MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

Overview

- *Part 1: Background on MLS*
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

Background

- What is multilevel security (MLS)?
- MLS implementation examples from the past and present
- The Bell–LaPadula model
- Comparing MLS with MCS

What is multilevel security (MLS)?

- The application of a computer system to process information with different sensitivities (i.e., at different security levels), permit simultaneous access by users with different security clearances and needs-to-know, and prevent users from obtaining access to information for which they lack authorization
  - http://en.wikipedia.org/wiki/Multilevel_security

MLS implementation examples from the past and present

- Specialized operating systems
- Forked variants mainstream operating systems
- Red Hat Enterprise Linux

The Bell–LaPadula model

- Focuses on data confidentiality and controlled access to classified information
  - “No read up, no write down”
  - http://en.wikipedia.org/wiki/Bell%E2%80%93LaPadula_model

The Bell–LaPadula model

- Focuses on data confidentiality and controlled access to classified information
  - “No read up, no write down”
  - http://en.wikipedia.org/wiki/Bell%E2%80%93LaPadula_model

The Bell–LaPadula model

- Focuses on data confidentiality and controlled access to classified information
  - “No read up, no write down”
  - http://en.wikipedia.org/wiki/Bell%E2%80%93LaPadula_model

The Bell–LaPadula model with write equality

- No write up
- Adds integrity, prevents noise
- Red Hat Enterprise Linux 6 MLS implements this

MLS is not MCS

- Multilevel security (MLS)
  - Read up/write down features (“security levels”)
  - Mostly military and intelligence community applications
- Multicategory security (MCS)
  - No concept of read up/write down (“categories”)
  - Military and intelligence community applications
  - Useful in other industries (healthcare, financial services)
    - Separate billing access from medical record access
  - Often easier to implement and maintain
    - When you want category separation and you don't have levels
    - Default RHEL SELinux targeted policy does MCS

Overview

- *Part 1: Background on MLS*
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

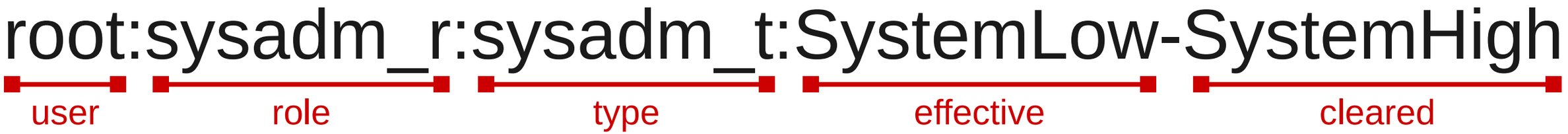
Overview

- Part 1: Background on MLS
- *Part 2: MLS with RHEL 6*
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

MLS with Red Hat Enterprise Linux 6

- selinux-policy-mls RPM
- Implements the Bell–LaPadula model with write equality
- Provides role based access control (RBAC)
  - Can separate system admin from security admin from auditor, etc.
- Provides extra protection of type enforcement (TE)
  - httpd, etc., are confined by **both** MLS and TE

id -Z


root:sysadm_r:sysadm_t:SystemLow-SystemHigh
user
role
type
effective
cleared

SELinux sensitivity and category

- SystemLow-SystemHigh = s0-s15:c0.c1023
- s = sensitivity (“classification level”)
  - 16 levels by default
  - Can only effectively be in one at a time
- c = category (“program you're read into”)
  - 1024 categories by default
  - Can have multiple categories
    - Can be read into multiple programs
- 16 * 2^{1024} possible labels!

SELinux sensitivity and category example

- s0 < s5
  - s5 has a higher sensitivity (“classification level”) than s0
  - s5 can read s0 to s5 content
  - s0 can't read s5 content
  - s5 will write exactly s5 content
  - s0 will write exactly s0 content (no higher)
  - Neither have categories
    - “Clearances but not read into any compartmentalized programs”

SELinux sensitivity and category example

- s6:c133 <> s9:c296
  - Neither can read the other
  - s9 is a higher sensitivity (“classification level”) but isn't read into category c133
  - c296 does not dominate c133
    - Unlike sensitivities, categories have no concept of domination
    - c296 and c133 are just different
  - s6:c133 will write exactly s6:c133 content
    - Role change needed to write s6 with no category

SELinux sensitivity and category example

- s1:c2,c4,c5 < s9:c2.c6,c10
  - Multiple categories
  - Dot notation defines a contiguous range of categories (“c2 through c6”)
  - s9:c2.c6,c10 can read the s1:c2,c4,c5 content
    - s9:c2.c6,c10 has a higher classification level and is read into c2, c4, and c5 (as well as c3, c6, and c10)
  - s1:c2,c4,c5 can't read the s9:c2.c6,c10 content
    - s1:c2,c4,c5 is a lower sensitivity
    - s1:c2,c4,c5 isn't read into c3, c6, and c10
  - s9:c2.c6,c10 will write exactly s9:c2.c6,c10 content

