SUMMIT

JBoss WORLD

PRESENTED BY RED HAT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

Red Hat Enterprise Linux 6
Writing SELinux Policy

Dan Walsh
May 5, 2011

SELinux Labels

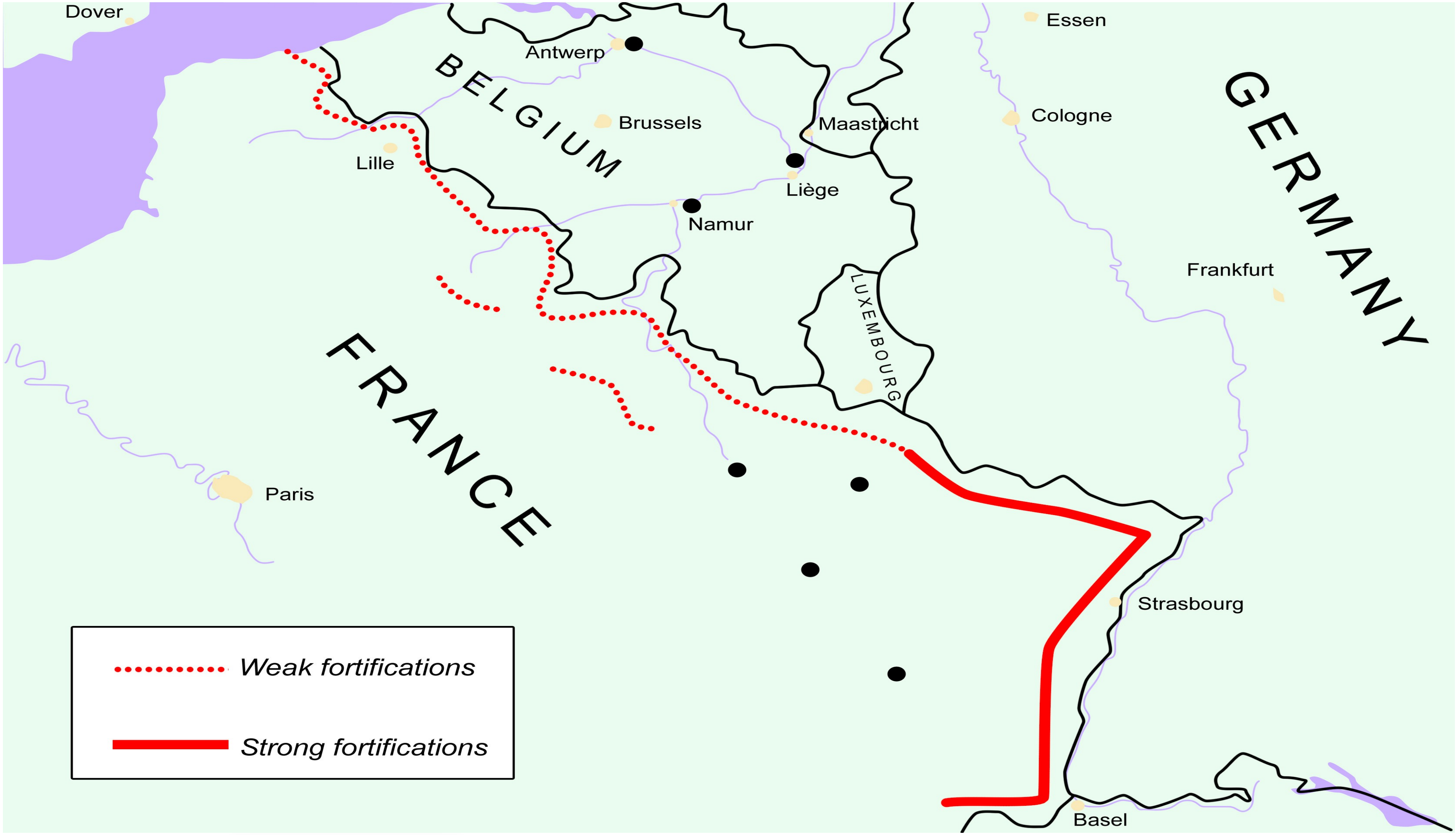
staff_u:staff_r:staff_t:s0-s0:c0.c1023

staff_u:webadm_r:webadm_t:s0

system_u:object_r:dictd_exec_t:s0

system_u:system_r:dictd_t:SystemHigh

Security Goals



http://en.wikipedia.org/wiki/Maginot_line

Policy Modules

Three Components

Type Enforcement (te) File

Contains all the rules used to confine your application

File Context (fc) File

Contains the regular expression mappings for on disk file contexts

Interface (if) Files

Contains the interfaces defined for other confined applications, to interact with your confined application

Policy Package (pp)

Compiler/packager roles generates policy package to be installed on systems.

Type Enforcing File - Language

Name the module

policy_module(dictd,1.0)

M4 macro grabs all definitions for classes, perms

Type Field
system_u:system_r:dictd_t:s0

```
type dictd_t;  # Process Type (domain)
type dictd_exec_t; # File Type (file_type)
gen_require(`
        type shadow_t;
')
```

COMMAND SOURCETYPE TARGETTYPE : CLASS PERMS;

- allow
  - Most common
  - Everything denied by default
- dontaudit
  - Deny Access, but do not log
- auditallow
  - Allow access but log a message
- neverallow
  - Conflicting rule will cause policy install to fail

COMMAND SOURCETYPE TARGETTYPE : CLASS PERMS;

- SOURCETYPE
  - Always a process type.
- TARGETTYPE
  - Object-type
  - Usually a file type
  - Sometimes another process
    - self is a special field
- allow { httpd_t httpd_sys_script_t } etc_t : file read;

COMMAND SOURCETYPE TARGETTYPE:CLASS PERMS;

- Classification of different target objects
- file, dir, sock_file, tcp_socket, process, capability

COMMAND SOURCETYPE TARGETTYPE:CLASS PERMS;

