九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Mac OS X : Mac OS Xのネットワーク

内藤, 久資
名古屋大学多元数理科学研究科

https://doi.org/10.15017/1470684

出版情報：九州大学情報基盤センター広報 : 学内共同利用版. 5 (1), pp.7-46, 2005-04. 九州大学情報基盤センター
バージョン：
権利関係：

Mac OS X

－Mac OS X のネットワーク－ *

内藤 久資
名古屋大学多元数理科学研究科
naito@math.nagoya-u.ac.jp

これまでの解説では, ネットワークに接続された MacOS X のホストをスタンドアロン（単独）で利用する形態を前提としてきたが, 今回の解説では, 複数台の MacOS X のホストや, 他のプラットフォームを含めた, MacOS X のネットワーク設定を解説しよう.[1] 特に, 複数台の Mac OS X のホスト間でユーザ情報の共有やディスクの共有を行う方法に焦点を絞って解説する. また, Panther になって新たに追加された, セキュアなネットワーク接続機能である「802.1x 接続」と「VPN 接続」に関しても解説を行う.

11 MacOS X のネットワーク

MacOS X のベースシステムは UNIX (BSD) システムであるので, 基本的なネットワーク設定の考え方は UNIX のそれと同様であると考えてもよい. しかし, MacOS X のデフォールトのネットワーク設定は, 旧来の BSD システムとは大きく異なる部分を持つ. そのため, 旧来の BSD システムと同じと考え設定を行おうとすると, いろいろと戸惑うことが多いだろう. ここでは, 旧来の BSD システムとの違いを中心に, MacOS X のネットワーク設定の基本を解説し, MacOS X 同士や他のプラットフォームのホストを加えたネットワークをどのように構築するのかを調べていこう.

以下では, どのような状況をネットワークを利用して実現するのかと, そのために必要な「ネーミングサービス」について概略を解説し, 詳細な設定方法は次章で解説を行う.

11.1 ネットワーク設定の目的

UNIX システムのネットワーク設定では以下にあるような項目を設定することが基本的な内容となる.

- ユーザ情報の設定と共有
- ネットワークファイルシステムの設定

この他にも, 電子メールやウェブサーバの設定など各種のネットワークサービスの設定が考えられるが, 複数台のホスト間でデータを共有したり, 複数台の MacOS X ホストを同一の設定にして, ど

*この記事は名古屋大学情報連携基盤センターニュース（Vol. 3, No. 2, pp. 105–149）に掲載されたものを加筆・訂正したものである.「【注】」とある部分は, 九州大学広報に掲載するにあたって加筆した部分である.

[1]この原稿を執筆している２００４年３月現在, MacOS X の最新リリースは 10.3.2 である. この解説は, 特に断らない限り, MacOS X 10.3.2 に沿うものとご理解いただきたい.

【注】２００５年２月現在の最新リリースは 10.3.8 である.

のホストを利用しても同一の環境でユーザが利用できる状況を構築するには, 上記の設定を行うことが必要最小限の内容となる. 上記の設定項目のうち最初のものは「ネーミングサービス」または「ディレクトリサービス」と呼ばれるネットワークサービスによって実現される.[2]

例えば, ある部屋に複数台の MacOS X の機器が並んでいる状況で, ユーザがどのホストを利用しても全く同じ環境で利用できるためには, 最低限でも以下の状況が実現できなくてはならない.

- 同一のユーザ名とパスワードの組（認証情報）によってログインできること. また, あるホスト上でパスワードを変更した場合には, 変更したパスワードで他のホストにログインできること.
- あるホスト上で個人のファイルを作成・変更した場合には, 他のホストにもその変更が及ぶこと.

この状況のうち前者は「ユーザ情報の設定と共有」に関わり, 後者は「ネットワークファイルシステムの設定」に関わっている. さらに, 後者は個人のファイルの所有権情報を含めてファイルの作成・変更が行われなければならないので, このような状況の実現のためにはネーミングサービスが欠かすことができないサービスであることを理解していただけるだろう.

しかしながら, MacOS X のネットワーク設定で, 旧来の BSD システムと大きく異なるものがこのネーミングサービスである. 例えば, ユーザ情報（ユーザ名とパスワードの組）を格納するデータベースとしては, 旧来の BSD システムでは「BSD フラットファイル」と呼ばれる `/etc/passwd` ファイルが基本となり, NIS (Network Information Service) を利用して, 複数台のホスト間でユーザ情報を共有していた. しかし, MacOS X のデフォールト設定では, BSD フラットファイルを用いることはなく,「NetInfo」と呼ばれるネーミングサービスを用いる. また, 他のプラットフォーム（例えば, Linux, BSD, Solaris などを含む他の UNIX や Windows NT/2000/XP など, ユーザ認証を利用する Windows システム）とのユーザ情報の共有のためには LDAP と呼ばれるディレクトリサービスを用いることが推奨されている.

つぎの章では, 複数台の Mac OS X をつぎのような環境に設定することを考えてみる.

- 複数台の Mac OS X のどのホストから利用しても, 同一のユーザ認証情報を利用してログインすることができ, ログイン後は同一のホームフォルダやアプリケーションを利用できる.

このような環境を実現するためには, ユーザ認証情報の共有とネットワークを利用したディスクの共有が必要となる. そのため, ネーミングサービスを利用した, これらの共有設定の方法を考察する.

11.2　ネーミングサービスの概要

ネーミングサービスは各種のプロトコルが存在するが, ここでは MacOS X で用いられるもののうち NetInfo と LDAP に焦点を絞って, その概要をみていくことにしよう.

11.2.1　NIS

はじめに, 旧来の UNIX システムで用いられてきたネーミングサービスである NIS の概要についてまとめておこう. UNIX システムの設定に直接関わるものとして

[2]より正確には,「ネーミングサービス」とは, 複数台のホスト間でシステム設定に必要なユーザ名やホスト名の解決をサポートするサービスのことであり,「ディレクトリサービス」はシステム設定には直接は関わらないような情報も含めたデータの提供を行うサービスを指す.

- ユーザ情報 /etc/passwd （グループ情報 /etc/groups）
- ホスト情報 /etc/hosts

などのファイルが存在している. NISはこれらのファイルを1台のサーバ（NISマスタサーバ）に集約し[3], その他のホスト（NISクライアント）はマスタサーバへデータの問い合わせを行うことによりファイルの共有を行うシステムである. 同一のNISシステムを利用するホスト群は「NISドメイン」と呼ばれ, システム起動時にサーバを明示的に指定するか, ブロードキャストを用いたサーバの検索を行う.

NISは古くから利用され, 設定が容易であるという利点があるが, 階層的なデータベースの構築ができなかったり, セキュリティ上いくつかの欠点を持つことが知られている.[4]

11.2.2 NetInfo

「NetInfo」とは, NeXTSTEPで採用された, 階層的データベースを構築できるネーミングサービスである. MacOS XはNeXTSTEPを基本として構築されたBSDシステムであるため, 最も基本的なネーミングサービスとしてNetInfoが採用されている. そればかりか, スタンドアロンシステムのユーザ情報の設定にさえもBSDフラットファイル (/etc/passwd) を用いるのではなく, NetInfoのローカル・データベースを構築し, そこへの問い合わせを実現する形でユーザ情報を取得している.

NetInfoでは単にユーザ情報やホスト情報だけでなく, ネットワークファイル共有の情報など, MacOS Xのシステム設定に関わるほとんどすべての情報をサーバから提供することができる. また, 階層的なネーミングサービスを実現できるため, より細かいネーミングサービスを実現可能である.

しかしながら, NetInfoを実装するプラットフォームはMacOS XとNeXTSTEP以外には存在しないため, Linux, BSD, Solarisなど他のUNIXシステムとのデータの共有ができないことが最も大きな問題である. そのため, MacOS Xのみのシステムであれば, 設定も容易なNetInfoを用いることが一つの方法であるが, ユーザ情報などを他のプラットフォームと共有したい場合にはつぎにあげるLDAPを利用する必要がある.

11.2.3 LDAP

「LDAP」は, 単なるシステム設定のためのネーミングサービスではなく, 多くの情報を提供できるディレクトリサービスである. 近年広く用いられるようになってきた. LDAPでは単一の問い合わせに対して, そのサーバ内だけで問い合わせを解決できない場合には, 他のLDAPサーバへの問い合わせを行うことができるなど, 単なる階層的データベース以上の複雑な構成が可能であり, SSLによる暗号化通信を仕様に含む安全で高機能なディレクトリサービスである.

MacOS Xでは, ネットワークを跨ったネーミングサービスとして「将来の拡張に備えて」LDAPの利用が推奨されているだけでなく, 「アドレスブック」などの検索サービスにもLDAPを利用する機能がデフォールトで備わるなど, LDAPの利用が最大限に考えられている. また, MacOS Xのインストール時にはLDAPサーバもインストール[5]され, MacOS Xで利用する場合には, サーバも

[3]必要に応じてマスタサーバのコピーを持つ「スレーブサーバ」をおくこともできる. なお, 後述のNetInfoやLDAPでもマスタサーバのコピーを持つサーバをおくことができ, NetInfoでは「クローンサーバ」, LDAPでは「レプリカサーバ」と呼ばれる.

[4]NISの通信の安全性を保てないことだけでなく, 不特定のポートを開けておく必要があり, ポートスキャンの対象となってしまう欠点を持つ.

[5]MacOS Xにインストールされている LDAP は"OpenLDAP" である.

含めてソフトウェアを追加することなく利用することが可能であるが, サーバの設定が少しばかり面倒なのが欠点である.

12 ネットワーク設定

ここからは具体的なネットワーク設定の方法をみていこう. ここでの設定の目標は「複数台のMac OS X のホストでユーザ情報とディスクを共有」することである. すなわち, それらのホスト間では同一のユーザ情報でログインでき, どのホストでログインしても同一のホームディレクトリを共有しているという状況をつくることである.

最初にスタンドアロンシステムの NetInfo の状況を確認した後, NetInfo を用いて複数の MacOS X ホスト間でのユーザデータベースの共有を行おう. さらに, 既存の LDAP サーバを利用して, ユーザデータベースの共有を行うことを考える.[6] 最後に, NetInfo, LDAP を利用してディスクの共有を行う方法を考察する.

12.1 スタンドアロンシステム

スタンドアロンのシステム, すなわち, 他のホストとユーザ情報などの共有を行っていないシステムの状況を確認しておこう.

スタンドアロンのシステムで「ターミナル」アプリケーションを開き, /etc/passwd ファイルをみてみると

```
nobody:*:-2:-2:Unprivileged User:/nohome:/noshell
root:*:0:0:System Administrator:/var/root:/bin/tcsh
daemon:*:1:1:System Services:/var/root:/noshell
smmsp:*:25:25:Sendmail User:/private/etc/mail:/noshell
www:*:70:70:World Wide Web Server:/Library/WebServer:/noshell
mysql:*:74:74:MySQL Server:/nohome:/noshell
sshd:*:75:75:sshd Privilege separation:/var/empty:/noshell
unknown:*:99:99:Unknown User:/nohome:/noshell
```

という内容であり, インストール時に指定した「ユーザ」に関する情報が含まれていないことがわかる.[7]

12.1.1 NetInfo データベースをみる

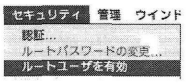
ここで,「ユーティリティ」フォルダ内にある「NetInfo Manager」を開いてみよう.

[6]実際に利用できるのは NetInfo, LDAP のいずれか一方である.

[7]/etc/passwd ファイルの読み方は以下のとおりである. 1 行に一人のユーザの情報があり, 各フィールドは ":" で区切られている. 第一フィールドが「ユーザ名」, 第二フィールドが「暗号化されたパスワード」である.

ここで表示されているものはNetInfoの「Localドメイン」と呼ばれるもので，システムインストール時に設定された内容である．左のウィンドウ中央の列には，このデータベースに含まれる「ディレクトリ」と呼ばれる各種データベースがリストされている．その中で「usersディレクトリ」がユーザ情報を含むデータベースであり，その中をみると，右ウインドウの右列のように多くの（システムに関する）ユーザ情報だけでなく，下段のようにユーザ「naito」に対する情報が含まれていることがわかる．

またgroupsをみると，adminグループにrootユーザとnaitoが含まれていることがわかるが，前回に解説した「管理者権限」を持つユーザとは，adminグループに属しているユーザのことであり，その設定がNetInfoデータベースによって制御されていることがわかる．また，右はrootユーザの情報を示しているが，デフォールトではrootユーザにはパスワードが設定されていないことに注意しよう．NetInfo Managerのメニューでは「ルートユーザを有効」という項目があり，ルートユーザのパスワードを設定することが可能である．



しかしながら, 管理者権限を持つユーザの権限だけでシステム管理は十分なはずであるので, よほどの理由がない限りルートユーザを有効にする必要はない.

12.1.2 NetInfo データベースにデータを追加する

ここで, Local ドメインに対してデータを追加する方法を考えてみよう. 管理者権限を持つユーザであれば, 「🔑」マークをクリックし, NetInfo データベースを書き換える権限を入手すれば, 以下の図のように直接ウインドウ内で値などを書き換えるか, メニューを開いて値やプロパティを挿入・変更すればよい.

しかし, この方法はシステムに必要不可欠なデータを間違って削除することになりかねないので, よほどの理由がない限りスタンドアロンシステムでは NetInfo データベースを直接いじる必要はない.

12.2 NetInfo の利用

ここでは「local」ではない NetInfo データベースにユーザ情報を登録して, その情報を参照できるように設定してみよう.

12.2.1 NetInfo データベースの構築

NetInfo では他のホストの「local」のデータベースを参照することはできないため, 新しく「local」ではない NetInfo ドメイン（データベース）を構築する必要がある. そのデータベースは自身のホストに設定して自身からも他のホストからも参照できるように設定することが可能であるので, ここでは自身のホストに「local」ではない NetInfo ドメインを設定しよう. そのためには以下の手順を実行する必要がある.

1. そのホスト上で nibindd と呼ばれる NetInfo のサーバプログラムが起動するように設定する.
2. 新規に NetInfo ドメインを作成する.
3. 作成した NetInfo ドメインにデータベースを構築する.
4. 作成した NetInfo ドメインを参照できるように設定する.

以下で実際の手順を解説していこう. この手順の中では, いくつかは「ターミナル」アプリケーションからコマンドを入力する必要がある.

