SQL Injection prin Burp

Burp sau **Burp Suite** este un instrument grafic pentru testarea securității aplicațiilor web. Instrumentul este scris în Java și dezvoltat de PortSwigger Security.

Burp Decoder - este instrumentul ce are două versiuni: o versiune gratuită și o versiune completă care poate fi achiziționată după o perioadă de probă (Professional Edition). Versiunea gratuită are redusa semnificativ funcționalitatea. Acesta a fost dezvoltat pentru a oferi o soluție completă pentru verificarea securității aplicațiilor web. În plus față de funcționalitatea de bază, cum ar fi proxy server, scanner și intrus, instrumentul conține și opțiuni mai avansate, cum ar fi un spider, un repetor, un decodor, un comparter, un extender și un sequencer.

Compania din spatele suitei Burp a dezvoltat, de asemenea, o aplicație mobilă care conține instrumente similare compatibile cu iOS 8 și mai sus.

Instrumente sunt următoarele :

- Proxy HTTP - funcționează ca un server proxy web și este așezat ca un intermediar între browser și serverele de destinație web. Aceasta permite interceptarea, inspecția și modificarea traficului brut care trece în ambele direcții.
- Scanner - un scaner de securitate pentru aplicații web, utilizat pentru scanarea automată a vulnerabilităților aplicațiilor web.
- Intrus - Acest instrument poate efectua atacuri automate asupra aplicațiilor web. Instrumentul oferă un algoritm configurabil care poate genera solicitări HTTP rău intenționate. Instrumentul intrus poate testa și detecta SQL Injections, Cross Site Scripting, manipularea parametrilor și vulnerabilitățile susceptibile de atac brutal.
- Spider - Un instrument pentru crawlerea automată a aplicațiilor web. Acesta poate fi utilizat împreună cu tehnicile de cartografiere manuală pentru a accelera procesul de mapare a conținutului și a funcționalității unei aplicații.
- Repetor - un instrument simplu care poate fi folosit pentru a testa manual o aplicație. Acesta poate fi folosit pentru a modifica cererile către server, pentru a le trimite din nou și pentru a observa rezultatele.
- Decodor - instrument de transformare a datelor codificate în forma sa canonică sau de transformare a datelor brute în diverse forme codate și rulate. Este capabil să recunoască în mod inteligent mai multe formate de codare folosind tehnici euristice.
- Comparer - Un instrument pentru efectuarea unei comparații (un "diff" vizual) între oricare două elemente de date.

- Extender - permite testerului de securitate să încarce extensiile Burp, pentru a extinde funcționalitatea lui Burp folosind codul de testare de securitate propriu sau al unui terț (BAppStore)
- Sequencer - instrument de analiză a calității aleatoare într-un eșantion de elemente de date. Acesta poate fi folosit pentru a testa jetoanele de sesiune ale unei aplicații sau alte elemente de date importante care sunt intenționate a fi imprevizibile, cum ar fi jetoanele anti-CSRF, jetoanele de resetare a parolei etc.

Ce este injecția SQL

Mulți developeri web nu știu cum pot fi manipulate interpelările SQL, și acordă toata încrederea unei asemenea comenzi. Interpelările SQL pot ocoli controalele de acces, în consecință să treacă peste metodele de autentificare și verificările de autorizație, iar câteodată pot chiar să faciliteze accesul la comenzile de sistem.

Injectarea directă a comenzilor SQL este o tehnică în care atacatorul creează sau modifică comenzile SQL pentru a scoate la iveală datele sensibile, sau pentru a suprascrie o anumită valoare, sau chiar pentru a executa comenzi periculoase la nivel de sistem. Acest lucru este înfaptuit de către aplicația care preia inputul utilizatorului, îl combină cu parametrii statici pentru a forma o interpelare SQL. Următoarele exemple sunt bazate pe cazuri reale, cu regret.

Datorită lipsei validării inputului și conectării la baza de date cu drepturi de superuser, sau a unui user care poate crea la rândul lui alți useri, atacatorul poate crea un superuser în baza de date.