Permissions differ per class

- file - { read write append ... }
- process { fork signal sigkill ...}
- capability { setuid setgid ... }

Macro definitions

- Combine multiple different permissions for one logical access.
- read_file_perms, manage_sock_file_perms;

Common file patterns

- read_files_pattern(httpd_t, etc_t, net_conf_t)
- /usr/share/selinux/devel/include/support/obj_perm_sets.spt

Interfaces

Policy Function Calls

- Allow other domains to interact with your types
- /usr/share/selinux/devel/include/kernel/files.if
- Examples
  - mysql_stream_connect(httpd_t)
  - init_system_domain(dictd_t, dictd_exec_t)
  - corenet_tcp_connect_mssql_port(httpd_php_t)
  - apache_admin(webadm_t)

Attributes

- Group types together
  - attribute file_type
  - type etc_t, file_type
- Use as Source or Target
  - allow rpm_t file_type:file manage_file_perms;
  - allow domain self:process fork;
- Interfaces used to assign attributes:
  - files_type(etc_t)
  - domain_type(httpd_t)

Permissive Domains

- permissive dictd_t;
  - dictd_t will be allowed full access to the system, but will generate AVC messages.
  - “Learning Mode”

Control Writing

How does one process attack another?

- **Writing**
- Your domain owns the data?
  - Create a new type
- The data is labeled as system data
  - etc_r, usr_t, var_lib_t, var_run_t, root_t
  - Never write, you need a file trans type
- Data owned by another confined domain?
  - httpd_sys_content_t?
  - apache_write_content(dictd_t)

Process Class

```
allow guest_t guest_t : process sigkill;
allow guest_t self : process sigkill;
allow guest_t guest_dbusd_t : process sigkill;
allow guest_t httpd_t : process sigkill;
```

Capability Class

- Attempt to limit power of root
  - ~34 Capabilities
  - Explained in /usr/include/linux/capability.h
- dac_override, net_bind_service, setuid, kill
- allow ping_t self:capability net_raw;

Transitions

File Transition

```
filetrans_pattern(dictd_t, var_run_t, { file dir }, dictd_var_run_t)
```

Process Transitions:

```
allow dictd_t sendmail_exec_t:file { execute read ... }
can_exec(dictd_t, sendmail_exec_t)
sendmail_domtrans(dictd_t)
```

Using Modules

Makefile

```
# make -f /usr/share/selinux/devel/Makefile
```

Install

```
# semodule -i dictd.pp
```

Assigning file context

```
# restorecon -R /var/run/dictd.pid
```

Work Flow

Lather Rinse Repeat

Test application

Generate avc messages

audit2allow -lar >> dictd.te

make -f /usr/share/selinux/devel/Makefile

semodule -i dictd.pp

Audit2allow

```
# ausearch -m avc -ts recent -i
type=SYSCALL msg=audit(04/22/2011 11:53:51.194:49) : arch=i386 syscall=open success=yes exit=4 a0=2d89b8 a1=0
a2=b77ac910 a3=3 items=0 ppid=7694 pid=7695 auid=Tim uid=root gid=nobody euid=root suid=root fsuid=root
egid=nobody sgid=nobody fsgid=nobody tty=pts3 ses=1 comm=dictd exe=/usr/sbin/dictd
subj=unconfined_u:system_r:dictd_t:s0 key=(null)

type=AVC msg=audit(04/22/2011 11:53:51.194:49) : avc:  denied  { read } for  pid=7695 comm=dictd
scontext=unconfined_u:system_r:dictd_t:s0 tcontext=system_u:object_r:sysctl_kernel_t:s0 tclass=file
```

```
# audit2allow -la
allow dictd_t sysctl_kernel_t:file read;

# audit2allow -laR
require {
     type dictd_t;
}
#============= dictd_t ==============
kernel_read_kernel_sysctls(dictd_t)
```

MOST IMPORTANT THING TO LEARN TODAY

audit2allow – Just MAKE IT WORK?????

```
# audit2allow -M myprelink -R -i
/var/log/audit/audit.log
******************** IMPORTANT ***********************
To make this policy package active, execute:
semodule -i myprelink.pp

# ls myprelink*
myprelink.fc  myprelink.if  myprelink.pp  myprelink.te
```

