Aide-mémoire sur les failles en C

Victor Stinner

5 octobre 2005

1 Failles en C

1.1 Débordement (buffer overflow et integer overflow)

Les erreurs de programmation les plus courantes sont les erreurs de débordement. Les cas les plus connus sont :

- buffer overflow : l'entrée clavier est plus longue que prévue
- integer overflow : dépassement de capacité d'un nombre entier qui le fait passer dans les nombres négatifs

Ces failles sont très courantes, allez faire un tour sur le site frsirt(.com), pour vous en rendre compte. Elles permettent d'injecter du code arbitraire.

1.2 Fonctions (de la libc) connues pour être faillibles

De nombreuses fonctions standards contiennent des failles de sécurité (si elles sont mal utilisées). Liste volontairement incomplète : strcpy(), strcat(), sprintf(), vsprintf(), getwd(), gets(), realpath(), ...

1.3 Autres types de faille

Heap overflow, format string, déni de service, insecure temporary file creation, race condition, problème de droit d'accès aux fichiers, problème de stockage des mots de passe, missing input sanitising, ...

2 Trouver les failles

2.1 Sources d'information

- Événements clavier et souris
- Pipe Unix
- Fichiers
- Donnée en provenance d'un réseau (socket)
- Variable d'environnement (on n'y pense pas souvent)
- Signaux : ce n'est pas à proprement parler une source de données, mais on peut agir sur un programme
- etc.

2.2 Stresser un programme

Pour stresser un programme, il faut lui envoyer des données auquelles il ne s'attend pas. Pour une chaîne de caractères, cela peut être par exemple :

- Chaîne vide
- Chaîne contenant des caractères nuls
- Chaîne trop courte / trop longue
- etc.

3 Solutions

3.1 Protéger la pile

Ajout d'un canari sur le pile pour vérifier les débordements. Voir StackField et l'option de compilation GS dans Visual Studio .NET. OpenBSD et Linux (patch PaX) interdisent l'exécution de code sur le pile. Les processeurs AMD (flag NX) et Intel (flag XD) apportent une solution matérielle à ce problème.

3.2 Protéger les fonctions sensibles

La bibliothèque libsafe remplace les fonctions sensibles de la libc par une version plus sécurisée. http ://www.research.avayalabs.com/project/libsafe/

3.3 Utiliser un langage de haut-niveau ?

Les langages dits de "haut niveau" évitent les failles les plus communes telle que le buffer overflow. Quelques langages dits de "haut-niveau" : PHP, Python, Perl, Java, Haskell, etc.

4 Liens sur le web

- Splint : outil d'analyse statique de code C.
  http ://www.splint.org/
- Protections contre l'exploitation des débordements de buffer, article paru dans magazine MISC
  http ://www.miscmag.com/articles/index.php3 ?page=415
- FrSIRT 24h/24 7j/7, actualité des failles de sécurité
  http ://www.frsirt.com/
- Phrack, eZine sur la sécurité informatique
  http ://www.phrack.org/
- Packet storm security : annuaire d'outils, exploits, articles, etc.
  http ://packetstormsecurity.org/
- Insecure (auteur du logiciel nmap) : liste d'outils, liste de diffusion, exploits, etc.
  http ://www.insecure.org/
- Articles de sécurité sur mon site perso
  http ://www.haypocalc.com/wiki/Sécurité