Landlock : cloisonnement programmable non privilégié

Mickaël Salaün

ANSSI

7 juin 2017

Exemple : lecteur multimédia

Constat

- parsing complexe de formats multimédia
- source de vulnérabilités (p. ex. Android Stagefright)

Exemple : lecteur multimédia

Constat

- parsing complexe de formats multimédia
- source de vulnérabilités (p. ex. Android Stagefright)

Solutions

- développement sécurisé
- isolation des composants exposés

Objectif : sandboxer des applications sous Linux

Développeur : durcissement d'application

- confinement d'une compromission
- restriction à un sous-ensemble des accès initiaux
- prise en compte de la logique de l'application
- disponible pour tous les utilisateurs

Objectif : sandboxer des applications sous Linux

Développeur : durcissement d'application

- confinement d'une compromission
- restriction à un sous-ensemble des accès initiaux
- prise en compte de la logique de l'application
- disponible pour tous les utilisateurs

Utilisateur : principe du moindre privilège

- ne pas donner plus d'accès que ceux requis pour l'utilisation
- ne pas nécessiter de privilèges d'administration

Solutions d'isolation existantes sous Linux

	contrôle fin	non privilégié	intégration applicative autonome
SELinux. . .	✓		

Solutions d'isolation existantes sous Linux

	contrôle fin	non privilégié	intégration applicative autonome
SELinux...	✓		
seccomp-bpf		✓	✓
namespaces		∼	✓

Solutions d'isolation existantes sous Linux

	contrôle fin	non privilégié	intégration applicative autonome
SELinux...	✓		
seccomp-bpf		✓	✓
namespaces		∼	✓

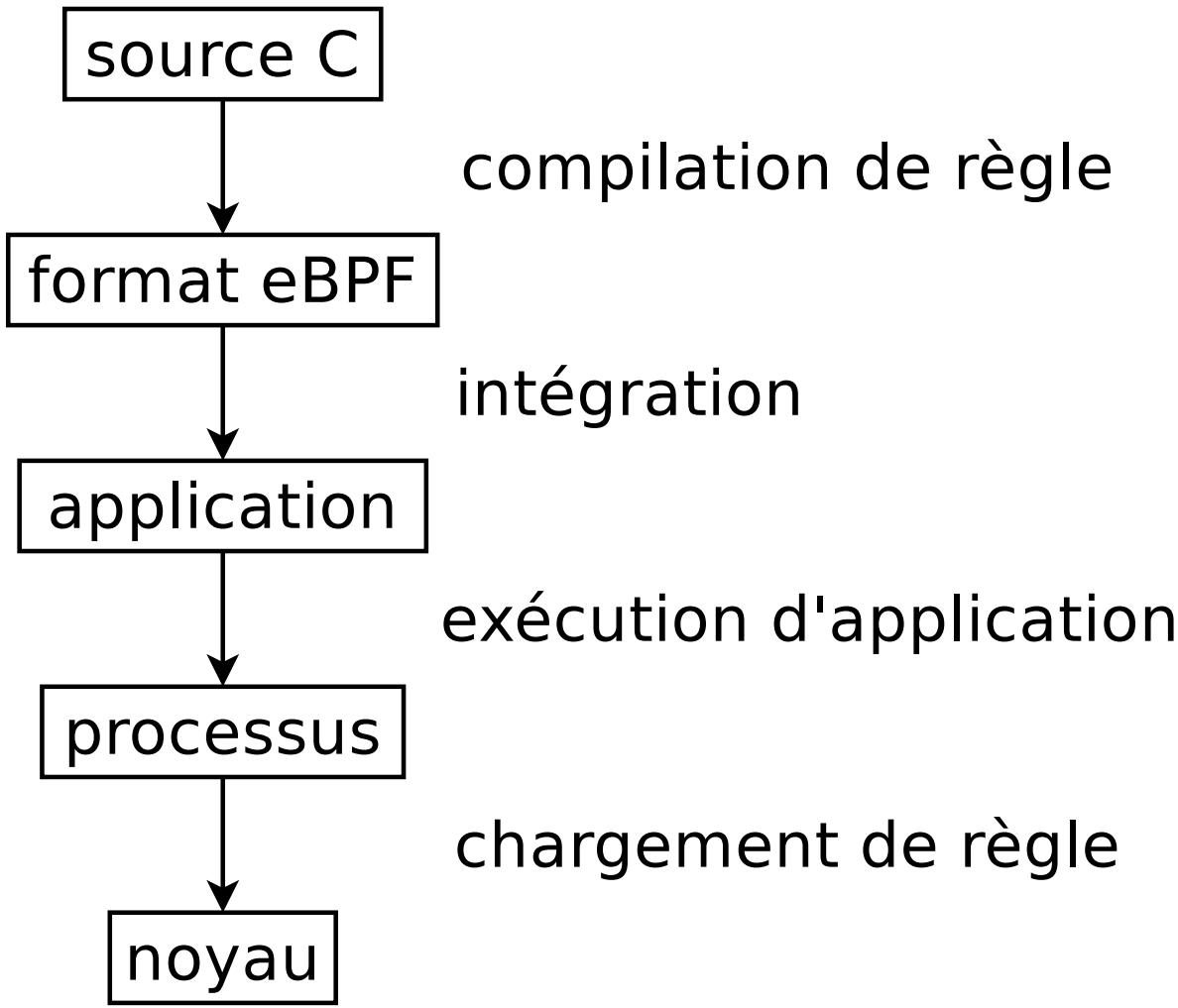
Exemples d'applications utilisatrices

- service : OpenSSH, systemd...
- navigateur : Chromium
- gestionnaire de sandbox : Firejail, Subgraph/Oz, Minijail, StemJail...
- gestionnaire de conteneur : Docker, LXC...

Propriétés du LSM Landlock

- contrôle fin, non privilégié et intégrable dans les applications
- complémentaire aux autres mécanismes de sécurité
- choix du modèle de contrôle d'accès

Cycle de vie d'une règle Landlock



Exemple de règle Landlock

- accès en lecture-seule au système de fichier...
- ...mais écriture autorisée sur les pipes
- règle appliquée sur chaque demande d'accès

Exemple de règle Landlock

```
 1 SEC("landlock1")
 2 int landlock_fs_rule1(struct landlock_context *ctx)
 3 {
 4     int mode;
 5
 6     /* allow non-write actions */
 7     if (!(ctx->arg2 & LANDLOCK_ACTION_FS_WRITE))
 8         return 0;
 9     /* get the file mode */
10     mode = bpf_handle_fs_get_mode(ctx->arg1);
11     /* allow write on pipes */
12     if (S_ISFIFO(mode))
13         return 0;
14     return 1;
15 }
```

Exemple de règle Landlock

```
1  SEC("landlock1")
2  int landlock_fs_rule1(struct landlock_context *ctx)
3  {
4      int mode;
5
6      /* allow non-write actions */
7      if (!(ctx->arg2 & LANDLOCK_ACTION_FS_WRITE))
8          return 0;
9      /* get the file mode */
10     mode = bpf_handle_fs_get_mode(ctx->arg1);
11     /* allow write on pipes */
12     if (S_ISFIFO(mode))
13         return 0;
14     return 1;
15 }
```

Exemple de règle Landlock

```
 1 SEC("landlock1")
 2 int landlock_fs_rule1(struct landlock_context *ctx)
 3 {
 4     int mode;
 5
 6     /* allow non-write actions */
 7     if (!(ctx->arg2 & LANDLOCK_ACTION_FS_WRITE))
 8         return 0;
 9     /* get the file mode */
10     mode = bpf_handle_fs_get_mode(ctx->arg1);
11     /* allow write on pipes */
12     if (S_ISFIFO(mode))
13         return 0;
14     return 1;
15 }
```

Exemple de règle Landlock

```
1  SEC("landlock1")
2  int landlock_fs_rule1(struct landlock_context *ctx)
3  {
4      int mode;
5
6      /* allow non-write actions */
7      if (!(ctx->arg2 & LANDLOCK_ACTION_FS_WRITE))
8          return 0;
9      /* get the file mode */
10     mode = bpf_handle_fs_get_mode(ctx->arg1);
11     /* allow write on pipes */
12     if (S_ISFIFO(mode))
13         return 0;
14     return 1;
15 }
```

Exemple de règle Landlock

```
1  SEC("landlock1")
2  int landlock_fs_rule1(struct landlock_context *ctx)
3  {
4      int mode;
5
6      /* allow non-write actions */
7      if (!(ctx->arg2 & LANDLOCK_ACTION_FS_WRITE))
8          return 0;
9      /* get the file mode */
10     mode = bpf_handle_fs_get_mode(ctx->arg1);
11     /* allow write on pipes */
12     if (S_ISFIFO(mode))
13         return 0;
14     return 1;
15 }
```

Exemple de règle Landlock

```
 1 SEC("landlock1")
 2 int landlock_fs_rule1(struct landlock_context *ctx)
 3 {
 4     int mode;
 5
 6     /* allow non-write actions */
 7     if (!(ctx->arg2 & LANDLOCK_ACTION_FS_WRITE))
 8         return 0;
 9     /* get the file mode */
10     mode = bpf_handle_fs_get_mode(ctx->arg1);
11     /* allow write on pipes */
12     if (S_ISFIFO(mode))
13         return 0;
14     return 1;
15 }
```

Exemple de règle Landlock

```
 1 SEC("landlock1")
 2 int landlock_fs_rule1(struct landlock_context *ctx)
 3 {
 4     int mode;
 5
 6     /* allow non-write actions */
 7     if (!(ctx->arg2 & LANDLOCK_ACTION_FS_WRITE))
 8         return 0;
 9     /* get the file mode */
10     mode = bpf_handle_fs_get_mode(ctx->arg1);
11     /* allow write on pipes */
12     if (S_ISFIFO(mode))
13         return 0;
14     return 1;
15 }
```

Landlock, un LSM pour l'utilisateur non privilégié

Application de contrôle d'accès

- besoin de privilèges
- approche alternative de l'interface traditionnelle (SUID) par une nouvelle interface dédiée à la restriction d'accès

Landlock, un LSM pour l'utilisateur non privilégié

Application de contrôle d'accès

- besoin de privilèges
- approche alternative de l'interface traditionnelle (SUID) par une nouvelle interface dédiée à la restriction d'accès

Protection du noyau et de ses ressources

- surface d'attaque limitée :
  - interprétation eBPF : analyse statique
  - code LSM : exécution après les autres contrôles
- protection contre le déni de service
- protection contre les canaux auxiliaires

Landlock

Démo

Récapitulatif de Landlock

Durcissement d'application

- non privilégié
- contrôle d'accès sur objets noyau
- politique de sécurité évolutive

Intégration upstream incrémentale

- envoi de plusieurs séries indépendantes de patchs
  - LKML
  - github.com/landlock-lsm
- incubation dans LinuxKit (Docker)