SUMMIT JBoss WORLD

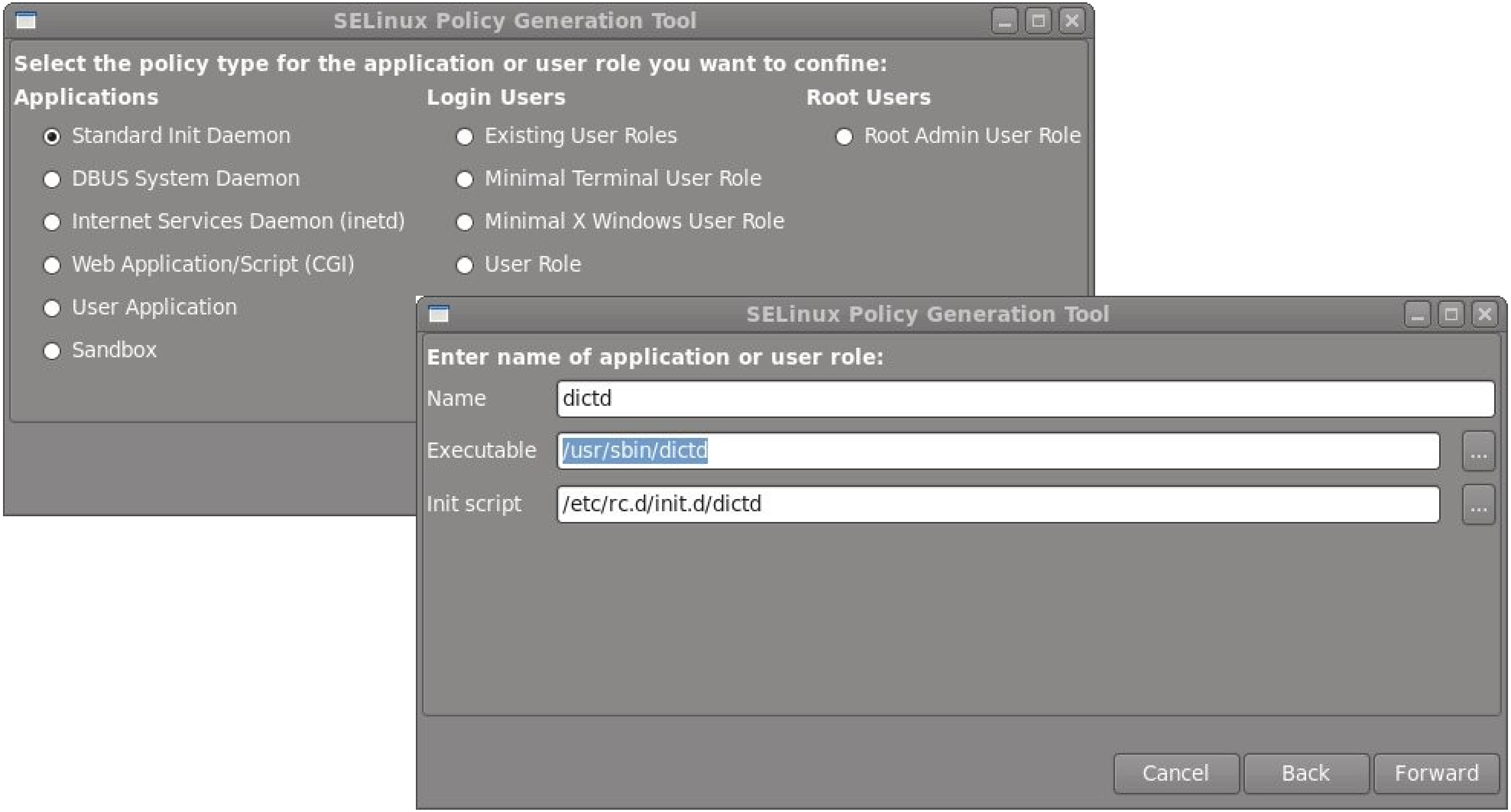
PRESENTED BY RED HAT

sepolgen

```
# sepolgen /usr/sbin/dictd
Created the following files:
Type Enforcement file    ./dictd.te
Interface file           ./dictd.if
File Contexts file       ./dictd.fc
Setup Script            ./dictd.sh
# sh dictd.sh
```

Lets Start Generating Policy

selinux-polgengui



SELinux Policy Generation Tool

Select the policy type for the application or user role you want to confine:

Applications

- ● Standard Init Daemon
- ○ DBUS System Daemon
- ○ Internet Services Daemon (inetd)
- ○ Web Application/Script (CGI)
- ○ User Application
- ○ Sandbox

Login Users

- ○ Existing User Roles
- ○ Minimal Terminal User Role
- ○ Minimal X Windows User Role
- ○ User Role

Root Users

- ○ Root Admin User Role

SELinux Policy Generation Tool

Enter name of application or user role:

Name: dictd

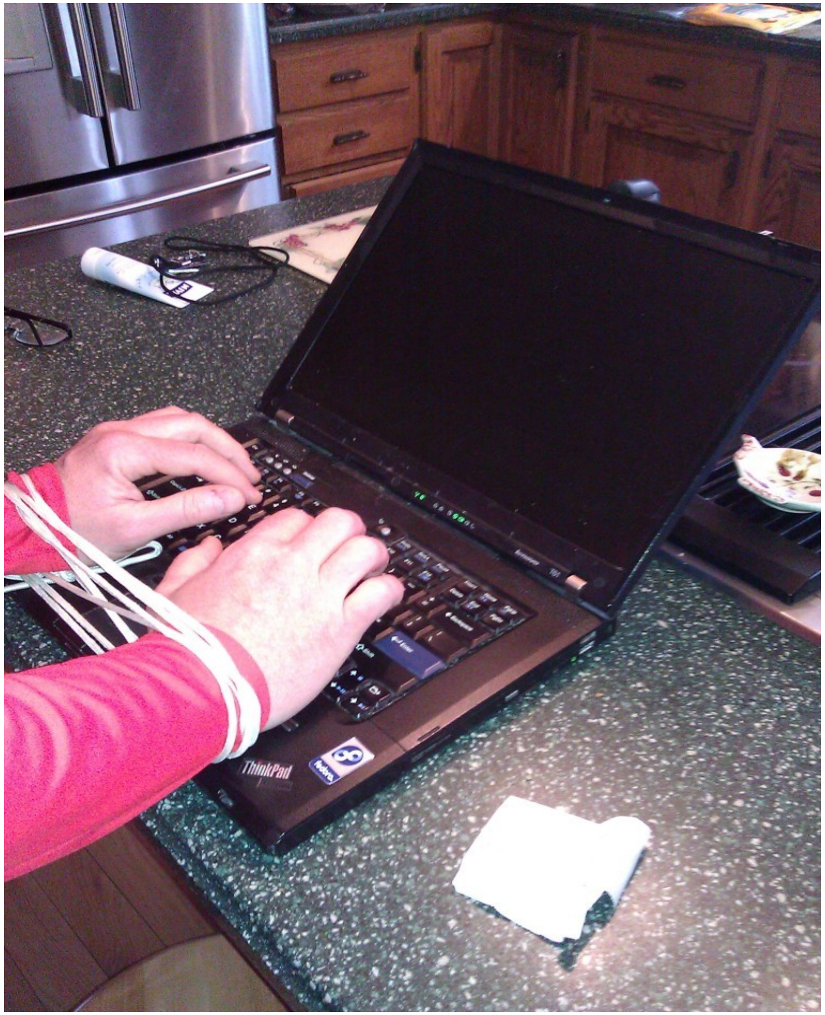
Executable: /usr/sbin/dictd

Init script: /etc/rc.d/init.d/dictd

Now your turn, I want you to confine rwhod.
Hint, it listens on port 513.

