UNIX ― Windows 統合認証

葛 生 和 人

Ⅰ．はじめに

プライベートな時間から日常の業務まで，さまざまなコンピュータにアクセスしようとするとき，私たちは，必ずユーザ ID やパスワードの入力が求められます。いわゆるログオン認証プロセスと呼ばれるものですが，これはコンピュータ上のデータやネットワーク上のリソースを不特定多数の第３者から保護するという意味で必要不可欠なものです。このような認証プロセスを実現するためには，個人を特定するための情報やパスワードがコンピュータデータベース上に格納され管理されていなければなりません。例えば，個人的に使用するデスクトップマシンやノートパソコンでは，最低でも管理者（Administrator）と使用者の ID とパスワードがそのマシン上に保存されています。一方，企業や研究機関，大学などの組織で管理運用されているコンピュータは，特定のサーバマシン上に，その組織に属する全員の ID やパスワード，あるいはその他のユーザプロファイルが保存されることになります。特に，大規模な組織になると，それぞれの部署，部局ごとで個人情報を管理し，また，そのための管理サーバも Windows, Mac, Linux, UNIX など異なる環境上に構築されるというような状況が生じます。そのような場合，情報の管理に費やされる時間やコストは必然的に増加し，さらには他の業務における生産性の低下をもたらすということさえ十分予想されます。そのようなことから，認証のための情報を一括して管理できるシステム，いわゆる統合認証という考え方が重要となってきます。ここでは，さまざまな組織で幅広く運用されている OS 環境として UNIX/Linux と Windows を取り上げ，それら認証システムの統合方法について，いくつか例をあげながら紹介していきます。

Ⅱ．個人情報管理と認証

プライベートや限られた仲間内での使用を目的とした PC や UNIX/Linux マシンでは，ユーザ登録はマシンごとに行い，登録された情報はマシンごとに管理されています。しかし，組織内でネットワークに接続した多くの端末から，共通のコンピュータリソースを利用できるようにするためには，サーバマシンを設定して，一括してユーザ情報を管理運用できるような認証システムを構築しなければなりません。そのような，情報管理の方法として最近多くのシステムで採用されているのが，ディレクトリサービスを使った情報管理です。これから取り上げる UNIX/Linux 系 OS と Windows 系 OS で使用している LDAP（Lightweight Directory Access Protocol）及び Active Directory はその代表的なものです。

１．ディレクトリサービスと統合認証

いま，仮にある組織においてUNIX/Linux系OS環境を利用するグループとWindows Server環境を利用するグループが存在し，それぞれのユーザの個人情報に関して，一方はLDAPにより，もう一方はActive Directoryにより管理運用されているものとします。このような状況では，ユーザが両方のコンピュータリソースを利用するためには，両方のディレクトリサーバにアカウントを登録しなければなりません。また，アカウントの削除，変更に関しても同様です。このような作業は，システム管理者にとってもユーザにとっても非常にわずらわしいものとなってしまいます。また，同じ個人情報が２箇所に存在するということは，情報への不正アクセス対策も２重に必要となるということでセキュリティ的にも問題が生じます。そこで，個人情報の管理をLDAP，Active Directoryのいずれか一方に任せてしまい，なおかつ，いずれの環境からでも同じ情報管理の元で認証が受けられるようにしたい，という発想が出てきます（図１）。これが，統合認証の考え方です。

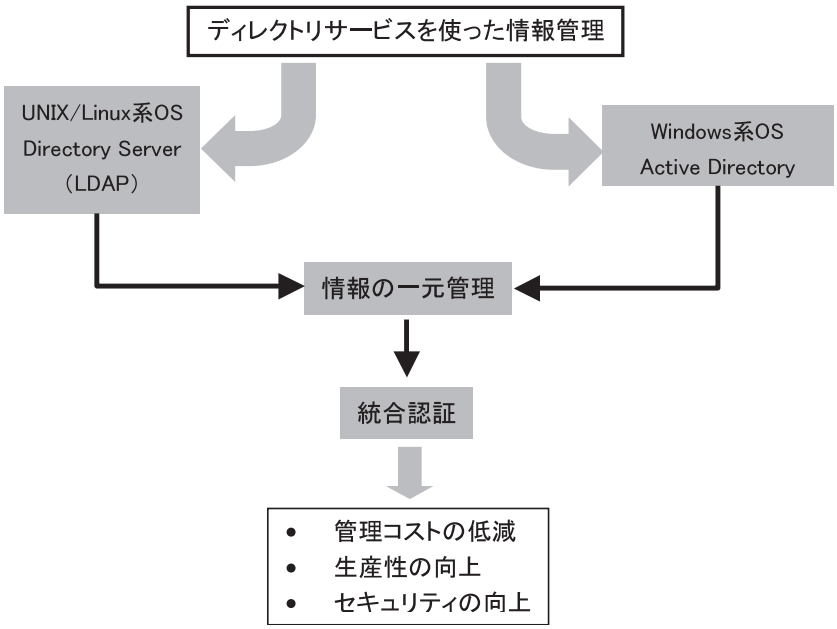

図１　ディレクトリサービスと統合認証

２．統合認証へのアプローチ

ここで，UNIX/Linux系システムとWindows Serverシステムにおける統合認証システムの構築を考えた場合，代表的な２つの方法があげられます。

第一の方法は，個人情報管理に関してはNISやLDAPなどのUNIX/Linux系システムで運用し，Windowsクライアントがシステムにログオンするときにはそれらの情報を利用して認証を受けるという方法，もう１つの方法は，その逆でWindowsサーバ側のActive Directoryで個人情報を管理運用し，UNIX/Linuxクライアントがシステムにログオンするとき

はActive Directory上の情報を利用して認証を受けるという方法です。いずれの方法に関しても，ログオン時に入力する情報はユーザアカウントとパスワードのみで，ユーザから見た使い勝手上の違いはまったくありません。しかし，UNIX/Linux系システムとWindows系システムでは，内部的に処理される情報の属性，フォーマット，認証プロトコルなどが異なるため，それらの情報を共通のデータベースに置いて管理する際には，認証時に整合性が取れるようにデータ処理がなされる形でなければなりません。そこで，そのための統合認証用モジュールが必要となってきます。Samba[1], GINA[2], winbind[1], SFU[3], AD4Unix[4] などは統合認証専用モジュールではありませんが，いずれも上で述べた要求を満たす機能を持っており，UNIX/Linux-Windows間での統合認証を実現することができます。後ほど，そのうちのいくつかを実際の統合認証システム構築の説明の中で紹介します。

