FORMAT STRING BUGS

Yannick Harson

4 décembre 2007

Table des matières

Première partie

Introduction

Découverts en 1999, les **Format String Bugs** (pouvant etre traduit par "bug de format de chaine") sont une classe de vulnerabilités au départ sous-estimée, jugée plutôt inoffensive par la communauté des programmeurs et donc quelque peu négligée en comparaison par exemple avec les buffer-overflows, eux reconnus dangereux depuis le début des années 90.

C'est en juin 2000 que le danger représenté par ces bugs fût justement évalué, mais le fait d'avoir négligé ces failles amène aujourd'hui à constater qu'elles comptent parmi les plus répandues : le projet CVE (Common Vulnerabilities and Exposures) enumère en effet 400 programmes vulnérables et classe les Format String Bugs à la neuvième place des vulnérabilités les plus signalées entre 2001 et 2006.

Deuxième partie

Fonctions de format

Les **fonctions de format** en langage C sont un genre de fonction de la norme ANSI qui prennent un nombre variable d'arguments, dont la chaine de format en question. Ce sont des fonctions de conversion, utilisées pour rendre lisible par l'utilisateur les types de base du langage C. Elles sont utilisées dans la quasi-totalité des programmes C, pour extraire des informations, pour renvoyer des messages d'erreur ou pour traiter les chaines de caractères.

La norme ANSI définit un certain nombre de fonction de format du langage C.

0.1 Quelques représentants des fonctions de format

fprintf écrit dans un fichier

printf écrit en sortie

sprintf écrit a l'intérieur d'une chaine de caractères

snprintf écrit a l'intérieur d'une chaine de caractères en contrôlant la taille

vprintf idem que printf en passant en argument une va_list qui pointe vers les arguments

vfprintf idem que fprintf en passant en argument une va_list qui pointe vers les arguments

vsprintf idem que sprintf en passant en argument une va_list qui pointe vers les arguments

vsnprintf idem que snprintf en passant en argument une va_list qui pointe vers les arguments

Ces fonctions de format sont utilisées pour convertir les types de données simples du langage C en chaines de caractères. Elles permettent de spécifier le type des données à représenter et de traiter la chaine produite.

Le fonctionnement est le suivant : la chaine de format contrôle le comportement de la fonction, elle détermine le type de paramètres à écrire, les paramètres sont sauvegardés sur la pile, par valeur ou par adresse (ou référence).

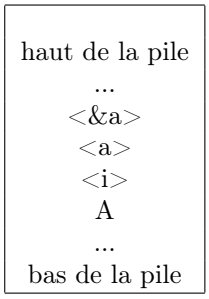
Un mot sur les chaines de format

Prenons pour exemple une fonction printf :
printf("j'ai %d frères",2)
qui doit afficher "j'ai 2 frères". Le %d définit le type (ici entier) du paramètre à afficher (ici 2) à l'intérieur du texte entre guillemets qui lui sera affiché tel quel.

On peut définir plusieurs types de paramètres, qui ont chacun leur type de passage (ex : %u pour les unsigned int, passés par valeur - %s pour les string, passés par référence).

0.2 Rôle de la pile pour les fonctions de format

La fonction de format récupère dans la **pile** la chaine de format ainsi que ses paramètres :



A étant l'adresse de la chaine de format, i et a les valeurs des variables i et a et &a l'adresse de la variable i.

La fonction de format analyse la chaine caractère par caractère : si le caractère lu n'est pas %, il est copié tel quel. Si la fonction lit le caractère %, le caractère derrière définit le type de paramètre a évalué et a retranscrire.

Troisième partie

Format String Bugs et exploitations possibles

0.3 Principe général

Les **Format String Bugs** exploitent donc les faiblesses des fonctions de format telles que printf, fprintf... Ils peuvent être utilisés pour planter un programme ou exécuter un code malveillant.

La faiblesse des fonctions de format survient lorsque la chaine à passer en argument à la fonction doit être partiellement ou totalement fournie par l'utilisateur.

C'est en fournissant la chaine de format que l'on peut prendre la contrôle du comportement de la chaine de format.

0.4 Exploitations

Une exploitation simple est de faire **planter un processus**. Cela peut être utile dans certains cas, par exemple dans certaines attaques réseau où il est utile d'avoir un service qui ne repond pas.

On peut également **regarder certains endroits de la mémoire** : vu que l'on peut voir la chaine sortie par la fonction de format, nous pouvons recueillir des renseignements utiles. Comme nous contrôlons le resultat de la fonction de format, nous pouvons l'utiliser pour savoir ce que fait notre chaine de format et savoir comment le processus est disposé.

On peut ainsi **reconstruire une partie plus ou moins importante de la pile** et ainsi savoir comment elle est disposée. Il est possible dans certains cas de reconstituer la totalité de la pile.

En affichant le contenu d'une adresse mémoire que nous avons préalablement fournie, nous pouvons également aller voir ce qu'il y a en mémoire a des endroits quelconques.

Mais ce qui se fait de mieux en matière d'exploitation demeure la **prise de contrôle du pointeur d'instruction du processus**. Pour réaliser cet exploit, il faut trouver des instructions qui modifient le pointeur d'instruction et savoir comment ces instructions le modifient. Ceci est réalisable de deux manières : la première s'apparente à l'exploitation d'un buffer-overflow, les faiblesses des chaines de format étant parfois des restrictions de longueur tampon permettant ce type d'exploitation. La seconde se fait en fournissant une vraie chaine de format.

Ces deux exploitations permettent donc de prendre le contrôle du pointeur d'instruction et ainsi d'écrire librement sur la mémoire à des endroits quelconques.

Quatrième partie

Conclusion

Comme il a été dit en introduction, les faiblesses liées de format de chaines sont très répandus. Le langage C semble être le plus touché par ces failles : effectivement, si on peut observer ce type de failles dans d'autres langages, elles apparaissent tout de même moins fréquemment et ne peuvent normalement pas être utilisées pour exécuter le code d'un attaquant.

Les bugs de format ont été observés pour la première fois en 1990 lors du fuzz testing effectué à l'université du Wisconsin. Ils furent alors appelés "interaction effect"(effets d'interaction) et repérés lors des tests du C shell. L'utilisation des bugs de format de chaine comme vecteur d'attaque a été découvert par T.Twillman pendant une conférence sur la sécurité du demon ProFTPD. Ceci a abouti au premier rapport en septembre 1999 considérant cette catégorie de faiblesses, incluant un exploit basique. Cependant, c'était plusieurs mois avant que la communauté de sécurité ne se mette au courant de tous les dangers des failles des chaines de format en tant qu'exploits pour d'autres software utilisant cette méthode. Le rapport "Format String Attacks" par T.Newsham fut publié en septembre 2000.

SOURCES :

http ://en.wikipedia.org/wiki/Format_string_attack
Exploiting Format String Vulnerabilities par Scut and Team Teso.