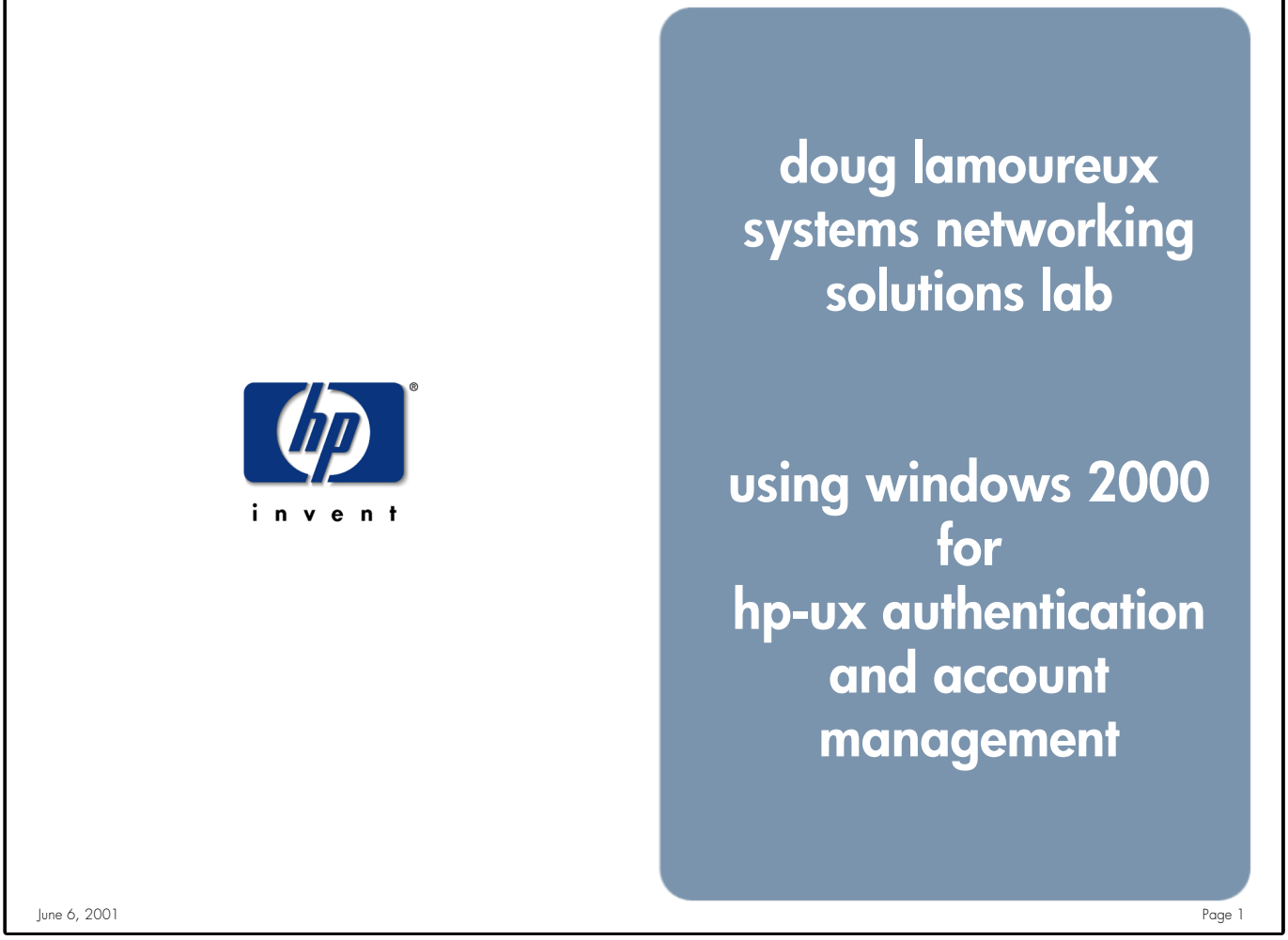

doug lamoureux
systems networking
solutions lab
using windows 2000
for
hp-ux authentication
and account
management
invent
June 6, 2001
Page 1

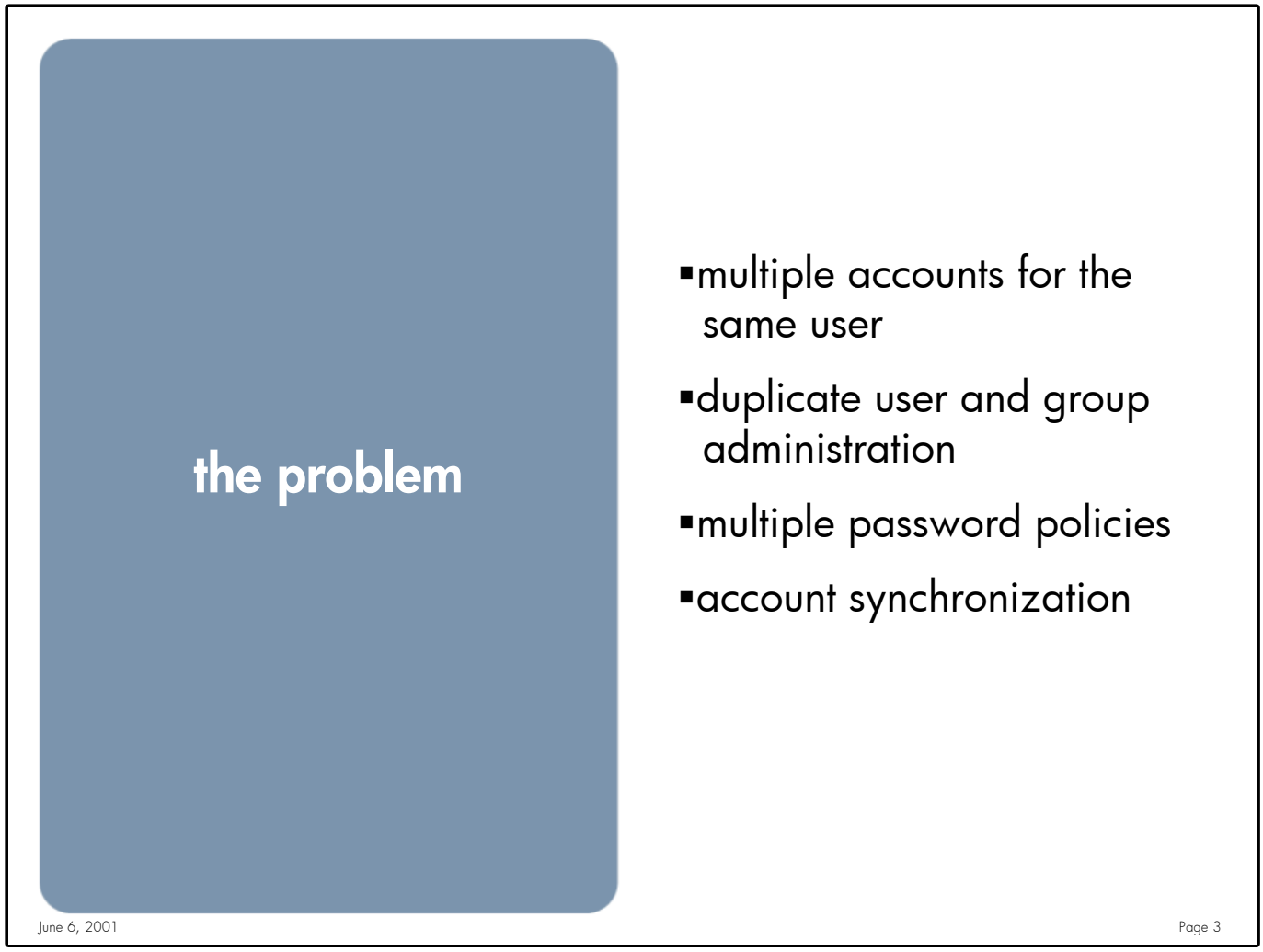
Notes:

agenda

- problem description
- authentication and account management history
- a solution and technologies
- the "big" picture
- other technologies
- limitations

June 6, 2001 Page 2

Notes:



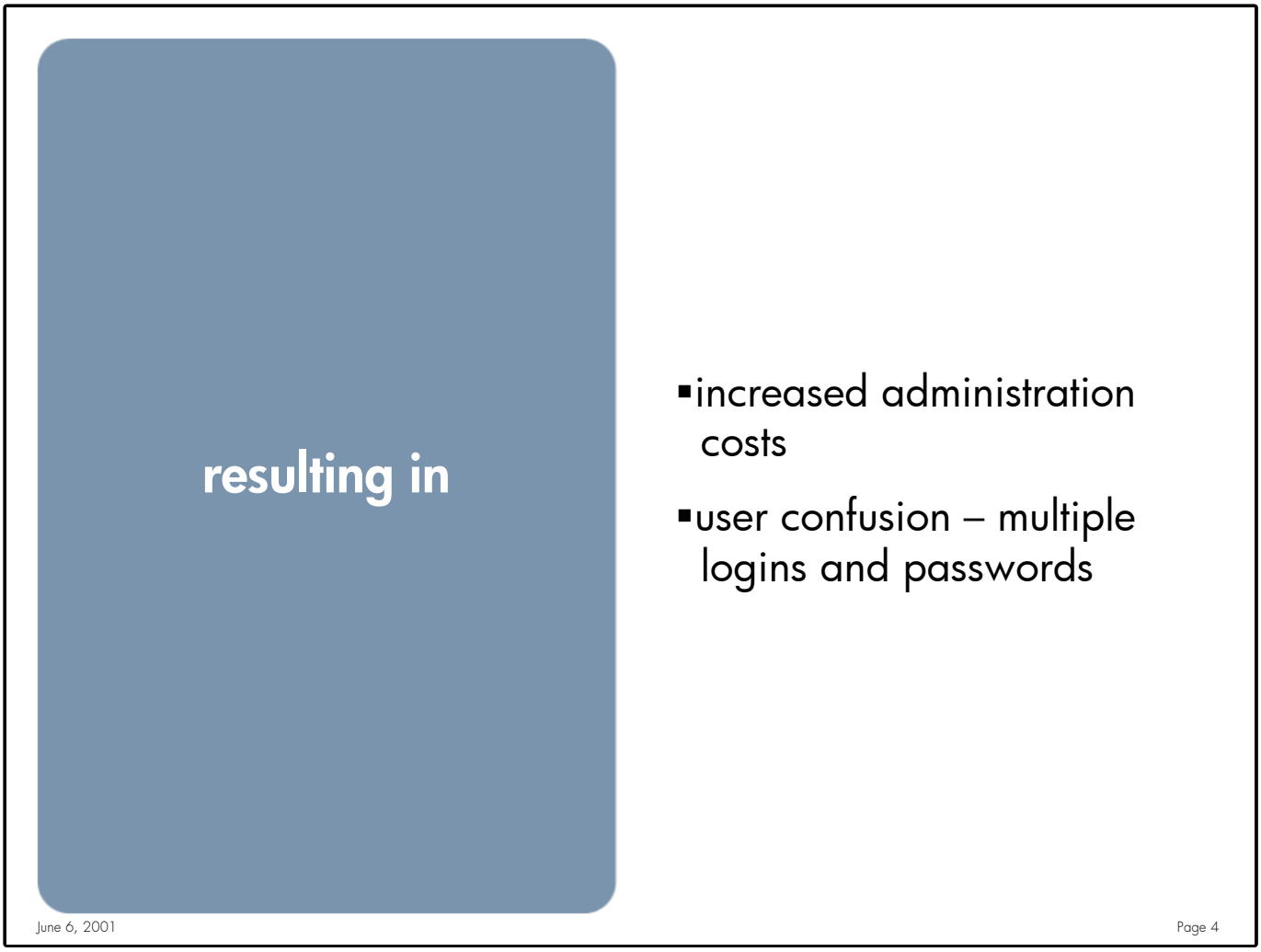
Notes: An account or user name is just a way for a computer to identify a user/client in order to grant access to a resource. Since I will always be <Insert your name here> I should always be identified as myself weather I'm logged into a Unix system or a Windows system.

Because we've duplicated our User's and Account's we need to duplicate our administration effort. You now need to add, update, remove the same information in multiple locations

Password policies may differ, or need to be closely synchronized

Account modifications, deletions must be synchronized between all account databases
Example:

- Name change
- Employee leaves the company



Notes: Maintaining multiple User databases

- Training
- Infrastructure
  - Home grown product
  - 3rd party product
- Administration costs
  - Data synchronization
    - Policy: Could be missed leaving an account active or unchanged
    - Timing: If updates or deletions aren't done immediately you have mismatched data

User confusion

- We have enough user ID's and PIN's to remember
  - ATM, Brokerage, Alarm, Internet Access, Unix, Windows, Web
  - We all use the same, easy to remember EASY to crack password

agenda

- ✓ *problem description*
- **authentication and account management history**
- a solution and technologies
- the "big" picture
- other technologies
- limitations

June 6, 2001 Page 5

Notes:

Notes: In the past you had your Unix users or your Windows users, seldom were they the same

Windows
- NT CHAP (Challenge-Handshake Authentication Protocol)
- NT NTLM

Unix
- Files: Not "network" friendly
- NIS: Not a standard, but widely adopted by most (all?) Unix vendors

hp-ux authentication and account management history

yesterday

account management and authentication technology are tightly integrated

- files
  - insecure
  - not scaleable
  - localized
- nis
  - insecure
  - not scalable
  - distributed

June 6, 2001 Page 7

Notes: Prior to 10.30 most Unix authentication was done using a "crypt'ed password string compare" in which the users crypted password needed to be stored in their password entry.

Files

- Encrypted passwords can be viewed by anyone allowing users to use "crack" programs against your password file
- Data stored in a flat file that doesn't scale well with large user databases
- File is local to the system, not shared with other computers

NIS

- Still insecure, encrypted passwords can be read using ypcat
- "DB" is file based so it still won't scale
- Is client/server based, can be shared by multiple clients

hp-ux authentication and account management history

yesterday *(cont.)*

- nis+
  - secure
  - more scalable
  - distributed
  - too complex to administer
  - not widely accepted
  - future??