また，上で述べたアプローチとは異なり，UNIX/Linux系とWindows系でのディレクトリサーバはそのまま残した状態で双方のディレクトリ情報を常時監視し同期をとるようなサーバを追加する，という方法も考えられます。この方法では，システムごとに情報データ処理の整合性をとるというわずらわしい作業を省かれますが，監視システムに関しては，独自に開発するか，専用の商用システムを導入することが必要となってきます。ここでは，最後にそのような監視機能を実現した商用サーバとしてSun Java™ System Identity Synchronization for Windows[5] を紹介します。

なお，上で述べた統合認証に関して，システム構築へのアプローチごとに分類すると図２のようになります。

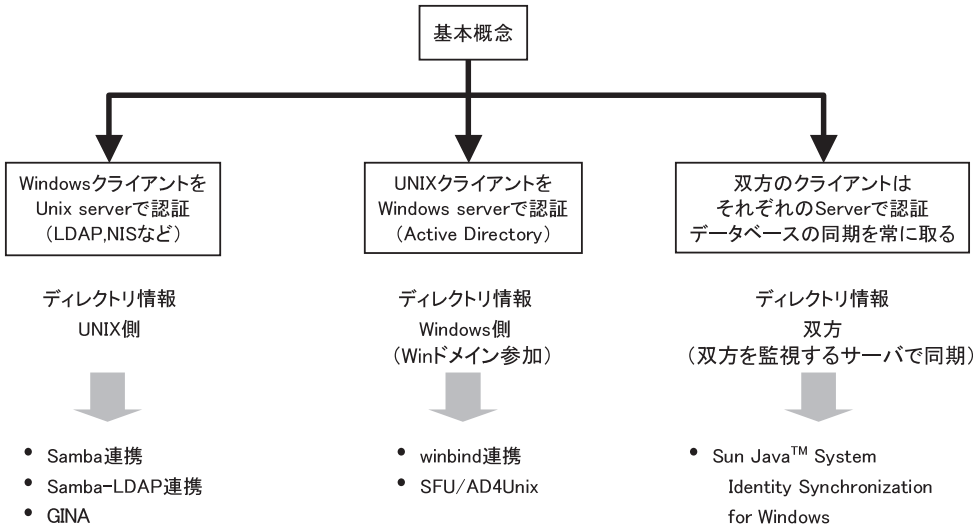

図２　統合認証へのアプローチ

1　Sambaに含まれるモジュール

Ⅲ．統合認証方法

前節で紹介したように，UNIX － Windows の統合認証を実現するための方法やその際利用するモジュールはいくつか存在します。ここでは，図２に示した統合認証のための３つのアプローチに対応して，Samba，GINA，SFU，Sun Java™ System Identity Synchronization for Windows を利用した場合のシステムの構築方法とその具体的手順について簡単に紹介します。

１．Samba-LDAP 連携

１．１　システム構成

最初に紹介するのは，Windows クライアントを LDAP で認証する統合認証システムです。

このシステムでは，UNIX/Linux システムの LDAP 上に Windows ドメインの情報を格納し，それらの情報を利用して Windows クライアントのドメイン認証を行えるようにします。なお，ここでは UNIX/Linux クライアントのログオン認証用として，LDAP 認証システムがすでに構築されているものとします。

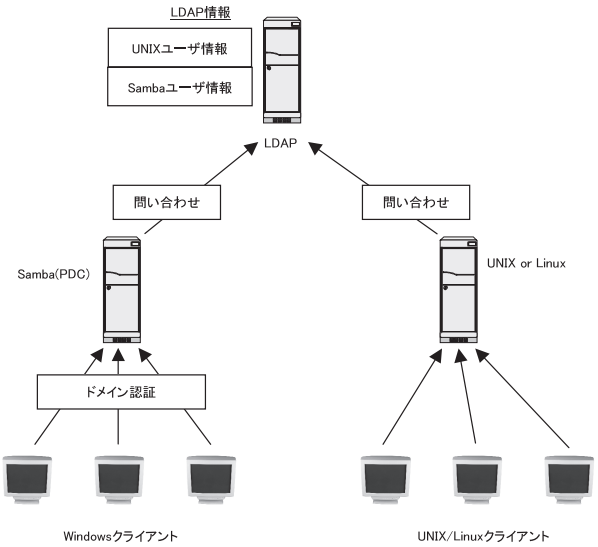

図３　Samba-LDAP 連携のためのシステム構成

上記のシステムを実現するためにまず必要なモジュールは Samba です。Samba は UNIX/Linux 上に Windows 互換のサービスを提供することのできるモジュールで，具体的サービスにはファイルサーバ機能，プリントサーバ機能，ドメイン管理機能などがあります。ここでは，Windows クライアントからのドメイン認証システムの構築が目的なので上記機能のうちドメイン管理機能を利用することになります。ただし，Samba 自体は LDAP サーバとしての機能は持っていないため，Samba と LDAP サーバを連携させる手続きを別途踏まなければなりません。以上述べたシステム全体の構成をまとめると図３のようになります。なお，UNIX/Linux 側での

LDAP 認証システムは，Windows 側のシステム構築からの影響をなんら受けることはありません。

1．2　ドメイン管理

Samba を利用した統合認証での重要なポイントは，Samba のドメイン管理機能を利用して UNIX 上に Windows ドメインに代わる Samba ドメインを構築することです。Windows ドメインとは，ネットワーク上に接続された Windows クライアントの集合（コンピュータ名，ユーザ名，グループ名などの属性で構成される）の組織単位を指し，Windows Server はその管理サーバとしての機能を提供しています。なお，Windows ドメインには，Windows NT 4.0 Server 以前に提供されていた NT ドメインと，Windows 2000 Server 以降に提供されている Active Directory ドメインの２つの異なる形態が存在します。これらは，表１に示したように管理体系を構成するためのポリシーとその使用モジュールに大きな違いがあります。Samba の提供するドメイン管理機能は現在のところ NT ドメインとの互換性を念頭においており，最新の Active Directory ドメインとの完全な互換性はありません。したがって，Samba ドメインに移行できる Active Directory ドメイン上の情報は，ユーザ情報，グループ情報など一部の情報に限られることになり，グループポリシー，Kerberos 認証に関わる情報など Active Directory ドメイン固有の情報は移行できないことに注意しなければなりません。

表１　NT ドメインと Active Directory ドメインの違い

	情報管理	名前解決	認証プロトコル
NT ドメイン	SAM（Security Account Manager）	WINS（Windows Internet Name Service）	NTLM（NT Lan Manager）認証
Active Directory ドメイン	Active Directory（LDAP 準拠）	DNS	Kerberos 認証

