Security Objects and Parameters Issues in Windows and Linux

Sanjay Kumar Anand
CURAJ
Ajmer

Sudesh Kumar Prajapat
CURAJ
Ajmer

ABSTRACT
In any operating system, various objects are existed which are graphically shown to the users on his/her operating system. Each object has a security attributes that identifies its access control list. This list has an entry for each system user with access privileges. Reading the files (or all the files in a directory), writing the file or files and execute the file or files (if it is an executable file, or program) are most common privileges included.

Thus, an Access Control List (ACL) is a table that tells a computer which user has to given the access rights for a particular system object, such as a file directory or individual file. Each operating system uses the access control list but in different manner. Linux operating system is based on the POSIX ACL and Windows O/S (Microsoft Windows NT/2000) is based on the ACL. This paper presents the analysis of Linux POSIX ACL and the Windows ACL, based on the various parameters.

General Terms
Operating System, System Security et.al

Keywords
Access Control List (ACL), POSIX, Security objects, Security Parameters

1. INTRODUCTION
Access control [1] is the very basic part in computer security. It is applied in the computer system where security is concerned. Its function is to control which principals (i.e. persons, processes, machines) have access to which resources in the system— which files they can read, which programs they can execute, how they share data with other principals, and so on.

Access Control Lists (ACLs) [2] [3] are an additional method to grant specific permissions to certain users. ACLs are a supplement to the existing POSIX permissions, the conventional rules for access rights still apply, but some optional new rules can be added.

Technically, an ACL is a list of individual rights which can be attached to a file system object.

2. ACCESS CONTROL
Access control[15][16][17] basically a security features or services that control that can access resources in the operating system and that regulates the use of system resources according to a security policy and permits their use only by authorized entities (users, programs, processes, or other systems in a network). Firewalls are often referred to as access control devices between networks. It may be the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner. Various examples can be seen in an everyday phenomenon such as - A lock on a car door is essentially a form of access control. A PIN number on an ATM system at a bank is another means of access control.

Thus, it can be said that Access control is the process by which users are identified and granted certain privileges to information, systems, or resources. The basic of access control is to understand how to manage proper disclosure of information.

The primary objective of access control is to preserve and protect the confidentiality, integrity, and availability of information, systems, and resources. It's important that only the right people have access to the data, but it's also important that the data is the right data, and not data that have been modified either accidentally or on purpose.

2.1 Access Control Lists
Access Control List basically consists the user permissions for a file, folder, or other object. It defines what users and groups can access the object and what operations they can perform. Thus, these operations typically include read, write, and execute. For example, if an ACL specifies read-only access for a specific user of a file, that user will be able open the file, but cannot write to it or run the file.

2.1.1 Advantages of Access Control Lists
Access control lists are a straightforward method of managing the files and folders permissions. They are used by most operating system including Windows, Mac, and UNIX systems. While ACLs are typically hidden from the user, they can often be modified using a graphical interface. The access control settings can be modified within the "Sharing and Permissions" section of the window. On LINUX systems, POSIX ACLs can be edited using the chmod () command.

Another advantage is that access control lists filter the network traffic by controlling whether routed packets are forwarded or blocked at the router's interfaces. Based on the criteria specified within the access lists, each router examine the each/every packets to determine whether to forward or drop the packet.

Generally the ACL criteria could be the source address of the traffic, the destination address of the traffic, the upper-layer protocol, or other information.

3. ACL IN WINDOWS
In Windows, an ACL is associated with each system object. Each ACL has one or more access control entries (ACEs) consisting of the name of a user or group of users. The user

can also be a role name, such as "programmer," or "tester" etc. For each of these users, groups, or roles, the access privileges are stated in a string of bits called an access mask. Generally, the system administrator or the object owner creates the access control list for an object (see figure1).

