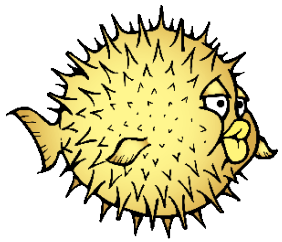
Mesures de protection dans OpenBSD

Matthieu Herrb



LAAS-CNRS

Université de Toulouse

Resist, 23 juin 2009

Agenda

Agenda

OpenBSD...

- Système d'exploitation Unix-like multi-plateformes
- Dérivé de BSD 4.4
- Noyau + userland + doc maintenus ensemble
- Applications tierces disponibles via ports
- Une release tous les 6 mois
- Architectures matérielles : i386, amd64, alpha, sparc, sparc64, macppc, arm,...

Objectifs

- Fournir du code libre (licence BSD...)
- de qualité
- correct
- conforme aux standards (POSIX, ANSI)
- avec outils crypto (SSH, IPSEC)

$\longrightarrow$ meilleure sécurité

Version courante

OpenBSD 4.5 sortie le 1^{er} mai 2009.

Nouveautés :

- plus de support pour machines sparc64 (UltraSparc T2+, ...)
- nouveaux drivers
- ypldap
- nouveau sous-système audio
- support C99 dans libm (complex)
- ...

Agenda

« Sûr par défaut »

- Leitmotiv depuis 1996
- Adopté depuis par tous les OS
- Services non indispensables désactivés à la fin d'une installation standard
- Configuration par défaut des services pour la sécurité.
- Nécessite une intervention explicite de l'utilisateur
- Conserver un système utilisable

$\longrightarrow$ seulement 2 failles exploitables à distance en plus de 10 ans.

Règles de codage

- Correction du code d'abord $\longrightarrow$ meilleure fiabilité, meilleure sécurité.
- Conception recherchant la simplicité.
- Principe de revue par les pairs à tous les niveaux
- Recherche systématique des erreurs
- Nouvelles fonctionnalités de gcc :
  - option `-Wbounded`,
  - attribut `__sentinel__`
- Outils : llvm/clang, Parfait, etc.

Technologies pour la sécurité

- `strlcpy/strlcat`
- protection de la mémoire
- révocation des privilèges (ex. `ping`)
- séparation des privilèges (ex. OpenSSH)
- mise en cage (`chroot`)
- uids distincts par service
- protection de la pile (SSP) & Stackgap
- introduction d'aléas (ld.so, malloc, mmap)

Protection de la pile et de la mémoire

Débordements de pile : faille la plus facile à exploiter...

- Stackgap
- GCC + Propolice actif sur l'ensemble du système et des bibliothèques.
- Généralisation de la protection automatique contre les erreurs dans les formats de printf() & Co.

Propolice

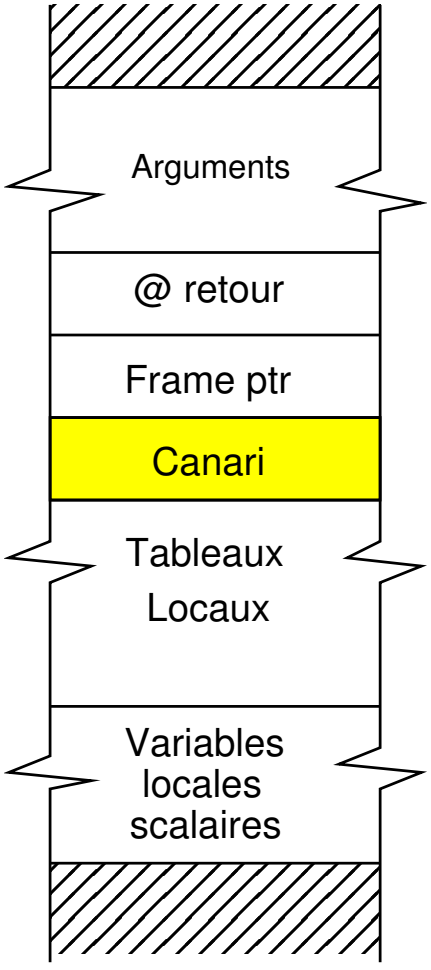
`http://www.trl.ibm.com/projects/security/ssp/`

Principe placer un « canari » sur la pile avant les variables locales

- vérification lors du retour.
- si toujours vivant : pas de débordement
- si mort (écrasé) $\longrightarrow$ débordement détecté -> abort

Seulement si tableaux présents dans les variables locales

Adopté dans gcc 4.1.



W^X

(W $\oplus$ X pour les puristes)

Écriture **ou exclusif** exécution autorisée sur une page..
(PAX sur Linux...)