1．3　システム構築

それでは，Samba-LDAP 連携によるシステム構築の具体的手順について説明します。

システム環境としては，Linux（CentOS-4[2]）のクライアントに対してすでに LDAP 認証の環境が構築されており，Samba もインストール済みであるものとします。一方，Windows システム側では Windows サーバ（Windows Server 2003 R2）により Windows ドメインが構築されているものとします。なお，Windows ドメインに登録されたユーザ情報の移行については最後に紹介します。

2　ここでは UNIX/Linux 系システムとして CentOS-4 を使用していますが基本的な構築手順に差はありません。なお，以降の具体的手順の説明においてはこれまでの UNIX/Linux の代わりに Linux という表記を使用します。

Samba ドメインの構築

Samba-LDAP 連携システム構築のための最初の手順は，Samba ドメインの構築です。

構築のための流れは以下のようになります。

１）プライマリドメインコントローラ（PDC）として Samba を起動（PDC 用 smb.conf の設定）

２）クライアントマシンアカウントを Linux, Samba に登録

３）Windows クライアントの Samba ドメインへの参加

まず，Samba にプライマリドメインコントローラ（PDC）の機能を持たせるように smb.conf を設定し起動します。リスト 1 は smb.conf ファイルの一部で，太字の部分は PDC 用に設定されたパラメータです。リスト中の workgroup, netbios name には構築するドメイン名とドメインコントローラとしてのコンピュータ名を設定しています。また，os level は通常のクライアントより高い優先順位をもつように 64 と指定します。

smb.conf の設定が済んだら，

```
#  /etc/init.d/smb start
```

により Samba を起動します。

これで，Samba サーバは PDC として Samba ドメインを運用するようになります。

Samba ドメインの運用が開始したら，つぎに Windows クライアントがドメインに参加できるように，ドメインに参加するクライアントマシンを PDC に登録しなければなりません。クライアントマシンの登録は，まず，Linux ユーザとしてのマシンアカウントを Linux 上に登録した後，Samba 上にも同じマシンアカウントを登録します。

いま，仮にマシンアカウント登録用に UNIX グループとしてグループ名 **PCgroup** を作成し，そこに Windows マシン名 **testpc** を登録するものとすると，登録手順は以下のようになります。

```
#  /usr/sbin/groupadd PCgroup
#  /usr/sbin/usradd -g PCgroup -s /bin/false -d /dev/null testpc$
#  pdbedit -a -m testpc
```

なお，上記コマンドライン 2 行目は UNIX 用の登録[3]，3 行目は Samba 用の登録です。

最後に Windows クライアントがドメインに参加するためには管理者権限が必要となるため，そのための管理者権限の登録も行います。通常は root ユーザの権限を用いてドメインに参加す

3　UNIX 登録用のクライアントマシン名はすべて小文字で最後に $ を付けます。また，-s /bin/false と -d /dev/null の指定は，ログインシェルとホームディレクトリの指定ですが，ここでは UNIX マシンへのログインは許可しない設定としています。

るため，Samba ユーザアカウントとして root を登録します。

```
# pdbedit -a root
```

以上の操作により，Windows クライアントが Samba ドメインに参加するための環境が設定されました。Windows クライアントの Samba ドメインへの参加は，通常の Windows ドメインへの参加と同様に Windows メニューから行うことができます[4]。

```
[global]
     dos charset = CP932
     unix charset = EUCJP-MS
     display charset = EUCJP-MS
     workgroup = SAMBADOM
     netbios name = SAMBA30
     server string = Samba Server
     obey pam restrictions = Yes
     pam password change = Yes
     unix password sync = Yes
     log file = /var/log/samba/%m.log
     max log size = 50
     socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
     cups options = raw
     domain logons = Yes
     domain master = Yes
     security = user
     os level = 64
     preferred master = Yes
     local master = Yes
.....................................
.....................................
```

リスト１　PDC 用の smb.conf 設定

Samba-LDAP 連携設定

ここまでの設定で Linux 上に Windows ドメインに代わる Samba ドメインを構築することができました。ただし，この時点では Linux 上に異なる２つの認証システムが混在しているに過ぎません。これを統合認証という形にするために Samba-LDAP 連携の設定が必要となります。Samba-LDAP 連携設定の流れは以下のようになります。

１）OS-LDAP 間の連携設定（NSS,PAM の設定）
２）Samba-LDAP 連携用 LDAP 側の設定（slapd.conf の設定）
３）Samba-LDAP 連携用 Samba 側の設定（smb.conf の設定）
４）LDAP 管理者 pw の Samba への登録

最初の OS-LDAP 間の連携設定とは，LinuxOS 自体が LDAP データベースを参照し，ログオ

4　Windows ドメイン参加手順は通常の Windows 操作と同様なのでここでは省略します。

ン認証に LDAP の情報を使用できるようにするための設定です。ここでは，すでに LDAP 認証の設定が済んでいる環境を想定しているため詳細は省きますが，多くの Linux では標準装備された LDAP 認証設定用モジュールを利用することができます。例えば，RedHat 系 LinuxOS では authconfig モジュールを起動[5]することにより NSS（ユーザ情報の取得先を切り替える機能）と PAM（認証システムを切り替える機能）の設定を比較的簡単に行うことができます。図 4 は authconfig の起動画面です。

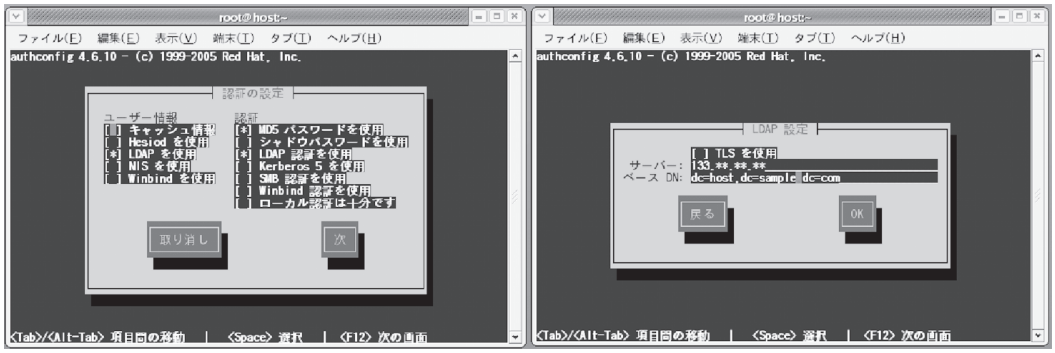

図 4　authconfig の設定画面

つぎに，Samba-LDAP 連携用の LDAP 側の設定を行います。

まず，ldap サーバに対して Samba のもつデータ属性や構造を扱えるようにするために，ldap 設定ファイルである slapd.conf の中に samba.schema を追加します。samba.schema ファイルは Samba ソースディレクトリの examples/LDAP にありますのでそれを ldap 用スキーマのディレクトリにコピーし，slapd.conf のスキーマファイルの指定（リスト 2 ：太字部分）を追加します[6]。

```
…
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/samba.schema
…
```

