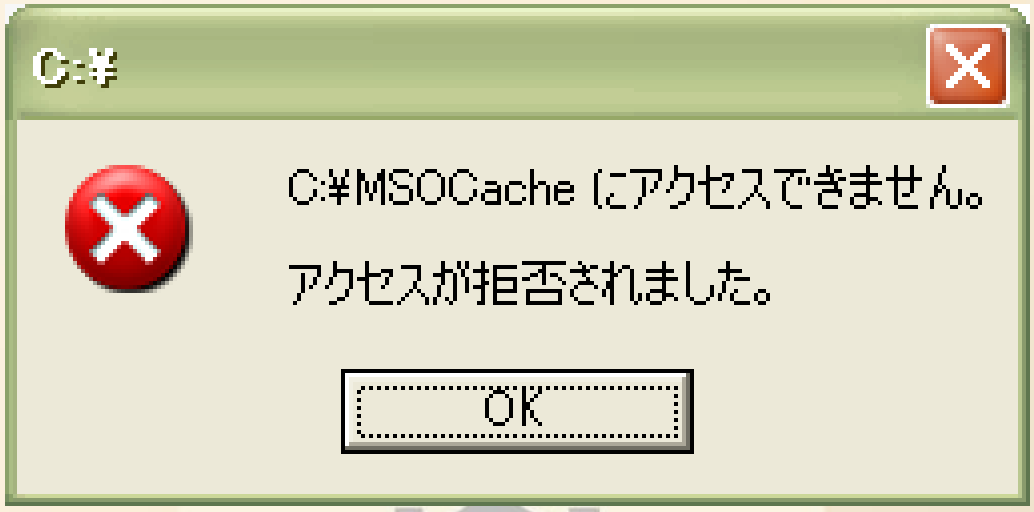
Windows Access Control Model

by ちゃっぴ

はじめに

こんな例外発生してハマった経験ありませんか？

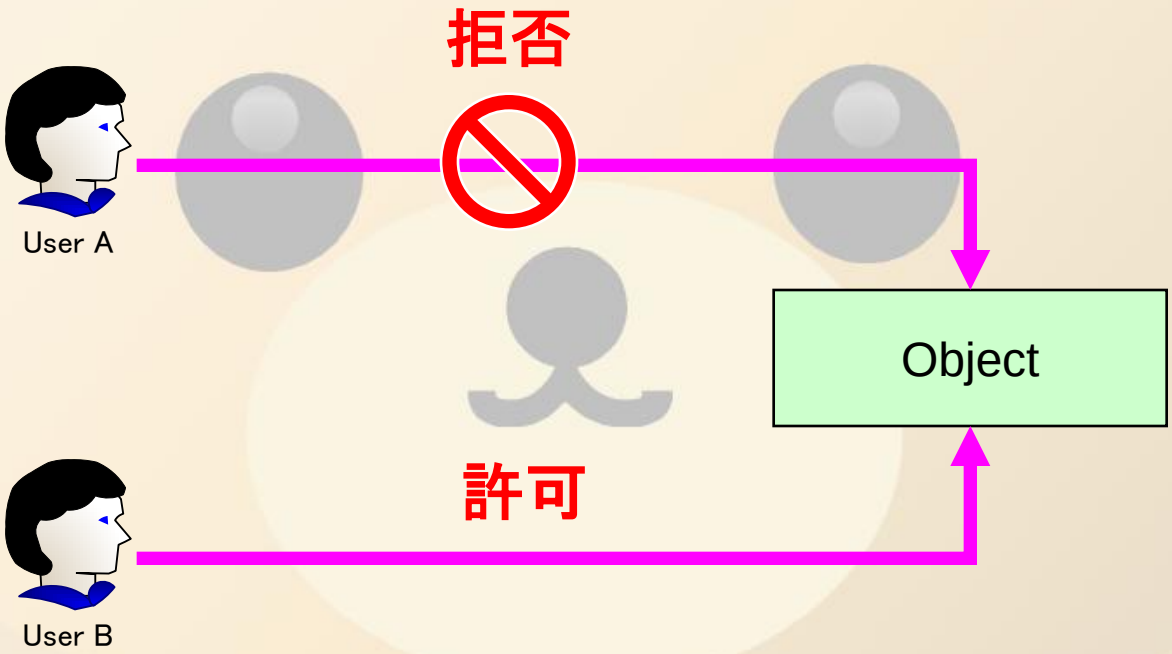


とりあえず、しくみがわかっていないことには適切な対処ができません。

ということで、しくみをちょっくら勉強しましょう。

Access control とは？

利用者が対象 (object) を扱えるかを制御すること



Access control がなぜ必要か？

- 機密情報を保護するため
  Access control が無いとすべての人がすべての情報を扱える。
  → 情報漏えい等 security 事故の発生

- 不用意な操作による人為的な障害の防止
  Access control が無いとちょっとしたことで system を壊せます。

- 攻撃されたときの被害の最小化
  Access control が無いと攻撃し放題。

ということで、access control が全く無いと非常に怖いことになります。

前提条件

対象は Windows Server 2003 以前の NT系 OS。

Windows Vista 以降で拡張されたものに関しては扱いません。

Windows 9x 系の OS はそもそもまともな access control が存在しないため対象外です。

実は Windows Server 2003 以前でももうちょっと複雑です。Session の都合上、最低限のことしか書いていませんのでその点よろしく。

ちょいと予備知識 (SID について)