O reală posibilitate de a afla parole este de a manipula rezultatele din paginile de căutare. Singurul lucru de care are nevoie atacatorul este să vadă dacă există variabile în declarațiile SQL care nu sunt protejate corespunzător. Se pot manipula variabilele din formularele care utilizează *WHERE, ORDER BY, LIMIT* sau condițiile *OFFSET* din declarațiile *SELECT*. Dacă baza de date suportă construcții *UNION*, atacatorul poate încerca să lipească o interpelare întreagă la cea originală pentru a lista parolele dintr-un tabel arbitrar. Folosirea parolelor criptate este pe deplin încurajată.

Partea statică a interpelării poate fi combinată cu înca un *SELECT* care să arate parolele:

'union select '1', concat(uname||'-'||passwd) as name, '1971-01-01', '0' from usertable;--

Dacă această interpelare (ne-am jucat cu ′ și --) ar fi fost atribuită unei variabile utilizate la formarea *$query*, am fi dat de belea.

Comanda SQL UPDATE nu este nici ea ocolită de probleme. Aceste interpelări sunt amenințate de atacurile prin tăierea și alipirea unei noi interpelări. În plus, atacatorul se mai poate juca și cu declarația *SET*. În acest caz, atacatorul trebuie să cunoască careva informații despre

schemă, de ex. structura tabelului din care dorește să extragă sau să manipuleze informația. Acest lucru poate fi facut prin examinarea denumirilor variabilelor din formulare, sau prin procedeul brute-force. Nu există multe convenții prin care se delimitează câmpurile pentru user sau parolă.

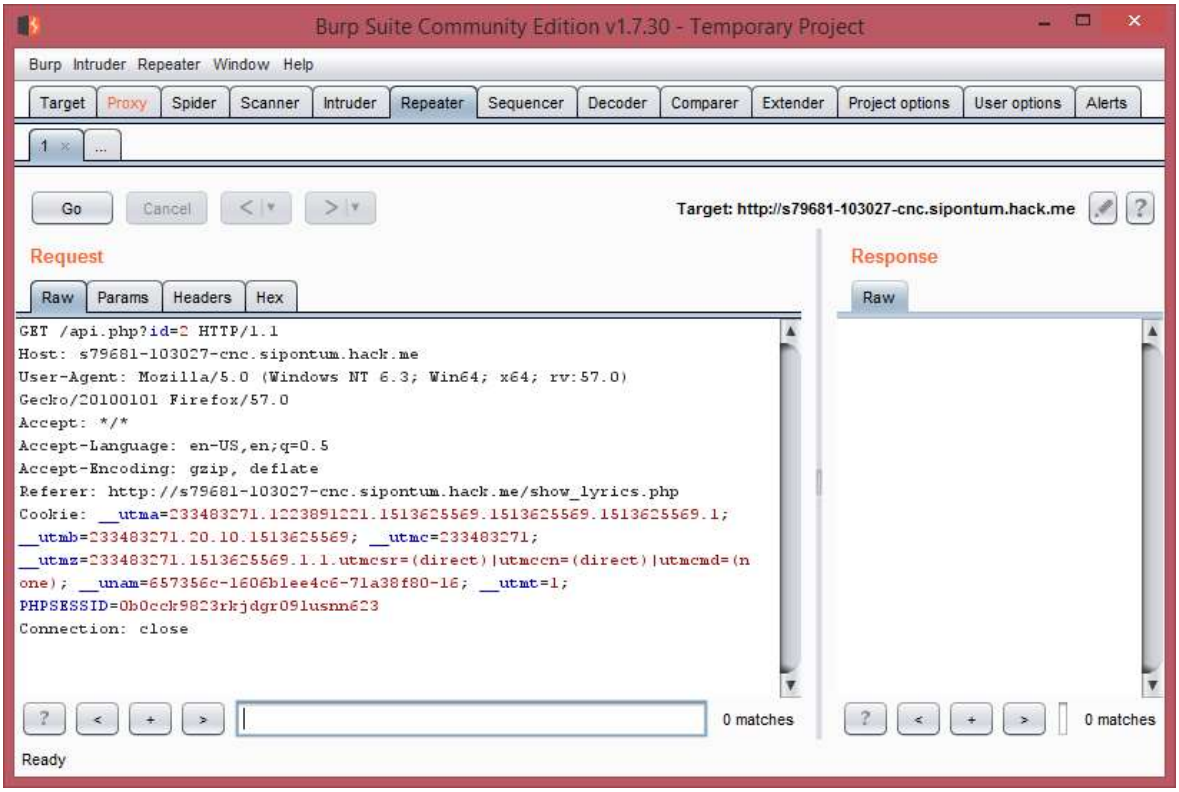
Aflarea datelor sensibile cu ajutorul injecției SQL