Confining Administrators

- RBAC
  - Roles Based Access Control
- At most two roles:
  - User Role - Always
  - Administrator Role
    - Role as root.



SELinux USER Selection
staff_u:webadm_r:webadm_t:s0

- Linux User :
  - dwalsh
  - root
  - __default__
- SELinux User :
  - staff_u
  - unconfined_u
  - guest_u
  - xguest_u

photo by _Zeta_ on Flickr

SELinux ROLE Selection

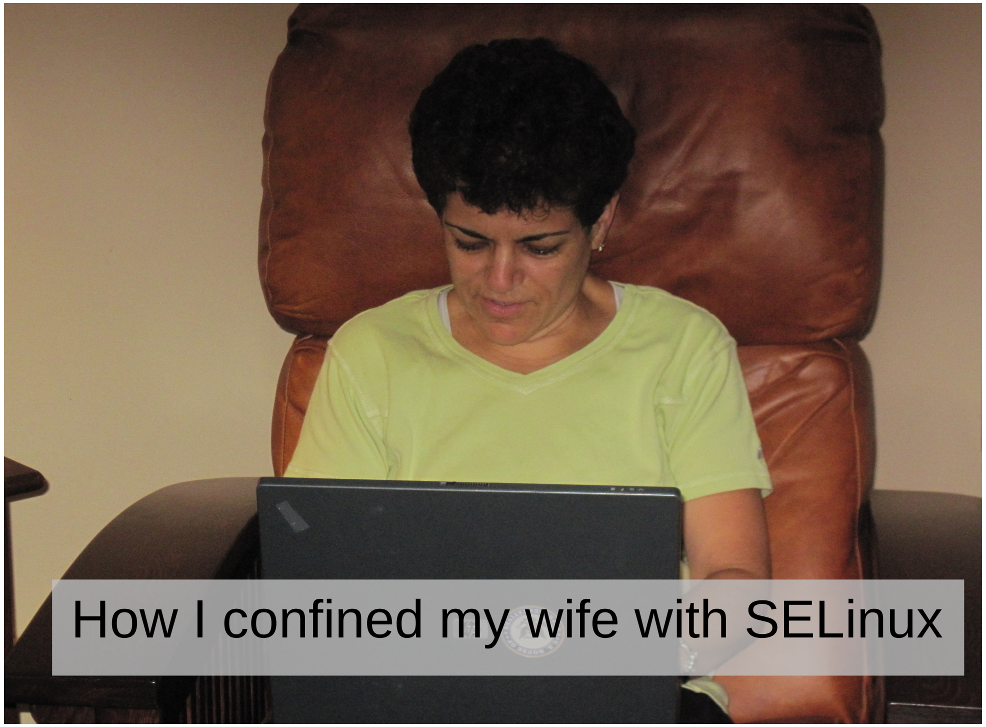
- SELinux User : dwalsh
  - staff_u
- SELinux Roles : staff_u
  - staff_r webadm_r system_r
    - SELinux Types: staff_r
      - staff_t sudo_staff_t