- facile sur certaines architectures (x86_64, sparc, alpha) :
  bit ’X’ par page
- plus difficiles sur d’autres (x86, powerpc) :
  bit ’X’ par segment
- impossible dans certains cas (vax, m68k, mips)

Nombres aléatoires dans OpenBSD

Source de nombres aléatoires importante pour la sécurité.

Collecte d'entropie :

- dans les E/S : clavier, souris, cartes réseau, audio, etc.
- sources matérielles (CPUs VIA, cartes accélération crypto)

Utilisation :

- Nombres pseudo-aléatoires utilisant `arc4random()` pour ne pas épuiser l'entropie trop vite.
- Système centralisé $\rightarrow$ augmente la sécurité. (Difficilement observable ou répétable).

Aléa dans `ld.so`

Chargement des bibliothèques dynamiques en ordre aléatoire
+ offset aléatoire pour chacune.

Conséquences :
Adresse de chargement aléatoire d'un système à l'autre pour tous les objets partagés
$\longrightarrow$ retour vers libc beaucoup plus difficile

Aléa dans mmap()

Adresse retournée par `mmap()` :

Si `MAP_FIXED` n'est pas spécifié : retourne une adresse aléatoire.

(Comportement traditionnel : première page libre après une adresse de base)

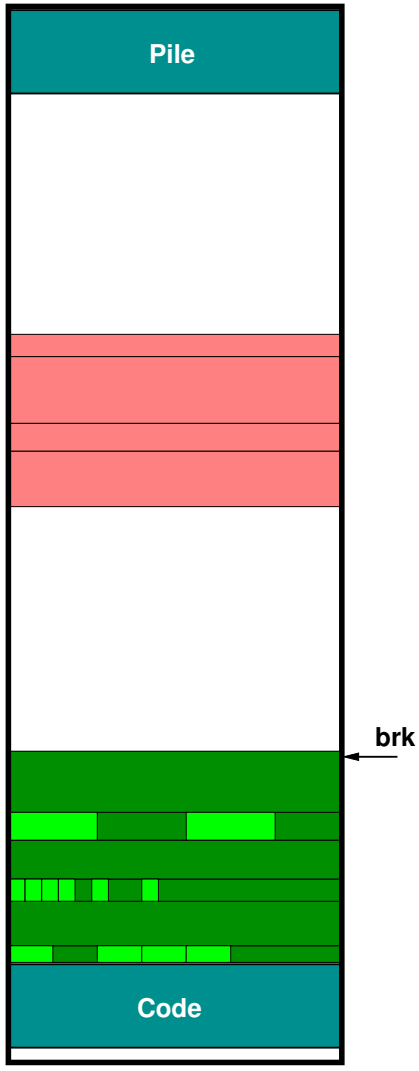
Aléa dans malloc()

- Allocations $\geqslant$ 1 page : mmap() $\rightarrow$ adresses aléatoires.
- Allocations $<$ 1 page : allocateur classique par blocs de taille fixe, mais sélection aléatoire du bloc dans la liste libre.

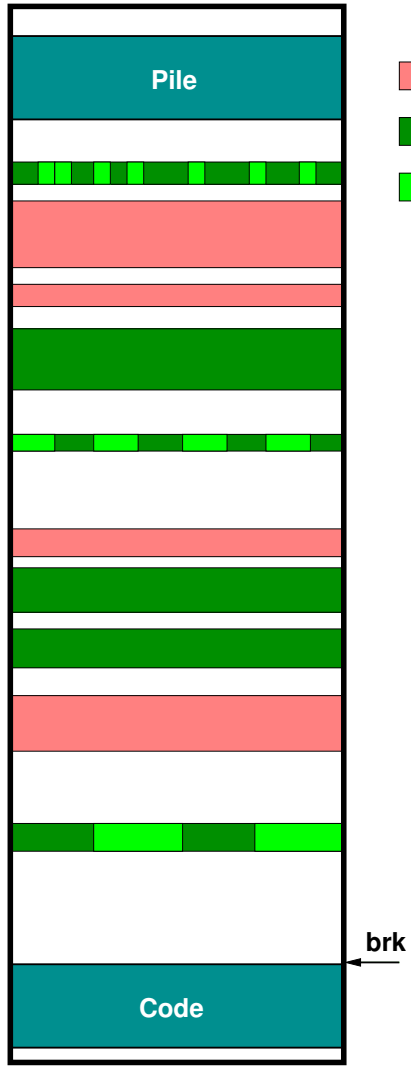
$\Rightarrow$ attaques sur le tas plus difficiles.

Protection de la mémoire dynamique

- Activée avec `/etc/malloc.conf` $\longrightarrow$ `G`
- Chaque allocation plus grande qu'une page est suivie d'une page de garde $\Rightarrow$ erreur de segmentation si débordement.
- Les allocations plus petites sont ordonnées aléatoirement dans la page.



Modèle traditionnel



Modèle OpenBSD

Réduction des privilèges

Révoquer définitivement les privilèges des commandes privilégiées (setuid) ou démons lancés avec privilège (named) une fois que toutes les opérations nécessitant un privilège sont effectuées.
Grouper ces opérations le plus tôt possible.
Exemples :

- ping
- named

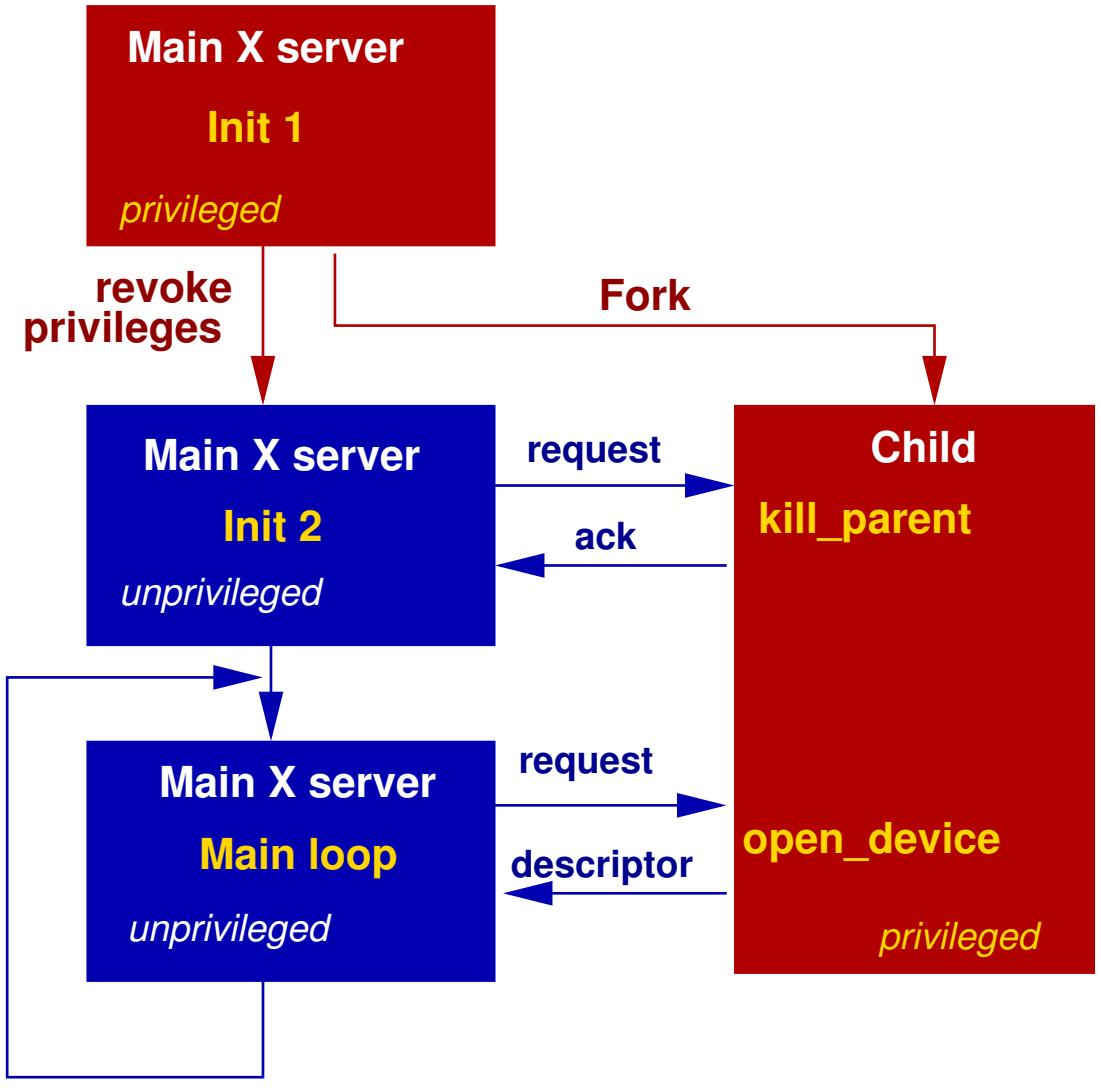
Séparation des privilèges

[Provos 2003]

- exécuter les démons système
  - avec un uid $\neq 0$
  - dans une cage chroot(2)
- processus additionnel d'aide qui reste privilégié mais vérifie de manière paranoïaque toutes ses actions.

Une douzaine de démons ainsi protégés

Exemple : serveur X



Niveaux de sécurité

Parti-pris : pas de politique à grain fin :
trop complexe donc potentiellement dangereux.

Par défaut 3 niveaux de privilège :

- noyau
- root
- utilisateur

Flags du système de fichiers (immutable, append-only) pour limiter les accès de root.
Interdiction de modification de certains réglages de sécurité.
Limite les accès à `/dev/mem`.
Exception : X...

Agenda

Menaces sur les protocoles

Internet : favorise les choses qui marchent au détriment de la sécurité.

- valeurs faciles à deviner
- permettent de forger des paquets acceptés comme valides
- fuites d'information
- utilisation de l'heure comme identificateur secret ? !

Principe de protection

Utiliser des données impossibles (difficiles) à trouver partout où des données arbitraires sont autorisées, même s'il n'existe pas (encore) d'attaque connue.

- compteurs
- horodatage
- identificateurs de paquet, de session, d'hôte, ...

Respecter les contraintes pour ne rien casser :

- non répétition
- intervalle minimal entre 2 valeurs
- éviter les valeurs magiques

Exemple : TCP Reset

« Slipping in the window », Paul T. Watson (2004)

Le redimensionnement des fenêtres TCP facilite la recherche d'un numéro de séquence TCP valide.

Cible : connexions de longue durée, entre machines qui font peu de connexions différentes (sessions BGP par ex).

Solution OpenBSD :

- ports source vraiment aléatoires
- exige que les paquets RST soient à l'extrême droite de la fenêtre
- et bien sûr aussi : authentification TCP MD5 et/ou IPSec (OpenBGPD n'accepte pas la négociation de fenêtre TCP sans l'un ou l'autre).

DNS

Sécurité basée sur

- (ip source, port source, ip dest, port dest)
- identificateur 16 bits

OpenBSD :

- identificateurs pseudo-random depuis 1997
- port source aléatoire

Aléas dans la pile réseau

Utilisation :

- IPID (16 bits, non répétition)
- DNS Queries (16 bits, non répétition)
- TCP ISN (32 bits, non répétition, écart de 2^{15} entre 2 valeurs)
- Ports source (ne pas réutiliser un port encore actif)
- Horodatage TCP (valeur initiale aléatoire, puis croissant à taux constant)
- Id NTPd (64 bits, aléatoire) à la place de l'heure courante
- RIPdMD5 auth...x

PF : plus d’un tour dans son sac

Packet Filter

- Filtrage de base à état + possibilités de réécriture (NAT) :
- **Scrub** pour ajouter de l’aléa sur les paquets :
  - TCP ISN
  - IP ID
  - TCP timestamp
  - NAT : Réécriture des ports sources (voire des adresses)

Protège aussi les machines non-OpenBSD.

Agenda

Protection des outils utilisateur

- Navigateurs web,
- outils multimédia,
- mail...

Pas encore assez protégés. (piste : Chrome ?)

Problème du virus « corse »

Les problèmes de X

Code privilégié niveau noyau dans l'espace utilisateur
failles dans X particulièrement dangereuses (cf. Loïc Duflot Cansecwest)

- Séparation des privilèges (pas suffisant malheureusement)
- Supprimer accès direct au matériel : vesafb
- Futur : KMS + DRI : programmation de la carte dans le noyau, accès filtrés par DRI.

Agenda

Conclusion

- Nombreux progrès depuis le début
- Contribution à la correction de bugs dans de nombreuses applications tierces
- Souvent copié (bien)
- Restent de nombreuses choses à faire
- Syndrome du mythe Sisyphe ?

Bibliographie

`http://www.openbsd.org/papers/index.html`

- *Using OpenBSD Security Features to Find Software Bugs*, Peter Valchev, Reflections/Projections, Champaign-Urbana, 2007
- *Time is not a secret : Network Randomness in OpenBSD*, Ryan McBride Asia BSD Conference 2007
- *Security Measures in OpenSSH*, Damien Miller Asia BSD Conference 2007
- *The OpenBSD Culture*, David Gwyne : OpenCON 2006
- *Security issues related to Pentium System Management Mode*, Loïc Duflot, CansecWest 2006.
- *Exploit Mitigation Techniques*, Theo de Raadt OpenCON 2005, Venice, Italy
- *A Secure BGP Implementation*, Henning Brauer SUCON 04
- *Preventing Privilege Escalation*, Niels Provos, Markus Friedl and Peter Honeyman, 12th USENIX Security Symposium, Washington, DC, August 2003.
- *Enhancing XFree86 security*, Matthieu Herrb LSM, Metz 2003.

Questions ?