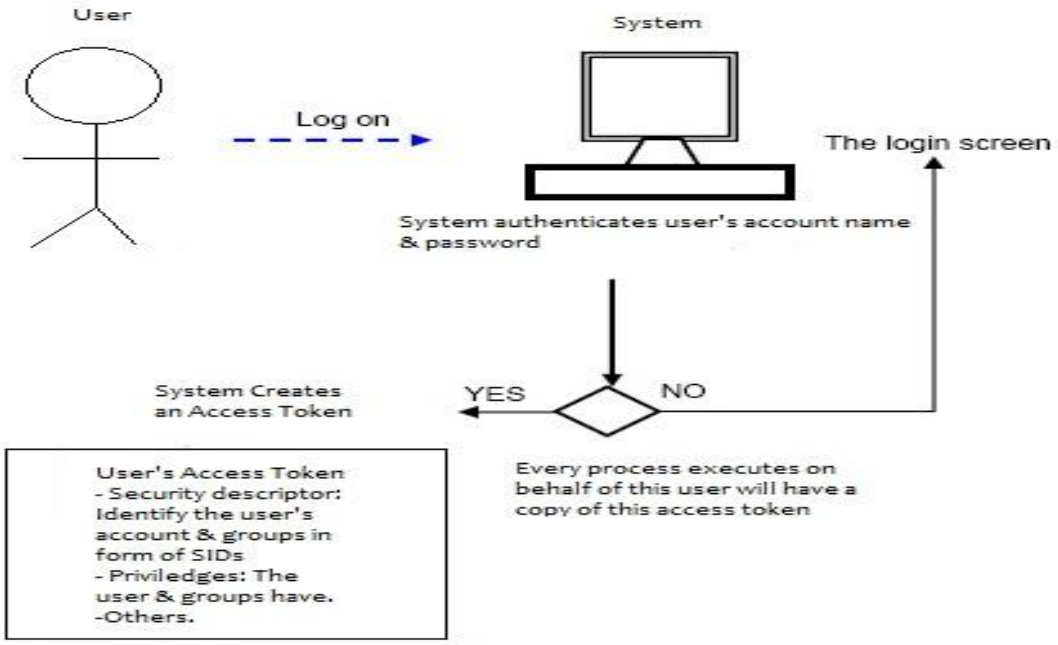

Fig. 1: Login screen for user

3.1 Structure of Windows ACL

In Windows Access Control List (ACL), Each ACE identifies a trustee and specifies the access rights allowed, denied, or audited for that trustee. In windows, basically there are two parts which makes the Access Control Model [4] –

- Access Token and
- Security Descriptor.

Access Token contains information about a logged-on user and the security descriptor contains the security information that protects a securable object and can contain two types of ACLs:

- DACL(Discretionary Access Control list)
- SACL(system access control list)

as shown in the above figure (see figure.1).

DACLs identify the users and groups that are assigned or denied access permissions on an object. If a DACL does not explicitly identify a user, or any groups that a user is a member of, the user will be denied access to that object. By default, a DACL is controlled by the owner of an object or the person who created the object, and it contains access control entries (ACEs) that determine user access to the object.

Generally, DACL has cases-

- For accessing a securable object when it is tried by a process, the system checks the ACEs in the object's DACL to determine whether to grant access to it.
- The system gives full access to everyone in case when object does not have a DACL. In case when the object's DACL has no ACEs, the system denies all attempts to access the object because the DACL does not allow any access rights. The system checks the ACEs in sequence unless it does not find one or more ACEs that allow all the requested access rights, or until any of the requested access rights are denied.

SACLs identify the users and groups that we want to audit when they successfully access or fail to access an object. Auditing is used to monitor events related to system or network security, to identify security breaches, and to determine the extent and location of any damage. By default, a SACL is controlled by the owner of an object or the person who created the object. A SACL contains access control entries (ACEs) that determine whether to record a successful or failed attempt by a user to access a object using a given permission, for example, Full Control and Read.

A system access control list (SACL) basically attempts us to access a secured object. Each ACE specifies the types of access attempts by a specified trustee that cause the system to generate a record in the security event log. An ACE in a SACL can generate audit records when an access attempt fails, when it succeeds, or both. These procedures are given in the following figure (see figure.2).