June 6, 2001 Page 8

Notes: **NIS+**

- Secure
- Scales better than NIS
- Very complex to administer
- Because of it's complexity it is not being widely implemented

hp-ux authentication and account management history

today

account management and user authentication can now be de-coupled

- ldap (account mgmt & authentication)
  - scalable
  - distributed
  - standards based
  - secure and insecure
- kerberos (authentication)
  - secure
  - distributed
  - standards based

June 6, 2001 Page 9

Notes: With the implementation of NSS(Name Service Switch) and PAM (Plugable Authentication Module) in 10.30 we can now decouple user authentication from account management

LDAP

- True Database that can hold millions of entries, not just users and groups (extensible)
- Network based PROTOCOL (See RFC list at the end of slide set)
- Multiple master and slave architecture (High Availability built in)
- Industry standard defined protocol

Kerberos

- Secure
  - Uses a "trusted 3rd party" authentication system
  - Passwords are not transmitted across the wire
  - 2-way authentication (server authenticates client; client authenticates server)
- Network based protocol
- Industry standard defined protocols (See RFC list at the end of slide set)

windows authentication and account management history

yesterday

- lan manager
- nt chap
- ntlm

today

- ldap
- kerberos

June 6, 2001 Page 10

Notes: Yesterday

Proprietary authentication

Today

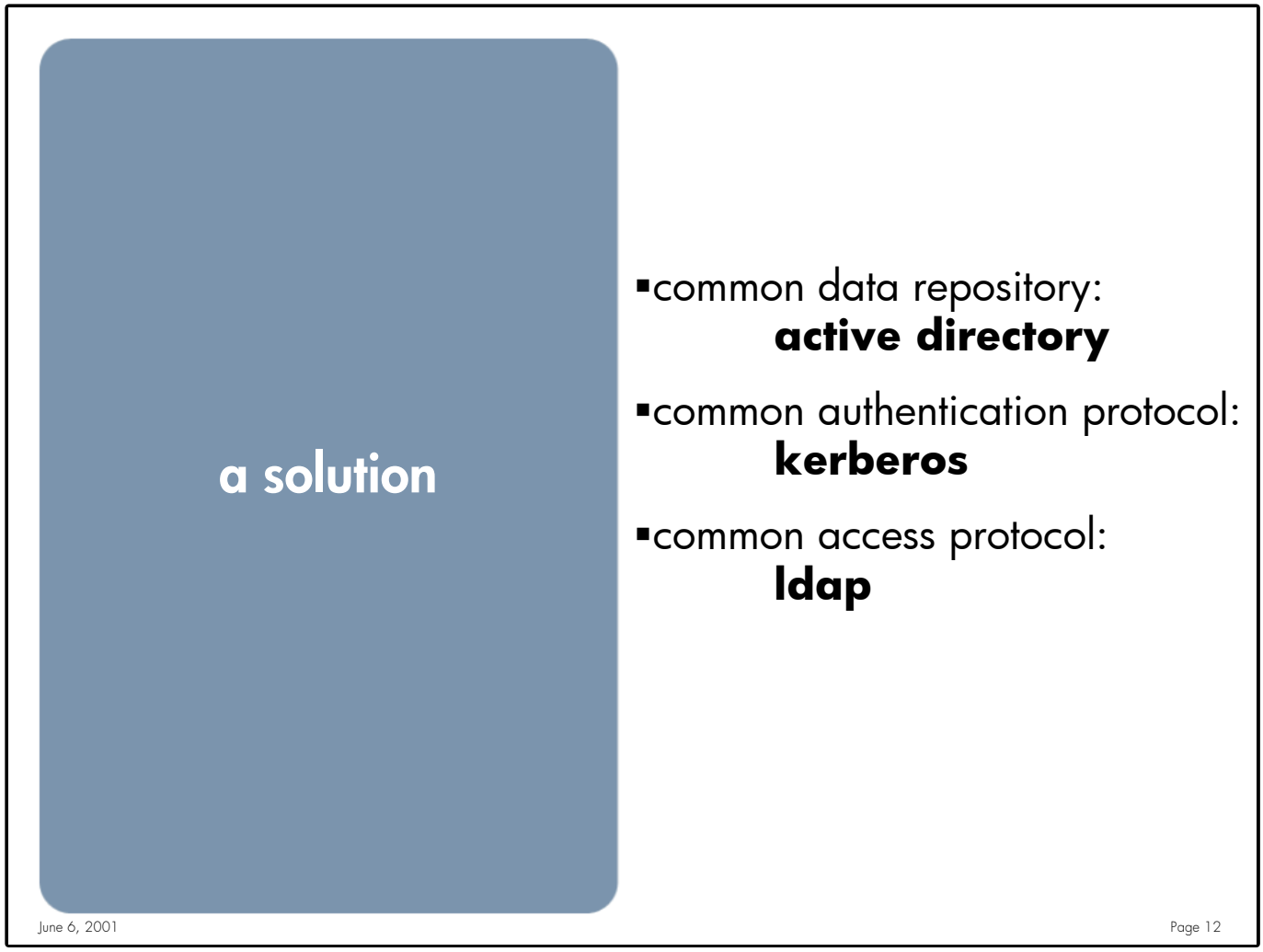
Standard based protocols (plus MS “enhancements”)

agenda

- ✓ *problem description*
- ✓ *authentication and account management history*
- ❑ **a solution and technologies**
- ❑ the “big” picture
- ❑ other technologies
- ❑ limitations

June 6, 2001 Page 11

Notes:



Notes: This is a single solution. There are other solutions out there:

- Meta-directories – An "umbrella" directory that integrates data from existing repositories into a single directory with central administration
- Synchronizing Active Directory data with other Directories – There are no current ldap replication standard (there are drafts in the IETF) but it is possible to create custom scripts/applications to loosely synchronize data between different directories

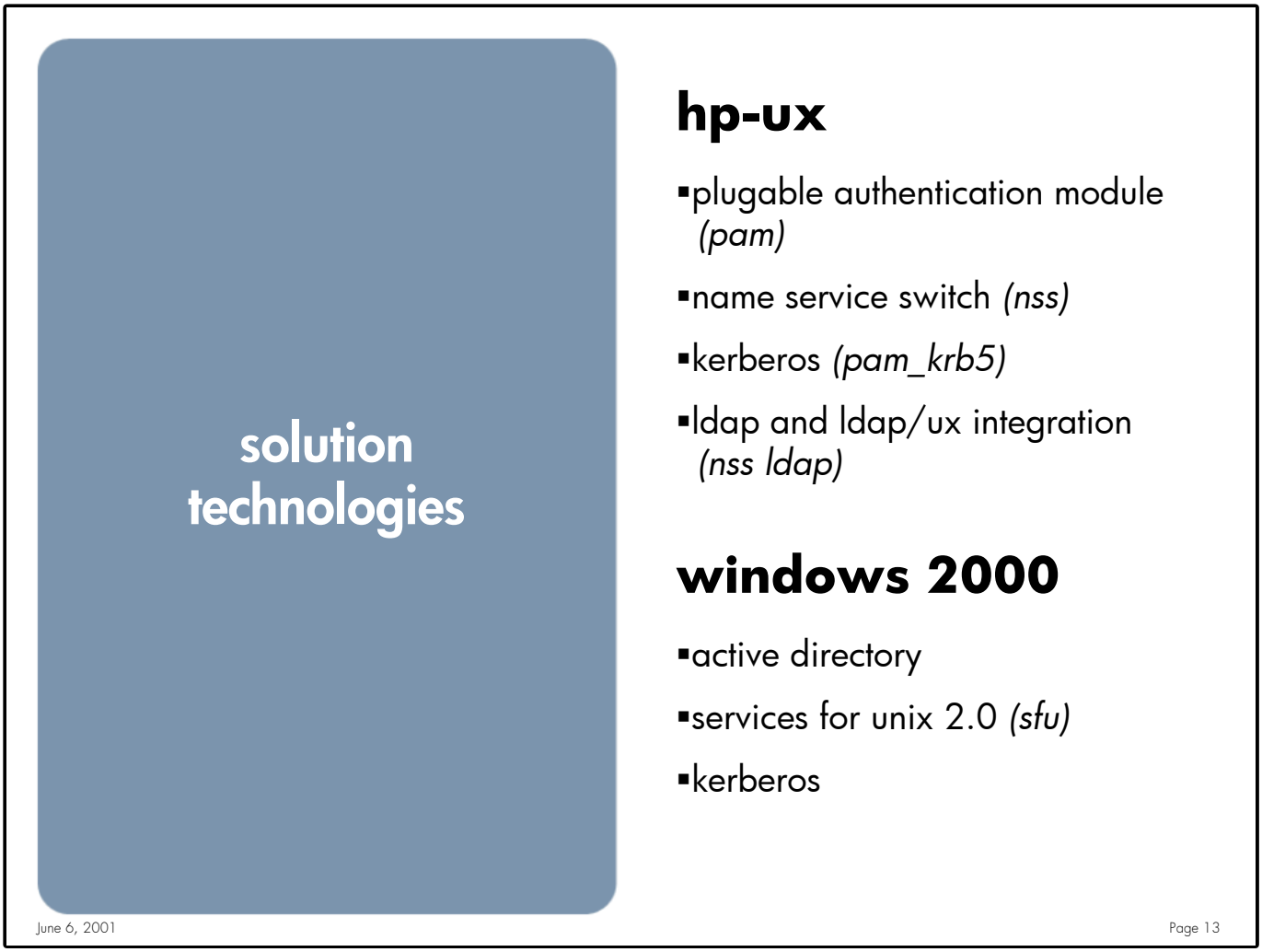
By using a single repository for users we eliminate the need for multiple "accounts" for the same user.

There is no need to synchronize user/group data

A single password policy can be instituted across platforms

Account management is done in one place.

This does not mean you no longer need Unix Administrators



Notes: Since Authentication and Account lookups are decoupled it's possible to use different technologies for each. In this solution LDAP is used to lookup account information (Name Service Switch) and Kerberos is used for authentication (PAM).

kerberos

kerberos

is

- a network authentication protocol
- uses a trusted 3rd party (KDC) to distribute "tickets"
- can be used to encrypt data between clients and servers

is not

- a network authorization protocol
- a repository for user account information

June 6, 2001 Page 14

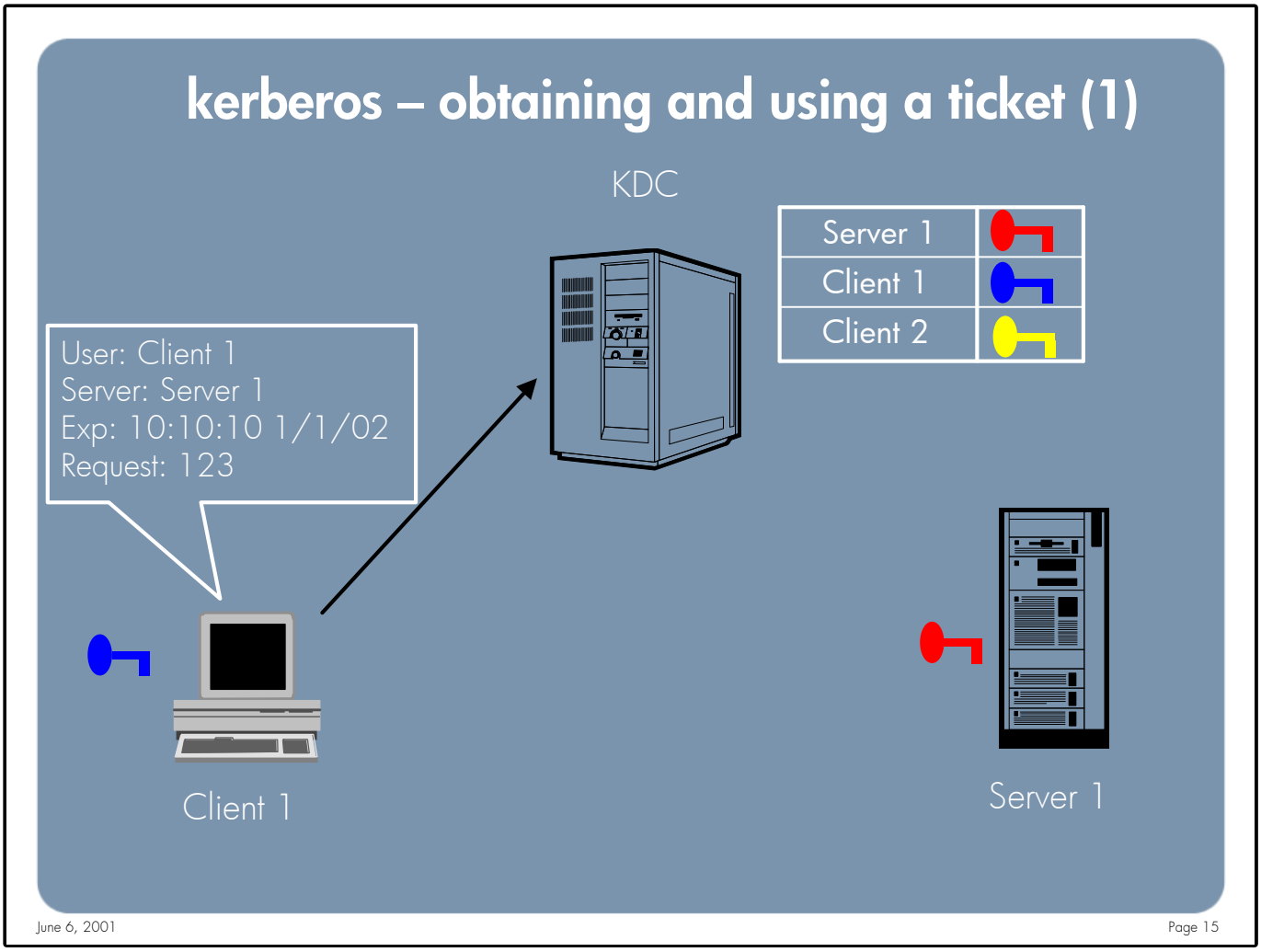
Notes: The name Kerberos comes from Greek mythology; it is the three-headed dog that guarded the entrance to Hades.

Kerberos is used to authenticate both clients and servers, it does **not** provide authorization to services or resources. Kerberos does, however, provide the framework for services to pass authorization data.

Kerberos assumes that the 3rd party is secure and trustworthy. If the 3rd party is compromised then all of your users are too.

While Kerberos does not itself provide encryption for application data it can be used to exchange encryption keys securely, leaving it up to the application to encrypt data.