SELinux MLS and type enforcement example

- `ps -ZC httpd`
  - system_u:system_r:httpd_t:s15:c0.c1023
- `ls -Z /etc/shadow`
  - system_u:object_r:shadow_t:s0
- httpd can't read /etc/shadow
  - But isn't httpd at the highest security level and is read into all categories?
  - httpd's SELinux type enforcement policy doesn't allow access to shadow_t!
  - Most other MLS implementations don't provide this additional layer of security

Overview

- Part 1: Background on MLS
- *Part 2: MLS with RHEL 6*
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- *Part 3: Basic setup of MLS with RHEL 6*
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

Setup

- Install a RHEL 6 system (virtual guest totally fine)
  - Select “Basic Server” install
- Register system with RHN
- Update system and reboot
  - `yum -y update && reboot`
- Install the SELinux MLS policy and additional tools
  - `yum install selinux-policy-mls policycoreutils-python`

Targeted policy: one sensitivity

```
[root@rhel6 ~]# id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@rhel6 ~]# _
```


Translation table

```
[root@rhel6 ~]# id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@rhel6 ~]# cat /etc/selinux/targeted/setrans.conf
#
# Multi-Category Security translation table for SELinux
#
# Uncomment the following to disable translation libary
# disable=1
#
# Objects can be categorized with 0-1023 categories defined by the admin.
# Objects can be in more than one category at a time.
# Categories are stored in the system as c0-c1023.  Users can use this
# table to translate the categories into a more meaningful output.
# Examples:
# s0:c0=CompanyConfidential
# s0:c1=PatientRecord
# s0:c2=Unclassified
# s0:c3=TopSecret
# s0:c1,c3=CompanyConfidentialRedHat
s0=SystemLow
s0-s0:c0.c1023=SystemLow-SystemHigh
s0:c0.c1023=SystemHigh
[root@rhel6 ~]# _
```


Switch from targeted to mls SELINUXTYPE

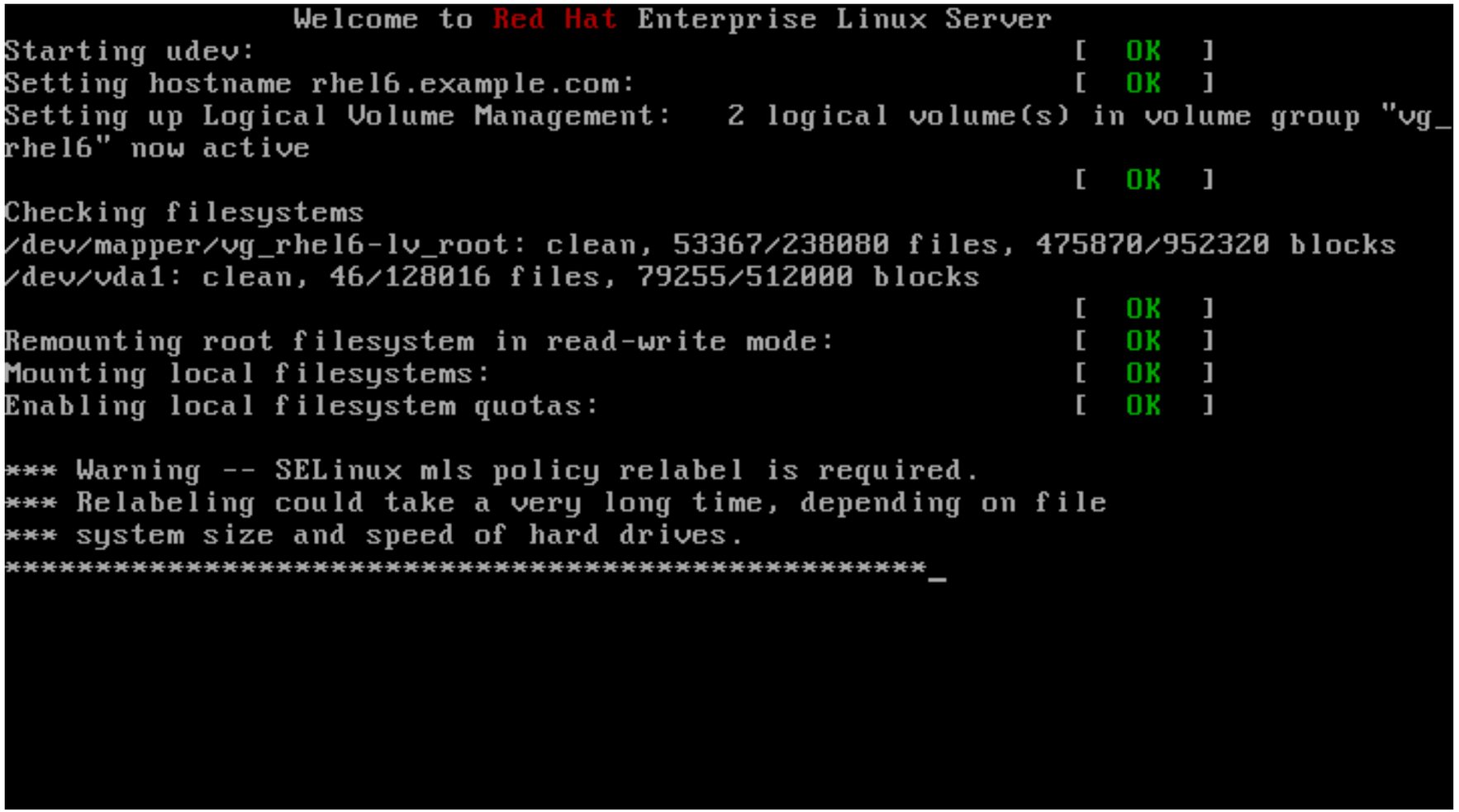
```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=mls


~
~
~
~
~
~
~
~
~
~
~
"/etc/sysconfig/selinux" 13L, 453C written
```


