UAC
(User Account Control)

ちゃっぴ

はじめに

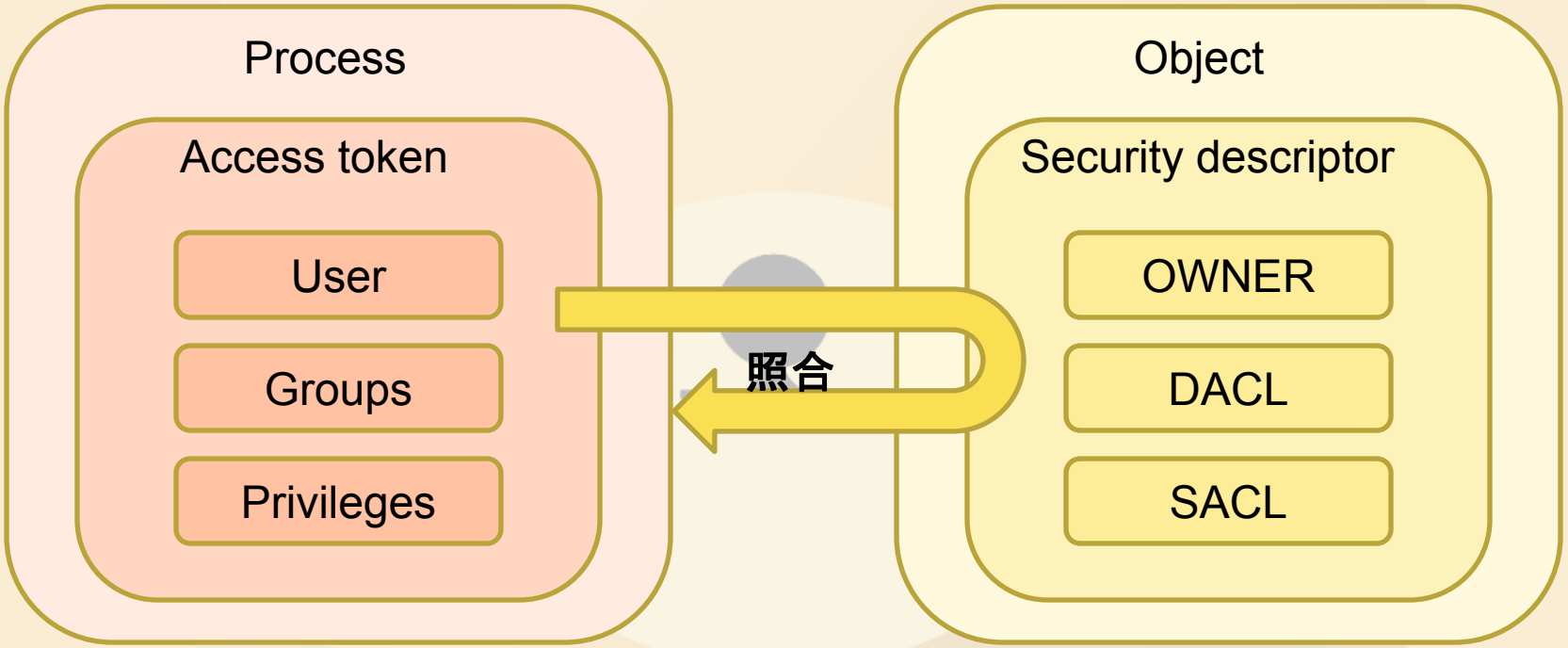
Windows Vista から UAC (User Account Control) が導入されましたが、これはなぜ導入されたのでしょうか？その背景を説明するとともに UAC の内部仕様を解説します。

なお、本資料のところどころに参考資料の link を埋め込んでいます。興味がある方は参考資料を参照し、より深い知識を身につけてください。

Windows Access Control の基本

Access Control のしくみ

Windows での access control は process または thread の access token と扱う object の security descriptor の照合によって行われる。



- Authorization and Access Control Technologies – Technet
- How DACLs Control Access to an Object - MSDN

Windows Access Control の基本
SID

Windows で利用される account (user, group 等) は内部では SID (Security IDentifier) と呼ばれる ID によって管理されている。

SID	Name
S-1-1-0	Everyone
S-1-5-32-544	BUILT IN¥Administrators
S-1-5-32-545	BUILT IN¥Users
S-1-5-21-185539546-1431137498-1249753232-1000	TyappiPC¥Tyappi

