Avaya Aura® Communication Manager Administrator Logins

Release 6.2
Issue 2
July 2012

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya generally makes available to users of its products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked Web sites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on its Hardware and Software (“Product(s)”). Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this Product while under warranty is available to Avaya customers and other parties through the Avaya Support Web site: http://support.avaya.com. Please note that if you acquired the Product(s) from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by said Avaya reseller and not by Avaya.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO/ ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AUTHORIZED AVAYA RESELLER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AUTHORIZED AVAYA RESELLER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA AUTHORIZED RESELLER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Avaya grants End User a license within the scope of the license types described below. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the Documentation or other materials available to End User. “Designated Processor” means a single stand-alone computing device. “Server” means a Designated Processor that hosts a software application to be accessed by multiple users. “Software” means the computer programs in object code, originally licensed by Avaya and ultimately utilized by End User, whether as stand-alone Products or pre-installed on Hardware. “Hardware” means the standard hardware originally sold by Avaya and ultimately utilized by End User.

License types

- Designated System(s) License (DS). End User may install and use each copy of the Software on only one Designated Processor, unless a different number of Designated Processors is indicated in the Documentation or other materials available to End User. Avaya may require the Designated Processor(s) to be identified by type, serial number, feature key, location or other specific designation, or to be provided by End User to Avaya through electronic means established by Avaya specifically for this purpose.
- Concurrent User License (CU). End User may install and use the Software on multiple Designated Processors or one or more Servers, so long as only the licensed number of Units are accessing and using the Software at any given time. A “Unit” means the unit on which Avaya, at its sole discretion, bases the pricing of its licenses and can be, without limitation, an agent, port or user, an e-mail or voice mail account in the name of a person or corporate function (e.g., webmaster or helpdesk), or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software. Units may be linked to a specific, identified Server.
- Named User License (NU). End User may: (i) install and use the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use the Software on a Server so long as only authorized Named Users access and use the Software. “Named User”, means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya's sole discretion, a “Named User” may be, without limitation, designated by name, corporate function (e.g., webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.
- Shrinkwrap License (SR). Customer may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as “shrinkwrap” or “clickthrough” license accompanying or applicable to the Software (“Shrinkwrap License”). (see “Third-party Components” for more information).

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, or Hardware provided by Avaya. All content on this site, the documentation and the Product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Third-party components

Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements (“Third Party Components”), which may contain terms that expand or limit rights to use certain portions of the Product (“Third Party Terms”). Information regarding distributed Linux OS source code (for those

Products that have distributed the Linux OS source code), and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply to them is available on the Avaya Support Web site: http://support.avaya.com/Copyright.

Preventing Toll Fraud

"Toll fraud" is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of Toll Fraud associated with your system and that, if Toll Fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Toll Fraud Intervention

If you suspect that you are being victimized by Toll Fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support Web site: http://support.avaya.com. Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

Trademarks

Avaya® and Avaya Aura® are trademarks of Avaya Inc.

The trademarks, logos and service marks ("Marks") displayed in this site, the documentation(s) and product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the documentation(s) and product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

All non-Avaya trademarks are the property of their respective owners.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support Web site: http://support.avaya.com.

Contact Avaya Support

See the Avaya Support Web site: http://support.avaya.com for product notices and articles, or to report a problem with your Avaya product. For a list of support telephone numbers and contact addresses, go to the Avaya Support Web site: http://support.avaya.com, scroll to the bottom of the page, and select Contact Avaya Support.

Contents

Chapter 1: Overview

Overview

This document describes how the administrator logins in Communication Manager are processed. Communication Manager 4.0 and later versions support access to Linux Pluggable Authentication Module (PAM) subsystem's configuration files. The PAM subsystem controls administrator login processing. PAM supports local host accounts as well as Authentication, Authorization, and Accounting (AAA) via an external server such as Lightweight Directory Access Protocol (LDAP).

Communication Manager R4.0 eliminated the requirement that all logins be host accounts.You can configure and manage administrator logins using standard Linux commands, such as **useradd**, and wrapper commands such as **cmuseradd**, as well as the System Management Interface of Communication Manager.

This document is not a programming or administration manual. It describes the features of Linux PAM subsystem supported by Communication Manager. It also describes how to configure Linux PAM subsystem for Communication Manager.

Important:

You must have root level access to the Communication Manager server to administer it for PAM.

This guide is intended for experienced Linux administrators.

Support

Visit the Avaya Support website at http://support.avaya.com for the most up-to-date documentation, product notices, and knowledge articles. On the Avaya Support website at http://support.avaya.com, search for notices, release notes, downloads, user guides, and resolutions to issues. Use the Web service request system to create a service request. Chat with live agents to help answer questions. If an issue requires additional expertise, agents can quickly connect you to a support team.

Chapter 2: The Linux Pluggable Configuration Module

Overview

The following mechanism needs to be in place when a user logs in to a computer system:

- Authentication – The system needs to identify the user.

  The most common way for identifying a user on a computer system is an ID and a password. Some other means to identify a user include:

  - Retinal scan
  - Finger print
  - Voice sampling
  - X.509 certificate
  - A one-time password implementation such as RSA SecurID® or SafeWord®

- Authorization/Accounting – The access restrictions and permissions must be set by a system administrator.

  In Linux, system administrator adds a user to one or more groups. The user also needs a home directory and a program (shell) to start with. The system administrator can also specify specific hours of the day or days of the week, when the user cannot access the system.

- Password – Forcing a user to change identifier.

  The most common type of identifier is a password. Some other types of identifiers include encryption keys, PINs, token serial numbers, etc.

- Session – The system needs to allocate resources to the user.

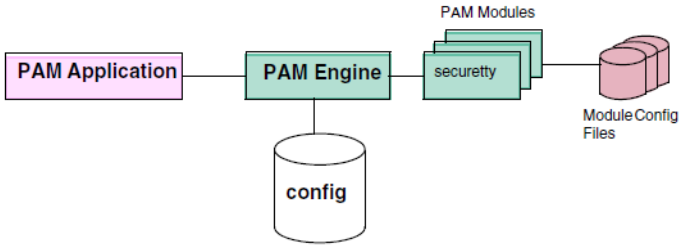
  For example, the system needs to create a home directory for the user, when they log in for the first time.

Users generally have access to multiple systems. In the absence of PAM, an administrator would have to administer access to each service for each user on every system they access.

The PAM subsystem centralizes the user identification process on one server so that individual service access modules do not have to understand exactly how a user's identity is proven.

The PAM subsystem consists of the following three components as shown in Figure 1.

- The PAM engine with PAM modules, a collection of libraries called by PAM applications
- PAM engine configuration files
- Module configuration files



The PAM application interacts with the PAM engine through a PAM conversation. When the PAM application needs to process a new login session, it calls the PAM engine to begin the conversation. The PAM engine then scans the configuration files to check the specified login mechanism. A PAM configuration file consists of the following four sections:

- authentication
- accounting
- password
- session processing

Each section of the PAM configuration file contains a list of the relevant PAM modules and the rules for using them.

Not all access points can interact with the PAM engine directly. Modules such as Communication Manager SAT interface, Secure FTP (SFTP), trivial FTP (TFTP) and telnet (in.telnet) interact with the PAM engine through xinet.d. These modules use the login module to process logins. Other modules such as PPP and SSH can interact with the PAM engine directly because they are PAM applications. In Figure 2, http.d is crossed out. Although http.d can use the PAM subsystem, the Communication Manager SMI itself is a PAM application. The SMI can authenticate users directly, eliminating the need to do the authentication through the Apache web server.

You can change the way PAM processes logins by modifying the configuration files, or adding another module to the subsystem and modifying the configuration file of the new module.

Communication Manager system comes with all the basic PAM modules it supports. You must configure it to make it work with Communication Manager. You must modify the PAM configuration file and the configuration files of the individual PAM modules to change the way

PAM handles AAA. Figure 2 shows a PAM configuration file structure:

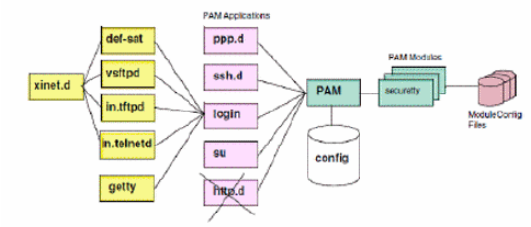


Figure 2. AAA Structure

PAM configuration file structure

There are multiple methods to configure PAM.

Note:

This document describes only the method that works with Communication Manager. For complete information on other methods, see *PAM Administrator's Guide*.

You can place the configuration information for the PAM engine in a single file, `/etc/pam.conf`, or you can provide one configuration file for each PAM application in the `/etc/pam.d` directory. Communication Manager uses the second method. The `/etc/pam.d` directory contains a series of files, one for each PAM application. For example,

- crond
- sshd
- login
- vsftpd
- passwd
- su
- sudo
- other
- mv-auth

There are two methods for PAM applications to share all or portions of their configuration data among them. One method uses a special module named pam_stack and the other method uses the pam_include module. Communication Manager currently uses the pam_stack method.

Important:

The pam_stack method is currently deprecated by Red Hat and will be replaced with the pam_include method. Communcation Manager will support pam_include method at that time.

By default, the pam_stack method uses common file called system_auth. Communication Manager does not use this file. Communication Manger uses a file named mv-auth. This allows the system_auth file to remain unchanged and used. This way Communication Manager can deliver mv-auth without concern that other tools might modify it as they might system-auth. In most cases, the only PAM configuration files you must modify are mv-auth and su.

When a PAM application initiates a PAM conversation with the PAM engine, it uses the configuration file called *other*. The *other* configuration file denies access as a safety measure if your system is not properly configured.

All configuration files are text files and have a similar structure. All configuration files have the following four sections:

- Authentication
- Accounting
- Password
- Session

The following table shows contents of an example configuration file:

Module Type	Control Flag	Module Path	Args.
• auth • account • password • session	• required • required • required • required	`/lib/security`	

- Module Type – identifies one of the four sections of the configuration file, and must contain one of the values in Table 2.
- Control Flag – defines how the PAM engine processes.
  - Required – the PAM module identified on this line in Module Type must process successfully. If the PAM module fails, PAM engine still processes the lines following this line, but the PAM application fails.
  - Requisite – the PAM module identified on this line in Module Type must process successfully. If the PAM module fails, PAM engine does not process the lines following this line, but the PAM application fails.
  - Sufficient – If the PAM module fails, PAM engine still processes the lines following this line. If the PAM engine successfully processes all other required and requisite modules, the PAM application still succeeds. If he PAM module succeeds and if there are no previous entries marked as required or requisite, the PAM engine does not process any more instances of this module type.
  - Optional – modules with optional as control flag do not affect the result.
  - Include – includes lines from the configuration file identified by the module path for this line.
- Module Path – specifies path for the PAM module to be invoked.

Note:

if the path does not begin with a forward slash, the module must be stored in the /lib/security directory.

- Args. – specifies arguments for the PAM module in the module path. For more information on arguments, see documentation for the PAM module.

PAM modules

Communication Manager is built upon the Linux operating system from Red Hat. Red Hat delivers operating system components in files known as Red Hat Package Manager (RPM) files. RPM files are somewhat like a sophisticated version of ZIP files and contain the software as well as scripts to install the operating system appropriate components in appropriate places as well as perform other tasks. Although Communication Manager does not load all the available RPMs, when Communication Manager needs some portion of the software in an RPM, the entire RPM is usually loaded. If the RPM contains components Communication Manager does not need, these components are usually still loaded but just never configured to be used. Components within an RPM may have dependencies on each other, so even though Communication Manager does not use a component directly, some other component may have a dependency on it. Components within an RPM may change with each release of the operating system. It is much safer and easier to load the entire RPM. It also makes it easier to apply security updates.

There are a number of components in the RPM for PAM modules that Communication Manager does not use or that may not be appropriate for use on Communication Manager. For example, the module pam_xauth is related to X-windows which Communication Manager does not support. Table 3 illustrates the PAM modules that might be resident in /lib/security. However, just because the module is here doesn't imply that its use is recommended or supported. Comments in the table identify PAM modules not suited for use with Communication Manager.

The following table lists all the module that might be present in the `/lib/security` directory:

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
pam_access	account	Controls access by individual users or groups through specific ports or from specific hosts.	`/etc/security/access.conf`		

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
pam_chroot	auth account session	Isolates a user to a subset of the total file system by changing the meaning of "/" for this user to be some other directory. For example, `/a/b/c`		No	This is more applicable to general computing environments and not used in Communication Manager.
pam_console	auth session	Allows a user special permissions and control if logged in through the system console.		No	Since Communication Manager systems have no local keyboard and monitor, this is not useful.
pam_cracklib	password	Defines acceptable user password characteristics.			
pam_debug				Yes	This module is used by debugging code and should only be used by Avaya Tier IV support.
pam_deny	auth account password session	Always denies access.			Generally, this should appear at the end of a PAM section to deny by default.
pam_env	auth	Used to set environment variables for a particular user.			
pam_filter	auth account	Designed to invoke "filters".		No	A filter is a program that

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
	password session				needs to be provided by a software developer to work in conjunction with a PAM application. There are no useful filters provided so this module has no purpose on a Communication Manager server.
pam_ftp	auth	Intended to be used with FTP to provide anonymous login.		No	Use of FTP is not secure and not recommended. Even when FTP is enabled on a Communication Manager server, this module is not used.
pam_group	auth	Used to assign group membership based on requested service.		No	This module is generally not used on Communication Manager servers.
pam_issue	auth	Prepends the content of an issue file to the ID prompt during login.	`/etc/issue`	No	Use of this module is not recommended because not all clients support it and its use can

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
					sometimes prevent users from logging in at all.
pam_lastlog	session	Displays time of last login.	`/var/log/lastlog`		
pam_ldap	auth account password	LDAP authentication module.	`etc/ldap.conf`		Although not a module to be configured via mv-auth, nss-ldap uses LDAP and uses the same configuration file, `/etc/ldap.conf`. Default ports: • 389 TCP for LDAP • 636 TCP for LDAPS
pam_limits	session	Sets resource limits for groups and users. For example, max logins, max syslogins, etc.	`/etc/security/limits.conf`		You should set only maxlogins, and maxsyslogins on Communication System.
pam_listfile	auth	Used to grant or deny access to a user based on the content of a specified file.		No	Generally not used on Communication Manager servers.
pam_localuser	account	Allows a users authorization information to be obtained from the local files in order to			This module needs to be used in the account section to

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
		prevent attempts to access an external AAA server.			support local host accounts whenever there are also external accounts in LDAP or RADIUS.
pam_loginuid	session	Sets the loginuid for the process that was just authenticated.		No	Use of this module is not appropriate for the software supplied with Communication Manager. It should never be placed in mv-auth as it will interfere with things like su or sudo whose purpose is to change the effective UID of the user.
pam_mail	auth session	Displays `you have new mail` message to the user.		No	Since Communication Manager does not support incoming mail, this module has no use.
pam_mkhomedir	session	Creates home directories on the fly.			See section 10 on page 20 for use of this module.
pam_motd	session	Generates a confirmation	`/etc/motd`		

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
		message after successful login.			
pam_nologin	auth account	If the file `/etc/nologin` exists, only root user may login. Other users are denied access but are shown the content of `/etc/nologin`.	`/etc/nologin`		Do not use this feature if *root* user does not have direct login access.
pam_permit	auth account password session	Always allow access.		No	Do not use this module.
pam_postgresok				No	Not supported on Communication Manager.
pam_pwdb	auth account password session	Specifies locations for user credentials.	`/etc/pwdb.conf`	No	Not supported on Communication Manager.
pam_radius_auth	auth account	RADIUS authentication module.	`/etc/raddb/server`	Yes	Default ports: • 1812 • 1813 udp.
pam_rhosts_auth	auth	Allows access by users already logged in at another specified host to login without additional authentication.	`/etc/hosts.equiv` `~/.rhosts`	No	Not recommended.
pam_root_login	auth	Restricts unauthorized root logins based on product offer.		Yes	Should always be present.

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
pam_rootok	auth	Used to allow root access to a service without having to enter a password.		No	Not recommended.
pam_rps	auth	Provides challenge response authentication.		No	Not supported on Communication Manager.
pam_securetty	auth	Limits root login to a specified list of ports which may be a null list.	`/etc/securetty`		
pam_selinux	session	Used to set the default security context.		No	Communication Manager does not support selinux due to performance issues.
pam_shells	auth	Authentication is granted if the users shell is listed in /etc/shells. If no shell is in /etc/passwd (empty), the /bin/sh is used (following ftpd's convention). Also checks to make sure that /etc/shells is a plain file and not world writable.		No	Not used on Communication Manager.
pam_stack	auth account session password	Supports a common configuration for multiple services.			See a discussion of this module in the previous section of this document.
pam_stress				No	Not supported on

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
					Communication Manager.
pam_succeed_if	account	Succeeds based on characteristics of the account such as UID value.			
pam_tally	auth account	Counts user login attempts and denies access after a specified number of failed attempts.	`/var/log/faillog`		
pam_time	account	Used to restrict access by time of day or day or week.	`/etc/security/time.conf`		
pam_timestamp	auth	When an application opens a session using pam_timestamp, a timestamp file is created in the timestampdir directory for the user. When an application attempts to authenticate the user, a pam_timestamp will treat a sufficientlyrecent timestamp file as grounds for succeeding.		No	
pam_unix	auth account session password	This is the standard Linux module for authentication of local host accounts.			
pam_unix_acct				No	Not used on Communication Manager servers.

Module Name	Module Type	Purpose	Configuration or other related file	Used by Communication Manager?	Comments
pam_unix_auth				No	Not used on Communication Manager servers.
pam_unix_passwd				No	Not used on Communication Manager servers.
pam_unix_session				No	Not used on Communication Manager servers.
pam_userdb	auth	Authenticates users based on content of a Berkeley DB.		No	This module is not supported on Communication Manager.
pam_warn	auth password	Logs information about a login attempt to syslog. Useful in the "other" configuration file to warn of attempts to use unknown services.			
pam_wheel	auth account	Restricts root access to members of the wheel group.		No	This is not used by default on Communication Manager because root accounts are very restricted.
pam_xauth	session	Used for x-windows environments.		No	Not supported on Communication Manager

In addition to these PAM modules, you can also load the following licensed modules:

- RSA SecurID
- SafeWord

You must obtain license for these modules from the respective vendors.

Related modules

Communication Manager runs unused_login_audit to look for and lock user logins that have not been used for a specified period of time. You must create the configuration file, `/etc/opt/ecs/unused_login_audit.conf`. This file must contain at least the following two lines:

`MaxUnusedDays=N`, where the *N* is the number of days a login may remain unused before it is locked.

```
Exceptions=root,sroot,init,inads,craft,adadmin
```

The Exceptions line contains a list of logins that Communication Manager must ignore when running this audit. You can add other logins to the list. You can also add as many *Exceptions* lines as needed. The unused_login_audit utility depends on the output of `pam_lastlog` in `/var/log/lastlog`. Communication Manager does not run this audit by default. You must define a schedule to run this audit via the Linux CRON service.

You must create the `/etc/cron.d/unused_login_audit.cron` file. The content of this file is similar to the following:

```
# [minute] [hour] [day of month] [month] [day of week] [program to be run]

00*** /opt/ecs/bin/unused_login_audit >>/dev/null>&1
```

The first line indicates the structure for lines in this file. The second line causes the audit to run every day at midnight. An asterisk (*) means *all*. For more information, see *man S 5 crontab*.

PAM configuration file contents

Figure 3 illustrates configuration file for a PAM application.

```
auth        required    pam_env
#auth       required    pam_issue (optional)
#auth       required    pam_tally (optional)
auth        required    pam_root_login
#auth       required    securetty (optional)
auth        required    pam_asg collect_password
auth        sufficient  pam_asg audit
#auth       sufficient  pam_external_AAA_goes_here (optional)
auth        sufficient  pam_unix
auth        required    pam_deny

account     required    pam_unix
account     required    pam_access
#account    required    pam_time (optional)
#account    required    pam_tally (optional)
#account    sufficient  pam_local_user (optional)
#account    sufficient  pam_external_AAA_goes_here(optional)

password    sufficient  pam_asg
#password   required    pam_cracklib (optional)
password    sufficient  pam_unix
#password   sufficient  pam_external_AAA_goes_here (optional)
password    required    pam_deny

#session    required    pam_limits (optional)
session     required    pam_lastlog never
#session    required    pam_motd (optional)
session     required    pam_unix
```

Figure 3. Generic PAM Configuration File

Note:

The illustration does not show the complete syntax and It omits certain arguments for clarity. The notation (optional) is not a parameter but indicates that the entry is optional in the PAM configuration.

The order of lines in the configuration file is important as the lines are processed in each section in the order in which they appear in the file.

Note that pam_asg appears twice in Figure 3. You must enter the two entries in the same order as Figure 3. The two entries act as a single module. This double-entry prevents subsequent modules from processing Access Security Gateway (ASG) logins.

The pam_asg module processes accounts authenticated using Avaya's propreitery ASG one-time-password method. All Avaya services accounts are ASG authenticated accounts.

The pair of pam_asg entries must be the first authentication module that verifies user identity. That is, not necessarily the first modules in the section, but the first module that can authenticate the user.

The following example explains the importance of this order:

pam_asg before pam_unix

If the configuration file is similar to the following:

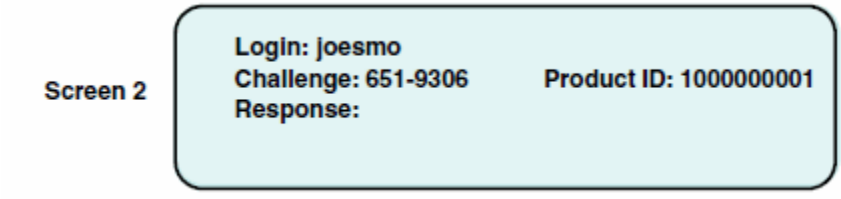
auth	required	pam_asg
auth	sufficient	pam_asg audit
auth	sufficient	pam_unix

Sequence of events:

- pam_asg prompts the user for an ID and then locates this ID in the ASG database.
- If pam_asg does not find the ID, it passes the ID to the pam_unix module.
- pam_unix receives the ID from pam_asg and prompts the user for a password.
- pam_unix then looks for this user in the local /etc/passwd and /etc/shadow files.

The user sees a prompt similar to Screen 1:

If the pam_asg module finds the ID, it prompts the user with a challenge question. The user sees a prompt similar to Screen 2:



pam_unix before pam_asg

If the configuration file is similar to the following:

auth	sufficient	pam_unix

auth	required	pam_asg
auth	sufficient	pam_asg audit

Sequence of events:

- pam_unix prompts the user for an ID and a password.
- The user sees a screen similar to Screen 1.
- pam_unix looks for the user in the`/etc/passwd`, and `/etc/shadow` files.
- If pam_unix finds the ID, the user sees a typical login prompt.
- If pam_unix does not find the ID, it passes the ID and password to pam_asg
- If pam_asg finds this user in the ASG data base, it ignores the entered password and prompts the user with an ASG one-time-password challenge.

In this case the user first sees a prompt similar to Screen 1. Because the user expects an ASG authentication challenge, the user leaves the password field blank. The system displays another prompt similar to Screen 3:

Screen 3

```
Login: joesmo
Password: *****
Challenge: 651-9306          Product ID: 1000000001
Response:
```

Automated services tools would not know this workaround. These tools would have to be reprogrammed to overcome this situation.

Pam modules that are capable of authenticating a user, generally accept two parameters, try_first_pass and use_first_pass. These parameters control how the module prompts the user when multiple pam_modules that can authenticate the user are active in the PAM configuration file. Generally the first of these modules, must be pam_asg for Communication Manager, does not accept either parameter and subsequent modules accept use_first_pass.

The pam_asg module passes the credentials entered by the user to the subsequent modules. These modules use this information to try to authenticate the user. This causes the user to be prompted once. If the subsequent modules accept the try_first_pass parameter, they may re-prompt the user again if the passed credentials are not valid.

Pam_asg prompts the user for an ID. It then looks for this ID in the ASG files on the server. If pam_asg finds the user, then pam_asg handles the user validation. However, if pam_asg does not find the user in the ASG files, then pam_asg passes the ID to the subsequent modules. Pam_asg supports a special command line parameter, collect_password, that causes pam_asg to prompt the user for a password if the user is not found in the ASG files. This password is not used by pam_asg but is passed to the subsequent modules.

If the PAM configuration file is as follows:

auth	required	pam_asg
auth	sufficient	pam_asg audit

auth	sufficient	pam_ldap try_first_pass

and pam_asg does not find the user in the ASG database, the user sees prompt similar to Screen 4.

If the PAM configuration file is as follows:

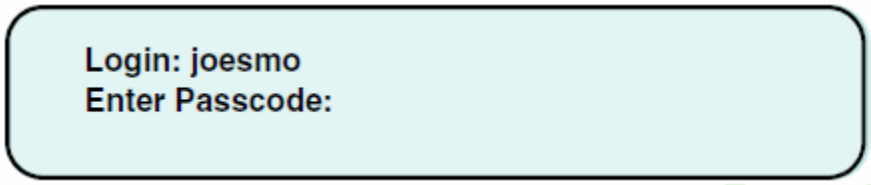
auth	required	pam_asg collect_password
auth	sufficient	pam_asg audit
auth	sufficient	pam_ldap try_first_pass

and pam_asg does not find the user in the ASG database, the user sees prompt similar toScreen 5.

If the PAM configuration file is as follows:

auth	required	pam_asg
auth	sufficient	pam_asg audit
auth	sufficient	pam_securid

and pam_asg does not find the user in the ASG database, the user sees prompt similar to Screen 6, because the pam_securid does not support the use_first_pass parameter.



If the PAM configuration file is as follows:

auth	required	pam_asg collect_password
auth	sufficient	pam_asg audit
auth	sufficient	pam_securid

and pam_asg does not find the user in the ASG database, the user sees prompt similar to Screen 7.

Even if the user enters the correct SecurID credentials to the password prompt, they will still see the Passcode prompt and must respond to it with the correct pass code. A local host account that is neither ASG protected nor SecurID protected also receives the **Enter Passcode** prompt because the pam_securid entry occurs in the mv-auth file before the pam_unix entry.

If the pam_asg line does not contain the collect_password parameter, then the user sees the **Passcode** prompt followed by the **Password** prompt. If the pam_asg line has the collect_password parameter, then the user sees the **Password** prompt first, followed by the **Passcode** prompt. The user must enter the correct password at the **Password** prompt and press `Enter` at the **Passcode** prompt in either case. This is a consequence of the way the SecurID module is designed.

Pam_deny always denies access, when it is the last entry in a section and its control flag is set to required. This means that if all the preceding modules were not able to validate the user, the default behavior is to deny access. For example, if the auth section of a configuration file is as follows:

auth	required	pam_env
auth	sufficient	pam_ldap
auth	required	pam_deny

The user cannot log in, if the LDAP server is unreachable, the user is denied access. This example illustrates a particularly bad configuration to make a point. If the LDAP server is not reachable, no one logs into this machine.

The administrator must always provide a local host account, so that the user can access the machine locally regardless of network connectivity. The control flag values are very important. Generally, all entries in a section of the PAM configuration file are not set to `required`.

Pam_securetty is used to control which ports root may log in from. Ideally, a user must never be able to log in as root directly. A user must log in first as a non-root user and then "su" to root. Pam_access provides a more sophisticated and flexible way to accomplish this. Communication Manager uses pam_access instead of pam_securetty because it provides greater flexibility in configuration.

Various Avaya services and administration tools automatically connect to the Communication Manager server. These tools parse the prompt strings during the login sequence to understand how to respond. These tools were developed over time for various systems and not all of them parse the prompt strings in the same way. For this reason, when constructing a Message of the Day (pam_motd) or messages for pam_issue, the following strings are not permitted in the message:

- `[513]` — used by FPM, CMSA, VAM.
- `513]` — used by connect2
- `]` — used by MSA
- `Login:` — used by ASA
- `Password:` or `password:` — used by ASA
- `Ogin:` — with or without a colon.
- `incorrect login`
- `assword` — from `Password` or `password:`
- `hallenge` — from `Challenge` or `challenge`
- SAT

Note:

These strings are case sensitive. For example, SAT is not permitted, but `sat` is OK. `Software Version` is not permitted but `software version` or `Software-Version` is OK.

