Linux Security Modules: General Security Support for the Linux Kernel

米澤研究室 増山隆*

2002 年 11 月 25 日

1 動機

サイト特有の強化したセキュリティーポリシーを組み込む仕組みは今の Linux カーネルにはない。既存のセキュリティー強化のモデルはカーネルのソースを書き換えてパッチの形で提供されているが、パッチではメインの拡張としてとりいれてもらうことは難しい。そこで、 Linux のローダブルモジュールの機能を用いて、セキュリティの論理をモジュール化する方法を LSM が与える。LSM は以下のことを目的とする。

- kernel の変更を少なくする
- なるべく多くのセキュリティーモデルをサポートできるようにする
- オーバーヘッドを小さくする
- POSIX.1e ケイパビリティをカーネルから分離する

2 Opaque "security" Fields

カーネル内で使用されている以下の構造体に void* 型の "security" という名のフィールドを付け加える。このフィールドはモジュールが与えるフックによって管理される。たとえば SELinux では task_security_struct, inode_security_ops といった構造体を定義して "security" フィールドに結びつけている。LSM は "security" フィールド更新のためのロックを提供しないので、モジュールがロックを管理しなければならない。

- task_struct
  プロセス
- linux_binprm
  プログラム
- super_block
  ファイルシステム
- inode
  パイプ、ファイル、ソケット

* tak@yl.is.s.u-tokyo.ac.jp

- file
  開いているファイル
- sk_buff
  パケット
- net_device
  ネットワークデバイス
- kern_ipc_perm
  セマフォ、共有メモリ領域、メッセージキュー
- msg_msg
  個々のメッセージ

3 Hooks

大域変数 security_ops にフックを登録する表 (構造体) がある。以下のようなフィールドがある。

```
struct inode_security_ops      * inode_ops;
struct file_security_ops       * file_ops;
struct task_security_ops       * task_ops;
  /* etc .. */
```

それぞれのフィールドに、関連したシステムコールがまとめられている。たとえば inode_ops には inode 操作に関連するシステムコールに対応するフックが登録されている。mkdir システムコールについては、security_ops-¿inode_ops-¿mkdir 、security_ops-¿inode_ops-¿post_mkdir [*1] というプレフック、ポストフックがある。

プレフックは int 型の返り値をもつ。これは error 変数にセットされて実際に処理が行われる前に error の値がチェックされる。 0 以外の値であれば、処理を行わずにリターンする。主にアクセス権限のチェックを行うためにある。

ポストフックは返り値を持たない (void)。これは処理終了後に呼ばれ主に、カーネルオブジェクトに追加された "security" フィールドの管理を行うためにある。

実際のコードは以下のようになる。

```
int vfs_mkdir(struct inode *dir,
              struct dentry *dentry,
              int mode) {
  /* some checks */
  error =
    security_ops->inode_ops->mkdir(dir, dentry, mode);
  if(error)
   goto exit_lock;
```

[*1] 最新版では、それぞれ security_ops-¿inode_mkdir, security_ops-¿post_mkdir になっている。古いパッチには、ポストフックはなかった

```
  error = dir->i_op->mkdir(dir, dentry, mode);
  security_ops->inode_ops->post_mkdir(dir, entry, mode);

  exit_lock:
  /* ... */
}
```

LSM はこれらのフックの呼出をカーネルパッチとして提供する。このパッチはカーネルから cap_set_full, cap_clear などの capability 関連の関数呼出を取り除く役割も果たす。

```
struct inode_security_ops {
  int (* create)(struct inode *dir, struct dentry *dentry,
                 int mode);
  int (* link)(struct dentry *old_dentry, struct inode *dir,
               struct dentry *new_dentry);
  int (* unlink)(struct inode *dir, struct dentry
                 *dentry);
  /* etc ... */
};
```

フックの種類は以下のように分けられる。

- Task Hooks
  kill , nice などプロセスに関わる操作に対するフック。 task_struct の "security" フィールドを管理する
- Prgram Loading Hooks
  プログラムのロードに関わる。execve から呼ばれる。
- IPC Hooks
  メッセージ毎のセキュリティーを管理するフック、IPC オブジェクトを触るときにそのオブジェクトに対する権限をチェックするフックがある。
- Filesystem Hooks
  mkdir などの inode 操作を行うシステムコールから呼ばれるフック。
- Network Hooks
  ソケット関連のシステムコールから呼ばれるフック。ネットワークデバイスを管理するための"security" フィールド
- その他
  モジュールの生成、初期化、削除に関するフックや、ホスト名の変更、I/O ポートの使用に関するフックがある。[*2]

[*2] security_ops-¿sethostname など

4 System call

4.1 security system call

security(unsigned int id, unsigned int call, unsigned long *args)
sys_security hook を呼ぶ sys_security システムコールを LSM は提供している。

5 LSM を使っているセキュリティーフレームワーク

- SELinux
- DTE Linux (Domain and Type Enforcement)
  SELinux / DTE Linux はともにはじめパッチとして提供されていたが LSM によってモジュールとして提供されるようになった。
- Openwall
  バッファフロー、一時ファイルレースなどの攻撃からシステムを守る
- POSIX.1e capabilities
  Linux のカーネルから capability の論理を外してモジュール化している。

6 性能評価

6.1 LMBench

LMBench とはファイルのオープン、削除のようなシステムコールの性能をはかるための小さなベンチマークである。デフォルトの (POSIX.1e) capability モジュールを使用して性能実験したところ、オーバーヘッドは大半が 0.1 ~ 3 % ほどに収まっている。 select が最も顕著なオーバーヘッドを示し 5.1 % であった。

6.2 Kernel Compilation

(おそらく default の capability モジュールを使用して) Linux kernel をコンパイルする実験を行ったがオーバーヘッドは 0.3 % 以下であった。

6.3 Webstone

典型的な応用。Netfilter と capability を組み込んで実験した。普通の kernel と LSM を使用したものと、SELinux とで単位時間あたりの接続回数を比較した。まず普通の kernel と LSM を使用したものを比較したところ、4 ~ 7 % のオーバーヘッドがかかることがわかった。普通の kernel と SELinux を比較した場合 16 ~ 21 % のオーバーヘッドがかかった。特に SMP ではロックに関する問題でオーバーヘッドが大きくなる。

当然 LSM のオーバーヘッドはモジュールが実装するセキュリティーエンジンの性能に依存する。

7　結論

LSM はセキュリティーの論理をモジュール化し、セキュリティー拡張をモジュールとして容易に提供できるようにした。

参考文献

[1] Linux Security Modules : General Security Support for Linux Kernel