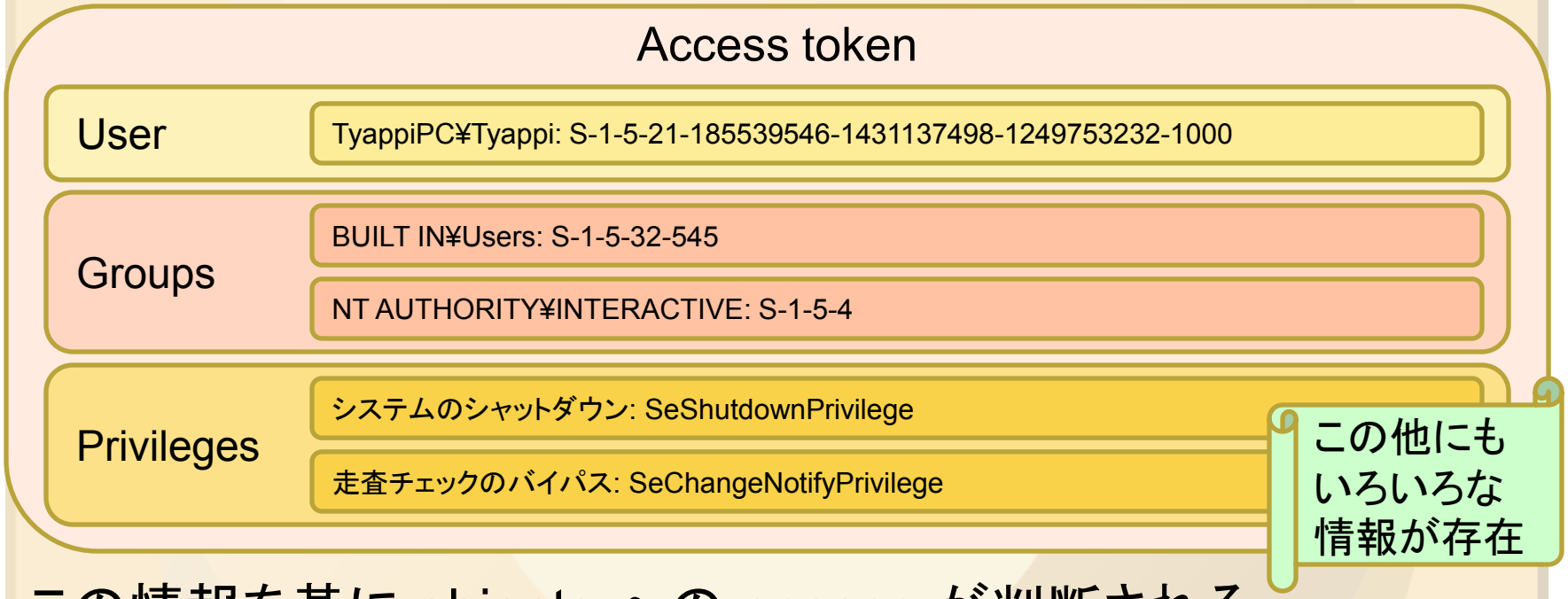
RID: Relative IDentifier

Computer または domain 毎に固有の ID が割り当てられる。

- Security Identifier Architecture - Technet
- Security Identifiers - MSDN

Windows Access Control の基本
Access Token

Access token とは log on した user の資格情報 (user や所属する group の SID や privileges (特権)) が格納されたもの。



この情報を基に objects への access が判断される。

- Access Tokens Technical Reference – Technet
- Access Tokens - MSDN

Windows Access Control
Access Token 確認方法

Access token に含まれる一部の情報は下記 tools を用い確認することができる。

- Whoami.exe

  cmd.Exe の process に含まれる access token 情報の一部を確認できる。
  Windows Vista 以降では標準搭載。Vista 以前は Support Tools に含まれる。
  Whoami - Technet

- Process Explorer

  現在起動している process の access token 情報の一部を確認できる。
  Process Explorer

Windows Access Control の基本
Privilege

Windows で利用される account (user, group 等) は内部では SID (Security IDentifier) と呼ばれる ID によって管理されている。

SID	Name
S-1-1-0	Everyone
S-1-5-32-544	BUILT IN¥Administrators
S-1-5-32-545	BUILT IN¥Users
S-1-5-21-185539546-1431137498-1249753232-1000	TyappiPC¥Tyappi