Relabel the file system with the MLS policy

- Tell SELinux to relabel the file system with the current (MLS) SELinux policy on next boot and reboot
  - `touch /.autorelabel && reboot`
- Reboot will take longer than usual

File system relabeling



MLS policy: 16 sensitivities

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# _
```


Different translation table

```
s2:c1=B

# ranges for Unclassified
s0-s1=SystemLow-Unclassified
s1-s2=Unclassified-Secret
s1-s15:c0.c1023=Unclassified-SystemHigh

# ranges for Secret with compartments
s0-s2=SystemLow-Secret
s0-s2:c0=SystemLow-Secret:A
s0-s2:c1=SystemLow-Secret:B
s0-s2:c0,c1=SystemLow-Secret:AB
s1-s2:c0=Unclassified-Secret:A
s1-s2:c1=Unclassified-Secret:B
s1-s2:c0,c1=Unclassified-Secret:AB
s2-s2:c0=Secret-Secret:A
s2-s2:c1=Secret-Secret:B
s2-s2:c0,c1=Secret-Secret:AB
s2-s15:c0.c1023=Secret-SystemHigh
s2:c0-s2:c0,c1=Secret:A-Secret:AB
s2:c0-s15:c0.c1023=Secret:A-SystemHigh
s2:c1-s2:c0,c1=Secret:B-Secret:AB
s2:c1-s15:c0.c1023=Secret:B-SystemHigh
s2:c0,c1-s15:c0.c1023=Secret:AB-SystemHigh
[root@rhel6 ~]# _
```


Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- *Part 3: Basic setup of MLS with RHEL 6*
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- *Part 4: Separating system and security admin roles*
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

Change to secadm_r

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# newrole -r secadm_r
Password:
[root@rhel6 ~]# id -Z
root:secadm_r:secadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# _
```


Separate sysadm_r from secadm_r

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# newrole -r secadm_r
Password:
[root@rhel6 ~]# id -Z
root:secadm_r:secadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# semodule -d sysadm_secadm
[root@rhel6 ~]# _
```


Leave secadm_r

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# newrole -r secadm_r
Password:
[root@rhel6 ~]# id -Z
root:secadm_r:secadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# semodule -d sysadm_secadm
[root@rhel6 ~]# exit
logout
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# _
```


Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- *Part 4: Separating system and security admin roles*
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- *Part 5: Mapping sensitivities and categories to mission specific names*
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

Customizing the translation table

```
[root@rhel6 ~]# cd /usr/share/mcstrans/examples/urcsts
[root@rhel6 urcsts]# ls
README  secolor.conf  setrans.conf  urcsts.color  urcsts.test
[root@rhel6 urcsts]# cat README
Simple handling of
 UNCLASSIFIED
 RESTRICTED
 CONFIDENTIAL
 SECRET
 TOP SECRET

To use:
cp setrans.conf /etc/selinux/mls/setrans.conf
cp secolor.conf /etc/selinux/mls/
run_init /etc/init.d/mcstrans restart

To test:
/usr/share/mcstrans/util/mlstrans-test urcsts.test
/usr/share/mcstrans/util/mlscolor-test urcsts.color
[root@rhel6 urcsts]# _
```


Sample translation table

```
# UNCLASSIFIED

s0=SystemLow
s15:c0.c1023=SystemHigh

s1=UNCLASSIFIED
s1=UNCLAS
s1=U

s3=RESTRICTED
s3=R E S T R I C T E D
s3=R

s5=CONFIDENTIAL
s5=C O N F I D E N T I A L
s5=C

s7=SECRET
s7=S E C R E T
s7=S

s9=TOP SECRET
s9=T O P S E C R E T
s9=T O P  S E C R E T
setrans.conf
```


Sample colors

```
[root@rhel6 urcsts]# less secolor.conf

color black = #000000
color green = #008000
color yellow = #ffff00
color blue = #0000ff
color white = #ffffff
color red = #ff0000
color orange = #ffa500
color tan = #D2B48C

user * = black black
role * = black black
type * = black black
range s0-s0:c0.c1023 = black green
range s1-s1:c0.c1023 = black green
range s3-s3:c0.c1023 = black tan
range s5-s5:c0.c1023 = white blue
range s7-s7:c0.c1023 = black red
range s9-s9:c0.c1023 = black orange
range s15:c0.c1023 = black yellow

secolor.conf (END)
```


