Access-control list

In computer security, an **access-control list** (**ACL**) is a list of permissions associated with a system resource (object). An ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects.[1] Each entry in a typical ACL specifies a subject and an operation. For instance, if a file object has an ACL that contains `(Alice: read,write; Bob: read)`, this would give Alice permission to read and write the file and only give Bob permission to read it.

Contents

Implementations

Many kinds of operating systems implement ACLs, or have a historical implementation. The first of which was in the filesystem of Multics in 1965.[2]

Filesystem ACLs

A filesystem ACL is a data structure (usually a table) containing entries that specify individual user or group rights to specific system objects such as programs, processes, or files. These entries are known as access-control entries (ACEs) in the Microsoft Windows NT,[3] OpenVMS, and Unix-like operating systems such as Linux, macOS, and Solaris. Each accessible object contains an identifier to its ACL. The privileges or permissions determine specific access rights, such as whether a user can read from, write to, or execute an object. In some implementations, an ACE can control whether or not a user, or group of users, may alter the ACL on an object.

One of the first operating systems to provide filesystem ACLs was Multics. PRIMOS featured ACLs at least as early as 1984.[4]

In the 1990s the ACL and RBAC models were extensively tested and used to administer file permissions.

POSIX ACL

POSIX 1003.1e/1003.2c working group made an effort to standardize ACLs, resulting in what is now known as "POSIX.1e ACL" or simply "POSIX ACL".[5] The POSIX.1e/POSIX.2c drafts were withdrawn in 1997 due to participants losing interest for funding the project and turning to more powerful alternatives such as NFSv4 ACL.[6] As of December 2019, no live sources of the draft could be found on the Internet, but it can still be found in the Internet Archive.[7]

Most of the Unix and Unix-like operating systems (e.g. Linux since 2.5.46 or November 2002,[8] BSD, or Solaris) support POSIX.1e ACLs (not necessarily draft 17). ACLs are usually stored in the extended attributes of a file on these systems.

NFSv4 ACL

NFSv4 ACLs are much more powerful than POSIX draft ACLs. Unlike draft POSIX ACLs, NFSv4 ACLs are defined by an actually published standard, as part of the Network File System.

NFSv4 ACLs are supported by many Unix and Unix-like operating systems. Examples include AIX, FreeBSD,[9] Mac OS X beginning with version 10.4 ("Tiger"), or Solaris with ZFS filesystem,[10] support NFSv4 ACLs, which are part of the NFSv4 standard. There are two experimental implementations of NFSv4 ACLs for Linux: NFSv4 ACLs support for Ext3 filesystem[11] and the more recent Richacls which brings NFSv4 ACLs support for Ext4 filesystem.[12] As with POSIX ACLs, NFSv4 ACLs are usually stored as extended attributes on Unix-like systems.

NFSv4 ACLs are organized near-identically to the Windows NT ACLs used in NTFS.[13] NFSv4.1 ACLs are a superset of both NT ACLs and POSIX draft ACLs.[14] Samba supports saving the NT ACLs of SMB-shared files in many ways, one of which is as NFSv4-encoded ACLs.[15]

Active Directory ACLs

Microsoft's Active Directory Directory Service implements an LDAP server that store and disseminate configuration information about users and computers in a domain.[16] Active Directory extends the LDAP specification by adding the same type of access-control list mechanism as Windows NT uses for the NTFS filesystem. Windows 2000 then extended the syntax for access control entries such that they could not only grant or deny access to entire LDAP objects, but also to individual attributes within these objects.[17]

Networking ACLs

On some types of proprietary computer-hardware (in particular routers and switches), an access-control list provides rules that are applied to port numbers or IP addresses that are available on a host or other layer 3, each with a list of hosts and/or networks permitted to use the service. Although it is additionally possible to configure access-control lists based on network domain names, this is a questionable idea because individual TCP, UDP, and ICMP headers do not contain domain names. Consequently, the device enforcing the access-control list must separately resolve names to numeric addresses. This presents an additional attack surface for an attacker who is seeking to compromise security of the system which the access-control list is protecting. Both individual servers as well as routers can have network ACLs. Access-control lists can generally be configured to control both inbound and outbound traffic, and in this context they are similar to firewalls. Like firewalls, ACLs could be subject to security regulations and standards such as PCI DSS.

