

開放的  
열린  
مفتوح  
libre  
মুক্ত  
ମୁକ୍ତ  
livre  
libero  
ମୁକ୍ତ  
开放的  
açık  
open  
nyílt  
ଓଡ଼ିଆ  
ଓଡ଼ିଆ  
オーブン  
livre  
ανοικτό  
offen  
otevřený  
öppen  
открытый  
ବେଳିପ୍ଲଟେ

# open



USE



IMPROVE



EVANGELIZE

## Flexible Mandatory Access Control (FMAC)

John Weeks  
Senior Information Assurance Architect  
Sun Microsystems, Inc.



# Agenda

- Design Goals
- Early Stages
- Timeline of Completed Tasks
- Implementation
- Examples
- Next Steps
- Community
- Q&A

# Design Goals

- Bring the Flask architecture and type enforcement (TE) to the OpenSolaris™ Operating System
- Complement existing Solaris™ security mechanisms
- Preserve existing Solaris APIs
- Provide Flask-compatible APIs
- Specify a single policy for a system

# Early Stages

- Provide project foundation
  - Create opensolaris.org project page and fmac-discuss discussion list
  - Set up Mercurial project repository
  - Create code contribution charter
- Provide development foundation
  - Discuss initial design concepts on list
  - Integrate Flask/TE v15 into ONNV
  - Add process and file context support
  - Add library and utility support

## Timeline of Completed Tasks

- Project proposal submitted 02/14/2008
- Project proposal approved 02/14/2008
- Project site created 03/04/2008
- Press release 03/13/2008
- Jonathan Schwartz blog 03/25/2008
- Alpha 1 source code drop 05/02/2008
- System call support 06/20/2008
- Process context support 07/10/2008



# Implementation – Utilities

- checkpolicy
- loadpolicy
- getenforce
- setenforce
- setfiles
- pcon



# Implementation – libc Interfaces

- security\_load\_policy()
- security\_compute\_av()
- security\_check\_context()
- security\_getenforce(),  
  security\_setenforce()
- is\_fmac\_enabled()
- getcon(), getpidcon()
- getexeccon(), setexeccon()
- getprevcon()
- freecon()



# Implementation - /etc/system Options

- set fmac\_enabled = [0,1]
- set fmac\_enforcing = [0,1]
- set fmac\_default\_policy\_file = "/etc/security/fmac/ss\_policy"



# Implementation – Boot Flags

- -p [disabled|enforcing|permissive]



# Implementation – Source Structure

- usr/src/head/fmac
  - User header files
- usr/src/common/fmac
  - Code shared by user-space and the kernel
- usr/src/cmd/fmac
  - FMAC-specific commands e.g., checkpolicy
- usr/src/uts/common/sys/fmac
  - FMAC-specific kernel header files
- usr/src/uts/common/fmac
  - Kernel files



# Example – Process Contexts

The screenshot shows a terminal window titled "Terminal". The window contains the following text output from a shell session:

```
# uname -a
SunOS zone1 5.11 fmac-proc i86pc i386 i86pc
# zonename
zone1
# /sbin/getenforce
permissive
# pcon $$

101745: system_u:system_r:kernel_t:unclassified
# ■
```



# Example – Policy Enforcement

The screenshot shows a terminal window titled "Terminal". The window contains the following command-line session:

```
fmac$ uname -a
SunOS fmac 5.11 fmac-proc i86pc i386 i86pc
fmac$ getenforce
permissive
fmac$ loadpolicy /etc/security/fmac/ss_policy
fmac$ setenforce enforcing
fmac$ loadpolicy /etc/security/fmac/ss_policy
loadpolicy: policy load of /etc/security/fmac/ss_policy failed: Permission denied
fmac$ setenforce permissive
setenforce: setting status to permissive failed: Permission denied
fmac$ █
```



# Example – AVC Messages

The screenshot shows a terminal window titled "Terminal". The window contains the following command and its output:

```
fmac$ dmesg | grep avc
Jul 15 07:06:24 fmac genunix: [ID 702911 kern.notice] avc: denied { load_policy } for scontext=system_u:system_r:kernel_t:unclassified tcontext=system_u:object_r:security_t:unclassified tclass=security
Jul 15 07:06:33 fmac genunix: [ID 702911 kern.notice] avc: denied { setenforce } for scontext=system_u:system_r:kernel_t:unclassified tcontext=system_u:object_r:security_t:unclassified tclass=security
Jul 15 07:06:46 fmac genunix: [ID 702911 kern.notice] avc: denied { load_policy } for scontext=system_u:system_r:kernel_t:unclassified tcontext=system_u:object_r:security_t:unclassified tclass=security
Jul 15 07:08:55 fmac genunix: [ID 702911 kern.notice] avc: denied { setenforce } for scontext=system_u:system_r:kernel_t:unclassified tcontext=system_u:object_r:security_t:unclassified tclass=security
fmac$ █
```



# Example – Truss Output

```
Terminal
File Edit View Terminal Tabs Help
101547: getcon("system_u:system_r:kernel_t:unclassified") = 0
101547: getexeccon("") = 0
101547: setexeccon("root:user_r:user_t") = 0
101547: getexeccon("root:user_r:user_t:unclassified") = 0
101547: getpidcon(1, "system_u:system_r:kernel_t:unclassified") = 0
101547: getpidcon(0, "system_u:system_r:kernel_t:unclassified") = 0
101547: is_fmac_enabled() = 1
101547: security_getenforce() = 0
101547: security_setenforce(0) = 0
101547: security_check_context("root:user_r:user_t") = 0
101547: security_compute_av("root:user_r:user_t", "root:user_r:user_t", 2, 32, 0
x0804725C) = 0
~
```



# Example – dtrace

```
# ./fmacssyscall.d
dtrace: script './fmacssyscall.d' matched 25744 probes
CPU FUNCTION
0  -> fmacsy          101692
0    -> fmacsy_getcon  101692
0      -> prfind        101692
0      <- prfind        101692
0      -> crgetsecid    101692
0      <- crgetsecid    101692
0      -> crgetsecid    101692
0      <- crgetsecid    101692
0      -> avc_has_perm_audit 101692
0        -> avc_has_perm_ref_audit 101692
0          -> avc_lookup    101692
0          <- avc_lookup    101692
0          <- avc_has_perm_ref_audit 101692
0        <- avc_has_perm_audit 101692
0        -> security_sid_to_context 101692
0          -> sidtab_search   101692
0          <- sidtab_search   101692
0          -> context_struct_to_string 101692
0            -> mls_compute_context_len 101692
0              -> ebitmap_cmp   101692
0                <- ebitmap_cmp   101692
```



# Source Example – Ubiquity

```
Terminal
File Edit View Terminal Tabs Help
#include <fmac/fmac.h>
#include <sys/fmac/flask.h>
#include <sys/fmac/av_permissions.h>
#include <stdio.h>

int
main(int argc, char *argv[])
{
    struct av_decision      avd;

    if (security_compute_av("jdoe:user_r:user_t",
                            "system_u:object_r:security_t", SECCLASS_SECURITY,
                            SECURITY_COMPUTE_AV, &avd)) {
        perror("security_compute_av failed");
        return 1;
    }
    if (SECURITY_COMPUTE_AV & avd.allowed)
        printf("allowed\n");
    else
        printf("denied\n");

    return 0;
}
```



# Source Example – Run Under Truss

The screenshot shows a terminal window titled "Terminal". The window has a standard OS X-style title bar with "File Edit View Terminal Tabs Help" menu items and standard window controls. The terminal itself displays the following text:

```
fmac$ ./truss/truss.fmac t1_i386
101633: execve("t1_i386", 0x080472C8, 0x080472D0)  argc = 1
101633: argv: t1_i386
101633: security_compute_av("jdoe:user_r:user_t", "system_u:object_r:security_t"
, 1, 4, 0x0804727C) = 0
denied
fmac$ █
```

# Next Steps

- Rebase to ONNV 93+
- Add file context support
- Continue to expand library and utilities
- Hook AVC into Solaris audit system
- Continue with design discussions on list
  - Zones
  - Networking
  - Labeling
  - RBAC convergence
  - Improved policy and system usability



# Community

- FMAC is a community project
- We would like to align and share
- Contributors welcome in any and all areas
- Lots of interesting work to do
- Join us:
  - <http://opensolaris.org/os/project/fmac/>
  - [fmac-discuss@opensolaris.org](mailto:fmac-discuss@opensolaris.org) (list membership required)



USE



IMPROVE



EVANGELIZE

# Questions?

開放的  
열린  
مفتوح  
libre  
মুক্ত  
ମୁକ୍ତ  
livre  
libero  
ମୁକ୍ତ  
开放的  
acık  
open  
nyílt  
ଓପନ  
オーブン  
livre  
ανοικτό  
offen  
otevřený  
öppen  
открытый  
ବେଳିପ୍ଲଟେ

# open



USE



IMPROVE



EVANGELIZE

## Thank you!

John Weeks

Senior Information Assurance Architect

john.weeks@sun.com

<http://blogs.sun.com/johnw/>

“open” artwork and icons by chandan:  
<http://blogs.sun.com/chandan>