Kerberos does NOT provide a means to store user account information like:
- uid number
- Home directory
- etc

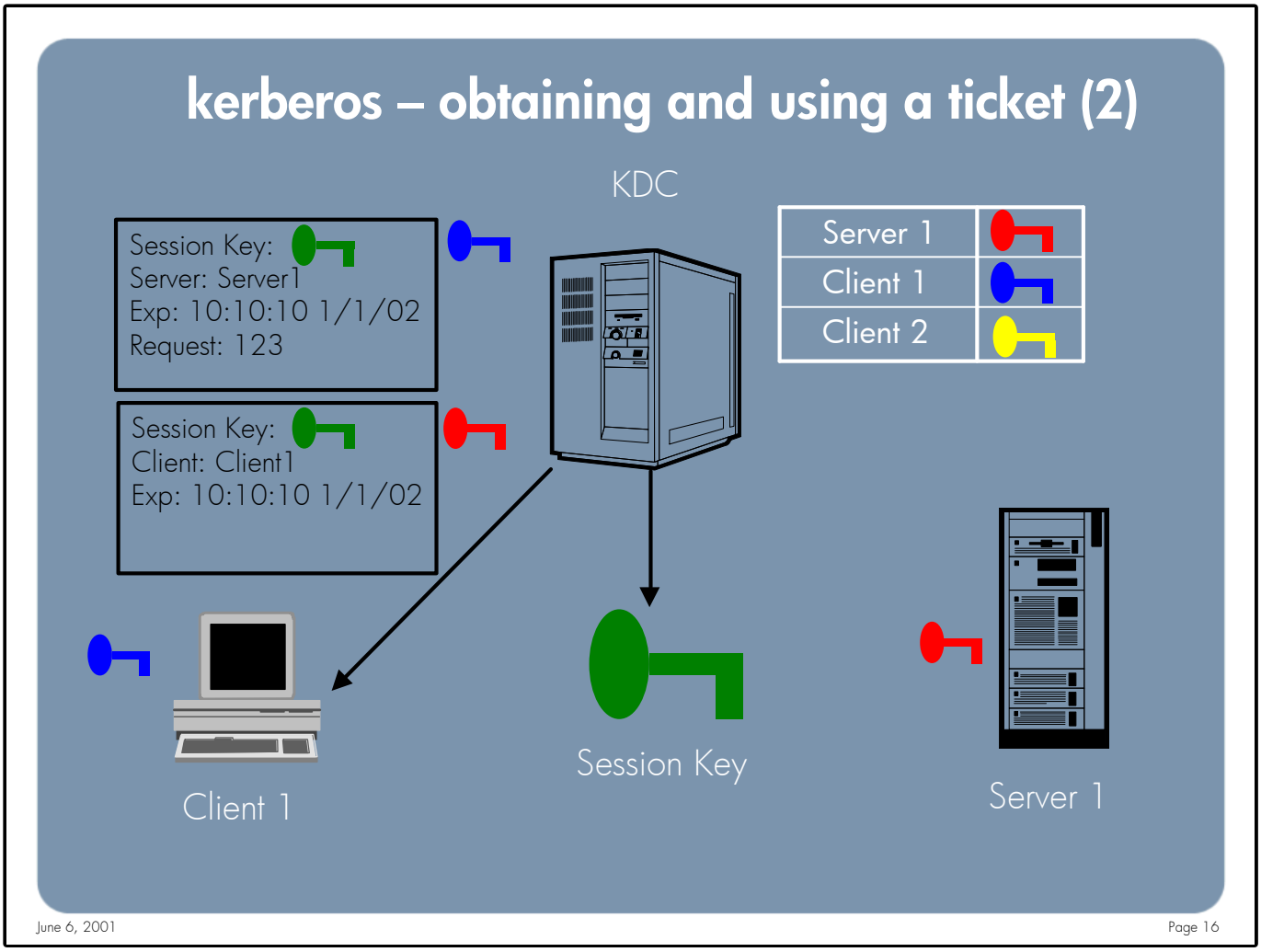


Notes: A user/client that wishes to communicate with a server/service must 1st request a "ticket" and session key for that service from the trusted 3rd party (Authentication Server). To obtain a ticket for the server the following actions take place:

The client sends a message to the Authentication Server requesting a ticket for Server 1.

In this request the client includes:

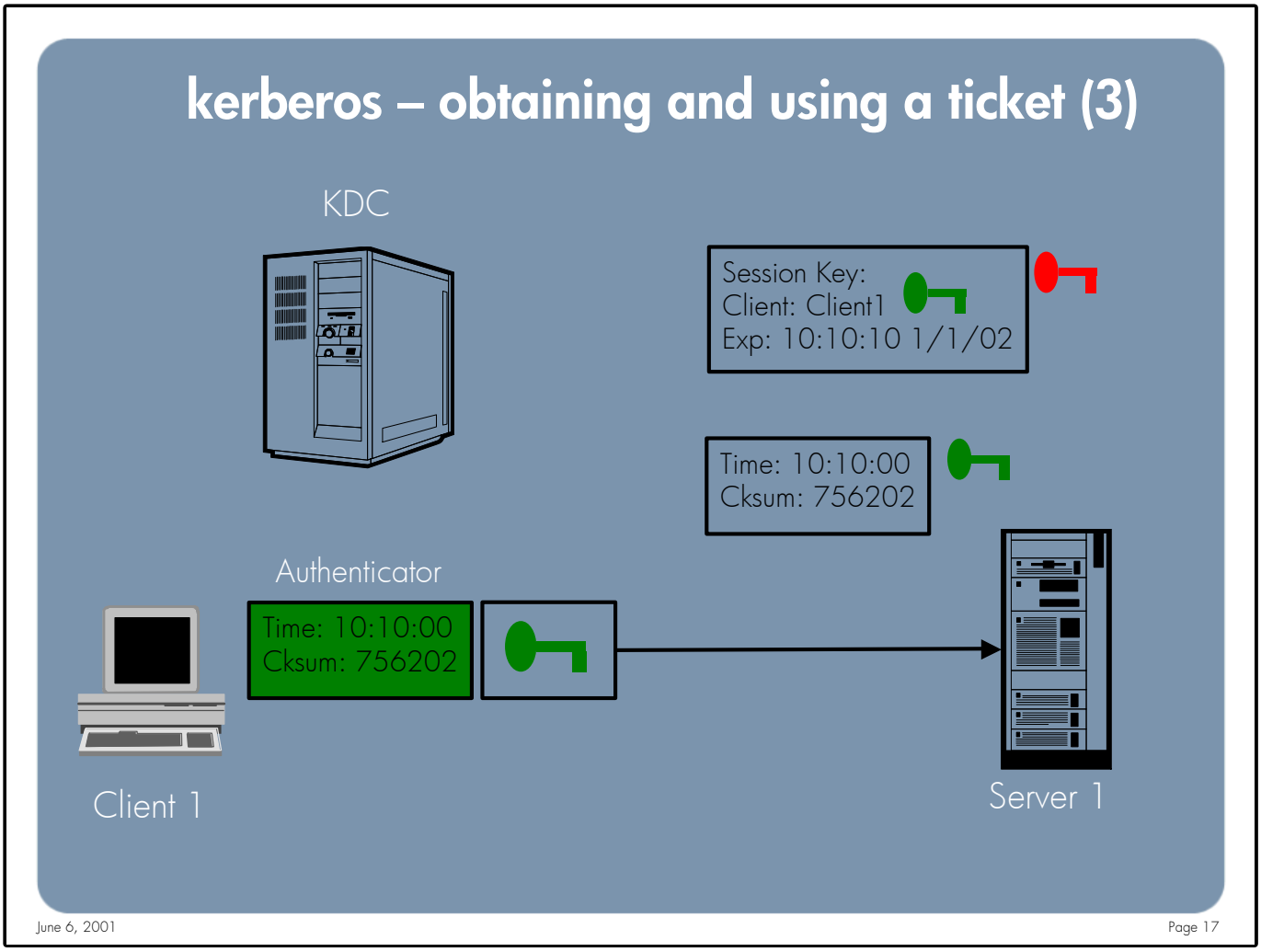
- users name
- servers name
- expiration time for the session ticket (requested)
- random "request" number



Notes: The Authentication Server generates a session key then replies with 2 "packages":

1. Encrypted with the users key
   - Session Key for the Server
   - Expiration time for the session key
   - Random Request number(sent by the client)
   - Server name
2. Encrypted with the Servers Key (not readable by the client)
   - Session Key
   - Client name
   - Expiration time for the session key

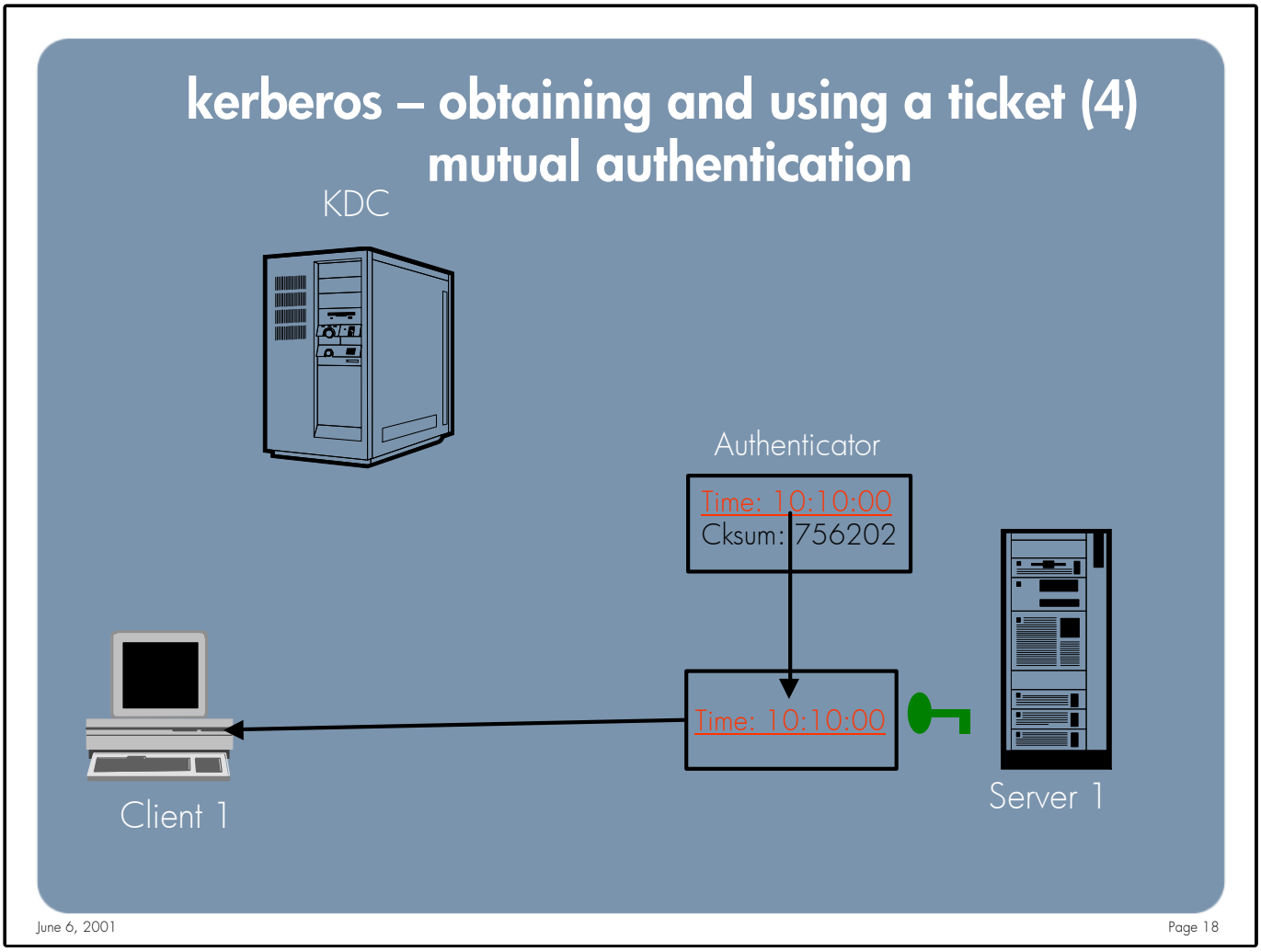
NOTE: The users key is not their password. A Key is a much longer string derived from the users password.



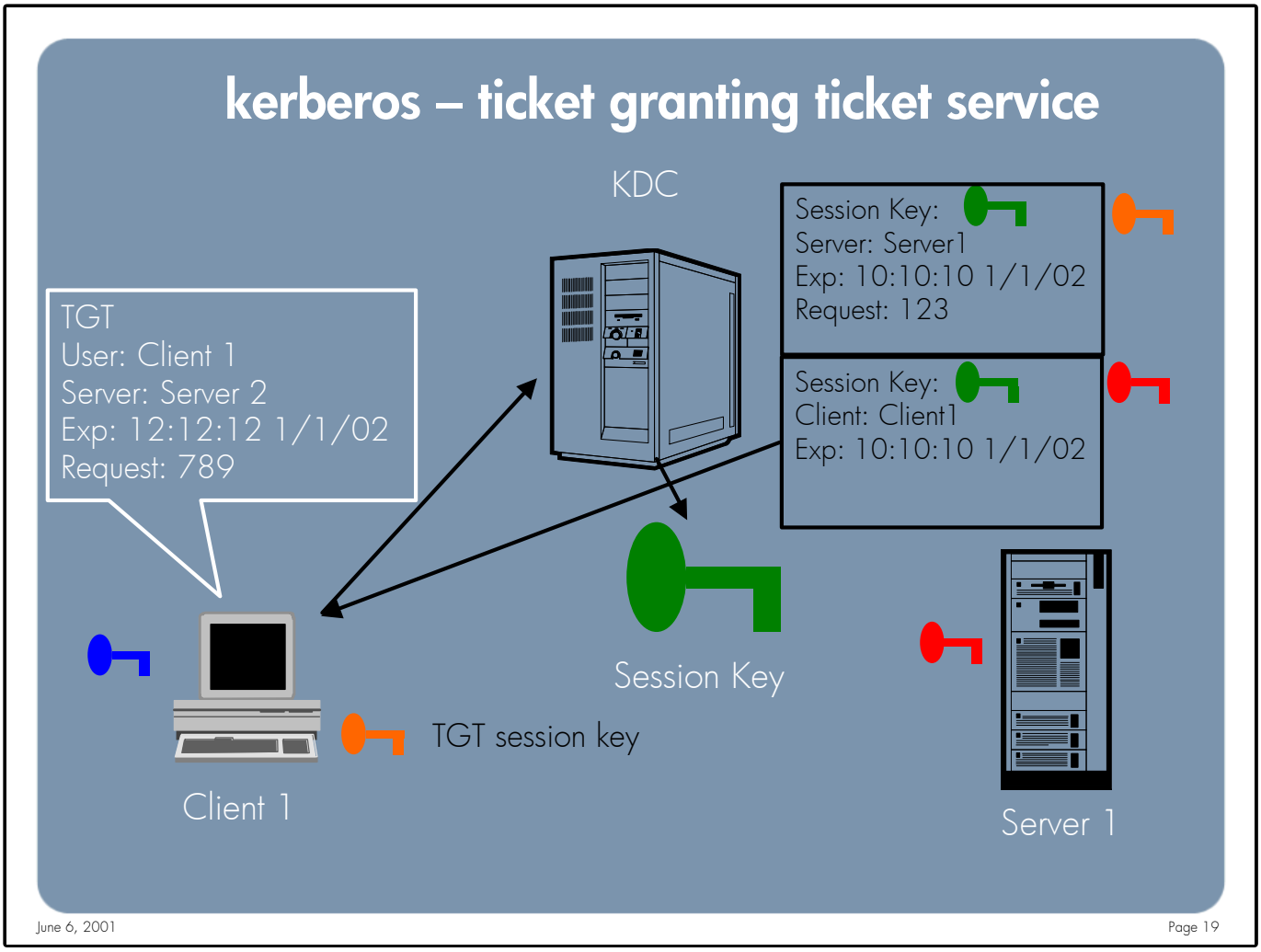
Notes: The client sends an "application request" to the server. This application requests contains:

- The ticket encrypted in the servers key
- An authenticator that is encrypted with the session key. The authenticator contains:
  - The current time
  - Checksum

The server then uses it's key to decrypt the ticket. The session key is extracted and used to decrypt the authenticator. It then compares the checksum computed with the checksum the client provided in the authenticator, if they are the same then user named in the ticket (who the session key was generated for) created the authenticator. This does NOT completely authenticate the user (the authenticator could be replayed in the future if captured on the wire). To complete the authentication the Server compares the current time with the timestamp in the ticket, if they are with the allowed time period AND there were no other requests within the allowed time period with the same timestamp then the server excepts the client's "application request".



Notes: Optionally the client can request that the server authenticate itself (mutual authentication). To authenticate itself to the client the server takes the timestamp that the client included in it's authenticator and encrypts it, along with other information, using the session key. If the client decrypts the timestamp and it matches what it sent in it's authenticator then the server was able to extract the session key and therefore must know the key that was used by the Authentication Server to encrypt the ticket.



Notes: As you can see from the previous slides the user will need to supply it's password every time it wants to get a ticket for a new service/server. To solve this problem the user can obtain a ticket (TGT) for the "Ticket Granting Server" (TGS). This is a special ticket used to obtain tickets for other server/services without requiring the user to enter their password each time. This ticket is obtained using the same procedure described in the previous slides.

Here's how it works:

- The client obtains a ticket for the Ticket Granting Server (as described in the previous slide)
- The client decrypts the ticket and caches(in a file on HP-UX) the ticket and the session key
- The client needs a ticket for a server/service so it sends an "authentication request" to the TGS, just as it would any other server except it contains information (the TGT it previously obtained) to authenticate itself to the TGS
- The TGS then uses the session key generated in the clients TGT request, not users key, to encrypt the response to the client.
- The client uses the session key from the TGT to decrypt the response and proceeds as normal.

kerberos configuration on hp-ux

kerberos configuration file: /etc/krb5.conf

```
[libdefaults]
        default_realm = ACME.COM
        default_tkt_enctypes = DES-CBC-CRC
        ccache_type = 2
        default_keytab_name = FILE:/etc/krb5.keytab

[realms]
        ACME.COM = {
                kdc = kdc1.west.acme.com:88
                kdc = kdc2.west.acme.com:88
                admin_server = kdc1.west.acme.com
        }

[domain_realm]
        acme.com = ACME.COM
```

June 6, 2001 Page 20

Notes: **default_realm:** Used to determine what Kerberos Realm a client is to use if it's not specified. Using the kinit program I could specify the realm I want to login to (jcool@REALM1.ACME.COM) or just specify my user (jcool) and kinit will attempt to log me into the default realm

default_tkt_enctypes: The supported list of session key encryption types that should be requested by the client

ccache_type: Credential cache format (must be set to 2 in order to use the DCE klist program with PAM Kerberos)

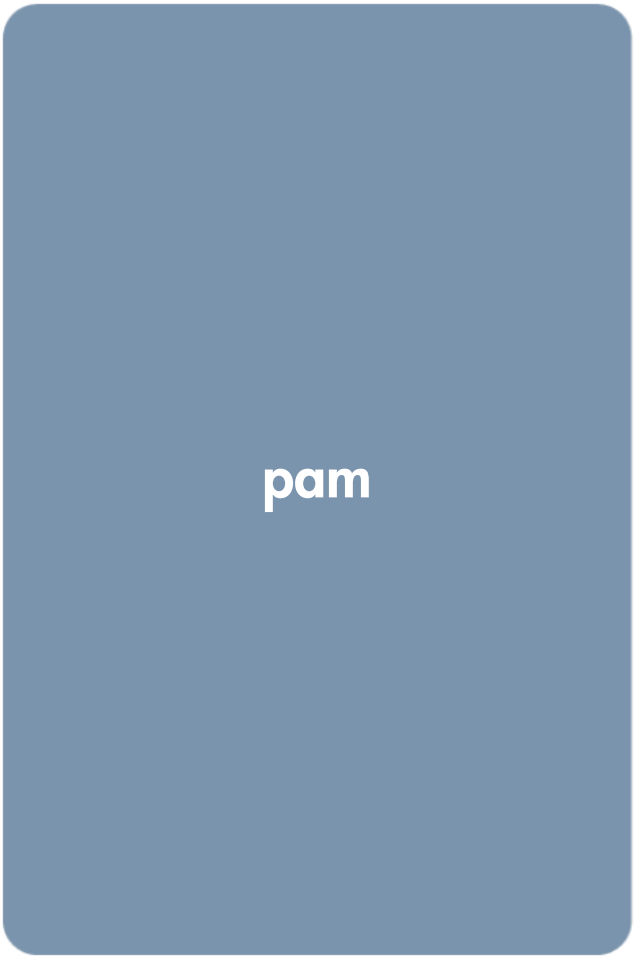
default_keytab_name: The default keytab name to be used by application severs such a telnetd and rlogind (not needed for PAM Kerberos, but required for Secure Internet Services, SIS)

[realms]: Section to define all Kerberos Realms that can be used by this system.

kdc: Specifies the Key Distribution Center (KDC) for the realm (multi-valued)

admin_server: Specifies where the administration server is running. Typically this is the Master Kerberos server.

domain_realms: Maps a domains (and subdomains) to a Kerberos Realm
There are several other parameters, see the krb5.conf man page for more details.



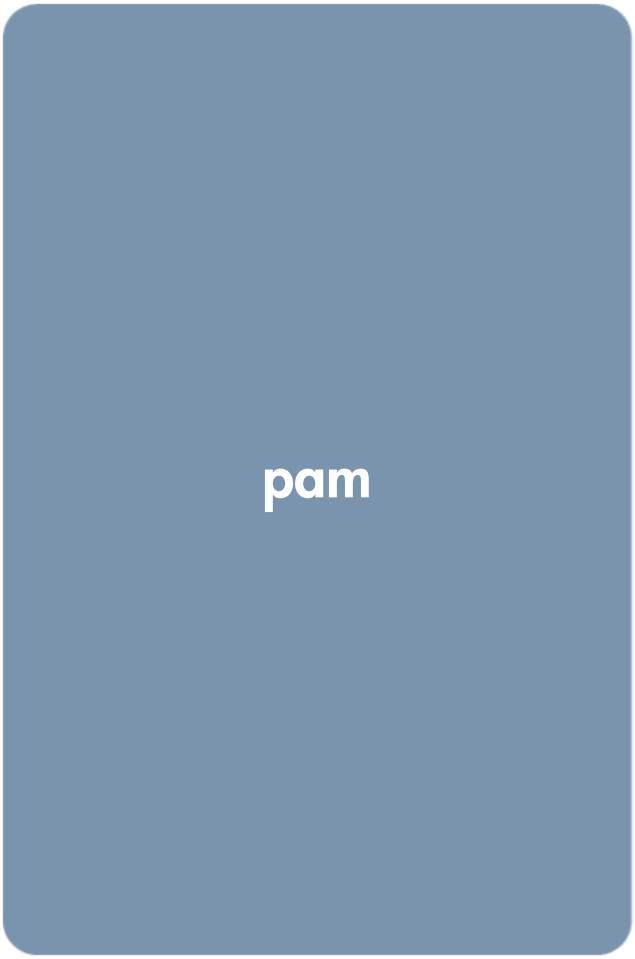
plugable authentication module

- allows server/service developers to write their authentication code to a standard api
- allows system administrators to choose which authentication module(s) to authenticate users with
- by "stacking" modules a user can be authenticated by multiple modules

Notes: By writing to an abstract API the application need not concern itself with the authentication technology used to authenticate the user. The application will call the pam_XXX() API which will in turn call the PAM module the system administrator has configured.

Administrators have the flexibility to choose the authentication modules that best fit their environment. Or they can develop their own PAM module. Example:
Secure: PAM_Kerberos
Less Secure: PAM_Unix

It's possible to configure PAM so that a user can be authenticated by multiple modules. There are options that tell PAM how to handle multiple "stacked" modules. For example I may want all of my modules to authenticate the user before he is authenticated on the system. Or it may be sufficient to authenticate the user to the system if any of the stacked modules successfully authenticate the user.



pam configuration

pam configuration file:
/etc/pam.conf

each entry is a single line containing:

service_name –
a service such as login, dtlogin, ftp
module_type –
the type of module (auth, account, session & password)
control_flag –
controls how stacking is interpreted
module_path –
path to the library of this module
options –
a list of options to passed to the module

Notes:

pam configuration (cont.)

example:

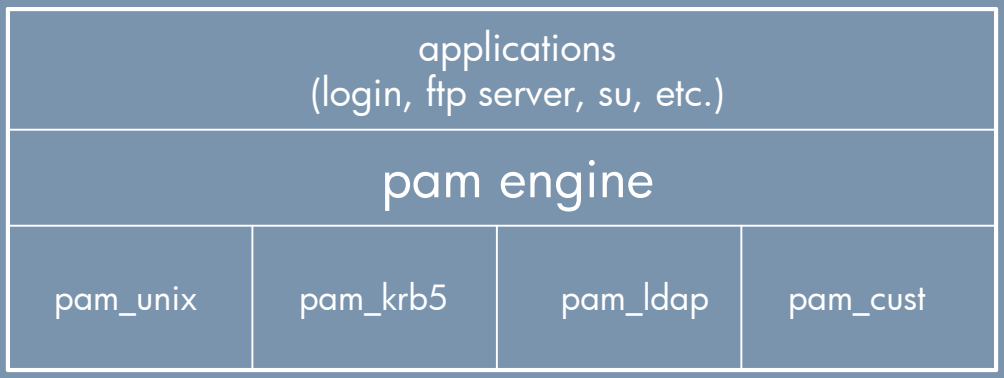
```
login     auth sufficient   /usr/lib/security/libpam_krb5.1 forwardable
login     auth required     /usr/lib/security/libpam_unix.1 try_first_pass
ftp       auth  sufficient   /usr/lib/security/libpam_krb5.1
ftp       auth  required     /usr/lib/security/libpam_unix.1 try_first_pass

login       password required /usr/lib/security/libpam_krb5.1
login       password required /usr/lib/security/libpam_unix.1
passwd      password required /usr/lib/security/libpam_krb5.1
passwd      password required /usr/lib/security/libpam_unix.1
```

June 6, 2001 Page 23

Notes: The top section of this example shows how stacking is used for the auth{entication} module. For the login service it is "sufficient" if the user is authenticated by the PAM_KRB5 (kerberos) module. If that fails, or the user does not exists in the kerberos "world" the user can still be authenticated by the PAM_UNIX module. There are also options at the end of each module. The "forwardable" option tells the PAM_KRB5 module to obtain Forwardable Credentials, this is a specific option that only the PAM_KRB5 module understands. The "try_first_pass" option on the PAM_UNIX module tells PAM to "try" the password that the user typed in for the 1st module. If that fails then the user will be prompted for a password.

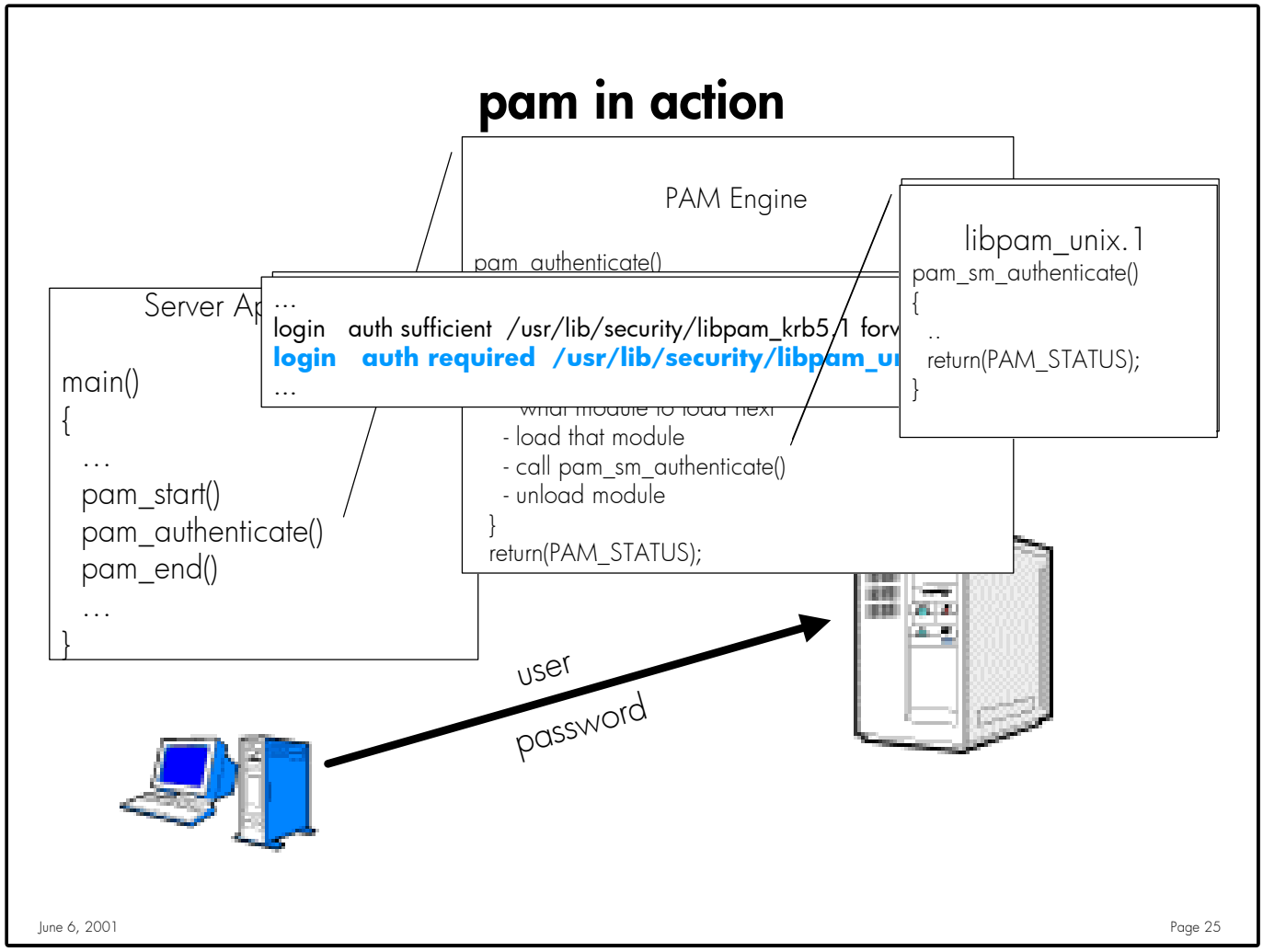
The bottom half of the slide shows an example of the password module. Again we are stacking the modules so that if a user exists in both "worlds" (pam_krb5 and pam_unix) then they can change their password in both. There are 2 services listed, login & passwd, in this example. The 1st service (login) would be used in the event that your password has expired and the "login" process is requiring you to change your password. The 2nd service (passwd) is used when a user types the passwd command to change their password. There are other services (dtlogin, dtaction, ..) that can be configured for the password module, but you won't find ftp. The reason for this is that the ftp protocol does not provide a mechanism to change passwords.