SQL implementations

ACL algorithms have been ported to SQL and to relational database systems. Many "modern" (2000s and 2010s) SQL-based systems, like enterprise resource planning and content management systems, have used ACL models in their administration modules.

Comparing with RBAC

The main alternative to the ACL model is the role-based access-control (RBAC) model. A "minimal RBAC model", *RBACm*, can be compared with an ACL mechanism, *ACLg*, where only groups are permitted as entries in the ACL. Barkley (1997)[18] showed that *RBACm* and *ACLg* are equivalent.

In modern SQL implementations, ACLs also manage groups and inheritance in a hierarchy of groups. So "modern ACLs" can express all that RBAC express, and are notably powerful (compared to "old ACLs") in their ability to express access-control policy in terms of the way in which administrators view organizations.

For data interchange, and for "high level comparisons", ACL data can be translated to XACML.[19]

See also

- Cacls
- Capability-based security
- C-list
- Confused deputy problem
- DACL
- Extended file attributes
- Role-based access control (RBAC)

References

1. RFC 4949 (https://tools.ietf.org/html/rfc4949)
2. *Elementary Information Security* by Richard E. Smith, p. 150
3. "Managing Authorization and Access Control" (https://technet.microsoft.com/en-us/library/bb457115.aspx). Microsoft Technet. 2005-11-03. Retrieved 2013-04-08.
4. "P.S.I. Pacer Software, Inc. Gnet-II revision 3.0" (https://books.google.com/books?id=KAUpSdv4AO4C). Communications. *Computerworld*. **18** (21). 1984-05-21. p. 54. ISSN 0010-4841 (https://www.worldcat.org/issn/0010-4841). Retrieved 2017-06-30. "The new version of Gnet-II (revision 3.0) has added a line-security mechanism which is implemented under the Primos ACL subsystem."
5. Grünbacher, Andreas. "POSIX Access Control Lists on Linux" (https://www.usenix.org/legacy/publications/library/proceedings/usenix03/tech/freenix03/full_papers/gruenbacher/gruenbacher_html/main.html). *Usenix*. Retrieved 12 December 2019.
6. wurtzkurdle. "Why was POSIX.1e withdrawn?" (https://unix.stackexchange.com/a/506641). *Unix StackExchange*. Retrieved 12 December 2019.
7. Trümper, Winfried (February 28, 1999). "Summary about Posix.1e" (https://web.archive.org/web/20080723061358/https://wt.xpilot.org/publications/posix.1e/). Archived from the original (https://wt.xpilot.org/publications/posix.1e/) on 2008-07-23.
8. "Red Hat Enterprise Linux AS 3 Release Notes (x86 Edition)" (https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/3/html/Release_Notes/as-x86/index.html). Red Hat. 2003. Retrieved 2013-04-08. "EA (Extended Attributes) and ACL (Access Control Lists) functionality is now available for ext3 file systems. In addition, ACL functionality is available for NFS."

