Esercizio Admin Panel

Allora, andando in http://localhost:8075/ si trova un link alla pagina di admin http://localhost:8075/admin/ ma c'è una scelta tra 2 file. Andando sul .bak posso vedere tutto il sorgente mentre in .php posso andare nella pagina eseguibile.

Dal sorgente vedo delle cose molto interessanti. Per essere autenticato vuole un cookie chiamato otadmin ma non basta. Vuole che otadmin contenga una cosa del tipo {"hash" : REG } dove REG è un qualsiasi cosa di lettere (maiuscole) e numeri, eventualmente tra ". Per esempio otadmin={"hash": "ciao"}

Se setto questo cookie la pagina mi ritorna una serie di 0 e 64. Dal sorgete vedo che questi 0 e 64 sono ottenuti tra una AND bit a bit tra l'hash della chiave (che devo trovare) e 0xC0. Tuttavia sappiamo che la AND è una operazione LOSSY quindi sto perdendo informazioni. Qui la soluzione VERA vuole l'analisi di questa serie di 0 e 64 ma ho usato un altro modo. Vedo il confronto $session_data['hash'] != strtoupper(MD5($cfg_pass)). È molto interessante perchè utilizza l'operatore !=

In php esistono 2 classi di operatori di confronto.

I type safe: === e !== i quali VOGLIO un confronto tra lo stesso tipo

Gli altri: == e != che permettono confronti tra variabili di tipi diversi.

Per esempio 4 == "4" è vero ma 4 === "4" è falso

Nella riga che ho citato prima, a sinistra ho il contenuto di session_data, che è quello nel cookie, e a destra l'hash del token. Con {"hash": "ciao"} ho un confronto tra stringa e stringa ma se mandassi {"hash": 4} allora ho un confronto tra int e string che È CONCESSO da !=

Quindi, come funziona != tra int e stringa? La stringa viene convertita in int troncando la parte int iniziale fino alla prima lettera, per esempio "12ABC" viene trasformata in 12 in int, "1ABC2" viene convertita in 1. Quindi, perchè non provare a mandare degli int a caso sperando che l'hash inizi con un numero piccolo? Facciamolo:

```
import requests

s = requests.Session()

for i in range(2**16):

    data = {'otadmin':'{"hash": ' + str(i) + '}'} #definition of the cookie

    url='http://127.0.0.1:8075/admin/login.php'

    r = s.get(url, cookies = data)

    if("0006464640640064000640006464000646400640000" in r.content):

        continue

    else:

        print("FOUND:")

        print (i)

        exit(0)
```

Lo script si interrompe a 389

quindi *otadmin={"hash": 389}* e aggiorno la pagina