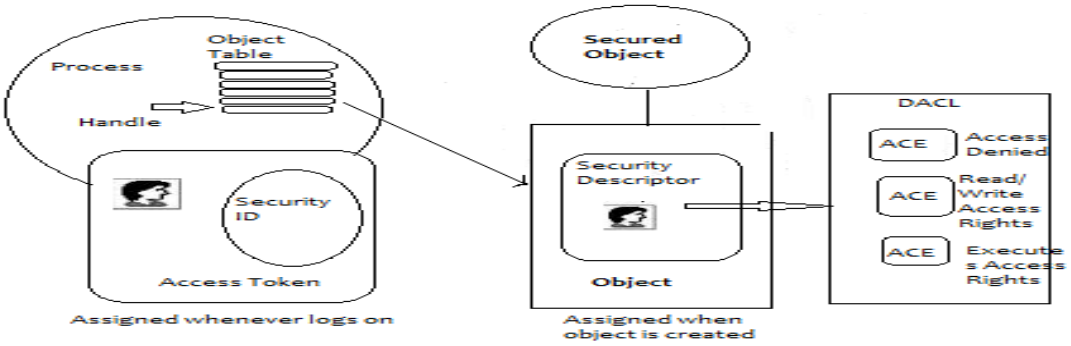

Fig 2: SACL & DACL Procedure

4. ACL IN LINUX

Linux supports the POSIX (Portable Operating System Interface) [5] Access Control List in place of ACL. Generally, POSIX file system defines three types of permissions i.e. read (r), write (w), and execute (x) to different classes of users i.e. owner, group, and other. Each of these classes is associated with a set of permissions. In this model, the owner class permissions define the access privileges of the file owner, the group class permissions define the access privileges of the owning group, and the other class permissions define the access privileges of all users that are not in one of these two classes.

The **ls -l** command in Linux displays the owner, group, and other class permissions in the first column of its output (e.g., ``-rw-r- --" for a regular file with read and write access for the owner class, read access for the group class, and no access for others).

4.1 Structure of Linux POSIX ACL

Linux POSIX ACLs [6] [7] contains two basic classes- a minimum or standard ACL and extended ACL. Standard ACL has the entries for the type's owner, owning group, and other, which correspond to the conventional permission bits for files and directories and the group class permissions, are identical to the owning group permissions where the group class permissions map to the owning group entry permissions whereas, an extended ACL goes beyond this. It must contain a mask entry and may contain several entries of the named user and named group types and group class may contain entries for additional users or groups which maps to the mask entry permissions, whereas the owning group entry still defines the owning group permissions. The mapping of the group class permissions is no longer constant.

A POSIX [11][12] access control list is basically a UNIX permission [8] set plus four extra things:

- Additional named users
- Additional named groups
- A group mask
- Default ACLs for directories

Named users and groups come under the additional user category and group permission sets represented by (user ::) and a group owner (group ::). A named user or group is specified by a single character followed by a colon and the account name, followed by another colon and the permission set.

The following table (Table.1) provides a summary of the various types of POSIX ACL [13][14] entries that are possible.

Table 1. POSIX ACL entities

Security Aspect	Windows ACL	Linux POSIX ACL
Permission Support	10 (enable exact control of who can do what with that object.)	3 (restricted to read, write, and execute for one user, one group, or everyone.)
Permission Type	cumulative	Granted
ACL Model	Inheritance, Dynamic	Inheritance (Inherited at file create time only.)
Concept Based On	container and non-container objects	Orthogonal concepts
IEEE Standard	Windows ACL is not an IEEE Operating System.	POSIX is the IEEE Portable Operating System Interface for Computing Environments.

Entry Type	Text Form	Permission
owner	user ::	rwx
named user	user :	name : rwx
owning group	group ::	rwx
named group	group :	name : rwx
mask	mask ::	rwx
other	other ::	rwx

The permissions are always effective, defined in the entries owner and other. Except for the mask entry, all other entries (named user, owning group, and named group) can be either effective or masked. If permissions exist in one of the above-mentioned entries as well as in the mask, they are effective. Permissions contained only in the mask or only in the actual entry are not effective. This can be demonstrated by the following table (Table-2).

Table 2. Permission in Mask/actual entry

Entry type	Text form	Permissions
Named user	User : sanjay : r-x	r-x
Mask	mask::rw-	rw-
Effective permissions		r-

When an application changes any of the owner, group, or other class permissions (e.g., via the chmod command), the corresponding ACL entry changes as well. Their permissions are always effective and never masked.