photo by _Zeta_ on Flickr

Confining User

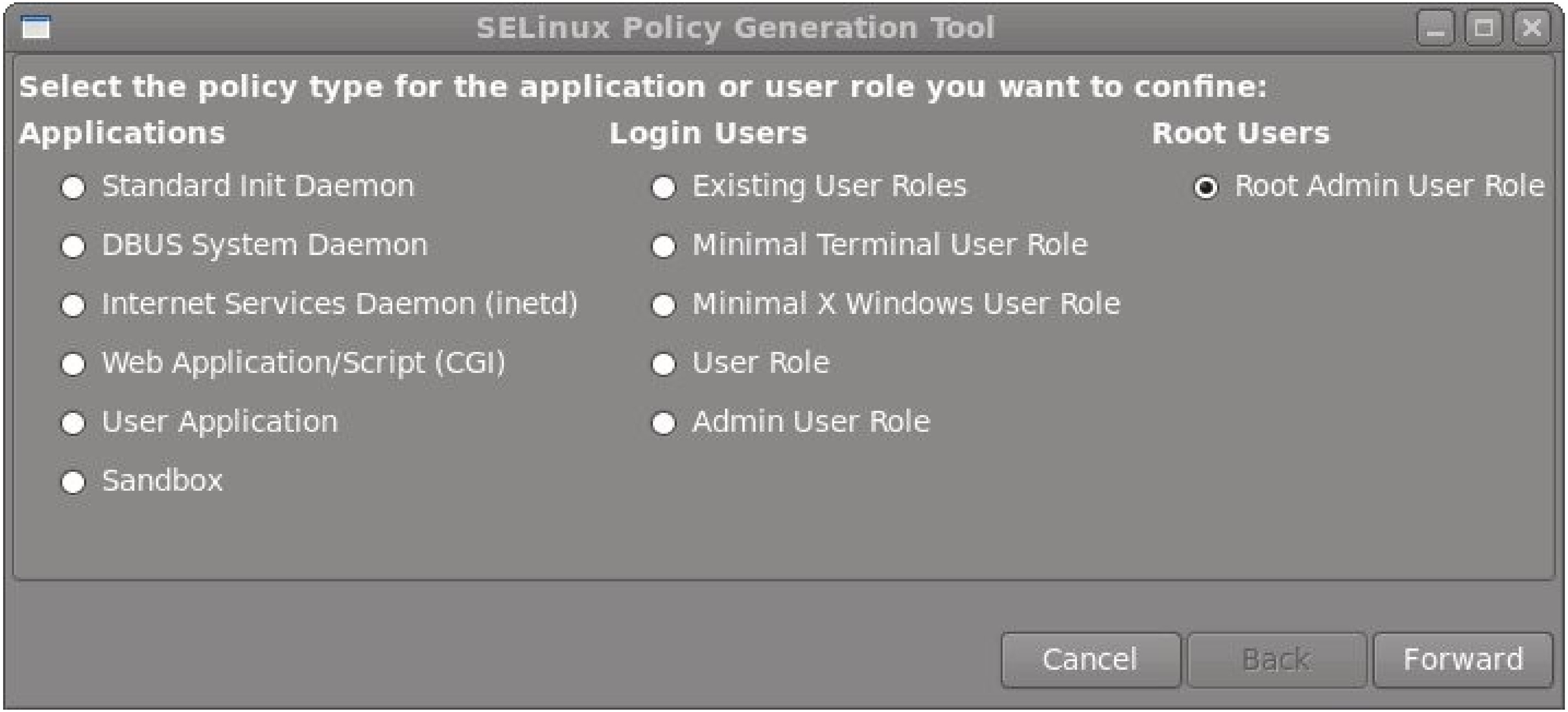
Step One

- Confine User
- Unconfined user/confined admin possible
- Can't know password



How I confined my wife with SELinux

```
# semanage login -a -s staff_u dwalsh
# semanage login -m -s user_u  __default__
```



Select the policy type for the application or user role you want to confine:

Applications

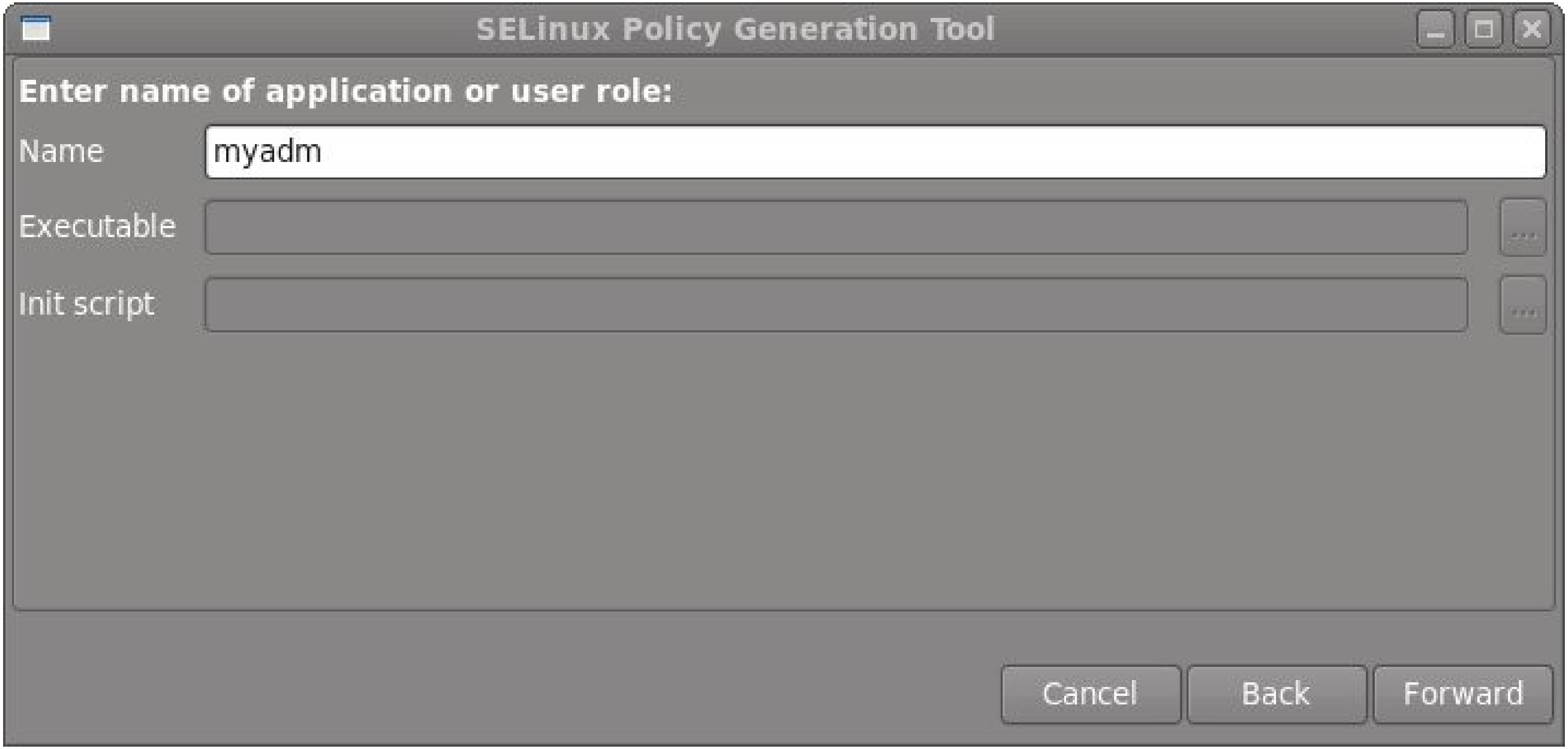
- ○ Standard Init Daemon
- ○ DBUS System Daemon
- ○ Internet Services Daemon (inetd)
- ○ Web Application/Script (CGI)
- ○ User Application
- ○ Sandbox

Login Users

- ○ Existing User Roles
- ○ Minimal Terminal User Role
- ○ Minimal X Windows User Role
- ○ User Role
- ○ Admin User Role