pam architecture

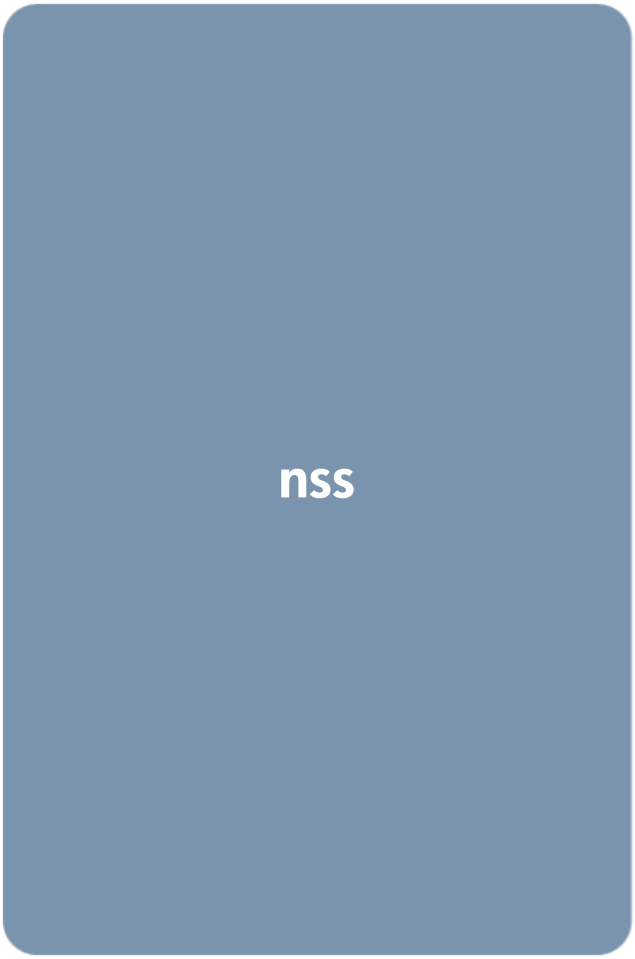


June 6, 2001 Page 24

Notes:


pam in action
PAM Engine
pam_authenticate()
libpam_unix.1
pam_sm_authenticate()
{
..
return(PAM_STATUS);
}
Server Ap
main()
{
…
pam_start()
pam_authenticate()
pam_end()
…
}
…
login auth sufficient /usr/lib/security/libpam_krb5.1 forv
login auth required /usr/lib/security/libpam_u
…
- load that module
- call pam_sm_authenticate()
- unload module
}
return(PAM_STATUS);
user
password
June 6, 2001
Page 25

Notes:



name service switch

- use to determine the source from which various system calls will attempt to retrieve data from
- multiple sources can be configured
- possible to configure different actions for each status returned from each source

June 6, 2001 Page 26

Notes: The operating system uses a number of "databases" of information about hosts, users (passwd), groups and so forth. Data for these can come from a variety of sources:

Similar to PAM's stacking ability it's possible to list multiple sources to be queried for each "database":

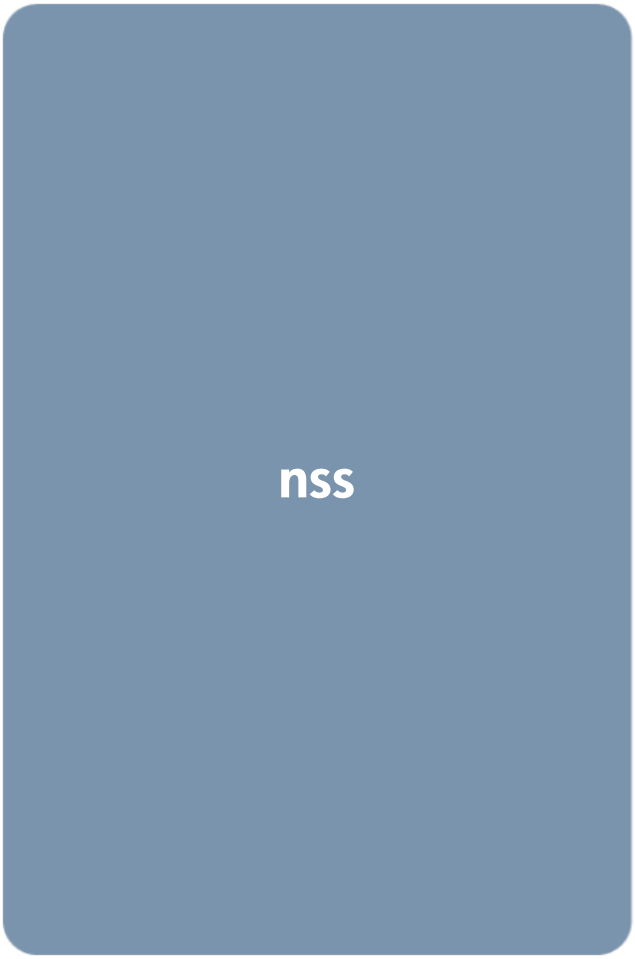
hosts: file dns ldap nis

It's possible to configure a different action based on the return status from each source. Each source must return one of the following:

SUCCESS	Requested database entry was found
UNAVAIL	Source is not responding or corrupted
NOTFOUND	Source responded "no such entry"
TRYAGAIN	Source is busy, might respond to retries

For each status code, two actions are possible:

CONTINUE	Try the next source in the list
RETURN	Return now



name service switch

there are 2 different types of nss apis

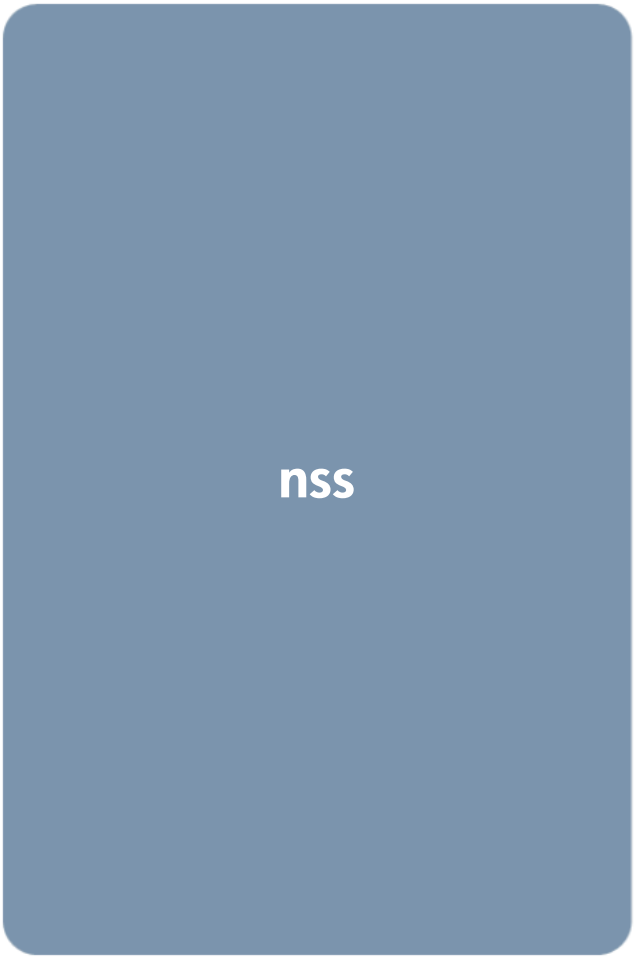
- searches
  - look for a specific entry
  - quick response
- enumeration
  - look at all entries in a “database”
  - increase network load
  - increase “server” load

June 6, 2001 Page 27

Notes: When an application makes a call like getpwent () or getgrent() it’s attempting to look at every entry in the “map”, one at a time. This means that the NSS back end will need to retrieve all entries from the server that’s supports this “map”. If your using an LDAP Directory Server that can store over a million entries you will impact the performance of your network and server.

It’s also worth noting that most Directory Servers limit (or can) the number of entries it will return to a query like this. MS Active Directory, by default, will only return 1000 entries. Its possible for an application to perform incorrectly if the number of entries in the search exceed the number of entries a server is willing to return.

For a more in-depth discussion on this topic please see the “Preparing your LDAP Directory for HP-UX Integration” white paper found at: http://docs.hp.com/hpux/internet/#LDAP-UX%20Integration



nss configuration

nss configuration file:
/etc/nsswitch.conf

each entry is a single line containing:

database –
the name associated with the type of information to be looked up (i.e. hosts, passwd)

source –
the module name used to query (multi-valued)

criteria –
controls what should be done for each possible return status (optional)

June 6, 2001 Page 28

Notes: The library functions contain compiled-in default entries that are used if the appropriate entry in nsswitch.conf is absent or syntactically incorrect.

The default criteria are to continue on anything except SUCCESS; in other words:

[SUCCESS=return NOTFOUND=continue UNAVAIL=continue
TRYAGAIN=continue]

nss configuration (cont.)

Example:

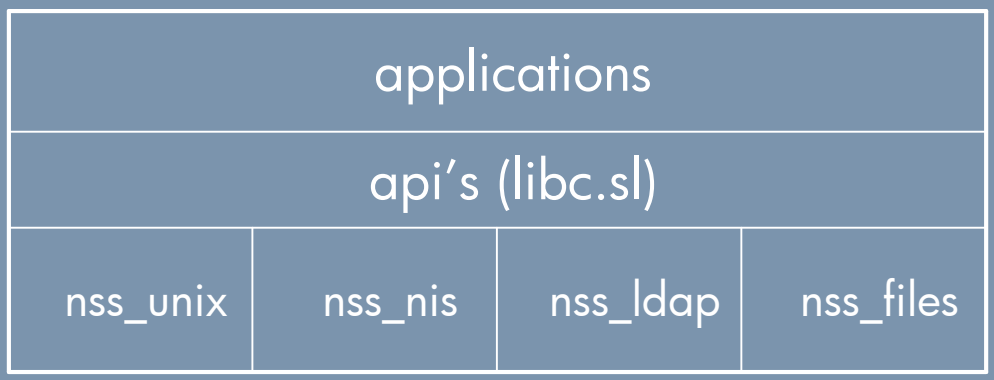
hosts:	dns [NOTFOUND=continue TRYAGAIN=return] files nis
passwd:	ldap [NOTFOUND=continue TRYAGAIN=continue] files
group:	ldap [NOTFOUND=continue TRYAGAIN=continue] files
networks:	files nis
protocols:	nis [NOTFOUND=continue TRYAGAIN=return] files

June 6, 2001 Page 29

Notes: For hosts DNS is tried 1st, if the server is "busy" (TRYAGAIN) stop and return back to the calling process without an answer. If the request is not found using DNS try files (/etc/hosts). If the answer is not found in files we will continue on to NIS (since nothing was specified as an action use the default which is CONTINUE). At this point what ever NIS returns will be passed up to the caller.

For user password entry (passwd) lookup's LDAP is 1st attempted. If the user is not found in the LDAP directory or the Directory server is too busy files (/etc/passwd) will be queried for the information.

nss architecture



June 6, 2001 Page 30

Notes: The nsswitch.conf file is only read once by an application, the 1st time it's needed. Any subsequent calls that require NSS will use the configuration that was 1st read. This can be a problem for long running programs. Since the configuration is only read once these application will not be aware of any subsequent changes made to the NSS configuration.

lightweight directory access protocol

defines:
add, delete, modify, and search operations

features:

- lightweight (x.500)
- open standard
- runs over tcp/ip
- flexible directory structure (flat or deep)

June 6, 2001 Page 31

Notes: LDAP is a PROTOCOL, it defines how you access the Directory, not how a Directory is implemented

The LDAP protocol is defined in several RFC's listed at the end of this presentation

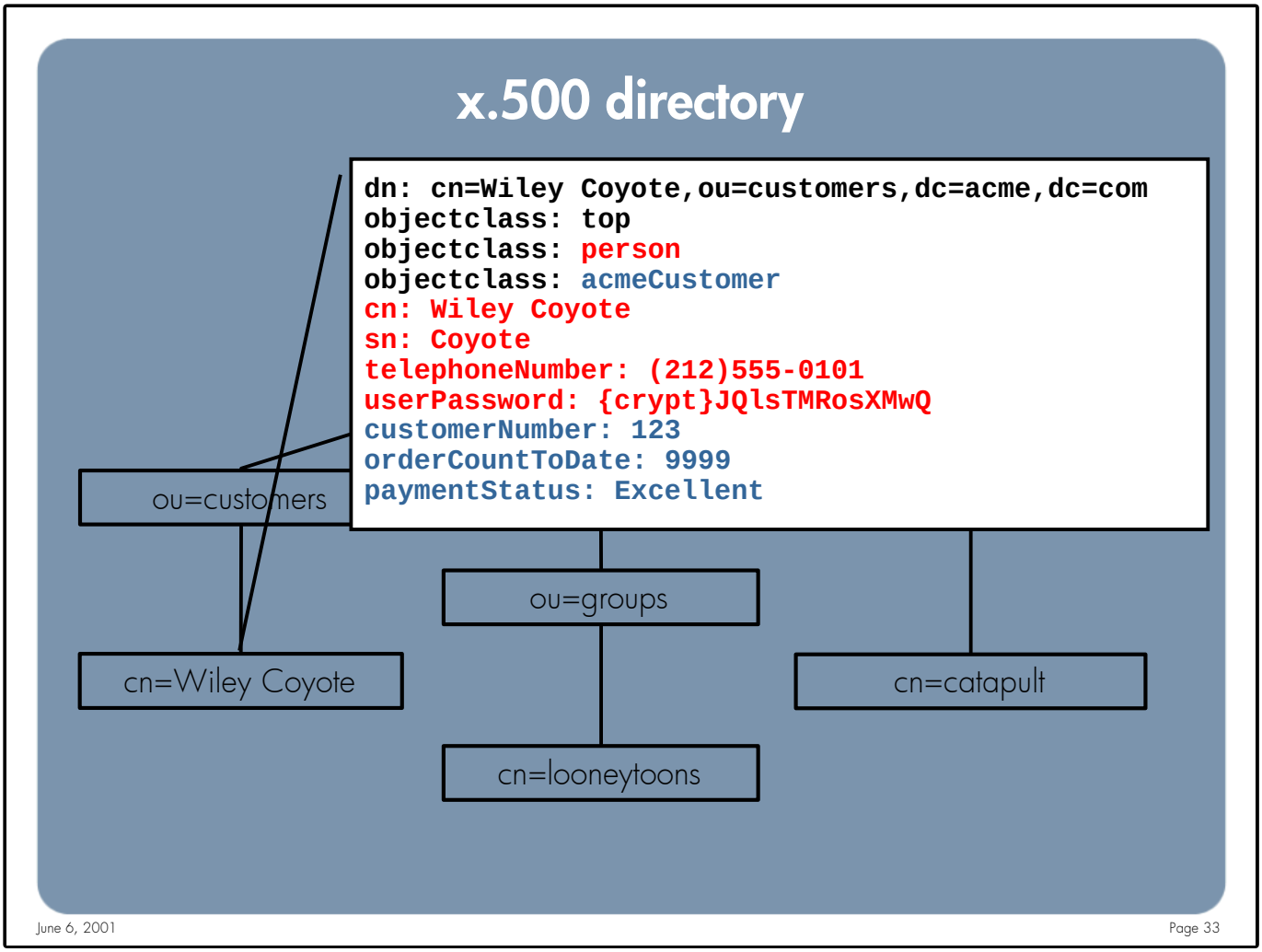
what is a directory?

- used to store information about an object
- an object can be just about anything (person, group, part)
- ldap based directories have generally been used to hold information about people
- similar to a database
  - designed for lookups (not updates)
  - the schema defines data format
- similar to a posix filesystem
  - hierarchical in structure
  - each object has a distinguished name (dn) and a relative distinguished name (rdn)

June 6, 2001 Page 32

Notes: Historically directories have been used to store information about users (internet portals). Because of it's extensibility directories are now being used to hold many different types of information (configuration, parts catalog, etc)

A distinguished name (DN) is synonymous to the full path of a file in a posix file system, for example:
/etc/hosts is the DN for a file name hosts in the /etc directory.
A relative distinguished name (RDN) is equivalent to the "short" file name, hosts in our example. Looking a a second file, /tmp/hosts, it has the same RDN as as /etc/hosts, but a different DN. Since an RDN is "relative" to the branch it resides in it's OK to have a duplicate RDN.



Notes: The "DIT" (Directory Information Tree) defines the structure of your Directory Tree

DITs are hierarchical and can be very flat or deep

The base name of an entry is called an RDN (Relative Distinguished Name.). For example: cn=Wiley Coyote

The DN (Distinguished Name) is built by appending all the parent RDNs. For example:
cn=Wiley Coyote,ou=customers,dc=acme,dc=com

A DN is expected to be unique.

active directory

- central part of the windows 2000 networking architecture
- allows enterprise resource to be stored to and retrieved from a distributed, replicated location
- is “the” central network security authority
- centralized point of user and group management
- supports the ldap protocol for remote access to directory objects
- multi-master

June 6, 2001 Page 34

Notes: Active Directory is the “brains” behind the Windows 2000 networking environment. All users, groups and network resources are stored in AD. Because all resources are stored in the same Directory administrators have a single point of administration.

ldap/ux integration consists of 2 products

nis/ldap gateway

- used to translate nis requests into ldap requests
- allows clients that don't support ldap natively to use an ldap directory to store/retrieve users and groups
- the gateway replaces existing nis slaves

ldap-ux client

- allows hp-ux systems direct access to nis information in ldap v3 compliant directories
- allows users to be authenticated against the directory

June 6, 2001 Page 35

Notes: LDAP/UX Integration is a free product available on the HP-UX Application CD and http://software.hp.com

NIS/LDAP Gateway is not being used in this solution

ldap-ux client

- nss_ldap module used to access nis "maps" (user, group, rpc, etc.) in an ldap directory
- pam_ldap module provides ldap based authentication
- flexible profile configuration
  - attribute mapping
  - search filters
  - configurable access
  - multiple profiles

June 6, 2001 Page 36

Notes: The profile currently being used by LDAP/UX is proprietary. There is a draft being proposed by HP, Sun & Open Source representatives to define a standard configuration profile. The current version of this draft can be found at:

`http://www.ietf.org/internet-drafts/draft-joslin-config-schema-01.txt`

The version of the draft is likely to change when revisions are made (..02.txt, 03.txt, etc.)

attribute mapping

allows you to map one attribute to another:

```
attributemap:
          passwd:userpassword=*NULL*
attributemap:
          passwd:gecos=cn building
                 telephonenumber
attributemap:
          passwd:uidnumber=employeeId

attributeMap:
          passwd:homedirectory=
                   msSFUHomeDirectory
```

June 6, 2001

Page 37

Notes: Map userpasswd to NULL so it's not viewable
Map gecos field to cn building telephonenumber
Map uidnumber to employeeId

Mapping that is used by LDAP/UX with Active Directory:
attributeMap: passwd:userpassword=msSFUPassword
attributeMap: pam:userpassword=*NULL*
attributeMap: pam:uid=msSFUName
attributeMap: group:userpassword=*NULL*
attributeMap: shadow:userpassword=*NULL*
attributeMap: shadow:uid=msSFUName
attributeMap: passwd:homedirectory=msSFUHomeDirectory
attributeMap: passwd:uid=msSFUName

search filters

- allows you to refine your search when looking up entries in the directory
- use to restrict access to systems

```
dn: cn=EngineerProfile,ou=ATC,dc=hp,dc=com
servicesearchdescriptor:passwd:ou=ATC,dc=hp,dc=com?sub? \
      (&(objectclass=posixAccount)(employeetype=engineer))
servicesearchdescriptor: pam:ou=ATC,dc=hp,dc=com?sub? \
      (&(objectclass=posixAccount)(employeetype=engineer))

# ./ldapsearch -b "ou=ATC,dc=hp,dc=com" uid=bsmith employeetype
dn: uid=BSmith,ou=People,ou=ATC,dc=hp,dc=com
employeetype: operator
# ./ldapsearch -b "ou=ATC,dc=hp,dc=com" uid=dougl employeetype
dn: uid=dougl,ou=People,ou=ATC,dc=hp,dc=com
employeetype: engineer
```

June 6, 2001 Page 38

Notes: In this example there is an additional search element placed on the search descriptor for passwd entry searches.
In order for a user to be returned from the directory they must also have the employeetype attribute set to “engineer”. This would prevent users that are not “engineers” from being able to login to the HP-UX system that uses this search filter in their profile.

configurable access

allows you to determine how ldap-ux will bind to the directory to lookup information

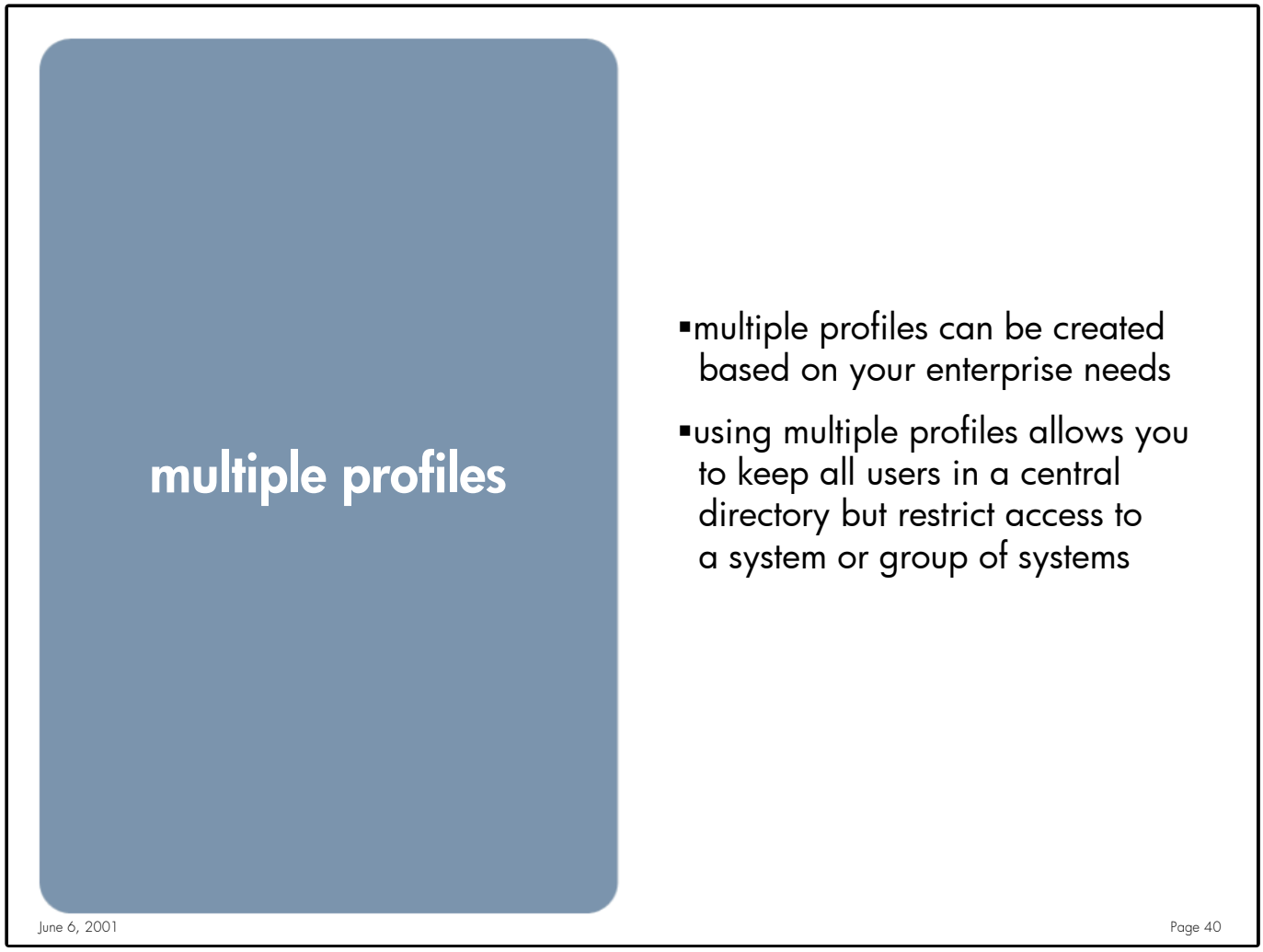
- anonymous
- proxy user

June 6, 2001

Page 39

Notes: Using a Proxy user allows you to set ACL's on user attributes. The Proxy users credentials are stored in kernel memory and on disk encrypted. By default you MUST use a proxy server to access information in the Active Directory

Caution: If the proxy users password/account expires then you will not be able to retrieve information for the Directory.



Notes: Using multiple profiles is one way to restrict which users can login to a set of systems, but can add to the administration overhead.

services for unix 2.0

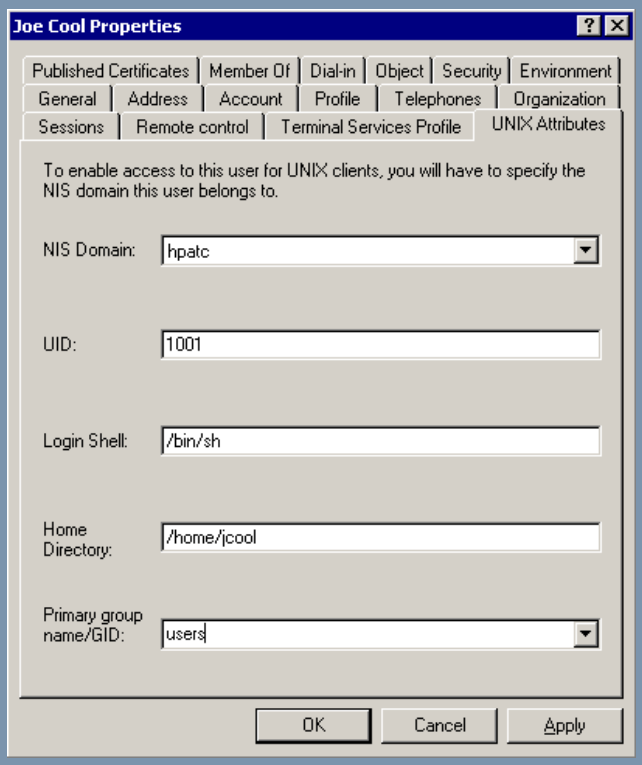
- provides several “unix” centric services for the windows 2000 platform
- extends the active directory’s schema to include unix attributes
- adds support for unix attribute management in the “active directory users and computers” tool

June 6, 2001 Page 41

Notes: Although SFU 2.0 provides many “Unix” services (NFS client/Server, **Server for NIS**, Posix shell, etc.) we only need SFU to extend Active Directory’s schema to include the Unix like (posixaccount) attributes. By installing SFU 2.0 an additional “Tab” is added to the “Active Directory Users and Computers” administration tool to manage these Unix attributes. SFU 2.0 does NOT use RFC 2307

Caution: Server for NIS is the only service that must be installed from SFU 2.0, however is not selected for installation by default, it MUST be manually selected during SFU 2.0 installation.

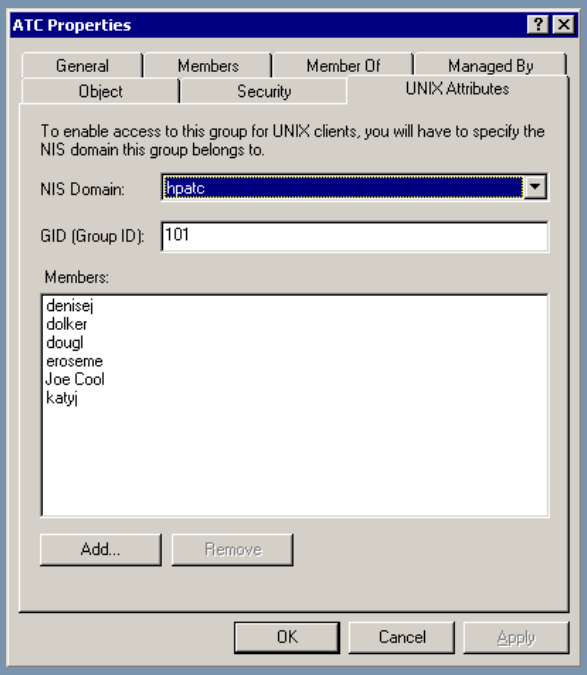
services for unix 2.0



June 6, 2001 Page 42

Notes: After installing SFU 2.0 on your Domain Controller you will have an additional tab "Unix Attributes" on both User and Group objects in the Active Directory Users and Computers utility. This slide shows the new tab of a User object. Here is where "Unix" attributes (such as uid number, login shell, home directory primary group) are set for a user

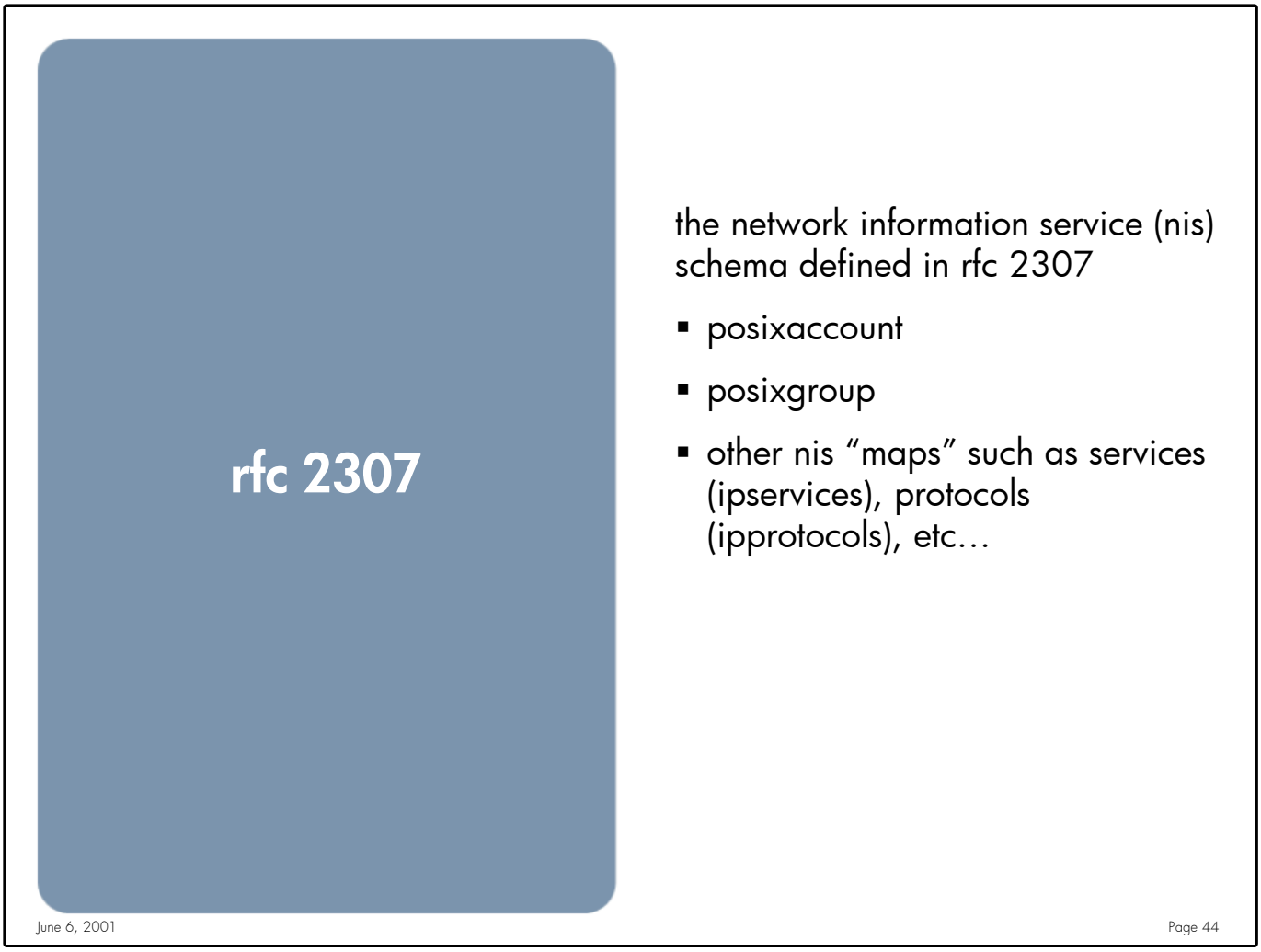
services for unix 2.0



June 6, 2001 Page 43

Notes: This slide shows the "Unix Attributes" Tab of a Group Object.

CAUTION: Because of the way SFU 2.0 stores members in a group object this tab can not be used to manage "Unix" group membership. See the slide "RFC 2307 vs. SFU 2.0 posixgroup" for more details



Notes: RFC 2307 defines the schema (objects and attributes) to map NIS entities into LDAP/X.500 objects. Currently the LDAP/UX integration product only supports Users (posixAccount) and Groups(posixGroup) in a Windows 2000 Active Directory.

rfc 2307 posixaccount

required attributes:

- cn *(common name)*
- uid *(user id, name)*
- uidnumber *(user id number)*
- gidnumber *(group id number)*
- homedirectory *(home directory)*

additional attributes:

- userpassword *(crypted password)*
- loginshell *(login shell)*
- gecos *(name, phone number, location)*
- description

June 6, 2001 Page 45

Notes:

rfc 2307 posixaccount

example in ldif *(ldap data interchange format)* format:

```
dn: uid=jdoe,ou=People,ou=ldap-ux,dc=acme,dc=com
uid: jdoe
cn: John Doe
objectclass: top
objectclass: account
objectclass: posixAccount
loginshell: /usr/bin/ksh
uidnumber: 223
gidnumber: 20
homedirectory: /home/jdoe
gecos: John Doe,,,
userpassword: {crypt}ask4kskHhFl=
```

June 6, 2001 Page 46

Notes:

required attributes:

- cn *(common name, group name)*
- gidnumber *(group id number)*

additional attributes:

- userpassword *(group password)*
- memberuid *(uid, name, of group members)*
- description

Notes:

rfc 2307 posixgroup

example in ldif *(ldap data interchange format)* format:

```
dn: cn=users,ou=Group,ou=ldap-ux,dc=acme,dc=com
objectclass: posixGroup
objectclass: top
cn: users
gidnumber: 20
memberuid: root
memberuid: jdoe
memberuid: jcool
```

June 6, 2001 Page 48

Notes:

rfc 2307 vs. sfu 2.0

posixaccount		
attribute	**rfc 2307**	**sfu 2.0**
user name	uid	mssfuname
user id number	uidnumber	uidnumber
primary group id number	gidnumber	gidnumber
users login directory	homedirectory	mssfuhomedirectory
users password	userpassword	mssfupassword
users login shell	loginshell	loginshell
users gecos information	gecos	gecos

June 6, 2001 Page 49

Notes:

rfc 2307 vs. sfu 2.0

posixgroup

attribute	rfc 2307	sfu 2.0
group name	cn	cn
group id number	gidnumber	gidnumber
group password	userpassword	-none-
group members	memberuid	posixmember memberuid*

June 6, 2001 Page 50

Notes: SFU 2.0 uses the posixmember attribute to signify group members. In this attribute sfu 2.0 uses the distinguished name of the user and not the uid (msSFUName) of the user. Since Unix commands and applications expect the members of a group to be listed by their uid ldap/ux (NSS_LDAP) had 2 choices:

Option 1: Take the DN listed as a member of the group and perform an additional ldapsearch to determine the uid (mssfuname) for each member
This option allows administrators to use the "Users and Computers" administration tool to manage Unix group membership. The major drawback to this option is that an additional ldapsearch will be preformed for every member of the group. This is not a problem if you have a small number of group members, but a major performance issue for larger groups.

Option 2: Use a different attribute (memberuid) to signify a member of the group
This option does not have the performance issue option 1 does since all of the attribute, including group members, can be retrieved in a single ldapsearch. The drawback to option 2 is that administrators can not use the "Users and Computers" administration tool to manage Unix group membership. To manage Unix group membership administrators must use the ADSI Edit utility on a Windows 2000 system, or the ldapmodify command on your HP-UX system.

Option 2 was selected because of the sever performance issues with option 1. When users and groups are migrated from /etc/passwd or NIS into AD the migration scripts use the memberuid attribute of the group object to show group membership. Both options may be supported in the future.

making it work

active directory

- plan your configuration carefully
- install sfu 2.0
- create a proxy user
- create a “host” user for all of your hp-ux systems
- create keytab file(s) for your hp-ux “host” accounts

June 6, 2001 Page 51

Notes:

- The most important step of deploying Active Directory is PLANING, PLANING, PLANING!
- SFU 2.0 is not a free product, it sells for around $150.00 US (per install?), check with your MS rep.
- Installing SFU enables several services by default. None of these services are needed, including Server for NIS, disable them except for the NIS Server. There is a bug that will cause the DC to abort when a user that has Unix attributes set changes their password, Hot fix from Microsoft will be available
- Active Directory does not allow unauthenticated users access to the attributes your HP-UX systems will need. A proxy user will be used by your HP-UX system to read information from Active Directory.
- A “host” account needs to be created for each HP-UX system.
- A keytab file is used for “servers” to read their key from in order to authenticate

hp-ux

- install ldap/ux integration product
- install pam_kerberos
- configure ldap/ux client services
  - run setup program
  - modify /etc/nsswitch.conf
  - download profile periodically (cron job)
- migrate users and groups into active directory
- configure pam_kerberos
  - modify /etc/pam.conf
  - create /etc/krb5.conf
  - install keytab file

June 6, 2001 Page 52

Notes: Available on the latest application CD's or from http://software.hp.com (free of charge)
•LDAP/UX (J4269AA) version B.01.20 and later
•PAM Kerberos (J5849AA) version B.11.00.11 (1.0) and later

There are step by step guides for installation and configuration for both integrating LDAP/UX with Active Directory and PAM Kerberos. Read BOTH before attempting the installation/configuration. All manuals and release notes can be found at http://docs.hp.com. You will also find very good white papers that talk about LDAP and Active Directory at: http://docs.hp.com/hpux/internet

The migration scripts will create an ldif file to be used by the ldapmodify program to add users and groups into AD. There will be several users, like root, that you don't want in AD. There are 2 options to exclude these accounts from being added to AD. :
•Modify the LDIF file after the migration script is run and BEFORE you execute the ldapmodify command
•Make a copy of the group and password files remove the users/groups you don't want to migrate and run the migration scripts against those files
I would recommend #2, an LDIF file can group large fast, making it more likely for errors to occur

Don't forget to remove the users and groups that have been migrated to AD from the local passwd/group file

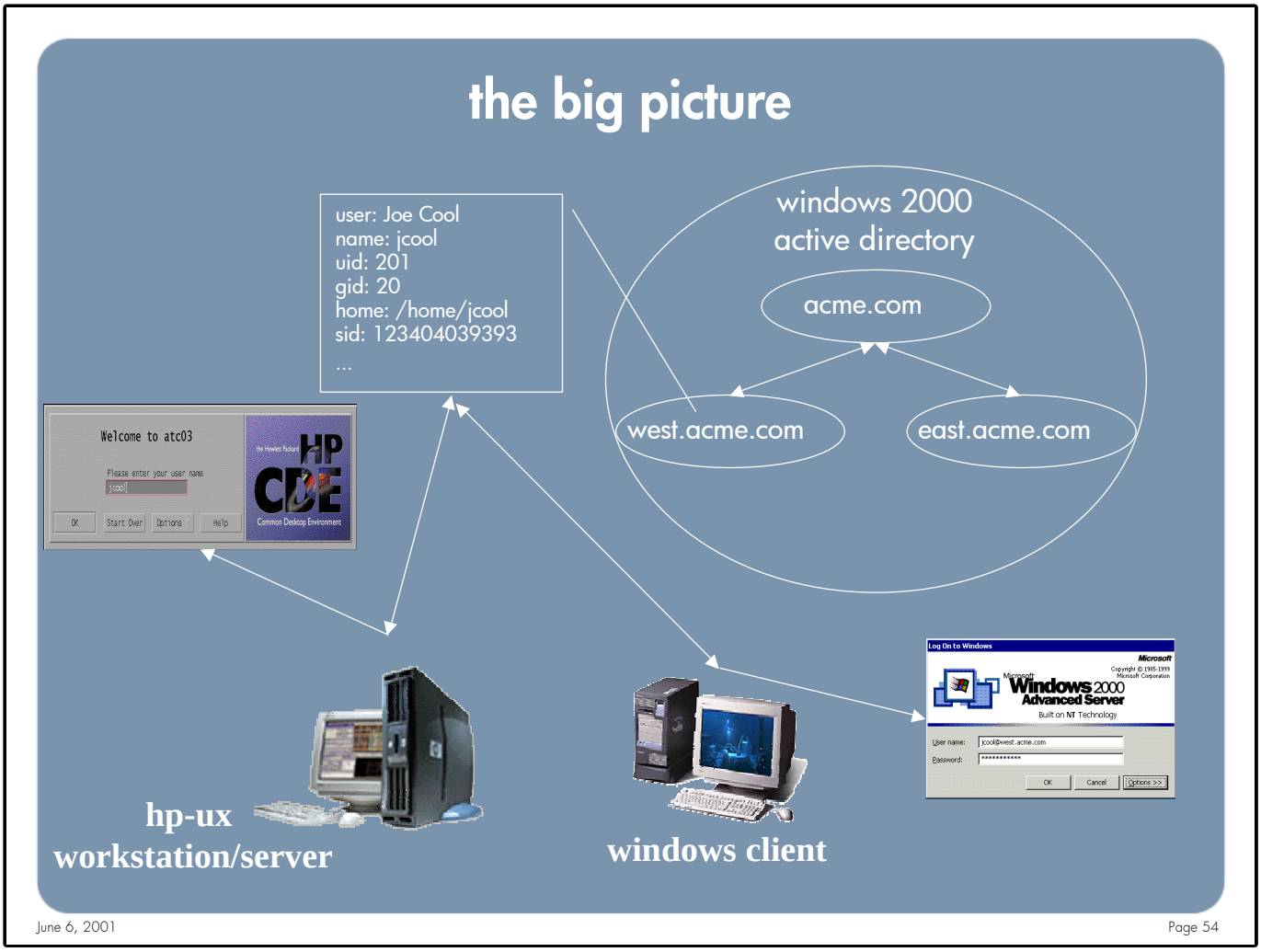
CAUTION: When users are added to AD their passwords are not set, and their accounts are not valid yet. An admin will need to set their accounts as valid, and set the passwords for each user.