Pentru a face o demonstrare ajustată scopului lucrării cu Injecție SQL vom folosi drept țintă situl https://hack.me care permite acest lucru, deoarece este un site vulnerabil la penetrări, iar autorii acestuia l-au creat pentru aceasta. La inceput vom naviga pe site și creăm un cont, după care navigăm pe https://hack.me/103027/lyrics-app.html unde pe butonul start la afișarea sitului propriu zis unde parola de intrare este *guest guest*.

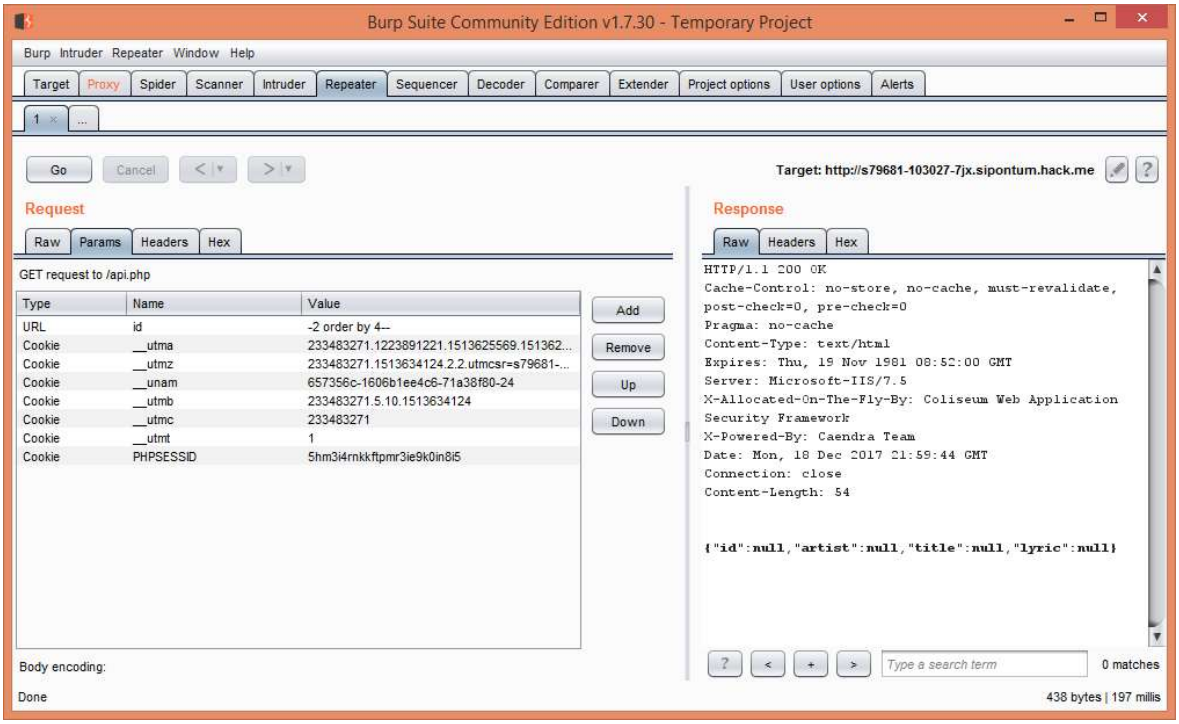
Pentru a face primul pas configurăm instrumentul Burp la Firefox astfel incit toate cererile din Firefox să treacă prin Burp (instrucțiunile a fi urmate sunt ușor de găsit pe web).

