TP 2 - Exploitation Vulnérabilités Logicielles – part 1

Florent Autréau

2016-2017

Pré-requis – Installation de la VM sur votre poste

Récupérer la vm disponible sur <...>

Le compte de la machine virtuelle est securimag / 123456

Compte-rendu

Le compte-rendu de ce TP devra être soumis au format pdf à florent.autreau@imag.fr avant le 25/12/16. Il comprendra les réponses aux questions ainsi que le code des programmes commentés si besoin.

Vulnérabilités logicielles :

Dans l'archive (cwe_samples.tgz), vous trouverez des programmes exemples comportant des vulnérabilités à exploiter. Cette première partie consiste à construire, éxécuter, analyser et corriger chacun de ses exemples.
Les vulnérabilités abordées sont catégorisées selon la dénomitation de http://cwe.mitre.org.
Soient :

- CWE-120 Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') - C
- CWE-131 Incorrect Calculation of Buffer Size - C
- CWE-134 Uncontrolled Format String -C
- CWE-190 Integer Overflow or Wraparound – C/Java
- *CWE-798 Reliance on Untrusted Inputs in a Security Decision – C/C++/Java*
- *CWE-807 Use of Hard-coded Credentials*

Pour chacune de ces catégories (donc 5 programmes) :

1) Consulter le site de référence et comprendre la vulnérabilité.
2) Compiler le programme (par ex : cd cwe_131/c ; make all ;)
3) Lancer le programme ainsi que l'exemple d'exploitation (make poc)
4) Analyser le défaut et proposer une correction
5) Implémenter et tester la correction

Note : on pourra utiliser gdb (voir partie suivante) pour éxécuter le programme pas a pas si besoin.

Annexe :

IDA :

Pour lancer IDA, dans un terminal « ida », ou lancer le script ida.sh se trouvant sur le bureau.

Utilisation de gdb :
x/x ADDR : affiche la valeur se trouvant à l'adresse ADDR
x/4x ADDR : affiche 4 octets depuis l'adresse ADDR
x/i ADDR : affiche l'octet sous forme d'une instruction

breakpoint * ADDR : place un breakpoint sur ADDR
disas func : désassemble la fonction func

run : lance le programme
run ARG : lance le programme avec l'argument
nexti : exécute la prochaine instruction
continue : continue l'exécution (après un breakpoint)

Utilisation de python :
Vous pouvez vous servir de python pour générer des entrées à vos programmes.

Exemple d'utilisation :
./programme $(python -c ' print "A" ')
Lance le programme avec A pour argument

./programme $(python -c ' print "A"*10 ')
Lance le programme avec AAAAAAAAAA pour argument

./programme $(python -c ' print "\x41"*10 ')
Lance le programme avec AAAAAAAAAA pour argument (0x41 étant le code ASCII de A)

Vous pouvez aussi utiliser la même syntaxe pour lancer d'autres commandes, par exemple pour donner un fichier en entrée d'un programme :
./programme $(cat fichier.txt)

Vous pouvez utiliser python aussi dans gdb, exemple :
run $(python cat fichier.tx)
Vous permettra d'avoir facilement le fichier fichier.txt en entrée du programme sous gdb.