12.2.1.1 NetInfo サーバの起動 MacOS X の各種サーバプログラムの起動は /etc/hostconfig という設定ファイルで制御されている. 通常 (デフォールトインストール) の状態では, /etc/hostconfig の中には

```
NETINFOSERVER=-AUTOMATIC-
```

と書かれた行が存在する. ここを emacs または vi などのエディタを用いて

```
NETINFOSERVER=-YES-
```

と修正する. ここで /etc/hostconfig の修正のためには管理者権限が必要となるため, 最初に sudo -s というコマンド[8]を入力し, パスワードとして管理者ユーザ自身のパスワードを入力して, 管理者モードに移行する必要がある. この時, /etc/hostconfig の他の行を決していじってはいけない.[9]

実際に入力するコマンド列は以下のようになる.

```
% sudo -s
Password:  *****
# cp /etc/hostconfig /etc/hostconfig.dist
# emacs /etc/hostconfig
# exit
```

この設定が終わったらホストを再起動する.

12.2.1.2 NetInfo ドメインの作成 再起動後に nidomain -l というコマンド[10]を入力してみる. するとつぎのような出力を得る.

```
% nidomain -l
tag=local udp=1033 tcp=1033
```

これは, 「local」という名前の NetInfo ドメインがこのホスト上に存在することを示している.[11]

新規に作成する NetInfo ドメインの名前を「test」と設定する場合には

```
% nidomain -m test
```

[8] sudo コマンドは, 管理者が一時的に root 特権を取得するためのコマンドであり, -s オプションを用いると, root 特権を持つシェルを起動することができる.

[9] /etc/hostconfig のいくつかの行は「システム環境設定」などから変更が加えられているため, 変に書き換えてしまうとシステム環境設定が正常に動作しなくなる可能性がある.

[10] nidomain は NetInfo ドメインの作成・削除などを行うコマンドである.

[11] 1033 という数値は異なっている可能性がある この数値は「local」ドメインが利用している TCP と UDP のポート番号を示している.

と入力する.[12] その後 nidomain -l コマンドで確認し,

```
% nidomain -l
tag=local udp=1033 tcp=1033
tag=test udp=1001 tcp=859
```

という出力を得ることができれば,「test」ドメインを作成できたことが確認できる.

この時点で「test」ドメインに格納されているデータベースをみるためには, niutil コマンドを用いる.

```
% niutil -list -t localhost/test /
1 machines
% niutil -list -t localhost/test /machines
2 myhost
% niutil niutil -read -t localhost/test /machines/myhost
name:  myhost
ip_address:  172.16.30.221
serves:  ./test
```

niutil コマンドは NetInfo ドメインを指定して, そのデータベースの閲覧・変更などを行うコマンドであり, ここで利用したオプションは以下のようなものである.

- -list オプションをつけてデータベースを指定すると, そのデータベースのデータのリストをみることができる. 最初のコマンドでは, ドメインのルートデータベースを出力している. そこには machines というデータベースのみが存在していることがわかる. つぎに /machines のリストを取ると, この NetInfo ドメインが存在するホストが記述されていることがわかる.
- -read オプションを指定すると, 各データベースのデータを得ることができる. この例では /machines データベースの myhost のデータを表示している.

12.2.1.3 NetInfo ドメインにデータベースを構築する　今回構築したいデータベースはユーザ情報のデータベースである. ユーザ情報データベースを構築するためには, niload コマンドを用いるのが簡単である.

はじめに, nidump コマンドを用いて, 既存のユーザ情報データベースを出力しよう.

```
% nidump passwd -t localhost/local | grep "^naito:"
naito:XXXXXXXXXXXXX:501:20::0:0:Hisashi NAITO:/Users/naito:/bin/tcsh
```

このように nidump コマンドは, NetInfo ドメインに含まれるデータベースのうち, 従来の UNIX システムで用いられている基本的なデータベース（この場合は passwd データベース）を, 既存の passwd 形式で出力するものである.

上記の出力をファイルに保存し, ユーザ名, ユーザ ID などを変更して,「test」のユーザ情報データベースに入力してみよう. そのためには, 上記の出力結果を

```
testuser:XXXXXXXXXXXXX:502:20::0:0:test user:/Users/testuser:/bin/tcsh
```

と変更し, ファイルに保存しておく.（そのファイル名は /tmp/passwd としておこう）その後, 以下のように niload コマンドで NetInfo データベースにデータを登録し, 登録内容を確認する.

[12] この操作にも管理者権限が必要なため, 実際には "sudo nidomain -m test" と入力する必要がある.

```
% niload -m passwd -t localhost/test < /tmp/passwd
% nidump passwd -t localhost/test
testuser:XXXXXXXXXXXXX:502:20::0:0:testuser:/Users/testuser:/bin/tcsh
```

また, niutil で「test」を表示すれば,

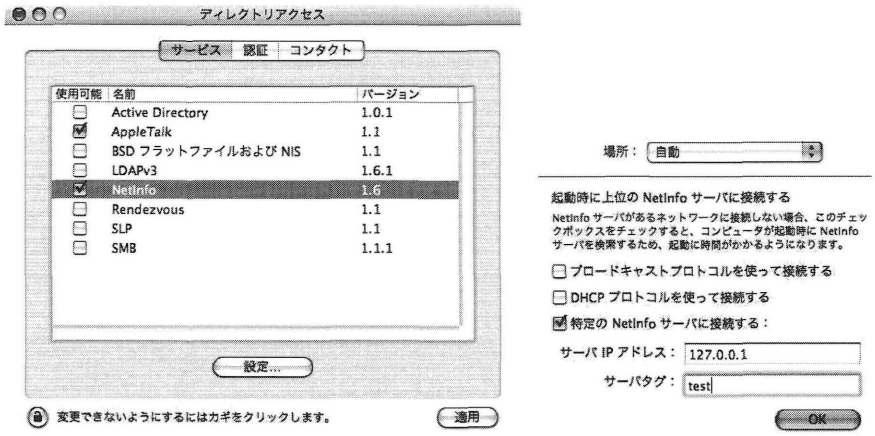
```
% niutil -list -t localhost/test /
1 machines
3 users
% niutil -list -t localhost/test /users
4 testuser
% niutil -read -t localhost/test /users/testuser
home:  /Users/testuser
name:  testuser
passwd:  XXXXXXXXXXX
realname:  testuser
shell:  /bin/tcsh
uid:  502
```

となり, users データベースに testuser のデータが登録されたことがわかる.

12.2.1.4 NetInfo ドメインの関連付け この時点では「local」ドメインと「test」ドメインとの間には何の関連もなく,「test」に追加したユーザ情報データベースを参照することができない. 既存の「local」ドメインに「test」を関連づけるためには,「local」ドメインの「上位層」として「test」を設定する必要がある. 各 NetInfo ドメインの上位層を確認するためには niutil コマンドを -rparent オプションをつけて利用する.

```
% niutil -rparent -t localhost/local
root domain:  no parent
```

この段階では「local」ドメインの上位層は存在しないことがわかる. ここで,「ディレクトリアクセス」ユーティリティを開き, NetInfo の設定を行う. （下の左図）ディレクトリアクセスの NetInfo を開く[13]と, 右図のように NetInfo サーバの上位層を指定することができるので, ここに上位層の NetInfo ドメインがあるホストの IP アドレスとそのドメイン名を指定する.

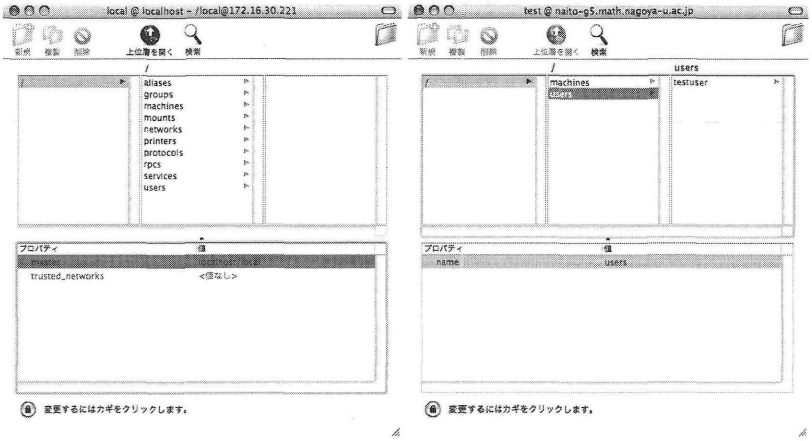


[13]"NetInfo" を選択して「設定」をクリックする.

すると,

```
% niutil -rparent -t localhost/local
localhost/test
```

となり, 上位層に "localhost/test" が指定されたことがわかる. この時点で NetInfo マネージャを開き,「local」データベースをみてみると, 下左図のように「上位層を開く」が有効になり, そこをクリックすると, 右図のように「test」ドメインを参照することができる.



なお, この状態では「test」ドメインに対して GUI から変更を加えることができない. なぜなら, 各ドメインに対する変更を行うには, 各ドメインの管理者権限が必要となるからである. もし,「test」ドメインに対して GUI を用いた変更を加えたいときにはつぎの 2 つの設定を行う必要がある.

- 「test」ドメインの users に root （UID の値が 0 のユーザ）を加える. （パスワード欄の値は "*" でかまわない）
- 「test」ドメインに admin グループを含む "groups" データベースを追加し, そのメンバーに root と管理者権限を与えるユーザを追加する. そのユーザは「test」ドメイン内部または,「test」ドメインの上位層に存在するユーザでなくてはならない.

この設定を行っておくと, 指定した管理者ユーザの権限で GUI からデータベースの変更を行うことができる. なお, ここで設定した「local」ドメインの上位層の情報は `/Library/SystemConfiguration/preferences.plist` に格納されている.

12.2.1.5　設定の確認　このようにして設定したユーザ情報を利用してログインができるかどうかを確認しておこう. それ以前に実行しておかなくてはいけないことは, 新たに作成した testuser の「ホームディレクトリ」を作成しておくことであるが, これは指定したホームディレクトリを作成して, その所有者を testuser にしておけばよい. （各種のフォルダなどは, 最初のログイン時に自動的に作成される）

ここで一旦ログアウトして, ログインウインドウをみると, これまでにはなかった「その他のネットワークユーザ」というアイコンがあらわれている.

これをクリックして, ユーザ名とパスワードを入力すれば, 「test」ドメインに設定したユーザ名を利用してログインすることができる.

なお, 「local」ドメインに属さないユーザのパスワード設定などは「アカウント」設定からは設定できないので, 上位層のドメインのユーザのパスワードは何らかの形で別に設定しておく必要がある. 実際には passwd コマンドを利用して新規パスワードを設定するのが最も簡単である.[14]

12.2.1.6 セキュリティ 以上の設定では, 新たに作成した「test」ドメインに対するアクセス制御は設定されていない. 実際, 他の MacOS X のホストから niutil を使って「test」ドメインにアクセスを行うと,

```
% niutil -list -t 172.16.xx.xxx/test /
1 machines
3 users
5 groups
```

のように, すべてのデータを閲覧可能となっている.[15] これを信頼できる他のホストからのみアクセスできるように設定するには, 「test」ドメインに新たな設定を行う必要があり, 「test」ドメインに対して trusted_networks というプロパティを追加する.

```
% niutil -t -createprop localhost/test / trusted_networks 172.16.254.0/24
% niutil -t -read localhost/test /
master:  myhost
trusted_networks:  172.16.254.0/24
```

このように NetInfo ドメインに対して trusted_networks プロパティを指定すると, そこで指定されたネットワークに属しないホストからの NetInfo の情報の読み取りが拒否される.[16]

12.2.2 NetInfo の階層化

ここまでで解説した NetInfo の利用法は, ただ一つの上位層を持つ NetInfo の構造であった. 以下ではより複雑な構造を持つ NetInfo の設定を考えてみよう. ここで試してみるのはつぎのような設定である.

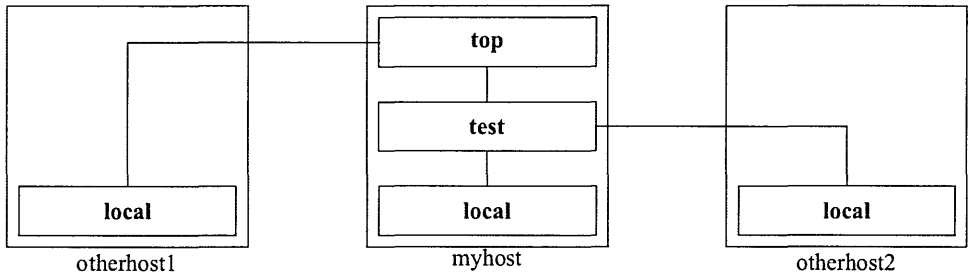
[14]別の方法としては, 何らかの方法で初期パスワードの暗号化文字列を生成して, niload でデータベースを入力する際に, その文字列を最初からデータベースに入れてしまう方法がある.

[15]実際にはファイヤーウォール設定が有効になっているとアクセスできない.

[16]筆者の経験では, ここに "172.16.254.1" または 172.16.254.1/32" のような「ホストアドレス」を書いた場合にはアクセス制限を実現することができなかった. マスクの深さは 31 以下にしなければアクセス制限を実現できないようである.

- あるホストに「test」,「top」という2つのNetInfoドメインを設定する.
- 「test」の上位ドメインとして「top」を設定する.
- 他のホストotherhost1の「local」の上位ドメインとして「top」を設定する.
- 他のホストotherhost2の「local」の上位ドメインとして「test」を設定する.

この関係を図であらわせば以下のようになる.

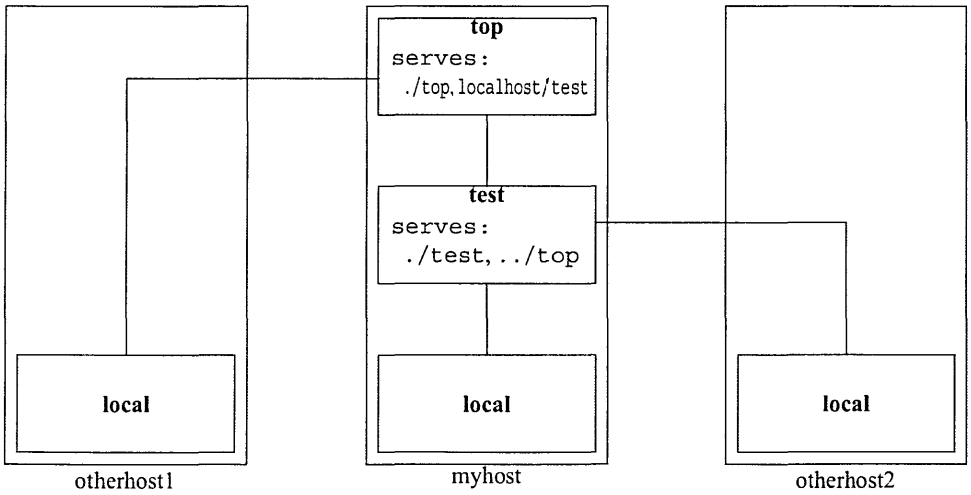


この時「local」ドメインの上位層を指定するためには,前に利用したディレクトリアクセスのNetInfoの「設定」で行うので,otherhost1, otherhost2のNetInfoの「設定」で,「local」ドメインの上位層として,それぞれ,「top」と「test」を指定する.