Using the sample translation table

```
[root@rhel6 urcsts]# cp /usr/share/mcstrans/examples/urcsts/setrans.conf /etc/se
linux/mls
cp: overwrite `/etc/selinux/mls/setrans.conf'? _
```


Using the sample translation table

```
[root@rhel6 urcsts]# cp /usr/share/mcstrans/examples/urcsts/setrans.conf /etc/se
linux/mls
cp: overwrite `/etc/selinux/mls/setrans.conf'? y
cp: cannot create regular file `/etc/selinux/mls/setrans.conf': Permission denie
d
[root@rhel6 urcsts]# _
```


Need to be secadm_r!

```
[root@rhel6 urcsts]# cp /usr/share/mcstrans/examples/urcsts/setrans.conf /etc/se
linux/mls
cp: overwrite `/etc/selinux/mls/setrans.conf'? y
cp: cannot create regular file `/etc/selinux/mls/setrans.conf': Permission denie
d
[root@rhel6 urcsts]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 urcsts]# _
```


Works after newrole to secadm_r

```
[root@rhel6 urcsts]# cp /usr/share/mcstrans/examples/urcsts/setrans.conf /etc/se
linux/mls
cp: overwrite `/etc/selinux/mls/setrans.conf'? y
cp: cannot create regular file `/etc/selinux/mls/setrans.conf': Permission denie
d
[root@rhel6 urcsts]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 urcsts]# newrole -r secadm_r
Password:
[root@rhel6 urcsts]# cp /usr/share/mcstrans/examples/urcsts/setrans.conf /etc/se
linux/mls
cp: overwrite `/etc/selinux/mls/setrans.conf'? y
[root@rhel6 urcsts]# id -Z
root:secadm_r:secadm_t:s0-s15:c0.c1023
[root@rhel6 urcsts]# _
```


Copy the sample colors and leave secadm_r

```
[root@rhel6 urcsts]# cp /usr/share/mcstrans/examples/urcsts/setrans.conf /etc/se
linux/mls
cp: overwrite `/etc/selinux/mls/setrans.conf'? y
cp: cannot create regular file `/etc/selinux/mls/setrans.conf': Permission denie
d
[root@rhel6 urcsts]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 urcsts]# newrole -r secadm_r
Password:
[root@rhel6 urcsts]# cp /usr/share/mcstrans/examples/urcsts/setrans.conf /etc/se
linux/mls
cp: overwrite `/etc/selinux/mls/setrans.conf'? y
[root@rhel6 urcsts]# id -Z
root:secadm_r:secadm_t:s0-s15:c0.c1023
[root@rhel6 urcsts]# cp /usr/share/mcstrans/examples/urcsts/secolor.conf /etc/se
linux/mls
[root@rhel6 urcsts]# exit
logout
[root@rhel6 urcsts]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 urcsts]# _
```


Load the new mapping using mcstrans

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:s0-s15:c0.c1023
[root@rhel6 ~]# run_init service mcstrans restart
Authenticating root.
Password:
Stopping mcstransd:                                        [FAILED]
Starting mcstransd:                                        [  OK  ]
[root@rhel6 ~]# run_init chkconfig mcstrans on
Authenticating root.
Password:
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# _
```


Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- *Part 5: Mapping sensitivities and categories to mission specific names*
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- *Part 6: Optionally relaxing security*
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

Optionally relaxing security

- newrole without a root password each time
- run_init without a root password each time

Add pam_rootok.so for newrole

```
#%PAM-1.0
auth       sufficient   pam_rootok.so
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    required     pam_namespace.so unmnt_remnt no_unmount_on_close
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/pam.d/newrole" 6L, 210C written
```


newrole works without root password each time

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# newrole -r secadm_r
[root@rhel6 ~]# id -Z
root:secadm_r:secadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# exit
logout
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# _
```


Add pam_rootok.so for run_init

```
#%PAM-1.0
auth       sufficient   pam_rootok.so
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
session    optional     pam_xauth.so
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/pam.d/run_init" 7L, 205C written
```


run_init works without root password each time

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# run_init service mcstrans restart
Authenticating root.
Stopping mcstransd:                                        [  OK  ]
Starting mcstransd:                                        [  OK  ]
[root@rhel6 ~]# _
```


Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- *Part 6: Optionally relaxing security*
- Part 7: Adding an MLS user
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- *Part 7: Adding an MLS user*
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

Add a user

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# useradd mlsuser
[root@rhel6 ~]# echo redhat | passwd --stdin mlsuser
Changing password for user mlsuser.
passwd: all authentication tokens updated successfully.
[root@rhel6 ~]# newrole -r secadm_r
[root@rhel6 ~]# _
```


SELinux user types

```
[root@rhel6 ~]# id -Z
root:secadm_r:secadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# semanage user -l

                Labeling   MLS/       MLS/
SELinux User    Prefix     MCS Level  MCS Range                      SELinux Rol
es

git_shell_u     user       SystemLow  SystemLow                      git_shell_r
guest_u         user       SystemLow  SystemLow                      guest_r
root            user       SystemLow  SystemLow-SystemHigh           auditadm_r
staff_r secadm_r sysadm_r system_r
staff_u         user       SystemLow  SystemLow-SystemHigh           auditadm_r
staff_r secadm_r sysadm_r system_r
sysadm_u        user       SystemLow  SystemLow-SystemHigh           sysadm_r sy
stem_r
system_u        user       SystemLow  SystemLow-SystemHigh           system_r
user_u          user       SystemLow  SystemLow                      user_r
xguest_u        user       SystemLow  SystemLow                      xguest_r
[root@rhel6 ~]# _
```


Linux login names mapped to SELinux user types

```
[root@rhel6 ~]# semanage login -l

Login Name                SELinux User                MLS/MCS Range

__default__               user_u                      SystemLow
root                      root                        SystemLow-SystemHigh
system_u                  system_u                    SystemLow-SystemHigh
[root@rhel6 ~]# _
```


Add login and assign user type and range

```
[root@rhel6 ~]# semanage login -l

Login Name                SELinux User                MLS/MCS Range

__default__               user_u                      SystemLow
root                      root                        SystemLow-SystemHigh
system_u                  system_u                    SystemLow-SystemHigh
[root@rhel6 ~]# semanage login -a -s staff_u -r U-TS mlsuser
[root@rhel6 ~]# semanage login -l

Login Name                SELinux User                MLS/MCS Range

__default__               user_u                      SystemLow
mlsuser                   staff_u                     UNCLASSIFIED-TOP SECRET
root                      root                        SystemLow-SystemHigh
system_u                  system_u                    SystemLow-SystemHigh
[root@rhel6 ~]# _
```


Log in as new user

```
Red Hat Enterprise Linux Server release 6.3 (Santiago)
Kernel 2.6.32-279.19.1.el6.x86_64 on an x86_64

rhel6 login: mlsuser
Password:
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ _
```


Create a file

```
[mlsuser@rhel6 ~]$ pwd
/home/mlsuser
[mlsuser@rhel6 ~]$ touch test
touch: cannot touch `test': Permission denied
[mlsuser@rhel6 ~]$ _
```


Can't write down!

```
[mlsuser@rhel6 ~]$ pwd
/home/mlsuser
[mlsuser@rhel6 ~]$ touch test
touch: cannot touch `test': Permission denied
[mlsuser@rhel6 ~]$ ls -Zd .
drwx------. mlsuser mlsuser user_u:object_r:user_home_dir_t:SystemLow-SystemHigh
.
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ _
```


Can read down

```
[mlsuser@rhel6 ~]$ pwd
/home/mlsuser
[mlsuser@rhel6 ~]$ touch test
touch: cannot touch `test': Permission denied
[mlsuser@rhel6 ~]$ ls -Zd .
drwx------. mlsuser mlsuser user_u:object_r:user_home_dir_t:SystemLow-SystemHigh
.
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific aliases and functions
[mlsuser@rhel6 ~]$ ls -Z .bashrc
-rw-r--r--. mlsuser mlsuser user_u:object_r:user_home_t:SystemLow .bashrc
[mlsuser@rhel6 ~]$ _
```


One fix: recursively relabel the home directory

```
[root@rhel6 ~]# id -Z
root:secadm_r:secadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# chcon -v -R -l U /home/mlsuser/
changing security context of `/home/mlsuser/.bash_logout'
changing security context of `/home/mlsuser/.bashrc'
changing security context of `/home/mlsuser/.bash_profile'
changing security context of `/home/mlsuser/'
[root@rhel6 ~]# ls -aZ /home/mlsuser/
drwx------. mlsuser mlsuser user_u:object_r:user_home_dir_t:UNCLASSIFIED .
drwxr-xr-x. root    root    system_u:object_r:home_root_t:SystemLow-SystemHigh ..
-rw-r--r--. mlsuser mlsuser user_u:object_r:user_home_t:UNCLASSIFIED .bash_logou
t
-rw-r--r--. mlsuser mlsuser user_u:object_r:user_home_t:UNCLASSIFIED .bash_profi
le
-rw-r--r--. mlsuser mlsuser user_u:object_r:user_home_t:UNCLASSIFIED .bashrc
[root@rhel6 ~]# _
```


Now writing works!

```
[mlsuser@rhel6 ~]$ pwd
/home/mlsuser
[mlsuser@rhel6 ~]$ touch test
[mlsuser@rhel6 ~]$ ls -aZ
drwx------. mlsuser mlsuser user_u:object_r:user_home_dir_t:UNCLASSIFIED .
drwxr-xr-x. root    root    system_u:object_r:home_root_t:SystemLow-SystemHigh ..
-rw-r--r--. mlsuser mlsuser user_u:object_r:user_home_t:UNCLASSIFIED .bash_logout
-rw-r--r--. mlsuser mlsuser user_u:object_r:user_home_t:UNCLASSIFIED .bash_profile
-rw-r--r--. mlsuser mlsuser user_u:object_r:user_home_t:UNCLASSIFIED .bashrc
-rw-rw-r--. mlsuser mlsuser staff_u:object_r:user_home_t:UNCLASSIFIED test
[mlsuser@rhel6 ~]$ _
```


Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- *Part 7: Adding an MLS user*
- Part 8: Using ssh
- Part 9: MLS with RHEL 6 in action!

MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- *Part 8: Using ssh*
- Part 9: MLS with RHEL 6 in action!

ssh as a non-root user

```
[mlsuser@rhel6 ~]$ ssh mlsuser@localhost
mlsuser@localhost's password:
Last login: Sun Feb 24 15:02:09 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ _
```


Escalating sensitivity within ssh

```
[mlsuser@rhel6 ~]$ ssh mlsuser@localhost
mlsuser@localhost's password:
Last login: Sun Feb 24 15:02:09 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ newrole -l S
Error: you are not allowed to change levels on a non secure terminal
[mlsuser@rhel6 ~]$ _
```


Figure out ssh's tty and the tty's SELinux type

```
[mlsuser@rhel6 ~]$ ssh mlsuser@localhost
mlsuser@localhost's password:
Last login: Sun Feb 24 15:02:09 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ newrole -l S
Error: you are not allowed to change levels on a non secure terminal
[mlsuser@rhel6 ~]$ tty
/dev/pts/0
[mlsuser@rhel6 ~]$ ls -Z /dev/pts/0
crw--w----. mlsuser tty staff_u:object_r:user_devpts_t:UNCLASSIFIED /dev/pts/0
[mlsuser@rhel6 ~]$ _
```


Add that type to securetty_types

```
console_device_t
sysadm_tty_device_t
user_tty_device_t
staff_tty_device_t
auditadm_tty_device_t
secureadm_tty_device_t
user_devpts_t
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/selinux/mls/contexts/securetty_types" 7L, 133C written
```


Now newrole within ssh works

```
[mlsuser@rhel6 ~]$ ssh mlsuser@localhost
mlsuser@localhost's password:
Last login: Sun Feb 24 15:02:09 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ newrole -l S
Error: you are not allowed to change levels on a non secure terminal
[mlsuser@rhel6 ~]$ tty
/dev/pts/0
[mlsuser@rhel6 ~]$ ls -Z /dev/pts/0
crw--w----. mlsuser tty staff_u:object_r:user_devpts_t:UNCLASSIFIED /dev/pts/0
[mlsuser@rhel6 ~]$ newrole -l S
Password:
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET-TOP SECRET
[mlsuser@rhel6 ~]$ ls -Z /dev/pts/0
crw--w----. mlsuser tty staff_u:object_r:user_devpts_t:SECRET /dev/pts/0
[mlsuser@rhel6 ~]$ _
```


ssh as a non-root user with a sensitivity

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/SECRET@localhost
mlsuser/staff_r/SECRET@localhost's password:
Last login: Sun Feb 24 15:23:46 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ _
```


Can change levels down within sensitivity range

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/SECRET@localhost
mlsuser/staff_r/SECRET@localhost's password:
Last login: Sun Feb 24 15:23:46 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ newrole -l U
Password:
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-SECRET
[mlsuser@rhel6 ~]$ _
```


Can't change levels outside sensitivity range

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/SECRET@localhost
mlsuser/staff_r/SECRET@localhost's password:
Last login: Sun Feb 24 15:23:46 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ newrole -l U
Password:
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-SECRET
[mlsuser@rhel6 ~]$ newrole -l TS
staff_u:staff_r:staff_t:TS-SECRET is not a valid context
[mlsuser@rhel6 ~]$ _
```


Can change levels up within sensitivity range

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/SECRET@localhost
mlsuser/staff_r/SECRET@localhost's password:
Last login: Sun Feb 24 15:23:46 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ newrole -l U
Password:
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-SECRET
[mlsuser@rhel6 ~]$ newrole -l TS
staff_u:staff_r:staff_t:TS-SECRET is not a valid context
[mlsuser@rhel6 ~]$ newrole -l S
Password:
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ _
```


ssh as a non-root user with a sensitivity range

```
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/C-TS@localhost
mlsuser/staff_r/C-TS@localhost's password:
Last login: Sun Feb 24 15:22:48 2013 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:CONFIDENTIAL-TOP SECRET
[mlsuser@rhel6 ~]$ _
```


Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- *Part 8: Using ssh*
- Part 9: MLS with RHEL 6 in action!

MULTILEVEL SECURITY (MLS) WITH RED HAT ENTERPRISE LINUX 6 AND SELINUX

David Egts, RHCA, RHCSS
Principal Architect
@davidegts

redhat.

Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- *Part 9: MLS with RHEL 6 in action!*

Set up secret and top_secret directories

```
[root@rhel6 ~]# id -Z
root:sysadm_r:sysadm_t:SystemLow-SystemHigh
[root@rhel6 ~]# mkdir -p /data/secret /data/top_secret
[root@rhel6 ~]# chmod 777 /data/secret /data/top_secret
[root@rhel6 ~]# newrole -r secadm_r
[root@rhel6 ~]# chcon -R -t user_home_t /data
[root@rhel6 ~]# chcon -l S /data/secret
[root@rhel6 ~]# chcon -l "TOP SECRET" /data/top_secret
[root@rhel6 ~]# ls -Z /data
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
drwxrwxrwx. root root root:object_r:user_home_t:TOP SECRET top_secret
[root@rhel6 ~]# exit
logout
[root@rhel6 ~]# _
```


ssh as mlsuser at the Secret level

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/S@localhost
mlsuser/staff_r/S@localhost's password:
Last login: Thu Dec 27 15:48:19 2012 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ _
```


ssh as mlsuser at the Secret level

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/S@localhost
mlsuser/staff_r/S@localhost's password:
Last login: Thu Dec 27 15:48:19 2012 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ cd /data
[mlsuser@rhel6 data]$ ls -Z
ls: cannot access top_secret: Permission denied
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
?---------  ?    ?                                     top_secret
[mlsuser@rhel6 data]$ _
```


Secret can write to Secret area

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/S@localhost
mlsuser/staff_r/S@localhost's password:
Last login: Thu Dec 27 15:48:19 2012 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ cd /data
[mlsuser@rhel6 data]$ ls -Z
ls: cannot access top_secret: Permission denied
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
?---------  ?    ?                                     top_secret
[mlsuser@rhel6 data]$ cd secret
[mlsuser@rhel6 secret]$ echo secrets > secret.txt
[mlsuser@rhel6 secret]$ cat secret.txt
secrets
[mlsuser@rhel6 secret]$ _
```


Can't read up

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/S@localhost
mlsuser/staff_r/S@localhost's password:
Last login: Thu Dec 27 15:48:19 2012 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ cd /data
[mlsuser@rhel6 data]$ ls -Z
ls: cannot access top_secret: Permission denied
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
?---------  ?    ?                                     top_secret
[mlsuser@rhel6 data]$ cd secret
[mlsuser@rhel6 secret]$ echo secrets > secret.txt
[mlsuser@rhel6 secret]$ cat secret.txt
secrets
[mlsuser@rhel6 secret]$ cd ../top_secret
-bash: cd: ../top_secret: Permission denied
[mlsuser@rhel6 secret]$ _
```


Can't write up (write equality only!)

```
[mlsuser@rhel6 ~]$ ssh mlsuser/staff_r/S@localhost
mlsuser/staff_r/S@localhost's password:
Last login: Thu Dec 27 15:48:19 2012 from localhost
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:SECRET
[mlsuser@rhel6 ~]$ cd /data
[mlsuser@rhel6 data]$ ls -Z
ls: cannot access top_secret: Permission denied
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
?---------  ?    ?                                     top_secret
[mlsuser@rhel6 data]$ cd secret
[mlsuser@rhel6 secret]$ echo secrets > secret.txt
[mlsuser@rhel6 secret]$ cat secret.txt
secrets
[mlsuser@rhel6 secret]$ cd ../top_secret
-bash: cd: ../top_secret: Permission denied
[mlsuser@rhel6 secret]$ echo secrets > ../top_secret/secret.txt
-bash: ../top_secret/secret.txt: Permission denied
[mlsuser@rhel6 secret]$ _
```


ssh as mlsuser at the Top Secret level

```
[mlsuser@rhel6 ~]$ ssh "mlsuser/staff_r/TOP SECRET"@localhost
mlsuser/staff_r/TOP SECRET@localhost's password:
Last login: Mon Dec 31 15:01:00 2012
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:TOP SECRET
[mlsuser@rhel6 ~]$ _
```


ssh as mlsuser at the Top Secret level

```
[mlsuser@rhel6 ~]$ ssh "mlsuser/staff_r/TOP SECRET"@localhost
mlsuser/staff_r/TOP SECRET@localhost's password:
Last login: Mon Dec 31 15:01:00 2012
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:TOP SECRET
[mlsuser@rhel6 ~]$ cd /data
[mlsuser@rhel6 data]$ ls -Z
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
drwxrwxrwx. root root root:object_r:user_home_t:TOP SECRET top_secret
[mlsuser@rhel6 data]$ _
```


Top Secret can write to Top Secret area

```
[mlsuser@rhel6 ~]$ ssh "mlsuser/staff_r/TOP SECRET"@localhost
mlsuser/staff_r/TOP SECRET@localhost's password:
Last login: Mon Dec 31 15:01:00 2012
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:TOP SECRET
[mlsuser@rhel6 ~]$ cd /data
[mlsuser@rhel6 data]$ ls -Z
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
drwxrwxrwx. root root root:object_r:user_home_t:TOP SECRET top_secret
[mlsuser@rhel6 data]$ echo secrets > top_secret/top_secret.txt
[mlsuser@rhel6 data]$ cat top_secret/top_secret.txt
secrets
[mlsuser@rhel6 data]$ _
```


Can't write down

```
[mlsuser@rhel6 ~]$ ssh "mlsuser/staff_r/TOP SECRET"@localhost
mlsuser/staff_r/TOP SECRET@localhost's password:
Last login: Mon Dec 31 15:01:00 2012
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:TOP SECRET
[mlsuser@rhel6 ~]$ cd /data
[mlsuser@rhel6 data]$ ls -Z
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
drwxrwxrwx. root root root:object_r:user_home_t:TOP SECRET top_secret
[mlsuser@rhel6 data]$ echo secrets > top_secret/top_secret.txt
[mlsuser@rhel6 data]$ cat top_secret/top_secret.txt
secrets
[mlsuser@rhel6 data]$ echo secrets > secret/top_secret.txt
-bash: secret/top_secret.txt: Permission denied
[mlsuser@rhel6 data]$ _
```


Can read down

```
[mlsuser@rhel6 ~]$ ssh "mlsuser/staff_r/TOP SECRET"@localhost
mlsuser/staff_r/TOP SECRET@localhost's password:
Last login: Mon Dec 31 15:01:00 2012
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:TOP SECRET
[mlsuser@rhel6 ~]$ cd /data
[mlsuser@rhel6 data]$ ls -Z
drwxrwxrwx. root root root:object_r:user_home_t:SECRET secret
drwxrwxrwx. root root root:object_r:user_home_t:TOP SECRET top_secret
[mlsuser@rhel6 data]$ echo secrets > top_secret/top_secret.txt
[mlsuser@rhel6 data]$ cat top_secret/top_secret.txt
secrets
[mlsuser@rhel6 data]$ echo secrets > secret/top_secret.txt
-bash: secret/top_secret.txt: Permission denied
[mlsuser@rhel6 data]$ cat secret/secret.txt
secrets
[mlsuser@rhel6 data]$ _
```


ssh as mlsuser without a sensitivity

```
[mlsuser@rhel6 ~]$ ssh mlsuser@localhost
mlsuser@localhost's password:
Last login: Tue Jan  8 12:22:21 2013
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ _
```


Can't read up

```
[mlsuser@rhel6 ~]$ ssh mlsuser@localhost
mlsuser@localhost's password:
Last login: Tue Jan  8 12:22:21 2013
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ ls /data
ls: cannot access /data/top_secret: Permission denied
ls: cannot access /data/secret: Permission denied
secret  top_secret
[mlsuser@rhel6 ~]$ _
```


Escalate sensitivity to TS

```
[mlsuser@rhel6 ~]$ ssh mlsuser@localhost
mlsuser@localhost's password:
Last login: Tue Jan  8 12:22:21 2013
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ ls /data
ls: cannot access /data/top_secret: Permission denied
ls: cannot access /data/secret: Permission denied
secret  top_secret
[mlsuser@rhel6 ~]$ newrole -l TS
Password:
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:TOP SECRET
[mlsuser@rhel6 ~]$ _
```


And read down works

```
[mlsuser@rhel6 ~]$ ssh mlsuser@localhost
mlsuser@localhost's password:
Last login: Tue Jan  8 12:22:21 2013
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:UNCLASSIFIED-TOP SECRET
[mlsuser@rhel6 ~]$ ls /data
ls: cannot access /data/top_secret: Permission denied
ls: cannot access /data/secret: Permission denied
secret  top_secret
[mlsuser@rhel6 ~]$ newrole -l TS
Password:
[mlsuser@rhel6 ~]$ id -Z
staff_u:staff_r:staff_t:TOP SECRET
[mlsuser@rhel6 ~]$ cat /data/top_secret/top_secret.txt
secrets
[mlsuser@rhel6 ~]$ cat /data/secret/secret.txt
secrets
[mlsuser@rhel6 ~]$ _
```


Overview

- Part 1: Background on MLS
- Part 2: MLS with RHEL 6
- Part 3: Basic setup of MLS with RHEL 6
- Part 4: Separating system and security admin roles
- Part 5: Mapping sensitivities and categories to mission specific names
- Part 6: Optionally relaxing security
- Part 7: Adding an MLS user
- Part 8: Using ssh
- *Part 9: MLS with RHEL 6 in action!*

References

- The SELinux Notebook
  - http://www.freetechbooks.com/the-selinux-notebook-the-foundations-t785.html
- Red Hat Enterprise Linux 6 Security-Enhanced Linux User Guide
  - https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html-single/Security-Enhanced_Linux/index.html
- Confining Users with SELinux
  - https://access.redhat.com/knowledge/videos/214723

Special thanks

- Dan Walsh
  - http://danwalsh.livejournal.com/
  - @rhatdan
- Paul Moore
  - http://paulmoore.livejournal.com/
  - @paul_via_tweet
- Ted Brunell
- Rick Ring
- Bob St. Clair
- Mark St. Laurent