Account (user, group) は Windows 内部では ID を使って管理している。
その ID を SID（Security Identifier）と呼ぶ。

SID の例

SID	Name
S-1-1-0	Everyone
S-1-3-0	Creator Owner
S-1-5-18	Local System
S-1-5-19	Local Service
S-1-5-20	Network Service
S-1-5-21-xxxxxxxxx-xxxxxxxxx-xxxxxxxxx-512	Domain Admin

Well-known SID と呼ばれる

この部分は domain (computer) 固有の ID が付与される

Domain Admin を表す

どんなものが制限できるの？

Windows 上では下記のものが制限できます。

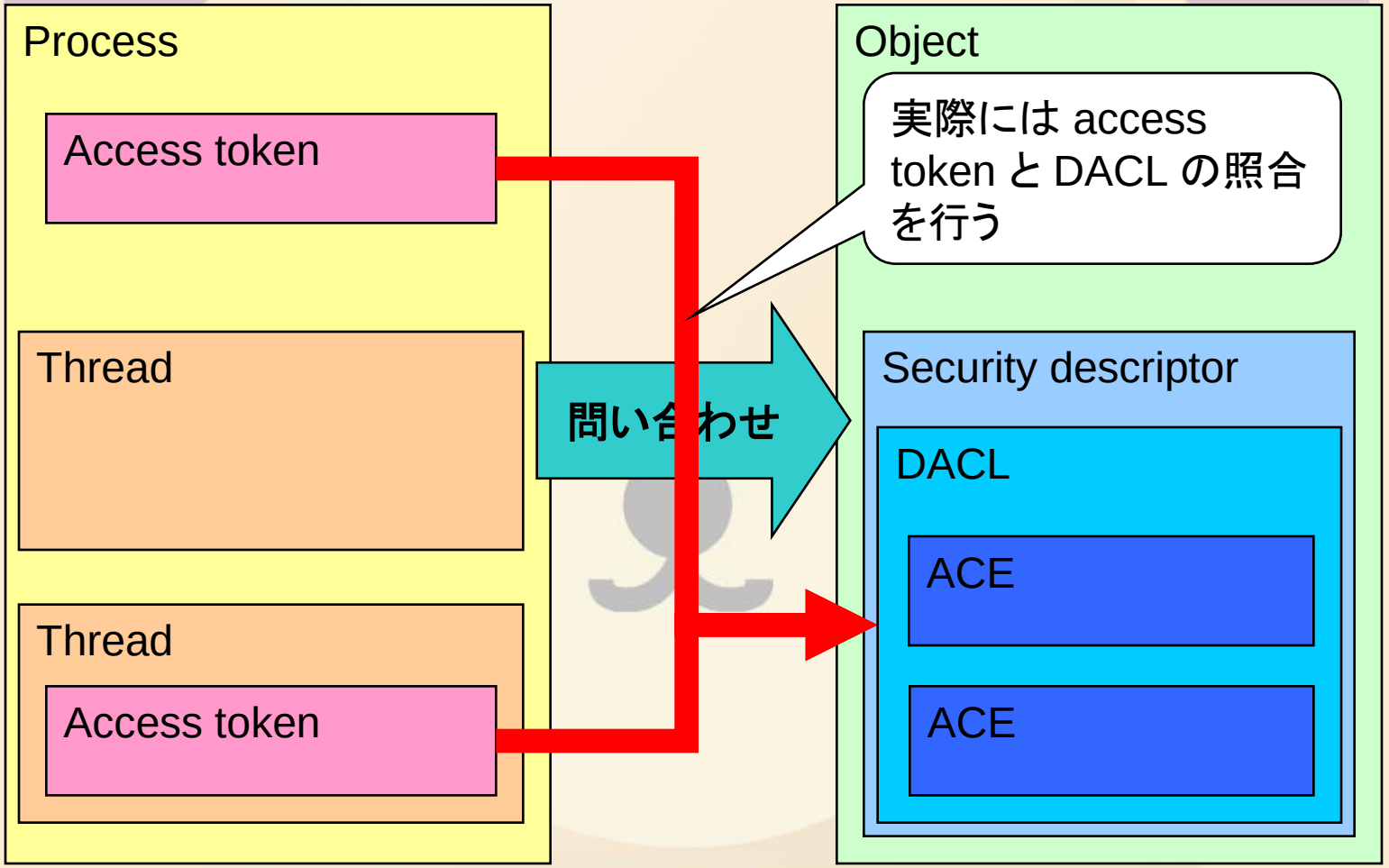
- Files, directories
- Registry key
- Network shares (Network 共有)
- Printers
- Job objects (Task Scheduler)
- Windows Services
- Directory Service objects
- Process, thread
- Named pipes (名前つきパイプ)
- File-mapping objects (共有メモリ)
- Interprocess synchronization objects (events, mutexes, semaphores, and waitable timers)
- Access token

