International Journal of Multimedia and Ubiquitous Engineering
Vol. 1, No. 3, September, 2006

Design and Implementation of File Access Control Module for Multi User Operating System Using on Windows

Bong-keun Lee[1], Yoon-ae Ahn[2], Wan Seung Jang[3], In-bae Oh[4] ,Sang-Joe Youk[5] , Yong-tae Kim[5] , Gil-cheol Park[5]
[1]Department of Computer Science , Chungbuk University
e-mail : bong9065@hanmail.net
[2]Department of Multimedia Engineering , Chungju National University
e-mail : yeahn@cjnc.ac.kr
[3]IDI Co. Ltd., DunSan-Dong, Seo-Gu, Daejeon, Korea
wsjang@idigroup.co.kr
[4]Department of Internet Information, Juseong College
e-mail : iboh@jsc.ac.kr
[5]Department of Multimedia Engneering, Hannam University
e-mail : {youksj,ky7763,gcpark}@mail.hannam.ac.kr

Abstract With the rapid development of information sharing through network, IT system is exposed to various threat and security incident are became a social problem. As a countermeasure, various security systems are been using such as IDS, Firewall, VPN etc. But, expertise or expert is required to handle these security systems, so it is not easy for ordinary users to operate them. This paper has developed the design and implementation of file access control module for secure operating system which can exclude from attacks on the operating system rather than detection of attacks. The module implemented in this paper is based on Windows and has effect on access control, integrity and non-repudiation for a file with an access control over files on the Windows based OS that are used by multi-user.

Keyword: Access control , ACL , ACE

1. Introduction

Today, information system from rapid development of information communication technology offers incredible convenience and speediness, but there are a lot of problems to cause damages by that.

In order to prevent invasion incidents and give effective confrontations, there have been developing several information protection techniques such as invasion interception technique, invasion detection technique, and so on. However, these techniques show a good result about prevention and detection on well-known weaknesses, but it does not confront nicely with unknown ones. It also stops main service immediately when most invasion incidents happen. In this case, it could cause a very important problem. So, an invasion incident confrontation with unknown weaknesses or attacks is required [1].

If the invasion detection system or the invasion interception system was the invasion confrontation through the network, the file access control proposed by this paper is the confrontation with invasions of in the system. This paper has done the design and implementation of file access control module that can be used on the security operating system, so it enables invasion correspondence by preventing new invasion types unknown from modifying and eliminating files. Even if the invader achieved Administrator's accounts authority, Administrator's authority, the access would exclude the authority used on the security OS for the important system files or the files that require security.

This paper is consisted as followed. Chapter 2 tells the file policy based on Windows and chapter 3 describes the design and implementation of module proposed by this paper. Finally, there are conclusions in chapter 4.

2. File Policy based on Windows

2.1 Security Model based on Windows

TCSEC(Trusted Computer Security Evaluation Criteria) is the evaluation guide that was printed by NCSC(National Computer Security Center of America) in 1985 for the security of computer system and defined 6 basic requirements in order to evaluate the security of computer system effectively. And it also presented 7 evaluation levels which satisfied the each [2].

The security model supported under the Windows OS supports up to C2 level required by Pentagon like UNIX does.

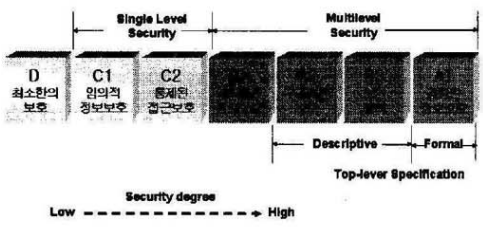


Fig. 1. Evaluation level of TCSEC

2.2 Security Descriptors

Security object is differed the use authority up to user by according to security establishment. If it makes the access control like this possible each object, a security establishment has to be remembered inside object itself. For instance, if there is security establishment remembered that User1 can only read in Admin.txt and User2 can both read and write, this security establishment information will be called security descriptors. Security descriptor is a kind of structure that remembers security establishment information and it is consisted of information as followed.

Possessor's SID: SID of the possessor who possesses an object. The possessor has all authorities over the object without reference to security descriptor's DACL information.

Possessor's group SID: It is the ID of group which possessor belongs to and this information exists for compatibility with other OS, but no big deal on the e Windows.

DACL: It is the list of authority information of each user. It remembers who can read this object or be denied to write and so on. This is the key information of security descriptor.

SACL: When someone accesses the object, it remembers inspection information to record. It includes information that leaves a record on the event log if someone does some action to this object.

2.3 Security Identifiers

A security identifier means the identifier used as identifying value which only identifies objects like user or group on Windows NT and 2000. SID which was allocated to user is a part of access allowance token which follows the process operated or the action tried by user or group. SID includes version, domain information, user information, etc, and is consisted of binary format.

Contents of SID

- The security descriptor of user and group
- The ID authentication of 48 bit
- Standard of modification
- Various lower rank authority values

2.4 Access Token

In order to protect from the user who are not allowed to security object, there should be a comparison between the information about the person who wants to access and security descriptor. Here let's call the information of the user access token, and it is an important element which composes of security system of Windows OS with security descriptor.

2.5 ACL

Access Control List(ACL) is an array of Access Control Entry(ACE) which is a piece of individual security information. Through comparing of SID and ACL, the decision whether the access to the object is possible or not will be made[2],[3].

Discretionary Access Control List(DACL): the list that permits or denies the use to specific user or group on the security descriptor of the object.

System Access Control List(SACL): the list that appoints events which have to be inspected on the security descriptor of the object.

2.6 ACE

ACL is an array of ACE and ACE is an array entry of ACL. ACL can have none of ACE or several ACEs. ACE is an entry which has real information and security descriptor or ACL are just a dish to put ACE. If someone wants to know or change the security establishments of the security object, it must read ACE and edits. There are 6 types of ACE in Windows 2000, theses 6 types of ACE are general ACE which can be applied to all objects like file system abject, Active Directory object, and so on, and object ACE which can be applied to only Active Directory object. General ACE and object ACE are basically the same but control density that is supplied to inheritance and object access is only the difference[4].

2.7 Access Authority

ACE information appoints whether it permits or denies which access authority to whom. Access authorities to each security object are various up to object. There are common access authorities like read, write, delete, etc, and individual access authorities that are up to object like add, attribute change, move, copy, end, question, priority change and so on. In order to express the combination of these various access authorities, an access mask, 32 bit integer, will be used. Each bit of access mask accommodates access authority one by one, so access mask can be called a structure of bit field type which has access authorities as members

3. Design and Implementation

3.1 Development Environment

The module implemented in this paper has used Windows 2000 Professional as basic operation system, also for the test, it has another account which is distinguished from Administrator. The program is called ACL(Access Control List) and the name of DLL library is ACLLib.

The function needed by module is made out DDL by using Visual C++ 6.0 and GUI by using Delphi 6.0.

3.2 Design and Implementation

To read and write security object of files on the Windows environment is possible to use by using aclapi.

It lets security descriptor be in the file in the NTFS file system, and a security descriptor is consisted of 2 ACL(Access Control List). ACL is an array of ACE (Access Control Entry) which is an individual security piece.

A security descriptor has two of ACL; one is DACL(Discretionary ACL), the list of access authority, and the other is SACL which controls inspection record making. DACL is consisted of several ACEs and each ACE expresses the information that who has which authority on this object.

This paper establishes the security policy on each file by reading, writing and modifying ACE of these each file. Next shows a part of the function that read ACE.

```
if (GetSecurityInfo(hFile, SE_FILE_OBJECT,
OWNER_SECURITY_INFORMATION |
DACL_SECURITY_INFORMATION, &pOwner, NULL,
```

```
&pDacl, NULL, (LPVOID *)&pSD) !=
ERROR_SUCCESS)
        return -1;

CloseHandle(hFile);

//owner's information
cbName = 0;
cbDomain = 0;
LookupAccountSid(NULL, pOwner, NULL,
&cbName, NULL, &cbDomain, &peUse);

Name = (char *)malloc(cbName);
Domain = (char *)malloc(cbDomain);

LookupAccountSid(NULL, pOwner, Name,
&cbName, Domain, &cbDomain, &peUse);

strcpy(Entry->Name, Name);
strcpy(Entry->Domain, Domain);

free(Name);
free(Domain);

nAce = 0;
//DACL information
GetExplicitEntriesFromAcl(pDacl, &nAce,
&pEntry);
Entry->Count = (int)nAce;

for(i = 0; i < (int)nAce; i++) {
   cbName = 0;
   cbDomain = 0;
   LookupAccountSid(NULL,
pEntry[i].Trustee.ptstrName, NULL, &cbName,
NULL, &cbDomain, &peUse);

   Name = (char *)malloc(cbName);
   Domain = (char *)malloc(cbDomain);

   LookupAccountSid(NULL,
pEntry[i].Trustee.ptstrName, Name, &cbName,
Domain, &cbDomain, &peUse);

   strcpy(Entry->AceName[i], Name);
   Entry->AccessMode[i] =
AccessModeToDword(pEntry[i].grfAccessMode);
   Entry->Permission[i] =
pEntry[i].grfAccessPermissions;
```

The function above can achieve owner's information, domain information, and DACL information of each file.

Next is a part of the function that writes ACE object of file.

```
GetNamedSecurityInfo(FileName,
                     SE_FILE_OBJECT,

DACL_SECURITY_INFORMATION,
                     NULL,
                     NULL,
                     &ExistingDacl,
                     NULL,
                     &psd);

BuildExplicitAccessWithName(&explicitaccess,
                            Trustee,
                            AccessMask,
                            option,
                            InheritFlag);

// add specified access to the object
SetEntriesInAcl(1,
            &explicitaccess,
            ExistingDacl,
            &NewAcl);

// apply new security to file
SetNamedSecurityInfo(FileName,
                SE_FILE_OBJECT, // object
type
                DACL_SECURITY_INFORMATION,
                NULL,
                NULL,
                NewAcl,
                NULL);
```

The source above is exported as a function as followed and used through DLL library.

```
extern "C" __declspec(dllexport) int
GetSecurity(char *filename, ACLEntry
*Entry);

extern "C" __declspec(dllexport) int
SetSecurity(char *FileName, char
*Trustee, DWORD AccessMode, DWORD
Permission);
```

Through DLL implemented above, ACL program is made out and the program which consists of GUI screen is made out Delphi. Next shows how to call the function that reads ACE information on the Delphi.

```
if GetSecurity(PChar(FolderName +'\' +
FileName), @Entry) < 0 then Exit;

Next shows how to call the function
that writes ACE information on the
Delphi.
Ret :=
SetSecurity(PChar(FileNameEdit.Text),
PChar(TrusteeEdit.Text), Access,
Permission);
```

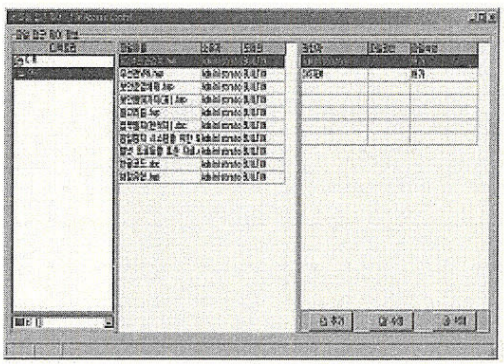


Fig. 2. ACL Program

Figure 2 shows if a file is selected by choosing driver

and directory, it is able to see ACE object of each file. And it is also possible to add, modify and delete ACE object of each file.

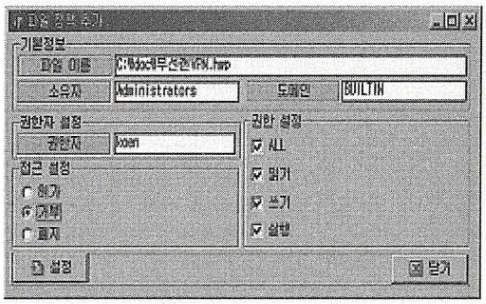


Fig. 3. Screen of Object Append

Figure 3 is the screen which shows adding ACE object of the file. For the test, it has made koen, a new account, which is not Administrator and established a policy for the file. The established authority is that it denies the account which corresponds to the file, Administrator, as above, all authorities are established as the account that has a correspondent file 'Administrator', but the authority 'no Administrator'.

After success the establishment, it accessed to correspondent file after log-in as Administrator on the Windows.

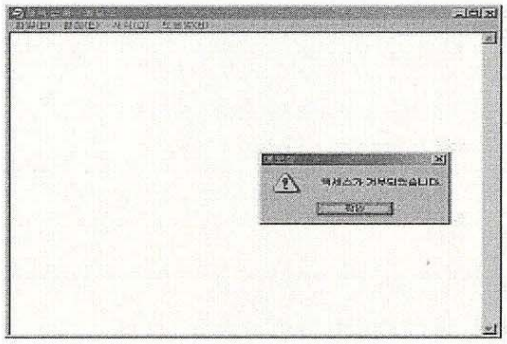


Fig. 4. Screen of Access denied

The message window that says 'access denied' can be seen when it accesses the correspondent file.

4. Conclusions

According to rapid use of computer and Internet as the rapid development of information network of the whole world, the convenience of information management has been enlarging. Meanwhile general users who are poor at protecting computer have met various problems of information security.

The source of system and important data have been threatened by intrusion through the Internet and sometimes even harmed fatally, so the need of security service on the Internet is required badly[3]. The need of this security service happens to be out of system invasion and it can be accommodated by security operation system.

This paper has designed and implemented the file policy control module that is used in the security OS and let it be used in the security operating system.

The module implemented in the paper controls the file policy for the security operating system, so it makes the system security strong.

References

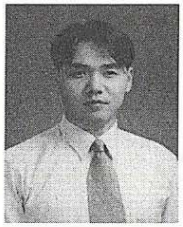
[1] Chris, Prosise. & Kevin, Mandia., "Incident Respinse: Investigating Computer Crime", McCraw-Hill, 2001, pp.371

[2] San Jose, "Common Criteria Solutions", Security Lab, http://www.fact-index.com/t/tc/tcsec.html.

[3] Woo-young Soh, "Computer Network Security", Green, pp.603-606

[4] Chul-woo Shin, "Windows 2000 Server", http://www.youngjin.com

Authors

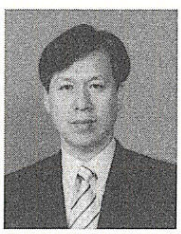


Bong-keun Lee

Received a B.S. degree in computer Science engineering from Han-Nam University , Korea, 1997, and M.S. degree in computer Science engineering from Han-Nam University, Korea, 1999. His research interests include Information Security, Mobile & Ubiquitous Web Service platform, Sensor Database, Data mining, Moving object Database . He is a Member of KIPS.

Yoon-ae Ahn

Received a B.S. degree in Computer Science Engineering from Han-Nam University, Korea, 1993, and M.S. degree in Computer Science from Chungbuk National University, Korea, 1996 and Ph D. degree in Computer Science from Chungbuk National University, Korea, 2003. From 2003 to 2005 she joined the faculty of Cheongju National College of Science & Technology, Korea. In 2006 she joined the faculty of Chungju National University, Korea, where she is currently a professor in Department of Multimedia Engineering. Her research interests include Moving Object Database, Mobile and Embedded System, Sensor Database. She is a Member of KISS, KIPS, KMMS, KOCIES.



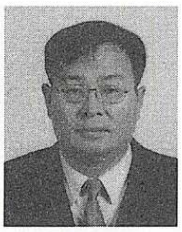
Wan-Seung Jang

1988-1982 HanNam Univ.(BA)
1996-1999 HanNam Univ. Graduate School(MA)
1993- present IDI Co., LTD., President

In-bae Oh

Received a B.S. degree in Computer Science from Han-Nam University 1987, and M.S. degree in Computer Science from Kon-Kuk University 1989 and Ph D. degree in Computer Science from Chungbuk National University 2004.

In 1992 he joined the faculty of Juseong College where he is currently a professor in Department of Internet Information. His research interests include XML Database, Mobile Database, Information Security, Virtual Reality. He is a Member of KISS, KIPS, KMMS, KOCIES.



Sang-jo Youk

Received a B.S. degree in Computer Science Engineering from Han-Nam University, Korea, 1990, and M.S. degree in Computer Science engineering from Han-Nam University, Korea, 1994 and Ph D. degree in Computer Science engineering from Han-Nam National University, Korea, 2004. Concerning and Interesting Recent research area Mobile & Ubiquitous Web Service platform, Real-time Multimedia Communication, Security Engineering.

Yong-tae Kim

Received a B.S. degree in Computer Science Engineering from Han-Nam University, Korea, 1984, and M.S. degree in Computer Science from Sung-Sil University, Korea, 1988. Engineering Director at Galim Corp. in 2002 until 2005.Concerning and Interesting Recent research area Mobile & Ubiquitous Web Service platform, Real-time Multimedia Communication, Security Engineering.

Gil-Cheol Park

Received M.S In computer science in Soong-Sil University in1983. And received Ph.D in Information Engineering from SungKunKwan University, Korea, in1998, respectively. He is currently an Professor in Department of Multimedia Engineering at Hannam University. He research interests include multimedia and mobile communication, network security.