5. PERFORMANCE OF LINUX POSIX ACL AND WINDOWS ACL

MS Windows is based on DOS, Linux is based on UNIX. MS Windows Graphical User Interface (GUI) is based on Microsoft-own marketing-driven specifications. Linux GUI is based on industry-standard network-transparent X-Windows. The performance of both Linux POSIX ACL and Windows ACL are compared in the following factors given the following Table-3.

Table 3. Comparison based on parameters

File System	Supported by FTP, NTFS	Supported by ext2, ext3, XFS, ReiserFS, and JFS file system
Policy	RBAC, DAC security groups	DAC, Rule-based ACL
Mechanism	ACL, Active Director Directory, Capability List, ACL , Active Directory Group Policy Object (GPO)	POSIX ACL

6. CONCLUSION

Based on the comparison between windows ACL and Linux POSIX ACL, the conclusion is that both provide the secure File system, policies and mechanism which they used. Also they both provide the ACL model and support the IEEE standard.

7. ACKNOWLEDGMENTS

We would like to give thanks to the experts who have given a valuable guidance to write this paper. We are also thankful to our colleagues and reviewer who have given the valuable suggestion.

8. REFERENCES

[1] Andreas Grünbache, 2003. Known Problems and Bugs in the Linux EA and ACL implementations. March 20, http://acl.bestbits.at/problems.html.

[2] Andreas Grünbacher, 2003, Preserving ACLs and EAs in editors and file managers. February 18, 348. http://www.suse.de/~agruen/ea-acl-copy/ for a description.

[3] Hewlett-Packard: acl (2): Set a file's Access Control List (ACL) information. HP-UXReference. http://docs.hp.com/.

[4] Microsoft Platform SDK: Access Control Lists. http://msdn.microsoft.com/.

[5] Robert N. M. Watson: acl(3): Introduction to the POSIX.1e ACL security API. FreeBSD Library Functions Manual. http://www.FreeBSD.org/.

[6] IEEE Std 1003.1-2001 (Open Group Technical Standard, Issue 6), Standard for Information Technology--Portable Operating System Interface (POSIX) 2001. ISBN 0-7381-3010-9, http://www.ieee.org/.

[7] IEEE 1003.1e and 1003.2c: Draft Standard for Information Technology--Portable Operating System Interface (POSIX)--Part 1: System Application Program Interface (API) and Part 2: Shell and Utilities, draft 17 (withdrawn). October 1997. http://wt.xpilot.org/publications/posix.1e/.

[8] Andreas Grünbacher SuSE Labs, SuSE Linux AG Nuremberg, Germany -POSIX Access Control ists on Linux. http://www.suse.de/~agruen/acl/linux-acls/online/.

[9] Iseminger, D. 2000. Active Directory Services for Microsoft Windows 2000 Technical Reference. Microsoft Press.

[10] IEEE Std 1003.1 (2004 ed.), Unix.org, retrieved 2009-07-26.

[11] "The Open Group announces completion of the joint revision to POSIX and the Single UNIX Specification" (Press release). The Open Group. January 30, 2002, Retrieved 2009-07-26.

[12] POSIX.1-2008, the Open Group.

[13] Native NFSv4 ACLs on Linux. Suse.de. Retrieved 2010-05-04.

[14] Richacls - Native NFSv4 ACLs on Linux, bestbits.at. 2011-09-01. Retrieved 2011-01-04.

[15] Hollingworth, D.; Redmond, T.; Rice, R. DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings Volume: 1 Digital Object Identifier: 10.1109/DISCEX.2000.825035 Publication Year: 2000, Page(s): 320 - 334 vol.1.

[16] Jihong Song; Guiying Hu; QuanSheng Xu Management and Service Science, 2009. MASS '09. International Conference on Digital Object Identifier: 10.1109/ICMSS.2009.5302077Publication Year: 2009 , Page(s): 1 – 4.

[17] Hyang-Chang Choi, Yong-Hoon Yi, Jae-Hyun Seo, Bong-Nam Noh, and Hyung-Hyo Lee, "A Privacy Protection Model in ID Management Using Access Control", O. Gervasi et al. (Eds.): ICCSA 2005, LNCS