gitlab.enssat.fr

Alexandre DOYEN / CTF Logiciels sécurisés · GitLab

3-4 minutes

Alpha Management Server

1) Contexte

Alpha Management Server est un logiciel de gestion de classes. Le but est d'entrer les différentes informations relatives aux étudiants, afin de pouvoir les consulter.

Dans le cadre d'une attaque, il serait idéal de pouvoir utiliser ce logiciel afin d'ouvrir un shell sur le serveur sur lequel il est exécuté.

2) Méthodologie d'analyse de l'application

Pour analyser le binaire de l'application, j'ai utilisé le logiciel Ghidra afin d'avoir une vue d'ensemble du logiciel désassemblé et décompilé dans le cadre d'une analyse statique. Ensuite, à l'aide de GDB j'ai pu effectuer une analyse dynamique.

En analysant ce binaire, j'ai découvert deux éléments intéressants dans le cadre d'une attaque de type ROP (Return Oriented Programming) :

- La présence d'une fonction "hide" qui fait un appel `system("/bin/sh");`
- La présence d'un stack buffer overflow dans la fonction `create()`, qui demande à l'utilisateur de rentrer les informations de l'étudiant. En effet, on peut remarquer que le développeur a commis une erreur de programmation.

3) Mise en place de l'attaque

Ainsi, l'erreur commise est la suivante :

```
printf("Nom : ");
fgets(stdinBuffer,50,stdin);
tailleDeLEntreeUtilisateur = strcspn(stdinBuffer,"\r\n");
stdinBuffer[tailleDeLEntreeUtilisateur] = '\0';
strncpy(class + current_max_index * 0x71 +
0x1f,stdinBuffer,30);
printf("City : ");
gets(stdinBuffer);
tailleDeLEntreeUtilisateur = strcspn(stdinBuffer,"\r\n");
stdinBuffer[tailleDeLEntreeUtilisateur] = '\0';
strncpy(class + current_max_index * 0x71 +
0x1f,stdinBuffer,50);
```

On peut remarquer que le remplissage du champ "City" comporte deux vulnérabilités : La présence, dans un premier temps, est l'appel à la fonction

`gets()` au lieu de la variante sécurisée `fgets()` ; et ensuite, la taille du champs à remplir qui n'est pas bon dans l'appel à `strncpy()` suivant (50 au lieu de 30). De plus, le remplissage du champ "City" écrase le champ "Nom", vu que le pointeur passé en paramètre à `strncpy()` est le même (`class + current_max_index * 0x71 + 0x1f`).

Donc, le champ "City" sera celui utilisé pour réaliser l'attaque.

4) Attaque

Après analyse avec GDB, il se trouve que la valeur du registre EIP (Compteur ordinal) sauvegardée sur la pile se situe à un décalage de 226 octets après le champ "Nom". Cette valeur a été trouvée en faisant un buffer overflow de 512 octets grâce à un patron généré par GEF (Extension de GDB permettant de faciliter l'analyse). Il est en effet possible d'utiliser le champ "Nom", car l'appel `fgets(ptr, 50, stdin)` ne va prendre que les 50 premiers octets passés sur l'entrée standard.

Ainsi, pour réaliser l'exploit, et appeler la fonction `hide()`, il faut mettre son adresse (`0x80494e6`) à 226 octets après le début du champ. Le script python associé permet de le faire avec l'aide de la bibiliothèque "Pwntool" ![1]

5) Notes

1. Le script est fourni en annexe de ce Writeup. ↩