După ce ne-am logat cu guest guest mergem mai departe și facem un request pe care-l interceptăm in burp (e suficient să clicăm pe un meniu). Interceptând cererea, o transferăm în zona Repeater al Burp unde ulterior o modificăm după instrucțiunile afișate pe situl http://www.securityidiots.com/Web-Pentest/SQL-Injection/Basic-Union-Based-SQL-Injection.html

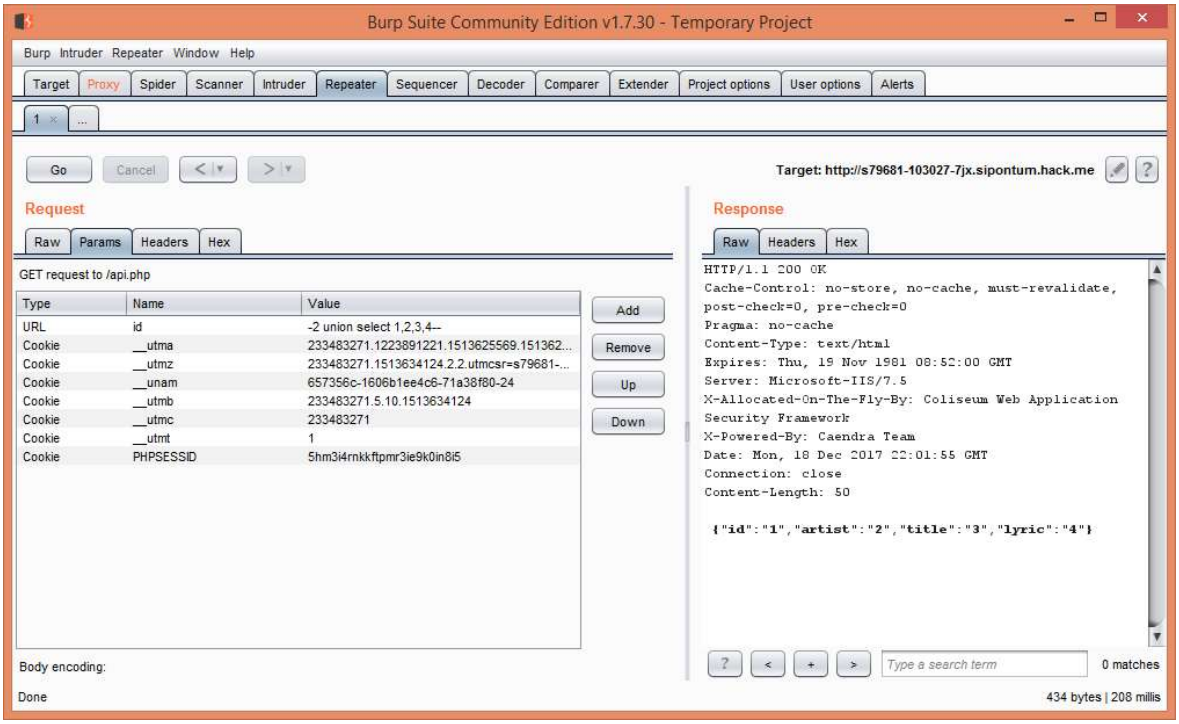
Astfel avem :