agenda

- ✓ *problem description*
- ✓ *authentication and account management history*
- ✓ *a solution and technologies*
- ❑ **the "big" picture**
- ❑ other technologies
- ❑ limitations

June 6, 2001 Page 53

Notes:



Notes: This picture shows that the same user "Joe Cool" logging into his HP-UX workstation/server and Windows 2000 workstation using the same account, jcool, and password.

agenda

- ✓ *problem description*
- ✓ *authentication and account management history*
- ✓ *a solution and technologies*
- ✓ the "big" picture
- ❑ **other technologies**
- ❑ limitations

June 6, 2001 Page 55

Notes:

other technologies

- secure internet services
- cifs/9000 server
- webservers
  - apache
  - iplanet
  - zeus
- any “pamized” application

June 6, 2001

Page 56

Notes:

secure internet services (sis)

- a set of traditional *"kerberized"* internet services (both clients and servers) that include ftp, telnet & rlogin
- allows secure authentication to remote services without requiring passwords be sent across the network
- comes disabled on the system by default
- both clients and servers can use non-kerborized protocol
- best used in conjunction with pam_kerberos **not** instead of
- shares configuration with pam_kerberos

June 6, 2001 Page 57

Notes: SIS can be enabled on 11.0 and later system by executing:
/usr/sbin/inetsvcs_sec enable
Use /usr/sbin/inetsvcs_sec status To determine the current status

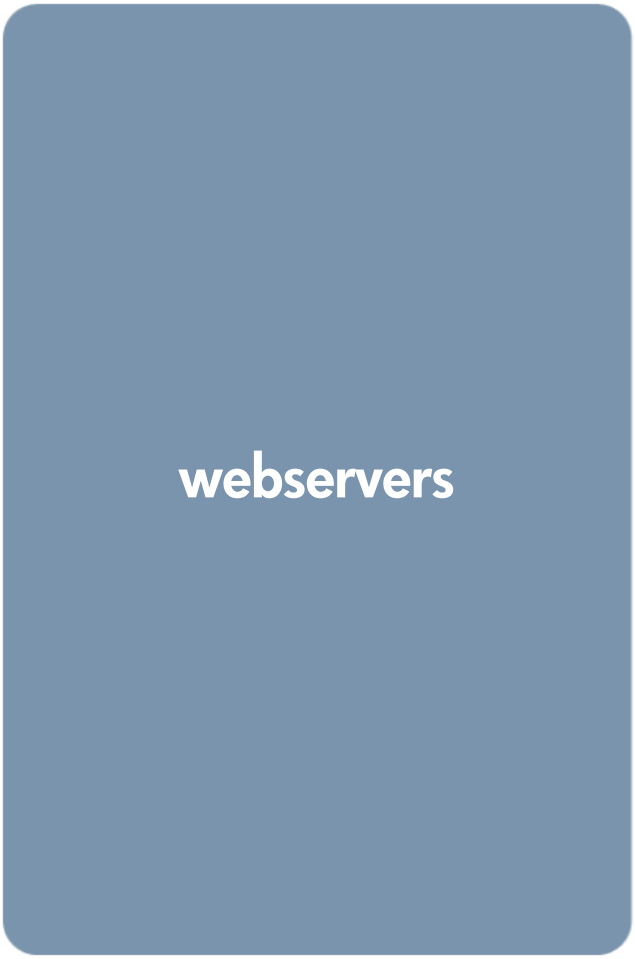
SIS requires that the /etc/krb5.conf file has been configured properly. For SIS Servers (rlogind, telnetd) to function properly there must be a "host" principal (host/<hostname>) configured in the KDC and a keytab file created on the local system. SIS clients can connect to "non-kerberized" servers by using the –P option of the client. By using both PAM_Kerberos and SIS users will only need to supply a password at the initial login to their HP-UX workstation to obtain a TGT, all authentication after that will be done securely. SIS and PAM Kerberos share the same configuration file, /etc/krb5.conf.

SIS clients and servers interoperate with MIT and Sun Kerberized clients and servers. (NOTE: patches are needed for both Sun and HP for complete interoperability. Patch Ids are not available at this time)

cifs/9000 (samba server)

- allows windows clients to "mount" network drives from an hp-ux server
- because unix accounts are stored in the active directory there is no need to map a unix account to a windows account *(user map file)*
- authentication is still done on the cifs server using ntlm passthrough
- for more a more detailed look at cifs/9000 and windows 2000 attend the:
  "cifs/9000 and windows 2000 interoperability"
  session on thursday at 4:30 pm

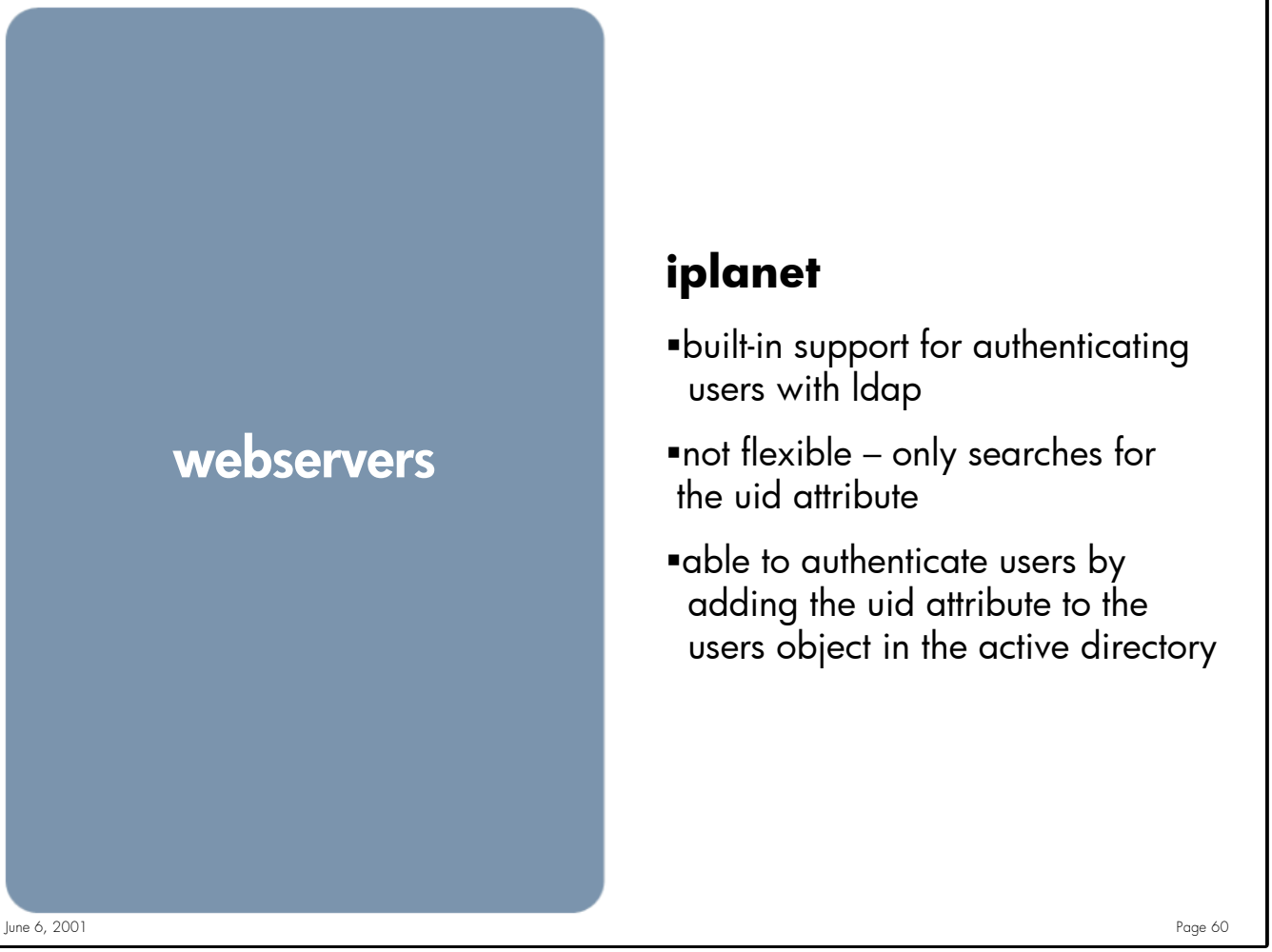
Notes:



apache

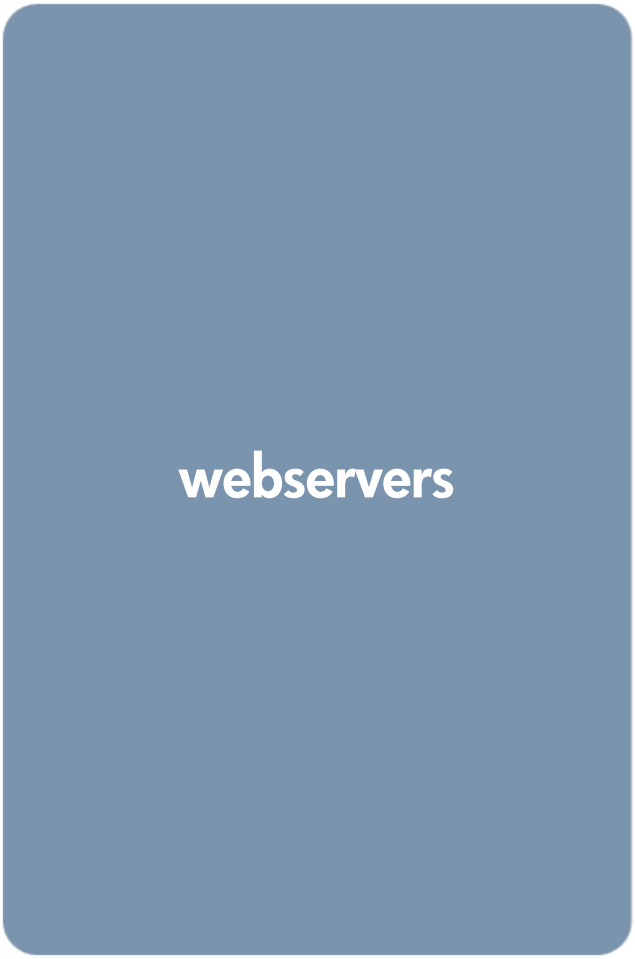
- using the auth_ldap module the apache webserver can authenticate users against the active directory
- configuration is flexible enough to map user name attributes
- auth_ldap module will ship with the apache release from hp in the future

Notes:



Notes: The iPlanet Webserver uses the uid attribute to look up a user. This works fine when using an iPlanet Directory Server, or RFC 2307 compliant Directory Server, but presents a problem for ADS.

There is no way to change/map the attribute that the webserver uses to lookup/authenticate a user.



zeus

- built-in support for authenticating users with ldap
- different approach then apache and iplanet
- webserver retrieves the users encrypted password to authenticate the user itself
- must have the mssfupassword attribute set and readable by a proxy user
- expired passwords/accounts are not enforced

Notes: Zeus takes a little different approach then iPlanet and Apache do. In the Zeus configuration you need to supply 2 LDAP URL's, one to retrieve the users password, and the other to retrieve a list of groups the user is a member of.
Zeus uses the crypted password from the directory to authenticate the user. Since the Webserver is doing the actual authentication it will not recognize an expired password or locked accounts.

It is possible to configure the Zeus Webserver to authenticate users using Active Directory, however the msSFUPassword attribute must be set.

pamized applications

any application that writes to the pam api can authenticate users against active directory

June 6, 2001

Page 62

Notes:

agenda

- ✓ *problem description*
- ✓ *authentication and account management history*
- ✓ *a solution and technologies*
- ✓ the “big” picture
- ✓ other technologies
- ❑ **limitations**

June 6, 2001 Page 63

Notes:

limitations

- single windows 2000 domain support
- user and group names limited to 8 characters
- only users and groups are supported by nss_ldap when using active directory
- no way to restrict logins

June 6, 2001 Page 64

Notes: Currently the LDAP/UX product only supports a single Windows 2000 domain. In Windows 2000 it is possible to have 2 users with the same user name in 2 domains (jcool@east.acme.com & jcool@west.acme.com) but HP-UX does not know how to differentiate the 2. Under investigation for future support.

HP-UX only supports user and group names up to 8 characters. When HP-UX supports greater the 8 character user and group names LDAP/UX will to.

Currently only the user and group "NIS" maps are supported by LDAP/UX (NSS_LDAP). Under investigation for more NIS Maps. (all standard NIS maps are supported using the iPlanet Directory Server v4 and later)

There is no easy way to restrict which users can login to which HP-UX systems. Because all systems are using the same "pool" of users and groups any user can login to any system with their account. It is possible to use search filters and different profiles to restrict access.

limitations

- users are not notified when their password must be changed
- no support for nested groups
- can't use "users and computers" application to manage unix groups

June 6, 2001 Page 65

Notes: In Windows 2000 groups can be members of groups (nested groups). This is not supported in HP-UX.

ldap rfc's

- 2307 An Approach for Using LDAP as a Network Information Service
- 2251 Lightweight Directory Access Protocol (v3)
- 2252 LDAP(v3) Attribute Syntax Definitions
- 2253 LDAP(v3) UTF-8 Rep of Distinguished Names
- 2254 String Representation of LDAP Search Filters
- 2255 The LDAP URL Format
- 2256 Sum. of X.500(96) User Schema for LDAPv3
- 2829 Authentication Methods for LDAP
- 2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security
- INTERNET-DRAFT LDAP-BIS

June 6, 2001 Page 66

Notes: IETF Draft LDAP-BIS (http://www.ietf.org/internet-drafts/draft-ietf-ldapbis-ldapv3-ts-00.txt) defines the set of RFCs comprising LDAPv3. This draft also addresses the "IESG Note" attached to RFCs 2251 through 2256 which discouraging implementation and deployment of LDAPv3 clients or servers implementing update functionality until a Proposed Standard for mandatory authentication in LDAPv3 is published.

kerberos rfc's

- 1510 The Kerberos Network Authentication Service (V5)
- 1964 The Kerberos Version 5 GSS-API Mechanism
- 2623 NFS Version 2 and Version 3 Security Issues and the NFS Protocol's Use of RPCSEC_GSS and Kerberos V5
- 2942 Telnet Authentication: Kerberos Version 5

Notes: