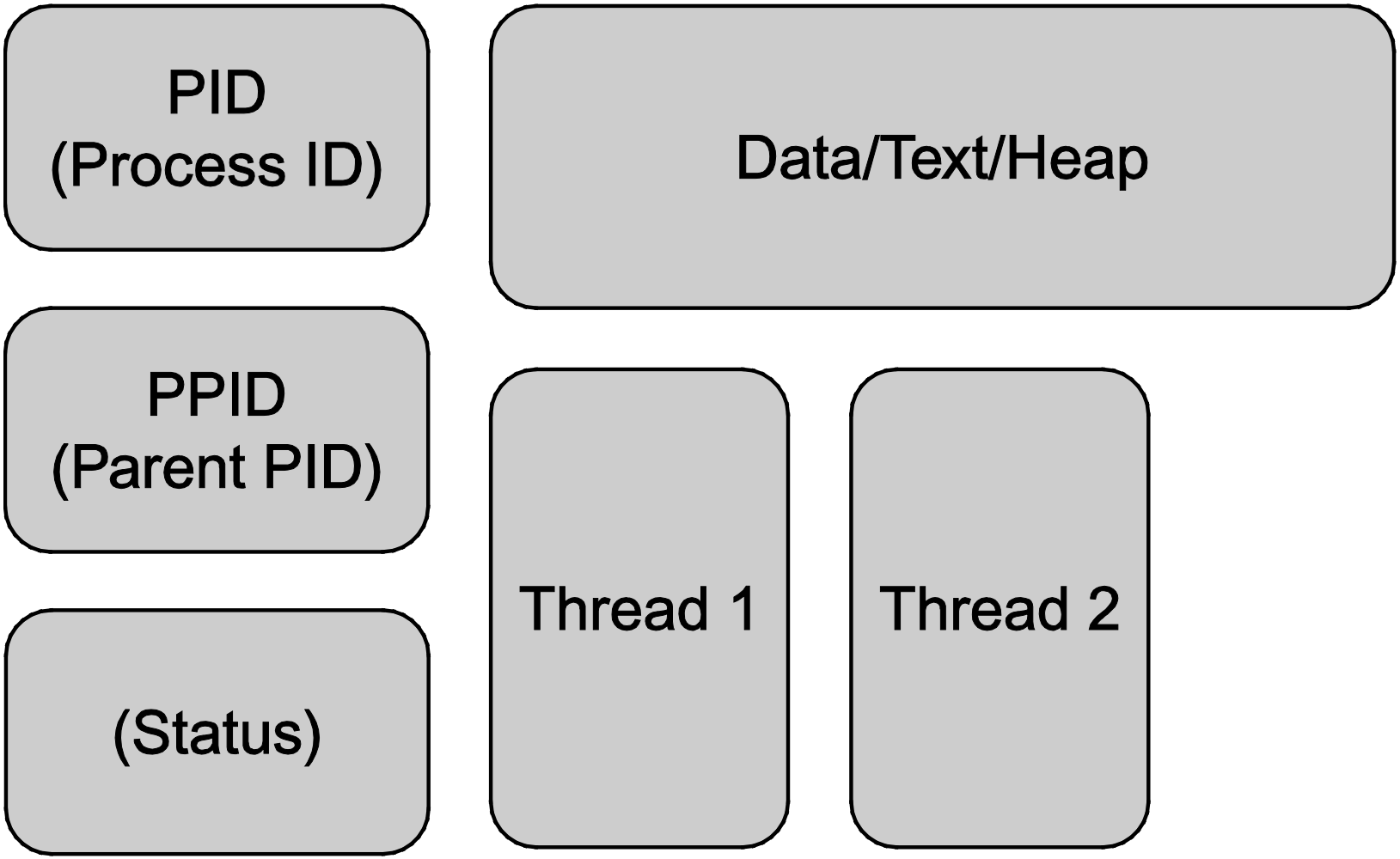
Chapter 09

Unix and Linux Security

The term POSIX stands (loosely) for “Portable Operating System Interface for uniX”. From the IEEE 1003.1 Standard, 2004 Edition:

“This standard defines a standard operating system interface and environment, including a command interpreter (or “shell”), and common utility programs to support applications portability at the source code level. This standard is the single common revision to IEEE Std 1003.1-1996, IEEE Std 1003.2-1992, and the Base Specifications of The Open Group Single UNIX Specification, Version 2.”
Partial or full POSIX compliance is often required for government contracts.

f09-01-9780123943972



Unix Process

f09-02-9780123943972

Making a directory readable for everyone:

```
# chmod o+r /tmp/mydir
# ls -ld /tmp/mydir

drwxr-xr-x   2 root     root         117 Aug  9 12:12 /tmp/mydir
```

Setting the SetID bit on an executable, thus enabling it to be run with super-user privileges:

```
# chmod u+s specialprivs
# ls -ld specialprivs

-rwsr-xr-x   2 root     root         117 Aug  9 12:12 specialprivs
```

Overview of Unix authentication methods

- Simple: a username and a password are used to login to the operating system. The login process must receive both in cleartext. For the password, the Unix crypt hash is calculated and compared to the value in the password or shadow file.

- Kerberos: The user is supposed to have a ticket-granting ticket from the Kerberos Key Distribution Server (KDC). Using the ticket-granting ticket, he obtains a service ticket for an interactive login to the Unix host. This service ticket (encrypted, time limited) is then presented to the login process, and the Unix host validates it with the KDC.

- PKI based Smartcard: the private key on the smart card is used to authenticate with the system.

```
# /etc/nsswitch.conf

#
# Example configuration of GNU Name Service Switch functionality.
#
passwd:         files nis

group:          files nis
shadow:         files nis

hosts:          files nis dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files
netgroup:       nis
```

```
# /etc/pam.d/common-password - password-related modules common to all services
#

# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords.  The default is pam_unix.

# Explanation of pam_unix options:
#
# The "nullok" option allows users to change an empty password, else
# empty passwords are treated as locked accounts.
#

# The "md5" option enables MD5 passwords.  Without this option, the
# default is Unix crypt.
#

# The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option in
# login.defs.
#

# You can also use the "min" option to enforce the length of the new
# password.
#
# See the pam_unix manpage for other options.

password   requisite   pam_unix.so nullok obscure md5

# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
# (Replaces the `OBSCURE_CHECKS_ENAB', `CRACKLIB_DICTPATH')
#

password required   pam_cracklib.so retry=3 minlen=6 difok=3

password required   pam_unix.so use_authtok nullok md5
```

f09-06-9780123943972

```
$ ssh host -luser1 -c aes192-cbc
```

f09-07-9780123943972

On Solaris simply edit the file /etc/default/login:

```
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.

# If CONSOLE is set, root can only login on that device.
# Comment this line out to allow remote login by root.
#

CONSOLE=/dev/console

# PASSREQ determines if login requires a password.
#

PASSREQ=YES

# SUPATH sets the initial shell PATH variable for root
#

SUPATH=/usr/sbin:/usr/bin



# SYSLOG determines whether the syslog(3) LOG_AUTH facility should be used
# to log all root logins at level LOG_NOTICE and multiple failed login
# attempts at LOG_CRIT.
#

SYSLOG=YES

# The SYSLOG_FAILED_LOGINS variable is used to determine how many failed
# login attempts will be allowed by the system before a failed login
# message is logged, using the syslog(3) LOG_NOTICE facility.  For
example,
# if the variable is set to 0, login will log -all- failed login attempts.
#

SYSLOG_FAILED_LOGINS=5
```

On Debian:

```
# The PAM configuration file for the Shadow `login' service
#


# Disallows root logins except on tty's listed in /etc/securetty
# (Replaces the `CONSOLE' setting from login.defs)

auth       requisite  pam_securetty.so
# Disallows other than root logins when /etc/nologin exists
# (Replaces the `NOLOGINS_FILE' option from login.defs)

auth       requisite  pam_nologin.so

# Standard Un*x authentication.

@include common-auth

# This allows certain extra groups to be granted to a user
# based on things like time of day, tty, service, and user.
# Please edit /etc/security/group.conf to fit your needs
# (Replaces the `CONSOLE_GROUPS' option in login.defs)
```

f09-08a-9780123943972

```
auth       optional   pam_group.so

# Uncomment and edit /etc/security/time.conf if you need to set
# time restrainst on logins.
# (Replaces the `PORTTIME_CHECKS_ENAB' option from login.defs
# as well as /etc/porttime)

account    requisite  pam_time.so



# Uncomment and edit /etc/security/access.conf if you need to
# set access limits.
# (Replaces /etc/login.access file)

account  required       pam_access.so

# Sets up user limits according to /etc/security/limits.conf
# (Replaces the use of /etc/limits in old login)

session    required   pam_limits.so

# Prints the last login info upon succesful login
# (Replaces the `LASTLOG_ENAB' option from login.defs)

session    optional   pam_lastlog.so

# Standard Un*x account and session

@include common-account
@include common-session
@include common-password
```

The following scheme is a good start for partitioning with read-only partitions:

- Binaries and Libraries: /bin, /lib, /sbin, /usr - read-only
- Logs and frequently changing system data: /var, /usr/var - writable
- User home directories: /home, /export/home - writable
- Additional software packages: /opt, /usr/local - read-only
- System configuration: /etc, /usr/local/etc - writable
- Everything else: Root (/) - read-only

Obviously, this can only be a start and should be evaluated for each system and application. Updating operating system files, including those on the root file system, should be performed in single-user mode with all partitions mounted writable.

f09-09-9780123943972

```
$ find /  \( -perm -04000 -o -perm -02000\) -type f -xdev -print
```

f09-10-9780123943972

```
$ find / -nouser
```

f09-11-9780123943972

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.1.0/24 --dport 22 -m state --state
NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

f09-12-9780123943972