その他いろいろ

実はほとんどのものが制限できちゃうんです。

詳しくは MSDN の「Securable Objects」を参照

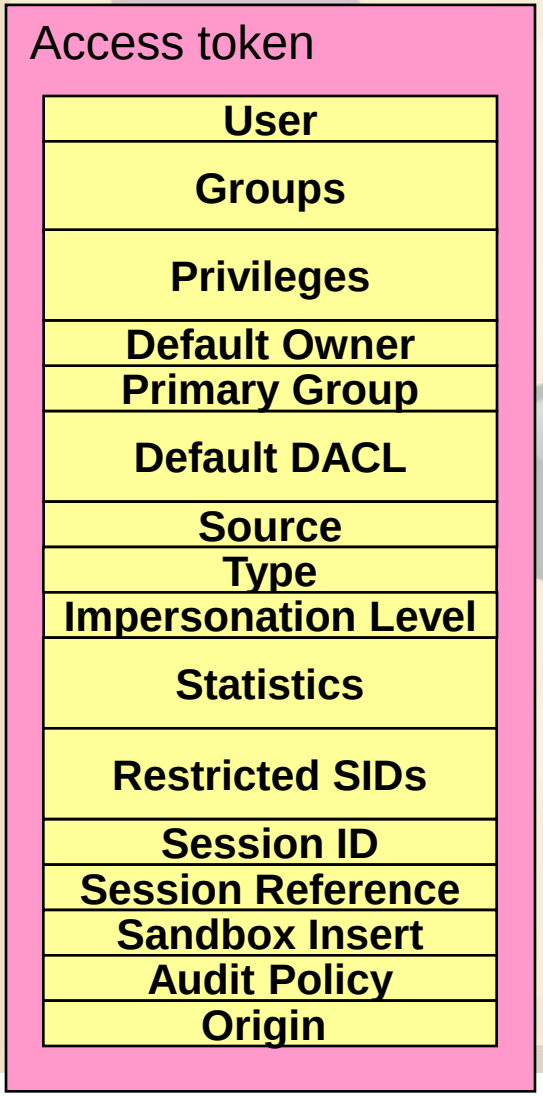
Windows access control のしくみ



Access Tokens

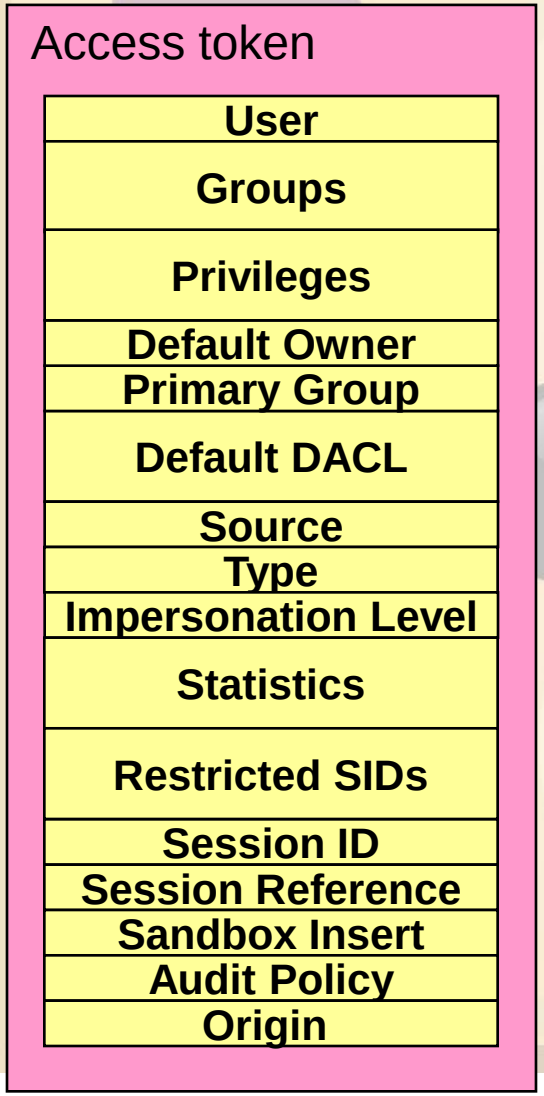
- Logon した user の security 情報が格納されているもの
- Process または thread の内部に保持されている
- Primary と impersonation (偽装) の2種類に分かれる
  - Thread が持つ access token が実は Impersonation token
  - 偽装していない thread は access token を持たない
    → Process が持つ access token (primary token) が使われる
- 基本的に読み取り専用
  - Logon 時に生成され原則変更できない
  - 例外としてより制限を課すことは一応できる
    → Restricted SIDs

Access token の内部構造 1



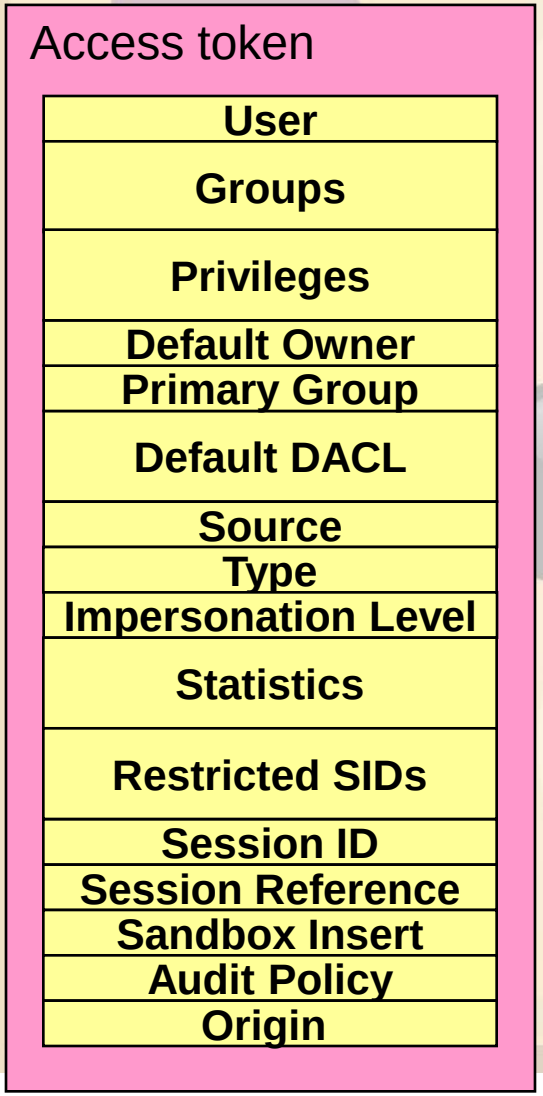
- User
  Logon した user の SID
- Groups
  Logon した user が所属する group SID の list
- Privileges
  Logon した user が所持している特権 (privileges)
- Default Owner
  Object を生成するときに使用される default owner
- Primary Group
  Logon した user が所属する primary group SID
  ※ POSIX sub system でのみ使用
- Default DACL
  Object を生成するときに使用される default DACL
- Source
  Access token の生成元
  (Session Manager, LAN Manager 等)

Access token の内部構造 2



- Type
  Access token の種類 (primary or impersonation)
- Impersonation Level
  偽装の種類
- Statistics
  いろんな情報
- Restricted SIDs
  Group SID を制限するために使用
- Session ID
  Session ID
- Session Reference
  予約
- Sandbox Insert
  SANDBOX_INERT flag 用

Access token の内部構造 3



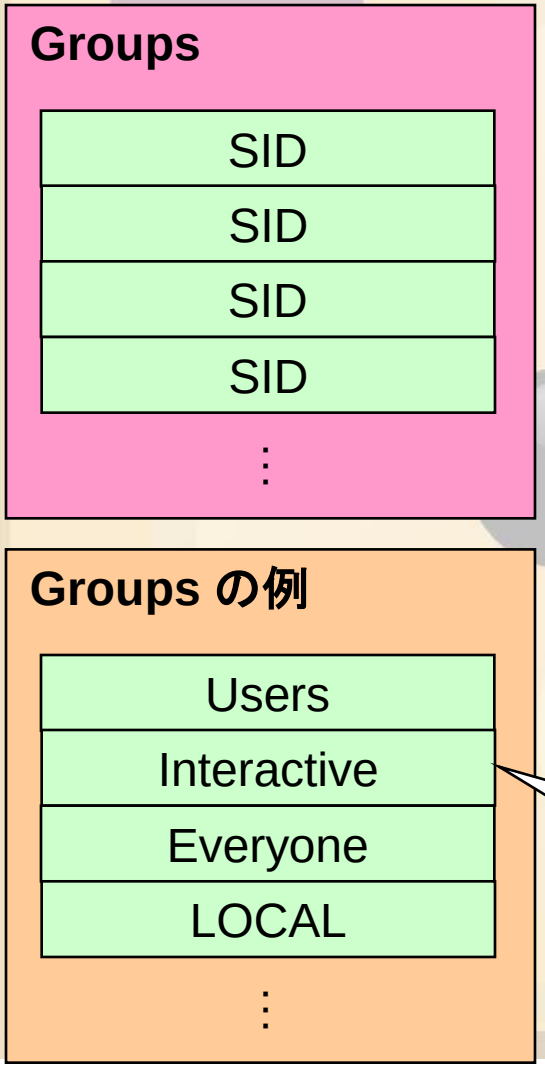
- Audit Policy
  予約
- Origin
  元の logon session ID (Windows Server 2003 以降)

だらだらと長く書きましたが、おぼえてほしいのはこれ以降で説明する User、Groups および Privileges です。

それ以外はとりあえずおぼえなくとも大丈夫です。

んなんじゃ満足できないという人は MSDN で該当の箇所 (GetTokenInformation 等) を読んでください。

Groups (Access Tokens)

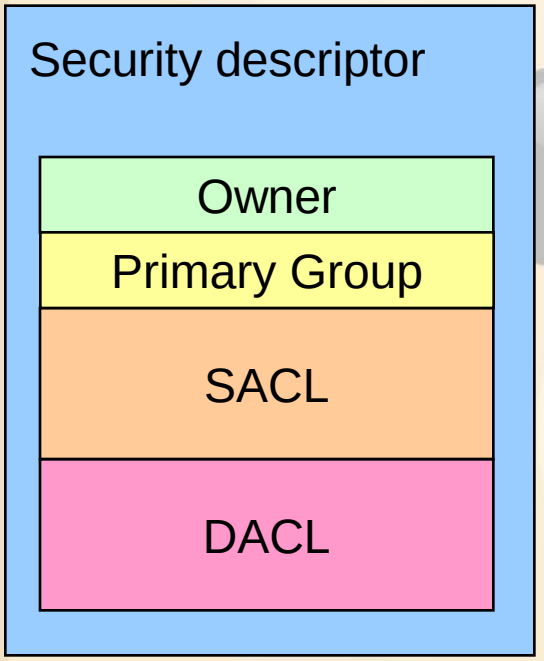


- 対象の user が所属している group の SID が展開されて格納される
- 格納される group は user の管理画面で確認できる静的なもののほか logon type 等によって動的に割り当てられるものも含まれる

  Interactive, Network, Batch, Service 等

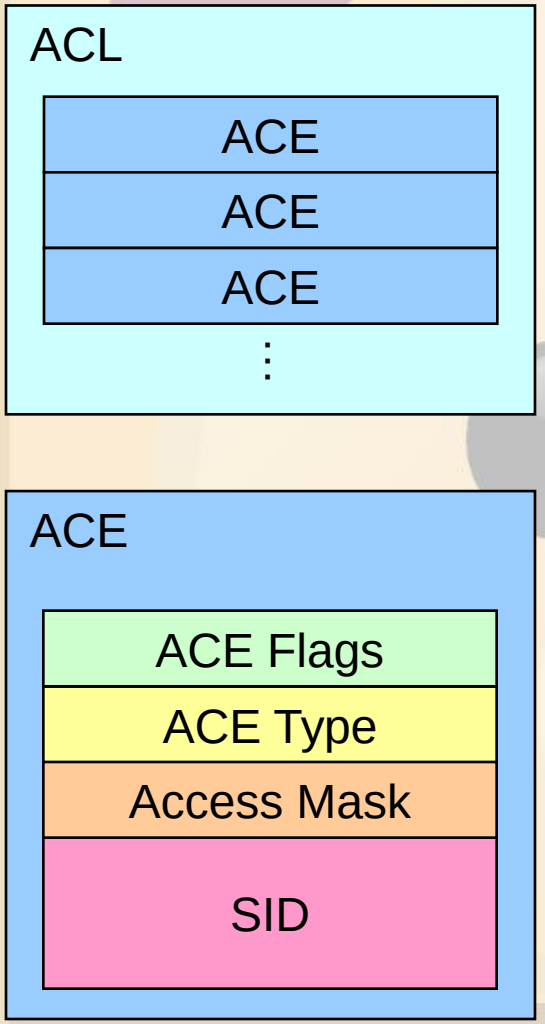
Security descriptor (セキュリティ記述子)

Object の security information が格納されている security descriptor には下記情報が格納されている。



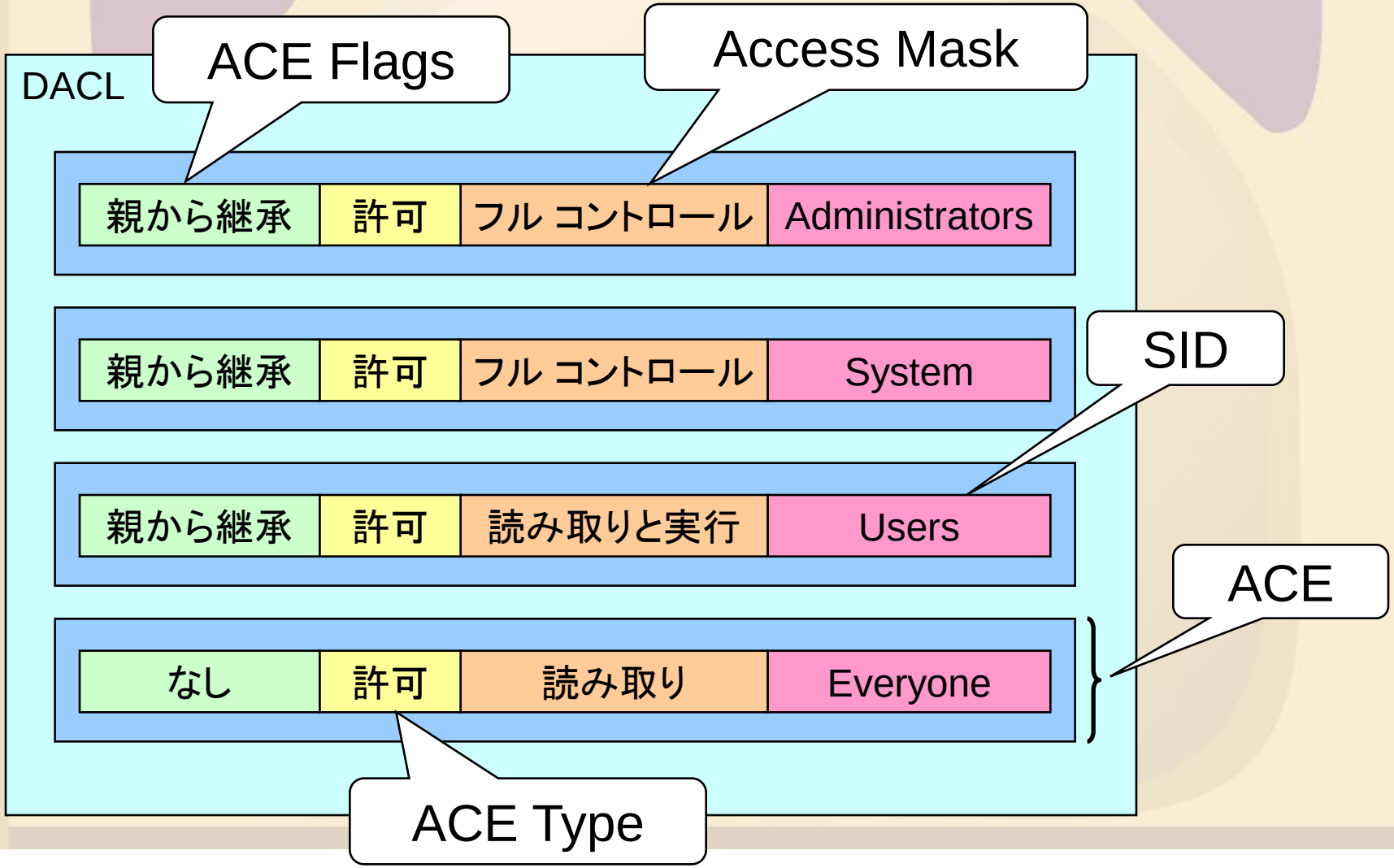
- Owner (所有者)
  Object の owner SID
- Primary Group
  Object の primary group SID
  ※ POSIX sub system でのみ使用
- SACL (System Access Control List)
  Object への監査等を制御する list
- DACL (Discretionary Access Control List)
  Object を扱えるか制御する list

ACL と ACE の内部構造



- ACL (Access Control List)
  - SID 毎の entry である ACE の集合
- ACE (Access Control Entry)
  - 対象への操作を制御する SID 毎の entry
- ACE Flags
  - Object の種類、継承および監査を制御
- ACE Type
  - 許可、拒否および監査のいずれか
- Access Mask
  - Control の種類 (読み取り、書き込み等)

DACL の例


DACL
ACE Flags
Access Mask
親から継承
許可
フル コントロール
Administrators
親から継承
許可
フル コントロール
System
SID
親から継承
許可
読み取りと実行
Users
ACE
なし
許可
読み取り
Everyone
ACE Type

Access Mask

- Access mask とは object を操作する権限のこと
  例) 読み込み、書き込み

- Access mask の種類は下記に大別できる
  - Generic Access Rights (汎用アクセス権)
    - Object の種類によって standard および object specific に mapping される
  - Standard Access Rights (標準アクセス権)
    - すべての objects に対して共通する権限
  - Object Specific Access Rights (特殊アクセス権)
    - Object 特有の権限 (その object の種類でのみ有効)

Object Specific Access Rights (files)

権利	日本語表記	対象
FILE_ADD_FILE	ファイルの作成/データの書き込み	directory
FILE_ADD_SUBDIRECTORY	フォルダの作成/データの追加	directory
FILE_APPEND_DATA	フォルダの作成/データの追加	file
FILE_CREATE_PIPE_INSTANCE	フォルダの作成/データの追加	named pipe
FILE_DELETE_CHILD	サブフォルダとファイルの削除	directory
FILE_EXECUTE	フォルダのスキャン/ファイルの実行	file
FILE_LIST_DIRECTORY	フォルダの一覧/データの読み取り	directory
FILE_READ_ATTRIBUTES	属性の読み取り	all
FILE_READ_DATA	フォルダの一覧/データの読み取り	file & pipe
FILE_READ_EA	拡張属性の読み取り	file & directory
FILE_TRAVERSE	フォルダのスキャン/ファイルの実行	directory
FILE_WRITE_ATTRIBUTES	属性の書き込み	all
FILE_WRITE_DATA	ファイルの作成/データの書き込み	file & pipe
FILE_WRITE_EA	拡張属性の書き込み	file & directory
STANDARD_RIGHTS_READ	読み取り	all
STANDARD_RIGHTS_WRITE	書き込み	all
SYNCHRONIZE	(オブジェクトの同期)	all

Standard Access Rights

権利	日本語表記 (file)	説明
DELETE	削除	削除
READ_CONTROL	アクセス許可の読み取り	Security descriptor の読み込み
SYNCHRONIZE	-	同期
WRITE_DAC	アクセス許可の変更	アクセス許可の変更
WRITE_OWNER	所有権の取得	所有権の取得
STANDARD_RIGHTS_ALL	-	他の権利の論理和 (SYNCHRONIZE 含む)
STANDARD_RIGHTS_EXECUTE	アクセス許可の読み取り	= READ_CONTROL
STANDARD_RIGHTS_READ	アクセス許可の読み取り	= READ_CONTROL
STANDARD_RIGHTS_REQUIRED	-	他の権利の論理和 (SYNCHRONIZE 含まない)
STANDARD_RIGHTS_WRITE	アクセス許可の読み取り	= READ_CONTROL

Generic Access Rights

権利	日本語表記 (file)	Mapping (file)
GENERIC_ALL	フル コントロール	全部
GENERIC_EXECUTE	実行	STANDARD_RIGHTS_EXECUTE FILE_READ_ATTRIBUTES FILE_EXECUTE SYNCHRONIZE
GENERIC_READ	読み取り	STANDARD_RIGHTS_READ FILE_READ_DATA FILE_READ_ATTRIBUTES FILE_READ_EA SYNCHRONIZE
GENERIC_WRITE	書き込み	STANDARD_RIGHTS_WRITE FILE_WRITE_DATA FILE_WRITE_ATTRIBUTES FILE_WRITE_EA FILE_APPEND_DATA SYNCHRONIZE

object の種類に応じた権限に展開される

Owner (所有者)

- Object の所有者 (通常の場合作成者) の SID が格納される
- Owner は DACL で ACL の編集が認められていない場合でも ACL の編集が可能

SeTakeOwnershipPrivilege
(ファイルとその他のオブジェクトの所有権の取得)

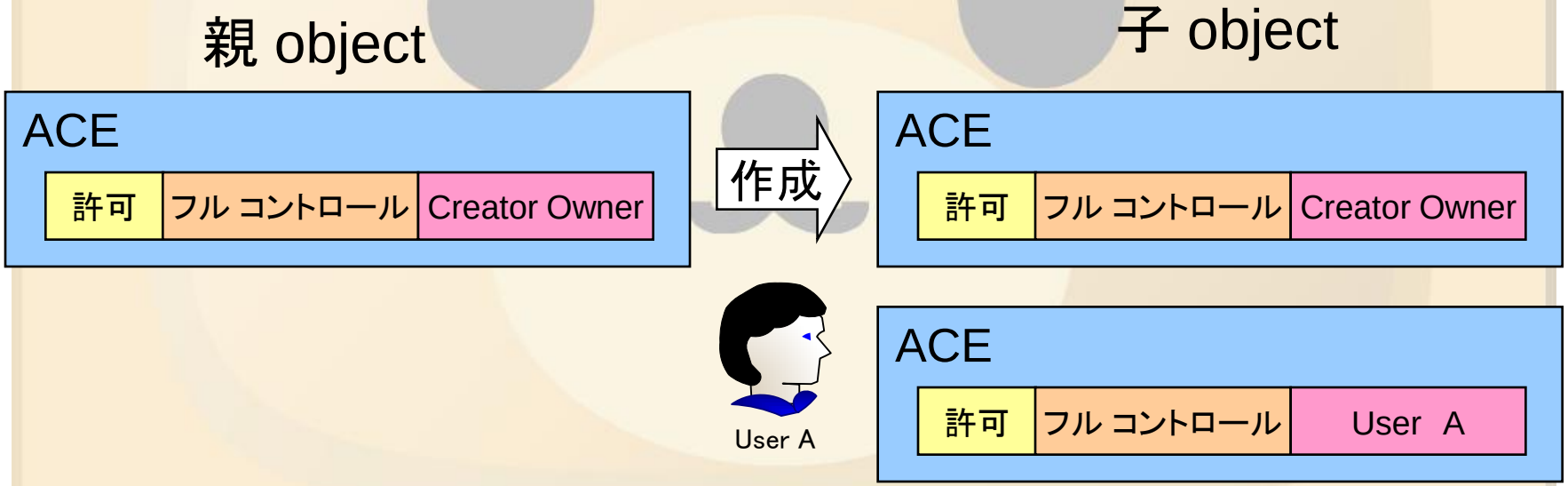
SeTakeOwnershipPrivilege を有する user は ACL で許可されていない場合でも所有権を任意の account へ置き換え可能。

この特権を所持している user であれば、所有権を置き換えた上で DACL を編集することにより、事実上すべての object を扱うことができる。

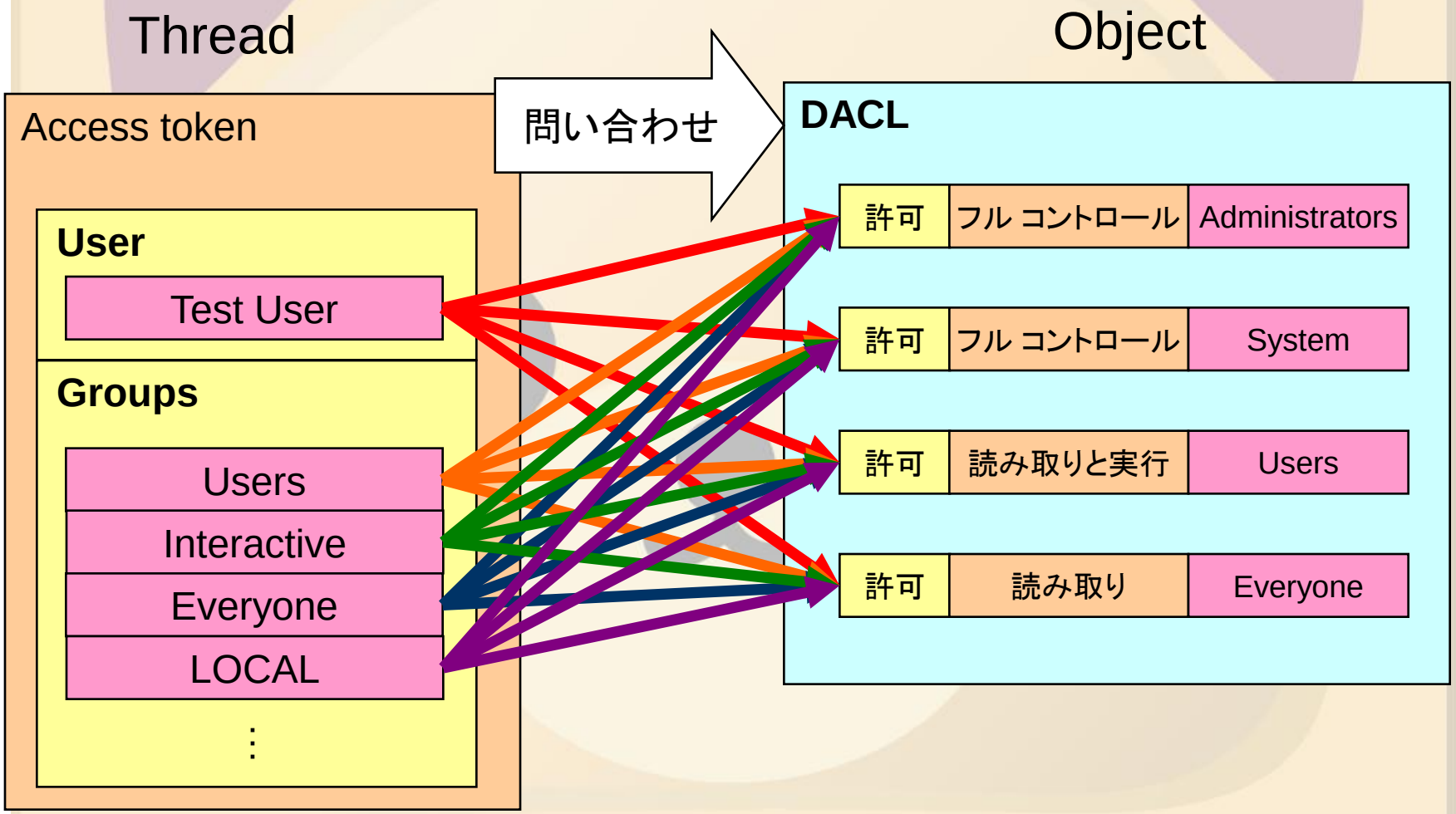
SeTakeOwnershipPrivilege は default で Administrators group がその特権を所持している。

Creator Owner

- Creator Owner を ACE に設定すると新規作成した object の DACL は Creator Owner の ACE に加え、作成者の ACE が追加される



Windows access control のしくみ (詳細)



Windows access control の弱点

- 格納されている data が暗号化されていない
  - 暗号化されていないため、OS の制御が及ばない場合には制限がかからない

  具体的には
  - Disk に保存されている data を直に扱えば data の読み書きがいくらでもできる
  - SeTakeOwnershipPrivilege を所持している account を利用できる端末につなげば、data の読み書きが可能

つまり、

物理的な secuirty が確保されない場所に保存されている data は保護されない。

そういう場合には暗号化しましょう。

まとめ

- Access control が無いと
  - 情報漏えいし放題
  - Data 改ざんし放題
  - 不注意で system 破壊されちゃう

  → **ということでちゃんとやりましょう！**

- Object を扱えるか否かの判断は process および thread に格納されている access token の user および groups と対象 object の DACL との照合によって行われる

- Access が拒否されたのを調査するにはその手の tool を使うと効果的

Tools

- Windows XP SP2 Support Tools
  - whoami.exe (2003 から標準搭載)
  - xcacls.exe (Vista 以降ではより高機能の icacls.exe が利用可能)
- Windows Server 2003 Resource Kit Tools
  - subinacl.exe (高機能)
- SysInternals
  - Process Explorer
  - Process Monitor
  - PsExec
  - WinObj

SysInternals suite でまとめて導入をお勧め

全て Microsoft の Web site から download できます。

参考文献

- MSDN (英語)
  - Win32 and COM Development
    - Security
      - **Authorization**

  http://msdn2.microsoft.com/en-us/library/aa375769.aspx

  基本ですね。

- “WinNT.h” (もち英語)

  内部構造とかはこれを読むのが一番。

- インサイド Microsoft Windows 第4版 下 (なんと日本語)

  David Solomon, Mark Russinovich 著

  いわずと知れた定番です。

参考文献

- Technet
  - Windows Server 2003 Technical Reference
    - Windows Security Collection
      - **Authorization and Access Control Technologies**

http://technet2.microsoft.com/windowsserver/en/library/addc004e-a1ad-4fba-8caa-1c9c3eb0fa861033.mspx

概念を理解するにはこれが一番かな？