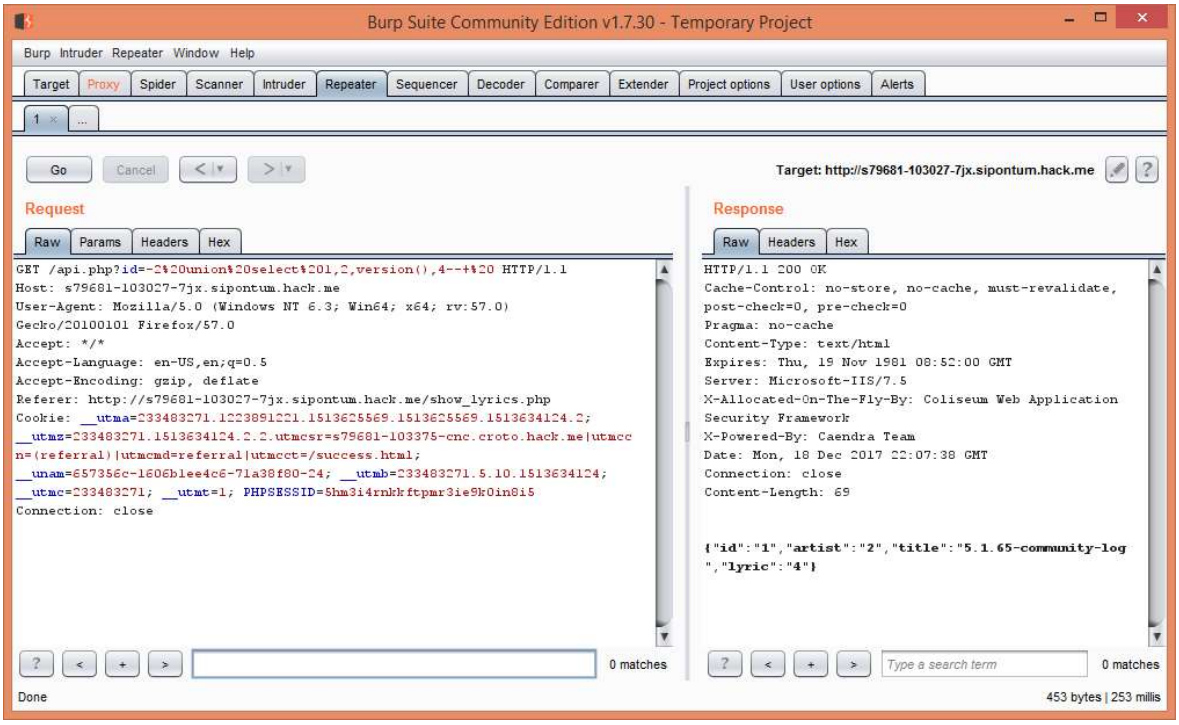
Pentru a face uz de Injecție vom modifica instrucțiunea GET atât timp cât vom primi un răspuns pozitiv și anume punem **order by 5--+** pentru inceput, iar dacă nu merge iterăm până dăm de numărul corect de coloeane, în cazul dat este 4 (dar modificam și id-ul cu unul invalid pentru a primi un răspuns pe pagină) :



Observăm rezultatele tabelului actual in răspunsul din BURP. Acum știm care sunt numele celor patru coloane.