9. "NFSv4 ACLs" (https://wiki.freebsd.org/NFSv4_ACLs). FreeBSD. 2011-09-12. Retrieved 2013-04-08.
10. "Chapter 8 Using ACLs and Attributes to Protect ZFS Files" (http://docs.oracle.com/cd/E19082-01/817-2271/ftyxi/index.html). Oracle Corporation. 2009-10-01. Retrieved 2013-04-08.
11. Grünbacher, Andreas (May 2008). "Native NFSv4 ACLs on Linux" (https://web.archive.org/web/20130620012339/http://users.suse.com/~agruen/nfs4acl/). SUSE. Archived from the original (http://users.suse.com/~agruen/nfs4acl/) on 2013-06-20. Retrieved 2013-04-08.
12. Grünbacher, Andreas (July–September 2010). "Richacls - Native NFSv4 ACLs on Linux" (https://web.archive.org/web/20130320080142/http://www.bestbits.at/richacl/). bestbits.at. Archived from the original (http://www.bestbits.at/richacl/) on 2013-03-20. Retrieved 2013-04-08.
13. "ACLs" (https://wiki.linux-nfs.org/wiki/index.php/ACLs#NFSv4_and_Windows_ACLs). *Linux NFS*.
14. "Mapping Between NFSv4 and Posix Draft ACLs" (https://tools.ietf.org/id/draft-ietf-nfsv4-acl-mapping-05.txt).
15. "vfs_nfs4acl_xattr(8)" (https://www.samba.org/samba/docs/current/man-html/vfs_nfs4acl_xattr.8.html). *Samba Manual*.
16. "[MS-ADTS]: Active Directory Technical Specification" (https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/d2435927-0999-4c62-8c6d-13ba31a52e1a).
17. Swift, Michael M. (November 2002). "Improving the granularity of access control for Windows 2000". *ACM Transactions on Information and System Security (Tissec)*. **5** (4): 398–437. doi:10.1145/581271.581273 (https://doi.org/10.1145%2F581271.581273). S2CID 10702162 (https://api.semanticscholar.org/CorpusID:10702162).
18. J. Barkley (1997) "Comparing simple role based access control models and access control lists (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.6366)", In "Proceedings of the second ACM workshop on Role-based access control", pages 127-132.
19. G. Karjoth, A. Schade and E. Van Herreweghen (2008) "Implementing ACL-based Policies in XACML (http://www.acsac.org/openconf2008/modules/request.php?module=oc_program&action=view.php&id=73)", In "2008 Annual Computer Security Applications Conference".

Further reading

- Rhodes, Tom. "File System Access Control Lists (ACLs)" (https://www.freebsd.org/doc/en/books/handbook/fs-acl.html). *FreeBSD Handbook*. Retrieved 2013-04-08.
- Michael Fox; John Giordano; Lori Stotler; Arun Thomas (2005-08-24). "SELinux and grsecurity: A Case Study Comparing Linux Security Kernel Enhancements" (https://web.archive.org/web/20120224213801/http://www.cs.virginia.edu/~jcg8f/GrsecuritySELinuxCaseStudy.pdf) (PDF). University of Virginia. Archived from the original (https://www.cs.virginia.edu/~jcg8f/GrsecuritySELinuxCaseStudy.pdf) (PDF) on 2012-02-24. Retrieved 2013-04-08.
- Hinrichs, Susan (2005). "Operating System Security" (https://web.archive.org/web/20120304040752/http://www.cs.uiuc.edu/class/fa05/cs498sh/seclab/slides/OSNotes.ppt). *CyberSecurity Spring 2005*. University of Illinois. Archived from the original (http://www.cs.uiuc.edu/class/fa05/cs498sh/seclab/slides/OSNotes.ppt) on 2012-03-04. Retrieved 2013-04-08.
- Mitchell, John. "Access Control and Operating System Security" (https://crypto.stanford.edu/cs155old/cs155-spring03/lecture9.pdf) (PDF). Stanford University. Retrieved 2013-04-08.
- Clarkson, Michael. "Access Control" (https://www.cs.cornell.edu/courses/cs513/2007fa/NL.accessControl.html). Cornell University. Retrieved 2013-04-08.
- Klein, Helge (2009-03-12). "Permissions: A Primer, or: DACL, SACL, Owner, SID and ACE Explained" (http://helgeklein.com/blog/2009/03/permissions-a-primer-or-dacl-sacl-owner-sid-and-ace-explained/). Retrieved 2013-04-08.
- "Access Control Lists" (http://msdn.microsoft.com/en-us/library/aa374872(VS.85).aspx). MSDN Library. 2012-10-26. Retrieved 2013-04-08.

- "How Permissions Work" (https://technet.microsoft.com/en-us/library/cc783530(WS.10).aspx). Microsoft Technet. 2003-03-28. Retrieved 2013-04-08.