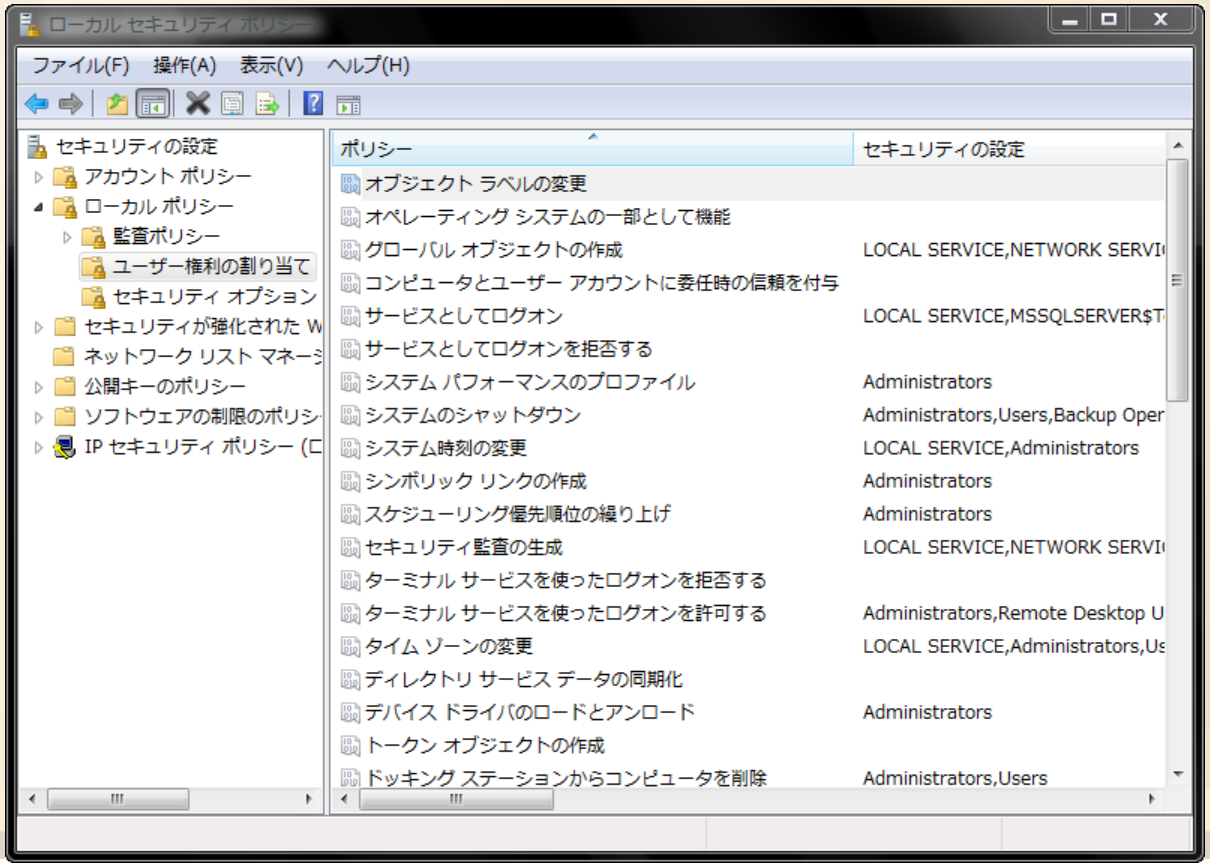
RID: Relative IDentifier

Computer または domain 毎に固有の ID が割り当てられる。

- Security Identifier Architecture - Technet
- Security Identifiers - MSDN

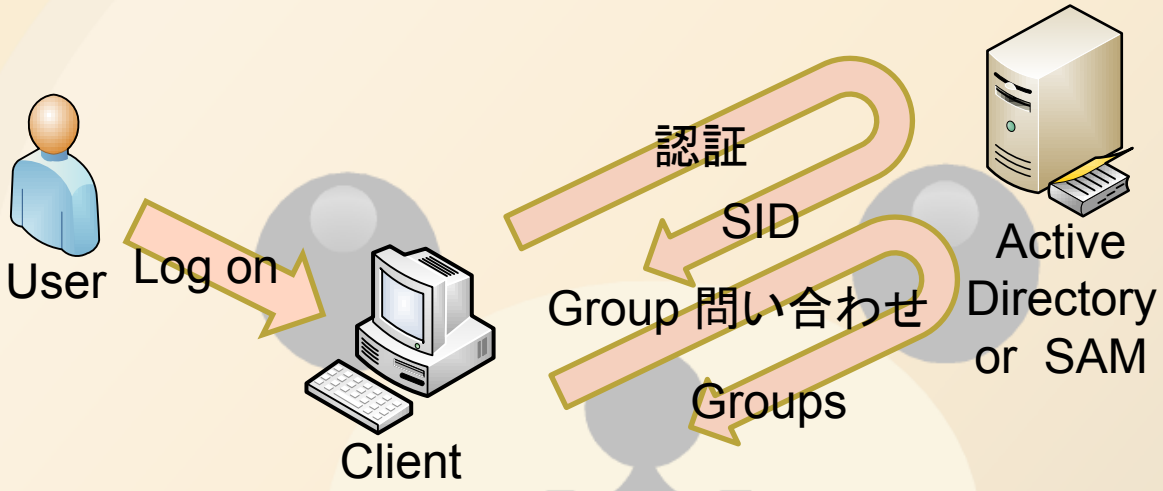
Windows Access Control の基本
Privilege 設定方法

