given. So, permissions are to be red in read, write execute in order of user group or others. So, take an example here balloon.text. Now, read write capabilities for the user read write for the group and read only for the others, maestro user balloon.text is the particular file maestro user is a user permissions can also be changed using chmod command.

(Refer Slide Time: 32:00)



Numeric File Permissions

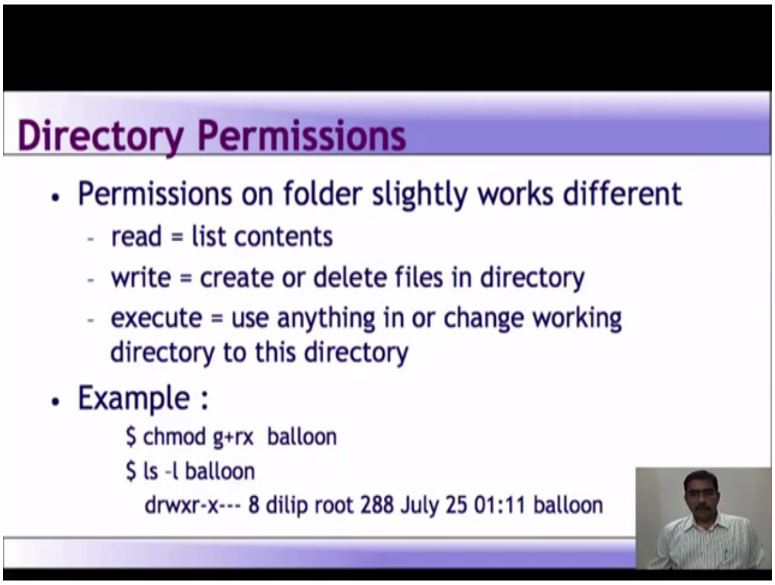
- Read (r) = 4
- Write (w) = 2
- Execute (x) = 1
- Example :

  drwxr-x--- 8 Dilip Root 288 July 25 01:11 Balloon

0 6 4 4
"other" users may read the file
group-owners may read the file
user-owner may read or write to the file (6 = 4 + 2)
no special permissions set

What you have to understand in Linux is, you can solve numerical file permissions. So, read is denoted by 4 write by 2 and execute by 1. Now, here it is 0 6 4 4, now how would arrive at 6 when there only 4 plus 2 plus 1, user owner may read or write to the file. So, you got 6 4 plus 2 read or write. Similarly, user owners may read the files, 4 other users may read the file. Now, the special permission file set is 0. So, if you have permission 0 7 5 5 try to decipher what is actually means.

(Refer Slide Time: 32:44)



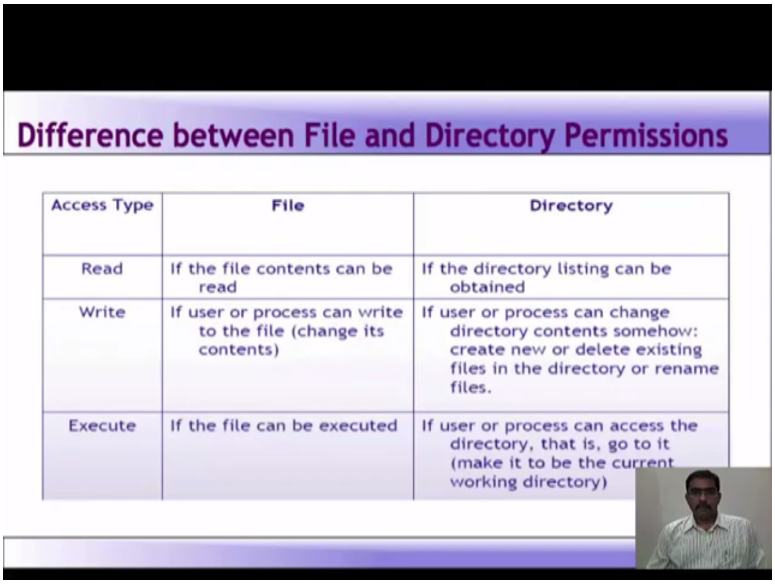
Directory Permissions

- 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 :

```
$ chmod g+rx  balloon
$ ls -l balloon
  drwxr-x--- 8 dilip root 288 July 25 01:11 balloon
```

Then there are directory permissions. Permissions on the folder slightly works different from how it is for a file. Read is for listing contents, write, create or delete files in a directory and execute use anything in or change working directory to this directory. Example chmod g plus rx balloon ls minus l listing balloon. So, you get group permission and this it is got root access group permission.