したがって,本質的に必要な設定は「test」の上位層に「top」を指定することである.そのためには以下の設定を行います.

「top」ドメインの設定　「top」ドメインの"/machines/myhost"にservesというプロパティを作成して,値を"./top","localhost/test"に設定する.

「test」ドメインの設定　「test」ドメインの"/machines/myhost"に"serves"というプロパティを作成して,値を"./test","../top"に設定する.



実際にこの設定を行うにはつぎのコマンドを入力すればよい.

```
% niutil -t -appendprop localhost/top /machines/myhost serves ./top
% niutil -t -appendprop localhost/top /machines/myhost serves localhost/test
% niutil -t -appendprop localhost/test /machines/myhost serves ./test
% niutil -t -appendprop localhost/test /machines/myhost serves ../top
```

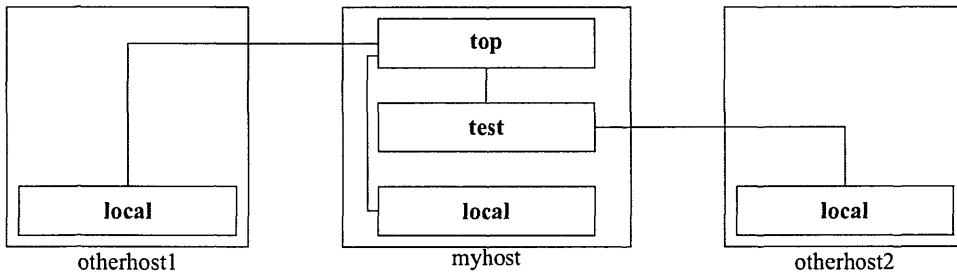
さらに，「test」の上位層が「top」になっていることを確かめるには

```
% niutil -t -rparent localhost/test /machines/myhost
localhost/top
```

という出力が得られればよい．もし“root domain: no parent”という結果がでてきた場合には，nibindd プロセスに対して HUP シグナルを送る．

この階層的な NetInfo の設定の利用例を挙げておこう．いま，「top」ドメインには研究室の教員のユーザ情報を入れ，「test」ドメインには研究室の学生のユーザ情報を入れておく．すると，myhost と otherhost2 は「test」ドメインを参照可能であるので，研究室の教員と学生の両方が利用できるが，otherhost1 は「test」ドメインを参照していないので，学生は利用ができず，教員だけが利用可能になる．

しかし，この設定ではサーバである myhost に学生がアクセス可能となり，少しばかりセキュリティに問題があると考えられる．その場合には myhost の「local」ドメインの上位層に「top」を指定しておけば，myhost は教員だけがアクセスできる状態を実現できる．



12.2.3 NetInfo のセキュリティ上の問題点

ここまでで解説してきた NetInfo は rpc を基本としたサービスであることに起因するセキュリティ上の問題がある．各 NetInfo ドメインに割り当てられるポートは portmapper を用いて動的に決定される．portmapper 自身は 111 番ポートを利用しているが，各 NetInfo ドメインが利用するポート番号は，起動ごとに異なる値となり，portmapper に問い合わせを発行しなければポート番号を知ることができない．したがって，portmapper が使う 111 番ポートをファイヤーウォールで防御することは可能であるが，各 NetInfo ドメインが使うポートをファイヤーウォールで防御しようとすると，利用する可能性のある 1024 番未満のすべてのポートを開けておく必要がでてくる．つまり，NetInfo はファイヤーウォール設定と親和性の悪いサービスであることがわかる．[17] また，後述の LDAP は SSL による通信路の暗号化が容易に実現できるが，NetInfo では通信の暗号化を行うことは（NetInfo 自身の機能では）実現できない．

このように NetInfo は Mac OS X 上で余分なソフトウェアの設定なしに利用でき，階層構成も可能という，それなりに優れたネーミングサービスの方法であるが，その設計思想が NeXTSTEP の時代から大きく進んでいるわけではなく，現状のネットワークセキュリティという観点からは問題があることがわかる．[18] したがって，可能であればつぎに述べる LDAP サービスを利用する方が，セキュリティ上の観点からも，機能的にも望ましい結果を得ることができると考えられる．

[17] この事情は NIS でも同じである．もちろん，NetInfo のクライアントになるホストに対してのみ通過を許可する設定を書けばよい．

[18] もちろん，信頼できる（ファイヤーウォールで分離された）ローカルネットワークの内部で利用する限りでは，非常に優れたサービスと考えることができる．

12.3 LDAP の利用

NetInfo の設定では, ファイヤーウォール設定との相性が悪く, その利用が MacOS X に限られるなどの不都合な点が多々ある. それに代って利用できるのが LDAP を利用したディレクトリアクセスの方法である. 前にも述べたとおり LDAP は種々の UNIX システムからも共通に利用できるディレクトリサービスであり, LDAP サーバの設定さえきちんとできていれば, MacOS X での利用が非常に容易であり, セキュリティも十分に考慮されているため, 非常に安全なシステムであると考えられる. また, ユーザ認証などの手段だけでなく, 「アドレスブック」を利用したユーザ検索にも利用できるため, 極めて有用なディレクトリサービスと考えられる.

ここでは MacOS X 上でユーザ認証などに LDAP を利用する方法, 「アドレスブック」を利用したユーザ検索の方法を解説しよう. また, MacOS X 上での LDAP サーバの構築方法についても概略を解説しよう.

12.3.1 LDAP クライアントの設定

ここでは, LDAP サーバが他の（または同一の）ホスト上で動作している場合に, LDAP サーバのデータベースを利用してユーザ設定を行う方法を考えてみよう. ユーザ認証のために LDAP クライアントを設定するには「ディレクトリアクセス」で設定を行うのであるが, そのための手順は以下のとおりである.

1. LDAPv3 クライアントの設定を行う.
   (a) LDAPv3 を有効にする.
   (b) LDAP サーバを指定する.
   (c) 検索ベース DN と認証データを設定する.[19]
2. MacOS X のユーザ認証及びネームサービスが LDAPv3 を向くように設定する.

以下ではこの手順の詳細をみていこう.

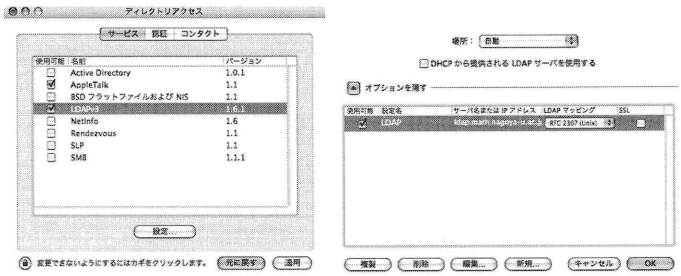
はじめに LDAPv3 クライアントの設定を行う. この設定手順で事前に調べておかなければいけない情報は

- LDAP サーバの FQDN または IP アドレス
- LDAP サーバの検索ベース DN
- LDAP サーバへの接続時に用いる認証データ

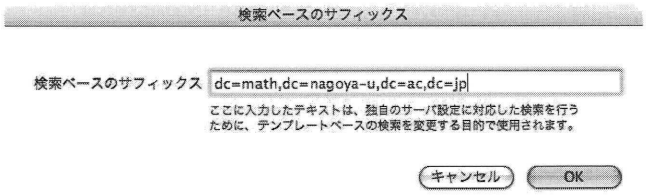
である.

そのためには「ディレクトリアクセス」を開いて “LDAPv3” を選択し「設定」をクリックする. すると下図の右のように LDAP サーバの設定ウインドウが開く.

[19] DN (Distinguished Name) の意味については, p. 17 を参照していただきたい.

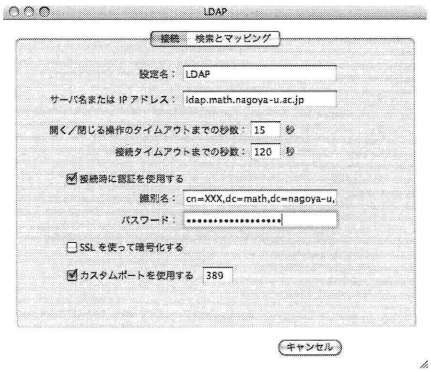


ここで「サーバ名」に LDAP サーバの FQDN または IP アドレスを入力し，「LDAP マッピング」として “RFC 2307 (UNIX)” を選択すると，以下のようなウインドウが開く．[20]



ここには LDAP サーバに格納されている（ユーザ情報などの）データベースの DN (Distinguished Name) に共通する DN （ベース DN）を入力する．

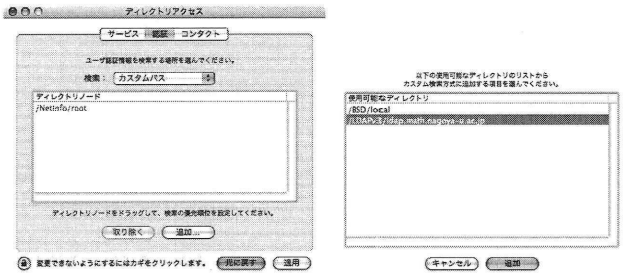
つぎに前の右図の「編集」ボタンをクリックして下図のウインドウを開き，LDAP サーバへのアクセス時の認証情報を入力する．



ここで入力する情報は「識別名」と「パスワード」のみで十分で，「検索とマッピング」タブを開く必要はない．ここまでで LDAPv3 クライアントの設定が終了した．

つぎに MacOS X のユーザ認証情報（ログイン時に入力するユーザ名とパスワードの組）を LDAP サーバから取得するための設定を行う．そのためには「ディレクトリアクセス」の「認証」タブを選択し，下左図のウインドウを開く．

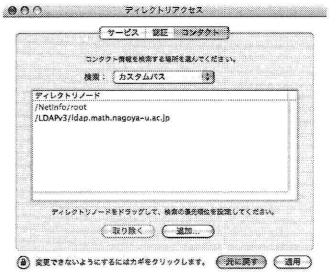
[20] LDAP サーバが SSL に対応しているのであれば “SSL” にチェックを入れておくのが望ましい．



ここで「検索」メニューから「カスタムパス」を選択すると，上右図のウインドウが開くので，

`/LDAPv3/ldap.math.nagoya-u.ac.jp`

を選択し「追加」をクリックする.[21] すると，つぎの図のように「ディレクトリノード」の LDAPv3 のサーバが指定される.



これを「コンタクト」タブを開いて同じ操作を繰り返す.

以上で LDAP を利用したユーザ認証とネームサービスの設定が終了したのだが，はじめてこの設定を行うと，何をやっているのか理解できないことが多いと思われるので，この設定の意味と LDAP の設定にあらわれる DN などの用語について説明しておこう.

ディレクトリアクセスの設定の意味　「ディレクトリアクセス」で設定する内容は以下の２項目である.

1. ユーザ認証のための情報をどのディレクトリサービス（ネームサービス）から取得するか.
2. ユーザ認証情報以外の情報（ホスト名の解決や起動時のディスクのマウントなど）をどのディレクトリサービスから取得するか.

前者を設定するのが「認証」タブで指定したサービスであり，後者を設定するのが「コンタクト」タブで指定したものである. これらには複数のサービスを（優先順位をつけて）指定することができ，多様なネームサービスから情報を得ることができる. しかしながら，常に最優先となるのが “NetInfo「local」ドメイン” であり[22]，これを削除することはできない. また，ホスト名の解決のためには “NetInfo「local」ドメイン” で解決できない場合には，通常の DNS を用いた解決が行われる. そのため，ネームサービス内に明示的に DNS を指定する必要はない.

[21] もちろん `ldap.math.nagoya-u.ac.jp` の部分は，LDAP サーバの名称であるので，一般には異なった表示となる.
[22] ディレクトリマネージャでは “NetInfo root” と表現されている. これは，「local」ドメインとその上位層を含めた NetInfo ドメインをあらわす.

LDAP の DN とは つぎに, LDAP の設定で用いられる "DN" というものの意味をみていこう. LDAP サーバに格納されたデータベースの各項目を区別するための識別子のことを DN (Distinguished Name) と呼ぶ. LDAP データベースはそれ自身が階層的な構造を持ち, さらに他の LDAP サーバに格納されたデータベースを参照する機能を持つため, 原則としては, データベースの各項目は「世界中で一意な識別子」を持つ必要がある. そのため, (例えば) `math.nagoya-u.ac.jp` ドメインで利用される LDAP データベースには

`dc=math,dc=nagoya-u,dc=ac,dc=jp`

または

`o=Graduate_School_of_Mathematics_Nagoya_University_JAPAN`

などの `math.nagoya-u.ac.jp` ドメインを明示する識別子の下にデータベースの各項目に識別子を与えていく必要がある. この `dc=math,dc=nagoya-u,dc=ac,dc=jp` 以下には, ユーザ情報をあらわす

`ou=people,dc=math,dc=nagoya-u,dc=ac,dc=jp`

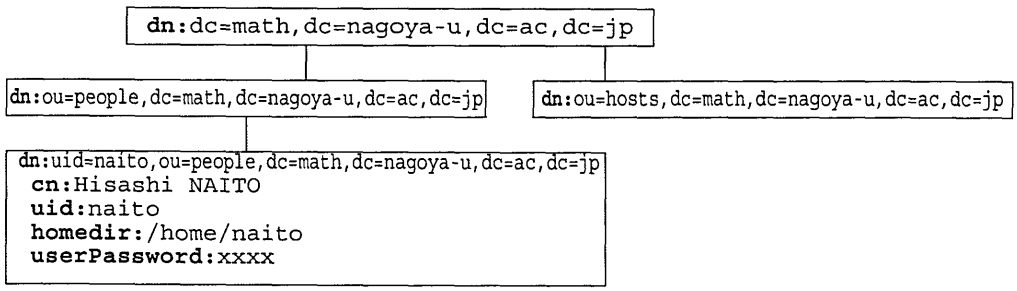
という DN を持つデータベース, ホスト情報をあらわす

`ou=hosts,dc=math,dc=nagoya-u,dc=ac,dc=jp`

という DN を持つデータベースなどがあり, 一人のユーザのユーザ情報は

`uid=naito,ou=people,dc=math,dc=nagoya-u,dc=ac,dc=jp`

という DN を持つデータベース (データベース項目) として格納されている.