Root Users

- ● Root Admin User Role

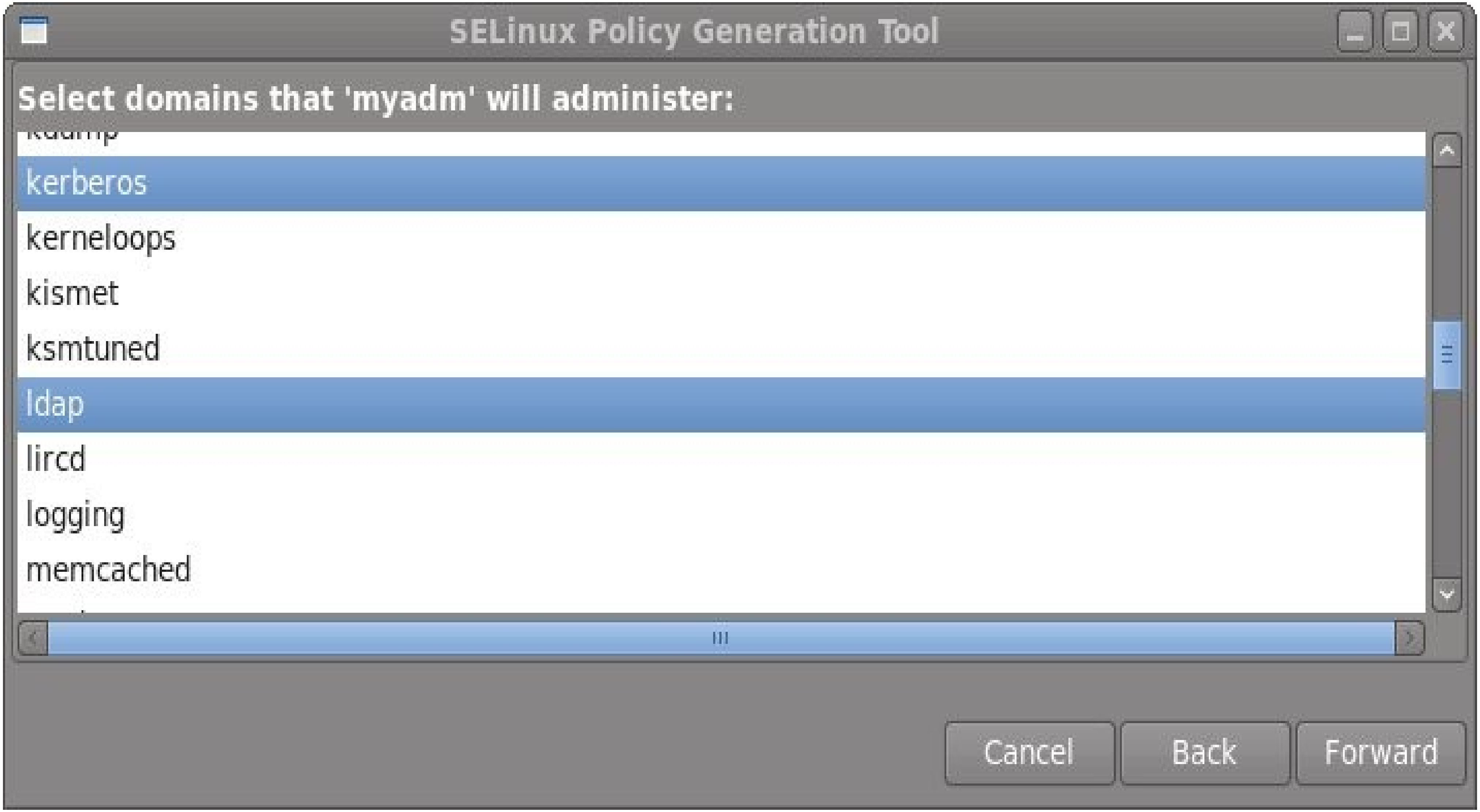


Enter name of application or user role:

Name: myadm

Executable:

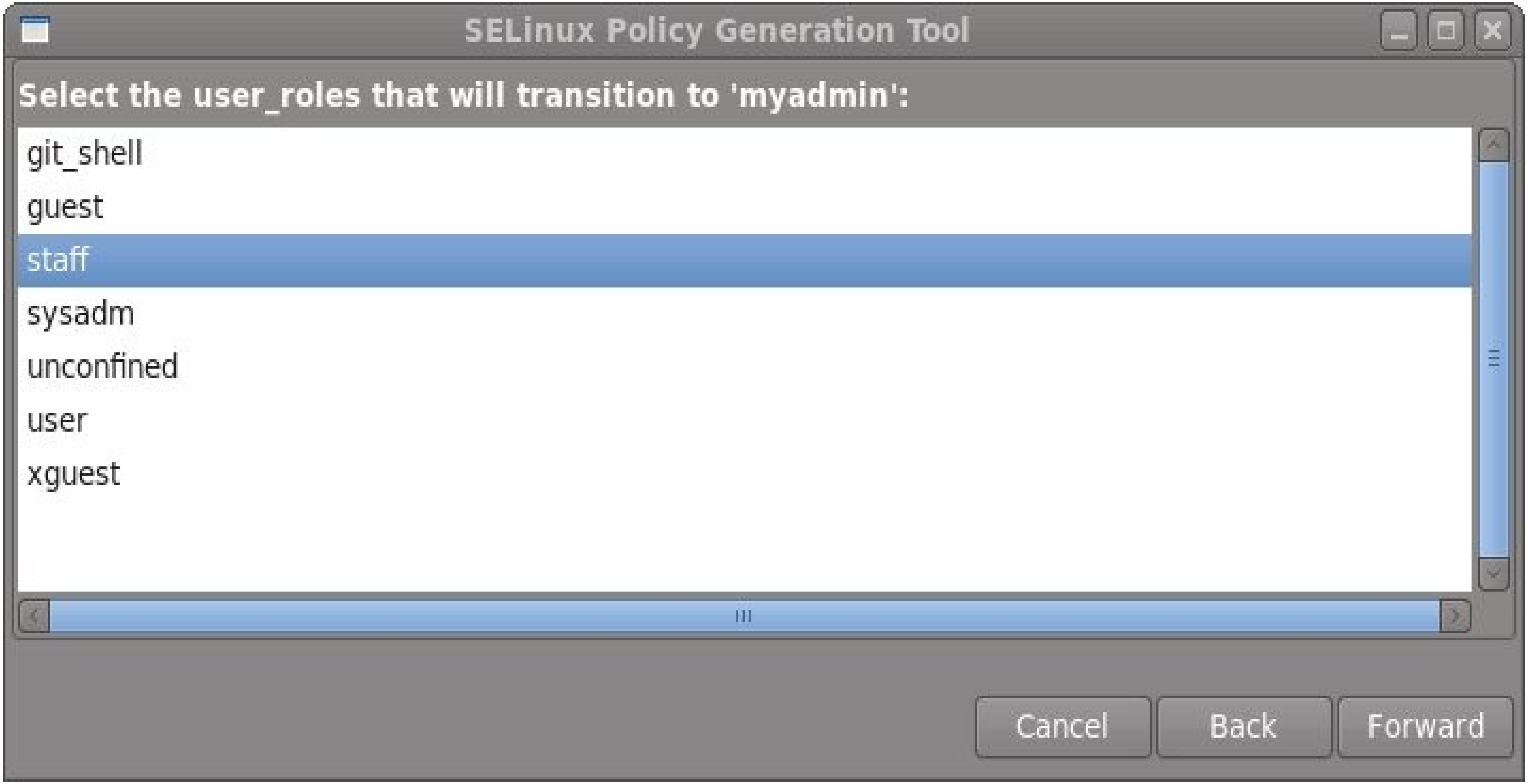
Init script:



SELinux Policy Generation Tool

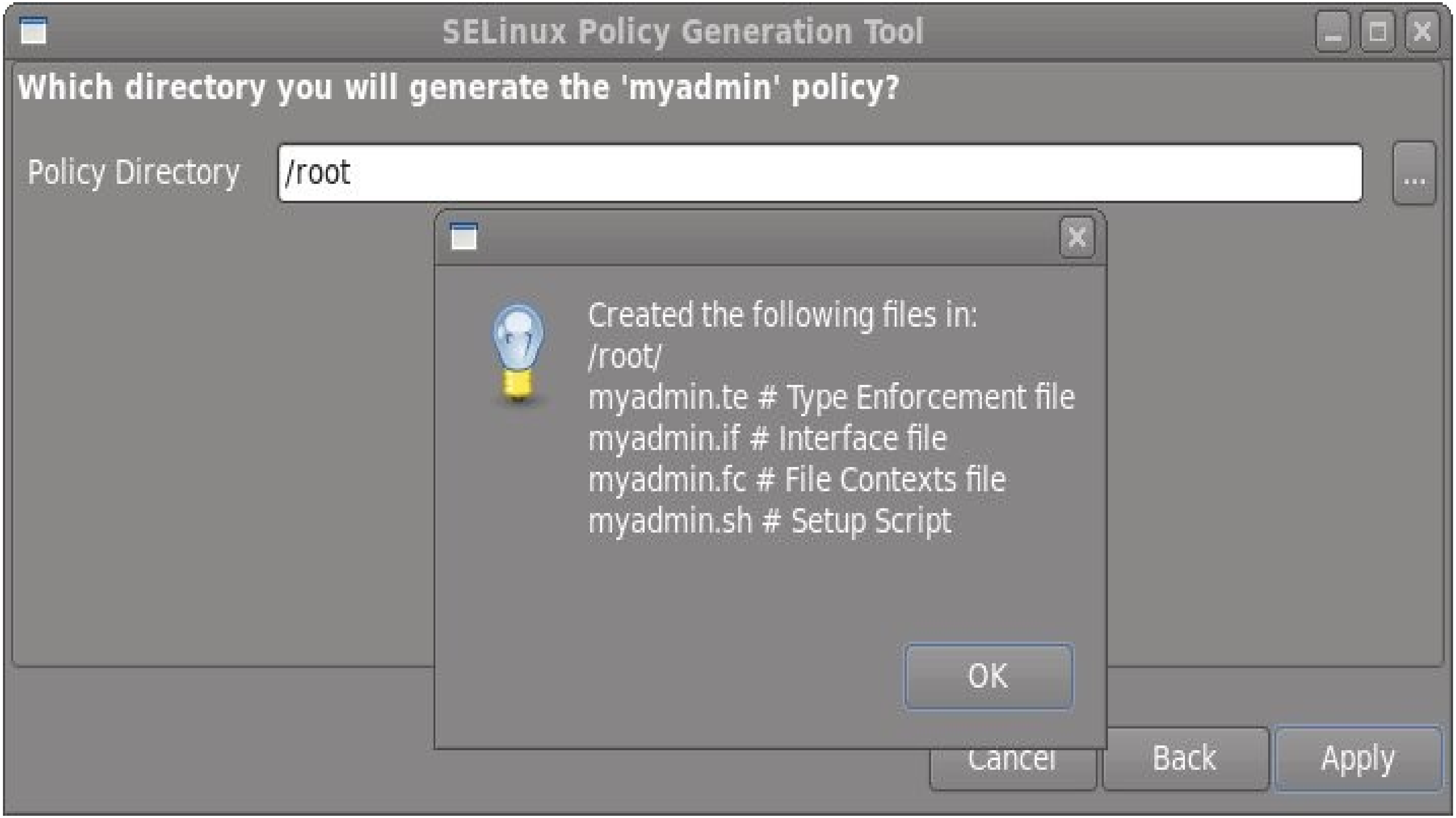
Select domains that 'myadm' will administer:

kdump
kerberos
kerneloops
kismet
ksmtuned
ldap
lircd
logging
memcached



Select the user_roles that will transition to 'myadmin':

- git_shell
- guest
- staff
- sysadm
- unconfined
- user
- xguest



Which directory you will generate the 'myadmin' policy?

Policy Directory: /root

Created the following files in:
/root/
myadmin.te # Type Enforcement file
myadmin.if # Interface file
myadmin.fc # File Contexts file
myadmin.sh # Setup Script

Compile and install policy

```
# sh ./myadmin.sh
Building and Loading Policy
+ make -f /usr/share/selinux/devel/Makefile
Compiling targeted myadmin module
/usr/bin/checkmodule:  loading policy configuration from tmp/myadmin.tmp
/usr/bin/checkmodule:  policy configuration loaded
/usr/bin/checkmodule:  writing binary representation (version 10) to tmp/myadmin.mod
Creating targeted myadmin.pp policy package
rm tmp/myadmin.mod.fc tmp/myadmin.mod
+ /usr/sbin/semodule -i myadmin.pp
```

Add roles to confined user

```
# semanage user -m -R "staff_r sysadm_r system_r myadm_r" staff_u

# semanage user -a -R "staff_r system_r myadm_t" myadm_u
```

Add line to /etc/sudoers to allow dwalsh root access

```
# visudo
dwalsh ALL=(ALL) TYPE=myadmin_t ROLE=myadmin_r ALL
```

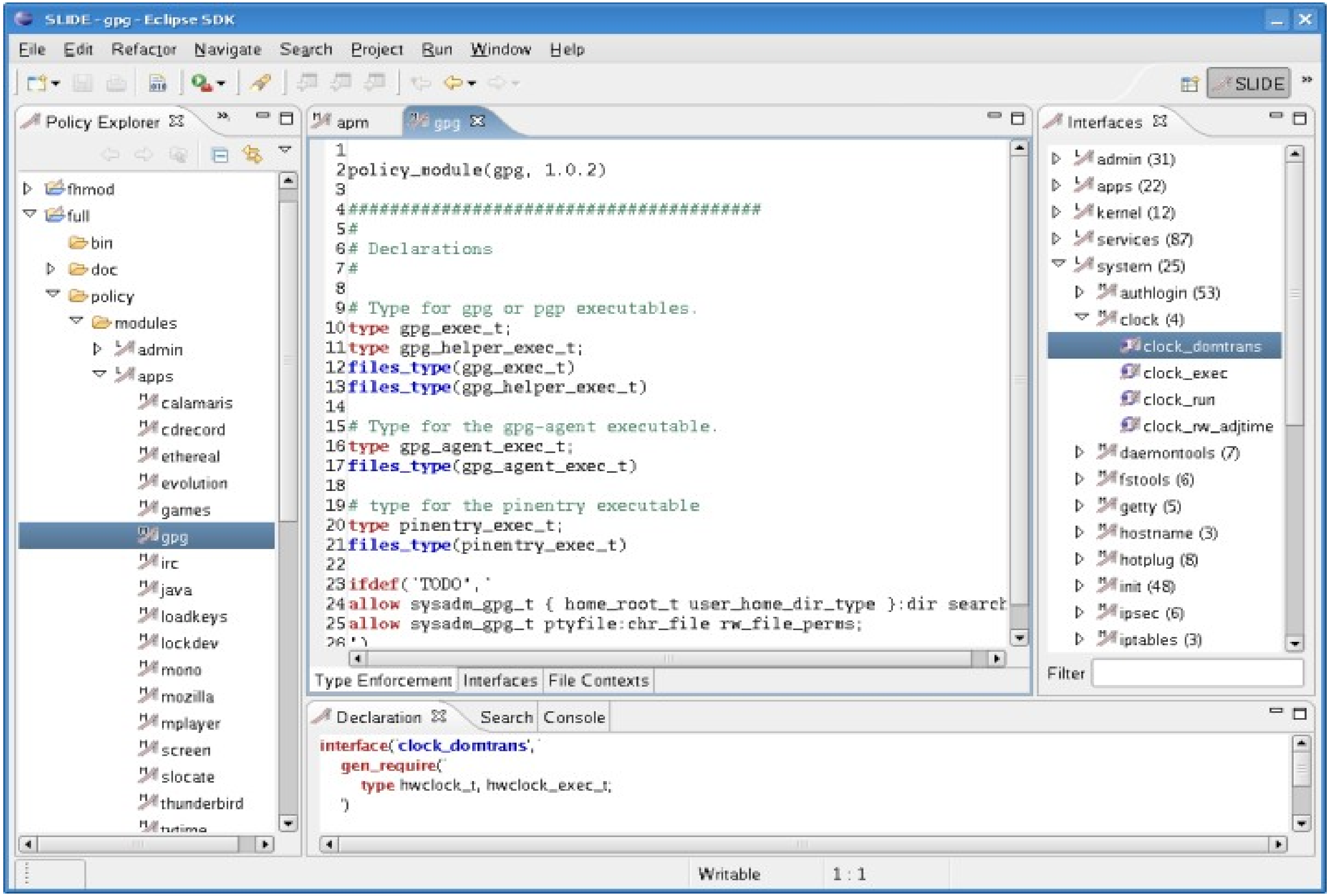
Login as dwalsh

```
# ssh dwalsh@locahost
> id -Z
staff_u:staff_r:staff_t:s0-s0:c0.c1023

> sudo sh
# id -Z
staff_u:myadmin_r:myadm_r:s0-s0:c0.c1023
```

I want you to build an admin who can manage httpd and databases.

Slide - Eclipse Policy Editor



```

policy_module(gpg, 1.0.2)

############################################
#
# Declarations
#

# Type for gpg or pgp executables.
type gpg_exec_t;
type gpg_helper_exec_t;
files_type(gpg_exec_t)
files_type(gpg_helper_exec_t)

# Type for the gpg-agent executable.
type gpg_agent_exec_t;
files_type(gpg_agent_exec_t)

# type for the pinentry executable
type pinentry_exec_t;
files_type(pinentry_exec_t)

ifdef(`TODO',`
allow sysadm_gpg_t { home_root_t user_home_dir_type }:dir search
allow sysadm_gpg_t ptyfile:chr_file rw_file_perms;
')
```

```
interface(`clock_domtrans',`
  gen_require(`
    type hwclock_t, hwclock_exec_t;
  ')
```

LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey