SERVIZIO QUADRATICEQUATION CTF 2019

Team Jelly Hinge

Introduzione

Quadraticequation era uno dei servizi web da exploitare durante la UniCTF 2019, avente una vulnerabilità di tipo SQL Injection di secondo ordine.

Il servizio

Il servizio si presentava come un comune sito web con una home page, un form di registrazione, uno di login e un eventuale form di cambio password.

Indizi

Il nome del servizio e la descrizione, vogliono fare un'allusione al tipo di vulnerabilità SQL, cioè un injection del secondo ordine.
Un altro indizio è possibile trovarlo ispezionando il codice della home dove esiste un commento html nel suo footer:

```
<!- -
Hey this is an hint:
you don’t need to know anything about the
quadratiequation,
but think about the term “quadratic” …
- - >
```

La vulnerabilità

Registrando un utente attraverso la pagina dedicata, tutto va a buon fine. Provando a registrare un altro utente avente lo stesso username del precedente, viene visualizzato il seguente messaggio:

Username already exists!

Questo è chiaramente un punto debole da cui trarne profitto. È possibile quindi fare un attacco al campo username. In particolare risulta utile la scelta di un attacco di tipo bruteforce per enumerare tutti gli eventuali users già registrati.
Dal risultato ottenuto dall’attacco, vale la pena attenzionare gli account **root** e **admin**.

Avendo la possibilità di cambio password del proprio account, andiamo ad ispezionare il codice del file **profile.php**. Attenzioniamo la query di modifica della password:

```
$sql=“UPDATE Users SET password=‘$newpassword’ WHERE
username=‘$username’ AND password=‘$oldpassword’ “;
```

Il controllo degli apici è uno dei principali meccanismi di difesa per evitare attacchi di tipo SQL Injection. Nel codice è possibile notare l'assenza di un qualsiasi tipo di sanificazione dell'input. Andremo a sfruttare tale vulnerabilità per exploitare il servizio.

L'Exploit

Creiamo un nuovo account chiamato **root' --** avente una qualsiasi password, ad esempio **123**. Effettuiamo il login e andiamo a modificare la password in **12345**. La query che si occupa della modifica interpreterà in modo anomalo l'update della password. In particolare:

```
UPDATE Users SET password='12345' WHERE username='root' --' AND password='123'
```

La parte di query evidenziata in rosso verrà ignorata poiché risulta essere un commento per MySQL. Il commento è ottenuto grazie ai due trattini presenti nell'username. Così abbiamo cambiato la password dell'utente **root** in **12345** senza averne accesso.
Adesso è possibile fare il login nell'account **root** con la password appena inserita, ovvero **12345**.
Una volta dentro troviamo un indizio nella sua biografia che ci suggerisce di andare a controllare l'user **admin**.
Con lo stesso metodo appena visto, creiamo un account **admin' --** e reiteriamo il procedimento di modifica password.

In questo modo riusciremo a cambiare la password di **admin** e, un volta loggati con tale account, riusciremo a trovare la flag nella sua biografia.

La patch

Una possibile patch per poter risolvere è quella di prevedere un escaping dei caratteri in input, precisamente negli apici singoli e doppi.

Link

Link alla pagina github dell'evento: https://gitlab.com/shides-ac/quadraticequation