すると LDAP サーバに接続する場合の「検索ベース DN」は検索対象のデータベースをそれ以下に制限する意味を持つことがわかる.

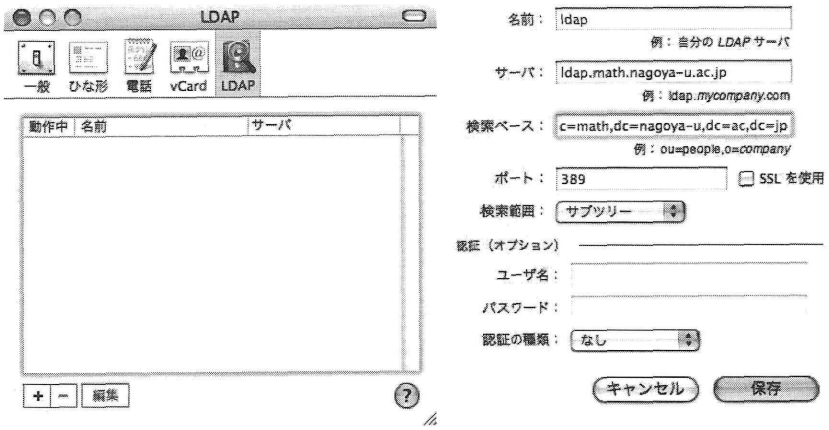
さて, LDAP サーバに接続する場合の「認証」とはどういう意味があるのかを調べておこう. 以下に「アドレスブック」の利用で述べるように, LDAP サーバは単なる認証サーバではなく, 一般のユーザに対して開放されたディレクトリサーバである. そのため, 一般のユーザからのアクセスに対して, 各ユーザの「パスワード情報」を丸見えにするわけにはいかない. したがって, クライアントの要求に対して「パスワード情報」を渡すためには, LDAP サーバへの接続に対して, 何かしらの認証を行う必要がある. そのために用いる方法が「バインド」と呼ばれる手続きである. LDAP サーバにバインドするためには, 認証に必要な全データベースの検索を許可された DN と, その DN の「パスワード」を用いてアクセスを行う.[23] すなわち, LDAP 認証を利用する場合の「LDAP サーバに対する認証」とは, ユーザ認証に必要な全情報を得るための DN とそのパスワードの組を用いてバインドの要求をすることである.

[23] LDAP では, バインドに用いた DN によって, どのデータベース領域のどの属性にアクセス可能かを細かに制御できる.

12.3.2 LDAP を利用した検索の設定

LDAP サーバがある場合には「アドレスブック」や「メール」などで LDAP サーバを用いたユーザやメールアドレスなどの検索が可能になる. ここでは「アドレスブック」を例にとり, LDAP を利用した検索の方法について調べてみよう.

アドレスブックの通常の設定では各ユーザが実際にアドレスを打ち込んだ "vCard" を用意する必要がある. しかし LDAP サーバが適切に設定されていれば[24]LDAP サーバに問い合わせを行うことによりメールアドレスなどの検索が可能となる. そのためにはアドレスブックに LDAP サーバを指定する必要がある. アドレスブックのメニューの「環境設定」を開くと左図のようなウインドウが開く. ここで左下の "+" マークをクリックすると, 下右図のようなウインドウが開き, LDAP サーバとその検索設定を設定することができる.



ここで,「サーバ」には問い合わせを行う権限のある LDAP サーバの FQDN を,「検索ベース」にはユーザ情報を検索するために必要な DN を入力[25] [26]して「保存」をクリックすると, つぎのように, 検索のための LDAP サーバが設定される.

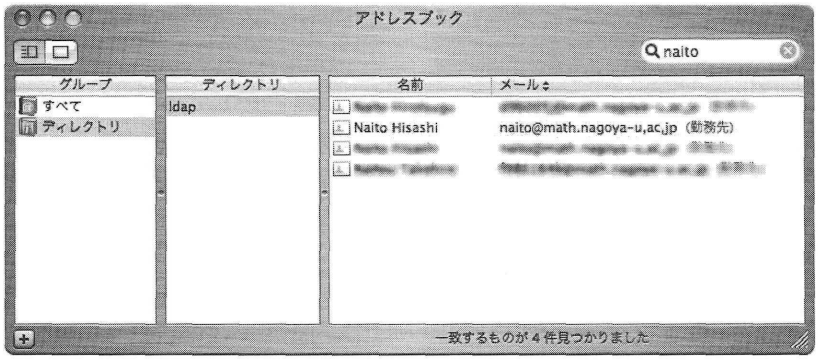
[24]検索対象となる LDAP の属性は cn, sn, givenName, mail であり, mail 属性の内容が「メールアドレス」として表示されるため, LDAP サーバ側でこれらの属性の設定を行っておかなければならない.

[25]ユーザ認証に LDAP を利用しようとした前節の状況では, 各ユーザのパスワードデータ（多くの LDAP サーバで userPassword 属性として指定されている）を読み取る必要があった. セキュリティ上の理由により, userPassword 属性の読み取りのためにはユーザ認証が必要なように LDAP サーバを設定する. 一方, 今回の「メールアドレス」などの検索のためには userPassword 属性まで読み取る必要はない. そのため, LDAP サーバの設定として, **IP アドレスなどによるフィルタリングのみ**を行い, 認証なしで LDAP サーバにアクセスできるようにするのがよいだろう.

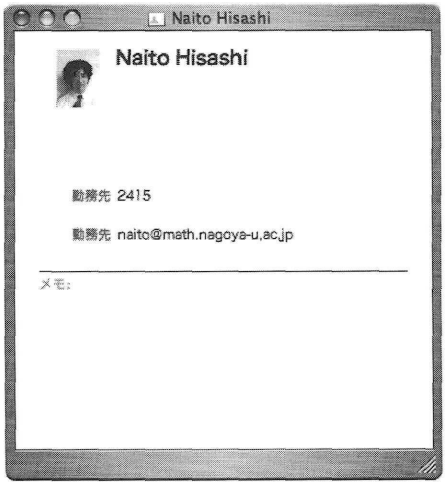
よりセキュリティを強化するのであれば, LDAP サーバへのすべてのアクセスは「パスワード」が必要なように設定する方法もある. その場合には「アドレスブック」の環境設定で, オプションとなっている「認証」の部分にユーザ名とパスワードを記入すればよい. この場合のユーザ名とは, そのユーザを示す DN であることに注意が必要である.

[26]ユーザ情報は, 通常 LDAP サーバに "ou=people, dc=. . . ." という DN をもつデータとして格納されている. そのため, 前節の認証のための検索ベース DN の前に ou=people をつけたものをアドレスブックの検索ベース DN として用いる.

ここで，アドレスブックのウインドウで「グループ」に「ディレクトリ」を指定し，右上の検索窓に検索したい文字列を入力すると，LDAP サーバへの問い合わせが発生し，文字列にマッチしたデータが表示される．[27]



ここで目的のデータをクリックすれば，その情報をみることができる．



[27] 下の図では“naito”に 4 件のデータがマッチしている．LDAP サーバに漢字の情報を入れておけば，漢字でもマッチさせることができる．

LDAP サーバを利用した検索はアドレスブックだけでなく，「メール」アプリケーションのユーザ検索でも利用可能である．

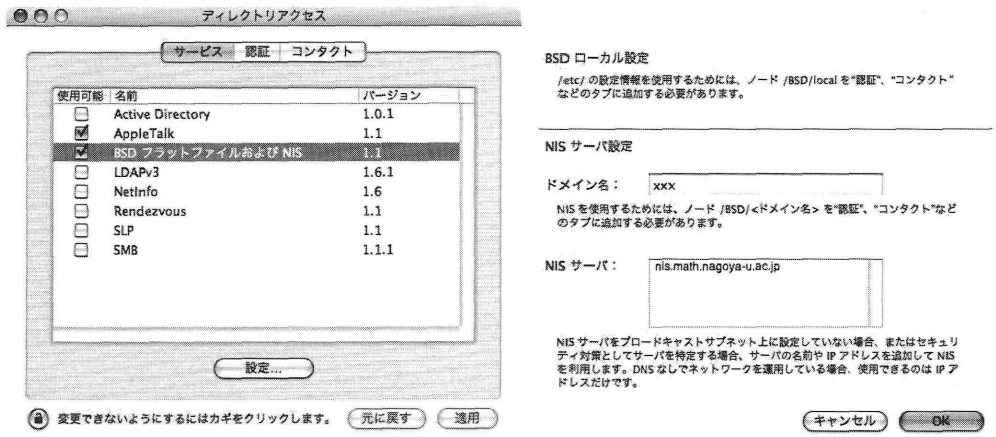
12.3.3 LDAP のセキュリティ

NetInfo の項では，そのセキュリティ上の問題点を列挙したのだが，LDAP にはそのような問題がないことを確認しておこう．NetInfo は rpc を基本としたサービスで，各 NetInfo ドメインのポートが動的に決まってしまうことが大きな問題点であった．しかし，LDAP は 389 番ポートのみを利用[28]するサービスであり，ファイヤーウォール設定との親和性が高い．また，LDAP の機構自身に SSL による暗号化通信機能があり，それを利用することにより，ネットワーク上を流れるユーザのパスワードデータを盗聴などから保護することが可能である．[29]

12.4 NIS とローカルデータベースの利用

とりあえず NIS と BSD ローカルデータベースを参照する方法をメモしておこう．Mac OS X では NIS や BSD ローカルデータベースの利用はあまり推奨されていないようだが，NIS は一時的に LDAP などへの移行の中途段階として必要になる場合があるだろう．

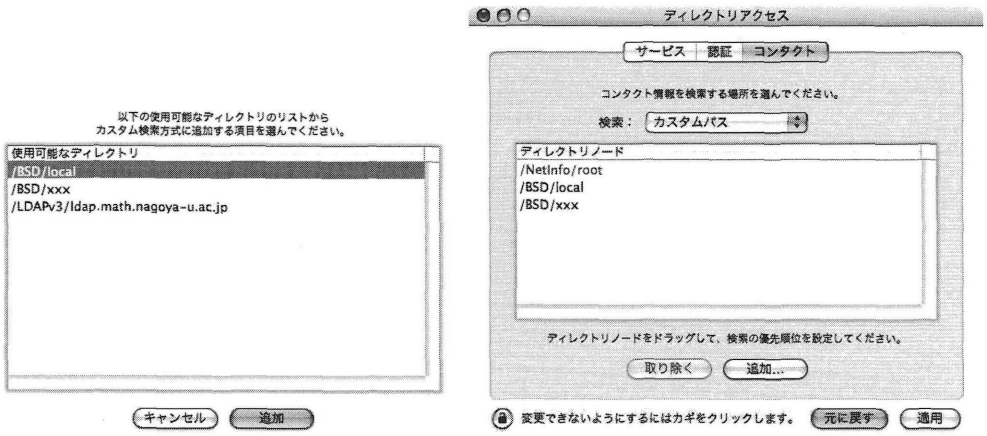
NIS または BSD ローカルデータベースを参照するためには，「ディレクトリアクセス」の設定で「BSD フラットファイル及び NIS」を選択し「設定」を行う．



「設定」をクリックすると，上右図のように「NIS ドメイン名」と「NIS サーバ」名の入力が求められる．もし「BSD フラットファイル」のみを利用する場合にはこれらの入力は必要ない．

この後，「ディレクトリアクセス」の「認証」及び「コンタクト」タブを開くと，`/BSD/local` と `/BSD/xxx` というエントリがみつかる．`/BSD/local` は「BSD フラットファイル」の参照をあらわし，`/BSD/xxx` は「NIS ドメイン」 `xxx` への参照をあらわすので，必要なネーミングサービスを「認証」や「コンタクト」に設定すればよい．

[28] SSL を使う場合には 443 番も利用する．
[29] SSL を利用した LDAP の設定については次回以降の LDAP サーバの設定の時に解説する．



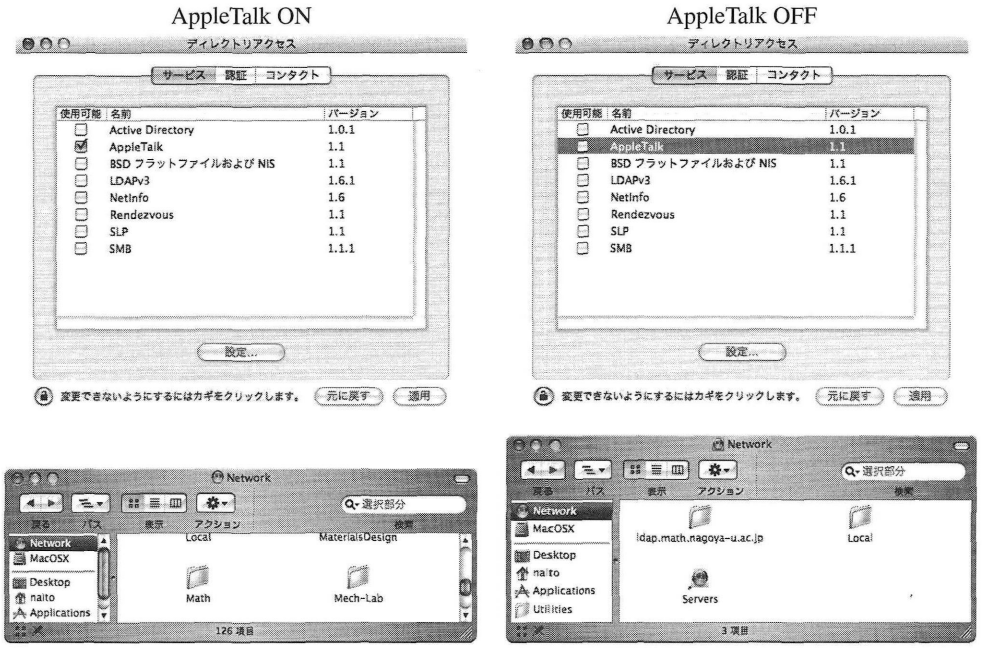
「認証」や「コンタクト」での参照の優先順位は，上右図でのならび順で指定されているので，必要であれば優先順位を変更する．[30]

12.5 AppleTalk

ディレクトリアクセスの項目に「AppleTalk」という項目があり，これが何を意味しているかを最後にメモしておこう．

ファインダの項目の中に「ネットワーク」なる項目がある．これは，ネットワーク上にあるサーバの一覧を示す項目なのだが，ディレクトリアクセスの「AppleTalk」が「ON」になっていて，「ネットワーク環境設定」で AppleTalk が利用できる状態となっていると，上記の「ネットワーク」には AppleTalk ゾーンがフォルダとして表示され，フォルダを開くと各ゾーン内の AppleTalk ファイルサーバが表示される．

[30]"/NetInfo/root" は常に最優先のネーミングサービスであり，これを変更することはできない．



逆に「ネットワーク」を開いても AppleTalk ファイルサーバが表示されないときには，ディレクトリアクセスの「AppleTalk」の項目が「ON」になっているかどうかを調べる必要がある．

12.6　ネットワークを使ったディスクの共有

ここまででみてきたように，NetInfo や LDAP を利用すると複数の Mac OS X のホストでユーザ情報などを共有することが可能となり，研究室などで複数台の Mac OS X のホストを同一の環境で利用できる．しかし，実際にこのことを実現するには，すなわち，複数台ある Mac OS X のホストのどれを使っても同一の環境でユーザが利用できるためには，各ユーザのホームフォルダがネットワークを利用して共有されていなければならない．また，いろいろなフリーソフトウェアなどをすべてのホストにインストールするのではなく，1 台のホストにインストールしておき，そのファイルを共有することによって，ソフトウェアのインストールの手間を省くことができる．[31]

ここでは，ユーザのホームフォルダが格納されたディスクとフリーソフトウェアが格納されたディスクをある 1 台の Mac OS X のホストにおき，そのディスクをネットワークを利用して他の Mac OS X のホストと共有するための設定を考えてみよう．

12.6.1　設定すべき状況

はじめに，どのような状況を実現すべきかをきちんと確認しておこう．

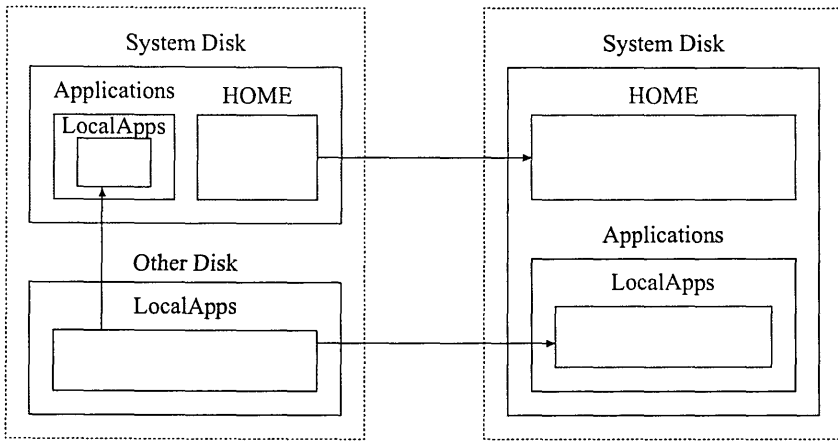
ディスクサーバ　ホスト名は nfsserver とする．

[31] フリーではないソフトウェア，すなわち，ライセンスが必要なソフトウェアに関しては，1 台のホストのみにインストールしてそれを共有することはライセンス違反になる可能性が高い．したがって，ソフトウェアを共有するときにはフリーソフトウェアに限ることが望ましい．

1. Mac OS X のホストで, ユーザのホームフォルダを格納するディスクとフリーソフトウェアを格納したディスクを持つ.
2. NetInfo または LDAP によるネームサービスが利用できる.
3. 上記のディスクは HFS+ フォーマットである.
4. ホームフォルダはディレクトリ "/home" 以下にあり, ユーザ "foo" のホームフォルダは /home/foo である.
5. フリーソフトウェアを格納したディスクは起動ディスクではなく,「ディスク名」software である. さらに,「アプリケーション」フォルダ内の「LocalApps」フォルダとしてその中身が見えているようにする.

クライアント ホストは client1, client2 の 2 台とする.

1. Mac OS X のホストで, ディスクサーバと同一の IP ネットワークに属している.
2. NetInfo または LDAP によるネームサービスが利用できる.
3. サーバ上のホームフォルダを共有し, ホームフォルダはディレクトリ "/home" 以下にあり, ユーザ "foo" のホームフォルダは /home/foo である.
4. サーバ上のフリーソフトウェアを格納したディスクを共有し,「アプリケーション」フォルダ内の「LocalApps」フォルダとしてその中身が見えているようにする.



ここでいくつかの用語の整理と Mac OS X のディスクシステムについて整理をしておこう. UNIX システムでは複数台からなるディスクを一つの論理的なファイルシステム（ディレクトリツリー）として構成している. 通常, 起動ディスクは「ルートディレクトリ」"/" に設定されるが, 起動ディスク以外のディスク（正確にはディスクパーティション）またはネットワーク上にある共有ディスクをルート以下のディレクトリのどこか（例えば /usr/local など）に設定しなければならない. このことを「ディスクをマウント (mount) する」と呼び, マウントするディレクトリの場所を「マウントポイント (mount point)」と呼ぶ. 各ディスクのマウントポイントの情報は, 旧来の UNIX システムでは, /etc/fstab というファイル[32]に記述され, 起動時にマウントされる.[33]

一方, Mac OS X では起動時にすべてのディスクを走査し, 起動ディスクをルートディレクトリに設定した後に, 他のディスクの自動的なマウントを行う.[34] その際のマウントポイントは, ディスク

[32] Solaris 2.x では /etc/vfstab になるなど, システムによってファイルの名称は異なることがある.
[33] ディスクがローカルにあるかネットワーク上にあるかに関わらず /etc/fstab に記述する.
[34] 起動後に IEEE 1394 のディスクを接続したときにも同じ処理が行われる

名 FOO を持つディスクの場合には /Volumes/FOO となる. したがって, 上記の「ディスクサーバ」の 5 の設定を行うには明示的にマウントポイントを指定する必要がある.

また, このようにネットワークを介してディスクの共有を行うことを「ネットワークファイル共有」と呼ぶ. 特に, 今回は「NFS」と呼ばれる[35]方法を用いて共有を行う.[36] その際, ディスクサーバ側では「どのディスク（またはディレクトリ）をどのような条件で, どのホストに共有を許可するか」という設定を行う必要があり, それを「エクスポート」(export) 設定と呼ぶ.

12.6.2 設定項目

この様な状況を設定するための項目は以下のとおりである.

ディスクサーバの設定

1. ディスクのエクスポートの設定を行う.
2. 「ディスクサーバ」の 5 の設定を行う.

ネームサービスの設定

クライアント及びディスクサーバのマウント情報をネームサービスに登録する.

「ディスクサーバの 5 の設定」は, ディスクサーバ上で単独で行うことも可能であるが, ネームサービスを利用して, クライアントと同時に設定することが可能である.

12.6.3 設定方法

以下では上記の設定項目を順を追って調べていこう.

12.6.3.1 ディスクサーバの設定　はじめに, ディスクサーバ上でエクスポート設定を行うことが必要となる. この部分の設定は, 標準的な BSD UNIX と同様の設定を行えばよい. 実際の設定は以下のように行う.

ディレクトリ /home をクライアント client1 と client2 にエクスポートするには, /etc/exports に以下の行を追加する.

```
/home client1 client2
```

また, ディスク FOO もエクスポートする必要があるため, /etc/exports に以下の行を追加する.

```
/Volumes/FOO client1 client2
```

このエクスポート設定を有効にするには mountd プロセスに対して HUP シグナルを送る.[37] もちろん, 再起動を行ってもよいのだが, すでにディスクをマウントしているクライアントホストがある場合にはクライアントホストのディスクのマウントを解除してからでなければ再起動してはいけない.

/etc/exports ファイルにはより詳細なエクスポート設定を記述することができる.

---

[35] 「NFS」は Network File Service または Network File System の略.

[36] Mac OS X の場合, AFS (Apple File Sharing) を用いてマウントする方法もないわけではないが, AFS はユーザ認証を必要とするため, システム全体にわたるネットワークファイル共有には適さない.

[37] 多くの BSD システムではディスクエクスポートデータを更新するには "exportfs -a" コマンドを実行する. Mac OS X には exportfs コマンドが存在しないので, 直接 NFS デーモンにシグナルを送ることになる.

- "-ro" オプションを指定すると, そのディスクは「読み取り専用」でエクスポートされる. すなわち, クライアント側ではディスクの内容に対する変更を加えることができない.
- エクスポート対象のホストは, 明示的なホスト名ではなく, ある範囲の IP アドレスをもつすべてのホストを指定することができる. 例えば "-network 172.16. -mask 255.255.0.0" を指定すると, IP アドレスが 172.16.0.0/16 に属するホストからのアクセスを許可することとなる.
- Mac OS X を含む BSD システムに特有なオプションが "-maproot" 及び "-mapall" である. 通常 NFS の書き込み及び読み出し権限は, 「ユーザ ID」 (UID) (ユーザ名ではなく, ユーザをあらわす数値) を使って判断される. 例えば, UID=501 を持つユーザがサーバ側ではユーザ名 foo, クライアント側では bar であったとしても, クライアント側でユーザ bar がファイルを作成した場合, サーバ側では同一の UID を持つ foo が書き込みを行ったと判断する. したがって NFS を利用する際には, サーバ側とクライアント側とで UID をそろえておく必要がある. BSD のオプション -maproot と -mapall は, この対応づけの一部分を変更するオプションであり, "-maproot=bar" と指定すると, クライアント側からのルートユーザのアクセスはサーバ側では bar のアクセスとみなされる. "-mapall=foo" と指定すると, クライアント側からのすべてのアクセスはサーバ側ではユーザ foo のアクセスとみなされる.[38]
- 通常はエクスポート対象のディレクトリはクライアント側からマウントするときには, そのサブディレクトリをマウントすることはできない. すなわち /etc/exports に /home と書いてあると, クライアントは /home/foo をマウントすることはできない. しかし "-alldirs" オプションを指定するとすべてのサブディレクトリをマウント対象とすることを許可する.

これらのオプションを利用したアクセス制御の例には以下のようなものがある. (すべて /etc/exports に書き込むエクスポート設定の例である)

- ディレクトリ /home へのアクセスを, client1 からは読み書き可能, その他の IP アドレス 172.16.xxx.yyy を持つホストからは読み出し専用でアクセスさせる.

```
/home client1
/home -ro -network 172.16 -mask 255.255.0.0
```

- ディレクトリ /shared へのクライアント client からのアクセスをユーザ foo としてアクセスさせる.

```
/shared -mapall=foo client
```

この設定は, 複数のユーザがあるフォルダ内のデータを共有しているときに有効にはたらく. 通常の「共有フォルダ」の設定では, 共有フォルダ内のファイルは作成者のみが書き換え可能となり, 他のユーザが作成したファイルを変更することはできない. しかし, 適当なフォルダ (上の例では /shared フォルダ) の所有者を foo にしておき, 共有したいファイルはそのフォルダ内に置く約束をしておけば, クライアント client からのアクセスであれば, クライアント側のユーザが誰であっても, サーバ側ではユーザ foo のアクセスと見なされるため, そのファイルは client のすべてのユーザから読み書き可能となる.

以上をまとめると, /etc/exports に

```
/home client1 client2
/Volumes/FOO -ro client1 client2
```

[38]ただし, これらのオプションを指定した場合の実際の書き込み動作は, クライアント側からの UID で書き込みを行った後に所有権の変更をするようなので, ディレクトリのアクセス権の設定には注意が必要である

と記述すれば, /home と FOO ディスクを共有し, FOO ディスクはクライアントからは読み出し専用となる.

なお, このエクスポート設定は, 本来なら NetInfo local ドメインのデータベースを用いて設定すべきものである. しかし, NetInfo データベースにエクスポート設定を記述するのは非常に面倒で, 間違えやすい操作をしなければならないため, 今回は /etc/exports に記述する方法を利用した. Mac OS X 10.3 では, ネーミングサービスに /BSD/local を指定しない状態でも, エクスポートデータベースだけは /etc/exports を読み出してくれる.

12.6.3.2　ネームサービスの設定　共有ディスクを実現するにはディスクサーバ上でエクスポートされたディスクをクライアント側でマウントする必要があるが, クライアントホストで共有すべきディスクの情報を得るために, Mac OS X ではネームサービスを利用してその情報を得る. ここでは NetInfo または LDAP を利用して共有ディスクのマウント情報を得るための NetInfo 及び LDAP の設定方法を調べよう. 現実にはクライアントで利用できる NetInfo または LDAP のいずれか一方の設定を行えばよい.

また, ネームサービスで流すべきディスクの情報は以下のとおりである.

ディスクサーバ名 どのホストからディスクがエクスポートされているか.
エクスポートディレクトリ マウントしたいサーバ上のディレクトリ名称.
マウントポイント クライアント上のどのディレクトリにマウントするか.
マウント方法 どのような方法（プロトコル）でマウントするか.（今回は NFS）
マウントオプション 読み出し専用または読み書き専用, ディスククオータの設定などのオプションを指定.

上記のディスクの情報は以下のようなものである.

/home

ディスクサーバ名 nfserver
エクスポートディレクトリ /home
マウントポイント /home
マウント方法 nfs
マウントオプション rw

/Volumes/FOO

ディスクサーバ名 nfserver
エクスポートディレクトリ /Volumes/FOO
マウントポイント /Applications/LocalApps
マウント方法 nfs
マウントオプション ro

12.6.3.2.1　NetInfo での設定　ここで利用する NetInfo ドメインは, 各クライアントの「Local ドメイン」ではなく, すべてのクライアント（及びディスクサーバ）で検索可能な「Local ドメイン」の上位層で設定することが望ましい. なぜなら「Local ドメイン」の設定をしてしまうと, すべてのクライアントの「Local ドメイン」に同じ設定を行わなければならず, 非常に面倒である. また, ディスク共有を行う前提としてユーザ情報の共有が行われているので, ユーザ情報が格納された NetInfo ドメインが存在するはずであり, そのドメインにディスク共有の設定を記述すればよい.

実際にNetInfoドメイン「top」に入れる情報は以下のようにして作成する.（ここで,NetInfoドメイン「top」はディスクサーバ上にあると仮定している）

fstabデータを作成する ディスクサーバ上で,/tmp/fstabとして,以下の内容のファイルを作成する.

```
nfsserver:/home          - /home                              nfs - yes rw
nfsserver:/Volumes/FOO   - /Applications/LocalSoftwares  nfs - yes ro
```

NetInfoにデータを入れる /tmp/fstabに記述した情報をNetInfoドメイン「localhost/top」に格納する.そのためには以下のコマンドを入力すればよい.

```
% niload -m fstab -t localhost/top < /tmp/fstab
```

NetInfoのデータの確認 その結果をniutilコマンドで調べてみると,

```
% niutil -list -t localhost/top /mounts
5 nfsserver:/home
6 nfsserver:/Volumes/FOO
% niutil -read -t localhost/top 5
dir:  /home
dump_freq:  0
name:  nfsserver:/home
opts:  rw
passno:  0
vfstype:  nfs
% niutil -read -t localhost/top 6fg
dir:  /Applications/LocalApps
dump_freq:  0
name:  nfsserver:/Volumes/FOO
opts:  ro
passno:  0
vfstype:  nfs
```

と出力される.

クライアント側で検索可能なNetInfoドメインに上記の情報が格納されていれば,起動時に自動的にマウントされる.

12.6.3.2.2 LDAPでの設定 LDAPデータベースでのディスクのマウント情報はRFC 2307に規定され,

```
ou=mounts,dc=....
```

というDNを持つエントリを作成すればよい.

実際には以下のようなLDIFデータをLDAPデータベースに設定すればよい.（ここでは「ベースDN」はdc=math,dc=nagoya-u,dc=ac,dc=jpとなっているが,この部分は各LDAPサーバで適切に設定しなければならない）

```
dn: ou=mounts,dc=math,dc=nagoya-u,dc=ac,dc=jp
ou: mounts
objectClass: top
objectClass: organizationalUnit

dn: cn=nfsserver:/home,ou=mounts,dc=math,dc=nagoya-u,dc=ac,dc=jp
mountOption: rw
mountType: nfs
cn: rabbit:/home
objectClass: mount
objectClass: top
mountDumpFrequency: 0
mountDirectory: /home
mountHost: nfsserver

dn: cn=nfsserver:/Volumes/FOO,ou=mounts,dc=math,dc=nagoya-u,dc=ac,dc=jp
mountOption: rw
mountType: nfs
cn: rabbit:/home
objectClass: mount
objectClass: top
mountDumpFrequency: 0
mountDirectory: /Applications/LocalSoftwares
mountHost: nfsserver
```

LDAP サーバに上記の情報が格納されていれば, クライアント側では「コンタクト」にその LDAP サーバが指定されているだけで, 起動時に自動的にマウントされる.

12.6.3.3　ディスクサーバ上でのディスクのマウント　ディスクサーバ自身が上記の NetInfo または LDAP データベースにアクセス可能であれば, `/Volumes/FOO` に（実体の）あるデータが `/Applications/LocalApps` にマウントされ,「ディスクサーバの 5 の設定」の設定が実現できる. この場合, 設定上はローカルなディスクとしてではなく NFS を経由しているが, 実際にはローカルなアクセスが行われる.

12.6.3.4　NFS の注意事項　Mac OS X で NFS を利用するにあたっては, 以下のような注意点がある.

- NFS を経由して Microsoft Office の文書か Excel のファイルを差し込みデータとして読み込もうとするときに,「データファイルが見つからない」というエラーが発生することがある.
- Mac OS X での標準的なディスクシステムのフォーマットは HFS+ である. 前回にも解説したとおり, HFS+ の特徴は「カタログ B ツリー」内に「リソースデータ」を持つことである. しかし, NFS を経由してしまうと「カタログ B ツリー」にはアクセスすることができない. そのため, Mac OS X では次のような方法で「分離リソース」を実現している.
  ファイル `foo` のリソースデータは, 通常は UNIX の `ls` コマンドではみることができない. しかし, NFS （UFS を用いても同じだが）を利用する場合には `._foo` というファイルがつくられる. この `._foo` が `foo` のリソースデータである. したがって, `._foo` などというファイルが見つかったとき `rm` コマンド等でこれを消去してはいけない.

なお,「開発環境」に含まれる `/Developer/Tools/SplitForks` コマンドを利用することにより, カタログ B ツリーに含まれるリソースデータを上記のように単独のファイルとして取り出すことができ, `/System/Library/CoreServices/FixupResourceForks` コマンドを利用すれ

ば，単独のファイルとして取り出されたリソースフォークを元のようにカタログBツリー内のデータに戻すことができる.[39]

13 インターネット接続

ここでは，「インターネット接続」アプリケーションで提供されている，「802.1x」と「VPN」について解説してみよう．これらは，ここまでの解説とは異なり，Mac OS Xのホスト（特にノート型のもの）を「安全に」ネットワークに接続する手段を提供する.

13.1 ネットワークのセキュリティ

はじめに，この章での設定の目的である，ネットワークのセキュリティとは何かを考えてみよう．通常，「ネットワークのセキュリティを保つ」という表現をした場合には，以下の項目がその目的となる.

- ネットワーク上を流れるデータを「盗聴」から守る.
- 許可されないユーザが勝手にネットワークを利用することを禁止する.

現在「ローカルエリアネットワーク」(LAN) で用いられているネットワーク (Ethernet) を使った通信では，ホスト間の通信は必ずしもそのホストの間でやり取りされるわけではなく，LAN内のすべてのホストにデータ（パケット）が到達し，その宛先が自分自身でないパケットは破棄する仕組みを使っている.[40] したがって，LAN上を流れるパケットは容易に盗聴[41]することが可能であり，LAN内に悪意のあるホストが存在した場合には，各人のパスワードなどが含まれたパケットが盗聴され，パスワードが流出することが容易に想像できる.

近年のネットワークの利用形態では，インターネット[42]を利用して，学外から学内へログインしたり，各種のサイトでクレジットカード番号などを入力する機会が増えている．このような場合にもインターネット上でパケットの盗聴が行われていない保証はなく，何らかの形でパケットを盗聴から防ぐ方策が必要となり，現在，盗聴を防ぐ最も有効な方法は「パケットの暗号化」と考えられている．パケットを暗号化して通信する手段として，UNIXホストへログインするためのssh，ウェブサーバにアクセスする際に，アプリケーションレイヤで暗号化を提供するSSLを利用したhttpsなどが広く用いられている．これらのプロトコルは，特定の通信手段の暗号化通信であるのに対して，この後に解説するVPNは通信手段を選ばず，[43]すべてのパケットを暗号化する方法である.

一方，後者の「許可されないユーザがネットワークを利用する」とは何が問題となるのだろうか．通常ネットワークを利用するためには各ホストにIPアドレスを割り当てる必要がある．よく知られたとおり，IPアドレスは世界中で一意なアドレス[44]を割り当てる必要があり，その設定も初心者にとっては必ずしも容易なことではない．そのため，近年ではDHCPと呼ばれる方法[45]を用いて，ネットワークに接続されたホストに対して自動的にIPアドレスを割り当てる仕組みが確立している.

[39]ただし，`FixupResourceForks` はファイルに対するコマンドではなく，ディレクトリに対するコマンドであることに注意が必要である.

[40]名古屋大学のネットワーク (NICE) では，各建物ごとにLANが構成されている．また，ネットワーク機器には「レイヤ2スイッチ」と呼ばれる機器を利用しているため，必ずしもLAN内のすべてのホストにパケットが到達するわけではない.

[41]必ずしも自分宛ではないパケットを保存し，その中身を解析することを指す.

[42]ここで「インターネット」と言った場合には，学内と学外を結ぶ「公衆網」のことである.

[43]より低いレイヤで暗号化を実現する手段という意味.

[44]近年，IPアドレスの枯渇が問題となり，「プライベートIPアドレス」を用いたLANも多くなっている.

[45]本来DHCPは，ユーザに対する負担の軽減ではなく，ノートPCを中心とするモバイル機器をネットワークに接続するための方法として開発されたものである.

逆に言えば, DHCP を用いたネットワークではノート PC をネットワークにつないでしまえば, 自動的に IP アドレスが割り当てられ, 誰もが自由にネットワークを利用することが可能である. 仮に, そのようなユーザの中に悪意を持ったユーザがいて, 前述のパケットの盗聴を行ったり, ネットワーク上のホストにクラッキングを行ったりする可能性も多い. そのため, DHCP ではあらかじめ登録された MAC アドレス[46]を持つ機器にのみ IP アドレスを割り当てる機能が用意されている. しかしながら, MAC アドレスも容易に詐称が可能であるばかりか, パケットの盗聴を行うだけであれば, IP アドレスを割り当てる必要さえない. このような状況で, ネットワークの不正な利用を防ぐ手段の必要性が重要視されている.

13.2　802.1x 認証を用いた無線 LAN

13.2.1　無線 LAN の脆弱性

数年前からノート型コンピュータをネットワークに接続する手段として, 無線 LAN を利用できるようになった.[47] 従来のネットワークでは（前述の）パケット盗聴や不正な利用を行おうとした場合には, 何らかの形で「ケーブルを接続する」ことが必要となり, 具体的にはネットワークの建物に入り込まれない限りはネットワークのセキュリティは保たれていた. しかし, 無線 LAN は電波の届く限りネットワークが広がっていると考えられるため, 建物外からもネットワークに接続できたり, セキュリティを脅かす行為を行うことができる. そこで近年は「無線 LAN のセキュリティ」問題が新聞紙上を賑わすことも多くなった.

そのため, これまでは無線 LAN の通信のセキュリティを保持するために以下のような方策をとることが多かった.[48]

- 無線 LAN ネットワーク (SSID) を「非公開」にする.
- 無線 LAN アクセスポイントに対するアクセスを MAC アドレスを用いてフィルタリングする.
- 無線 LAN 通信を WEP (Wired Equivalent Privacy) による暗号化を行う.

しかし, これらの方策もつぎのように問題点があることが知られている.

非公開ネットワーク　アクセスポイントの機器によっては,「ビーコン」と呼ばれるアクセスポイントから発信される探索用のパケットに（非公開ネットワークであっても）ネットワーク名が含まれるものがある.

MAC フィルタリング　ネットワークインターフェースカード (NIC) の MAC アドレスは, ソフトウェア的に改竄（変更）が可能である.

WEP 暗号化　これが最も問題がある内容で, 現在利用されている 40 ビットまたは 106 ビット WEP 暗号化は, そのアクセスポイントを利用するユーザ全員で共通のものを利用し, 暗号化手法が易しいものを利用しているため, 大量の通信パケットを傍受すれば暗号化鍵を推測可能となる. また, 40 ビット暗号化では, すべての鍵パターンを生成して「総当たり」的に暗号化鍵を発見することも容易である.[49]

このように, 標準的な無線 LAN 通信ではセキュリティ上多くの問題点があることがわかる.

[46] 各機器に（より正確にはネットワークインターフェースに）割り当てられたハードウェアのアドレス.
[47] もちろん, 無線 LAN はデスクトップ PC でも利用できるが,「モバイル」という視点からはノート PC での利用が本命だろう.
[48] 以下の設定を行っていない無線 LAN ネットワークは,「誰でも自由に使っていいよ」と言っているようなものである.
[49] これ以上に, ユーザ全員で同じ鍵を利用するため, 鍵が流失することも多い また, 106 ビット暗号化であっても, 鍵を推測することができると言われている

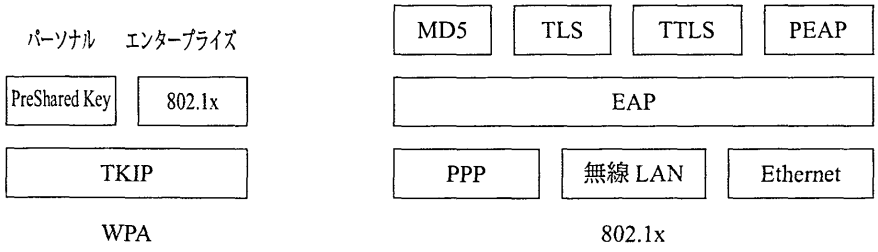
13.2.2 無線LANの新しいセキュリティ規格と802.1x

上に述べたような無線LANの脆弱性を克服するため策定されたものが「802.11i」と呼ばれる無線LANのセキュリティ規格[50]である．ここでは，802.11iの規格の一つであり，最近多くの無線LANアクセスポイントで採用されているWPA (Wireless Protected Access) の仕組みについて簡単に解説しよう．

WPAの柱は「認証」と「暗号化」の2点であり，何らかの認証をパスしたユーザにのみ，アクセスポイント経由のアクセスを許可し，暗号化通信を行うというものである．この場合の暗号化はTKIP (Temporal Key Integrity Protocol) と呼ばれるもので，アクセスする機器ごとに異なる暗号化鍵を用い，さらに一定時間ごとに異なった鍵に交換するもので，前述のWEPと比較して，セキュリティの強度は格段に上がっている．

WPAは認証方法によって2種類に分類され，一つは，アクセスポイントと無線LANクライアントの間で「共通鍵」を用いる「WPAパーソナルモード」と，[51] 802.1x認証を用いる「WPAエンタープライズモード」がある．

802.1x認証とは，ネットワークに接続する際にユーザ認証を行い，認証をパスしたユーザにのみネットワークの利用を許可する手順を与えた規格である．[52] 802.1xは，この章の冒頭で述べた「許可されていないユーザの不正なネットワークの利用」を防ぐためのものであり，「有線LAN」に対しても意味のあるものである．[53] 802.1xの認証の方法はEAP (Extensible Authentication Protocol) と呼ばれる以下の手順にしたがう．802.1xクライアントから認証要求が発生した場合，802.1x認証を用いるアクセスポイント[54]は「認証サーバ」[55]に認証を求め，それにパスした場合にのみアクセスポイントの利用を許可する．EAPを用いる際にはMD5 Digest認証，TLS認証，TTLS認証，PEAP認証などのいくつかの認証手順を用いることができる．



なお，これらの認証手順には以下のような長所と欠点がある．

MD5 digest認証 MD5という「ダイジェスト文字列」生成アルゴリズムによって，「チャレンジ＆レスポンス」による認証を行う．

長所 通常の認証のように「パスワード」以外の設定が必要ない．

欠点 認証サーバ上では「パスワード」を「平文」で格納する必要がある．また，認証パケットを大量に盗聴されるとパスワードが推測される可能性がある．

[50]無線LANの規格は，802.11b，802.11gのように802.11ではじまる．これはIEEEの802.11委員会によって規定された規格であることを示している．

[51] 「事前共有鍵」(Preshared Key) 方式とも呼ばれる．要するに共通の「パスワード」を用いるもので，多数のユーザがいる環境では適切な方法とは言い難い．

[52]このような状況を理解するには，自宅でADSLなどを利用している状況を思い出せばよい．ADSLを利用する際には，ADSLのユーザ名とパスワードを入力しなければネットワークへの接続はできない．

[53]実際「有線LAN」の場合には，ネットワーク認証で認証を得る前の段階では，ネットワークスイッチのポートが"disable"の状態となり，パケットの盗聴さえも不可能となる．

[54]無線LANの場合には「無線LANアクセスポイント」であり，「有線LAN」の場合には「ネットワークスイッチ」となる．

[55]通常"radius"と呼ばれるソフトウェアを用いる．

TLS 認証 電子証明書を用いた認証方法. クライアントは, 認証サーバの電子証明書によって署名されたクライアント証明書をサーバに提示することにより認証する.

長所 公開鍵暗号を利用しているため, 安全性が高い.
欠点 クライアントごとに電子証明書を発行する必要がある.

TTLS 認証及び PEAP 認証 電子証明書とパスワードを用いた認証方法. クライアントは認証サーバの電子証明書を持ち, 認証サーバはクライアントがもつ電子証明書と各ユーザのパスワードで認証を行う.

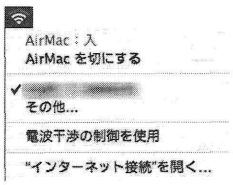
長所 公開鍵暗号を利用しているため, 安全性が高い. クライアントごとに電子証明書を発行する必要がない.
欠点 対応する radius サーバが非常に少ない.

なお, ここで出てきた「電子証明書」に関しては詳しくは解説しないが, ウェブサーバへのセキュアなアクセスである https で利用されている方法とほぼ同じものである.[56]

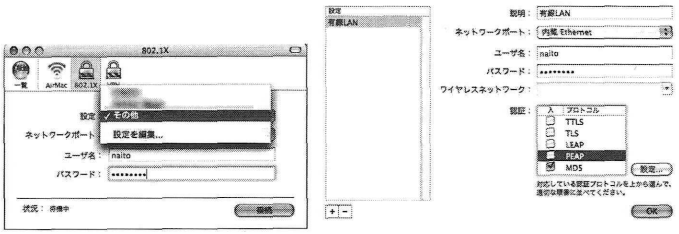
13.2.3 802.1x によるネットワークへの接続方法

ここから Mac OS X を用いて 802.1x ネットワーク認証が必要なネットワークへの接続方法を解説していこう. はじめに, (あまり実用的ではないかもしれないが)「有線 LAN」に 802.1x が必要な場合の設定を解説して, 802.1x の本質的な部分をみていく. その後に, 無線 LAN (特に, アクセスポイントが Apple 社の AirMac Extream BaseStation の場合) の 802.1x による接続方法を解説する. なお, この解説では 802.1x の認証手段として,「有線 LAN」の場合は MD5 Digest 認証, 無線 LAN の場合は TLS 認証を使うこととしよう.

13.2.3.1 有線 LAN の場合 最初に「インターネット接続」を開く. 無線 LAN が使えるときにはメニュー上の無線 LAN のアイコンから「インターネット接続を開く」でも開くことができる.



インターネット接続を開いた後, 上部のリストから 802.1X を選択する. さらに「設定」のプルダウンメニューを開き「その他」を選択する (下左図) と, 下右図のようなウインドウが開く.

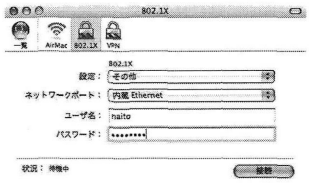


[56] ウェブブラウザには「ルート証明書」と呼ばれる電子証明書があらかじめ格納されており, ウェブサーバから提示された「サーバ証明書」の正当性が, 格納されているルート証明書により確認できた場合にのみ暗号化通信が確立する.

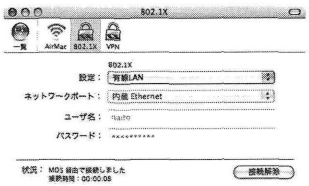
ここでつぎのような設定を行う.

ネットワークポート 「内蔵 Ethernet」を選択.
ユーザ名 ネットワーク認証のユーザ名.
パスワード MD5 認証のパスワード.
認証 MD5 のみ「入」.

これらを設定して「OK」をクリックすると, つぎのように設定される.



ここで「接続」をクリックすると認証が始まり, 正常に接続できれば



のようにウインドウ下部に「MD5 経由で接続しました」とのメッセージが表示される. もし, エラーが発生した場合には「ユーザ名」または「パスワード」が間違っている可能性があるので, それを修正すればよい.

このように「有線 LAN」で MD5 digest 認証を利用する場合には, radius サーバの設定さえ間違っていなければ, 極めて容易にネットワーク認証を行って接続することができる.

なお「有線 LAN」でも, 以下の「無線 LAN」と同様に TLS 認証を行うこともできる.

13.2.3.2 無線 LAN の場合 無線 LAN で 802.1x 認証を用いて接続する場合は, MD5 digest 認証が使えないため, 「有線 LAN」のようには簡単にはいかない.[57] 今回 Apple AirPort Extream BaseStation では TLS 認証が利用できるので, TLS 認証を使って接続することを考える.[58]

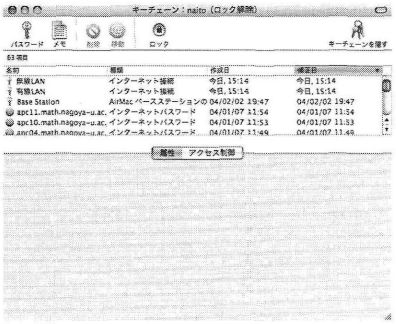
13.2.3.2.1 TLS 認証に必要なもの TLS 認証を行うためには, 事前に以下のものを入手しておく必要がある.

サーバ証明書 認証サーバ（この場合は radius サーバ）が正当なものであることを示す電子証明書. 以下のルート証明書によって署名されている必要がある.（最初は必要ない）
ルート証明書 サーバ証明書の署名の根拠を証明する電子証明書.
クライアント証明書 アクセスするユーザが正当であることを示す電子証明書. サーバ証明書により署名されていることが必要であり, 内部に「ユーザ名」と「パスワード（秘密鍵）」を含む.

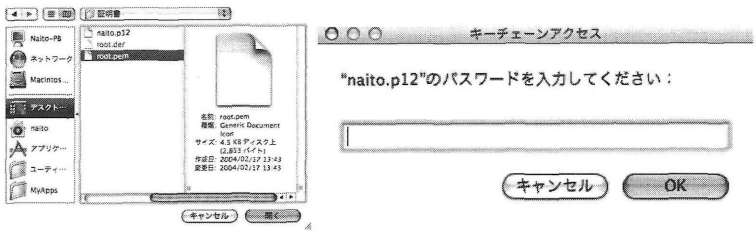
[57]MD5 digest 認証ではパスワードから生成されたダイジェスト文字列が平文のまま流れるため, 無線 LAN ではパスワードが流出する可能性がある. そのため, 無線 LAN アクセスポイントへのアクセスに MD5 digest 認証を用いることはできない. 実際 Apple AirMac Extream BaseStation の場合に MD5 digest 認証を用いると, 認証そのものはパスできるのだが, TKIP 暗号化の段階で鍵交換ができず, 実際の通信を確立することができない.
[58]TTLS, PEAP 認証は radius サーバとして freeradius を使う限りはうまくいかなかった.

TLS 認証の基本的な仕組みは https と同一であるが, 802.1x 認証クライアント（今回の場合は「インターネット接続」）があらかじめルート証明書を持っていないことと, 認証を行うためにクライアント証明書が必要となることの２点が異なっている. そのため, 無線 LAN アクセスポイントの管理者からルート証明書とクライアント証明書を入手する必要がある.

13.2.3.2.2　電子証明書のインストール　入手したルート証明書とクライアント証明書を Mac OS X の「キーチェーン」に登録する. そのためには,「ユーティリティ」フォルダにある「 キーチェーンアクセス」を開く. 最初キーチェーンは以下のようになっている.

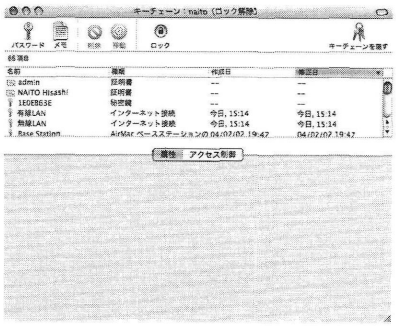


ここで「ファイル」メニューから「読み込み」を選択し, ルート証明書とクライアント証明書をキーチェーンに登録する. ルート証明書は拡張子 “pem” となっているものを利用する.[59] また, クライアント証明書は拡張子 “p12” となっているものを選択する.



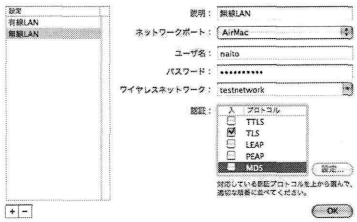
ここでクライアント証明書を読むときに, 上右図のように「パスワード」の入力が求められる. ここで入力するパスワードはクライアント証明書の秘密鍵であり, クライアント証明書を作成するときに入力を求められるパスワードのことである.

これらの読み込みがおわると, 以下のようにキーチェーンにルート証明書とクライアント証明書がインストールされたことがわかる.



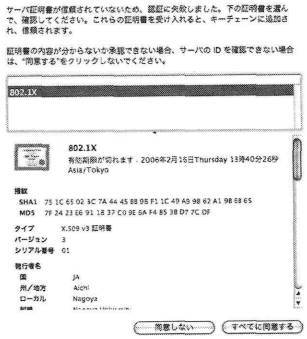
[59] Windows XP の場合は “der” を利用するらしい.

13.2.3.2.3 インターネット接続の設定 ここまで終わるとあとは「有線 LAN」の場合とほぼ同様な手順で設定すればよい.「有線 LAN」の時との違いは以下のものである.

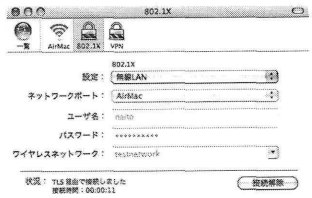


ネットワークポート （当然）"AirMac" を選択する.
ワイヤレスネットワーク ここには接続すべき無線 LAN ネットワークのネットワーク名を記入する.
認証 TLS のみを「入」にする.

この設定が終了したら「OK」をクリックし, 実際に接続を行ってみる. すると, 1 回目の接続に限り以下のようなウインドウが開く.



これはサーバ証明書を信頼するかどうかを聞かれているのであり,「全てに同意する」をクリックすると,



のように TLS を利用した無線 LAN 通信が確立する.

13.2.4 802.1x に必要な機材

802.1x ネットワーク認証は,（無線 LAN の場合）無線 LAN アクセスポイント自身が認証を行うのではなく, radius という認証サーバが動作しているホストに問い合わせを行う. そのため, 少なくとも EAP に対応している radius サーバが動作していなくてはならない.[60]

[60]筆者の属する研究科では, Solaris 9 (SPARC) 上の freeradius を利用している.

また, アクセスポイントが 802.1x に対応している必要がある. すなわち, アクセスポイントがネットワーククライアントからのアクセスを受けたとき, radius サーバに対して EAP にしたがう認証を要求する必要がある.[61] Apple 社の AirMac Extream BaseStation は, 最新版[62]のファームウェアを利用すれば 802.1x に対応する. また, 国内各社からも 802.1x に対応した無線 LAN アクセスポイントが販売されている.

最後に, クライアントソフトウェアであるが, Mac OS X 10.3 で AirMac Extream Card を使う場合は AirMac ソフトウェア が 3.2 以降, AirMac Card を使う場合は AirMac ソフトウェアが 3.3 以降ならば TLS を利用した認証が可能となる. また, Windows XP では標準的に TLS 認証が可能である.

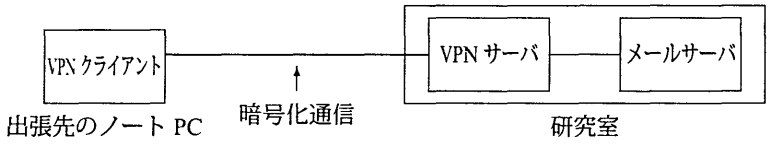
13.3 VPN の利用

13.3.1 VPN とは

VPN (Virtual Private Network) とは, インターネット（公衆回線）上に**暗号化された仮想的な専用回線を構築し**, インターネットを利用して安全な通信を行う手段のことである. VPN を用いることにより, この章の冒頭で述べた, 「インターネット上を流れるパケットを盗聴から保護する」ことが可能となり, 安全に LAN へ接続が可能となる. 具体的には VPN を有効に利用できる状況として, つぎのようなことを考えればよい.

離れた 2 ヶ所（以上）に拠点を持つ事業所の間でネットワーク通信をしようとした場合, その通信の安全性を保持しようとすると, 以前であれば高価な専用回線を利用する必要があった. しかし, 現在はインターネット（公衆回線）を利用して安価に拠点間を接続することができるが, そのままでは通信の安全性を保つことができない. そのため, 拠点間の通信を暗号化して仮想的には内部ネットワークと同じように安全性を保つ手段が必要となり, それは VPN を利用して実現可能である.

VPN は 2 つのネットワークを接続するだけでなく, 1 台の PC と LAN の間の暗号化通信も提供している. 我々はこれを利用して, LAN の外部から安全な通信を行うことができる. より具体的には, 出張先の大学やホテルから研究室のネットワークに接続したと考えた場合, 研究室のネットワークが外部からのアクセスを拒否するような設定になっている状況を考えよう. この場合, 出張先の大学やホテルで接続された PC は, あきらかに外部のネットワーク機器であるため, 研究室のネットワークに入ることはできない. しかし, 研究室に VPN サーバがあり, **ユーザ認証を経た上で** VPN サーバ経由でアクセスするのであれば, インターネット上の通信は暗号化され, ユーザ認証も通過しているので, そのようなホストからのアクセスは研究室内部のホストからのアクセスと同じにみなしてよい.



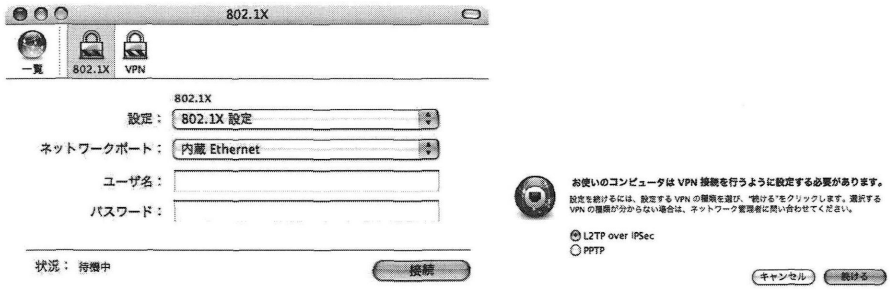
このように研究室などのネットワークが「閉鎖的」な状況になっている場合においても, VPN サーバを適切に運営できれば, 研究室のメンバーは外部から安全にアクセスが可能となる. 以下で Mac OS X での VPN クライアントの利用方法を解説するが, ここではすでに研究室などに VPN サーバが設置されていることを前提とする.[63]

[61] 筆者は CISCO 社のレイヤ 3 スイッチ Catalyst 3550 を利用して実験した.
[62] Version 3.2 以降.
[63] VPN アクセスに必要となる VPN サーバの設定に関しては次回以降に解説することにする.

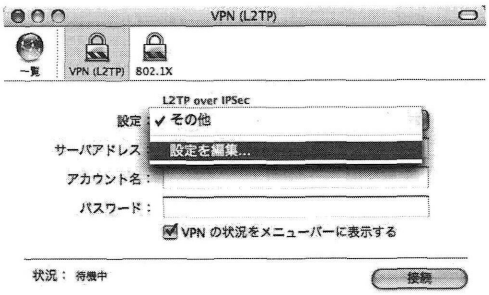
13.3.2 Mac OS X での VPN の利用法

以下では Mac OS X に標準的に含まれている「インターネット接続」の VPN 機能を使って, VPN サーバにアクセスする方法を解説しよう. その際, VPN サーバの管理者から入手しておかなければならない情報は,「サーバ名」と「共有シークレット」の２つである.[64]

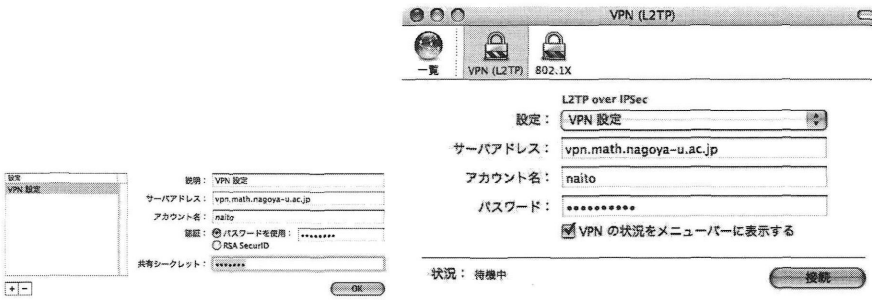
802.1x の時と同様に「インターネット接続」を開き,「VPN 接続」を選択する. すると, 下図のように "L2TP over IPSec" または "PPTP" の選択ウインドウが開く.[65]



ここでは "L2TP over IPSec" を選択し,「設定」メニューから「設定を編集」を選択する.



そこで開いたウインドウ（下左図）で「サーバアドレス」,「アカウント名」,「認証形式」,「共有シークレット」を入力し「OK」をクリックする. すると下右図のように設定が完了するので,「接続」をクリックすれば VPN (L2TP over IPSec) でサーバに接続し, 研究室などの内部ネットワークに接続が完了する.



この時, VPN サーバが通常の設定ならば, VPN は以下のような通信形態[66]をとる.

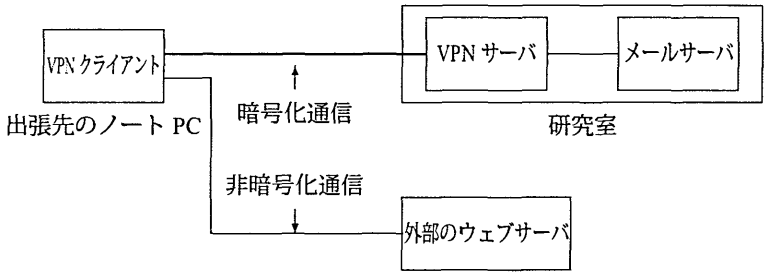
[64]通常「アカウント名」は各ユーザのユーザ ID なのだが,「パスワード」は通常と異なったものを利用する可能性もある.
[65]これらの意味は後述する.
[66]「スプリットトンネリング」と呼ばれる.

DNS 設定 VPN 接続を行っている状態では, DNS は VPN サーバから渡されるサーバを利用する. つまり, 研究室などの内部ネットワークの DNS サーバが利用される.

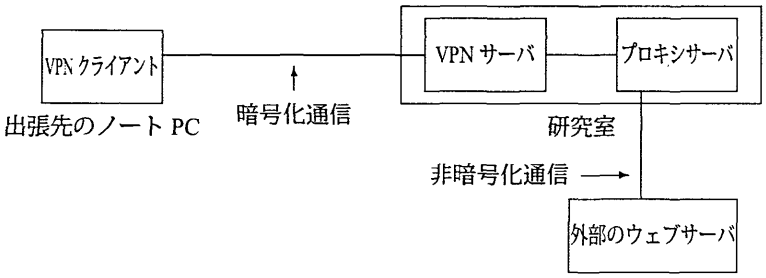
研究室などの内部ネットワークへのアクセス インターネット上を暗号化された通信で, 研究室の内部ネットワークへアクセス可能.

それ以外のホストへのアクセス VPN サーバから "Private" と指定されたネットワーク[67]以外へのアクセスは, 内部ネットワークを経由せず, **暗号化されない**状態で直接通信を行う.

したがって,「本当の意味」で内部ネットワークのホストとは異なるのだが, 内部ネットワークへのアクセスは公衆回線上（VPN クライアントと VPN サーバ間）は暗号化され, 内部ネットワークのホストにとっては, VPN サーバ上で割り当てられた内部アドレスを持つホストからのアクセスと見なされる. そのため VPN を経由したアクセスは, 内部ネットワークからは内部のホストからのアクセスとみなされ, POP サービスや内部に限定されたウェブサーバなどへアクセス可能となる.



しかし, よく誤解を招くのはつぎのような状況である. 電子ジャーナルなどアクセス元のアドレスによってアクセスが制限される外部ホストへのアクセスは, 内部ホストからのアクセスとは異なる. つまり, 外部ネットワークへのアクセスは VPN を経由せず直接行われる. この問題は研究室内部にプロキシサーバをおき, ウェブのアクセスを内部のプロキシサーバ経由で行えば, 外部ウェブサーバへのアクセスもプロキシ経由（内部ネットワーク経由）とすることができる.[68] [69]



[67] 上記の「内部ネットワーク」のこと.

[68] 電子ジャーナルなどの一部には, プロキシサーバ経由のアクセスを禁止（利用条件違反）としている場合がある. したがって, 電子ジャーナルなどへのアクセスの場合, プロキシサーバ経由でのアクセスが許可されているかどうかを事前に調べておく必要がある.

しかし, VPN 経由以外のアクセスを禁止したプロキシサーバを利用するのであれば, 公開プロキシサーバを利用しているのではないため, 上記の問題はクリアできる可能性がある.

【注】「http プロキシ」は http プロトコルのみを対象としていて, https プロトコルは対象とされていない. そのため, http プロキシだけでは, VPN を用いて https を用いるサイトにプロキシサーバ経由でアクセスすることはできない.

[69] お気づきの方も多いと思うが, この項には「成功例」が書いていない. 現在までのところ Mac OS X のインターネット接続を利用した VPN 接続には成功していない 筆者の属する研究科の環境では, CISCO VPN Concentrator とその専用 VPN クライアントソフトウェアを利用して VPN 接続を実現している. 次回までにインターネット接続による VPN 接続に成功したら, より詳細なレポートをすることにしよう. 【注】 VPN サーバとして BSD を用いた成功例はウェブ上に散見できる

13.3.3 PPTP ・ L2TP とは

前節で VPN の接続方法として PPTP と L2TP over IPSec という用語がでてきたので, ここでそれらを簡単に解説しておこう. VPN の実現方法としては PPTP (Point to Point Tunneling Protocol), L2TP (Layer 2 Tunneling Protocol), IPSec などがある.[70] ここではそれらの詳細について触れることはしないが, プロトコルの違いや長所・欠点について簡単に解説しておこう.

PPTP PPTP とは, その名のとおり, ネットワーク上の２点間を暗号化パケットを使って通信するためのプロトコルである. 基本的な考え方は PPP (Point to Point Protocol) を基本とするものであり, 実際の認証方法（ユーザ名とパスワードを送信して認証を得る方法）は PPP と同様に PAP, CHAP, MSCHAP, MSCHAPv2 などを用いることができる. PPTP の長所は, 認証のためにユーザ ID とパスワードの組のみを利用し, 余分な情報を必要としないことであるが, 逆にこれが欠点ともなり, 特に PAP 認証では, 認証情報（ユーザ ID とパスワード）を送信するフェーズで, その通信を暗号化することができない.[71] したがって, PAP 認証を用いた PPTP は本当の意味で安全な通信ではない. この問題を改善するために CHAP, MSCHAP, MSCHAPv2 などの「ダイジェスト認証」を用いることができるが, （802.1x の MD5 digest 認証と同様に）これらのダイジェスト認証ではサーバ側には「平文によるパスワード」を保存しておく必要がある.

IPSec IPSec は IP パケットのレベルで暗号化を行う方法であり, 認証フェーズ自身が IPSec の規格に取り込まれている. IPSec では認証フェーズも暗号化され, 安全に認証を行うことが可能であるが, そのかわりに, 認証フェーズにおいて, サーバとクライアントで共通の鍵（「共有シークレット」）を持たなければならない.

L2TP ここまでの２つのプロトコルは「レイヤ３暗号化」と呼ばれるもので, IP パケットを暗号化するプロトコルである. しかし, （Mac の利用者にとっては）AppleTalk[72] に代表される IP とは異なるプロトコルのパケットも VPN を通したいという要求に対して答えるものが L2TP である. L2TP 自身には認証フェースが定義されているわけではないが, L2TP によるアクセス時にも PPTP と同様な認証が行われる.

Mac OS X で利用できる L2TP over IPSec は IPSec の共有シークレットによる IP 層の暗号化パケットの中に L2TP パケットをカプセル化したものである.[73]

とりあえず今回の「最後に」

今回の解説ではネーミングサービスの利用法と, それを利用したネットワークファイル共有や, セキュアな無線 LAN 通信, VPN など, Mac OS X のネットワークの利用法を解説した. しかし, 今回の解説の中では, LDAP, 802.1x, VPN などを実際に構成する各サーバ[74]の設定方法や, これらのサービスとファイヤーウォール設定との関係を全く述べてこなかった.

[70] さらに, 各プロトコルで暗号化の形式や認証方法などに多くの種類がある.
[71] 暗号化のためには, 少なくとも一つの「鍵」を交換する必要がある
[72] AppleTalk が定義されているレイヤは IP 等と同じく「レイヤ２」である.
[73] Mac OS X の VPN 接続では PPTP, L2TP ともに認証は MSCHAPv2 を用いているため, サーバ側では MSCHAPv2 に対応した認証が可能となっていなければならない.
[74] 筆者の属する研究科ではこれらのネットワークサービスの利用が可能なのだが, 実際には Mac OS X Server を用いているのではなく, 各サービスは以下に挙げる機器またはソフトウェアを利用している.

LDAP: iPlanet Directory Server 5.1, Solaris 9.
radius: FreeRadius 0.9.3, Solaris 9,
VPN: CISCO 3005 VPN Concentrator.

次回にはこれらの内容を解説したいと考えているが, Mac OS X だけを利用しているのでは, これらの設定を容易に実現することは難しいものも含まれている. そこで, Apple 社が「サーバ用 Mac OS X」としてリリースしている "Mac OS X Server" を利用してこれらの設定を行うことを考えてみたい. Mac OS X Server は Mac OS X を基本として, ネットワークサーバとしての機能を充実させたアプリケーションを搭載した, 優れたサーバ用 OS である. これを用いることにより, 上記のサーバ設定だけでなく, Mac OS X を用いるクライアントの「ネットワークブート」設定や「ネットワークインストール」なども可能となり, ある程度の台数の Mac OS X ホストの管理が非常に容易になる.

Mac OS X を利用している研究室などで, いろいろなネットワークサービスを実現したいのだけど, 「UNIX を使うのはちょっと...」と思っているユーザの方々も, Mac OS X Server なら扱えるのではないだろうか.

参考文献

NetInfo の詳細な設定方法は NeXTSTEP のシステム管理マニュアル [1] に詳細な解説がある. Mac OS X の NetInfo は NeXTSTEP のものとは微妙に異なっている部分（「local」ドメインの上位層の指定方法やコマンドの詳細な利用方法）があるが, ほとんどは NeXTSTEP のそれと同一と思ってかまわない.

1 NeXT Computer, Inc., NEXTSTEP Network and System Administration, アジソン・ウェスレイ, 1993.

無線 LAN の通信規格である 802.11 とそのセキュリティについては [2,3,4] が参考になる.

2 B. Potter, B. Fleck 著, 802.11 セキュリティ, オライリー・ジャパン, 2003.
3 M.S. Gast 著, 802.11 無線ネットワーク管理, オライリー・ジャパン, 2003.
4 R. Flickenger 著, ワイヤレスコミュニティネットワーク, オライリー・ジャパン, 2002.

802.1x やリモートアクセスに利用される radius サーバについて解説された文献は非常に数が少ないが, 最近次の [5] が出版された.

5 J. Hassell 著, RADIUS, オライリー・ジャパン, 2003.

VPN や IPSec の利用形態や規格について解説された文献としては [6, 7] が参考になる.

6 C. Scott, P. Wolfe, M. Erwin 著, VPN オライリー・ジャパン, 2000.
7 馬場達也著, マスタリング IPsec, オライリー・ジャパン, 2001.