It is better to not use any form of these strings, but if the message requires them for any reason, a change of case or punctuation is needed.

Entries for external AAA servers can occur in any order, before or after local host account processing, except pam_asg, which must be the first authenticator in the auth section.

Unlike local host accounts (pam_unix) and LDAP (pam_ldap), RADIUS, SecurID and SafeWord, are not complete AAA services. If the auth section of mv-auth file specifies one of these services, either a parallel local host account or an LDAP server must provide the authorization information. The system uses Name Service Switch (NSS) in this case.

Constraints and recommendations

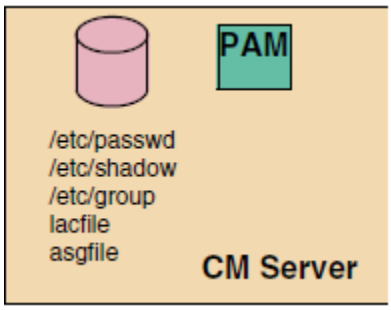
You must consider the following points when configuring the PAM subsystem:

- Pam_asg must be the first pair of modules that prompts a user.
- All ASG authenticated accounts must be local host accounts.
- At least one local host account should be present on all servers so that access is possible even if external AAA servers are not reachable.
- Password aging must not be enabled for Avaya Services accounts.
- Be careful when enabling password aging for accounts authenticated via external servers that do not support the user changing their password through the Communication Manager server. If an account expires, PAM prompts the user to change their password. If this is not possible through Communication Manager, then this user will be locked out. RADIUS accounts are an example.

- See constraints on the use of pam_limits in section 13.5 on page 45.
- PAM does not support SASL authentication.
- When configuring NSS for LDAP, you must specify files before LDAP in the search order in `nsswitch.conf`.
- If you have Tripwire enabled on the Communication Manager server, you may have to rebuild the Tripwire database after making changes to the PAM configuration.

Chapter 3: Communication Manager default PAM files

This section describes the default Communication Manager configuration. The following example uses the PAM application login. You must carefully study the actual server file contents before making changes. By default, Communication Manager is configured to support only local host accounts, as the following figure illustrates:



Note:

You can use local host accounts at the same time as any of the external AAA services. At least one local host account must always be present so that the server is accessible when access to an external AAA server is blocked for any reason.

The contents of the configuration file for the login process are similar to the example file in Figure 5:

```
#%PAM-1.0
auth            required        pam_securetty.so
auth            required        pam_stack.so service=mv-auth
auth            required        pam_nologin.so
account         required        pam_stack.so service=mv-auth
password        required        pam_stack.so service=mv-auth
#pam_selinux.so close           should be the first session rule
session         required        pam_selinux.so close
session         required        pam_stack.so service=mv-auth
session         required        pam_loginuid.so
session         optional        pam_console.so
#pam_selinux.so open            should be the last session rule
session         required        pam_selinux.so multiple open
```

In PAM configuration files, lines beginning with a pound symbol are comment lines. The lines for pam_selinux are inherited from the Red Hat distribution and not used. Communication Manager does not support Selinux due to serious performance problems. Notice the lines containing pam_stack.so. These lines invoke the content of the mv-auth file. Normally, only the mv-auth file needs to be changed to use an external authentication server.

Figure 6 illustrates contents of the mv-auth file:

```
#               <MESA:01:@(.............

auth            required        /lib/security/pam_env.so
auth            required        /lib/ecs/lib/pam_tally.so deny=5 unlock_time=600
auth            required        /lib/ecs/lib/pam_root_login.so
auth            required        /lib/security/pam_asg.so collect_password
auth            sufficient      /lib/security/pam_asg.so audit
  #auth         sufficient      /lib/security/pam_radius_auth.so use_first_pass
  #auth         sufficient      /lib/security/pam_ldap.so use_first_pass
  #auth         sufficient      /lib/security/pam_safeword.so.1 try_first_pass
  #auth         sufficient      /lib/security/pam_securid.so
auth            sufficient      /lib/security/pam_unix.so try_first_pass
auth            required        /lib/security/pam_deny.so
#
#               Account         modules
#
account         required        /lib/security/pam_unix.so
account         required        /lib/security/pam_access.so
  #account      required        /lib/security/pam_time.so
account         required        /lib/security/pam_tally.so
  #account      sufficient      /lib/security/pam_local_user.so
  #account      [default=die success=ok user_unknown=ignore service_err=ignore authinfo
                                /lib/security/pam_ldap.so
  #account      sufficient      /lib/security/pam_radius.so
#
#               Password        modules
#
password        sufficient      /lib/security/pam_asg.so
password        required        /lib/security/pam_cracklib.so retry=3 minlen=6
password        sufficient      /lib/security/pam_unix.so use_authtok
  #password     sufficient      /lib/security/pam_ldap.so use_authok
password        required        /lib/security/pam_deny.so
#
#               Session         modules
#
session         required        /lib/security/pam_limits.so
  #session      required        /lib/security/pam_lastlog.so never
  #session      required        /lib/security/pam_motd.so
session         required        /lib/security/pam_unix.so
```

Note:

The configuration includes lines for external AAA servers, but they are commented out.

Note:

You must obtain a license to use either RSA SecurID® or SafeWord® before loading these PAM modules.

Important:

You must edit the configuration files for the PAM modules to be able to use with the external AAA server.

There are two ways for specifying the second field in the PAM configuration files. The simple way uses a single keyword such as required or sufficient. The second way uses a series of keyword/value pairs to more precisely define behavior.

For example, see the LDAP entry in the account section in Figure 6. This entry reduces the delay time for logins if the LDAP server is not available or the LDAP server cannot identify the user.

Chapter 4: Configuration file for su

With the **su** (substitute user) command in Linux, you can run a command as different user, in most cases, as a root user. It is important that the account section of the configuration file for **su** does not invoke pam_access as the pam_access is configured to deny root access.

Chapter 5: Guidelines for modifying PAM configuration files

- Think carefully and document all changes in advance of making changes on the Communication Manager server.
- Perform a full system backup of all configuration files, including Communication Manager translation files.
- Whenever possible, test the PAM configuration on another computer running the same version of Linux that the Communication Manager uses. This way you can try different configurations that may be difficult to do on the production server.
- Make changes using a laptop computer connected to the Communication Manager server's dedicated laptop interface. If this is not possible, make changes using a laptop computer co-located with the Communication Manager server. The tools and procedure will work remotely, but if a network disruption causes the session to be disconnected at the wrong time, it could render the Communication Manager server inaccessible.
- Make sure the firewall is open when needed. Specifically, open the firewall ports first before enabling use of an external AAA service. Close the firewall last when removing use of an external AAA service.
- When configuring NSS for LDAP, specify files before ldap. If you specify ldap first, operations such as adding local host accounts may not work correctly. Also, it could result in long delays could result when the LDAP server is not reachable for any reason.
- When making changes on the Communication Manager server, use the three separate, simultaneous sessions described below:

    - Session 1 - using an SSH client, log in and then su to root. Change directory to /etc/pam.d and make a local copy of mv-auth. For example, cp mv-auth mv-auth.local. Do not press Enter. Minimize the window. This session provides a fall-back in case something goes wrong in Session 2.

      Do not use this session unless it is absolutely necessary to do so. Do not close this session before you complete Step f successfully. If the server appears to be locked up, try to return focus to this window/session and then press return to execute the pre--typed copy command. A reboot regains the server control. You must Wait for at least 30 minutes. when using LDAP, you must wait for 30 minutes or the amount of time equal to the sum of the bind and time-out values in the ldap.conf file , whichever is longer. The server may be very busy attempting to work with an external AAA service. Session 1 is a window with root access. If the server appears to hang while making configuration changes and the server is rebooted, the reboot does not fix it. Instead, the root window closes and a recovery action described in the next section is needed.

- Session 2 - Using an SSH client, log in a second time and as a substitute user. This is the session where changes are made.
- Using session 2 take a local backup of the files that you need to modify. This is in addition to the full backup in the second guideline. The purpose of this local backup facilitates easy restore to the original configuration if something doesn't work.
- Using session 2 make the necessary changes.
- Session 3 – Using an SSH client, verify that you can log in with a su login.
- Test other logins that might be affected by the change. For example, if an LDAP interface was added, log in using an ID that invokes this interface to verify that it works as expected.
- When all tests are complete, close all three sessions.

• Do not make large changes all at once. For example, when modifying PAM configuration files to:

- enable LDAP
- provide a message of the day
- remove most of the local host accounts
- change the login restriction rules or limits
- change password rules or expiration policy

• During the initial testing, begin with the least amount of security possible. Use a dummy account that you can be deleted later. Use unencrypted links. Not only is such a configuration easier to set up, it also allows more effective use of protocol sniffers if things aren't working. Add the security features once the basic configuration words as expected. This is one advantage of using a test AAA server and a test PC rather than the production servers to begin with.

Chapter 6: Recovery

Linux system can recover from serious problems by booting into single user mode. This provides the console root access without a password and bypasses the PAM system entirely. However, the Communication Manager server does not have a console. Some servers do not have a video card present; other servers such as S8400 do not support a video card. For those server models that support adding a monitor, keyboard, mouse, and possibly a video card, a single user boot is a possible recovery mechanism.

Another recovery option is to physically remove the hard drive from the Communication Manager server, mount this hard drive in a Linux computer as a second drive, edit the files on that computer, and re-install the hard drive on the Communication Manager server.

If a Linux computer is not available, you must boot the server from a Linux distribution disk and run the installation from the beginning.

If all these recovery options fail, you must return the system to Avaya for repairs.

Chapter 7: User login characteristics

You must assign Communication Manager administrator logins to one or more Linux login groups. You must assign each adminitrator login must be assigned a primary login group and possibly a second login group. The primary login group must be one of the groups shown in the following table:

Group Name	Group Number	Purpose
Susers	555	Privileged access to the CM server
Users	100	Non-privileged access to the CM server
Remote	888	PPP access to the CM server
Voice	102	Access to the co-resident voice mail product on the CM server.

Note:

Logins in susers group have root level access to many commands, including commands to create and modify logins.

Logins with access to either the Communication Manager telephony application or the server web pages must also be members of exactly one profile group. These groups have a default number in the range 10,000 to 10,069 inclusive and are named prof0 through prof69 respectively. You can specify a different range by administering the change on the server's web page. For example, you can specify 20,000 to 20,069 as the number range.

The output of the **id** command in Linux for the dadmin login would look similar to the following example:

```
$ id
uid=2000(dadmin) gid=555(susers) groups=555(susers),10002(prof2)
```

Chapter 8: Home directory

Communication Manager does not need many functionalities of the Linux operating system. One such capability is to assign a home directory to every user. Using a home directory, users can perform tasks such as configure their own environment, save their private files, etc.

In the Communication Manager versions older than 4.0, all users shared a common home directory in `/var/home/defty`. In Communication Manager version 4.0 and later, you can use the common home directory or create individual home directories. The default location of these individual home directories is `/var/home`.

The following are a few points you must consider when creating individual home directories:

- You must create all home directories in `/var/home`.
- The system backup does not cover any customization of contents of a home directory in `/var/home/defty`.
- For LDAP accounts, you must create home directory for a user on the Communication Manager server before the first access, or you must add the pam_mkhomedir module to the session section of the mv-auth configuration file.

Chapter 9: Configuring multiple servers

You can deploy multiple Communication Manager servers in three different roles:

- Main (active or standby)
- Survivable Remote Server (active or standby)
- Survivable Core Server

You must configure each of these servers to support AAA. This configuration information is not file synchronized among the servers, because it changes infrequently. The configuration information can differ from server to server.

However, to facilitate initial configuration of multiple servers, a special backup data set is supported. The pam_config data set includes the following files:

- `/etc/opt/ecs/lsfile`
- `/etc/asg/lacfile`
- `/etc/asg/asgfile`
- `/etc/passwd`
- `/etc/passwd-`
- `/etc/shadow`
- `/etc/shadow-`
- `/etc/group`
- `/etc/group-`
- `/etc/login.defs`
- all files in `/etc/aaa`
- all files in `/etc/pam.d`
- `/etc/ldap.conf`
- `/etc/openldap/ldap.conf`
- `/etc/nsswitch.conf`
- `/etc/nscd.conf`
- `/etc/sd_pam.conf`
- all files in `/var/ace`
- `/etc/pam_safeword.cfg`

- /etc/raddb/server
- /etc/opt/ecs/unused_login_audit.conf
- /etc/motd
- /etc/issue
- /etc/issue.net
- /var/home/defty/.hushlogin
- /etc/sshd/sshd_config
- all files in /etc/security
- /etc/securetty
- /etc/cron.d/unused_login_audit.cron

To take this backup, use the following command:

```
/opt/ecs/sbin/backup -b -d scp://username:password@hostname/dirname -k "pass
phrase" ---verbose pam_config
```

You must configure the main server before using this backup data set. In a duplicated setup, you can configure either server. Verify the file list in the backup data set before restoring the back data.

To restore the backup data set, use the following command:

```
opt/ecs/sbin/restore -r -d scp://username:password@hostname/dirname/name-of-
file -k "pass phrase" ---verbose -p /
```

Certain data in the backup such as unique digital certification backed up in /etc/aaa. You must configure appropriate certificates for the specific server. If the source server uses a different external AAA server, then you must edit appropriate files to change the address of the external server.

The pam_config backup is for manual movement of files to another server running the same software release. You must verify the data after restoring it.

You must not use the shared IP address of a duplicated pair of Communication Manager servers.

Chapter 10: Verified AAA server configurations

Verified AAA server configurations

When the Communication Manager software is first installed, only local host accounts are configured. You must edit the PAM files to incorporate support for any other type of account. The following configurations using external AAA servers have been tested:

- RSA SecurID for Authentication + LDAP/NSS/NSCD6)
- SafeWord for Authentication + LDAP/NSS/NSCD
- RADIUS for Authentication + LDAP/NSS/NSCD

This chapter describes configuration files for each of these configurations.

RSA SecurID

RSA SecurID is a token-based authentication method from RSA Security for authenticating users. You must provide user authorization through parallel host accounts or LDAP/NSS.

Communication Manager does not have the PAM application for RSA SecurID installed by default. You must purchase a license for this client from RSA Security. You must install the SecurID application in `/lib/security/pam_securid.so`,`/etc/sd_pam.conf`, and `/var/ace/sdconf.rec`. The default port for RSA SecurID is 5500 UDP. You can change the port number on the SecurID server. You must regenerate the `sd_pam.conf` file and re-install it on the Communication Manager server. verify in vendor documentation for your software.

The `sd_pam.conf` file is installed during the SecurID installation. You must generate the /var/ace/sdconf.rec on the RSA SecurID server and copy it to the Communication Manager server. For more information, see RSA SecurID documentation.

When you upgrade Communication Manager, you must reinstall the SecurID application in the new partition. Prior to the upgrade, edit the `mv-auth` file to disable the use of SecurID. After the upgrade is over, boot the system into the new partition and copy the SecurID files from the old partition to the running partition using the following command:

`cp /root2/lib/security/pam_securid.so /lib/security/pam_securid.so`

cp /root2/etc/sd_pam.conf /etc/sd_pam.conf.

You need to edit the `mv-auth` file to enable the SecurID application again.

SafeWord

SafeWord is a token-based authentication method from Secure Computing for authenticating users. You must provide user authorization through parallel host accounts or LDAP/NSS.

Communication Manager does not have the PAM application for SafeWord installed by default. You must purchase a license for this client from Secure Computing. You must install the SafeWord application in `/etc/pam_SafeWord.cfg` and`/lib/security/pam_SafeWord.so.1`. The default port for SafeWord is 5030 TCP. verify in vendor documentation for your software.

The SafeWord distribution CD-ROM has a Java based installer. If your Communication Manager server does not have a CD drive, you can install SafeWord on a separate Linux computer, and then copy it to the Communication Manager server.

When you upgrade Communication Manager, you must reinstall the SafeWord application in the new partition. Prior to the upgrade, edit the mv-auth file to disable the use of SafeWord. After the upgrade is over, boot the system into the new partition and copy the SafeWord files from the old partition to the running partition using the following commands:

cp /root2/lib/security/pam_safeword.so.1 /lib/security/pam_safeword.so.1

cp /root2/etc/pam_safeword.cfg /etc/pam_safeword.cfg

You need to edit the `mv-auth` file to enable the SafeWord application again.

RADIUS

You can use RADIUS for user authentication and accounting. You must provide user authorization through parallel host accounts or LDAP/NSS. The default ports for RADIUS is 1812 and 1813 UDP.

Chapter 11: Other PAM features

pam_access

You can use pam_access to control system access by individual users or groups of users. To use pam_access, you must enable it or add it to the PAM configuration file. You must define the access rules in the `/etc/security/access.conf` file. The syntax for defining these rules is as follows:

permission:	users:	origins:

pam_cracklib

You can use pam_cracklib to define the criteria for new passwords. For example,

- a new password must not be a palindrome, For example, `radar`.
- a new password must not be the old password, either with a changed case or with a changed order of letters.
- a new password must be a minimum certain number of characters in length.
- a new password must not be identical to the old one.
- a new password must be a combination of letters, numbers and special characters.
- a new password must not be one of the recently used passwords.

Note:

You can define the number of recently used passwords saved using the remember parameter of pam_unix in the password section of mv-auth. The default is `none`.

By default, pam_cracklib is enabled in the mv-auth file. There is no configuration file. Arguments on the cracklib invocation file control pam_cracklib.

Cracklib has two kinds of rules, internal compiled-in rules and rules which may be manipulated via command line parameters. You cannot change the internal rules except by recompiling cracklib and include the conditions listed at the beginning of this section.

In addition to the conditions described earlier, cracklib accepts the following command line parameters:

Parameter	Default	Purpose
debug		Enables additional messages in syslog. More useful to a developer with source code than to an administrator configuring cracklib.
type=xxx		When cracklib prompts for a password, it uses the string `New UNIX Password:`. You can can use this parameter to replace the word UNIX with the string xxx.
retry=N	3	Specifies the number of times the user can to try to enter a new password that meets the defined criteria before exiting. **Note:** This is the number of times a user may try to enter a new password during the password change process. This parameter does not control the number of times a user may fail to give the correct password during login.
difok=N	5	Sets the minimum number of characters in the new password that must be different from the old password. However, if the new password is at least twice as long as the old one, this parameter is ignored.
minlen=L	9	Specifies the minimum length of a password.
dcredit=N	1	Specifies the number of digits in a password.

Parameter	Default	Purpose
ucredit=N	1	Specifies the number of upper case letters in a password.
lcredit=N	1	Specifies the number of lower case letters in a password.
ocredit=N	1	Specifies the number of special characters in a password.

Note:

Cracklib has an internal rule that passwords must be at least 6 characters long. This rule takes precedence over any other rules regarding password length. You can increase the minimum required length beyond 6 by modifying the last 5 parameters. All 5 parameters together determine the minimum length requirement.

In addition to the parameter minlen, other parameters such as dcredit, ucredit, lcredit, and ocredit determine the minimum acceptable length of a password. These parameters can have either positive or negative values in any combination. Positive values for these parameters reduce the minimum length requirement. For example, if the cracklib configuration is similar to the following:

```
password        required        /lib/security/pam_cracklib.so
                                retry=3 minlen=10 dcredit=3
                                lcredit=0 ucredit=0 ocredit=0
```

and the user enters a password of 10 random lower case characters, it is acceptable. However, if the user enters a password with 4 random lower case letters intermixed with 3 digit characters, then a password 7 character in length is acceptable. The value in dcredit reduces the minimum length requirement by upto 3 characters. The password can have more than 3 digits, but the additional digits do not affect the minimum length requirement. Similar rules apply for lcredit, ucredit, and ocredit.

If the cracklib configuration is similar to the following:

```
password        required        /lib/security/pam_cracklib.so
                                retry=3 minlen=12 dcredit=1
                                lcredit=1 ucredit=1 ocredit=1
```

then an 8 character password is acceptable if it has 1 upper, 1 lower, 1 digit, and 1 special character. So the following password is acceptable:

```
4A3d.wpq
```

You can force the password to contain a mix of characters by setting one or more of the credit parameters to a negative value. The following configuration,

password	required	/lib/security/pam_cracklib.so retry=3 minlen=10 dcredit=-3 lcredit=0 ucredit=0 ocredit=0

requires passwords to be a minimum of 10 characters long and also require 3 of the characters to be digit characters.

Note:

You might encounter unexpected results when setting the minlen to 6 in combination with credit values. For example, when the cracklib configuration is minlen=6 dcredit=-1, `kdu8rg` should be acceptable, which is not the case. However, minlen=8 dcredit=-1 accepts `kdu8rgbd` as password. This is due to a bug.

Login messages

Login messages (pam_issue and pam_motd)

Linux supports displaying the following two types of messages at two different times to the users.

- issue message — which is displayed as part of the initial login prompt. This message often contains warnings about unauthorized access.
- Message of the Day - which informs legitimate users about upcoming outages, server status, such as approaching disk full conditions, or other information of interest to the user.

Related topics:

pam_issue on page 52
pam_motd (message of the day) on page 53
SSH on page 53
Telnet on page 54
HTTP on page 55
FTP/SFTP on page 55

pam_issue

Pam_issue displays the issue message from the file /etc/issue. To use this feature, you must edit the desired text in /etc/issue and place a call to its module as the second line in the mv-

auth file. However, pam_issue does not work in all cases and its use is not recommended. Some of the other means for displaying the issue message are described below.

Communication Manager installs a file named `/etc/issue.avaya` in the active partition and then copies this file to `/etc/issue`, again in the active partition. The `issue.avaya` file contains some default text that you can edit. After an upgrade, Communication Manager installs a new `/etc/issue.avaya` from the distribution. If an `/etc/issue` file exists in the current running partition, the system copies it without any modifications to the new partition. If it does not exist, then the the system copies `/etc/issue.avaya` file to `/etc/issue` in the new partition. A restore to defaults copies `/etc/issue.avaya` to `/etc/issue` and overwrites any changes made there.

pam_motd (message of the day)

You define pam_motd in the session section of the PAM configuration file. Pam_motd displays the text of file in `/etc/motd` to a user after successful login. You can edit the text in `/etc/motd` and then add the pam_motd line in the PAM configuration file.

SSH

You can configure the SSH daemon in the`/etc/ssh/sshd_config` file. You need to edit the following three entries in this file:

```
PrintLastLog no

PrintMotd no

Banner /etc/issue
```

If you set PrintLastLog to `yes`, the following message displays immediately after a successful login:

```
Last login: Mon Jul 17 11:37:10 2006 from someplace.dr.avaya.com
```

This line appears regardless of the configuration of pam_lastlog in the PAM configuration files. If pam_lastlog in the PAM configuration file is set to `never`, the system displays two messages on successful login.

The pam_lastlog message is similar to the following:

```
Last login: Mon Jul 17 11:37:10 2006 from someplace.dr.avaya.com on
pts/2
```

Setting `PrintMotd` to `yes` displays the content of the file`/etc/motd` to the user immediately after the last login information. If the mv-auth file contains the line session `required /etc/`

`security/pam_motd`, the system displays the message of the day twice. The two mechanisms work independently.

Setting `PrintMotd` to `yes` displays the content of the file `/etc/motd` to the user immediately after the last login information. If the mv-auth file contains the line session `required /etc/security/pam_motd`, the system displays the message of the day twice. The two mechanisms work independently.

The line, `Banner /etc/issue`, displays the contents of the `/etc/issue` file. You can specify any file on the banner line. However, SSH displays the content of the `/etc/issue` file before any other file you specify. The time of displaying the issue message depends on the client. Calling pam_issue in the mv-auth file has no effect on SSH as SSH is not integrated with pam_issue.

Telnet

Telnet is not configured to work with PAM. Telnet uses the login process to process user logins. This means that the user experience during login is a combination of the characteristics of the telnet daemon, in.telnetd, and the login process. The in.telnetd daemon is hard coded to display the content of `/etc/issue.net` prior to the login prompt. Additionally, the telnet daemon is incompatible with use of pam_issue. Adding pam_issue to the mv-auth file prevents a user from logging in via Telnet. If you are going to use Telnet, then you must copy `/etc/issue` to `/etc/issue.net`.

By default, login displays the time of last login from `/var/log/lastlog`, and also displays the message of the day file from `/etc/motd`. However, if a zero length file named `.hushlogin` is in the user's home directory, then login does not display this information.

Additionally, if the mv-auth file contains the following lines:

session	required	/etc/security/pam_lastlog never
session	required	/etc/security/pam_motd

The user sees the time of last login and message of the day from these entries. That is, if the `.hushlogin` file is NOT present, the user sees the time of last login and message of the day twice, once from the login process and again from the PAM entries. Telnet has the same behavior as SSH regarding display of time of last login when both methods are employed.

HTTP

The Communication Manager SMI is hard coded to display the content of the `/etc/issue` file on the "home" page and display the `/etc/motd` file after successful login.

The only way to remove this display is to remove the files.

FTP/SFTP

Communication Manager supports vsftp with an anonymous login only. FTP is disabled by default. You must enable before you can use it. FTP does not support display of last login information nor the message of the day.

FTP is not compatible with PAM and none of the message options in PAM work with FTP. FTP displays the content of the`/etc/issue` file if the `/etc/vsftpd.conf` file contains the following line:

```
banner_file=/etc/issue
```

pam_lastlog

You can use pam_lastlog to display a message with details of their last login to the user at the time of login. Pam_lastlog keeps track of user logins in the file `/var/log/lastlog`. To use this feature, uncomment the lastlog entry or add it to the mv-auth configuration file. Pam_lastlog accepts several parameters to control the message it displays. Pam_lastlog is required by the unused_login_audit and for generating reports regarding login activity.

pam_limits

You can use pam_limits to restrict resources such as the amount of CPU time, number of open files, number of processes, etc. that a user may access. These limits are more appropriate to a general purpose computing platform than to the Communication Manager server because these limits are controlled by the Communication Manager application. Attempting to change these limits via PAM may cause unexpected behavior. There are, however, two limits that you can set, maxlogins and maxsyslogins.

maxlimits controls the maximum number of sessions that may be simultaneously active for a particular user. maxsyslogins controls the maximum number of simultaneous sessions for all users taken together.

To use pam_limits, you must enable it in the PAM configuration file and edit the `/etc/security/limits.conf` file.

pam_tally

You can use pam_tally is used to deny access to a user after the maximum number of failed login attempts. You must configure this feature in the mv-auth file, before you can use it. For example, if the deny parameter of pam_tally has a value of 5, the system locks the particular user after 5 unsuccessful login attempts. A root user must needs to unlock it using the **`/sbin/pam_tally`** command.

If the parameter unlock_time = Tis set, then the system auto-enables the account after T seconds. For example, unlock_time = 600 automatically unlocks the account 10 minutes later. if the parameter deny is set to 3 and the parameter unlock_time is set to 600, and a user enters an incorrect password over a period of 3 days, the system still locks the account.

Avaya has modified pam_tally to accept a new parameter, unlock_reset. If this parameter is present on the pam_tally line, then both the attempt count and the time are cleared after unlock_time=T seconds of no activity before the next attempt. For example,

pam_tally.so deny=3 unlock_time=600 unlock_reset

The user enters three incorrect passwords, waits for 10 minutes or more, then tries a 4th time, the account is unlocked and the count of attempts is reset to 1.

The deny and unlock_time parameters work together to control the number of login attempts per unit time. This is important because it prevents a hacker using automated programs. The value in deny should be large enough to allow humans to make a reasonable number of typing errors and recover. The value in the unlock_time should be small enough for a user to wait rather than call the help desk for an unlock.

pam_time

You can use pam_time to control access based on time of day and day of week. To use pam_time, you must enable it or add it to the PAM configuration file. You must also define the rules in the `/etc/security/time.conf` file.

Index