User や group に特権を付与するには「セキュリティー ポリシー エディター」を利用する。



Windows Access Control の基本
Access Token 生成のしくみ

User が log on する過程において access token は自動的に生成される。



Access token が生成されるのは log on 時のみであるため、所属する group を変更した場合には、log on をやり直さないと結果が反映されない。

- How Access Tokens Work – Technet

Windows Access Control の基本
Restricted Token

Access token の内容は基本的に変更できないが、CreateRestrictedToken 関数を利用することにより制限を課すことはできる。より制限を課した token を restricted token と呼ぶ。Restricted token は token の内容に対し、下記変更を行う。

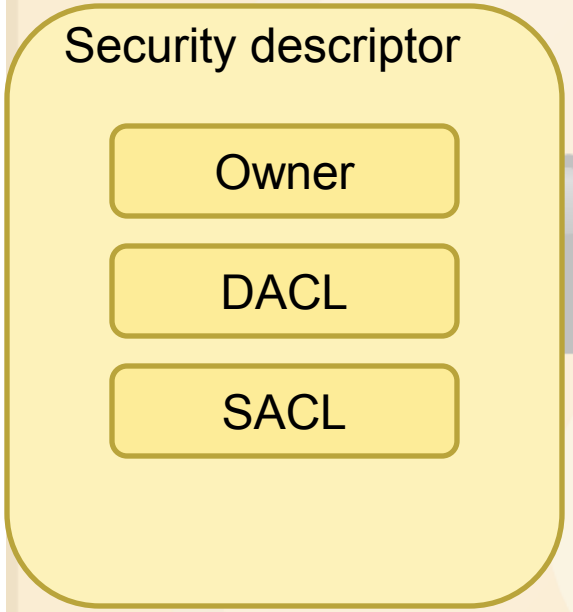
- Privileges
  制限を掛ける特権を消し去る
- Groups
  制限を掛ける SID に deny-only が設定される

Token に制限を課した場合、制限を課した token を制限取り払った状態に戻すことはできない。

- How Access Tokens Work - Technet
- Restricted Tokens - MSDN

Windows Access Control の基本
Security Descriptor

Objects に対する access 制御情報をまとめたもの。



Security descriptor の中には下記が存在する。

- Owner
  対象 object の所有者
- DACL (Discretionary Access Control List)
  対象 object への access を制御する list
- SACL (System Access Control List)
  対象 object への監査を制御する list

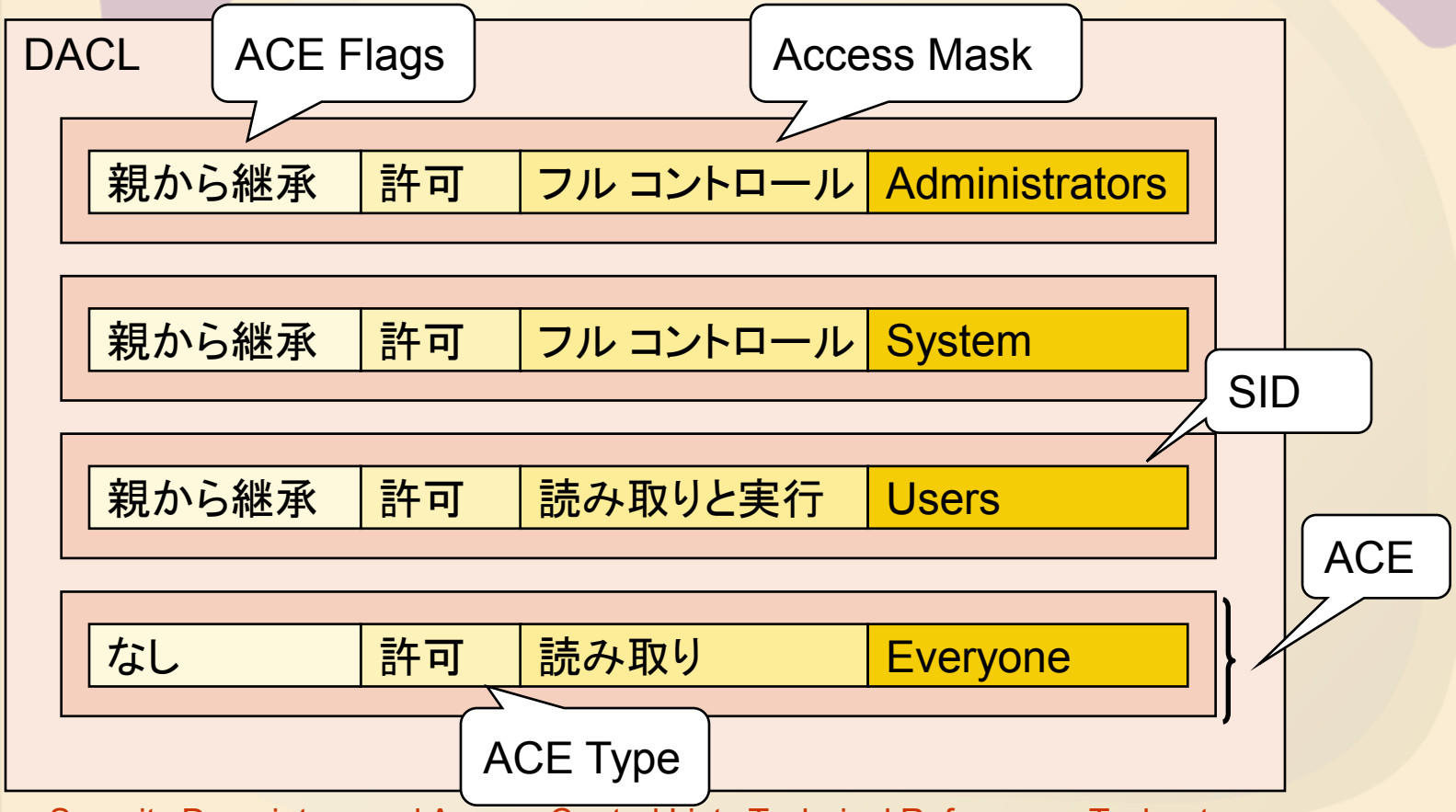
Security descriptor は SDDL (Security Descriptor Define Language) を利用し、文字列として記述することも可能。