リスト 2　slapd.conf 設定（samba.schema の追加）

さらに, slapd.conf の中に Samba 用 LDAP アクセス制限（リスト　3 :太字部分）を追加します。

5　root 権限でコマンドラインより authconfig を入力。
6　ここでは openldap を使用しています。

```
.....
access to attr=SambaLMPassword
     by dn="cn=Manager,dc=sample,dc=com" write
     by self read
     by anonymous auth
     by * none

access to attr=SambaNTPassword
     by dn="cn=Manager,dc=sample,dc=com" write
     by self read
     by anonymous auth
     by * none

access to attr=userPassword
     by dn="cn=Manager,dc=sample,dc=com" write
     by self read
     by anonymous auth
     by * none
.....
```

リスト３　slapd.conf 設定（Samba 用アクセス制限の追加）

```
[global]
     dos charset = CP932
     unix charset = EUCJP-MS
     display charset = EUCJP-MS
.....................................
#LDAP settings
     passdb backend = ldapsam:ldap://localhost
     ldap suffix    = dc=sample,dc=com
     ldap user suffix    = ou=People
     ldap group suffix   = ou=Group
     ldap machine suffix = ou=Computers
     ldap admin dn = cn=Manager,dc=sample,dc=com
     ldap passwd sync = yes
#system administrator
     Admin users = Administrator
.....................................
```

リスト４　ldap 用の smb.conf 設定

つぎに Samba に備わっている Samba　LDAP 機能を利用するために Samba 側の設定を行います。この場合は，smb.conf の中で ldap 用の設定（リスト４：太字部分）を追加します。この設定により Samba 側にバックエンドデータベースとして LDAP を利用することを指定し，Samba 側の情報が登録されている LDAP ツリー上の RDN などを指定しています。なお，パラメー

タの詳細に関しては文献 [6]（pp.306-307）を参照してください。

最後に，Samba が LDAP に接続できるように LDAP 管理者のパスワードを secrets.tbd ファイルに格納します。

```
# smbpasswd -w [rootdn のパスワード]
```

Samba-LDAP 連携支援ツールの設定

以上の設定により Samba-LDAP 連携機能が利用できるようになりました。

しかし，実際に Samba-LDAP 連携をとおしてドメインを運用するためには LDAP に Samba 上の情報を投入する必要があります。LDAP では，通常 LDIF 形式でデータが投入されますが，smbldap-tools という Samba-LDAP 連携支援ツールを利用することにより LDAP と連携した Samba ドメインのユーザ情報管理の運用を行えるようになります。smbldap -tools は Samba パッケージのソースからインストールことができますが，ここでは簡単のためすでにインストールされているものとして，smbldap-tools 用の設定から説明します。設定ファイルは，smbldap_conf.pm というファイルで /usr/local/sbin/smbldap_conf.pm にインストールされています。

まず，LDAP ツリーのベース DN やユーザ等各種情報の格納場所である RDN を suffix,usersou, computersou, groupsou パラメータ（文献 [6] pp.310-311 参照）を通して指定します。なお，ここでは Windows ドメインがユーザやグループを管理するための番号 SID を指定する必要があります。これは，Samba サーバを起動した状態で，

```
# net getlocalsid
SID for domain SAMBA30 is : S-1-5-21-**********************************
```

により値を取得することができますのでファイルに書き加えてください。この指定により smbldap-populate により作成したグループと Windows で設定されているグローバルグループのマッピングが自動的にとられるようになります。

つぎに，smbldap-populate.pl を実行します。

これは，Windows でデフォルトとして扱われているユーザ情報やグループ情報により LDAP エントリを初期化する操作です。

```
#  /usr/local/sbin/smbldap-populate.pl
Using builtin directory structure
adding new entry : dc=sample,dc=com
adding new entry : ou=People,dc=sample,dc=com
adding new entry : ou=Group,dc=sample,dc=com
```

```
adding new entry : ou=Computers,dc=sample,dc=com
adding new entry : uid=Administrator,ou=People,dc=sample,dc=com
……
```

この時点で，Samba ドメインにユーザやグループの情報が登録されていればそのまま LDAP からそれらの情報を参照できるようになります。なお，リスト 4 の smb.conf の中で Administrator を admin users として指定しておくことにより，上で示したように Administrator がユーザとして登録され，Windows クライアントのドメイン参加時に Administrator が使用できるようになります。ただし，Administrator パスワードの設定を，

```
# smbldap-passwd.pl Administrator
Changing password for Administrator
……
```

により行う必要があります。

Active Directory 情報の移行

最後に，Windows サーバの Active Directory にすでにユーザ，グループ情報が登録されている場合，それらを Samba ドメインに移行する手順について説明します。

これは，Samba 3.0 以降に追加されている，net vampire 機能を利用して行います。移行は，以下の 2 ステップの手順により行います。

1）バックアップドメインコントローラ（BDC）として Samba を起動し，Windows ドメインに参加する（BDC 用 smb.conf の設定）
2）net vampire 機能を利用して Active Directory 情報を移行する

まず，Windows サーバがネットワークで接続されているものとして，Samba をバックアップドメインコントローラ（BDC）として起動します。この場合の BDC 起動用 smb.conf ファイルの設定内容はリスト 5 のようになります。

```
[global]
     dos charset = CP932
     unix charset = EUCJP-MS
     display charset = EUCJP-MS
     workgroup = TOGONINSHO
     netbios name = SAMBA30
.....................................................................
```

```
     domain logons = Yes
     preferred master = Yes
     domain master = No
     security = user
     os level = 20
.....................................................................
     add user script = /usr/local/sbin/smbldap-useradd.pl -a -m "%u"
     add group script = /usr/local/sbin/smbldap-groupadd.pl "%g"; getent group "%g"
| cut -d: -f3
     add machine script = /usr/local/sbin/smbldap-useradd.pl -w "%u"
     delete user script = /usr/local/sbin/smbldap-userdel.pl -r "%u"
     delete group script = /usr/local/sbin/smbldap-groupdel.pl "%g"
     add user to group script = /usr/local/sbin/smbldap-groupmod.pl -m "%u" "%g"
     delete user from group script = /usr/local/sbin/smbldap-groupmod.pl -x "%u" "%g"
     set primary group script = /usr/local/sbin/smbldap-groupmod.pl -g "%g" "%u"
.....................................
```