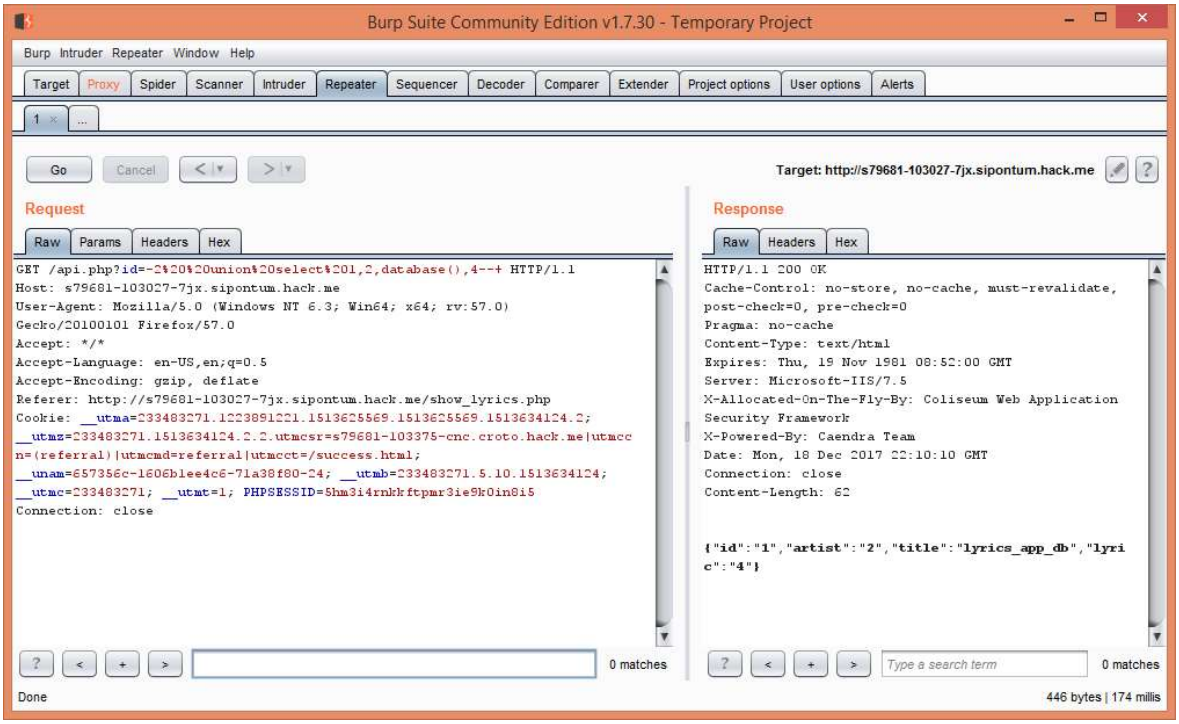


Aflăm care ste baza de date și versiunea acestea introducând GET /api.php?id=-2%20union%20select%201,2,version(),4--+%20 HTTP/1.1

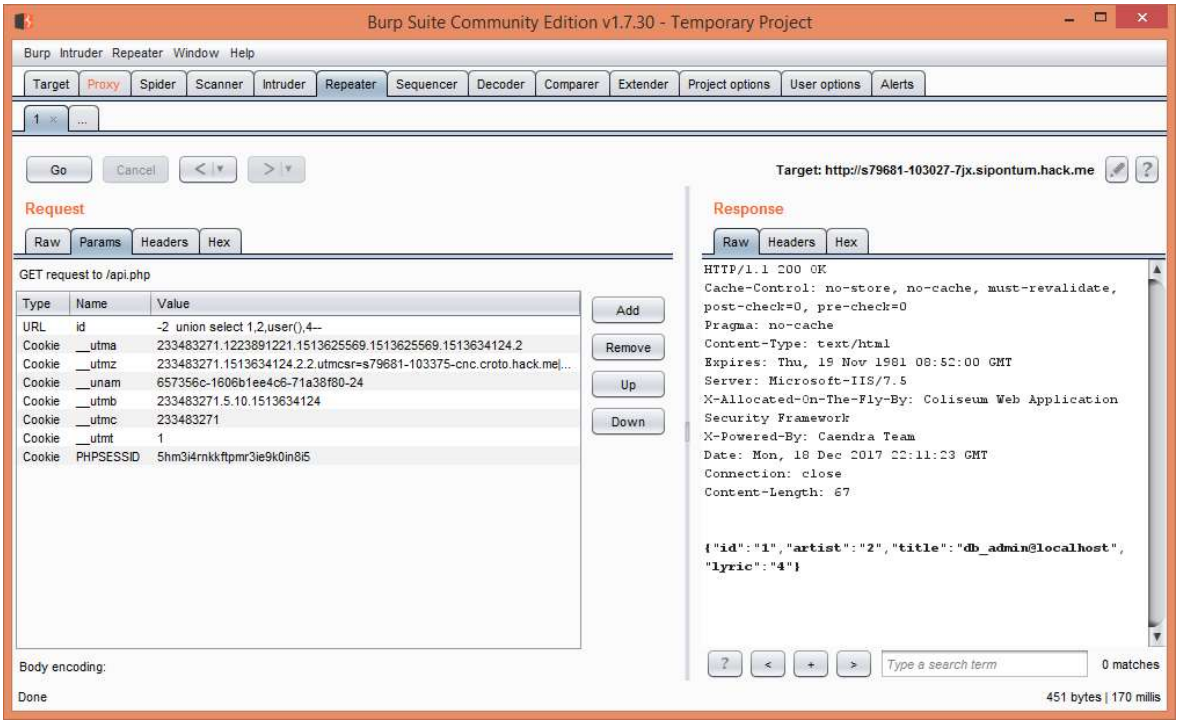


Observăm că este vorba de MySQL 5.1.65-community-log.

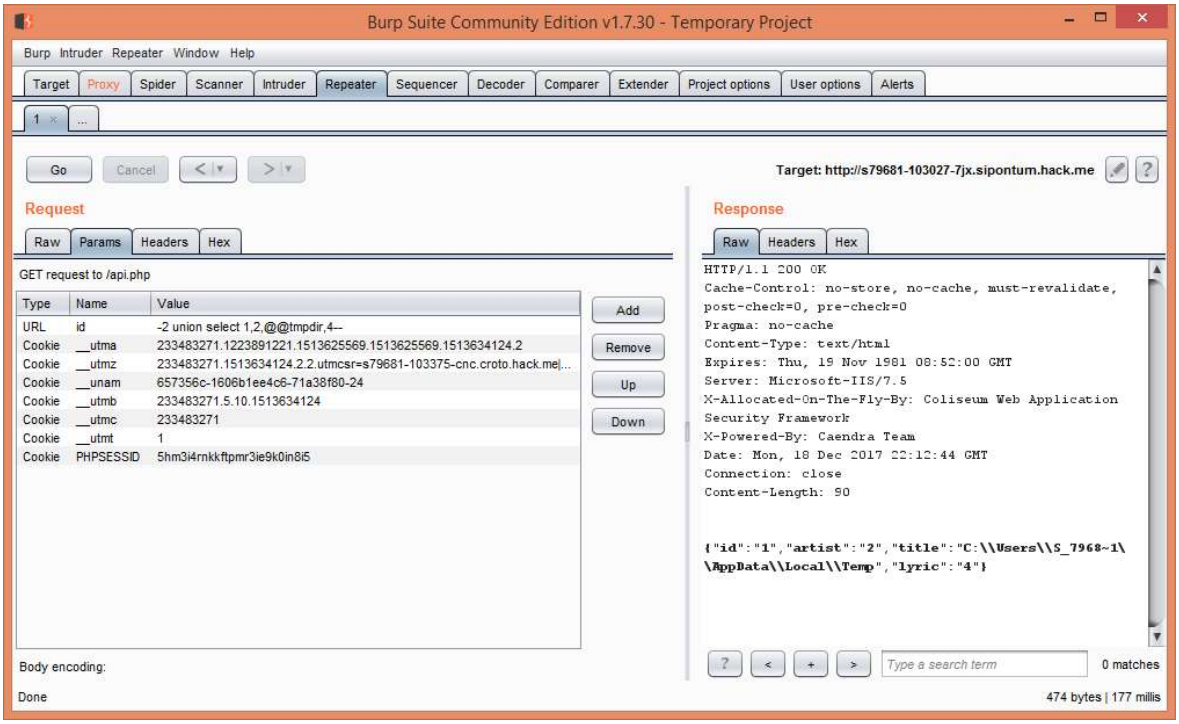
Aflăm în consecință și numele bazei de date :



Aflăm numele utilizatorului curent :



Aflăm numele și calea către directoriul temporar :



Astfel cu același succes prin penetrare SQL se pot afla și alte informații până la conținutul tabelelor, chiar dacă cheile sunt hashate cu MD5, acestea pot fi descifrate ușor online. Idea este că cele mai periculoase comenzi SQL in cazul dat devin cele de ștergere și anume *DROP TableName* sau *DROP DATABASE*.