"O:AOG:DAD:(A;;RPWPCCDCLCSWRCWDWOGA;;;S-1-0-0)"

- Security Descriptors and Access Control Lists Technical Reference - Technet
- Security Descriptors - MSDN
- Security Descriptor Definition Language – MSDN

Windows Access Control の基本
DACL

DACL の構成の例を下記に示す。



- Security Descriptors and Access Control Lists Technical Reference - Technet
- Security Descriptors - MSDN

Windows Access Control の基本
Security Descriptor 確認方法

Security descriptor の確認・設定方法は object の種類によって異なる。File の security descriptor は下記 tools を用い確認・設定することができる。

- ACL Editor
  Explorer.exe の「プロパティ」 - [セキュリティ] tab で表示される GUI editor
- Cacls
  Command line にて DACL の取得・設定を行える。
  Cacls
- Icacls
  Command line utility。Windows Vista 以降に標準搭載。
  Icacls
- Icacls
  Command line utility。さまざまな object の ACL を確認・編集できる。Windows Server 2003 Resource Kit Tools 同梱
  SubInACL

最小限の権限の原則

権限を分離する理由としては、異なる人の間で情報の共有を制限するというのが最有力だが、その他にも権限を分離した方がよい場合が存在する。

上図は一般的な攻撃であるが、利用している user の権限によって受ける影響は大きく異なる。このような攻撃の影響を緩和するためにも常に最小限の権限で処理を実施することが重要。

LUA

LUA (Least-privileged User Account)

Windows Vista で UAC 導入されるまでは、作業に必要最小限の user account を使い分けることが推奨されていた。この考え方を LUA と呼ぶ。

LUA を確実に実施することができれば、非常に安全な状態を作り出すことが可能。

Account 切り替え

LUA では複数の accounts を使い分ける必要があるため、下記方法を用いて accounts を切り替える必要がある。

- Log off – log on
  - 一端 log off し、別の account で log on する。
  - 非常に時間がかかる。作業が中断される。
- ユーザーの簡易切り替え (FUS: Fast User Switching)
  - ユーザーの切り替えを利用。
  - 比較的高速に切り替え可能。Windows XP では Domain 環境では利用できない。
- 別のユーザーで実行
  - Application を実行する際に context menu から「別のユーザーで実行」を選択し、user name, password を入力して実行する。または runas.exe を利用する。
  - この方法であれば比較的高速に切り替え可能。

上記方法は user の切り替えにそれなりに手間がかかる。

LUA
問題点

LUA を確実に実施することができれば非常に安全な状態を作り出すことが可能であるが、下記原因により LUA ほとんど実施されなかった。

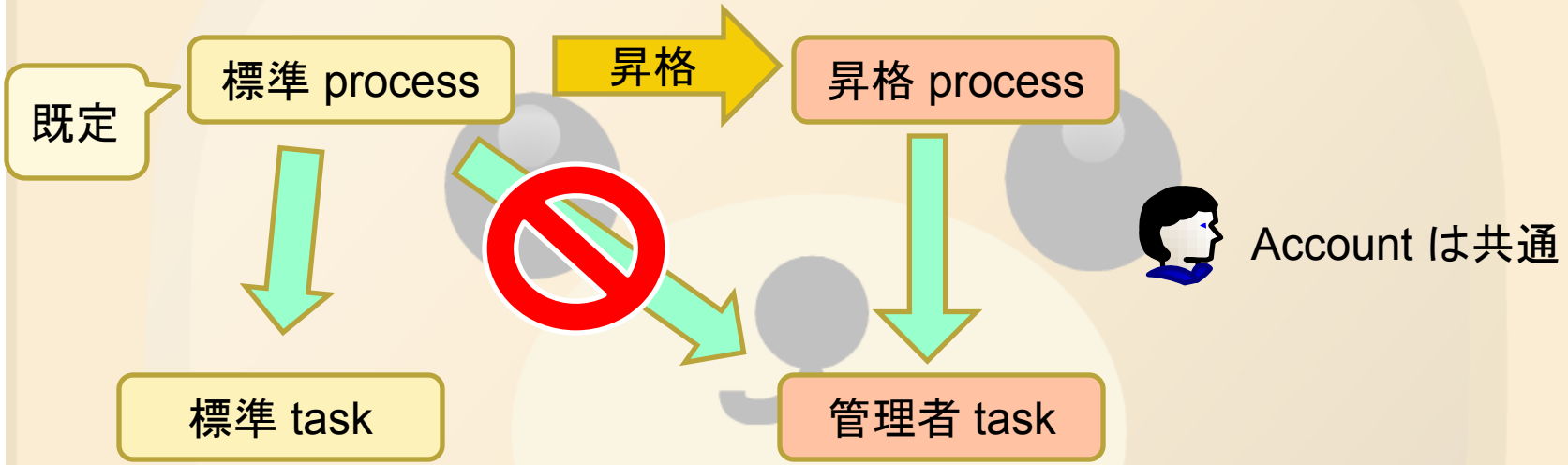
- Setup 時に作成される user が管理者権限を所持
  - 大多数の user は初期 setup 時に作成した user を何も考えず利用する。
- User 切り替えの煩雑性
  - 前頁で説明した方法はどれもそれなりに手間がかかる。
- User profile の問題
  - 異なる accounts を使い分けるため、user profile が一致しない。

と、いろいろ書きましたが、ぶっちゃけた話、興味も無く、気にしていないからそんなこと知りもしないというのが一番大きいのではないかと。。。

→ 詳しい user で無くても保護できるしくみが必要

UAC
概念

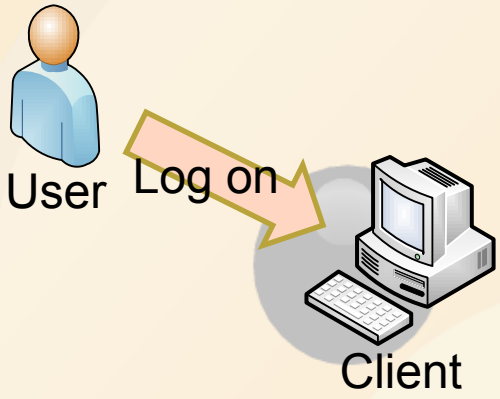
Windows Vista では UAC (User Account Control) が導入され、同じ account でも違った権限を持たせることができるようになった。



UAC が有効な状況では、標準 user と同等に制限された access token を持つ process が起動される。この process から管理者権限が必要な処理を実行することはできない。実行するためには「昇格」が必要になる。

UAC
Access Token 制限のしくみ

UAC が有効な状況では user の log on 時に「制限 token」と「昇格 token」の二種類の token が同時に生成される。



制限 token

標準 user と同程度の権限のみ所持

昇格 token

最大限の権限を所持

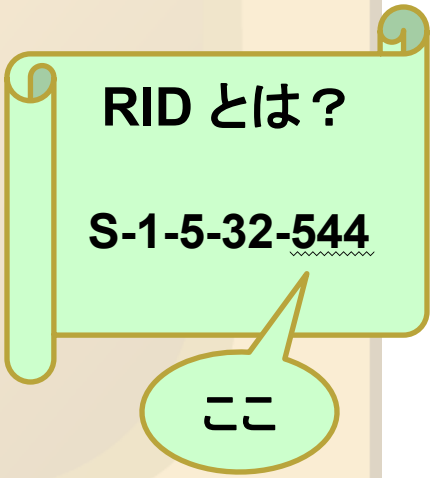
対話的に log on したときの Explorer は制限 token を使い起動されるため、user が普通に起動した process は制限 token で起動される。

制限 token はkernel 内部で CreateRestrictedToken を呼びだすことにより行われる。

UAC
Token 制限 - Groups

UAC が有効な状況では token から下記 groups が制限される。

Group	RID Name	RID Value
Domain Admins	DOMAIN_GROUP_RID_ADMINS	512
Domain Controllers	DOMAIN_GROUP_RID_CONTROLLERS	516
Cert Publishers	DOMAIN_GROUP_RID_CERT_ADMINS	517
Schema Admins	DOMAIN_GROUP_RID_SCHEMA_ADMINS	518
Enterprise Admins	DOMAIN_GROUP_RID_ENTERPRISE_ADMINS	519
Group Policy Creator Owners	DOMAIN_GROUP_RID_POLICY_ADMINS	520
Administrators	DOMAIN_ALIAS_RID_ADMINS	544
Power Users	DOMAIN_ALIAS_RID_POWER_USERS	547
Account Operators	DOMAIN_ALIAS_RID_ACCOUNT_OPS	548
Server Operators	DOMAIN_ALIAS_RID_SYSTEM_OPS	549
Print Operators	DOMAIN_ALIAS_RID_PRINT_OPS	550
Backup Operators	DOMAIN_ALIAS_RID_BACKUP_OPS	551
RAS and IAS Servers	DOMAIN_ALIAS_RID_RAS_SERVERS	553
Pre-Windows 2000 Compatible Access	DOMAIN_ALIAS_RID_PREW2KCOMPACCESS	554
Network Configuration Operators	DOMAIN_ALIAS_RID_NETWORK_CONFIGURATION_OPS	555
Cryptographic Operators	DOMAIN_ALIAS_RID_CRYPTO_OPERATORS	569



- New UAC Technologies for Windows Vista - MSDN

UAC
Token 制限 – 特権

Groups が制限された場合には下記特権のみ利用可能。

Friendly name	Constant name
走査チェックのバイパス	SeChangeNotifyPrivilege
システムのシャットダウン	SeShutdownPrivilege
ドッキング ステーションからコンピュータを削除	SeUndockPrivilege
プロセス ワーキング セットの増加	SeIncreaseWorkingSetPrivilege
タイム ゾーンの変更	SeTimeZonePrivilege

Groups が制限されない場合には下記特権が制限される。

Friendly name	Constant name
トークン オブジェクトの作成	SeCreateTokenPrivilege
オペレーティング システムの一部としての機能	SeTcbPrivilege
ファイルとその他のオブジェクトの所有権の取得	SeTakeOwnershipPrivilege
ファイルとディレクトリのバックアップ	SeBackupPrivilege
ファイルとディレクトリの復元	SeRestorePrivilege
プログラムのデバッグ	SeDebugPrivilege
認証後にクライアントを偽装	SeImpersonatePrivilege
オブジェクト ラベルの変更	SeRelabelPrivilege

- New UAC Technologies for Windows Vista - MSDN

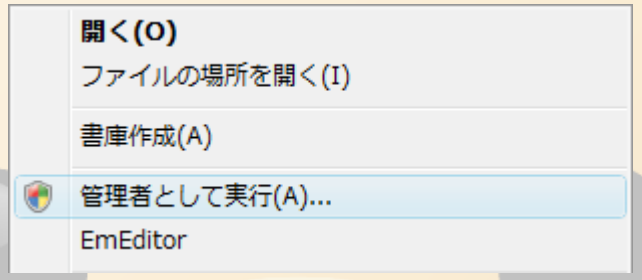
UAC
昇格方法

UAC 有効な環境で昇格を行う方法としては下記が存在する。

- 管理者として実行
- 互換性 tab
- Compatibility Fixes
- Application Manifest
- COM Elevation Moniker
- Installer の自動昇格

UAC
管理者として実行

Context menu にある「管理者として実行」を利用して昇格された token で実行させることができる。



互換性 tab の「管理者としてこのプログラムを実行する」を利用すると必ず昇格 dialog を表示させることも可能。

Microsoft Application Compatibility Toolkit 同梱の Compatibility Administrator を利用することで同じ設定を大量展開することも可能。

UAC
Application Manifest

Application manifest で requestedExecutionLevel を指定することにより、application 単位で昇格を制御することができる。

Value	Restrict	Virtualize	Description
(Unmarked)	Yes	Yes	呼び出し元と同じ
asInvoker	Yes	No	呼び出し元と同じ
highestAvailable	No	No	制限解除 (IL: High)
requireAdministrator	No	No	制限解除 (IL: High) Administrators group に所属している user のみ実行可能

Application を新規開発する場合、application manifest に requestedExecutionLevel を必ず指定する。

VS 2008 以降では既定で “asInvoker” が指定されている。Application 全体で昇格が必要な applications はできる限り “highestAvailable” を指定する。

UAC
COM Elevation Moniker

Application manifest による昇格は process 単位となるため、局所的に管理者権限が必要となる application では望ましく無い場合が多い。

Process 全体で無く局所的に昇格させる場合には COM Elevation Moniker を利用した昇格を行わせることができる。

- The COM Elevation Moniker – MSDN

UAC
Installer の自動昇格

過去との互換性のために Installer と予測されるものは自動的に昇格する機能がある。

対象の application manifest で requestedExecutionLevel が指定されておらず、名前に “install”, “update”, “setup” が含まれる場合、自動的に昇格 dialog が表示される。

本機能はあくまで過去との互換性用のみに用意されているため、積極利用してはならない。

IL
Integrity Level (整合性レベル)

Windows Vista 以降の kernel には IL (Integrity Level) という概念が導入された。

IL	Restrict	Description
Low	Yes	非常に限定的な process Internet Explorer の protected mode (保護モード) で利用
Medium	Yes	通常の process
High	No	昇格した process (UAC 無効化時既定)
System	No	NT Authority¥System, NT Authority¥Network Service, (NT Authority¥Local System で利用)

IL は昇格状態を表すとともに、旧来の access control とは別の access control にも利用される。

IL
SID

Integrity level を表す SID が存在し、access token の groups にその SID が格納される。この SID を調査することでどの IL で動作しているか調査が可能。

Name	Integrity level SID
Mandatory Label¥Low Mandatory Level	S-1-16-4096
Mandatory Label¥Medium Mandatory Level	S-1-16-8192
Mandatory Label¥High Mandatory Level	S-1-16-12288
Mandatory Label¥System Mandatory Level	S-1-16-16384

この SID は任意 objects の SACL に設定することができ、これを基に access control することができる。何も指定していないときは、Medium が設定されているものとして扱われる。

Windows Integrity Mechanism Design - MSDN

IL
Access Control

Vista 以降の kernel では下記順番で ACL の照合が行われる。

1. IL 照合 (Vista 以降から導入)

   Access token に格納された IL SID と対象 object の SACL に格納された IL SID を照合し、access token に格納された IL SID が低い場合には access 拒否する。

2. DACL 照合

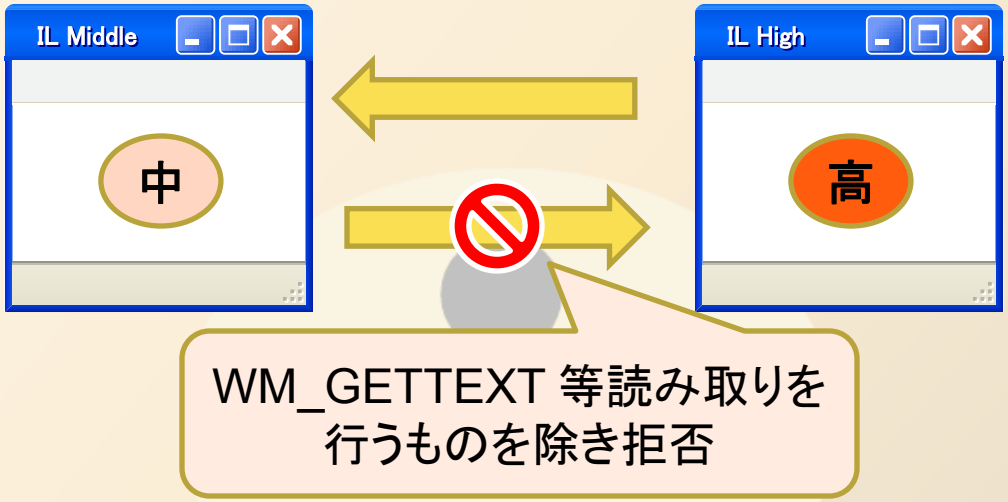
   Windows Vista 以前でおなじみの access control

IE で protected mode が有効な時には IL Low で起動しているため、SACL に IL Low が設定されている objects にしか access できない。

- Windows Integrity Mechanism Design – MSDN

UIPI

低い IL の処理からより高い IL に window message を送り、処理が実行できてしまうと IL を設ける意味が無くなってしまう。そのためより高い IL には明示的な許可無しには window message を受け取れないように仕様変更された。この仕組みを UIPI (User Interface Privilege Isolation) と呼ぶ。



より低い IL から高い IL への window message を許可するには message 受信側で ChangeWindowMessageFilter 関数を呼び出し、明示的に許可を与える必要がある。

- Windows Integrity Mechanism Design – MSDN

Secure Desktop

Windows Vista の log on 画面は「Secure Desktop」と呼ばれる通常 user が利用する desktop とは別の desktop で表示されている。この desktop は通常 user が利用する desktop から直接扱えることができないように厳密に保護されている。

Windows Vista の既定では昇格 dialog はこの「Secure Desktop」を使用する。昇格 dialog 直前に画面が暗転するのはこのため。

Policy を変更することにより、「Secure Desktop」を利用しないこともできるが、program 的に自動承認させることができる可能性が生じるため推奨しない。