リスト5　BDC 用の smb.conf 設定

リスト5の太字で示した部分は BDC 設定用としてリスト1と異なる部分です。Windows Active Directory ドメインのドメイン名は本来 DNS タイプですが同時に NT ドメイン名もマッピングされており，TOGONINSHO は NT ドメイン名として指定されているものです。また，os level は PDC より低い 20 を指定しています。リスト後半の add……の設定ラインは，smbldap-tools との連携のための設定です。

以上の設定により Samba を起動した上で，

```
# net rpc join -S [Windows Server 名] -w [NT ドメイン名] -U Administrator
```

により，Samba のドメインコントローラを BDC として Windows ドメインに参加させます。

ここで，Samba サーバを一時停止し，net vampire コマンド

```
# net rpc vampire -S [Windows Server 名] -U Administrator
```

により，Active Directory 上のユーザアカウント，グループアカウント，マシンアカウント情報を取り込みます。なお，Samba 側のドメインは NT ドメイン準拠のため Active Directory ドメインのグループポリシー，ケルベロス認証などに関する情報は移行することができません。

最後に，Samba サーバを PDC として再起動して情報の移行作業が終了します。

以上の設定により Samba-LDAP 連携によるシステム構築が完了しました。

なお，設定に使用されたパラメータの説明や設定手順に関して一部簡略化した部分もあります。

内容の詳細は文献 [6][7][8] を参照してください。

２．GINA-LDAP 連携

２．１　システム構成

GINA-LDAP 連携は前節で紹介した Samba-LDAP 連携と同様に Windows クライアントのログオン認証に LDAP 上の情報を参照できるようにするシステムです。GINA（Graphical Identification aNd Authentification）とは，Windows のログオン管理モジュール winlogon.exe に機能拡張を与えるための DLL モジュールでユーザに公開されているものです [7]。

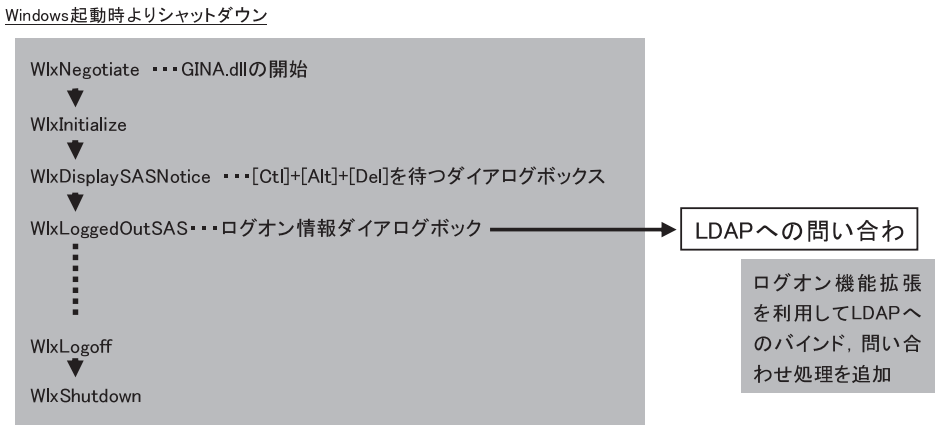

図５　GINA の実行シーケンス

基本的な実行シーケンスは図５に示したような形をとっていますが，ログオン情報ダイアログボックスを制御する WlxLoggedOutSAS プロセスの部分に LDAP へのバインドルーチンを追加すれば，LDAP 情報と Windows ログオンプロセスの連携が取れるようになります。このように機能拡張を行ったサードベンダ製 GINA モジュールとしては pGINA[9] [10]，CO-GINA[11] などがあります。

GINA の機能を利用した GINA-LDAP 連携のシステム構成は図６のようになります。

7　GINA モジュールは Windows Vista 以降，認証モデルの変更に伴い利用しなくなっています。

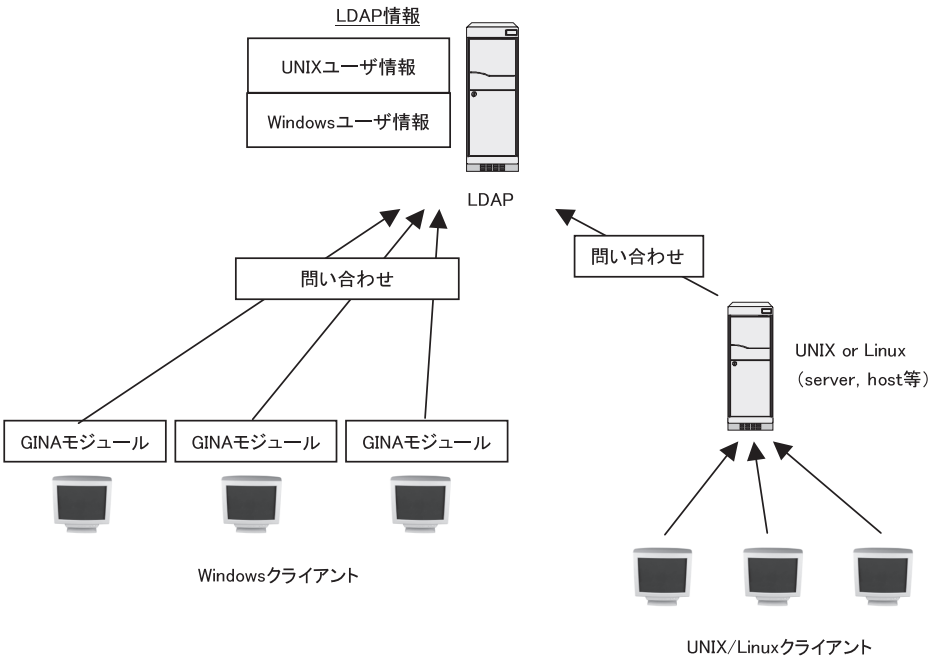

図６　GINA-LDAP 連携のためのシステム構成

２．２　GINA の設定

LDAP との連携が可能なサードベンダ製 GINA モジュールとしては pGINA（Pluggable Graphical Identification aNd Authentification）と CO-GINA があります。これらのモジュールは，LDAP 連携の基本的な考え方としては２- １で説明した内容で一致しますが，オプション機能や環境設定の方法，操作性に対するコンセプトなどの違いにより独自の設計が施されています。図７は，pGINA における LDAP 認証プロセス，すなわち LDAP バインドからログオン認証，さらにユーザ環境の構築までの流れを，従来の Windows で使われている msGINA-Active Directory 連携と比較したものです。pGINA では，ldapauth.dll という LDAP 連携オプション用モジュールを msGINA に対するアドオン DLL として挿入することにより，GINA-LDAP 連携からのログオン認証プロセスを実現しています。

