Security Module Stacks That Don’t Fall Over
Casey Schaufler

• Kernel developer from the 1970’s

• Supercomputers in the 1990’s

• Smack Linux Security Module

• Security module stacking

Photo Curtesy Ann Forrister
Linux Security Module

• Collection of security hooks

• Additions to traditional access controls

You security people are insane.
Security Module Stack

• A collection of security modules

• Called in order

• Bail on fail policy
Minor Security Module

- Checks based on available state
Major Security Module

• Checks on module managed state
• System managed security blobs
• Netlabel and/or secmarks
Stacking as of 4.18
Stumbling Block

Blob Pointers
Security Blobs
Stacking with infrastructure managed blobs

Capability → Yama → LoadPin → AppArmor → Success

Failure
Stumbling Block

secids
32 bits allows one module’s data

```c
rc = security_cred_getsecid(cred, &secid);

rc = security_secctx_to_secid(ctx, ctxlen, &secid);

rc = security_secid_to_secctx(secid, &data, &datalen);
```
Replace u32 with struct secids

With stacking ...

```c
struct secids {
    u32 selinux;
    u32 smack;
    u32 apparmor;
};
```

Without stacking ...

```c
struct secids {
    union {
        u32 selinux;
        u32 smack;
        u32 apparmor;
    };
};
```
Identify which to use

• Within a security module

    sec->sid = secid->selinux;
Identify which to use

• In netfilter

```c
    case SECMARK_MODE_SEL:
        info->secid = secid.selinux;
        break;
    case SECMARK_MODE_SMACK:
        info->secid = secid.smack;
        break;
```
Identify which to use

- Select by task attribute

\[
\text{prctl(PR\_SET\_DISPLAY\_LSM, \"selinux\", 7, 0, 0);}\]
Stacking with struct secids

Capability → Yama → LoadPin → APPArmor → Success

Failure
Stumbling Block

Mount Options
Unrecognized Option

- `mount -o seclabel,smackfsroot="*"`

- Stop failing on unknown options

- Multiple mount option structures
Stumbling Block

netlabel
Packet Labeling

- One CIPSO tag
- Security modules must agree
Pushed attributes

• Security modules push data to netlabel
  • Other sub-systems pull data

• Attributes stored in socket
  • In network format

• May not be used in the end
Netlabel Configuration

• Unlabeled networks
  • Default for SELinux

• Labeled networks
  • Default for Smack

• Address selectors
  • Defers labeling until delivery
Granularity

• Security modules won’t be synchronized

• May change after socket creation
Stacking with netlabel equality

Capability ➔ Yama ➔ LoadPin ➔ Success

Failure
What Can Still Cause Problems?
Redundant purpose

• Don’t use SELinux and Smack together

• Do use Smack and AppArmor together
Networking

- Don’t confuse IP
- Use one network enabled module
User Space

• May get confused

• /sys/kernel/security/lsm

• Updates needed for real support
  • systemd
  • id
  • ls
Advice For New Security Modules
Networking

• Make netlabel optional

• Read the netlabel code before trying to use it

• Define sane behavior on unlabeled networks
Process Attributes

• Create a subdir in /proc/.../attr

• Create user space wrappers for SO_PEERSEC
Think twice about using secids

• Do you need audit events?

• What about tmpfs?
Be careful with state

• Module hooks may not get called

• Avoid additional memory management
  • Let the infrastructure do it
Summary
• Stacks of dissimilar modules are good

• Stacks should avoid fighting over the network

• Modules should color within the lines
Thank You