Using CA eTrust Top Secret to authenticate users on zLinux

James Chaplin
Systems Programmer z/OS, z/VM & z/Linux
US Customs and Border Protection

Wednesday, August 10, 2011: 3:00 PM-4:00 PM
Session Number 9870

Agenda

- Need for centralized authentication
  - Why we choose to use CA ESM over other options
- What is CA **E**xternal **S**ecurity **M**anager (ESM)
  - Key Components of CA ESM client
  - CA **D**istributed **S**ecurity **I**nterface (DSI)
- The Install process of CA ESM client on each zLinux guest
- How PAM & CA ESM Client work
- Commands and Keywords in Top Secret (or ACF2)
- Known Issues & Troubleshooting

Centralized Authentication

- Centrally manage users
- Centrally manage systems & user access to them
- Keep same security policy across all servers
- Centrally tracking and audit of user logon violations
- Allow users to simplify the management process of the user's password
- Standardized UID & GID across multiple servers

Why CBP choose to use CA ESM

- CA Top Secret is an established security package on z/OS at our site
- Easy deployment
  - Took only a few hours to install and start testing
- Security Team using known TSS Commands to define Linux users and systems in Top Secret
- Allowed us to move the task of defining users to the legacy in-house user vetting process
  - Got me out of the process!
- Tried to install and test using (IBM) LDAP (failed)
  - Problems with LDAP and Top Secret (more effort)
    - Could not get password changes from LDAP to Top Secret
    - Administration of LDAP values still resided with System Admins, not Security

What is CA External Security Manager

- CA ESM Client allows existing CA ACF2 and CA Top Secret users to maximize their investment by extending their existing security implementation to Linux for zSeries.
- CA ESM Client uses PAM (Pluggable Authentication Modules) to authenticate users on UNIX and Linux systems with existing (password) values set in either CA Top Secret or CA ACF2.
- Allows a data center to manage, reuse and extend their existing z/OS security to the Linux for zSeries platform.
  - Existing z/OS security package as the authentication server for one or more Linux systems
  - Eliminating the need for redundant security administration to define users on a system-by-system basis.

What is CA ESM (continue)

- User administration is reduced by using existing user ID and passwords on another platform.
- Control who can log on, where they can log on to, and when they can log on.
- Reusing your existing security database also reduces the number of passwords that a user must maintain.
- This reuse also includes the password controls such as minimum length, history, minimum number of days before a change, and so on.
- There were no functional changes or patches on the z/Linux side from releases 14 and 15.

Disclaimer

CA External Security Manager 14.0/15.0

This is free software. You may redistribute copies of it under the terms of

the GNU Lesser General Public License <http://www.gnu.org/licenses/lgpl.html>.

There is NO WARRANTY, to the extent permitted by law.

Key Components of using CA ESM client

- CA Distributed Security Interface (CA DSI)
  - Comes as a part of CA Top Secret or ACF2 for zOS
- CA ESM proxy server (daemon)
- CA ESM PAM modules
  - pam_CA_esm.so
    - Authenticates users & interfaces with Linux PAM facility
  - libnss_CA_esm.so.2
    - Obtains user and group attributes & interfaces with Linux NSS facility
- Top Secret or ACF2
  - Add a LINUXNAM to and existing Top Secret ACID
  - Use the LINUXUID, LINUXHOM, and LINUXPGM attributes as LNXENTS as part of the LINUX segment and LINUXGID attribute from the Group definition record

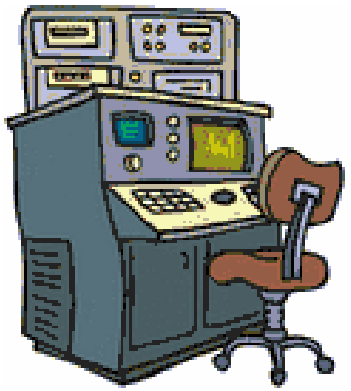
CA Distributed Security Interface (CA DSI)

- The CA Distributed Security Integration for z/OS (CA DSI Server) provides a remotely callable interface that uses TCP/IP to allow applications anywhere within the enterprise to communicate with the mainframe ESMs.
- The CA DSI Software Development Kits (SDKs) allow applications from MS Windows, IBM AIX and USS, Sun Solaris SPARC and x86, HP-UX PA-RISC and Itanium, Linux/Intel, and Linux for zSeries, to communicate to any CA ACF2 or CA Top Secret system in the enterprise.
- To help ensure secure connections between these z/OS-based servers and the application using it, SSL connections are used.

CA Distributed Security Interface (CA DSI)

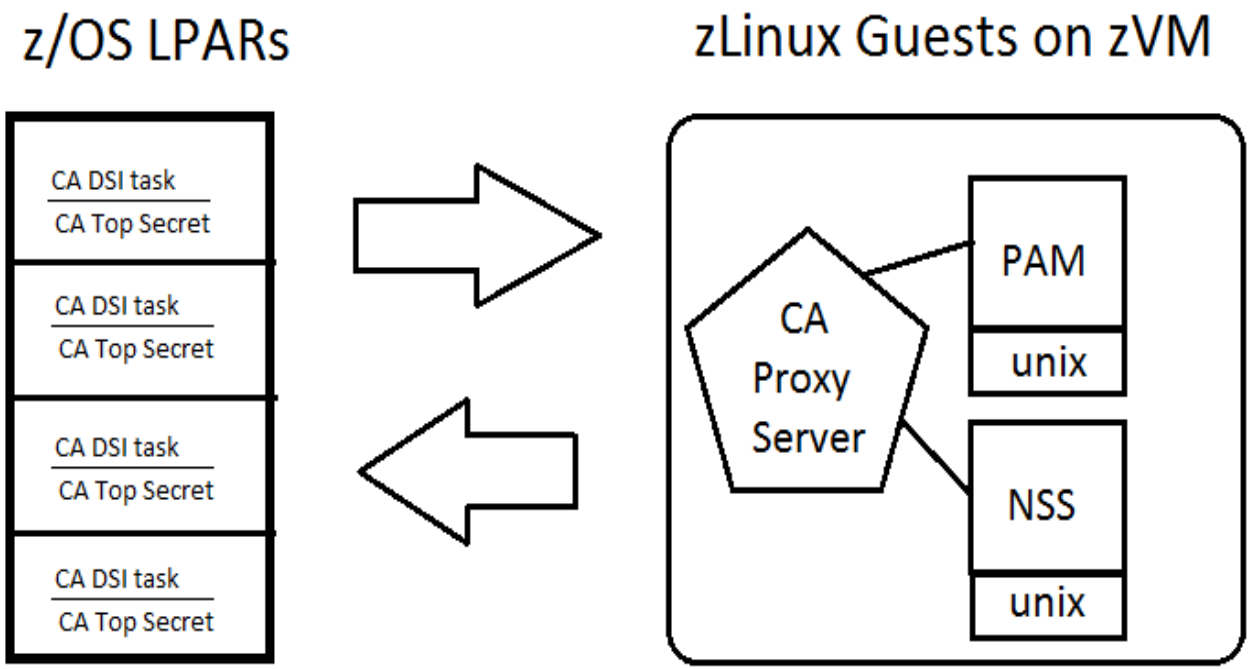
- With Release 14 of Top Secret, CA has merged the previous CA PAM Server functionality in to the CA DSI Server
- CA ACF2 and CA Top Secret act as the authentication server for one or more Linux systems from multiple LPAR locations
- CA DSI Server reduces the workload for user administration
- By using CA DSI Server, you can reuse all of your existing z/OS user ID and password security controls on your Linux for zSeries systems.
  - Minimum password length
  - History and violation tracking
  - Set minimum number of days before a change
  - Enforce additional password rules/requirements
  - Bypass need for **`/etc/login.defs`** values or setting options with **`pam_cracklib.so`** in pam.d directory

Installation of CA DSI & settings



- Now part of the CALDAP install process
- SMP/E or MSM based install
- Setup started task and security rules
- Modify the configuration file
  - Set port number used by the PAM Proxy client
  - Set debug/loglevel and log file location/name
  - *userid mixed* ← receive ID in any U/L case
  - *lowercase username* ← return Linux user name
  - TLS settings

Overview of how CA Pam Client interfaces with z/OS DSI task



CA PAM

- **pam_CA_esm.so**

Unix PAM

- **pam_unix.so**
- **/etc/passwd**

CA NSS

- **libnss_CA_esm.so**

Unix NSS

- **/etc/nsswitch.conf**

The Install process of CA ESM client on each zLinux guest

- The Client install files comes with CA Top Secret (or ACF2)
  - Is included as a part of the package for CA DSI on z/OS
- Requires Linux for zSeries Red Hat AS3 or SuSE ES8 and above
- Use rpm file for either 32 or 64 bit (s390 vs. s390x)
  - Two rpms, program and libraries
    - pam_ca_esm**.s390*.rpm
    - pam_ca_esm-libs**.s390*.rpm
  - Easiest to install and upgrade
- Tar file (binaries) install available for 32 or 64 bit
- Source Code to compile and install is also available

Steps to implement CA ESM Client

- Modify the common ESM configuration file
  - Located at /etc/CA_esm.conf
    - 4 basic parts
- Start the ESM proxy server
  - /sbin/service Caesm start
    - Set in chkconfig
- Enable the NSS module
  - /etc/nsswitch.conf
- Enable the PAM module
  - In /etc/pam.d/ directory

Modify the CA ESM configuration file

- Set Debug level
  - *Warning (debug can be set in two locations)*
- Define Host Systems on z/OS LPARs (DSI server location)
  - *Assign IP or DNS and port number (default 1091)*
  - *Select failover or round robin mode*
- Configure Transport Layer Security (TLS)
- Other options specific to PAM Components
  - *i.e. ignore-. . . . & min uid/gid option*
- Reference & Check:
  - man CA_esm.conf – display rules and syntax
  - CA_esm_showcfg – will checks and displays values
    - *Located at /opt/CA/PAMclient/sbin/CA_esm_showcfg*

Detail settings in /etc/CA_esm.conf

```
debug-level 0
log-facility AUTHPRIV
proxy-socket /var/run/CA_esm_proxy.socket
--------
esm-host 'DNSname' timeout=10 tls=yes
esm-host 10.333.143.23 timeout=10 tls=yes
esm-host . . . . . (set as many as you
  like)
esm-mode failover attempts=0 delay=5
esm-port 1230
--------
```

(one of two locations to set debug)

<-- Logging location

Each setting has a description in the configuration file explaining there default and usage/role

With **esm-mode**, set the **attempts** value to zero, then the server will keep trying to reconnect when a connection is lost

Settings in `/etc/CA_esm.conf` (cont.)

```
tls-supported yes
tls-required yes
tls-cacertfile /opt/CA/PAMclient/YRsiteKey.pem
tls-checkpeer off
tls-ciphers DEFAULT
--------
ignore-root yes
ignore-authinfo-unavail no
ignore-unknown-user no
min-uid 0
min-gid 0
--------
```

Values for tls-ciphers can be viewed by command:
openssl ciphers -tls1 –v 'DEFAULT‘

We set `ignore-root` to yes, the default is no

Transport Layer Security (TLS)

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide communication security over the Internet.

A TLS client and server negotiate a stateful connection by using a handshaking procedure. CA ESM client uses TLS with Transmission Control Protocol (TCP) protocol in this exchange.

Steps of a TLS secure connection

- The handshake begins when a client connects to a TLS-enabled server, presents a list of supported CipherSuites (ciphers and hash functions).
- The server picks the strongest cipher and hash function that it also supports and notifies the client of the decision.
- The server sends back its identification in the form of a digital certificate. The certificate usually contains the server name, the trusted certificate authority (CA) and the server's public encryption key.
- The client may contact the server that issued the certificate (the trusted CA as above) and confirm the validity of the certificate before proceeding.
- Generate the session keys used for the secure connection, the client encrypts a random number with the server's public key and sends the result to the server. Only the server should be able to decrypt it, with its private key.
- Both parties generate key material for encryption and decryption. This concludes the handshake and begins the secured connection, which is encrypted and decrypted with the key material until the connection closes.

Starting the CA ESM proxy server

- Command line:
  - CA_esm_proxy --runas-daemon --trace-file=<pathname>
- Startup file
  - /etc/init.d/Caesm
- Using the service command:
  - /sbin/service CAesm start
- Setup CAesm in chkconfig
  - chkconfig –add CAesm

Enable the NSS module

- NSS is set of libraries, APIs, utilities, and documentation designed to support cross-platform development of security-enabled client and server applications
- It provides a complete open-source implementation of the crypto libraries
- To enable the NSS module
  - Add the following changes in /etc/nsswitch.conf
    - passwd: files CA_esm
    - group: files CA_esm
      - *(do not need to change shadow)*
      - *Order determine which it checks first*
- Recycle nscd:
  - service nscd restart

Enable the PAM module

- Located in **/etc/pam.d/** directory
- Samples located:
  - **`/usr/share/doc/pam_CA_esm-14.0/examples/RedHat/pam.d/`**
  - **`/usr/share/doc/pam_CA_esm-14.0/examples/SuSE/pam.d/`**
  - Includes **`caesm-auth`**, **`login`**, **`passwd`** & **`sshd`** modules
- Issues with use of **`service=caesm-auth`** & **`debug`**
- For each PAM module type (auth, account, password, session)
  - Changed the control method from "**`required`**" to "**`include`**"
  - Changed the module and service arguments from "**`pam_stack.so service=caesm-auth`**" to "**`caesm-auth`**"
- We also modified **`/etc/pam.d/sudo`**

Sample of caesm-auth

```
cat /etc/pam.d/caesm-auth
#%PAM-1.0
auth       required     pam_env.so
auth       sufficient   pam_unix.so likeauth nullok
auth       sufficient   pam_CA_esm.so likeauth
auth       required     pam_deny.so

account    required     pam_unix.so
account    required     pam_CA_esm.so

password   required     pam_cracklib.so retry=3 type=
password   sufficient   pam_unix.so nullok use_authtok md5 shadow
password   sufficient   pam_CA_esm.so try_first_pass use_authtok
password   required     pam_deny.so

session    required     pam_limits.so
session    required     pam_unix.so
session    required     pam_CA_esm.so
```

Sample of sshd

```
cat /etc/pam.d/sshd

#%PAM-1.0
auth       include      caesm-auth
auth       required     pam_nologin.so
account    include      caesm-auth
password   include      caesm-auth
session    include      caesm-auth
session    required     pam_limits.so
session    optional     pam_console.so
session    optional     pam_mkhomedir.so skel=/etc/skel umask=0022
```

Sample of login

```
cat /etc/pam.d/login
#%PAM-1.0
auth       required     pam_securetty.so
auth       include      caesm-auth
auth       required     pam_nologin.so
account    include      caesm-auth
password   include      caesm-auth
session    include      caesm-auth
session    optional     pam_console.so
session    optional     pam_mkhomedir.so skel=/etc/skel umask=0022
```

Four CA ESM Utilities available

- CA_esm_ctrl
  - Send COMMAND to the CA External Security Manager proxy server
    - debug-level=num
    - reload
    - restart
    - shutdown
    - statistics
    - trace-level=num
- CA_esm_proxy
  - Start the proxy server for the CA External Security Manager.
- CA_esm_showcfg
  - Print the contents of the ESM configuration file.
- CA_esm_stash
  - Encrypt and store PASSWORD in FILE.

CA ESM Utilities output sample

#**CA_esm_showcfg**

```
ESM configuration data:
        codeset              <default>
        client_threads         10
        client_timeout         0 5
        debug_level            0x0000FFFF
        num_hostdefs             3
        mode                 failover
        attempts               0
        delay                5
        port                 1230
        log-facility          LOCAL1
        min-gid               0
        min-uid               0
        ignore-root            on
        ignore-authinfo-unavail off
        ignore-unknown-user     off
        proxy-socket           /var/run/CA_esm_proxy.socket
        tls-required           on
        tls-supported           on
        tls-cacertdir          <none>
        tls-cacertfile         /opt/CA/PAMclient/dhscatre.pem
        num_certdefs             0
        tls-checkpeer          off
        tls-ciphers            DEFAULT

Host definition:
        nodename       MydnsName
        port             1230
        cert             <default>
        codeset             <default>
        timeout            10
        tls              on
   . . . . . .
```

#**CA_esm_ctrl statistics**

Number of hosts defined: 3

Host 0: node MydnsName, port 1230, IP address 12.345.6.131
- connection status: **active**
- number of requests sent: 72
- number of responses received: 72
- shortest response time: 837
- longest response time: 120718
- average response time: 3122

Host 1: node MydnsName2, port 1230, IP address
- connection status: **inactive**

. . . . . .

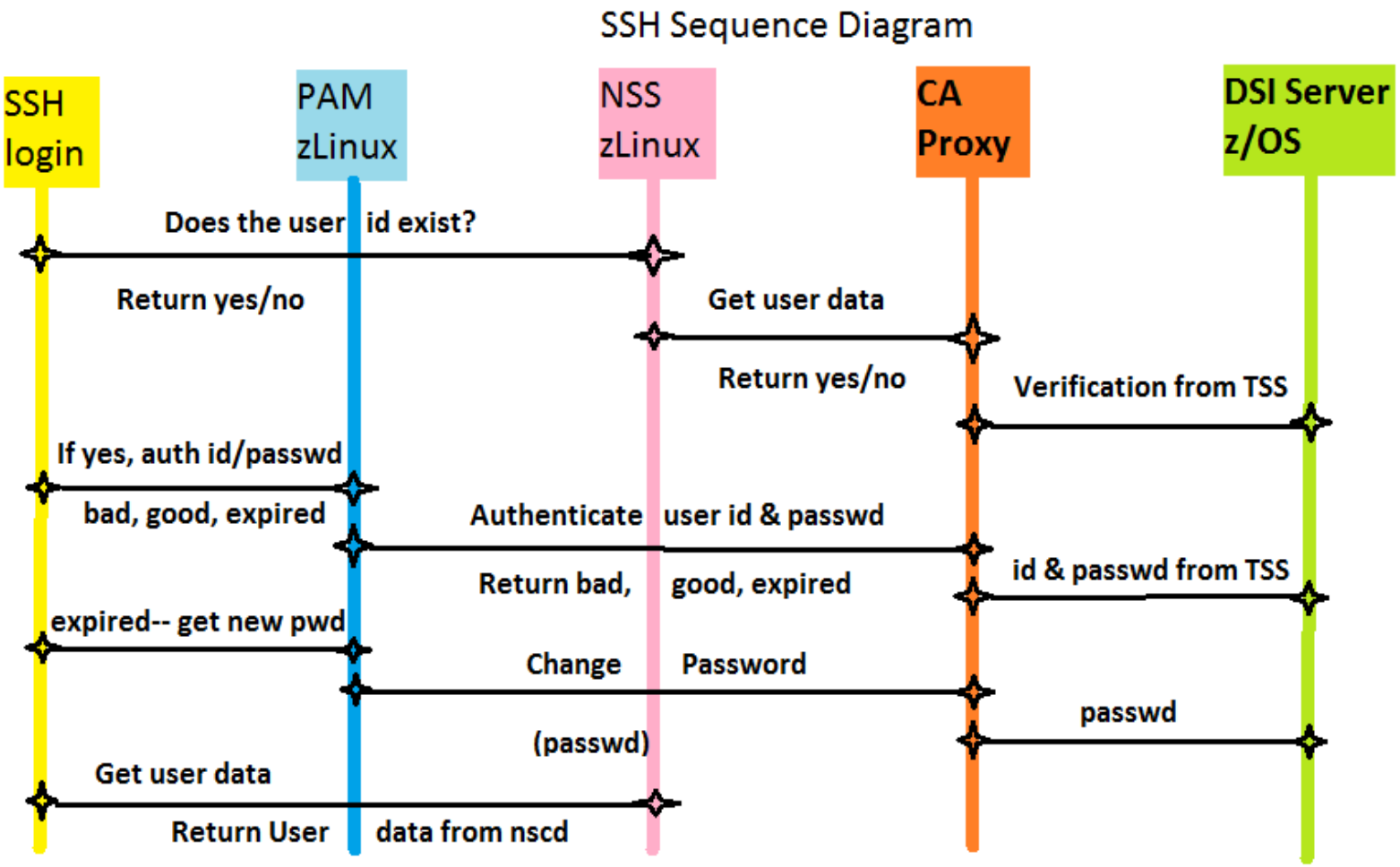
Debug settings and CA ESM Client

- When debug is set, output to the log file will include user passwords ☹ , but file is only readable by root.
- Value set in **`/etc/CA_esm.conf`** or dynamically by issuing the command:
  `CA_esm_ctrl --debug-level=`*`nnnn`*
- In the **`caesm-auth`** member located in **`/etc/pam.d/`**, if you place the word debug after the auth line for pam_CA_esm.so, this will put the client into a debug mode, writing more output to /var/log/caesm.log.

  Example:

auth sufficient pam_CA_esm.so try_first_pass likeauth debug debug debug

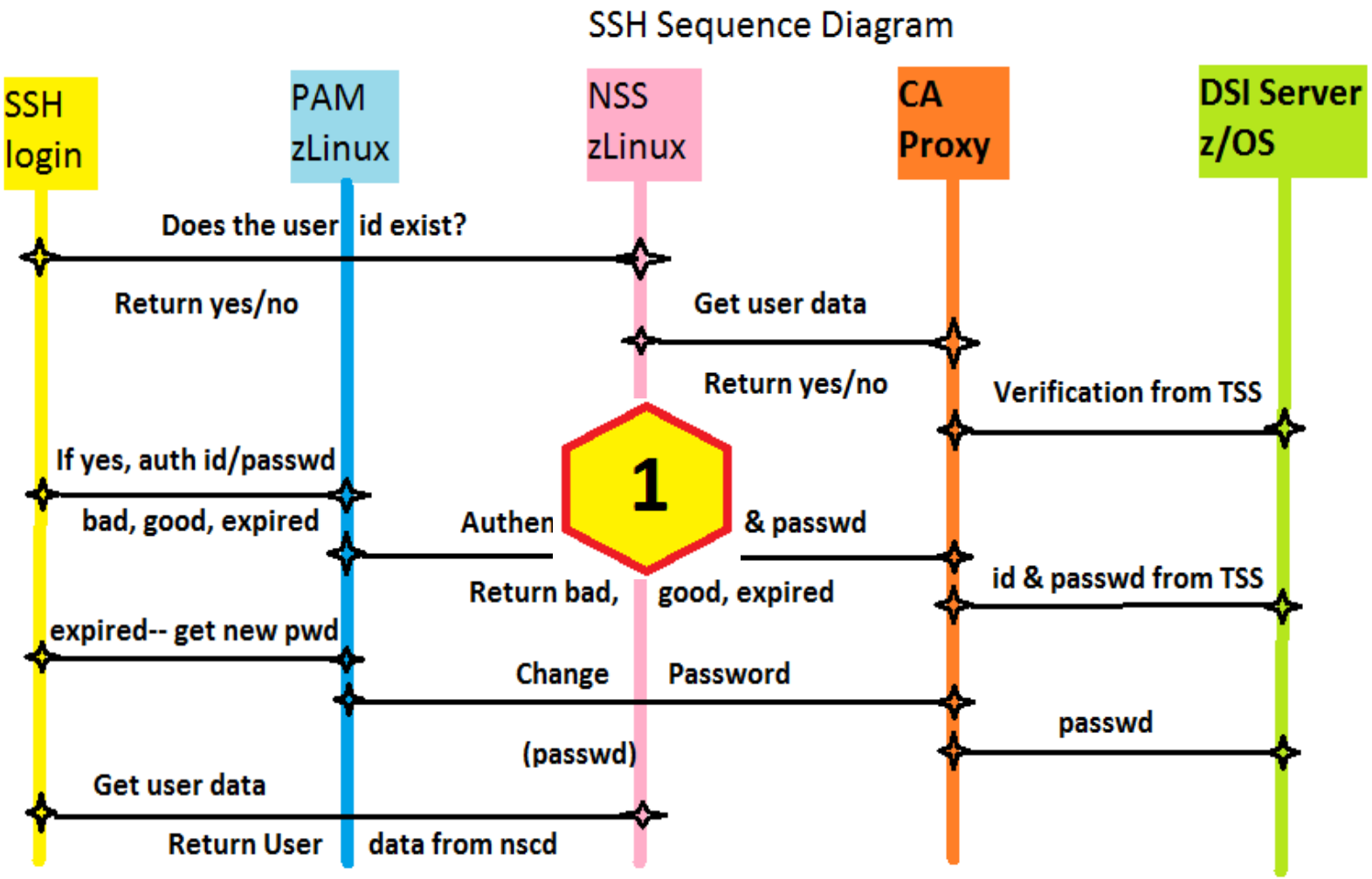
SSH Login process (example)



Sample SSH login process (debug on) - 1

Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]:
CHECKAUTH2,sshd,testlx1,"XXXX",ssh,w12345.customs.treas.gov,*,*,2011-07-15,14:17:50,testlx1
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Sent status to client: ESM_ERR_SUCCESS
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Sent response to client
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: 0
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: __CLIENTQUIT__
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Received close request from client
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Closing client socket
Jul 15 14:17:50 ztn004 nscd: _nss_CA_esm_initgroups_dyn: entering
Jul 15 14:17:50 ztn004 nscd: _nss_CA_esm_initgroups_dyn: user=testlx1, group=4294967295
Jul 15 14:17:50 ztn004 nscd: _nss_CA_esm_connect_send: entering
Jul 15 14:17:50 ztn004 nscd: _nss_CA_esm_connect_send: command=INITGROUPS,testlx1
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: INITGROUPS,testlx1
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Sent status to client: ESM_ERR_SUCCESS
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Sent response to client
Jul 15 14:17:50 ztn004 nscd: _nss_CA_esm_connect_send: leaving, err=0
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: 0,1,100
Jul 15 14:17:50 ztn004 nscd: _nss_CA_esm_initgroups_dyn: leaving, status=1
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: __CLIENTQUIT__
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Received close request from client
Jul 15 14:17:50 ztn004 CA_esm_proxy[1395]: Closing client socket

Logon – getpwbyname -- 1

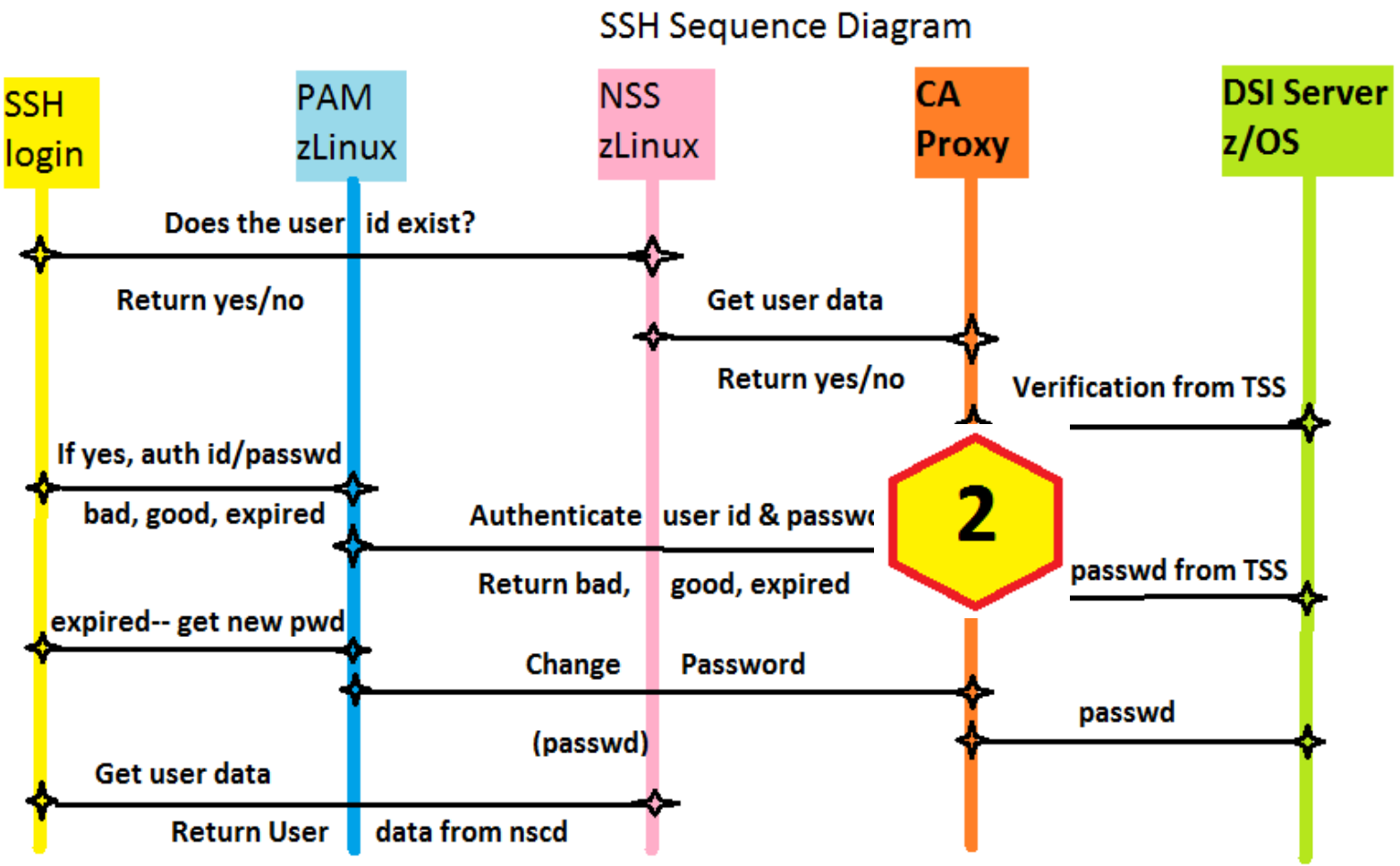


Sample SSH login process (debug on) - 2

nscd updating cache

Jul 15 14:20:02 ztn004 nscd: _nss_CA_esm_getpwnam_r: entering
Jul 15 14:20:02 ztn004 nscd: _nss_CA_esm_getpwnam_r: name=testlx1
Jul 15 14:20:02 ztn004 nscd: _nss_CA_esm_connect_send: entering
Jul 15 14:20:02 ztn004 nscd: _nss_CA_esm_connect_send: command=GETPWBYNAM,testlx1
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: GETPWBYNAM,testlx1
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: Sent status to client: ESM_ERR_SUCCESS
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: Sent response to client
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: 0,"5555","100","testlx1","/usr/local/home/testlx1","/bin/bash"
Jul 15 14:20:02 ztn004 nscd: _nss_CA_esm_connect_send: leaving, err=0
Jul 15 14:20:02 ztn004 nscd: _nss_CA_esm_getpwnam_r: leaving, status=1
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: __CLIENTQUIT__
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: Received close request from client
Jul 15 14:20:02 ztn004 CA_esm_proxy[1395]: Closing client socket

Return GETPWBYNAM -- 2



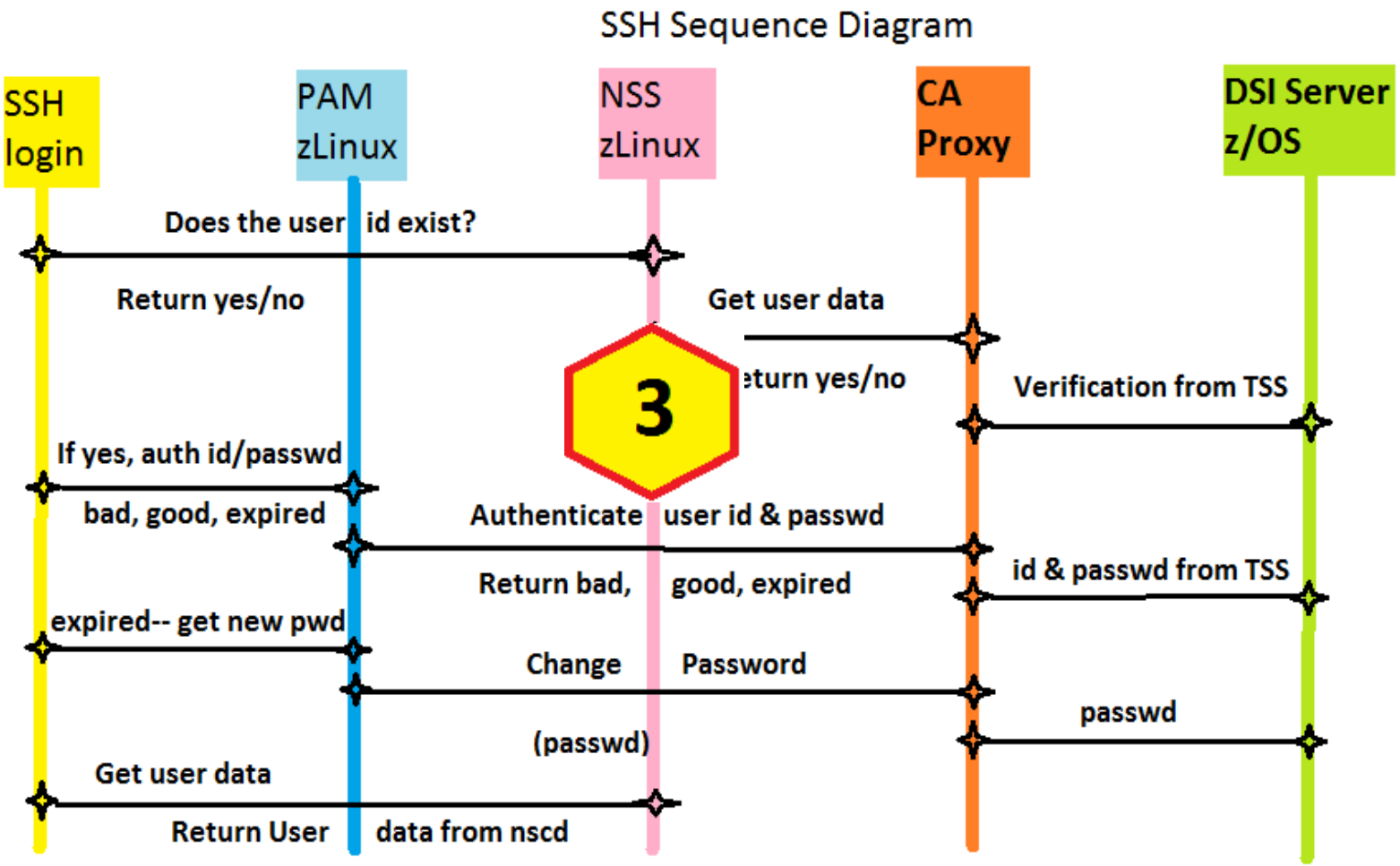
Sample SSH login process (failed logon) - 3

Failed logon due to incorrect/missing Facility in user definition

---

Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]:
CHECKAUTH2,sshd,testlx1,"XXXX",ssh,w12345.customs.treas.gov,*,*,2011-07-15,14:10:00,testlx1
Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]: Sent status to client: ESM_ERR_SUCCESS
Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]: Sent response to client
Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]: 1,2,"TSS7100E 002 J=PAM A=TESTLX1 T=012A460D
F=ZLINUX - Failed by Site Exit ","**TSS7175E Initiation Denied by Site Security Exit** "
Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]: __CLIENTQUIT__
Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]: Received close request from client
Jul 15 14:10:00 ztn004 CA_esm_proxy[1395]: Closing client socket
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]:
CHECKAUTH2,sshd,testlx1,"xxxx",ssh,w12345.customs.treas.gov,*,*,2011-07-15,14:10:23,testlx1
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]: Sent status to client: ESM_ERR_SUCCESS
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]: Sent response to client
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]: 3,2,"**TSS7100E** 009 J=PAM A=TESTLX1 T=012A460D
F=ZLINUX - Incorrect Password ","**TSS7101E Password is Incorrect**"
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]: __CLIENTQUIT__
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]: Received close request from client
Jul 15 14:10:23 ztn004 CA_esm_proxy[1395]: Closing client socket

Two examples of failed logins -- 3

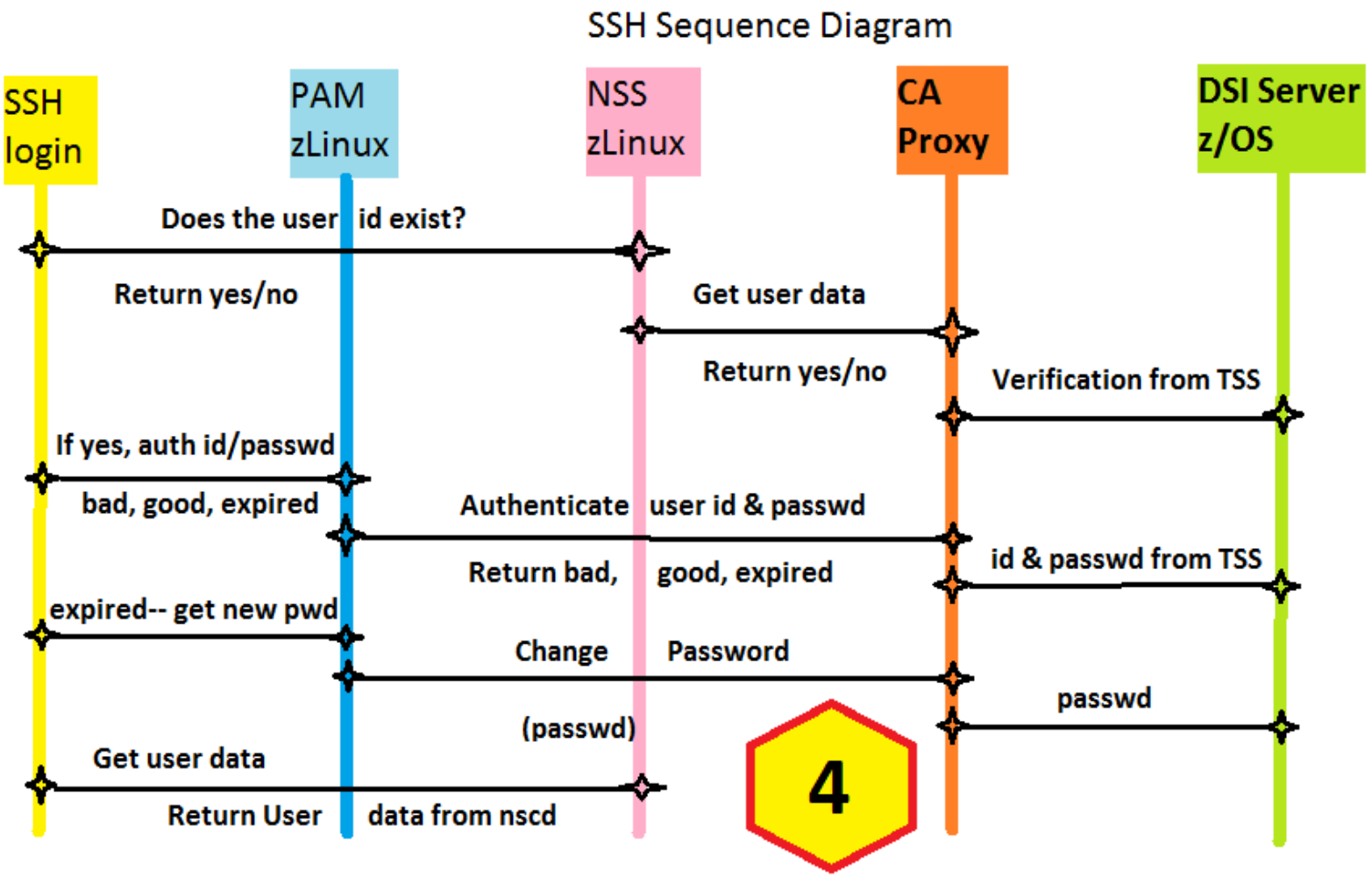


Sample SSH session (changing password) - 4

Log output with password change

Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: CHECKAUTH2,**passwd,**testlx1,"XXXX",pts/3,*,*,*,2011-07-15,14:20:47,testlx1
Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: Sent status to client: ESM_ERR_SUCCESS
Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: Sent response to client
Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: 0
Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: __CLIENTQUIT__
Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: Received close request from client
Jul 15 14:20:47 ztn004 CA_esm_proxy[1395]: Closing client socket
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: **CHANGEPW,passwd,testlx1,"XXXX","*******"**
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: Sent status to client: ESM_ERR_SUCCESS
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: Sent response to client
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: 0,1,"**TSS7030I Password Changed**"
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: Received request from client
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: __CLIENTQUIT__
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: Received close request from client
Jul 15 14:21:05 ztn004 CA_esm_proxy[1395]: Closing client socket

Changing password -- 4



Sample SSH login process (other files --/var/log/secure)

Failed & incorrect password:
Jul 15 14:10:00 ztn004 sshd[20395]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=w12345.customs.treas.gov user=testlx1
Jul 15 14:10:26 ztn004 sshd[20395]: Failed password for testlx1 from 10.159.234.168 port 1471 ssh2
Jul 15 14:10:37 ztn004 sshd[20396]: Received disconnect from 10.159.234.168: 13: Unable to authenticate
Jul 15 14:10:37 ztn004 sshd[20395]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=w12345.customs.treas.gov user=testlx1

Successful Login (sshd)
Jul 15 14:17:50 ztn004 sshd[27747]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=w12345.customs.treas.gov user=testlx1
Jul 15 14:17:50 ztn004 sshd[27747]: Accepted password for testlx1 from 10.159.234.168 port 1666 ssh2
Jul 15 14:17:50 ztn004 sshd[27747]: pam_unix(sshd:session): session opened for user testlx1 by (uid=0)

Jul 15 14:20:42 ztn004 passwd: pam_unix(passwd:chauthtok): user "testlx1" does not exist in /etc/passwd

Sample user view of SSH login process

----------start a ssh session (with a first time user)----------
login as: testlx1
You are accessing a U.S. Government information system, ...
testlx1@ztn004's password:
Creating directory '/usr/home/testlx1'.
testlx1@ztn004:(/usr/home/testlx1)#**ls -al**
total 20
drwxr-xr-x 2 testlx1 users 4096 Jul 15 14:17 .
drwxr-xr-x 33 root   root 4096 Jul 15 14:17 ..
-rw-r--r-- 1 testlx1 users   33 Jul 15 14:17 .bash_logout
-rw-r--r-- 1 testlx1 users 176 Jul 15 14:17 .bash_profile
-rw-r--r-- 1 testlx1 users 124 Jul 15 14:17 .bashrc
testlx1@ztn004:(/usr/home/testlx1)#**passwd**
Changing password for user testlx1.
(current) CA ESM Password:
New password:
Retype new password:
TSS7030I Password Changed
Password changed
passwd: all authentication tokens updated successfully.
testlx1@ztn004:(/usr/home/testlx1)#

Top Secret Keywords

- **LINUXNODE**
  - Modifies zLinux hosts to the TSS NDT
- **LINUXNAM**
  - Sets the user's zLinux ID to an established ACID
- **LNXENTS**
  - Assigns user values and facilities to an ACID
- **Linux Segment Attributes (set by LNXENTS):**
  - **LINUXUID**
    - Should be unique for each user on a given Linux system.
  - **LINUXGID**
    - Should represent a valid Linux group.
  - **LINUXHOM**
    - Directory should be valid for your Linux file system.
  - **LINUXPGM**
    - Value must specify a valid Linux shell.

Steps to define with Top Secret

1. Define the Facilities for the Linux Nodes
2. Define the Linux for zSeries Hosts (Linux node)
3. Define the Linux for zSeries Group ACID Records
4. Add the Group to the User ACID Record
5. Define the Linux for zSeries User Mappings
6. Use of Profiles to better manage Users and Facilities
    1. Use TSS CREATE command with TYPE(PROFILE)

- See Chapter 3 in the ***CA PAM Client for Linux for zSeries Product Guide*** for release 14 or release 15 for complete details, located in the CA LDAP bookshelf online.

Top Secret Commands

1	TSS MODI(FACILITY(*USER55*=NAME=facility_name, ID=55))
2	TSS ADD|REM|REP(NDT) LINUXNODE(node_name) IPADDR(ip_addr_here) FACILITY(facility_name) [ACTIVE(YES|NO)]
3	TSS CREATE(group_name) NAME('descr group name') DEPT(dept) TYPE(GROUP) TSS ADD(group_name) LNXENTS(facility,gid)
4	TSS ADD(acid_here) GROUP(group_here)
5	TSS ADD|REM|REP(acid_name) LNXENTS(<facility>,<uid>,<home_dir>,<shell_name>, <group_name>) LINUXNAM(long.name)
6	TSS ADD|REM(profile_name) FACILITY(facility_name) -- & then TSS ADD|REM(acid_name) PROFILE(profile_name) – instead of: TSS ADD|REM(acid_name) FACILITY(facility_name)

Top Secret and Using Profiles (and other notes)

- Used to simplify assigning users based on common roles.
- A Profile can have many Facilities
- A Profile can be assigned to many Users
- Group values are specified using the LNXENT
- A Facility can have many Linux nodes (IP Address), but a Linux node can only be associated with one Facility definition.

Steps to define with ACF2 (see Appendix B)

- Define the Linux for zSeries Hosts using GSO LINUX records with their Linux hostname and TCP/IP address.
- Defining the Linux for zSeries Group Profile Records with the Linux group name and the GID
- Define the Linux for zSeries User Profile Records, includes defining the UID, GID (group), home directory, and shell.
- Define the Linux for zSeries Resource Rules for who can access which Linux for zSeries LPAR
- See Chapter 3 in the ***CA PAM Client for Linux for zSeries Product Guide*** for release 14 or release 15 for complete details, located in the CA LDAP bookshelf online.

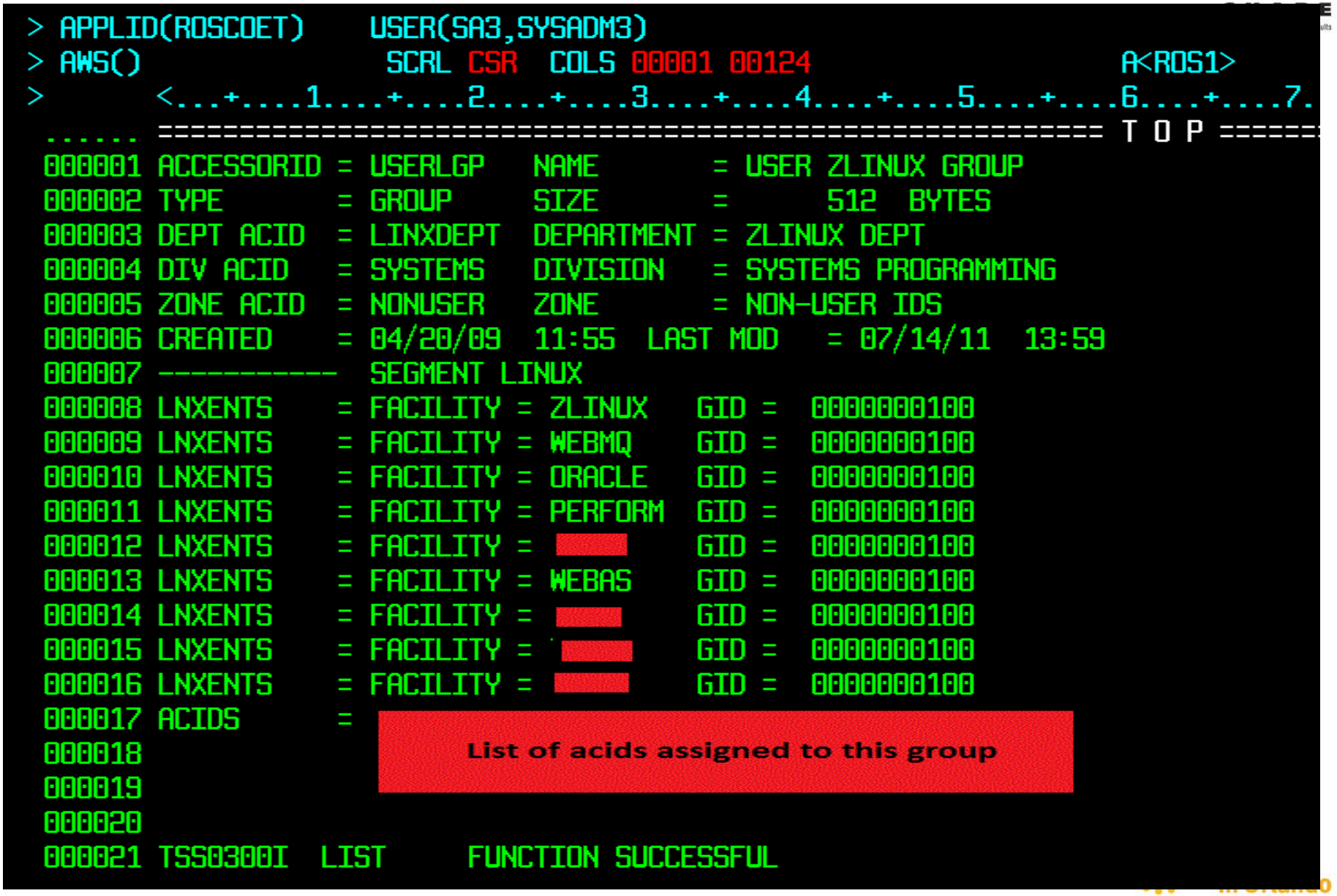
User Definition

```
> APPLID(ROSCOET)    USER(SA3,SYSADM3)
> AWS()               SCRL CSR  COLS 00001 00124                    A<ROS1>
>      <...+....1....+....2....+....3....+....4....+....5....+....6....+....7.
 ..... ================================================== TOP ======
 000001 ACCESSORID = TESTLX1   NAME       = TEST LINUX
 000002 TYPE       = USER      SIZE       =      512  BYTES
 000003 DEPT ACID  = R230202   DEPARTMENT = ENTERPRISE WIRELESS PRGM
 000004 DIV ACID   = R23Z      DIVISION   = OFFICE OF INFORMATION TECH
 000005 ZONE ACID  = USCS      ZONE       = DHS CUSTOMS AND BORDER PROTECT
 000006 CREATED    = 07/14/11  13:40  LAST MOD   = 07/15/11  09:14
 000007 PROFILES   = ZLINUX99
 000008 GROUPS     = USERLGP
 000009 INSTDATA   = 000000000
 000010 ----------   SEGMENT LINUX
 000011 LINUXNAM   = testlx1
 000012 LNXENTS    = FACILITY = ZLINUX   UID =  0000000100 GROUP = USERLGP
 000013              HOME     = /usr/local/home/testlx1
 000014              SHELL    = /bin/bash
 000015 ----------   SEGMENT OMVS
 000016 UID        = 0000005555
 000017
 000018 TSS0300I  LIST     FUNCTION SUCCESSFUL
 ..... ================================================== BOTTOM ===
```

User's assigned Facilities

```
> APPLID(ROSCOET)     USER( ,SYSADM3)
> AWS()                SCRL CSR  COLS 00001 00124                    A<ROS1>
>       <...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
 000035 -----------  SEGMENT LINUX
 000036 LINUXNAM    =
 000037 LNXENTS     = FACILITY = ZLINUX    UID =  0000001096 GROUP = USERLGP
 000038               HOME     = /usr/local/home/
 000039               SHELL    = /bin/bash
 000040 LNXENTS     = FACILITY = WEBMQ     UID =  0000001096 GROUP = USERLGP
 000041               HOME     = /usr/local/home/
 000042               SHELL    = /bin/bash
 000043 LNXENTS     = FACILITY = ORACLE    UID =  0000001096 GROUP = USERLGP
 000044               HOME     = /usr/local/home/
 000045               SHELL    = /bin/bash
 000046 LNXENTS     = FACILITY = PERFORM   UID =  0000001096 GROUP = USERLGP
 000047               HOME     = /usr/local/home/
 000048               SHELL    = /bin/bash
 000049 LNXENTS     = FACILITY = TPAC      UID =  0000001096 GROUP = USERLGP
 000050               HOME     = /usr/local/home/
 000051               SHELL    = /bin/bash
 000052 LNXENTS     = FACILITY = WEBAS     UID =  0000001096 GROUP = USERLGP
 000053               HOME     = /usr/local/home/
 000054               SHELL    = /bin/bash
 000055 LNXENTS     = FACILITY = WHTI      UID =  0000001096 GROUP = USERLGP
 000056               HOME     = /usr/local/home/
```

Group Definition



Known Issues

- sudo & CAesm
  - Redefine the sudo member in /etc/pam.d to use ESM
  - Where the wheel group is defined as a secondary group
    - Is wheel defined in CA ESM or /etc/group
  - Upgrade the sudo version
    - Version 1.6.9 of higher required
- User enters their User ID in incorrect case (resolved)
- Samples use of "**`pam_stack.so service=caesm-auth`**"
  - Creates a "*Deprecated pam_stack module called from service*" message in /var/log/secure
- Cannot Change a CA ESM Password with passwd when working with SuSE Linux
  - Returns "**`passwd: Unknown user <userID>`**"
  - Only affects passwd version 3.0.8 or lower

Troubleshooting Tips

- From the DSI server, look at the log files as defined in the startup task and configuration file (z/OS) for SAF messages.
  - F *TaskName*,SET,DEBUG,*nnnn* (0-65535)
  - F *TaskName*,SET,LOGLEVEL,*nnnn* (0-65535)
- On the proxy server use (on each Linux guest):
  - CA_esm_ctrl command to verify connection to DSI server
  - CA_esm_showcfg to check configuration file
- Put the Proxy server in Debug mode
  - Review log files for TSS, nscd & proxy messages (/var/log/caesm.log)
- To put nscd daemon in debug mode:
  - service nscd stop
  - nscd –d (will output messages to screen or pipe it to a file)
- Create a cron script (or other method) to monitor the proxy server and it's connection to the DSI server
  - *netstat -ntp | grep CA_esm_proxy*

References

- CA Doc for CA PAM ESM Client (login req.):
  - https://support.ca.com/cadocs/1/CA%20LDAP%20Server%20r15-ENU/Bookshelf_Files/PDF/PAM_Product_ENU.pdf
- CA Doc for Top Secret Commands (login req.):
  - https://support.ca.com/cadocs/1/g013431e.pdf
- CA Bookshelf for CA DSI and CA LDAP (login req.):
  - https://support.ca.com/cadocs/1/CA%20LDAP%20Server%20r15-ENU/Bookshelf.html
- Cool References to understand PAM & Linux:
  - http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/Linux-PAM_SAG.html
  - http://content.hccfl.edu/pollock/AUnix2/PAM-Help.htm
  - http://www.linux.com/archive/feed/39115

Questions????

Appendix A
TLS Settings (`reference from /etc/CA_esm.conf`)

The CA ESM proxy server uses the following options for establishing secure connection to the ESM host server using Transport Layer Security (TLS). The CA ESM proxy server recognizes these options if, and only if, the software has support for TLS.

tls-required option

Specifies whether or not the CA ESM proxy server is to establish a TLS connection with all hosts. If option is on, then the CA ESM server requests a TLS connection with all hosts. Conversely, if option is off, then the CA ESM server does not request a TLS connection with any host. You can use yes and true as alternatives for on. Similarly, you can use no and false as alternatives for off. The default for this option is off.

This option applies to all hosts unless explicitly overridden by the TLS parameter on the esm-host option. See the section "HOST DEFINITIONS" below for a description of the TLS parameter.

tls-supported option

Specifies whether or not the CA ESM proxy server can establish a TLS connection with any host. If option is on, then the CA ESM server can establish a TLS connection with a host. Conversely, if option is off, then the CA ESM server can not establish a TLS connection with a host. You can use yes and true as alternatives for on. Similarly, you can use no and false as alternatives for off. The default for this option is on.

tls-cacertdir pathname

pathname specifies the absolute pathname of a directory to use for server certificate verification. This directory must be in "hash format". See the manual page for the Openssl verify(1) command for more information. These are also used when building the client certificate chain.

tls-cacertfile pathname

pathname specifies the absolute pathname of a file containing trusted certificates to use during server authentication and to use when attempting to build the client certificate chain. The file can contain multiple certificates concatenated together. All the certificates in this file must be in PEM format.

tls-cert pathname1 [pathname2] [id=name] [password=option]

pathname1 specifies the absolute pathname of the file that contains the certificate. In addition to the certificate, this file may also contain all of the CA certificates needed to verify the certificate. If the CA certificates are present, they must be in the correct order. This file may also contain the private key for this certificate. All objects must be in PEM format.

pathname2 specifies the absolute pathname of the file that contains the private key for this certificate. This parameter is required if pathname1 does not contain the private key. The private key must be in PEM format.

Note that when the CA ESM proxy server sends this certificate to the ESM server, it will try to send the entire certificate chain. You can provide the needed certificates by placing them in the pathname1 file along with the certificate, or by placing them among the trusted certificates specified by the tls-cacertdir or tls-cacertfile option.

The id parameter attaches an identifier to this certificate. This allows the esm-host option to refer to this certificate. name is the identifier to attach. It must be a sequence of alphanumeric characters of any length. It should be unique. If more than one tls-cert options specify the same name, only the first one will be usable.

Appendix A
TLS Settings (continue)

The password parameter is required if the private key is encrypted. Option can have one of the following values:

pass:password

password must be the required password in clear text. Note that the NSS module must be able to read the configuration file. Therefore the configuration file must be readable by everyone. Since this makes that the password visible to everyone, this form should only be used where security is not important.

file:pathname3

pathname3 must be the absolute pathname of a file that contains the required password. The first line of this file is the password. The mode of pathname3 should be set to allow access to the owner of the file.

stash:pathname4

pathname4 must be the absolute pathname of a file that contains the required password. The contents of this file are the encrypted passwords. You can create this file with the command:

CA_esm_stash âpasswordâ pathname3

Note: this option is accepted if, and only if, the CA ESM proxy server has been configured to support stash files.

If the password parameter is not specified and a password is required then any attempt to establish a TLS connection using this certificate definition will fail. In particular note that the CA ESM proxy server will not prompt for the password.

tls-checkpeer option

Specifies whether or not to verify the ESM server certificate. If the ESM server certificate is not valid, the attempt to establish a TLS connection with the ESM server will fail. If option is on, then the CA ESM server verifies all ESM server certificates. Conversely, if option is off, then the CA ESM server does not verify any ESM server certificates. You can use yes and true as alternatives for on. Similarly, you can use no and false as alternatives for off. The default for this option is on.

Note that if the ESM server sends a certificate then it is always verified. (The ESM server does not send a certificate if the negotiated cipher suite is an "anonymous" cipher suite. That is, if it is a cipher suite that does not involve authenticating the server.) If option is off, then the CA ESM proxy server will log the results of this verification, but will proceed as if the certificate is valid.

tls-ciphers option

This option allows you to modify the cipher list sent by the CA ESM proxy server to the ESM server. Although the ESM server determines which cipher suite is used, it should take the first entry in this list that it supports. Option is a cipher list to convert to a cipher preference list. See the manual page for the OpenSSL ciphers(1) command for more information. If this option is not specified the default is the default cipher list.

Appendix B
ACF2 Commands

- Define the Linux for zSeries Hosts using GSO LINUX records with their Linux hostname and TCP/IP address.
  - To define hosts and TCP/IP addresses, issue the following command: SET CONTROL(GSO)
    INSERT LINUX.qual MACHNAME(linux name_here)
    IPADDR(ip_addr_here) ACTIVE

- Defining the Linux for zSeries Group Profile Records with the Linux group name and the GID
  - To define a Linux group name, issue the following command:
    SET PROF(GROUP) DIV(LINUX)
    INSERT group_here LINUXGID(gid_here)

- Define the Linux for zSeries User Profile Records, includes defining the UID, GID (group), home directory, and shell.
  - To define a user profile record to support ID mapping, issue the following command:
    SET PROF(USER) DIV(LINUX)
    INSERT lid_here.qual LINUXNAM(linux_long_name) LINUXUID(uid)
    LINUXGRP(group_name) LINUXHOM(home_directory)
    LINUXPGM(shell_name)

Appendix B
ACF2 Commands (cont)

- Define the Linux for zSeries Resource Rules for who can access which Linux for zSeries LPAR
  - To Define the SAFDEF records, issue the following command:

  INSERT SAFDEF.LINUX ID(LINUX) MODE(GLOBAL) -

  RACROUTE(REQUEST=AUTH REQSTOR=ACF9CSFV CLASS=LINUX)

- See Chapter 3 in the CA PAM Client for Linux for zSeries Product Guide for release 14 or release 15 for complete details

Glossary

- initgroups (initialize the supplementary group access list)
  - The **initgroups**() function initializes the group access list by reading the group database */etc/group* and using all groups of which *user* is a member. The additional group *group* is also added to the list.
- NDT – Node Descriptor Table
  - Contains all CPF, LDAP, LINUX, and PassTicket application and session key-related node information.
- nscd - name service cache daemon
  - nscd provides caching for accesses of the passwd, group, and hosts databases through standard libc interfaces, such as getpwnam, getpwuid, getgrnam, getgrgid, gethostbyname, and others.
- nss - Name Service Switch
  - The /etc/nsswitch.conf file determines the order of lookups performed when a certain piece of information is requested. Each call to a function which retrieves data from a system database like the password or group database is handled by the Name Service Switch implementation in the GNU C library. The various services provided are implemented by independent modules, each of which naturally varies widely from the other.
- pam_nologin.so
  - PAM module that prevents users from logging into the system when /var/run/nologin or /etc/nologin exists
- pam_securetty.so
  - PAM module that allows root logins only if the user is logging in on a "secure" tty, as defined by the listing in /etc/securetty. pam_securetty also checks to make sure that /etc/securetty is a plain file and not world writable.
- system-auth
  - Common configuration file for PAMified services by providing a common configuration file for all applications and service daemons calling the PAM libraries.

Transport Layer Security (TLS) definition

•Transport Layer Security is a protocol that guarantees privacy and data integrity between client/server applications communicating over the Internet.

•The TLS protocol is made up of two layers:

•The TLS Record Protocol -- layered on top of a reliable transport protocol, such as TCP, it ensures that the connection is private by using symmetric data encryption and it confirms that the connection is reliable. The TLS Record Protocol also is used for encapsulation of higher-level protocols, such as the TLS Handshake Protocol.

•The TLS Handshake Protocol -- allows authentication between the server and client and the negotiation of an encryption algorithm and cryptographic keys before the application protocol transmits or receives any data.

•TLS is application protocol-independent. Higher-level protocols can layer on top of the TLS protocol transparently.