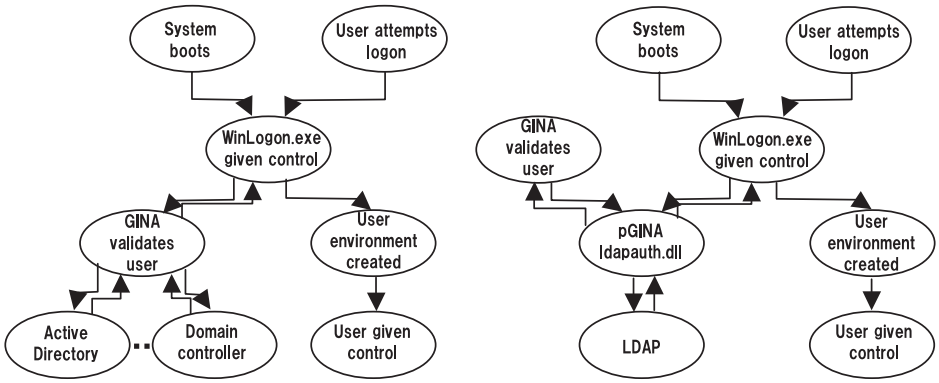

図７　pGINA における GINA-LDAP 認証プロセスの流れ [10]

また，上のように商用モジュールを導入するのではなく，GINA モジュールに LDAP バインドプロセスを書き加えることで，GINA-LDAP 連携の基本的な動作を実現することも可能です。リスト６は GINA の WlxLoggedOutSAS プロセスに挿入された LDAP バインドルーチンの一部です。

```
……
      LDAP *ld;
      int  version;
      int  status;
      LDAPMessage *result, *e;
      version = LDAP_VERSION3;
      ld = ldap_init("133.**.**.**",LDAP_PORT);
      if(ld==NULL) return 0;
      ldap_set_option(ld, LDAP_OPT_PROTOCOL_VERSION, &version);
      status = ldap_simple_bind_s(ld,"cn=Manager,dc=sample,dc=net","passwd");
      if(status != LDAP_SUCCESS)return 0;
……
```

リスト６　GINA に挿入された LDAP バインドプログラム

実際には，上のプロセスに続いて LDAP 情報を検索取得し Windows のログオン認証プロセスにそれらのデータを引き渡してユーザ ID，パスワード等の検証を行います。

２．３　GINA-LDAP 連携における留意点

以上説明したように GINA-LDAP 連携における認証プロセスの基本的な考え方は，Windows のログオンモジュールが LDAP データベースにアクセスし，取得した情報をもとにログオン判定を行うというものです。したがって，Samba-LDAP 連携の項で紹介したような，Windows ド

メイン情報に基づいて個人ごとのWindows環境が構築されるようなログオン機能をもたせるためには，モジュール自体の作りこみがある程度必要となります。また，認証時に必要な処理操作とそれに付随する動作（図８）をあらかじめ予想しなければならず，これらの内容は，GINA-LDAP連携システム構築に伴う留意点として挙げられます。

１．サーバ接続が停止したとき，認証プロセスが停止したとき
２．認証に失敗したとき
３．認証後にログオンするためのWindowsアカウント
４．認証成功後にWindows側にアカウントがない場合
５．パスワード変更手順
６．Windows側のアカウントのパスワードが異なる場合
７．Windows側のアカウントがロックされた場合
８．Windows側のパスワード変更が禁止された場合
・・・・・・・・・・・・

図８　GINA-LDAP連携システム構築時の留意点

３．SFU（Windows Services for UNIX）による統合認証

３．１　システム構成

１節のSamba-LDAP連携や２節のGINA-LDAP連携では，統合認証の形としてWindowsクライアントをLDAP上の情報をもとに認証するという方法をとっていました。統合認証の方法としては，それとは逆にUNIXクライアントのログオン認証をActive Directory上の情報をもとに行うという方法も考えられます。SFU（Windows Services for UNIX）を利用した統合認証はそのような考え方に基づくものです。

SFUとは，Windows環境とUNIX/Linux系システム環境との統合を目的としてMicrosoftから無償で提供されているUNIXシステム連携ソフトウェアです。このSFUに標準装備されているActive Directory用NISサーバ機能を利用して，Active DirectoryスキーマをNISベースで拡張することにより，UNIX/Linux側のユーザアカウント，グループアカウントをActive Directory内に格納することができます。したがって，UNIX/Linuxシステム側でActive DirectoryへのNSSマッピング，PAM認証設定を行えば，通常のLinuxクライアントからのLDAP認証と同等の機能が得られるようになります（図９）。

なお，上記の考え方に基く全体のシステム構成は図10のようになります。

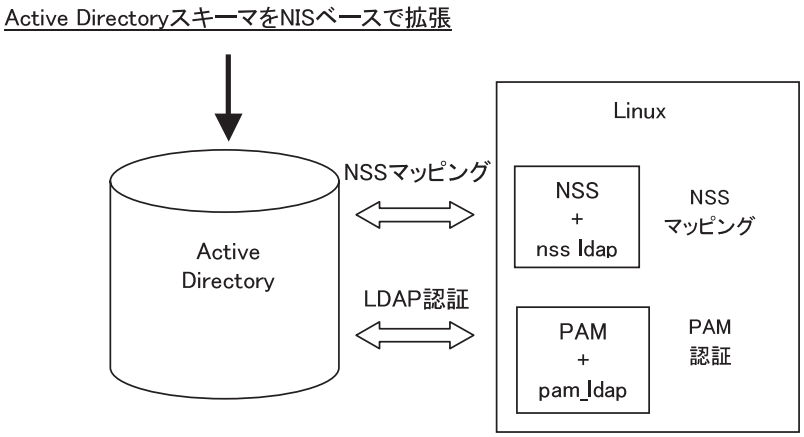

図９　Active Directory への NSS マッピング及び PAM の設定

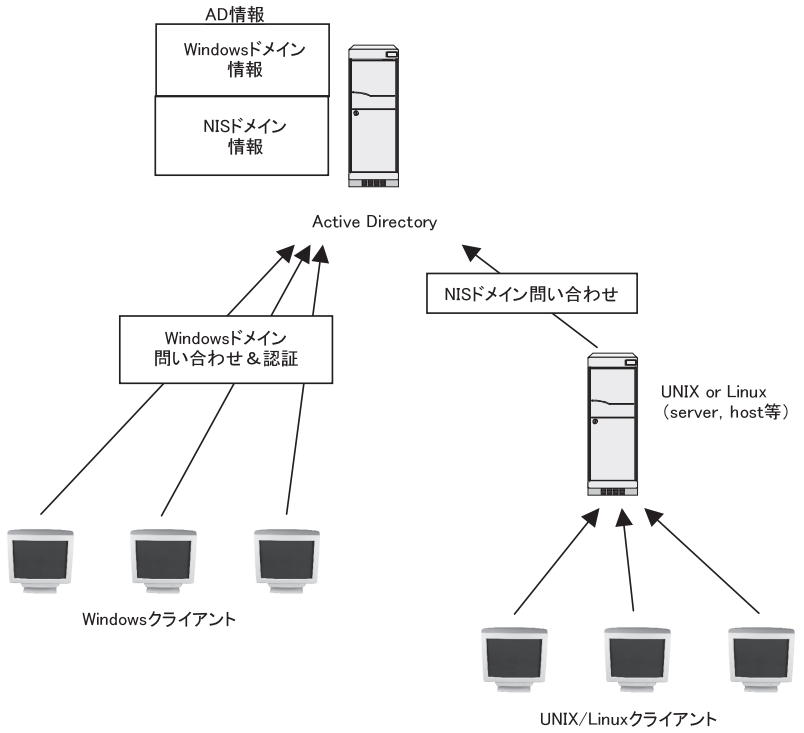

図 10　SFU を利用した統合認証システム構成

３．２　システムの構築

SFU を利用した統合認証システムの構築は，Windows Server 側の設定と UNIX 側の LDAP クライアントとしての設定の２つに分けられます。

Windows Server 側の設定

ここでは，Windows Server にはすでに SFU がインストールされているものとします。ただし，統合認証システムの構築には SFU の NIS サーバ機能のみを必要とするため，インストールすべきコンポーネントは NIS サーバのみで十分となります。

そして，Windows Server 上で「Active Directory ユーザとコンピュータ」のスナップインから UNIX 属性を持ったユーザアカウントオブジェクトの作成，プライマリグループの設定を行います（図 11）。これらの情報は，Active Directory の階層構造に組み込まれると同時に UNIX 側からの参照アカウント情報となります。

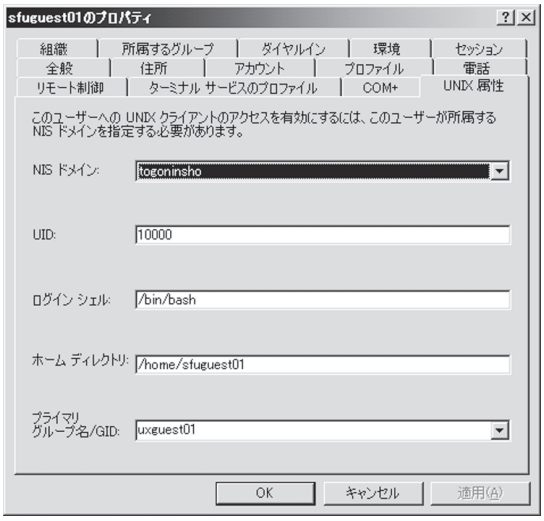

図 11　ユーザアカウント，プライマリグループの設定

さらに，Active Directory はデフォルトでは LDAP クライアントからの匿名アクセスを許可していないため，共通アカウントとしてのプロキシアカウントを作成しておく必要があります（図 12）。なお，ここで設定したプロキシアカウントはパスワードが LDAP クライアント側のバインド設定情報として ldap.conf に平文で記載されるため，Windows の管理者権限を持たないユーザとして設定しなければなりません。

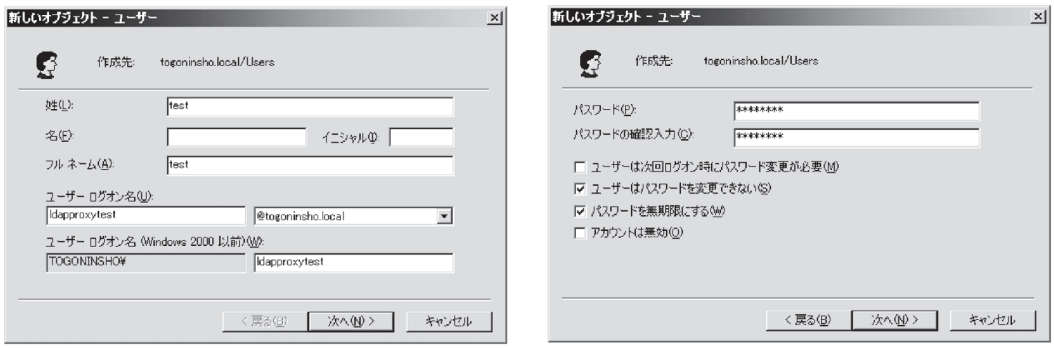

図 12　プロキシアカウントの設定

UNIX/Linux 側（LDAP クライアント）の設定

つぎに，UNIX/Linux 側の設定を行います。

まず, NSS 及び PAM の設定をとおして UNIX/Linux システムに LDAP 認証機能を追加します。LDAP 認証のための参照サーバの指定と NSS,PAM 設定方法は Samba-LDAP 連携設定の項で説明したように，RedHat 系 LinuxOS では authconfig モジュールを起動することにより簡単に行うことができます。参照サーバには Active Directory のアドレスを指定します。

