DEVGURU

Device Driver Security에 대한 고찰 Ⅱ

본 칼럼은 Windows 2000/XP에서 Device Driver에 대한 보안 문제를 기술하는 연재기사로서 파트 2에 해당한다. 지난 기사에 이어 좀더 구체적인 내용을 다루고자 한다. 지난 연재에서 마지막으로 살펴본 자료구조가 Access Control Lists(ACLs)이 였는데 이 부분에 대해서 먼저 보기로 하자.

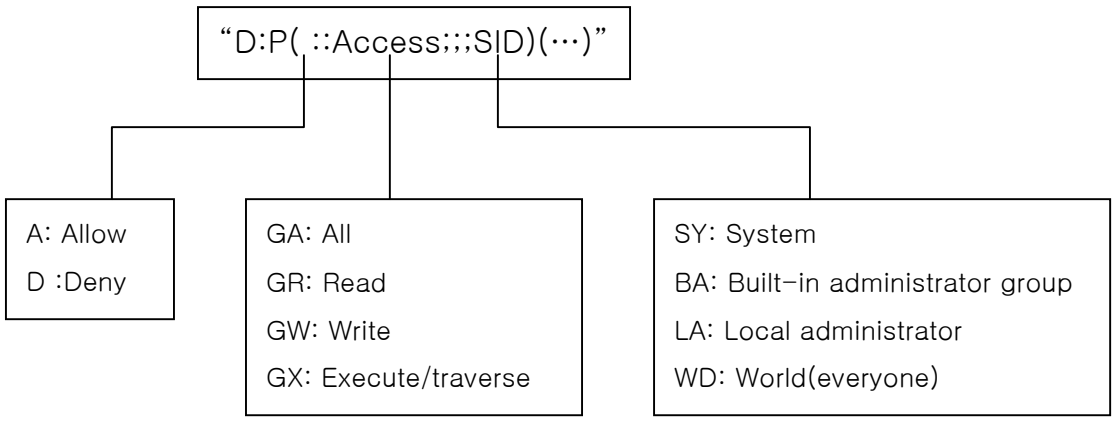
어떤 object의 Security Descriptor는 object를 만드는 함수에 의해 생성되어진다고 언급하였다. 따라서 Driver에서 IoCreateDevice or IoCreateDeviceSecure함수를 사용하여 device object를 만들 때 system은 Device object에 대해 security descriptor를 만들고 object에 대해 ACLs를 Set한다. 그리고 대부분의 디바이스들에 대해서는 ACLs는 INF file에 명시되어 있다.

Device object에 ACL를 set하는 방법은 다음의 3가지가 있다.

a. set in the default security descriptor for its device type.
b. RtlCreateSecurityDescriptor and RtlSetDaclSecurityDescriptor함수를 사용한다.
c. Specified in Security Descriptor Definition Language(SDDL) in the Device's INF file or in a call to the IoCreateDeviceSecure routine.

모든 새로운 drivers은 inf file에 ACE를 명시하는 SDDL방식을 사용하여야 한다.

SDDL은 SDK에 자세히 언급이 되어있지만 간단하게 설명하면 다음과 같다.



즉, 다음의 SDDL string은
"D:P(A;;GA;;;SY)(A;;GR;;;WD)"
System은 모든 access를 할 수 있고 everyone은 단지 read access만 가능하게 한다.
이렇게 object에 set된 ACL를 가지고 어떻게 Security check이 이루어지는지를 알아보자.
어떤 process가 어떤 object에 대한 접근을 요구하면 system은 object의 ACL과 caller's access token내의 SID를 비교한다.

다음의 예는 Jim's process가 특정 file을 open할 때 어떤 security check이 일어나는지를 설명한다.

Allow	Accounting	Write, Delete
Allow	Sales	Append
Deny	Legal	Append, Write, Delete
Allow	Everyone	Read

File's ACL

User	Jim(S-1-5-21…)
Group	Accounting(S-1-5-22…)
Group	Legal(S-1-5-23…)
Group	Everyone(S-1-1-0)

Security Token for Jim's process

File object의 ACL과 Access Token의 비교는 다음과 같은 순서로 이루어진다.

a. User Jim에 해당하는 File's ACL의 entry를 조사한다. Jim에 대한 Entry가 없으므로 다음으로 진행한다.
b. Accounting group에 해당하는 File's ACL의 entry를 조사한다. Accounting에 대해서 Write, Delete가 allow되어 있으므로 Jim's process는 file에 대해 접근할 수 있다.
c. comparison은 stop.

참고로 ACL's ACEs 들은 항상 top-down순으로 checking이 이루어지고 checking은 첫번째 match되는 것이 발견되면 stop한다. 그리고 항상 Deny ACE를 먼저 두자.

다음의 예는 Restricted Token에 대한 ACL checking이 어떻게 이루어지는가

를 나타낸다.

Allow	Accounting	Write, Delete
Allow	Sales	Append
Deny	Legal	Append, Write, Delete
Allow	Everyone	Read

File's ACL

User	Jim(S-1-5-21…)	Deny
Group	Accounting(S-1-5-22…)	Deny
Group	Legal(S-1-5-23…)	Deny
Group	Everyone(S-1-1-0)	

Security Token for Jim's process

시스템은 ACL과 Restricted token에 대한 비교를 이전의 non-restricted token과 같은 방식으로 한다. 단, 다른 점은 restricted token내에 있는 denial SID는 ACL내에 있는 Deny ACE와만 match 될 수 있다.
구체적으로 File object의 ACL과 restricted Token의 비교는 다음과 같은 순서로 이루어진다.

a. User Jim에 해당하는 File's ACL의 entry를 조사한다. Jim에 대한 Entry가 없으므로 다음으로 진행한다.
b. Accounting group에 해당하는 File's ACL의 entry를 조사한다. Accounting에 대해서 비록 ACE가 존재하지만 deny가 아니므로 다음으로 진행한다.
c. Legal group SID에 대해서 File's ACL은 deny entry를 가지고 있으므로 match가 이루어진다. 따라서 process는 해당 file에 대해서 write, append, delete를 수행할 수 없다.
d. comparison은 stop.

이상으로 두 가지 access token에 대한 access comparison이 어떻게 이루어 지는가를 보았다. 이 과정에서 사용된 ACL에 대해 좀더 알아보자
ACL은 어떤 object에 연관된 security 를 제어하기 위해 o/s에 의해 생성된 ACE들의 리스트이다. 이 ACE의 구조체를 한 번 살펴보자.

ACE Size	ACE flags	ACE type
ACCES_MASK		
Security Identifier		

Access Control Entry

주요 필드 중 ACE type과 ACCESS_MASK를 한 번 살펴보자.
ACE type은 다음과 같다.

- ACCESS_ALLOWED_ACE_TYPE – this indicates that the ACE specifies access rights that are to be granted to the specific SID.
- ACCESS_DENIED_ACE_TYPE – this indicates that the ACE specifies access rights that are to be denied to the specific SID.
- SYSTEM_AUDIT_ACE_TYPE – this indicates that the ACE specifies auditing behavior.
- SYSTEM_ALARM_ACE_TYPE – this indicates that the ACE specifies
- alarm behavior.
- ACCESS_ALLOWED_COMPOUND_ACE_TYPE – this indicates that the ACE is tied to a particular server and the entity it is impersonating.

ACCESS_MASK 필드에 대해 살펴보자
ACCESS_MASK는 특별한 type의 object에 대한 access 정보를 기술하기 위해 사용한다. 32bits의 값으로 크게 두 부분으로 구분할 수 있는데 상위 16bits는 Windows NT의 모든 object에 공통적인 access 정보를 가지고 하위 16bits는 object type 에 따라 다르게 적용되는 값을 가진다.

Generic Rights	Standard Rights	Specific Rights

Generic Rights는 programmer가 쉽게 사용하기 위한 값이며 다음의 값 중 하나를 가질 수 있다.

- *Read* – the ability to “read” the information maintained by the object
- *Write* – the ability to “write” the information maintained by the object
- *Execute* – the ability to “execute” or alternatively “look into” the object.
- *All* – Read, Write, and Execute privileges.

Programmer명시한 Generic Rights는 O/S에 의해 다음의 Standard Rights값으로 변환 된다.

- *Delete* – the ability to delete the object
- *Read control* – the ability to read the security information for the object
- *Write DAC* – the ability to modify the "discretionary access control" for the object. We discuss discretionary access controls later in this article (Part II)
- *Write Owner* – the ability to modify the "owner" for the object. We discuss the concept of ownership later in this article (Part II)
- *Synchronize* – the ability to use the object for synchronization. This allows the caller to "wait" on the given object.

Specific Rights는 object type에 따라 상이한데 file object인 경우의 예는 다음과 같은 값을 가질 수 있다.

- FILE_READ_DATA – the right to read from the file object.
- FILE_LIST_DIRECTORY –the right to enumerate the contents of the directory.
- FILE_WRITE_DATA –the right to write data to the file.
- FILE_APPEND_DATA – the right to append data to the end of the file.
- FILE_ADD_SUBDIRETORY – the right to create subdirectories.
- FILE_CREATE_PIPE_INSTANCE – the right to create an instance of a named pipe.
- FILE_READ_EA – the right to read the contents of the "extended attributes".
- FILE_WRITE_EA – the right to write the contents of the "extended attributes".
- FILE_EXECUTE – the right to execute the file.
- FILE_TRAVERSE – the right to access specific elements contained within a directory. Note that this differs from FILE_LIST_DIRECTORY because to traverse you need only know the name of the specific element being accessed.
- FILE_DELETE_CHILD – the right to delete an entry within a directory.
- FILE_READ_ATTRIBUTES – the right to read the attributes of the file.
- FILE_WRITE_ATTRIBUTES – the right to modify the attributes of the file.

이상으로 ACL에 대해서 살펴보았고 process가 어떤 object를 접근하고자 할 때 token과 ACL을 이용하여 어떻게 ACCESS check이 이루어지는지를 간단하게 보았다. 다음 번의 칼럼에서는 File을 create할 때 구체적으로 object manager, I/O Manger가 Driver가 각각 어떤 security checking을 담당하는지에 대해서 알아 보겠다.