(Refer Slide Time: 33:21)



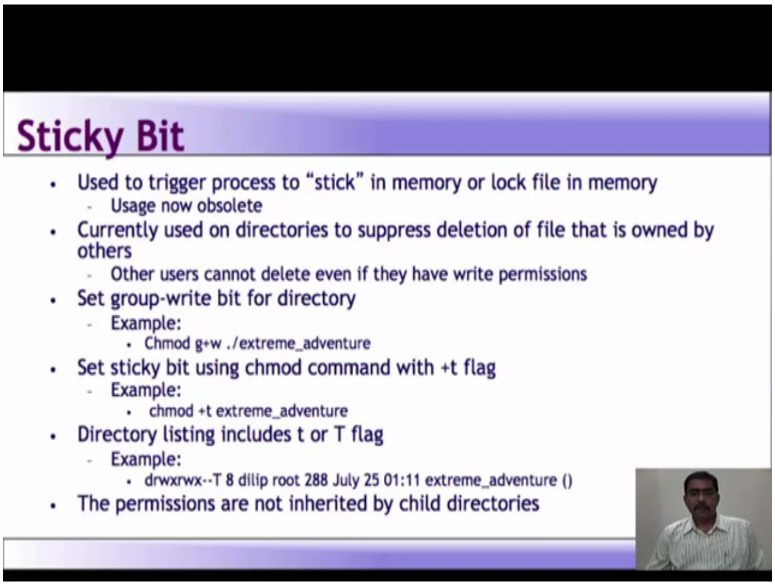
Difference between File and Directory Permissions

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)

Now, what is the difference between file and directory permissions. The access type is the read write and execute. If it s read for a file, if the file contents can be read you get access type read. If the directory listing can be obtained then it is read access. For the file if the user or process can write to the file or change its contents, then it is the right access. If user or process can change directory contents somehow, create new or delete existing files in the directory or rename files then it is write.

So, you can see the certain difference between a file and directory permission. Execute if the file can be executed then it is the execute permission. Now, in case of directory if user or process can access the directory, that is go to it make it to be the current working directory, then you have execute permission on a directory.

(Refer Slide Time: 34:21)



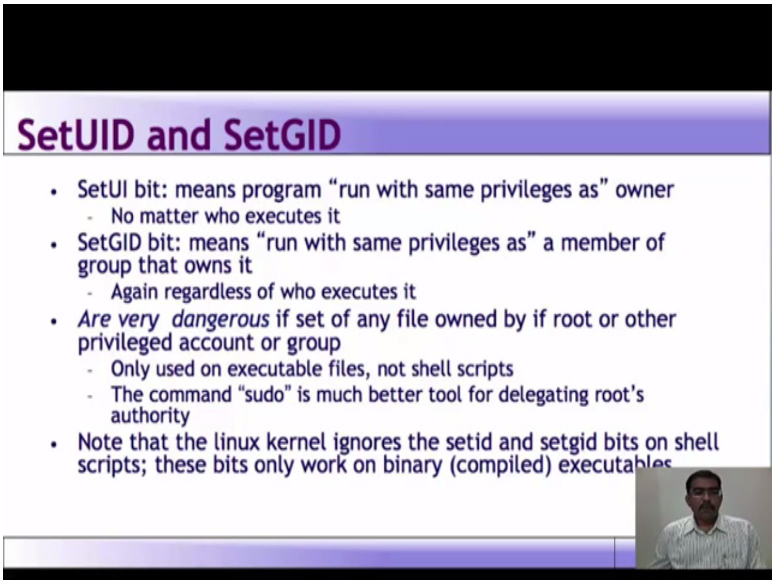
Sticky Bit

- Used to trigger process to "stick" in memory or lock file in memory
  - Usage now obsolete
- Currently used on directories to suppress deletion of file that is owned by others
  - Other users cannot delete even if they have write permissions
- Set group-write bit for directory
  - Example:
    - Chmod g+w ./extreme_adventure
- Set sticky bit using chmod command with +t flag
  - Example:
    - chmod +t extreme_adventure
- Directory listing includes t or T flag
  - Example:
    - drwxrwx--T 8 dilip root 288 July 25 01:11 extreme_adventure ()
- The permissions are not inherited by child directories

Sticky bit, it is used to trigger process to stick in memory or lock file in memory. Now, usage is very obsolete. It is used on directories to suppress of file that is owned by others. Others cannot other user cannot delete even if they have write permissions. So, set group write bit for directory example chmod g+w./extreme_adventure that is the file name. Set sticky bit using chmod command which are +T flag. So, chmod +T extreme_adventure you are enabling the sticky bit. Directory listing includes small t or capital t flag.

Now, that how we get it, the permissions are not inherited by the child directories. In Linux when you said the sticky bit on a directory it limits the user ability to do delete things in the directory. To delete given file in the directory you must either own that file own that directory.

(Refer Slide Time: 35:37)



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

SetUID and SetGID these are most dangerous permission bit in the Unix world. That is SetUID and SetGID. SetUID 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 who executes it. So, imagine the situation if that owner of the is root and somebody else running the program. It is very dangerous if set of any file owned by if root or other privileged by group. It is used or only used on an executable files not shell scripts. The sudo command is a much better tool for delegating roots authority. The Linux, Linux kernel ignores the SetUID and SetGID bits on shell scripts. These bits only work on binary compiled executables.