なお，LDAP クライアント上のバインド設定ファイル ldap.conf は，上の手順により自動的に LDAP 参照場所などが書き換えられますが，SFU スキーマとのマッピングのためのパラメータ設定に関しては手作業で書き換える必要があります。リスト 7 は Active Directory との連携用に設定された ldap.conf の内容です。

```
# @(#)$Id: ldap.conf,v 1.34 2004/09/16 23:32:02 lukeh Exp $
host 133.**.**.**

# The distinguished name of the search base.
base dc=host,dc=sample,dc=com

# The distinguished name to bind to the server with.
binddn cn=ldapproxy,cn=users,dc=ninsho,dc=local

# The credentials to bind with.
# Optional: default is no credential.
bindpw samplePW!
# Filter to AND with uid=%s
pam_filter objectclass=user
# The user ID attribute (defaults to uid)
pam_login_attribute msSFU30Name
```

```
# RFC2307bis naming contexts
nss_base_passwd   ou=togotest,dc=ninsho,dc=local?sub
nss_base_shadow   ou=togotest,dc=ninsho,dc=local?sub#

RFC 2307 (AD) mappingsnss_map_objectclass posixAccount User
nss_map_objectclass shadowAccount User
nss_map_attribute uid sAMAccountName
nss_map_attribute uidNumber msSFU30UidNumber
nss_map_attribute gidNumber msSFU30GidNumber
nss_map_attribute homeDirectory msSFU30HomeDirectory
nss_map_objectclass posixGroup group
nss_map_attribute uniqueMember member
nss_map_attribute cn sAMAcountName
nss_map_attribute loginShell msSFU30LoginShell
nss_map_attribute gecos name
pam_password md5

# SASL mechanism for PAM authentication - use is experimental
# at present and does not support password policy control
ssl no
tls_cacertdir /etc/openldap/cacerts
```

リスト７　Active Directory との連携用に設定された ldap.conf

以上の設定により SFU による統合認証システムの構築が完了しました。

なお，設定に使用されたパラメータの説明や設定手順の詳細に関しては，文献 [12][13] を参照してください。

４．Sun Java™ System Identity Synchronization for Windows による統合認証

４．１　システム構成

最後に Sun Java™ System Identity Synchronization for Windows を利用した統合認証を紹介します。

このシステムは，これまで説明してきた統合認証の方法とは異なり，Windows 系，UNIX 系のそれぞれのディレクトリサーバと認証環境をそのまま残し，双方のサーバの情報が常に同期が取れるようにディレクトリ監視サーバを追加するものです。そのための監視サーバ Sun Java™ System Identity Synchronization for Windows が商用サーバとして Sun Microsystems, Inc. より提供されています。システム構成は図 13 に示したような形になります。

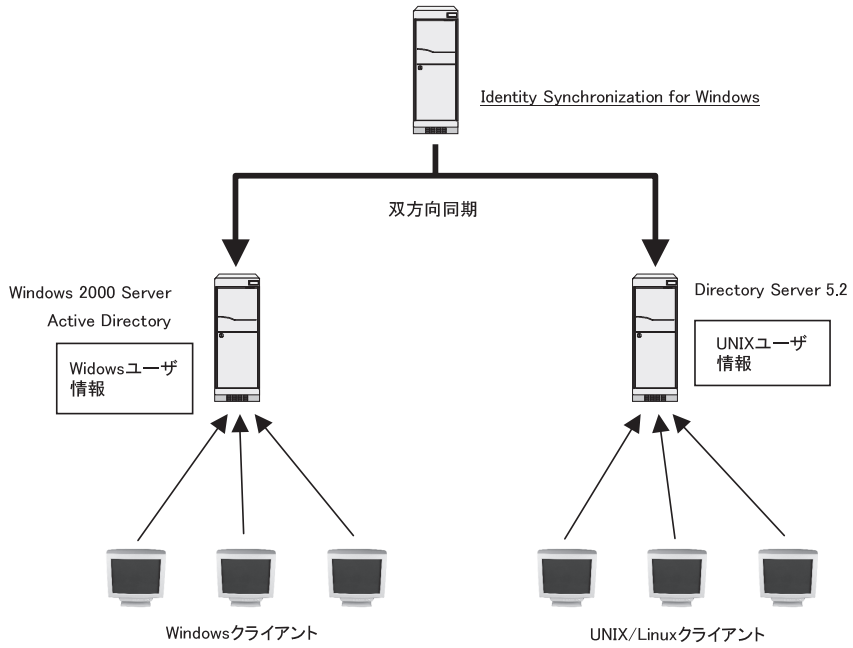

図 13　System Identity Synchronization for Windows を利用したシステム構成

なお，このシステムでは Windows,UNIX それぞれの環境は保持されるため，システムごとのデータベース管理の運用方法を変更する必要がないというメリットはありますが，適用可能なディレクトリサーバの種類が限定されることや設置コストなど，実際の導入に当たった場合に慎重に検討しなければならない項目もいくつかあげられます。

Ⅵ．おわりに

近年，ネットワーク環境が大規模かつ複雑化する中で，個人情報などさまざまな情報のセキュリティ強化やシステムの管理運用コスト節約など避けてとおることのできない問題がさらに増加しつつあります。UNIX 系 Windows 系の認証システムの統合は，そのような問題に対するソリューションの一つとして多くの企業，大学などで注目されています。今回，統合認証のためのシステムの構築の考え方，また，その構築方法についていくつか代表的な方法を紹介しました。紹介した内容は，具体的な手順について簡略化あるいは省略している部分もありますが，内容に関して興味を持ち，また，同様の統合認証を検討されている方にとって，今後，作業を進める上での一助となれば幸いです。

謝辞

ここで紹介した UNIX-Windows 統合認証のための調査，研究内容は，国立情報学研究所の最先端学術情報基盤(CSI)事業の一環として行われたものです。ここに記して謝意をあらわします。

参考文献

[1] Samba, http://us1.samba.org/samba/

[2] GINA, http://msdn.microsoft.com/msdnmag/issues/05/05/SecurityBriefs/

[3] SFU, http://www.microsoft.com/japan/technet/interopmigration/unix/sfu/default.mspx

[4] AD4Unix, http://sourceforge.net/projects/ad4unix/

[5] Sun Java™ System Identity Synchronization for Windows, http://jp.sun.com/products/software/javasystem/identitysynch/

[6] 武田保真,「徹底解説 Samba LDAP サーバ構築」, 技術評論社

[7] Red Hat Enterprise Linux 4: リファレンスガイド , http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/ja/ref-guide/s1-samba-servers.html

[8] Windows ネットワーク用統合認証サーバー構築 , http://fedorasrv.com/openldap.shtml

[9] pGINA, http://www.pgina.org/

[10] Dave Pickens and Kent Price, “Using pGINA to Authenticate Users in Microsoft Windows Environments” , http://www.sun.com/blueprints/0604/817-7043.pdf, Sun BluePrints ™ OnLine—June 2004

[11] CO-GINA, http://www.co-conv.jp/product/co-gina/

[12] 土井優子編 ,「LDAP Super Expert」, pp.159-162, 技術評論社

[13] 堀田元宣,「Active Directory と Linux によるシステム構築ガイド」, pp.155-187, 秀和システム

（くずう　かずと：名古屋大学情報連携基盤センター）