Mason Competitive Cyber

Exploiting in AWS

But First… a PSA

- Metropolis crew rolling out
- picoCTF

AWS Primer

- Owns at least 30% of the cloud market
- Easily programmable cloud infrastructure
- Offers like 100+ services to do a variety of tasks
- Traditionally known as expensive, reliable
- Tries to ship secure by default
  - Doesn’t always achieve that

AWS Account Options

•

Regular	Starter Accounts
- Run solely by AWS - Can stockpile codes - 100000 horror stories of running over credit	- Basically no concept of IAM - No credit card needed - Externally managed - Blows up when credit expires - No adding after the fact

AWS Coding Challenge

Services

- Owns at least 30% of the cloud market
- Easily programmable cloud infrastructure
- Offers like 100+ services to do a variety of tasks
- Basically everything is an API call
- Traditionally known as expensive, reliable to use
  - If you know how AWS works
    - Nobody does
- Tries to ship secure by default
  - Doesn’t always achieve that

Security/Common Services

- CloudWatch: Alerting, not for forensic logging
- CloudTrail: 90% of the good logging
- EC2: Runs virtual servers. VMs are “instances”
- S3: Files often called “objects” or “keys”
- RDS: Managed Database Servers
- Config: Horrible name, moreso security alerting
- Inspector: Automated Security Assessments
- Macie: New - Data Classification

https://aws.amazon.com/products/security/

flaws.cloud

- AWS “CTF”
- **TONS** of hints
- Starts with S3, moves towards EC2, etc
  - Blends harder concepts later

On S3…

- Top level folders = “Buckets”
- Filenames = “Keys”, Files = “Objects”
- Used for everything from secure file storage to content delivery
  - This is the main issue
- Oftentimes buckets set to less secure than should be

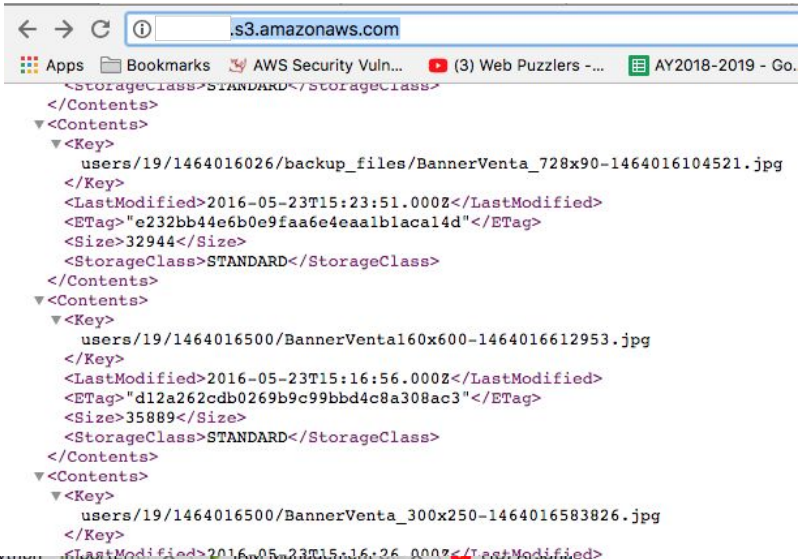
On Bucket Stream

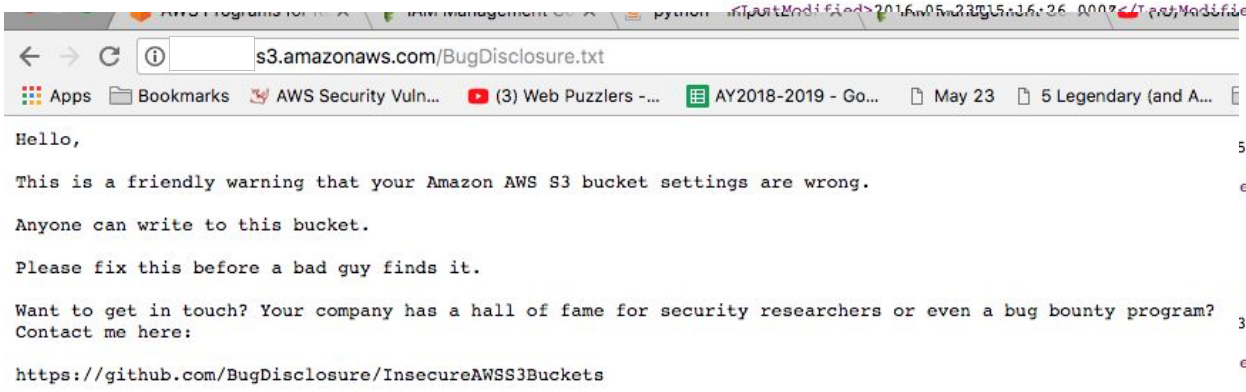
- On Github

```
96982 buckets checked (26b/s), 12 buckets found
97041 buckets checked (12b/s), 12 buckets found
97236 buckets checked (39b/s), 12 buckets found
97293 buckets checked (11b/s), 12 buckets found
97438 buckets checked (29b/s), 12 buckets found
97522 buckets checked (17b/s), 12 buckets found
97659 buckets checked (27b/s), 12 buckets found
97768 buckets checked (22b/s), 12 buckets found
Found bucket 'ispot-test.s3.amazonaws.com'. Owned by 'ecnuicloud'. ACLs = AllUsers: FULL_CONTROL | AuthenticatedUse
rs: (none)
97904 buckets checked (27b/s), 13 buckets found
97983 buckets checked (16b/s), 13 buckets found
98126 buckets checked (29b/s), 13 buckets found
98165 buckets checked (8b/s), 13 buckets found
98329 buckets checked (33b/s), 13 buckets found
98457 buckets checked (26b/s), 13 buckets found
98594 buckets checked (27b/s), 13 buckets found
98667 buckets checked (15b/s), 13 buckets found
Found bucket 'signal.s3.amazonaws.com'. Owned by '(unknown)'. ACLS = (could not read)
98819 buckets checked (30b/s), 14 buckets found
98925 buckets checked (21b/s), 14 buckets found
99030 buckets checked (21b/s), 14 buckets found
99077 buckets checked (9b/s), 14 buckets found
99236 buckets checked (32b/s), 14 buckets found
99307 buckets checked (14b/s), 14 buckets found
99461 buckets checked (31b/s), 14 buckets found
99520 buckets checked (12b/s), 14 buckets found
99724 buckets checked (41b/s), 14 buckets found
99788 buckets checked (13b/s), 14 buckets found
99943 buckets checked (31b/s), 14 buckets found
100022 buckets checked (16b/s), 14 buckets found
100150 buckets checked (26b/s), 14 buckets found
100182 buckets checked (6b/s), 14 buckets found
100348 buckets checked (33b/s), 14 buckets found
```

On S3 Indexing

- XML like a lot of AWS
- Not always possible
- Anonymous “ListBuckets” call





On EC2

- Servers = Instances
- Security Groups are like firewall rules
- VPC step above a subnet
- Volumes are…. volumes
- Like most VPS providers
- Instance types determine specs, cost
- Windows and Linux
- Can be bid on, on demand, etc

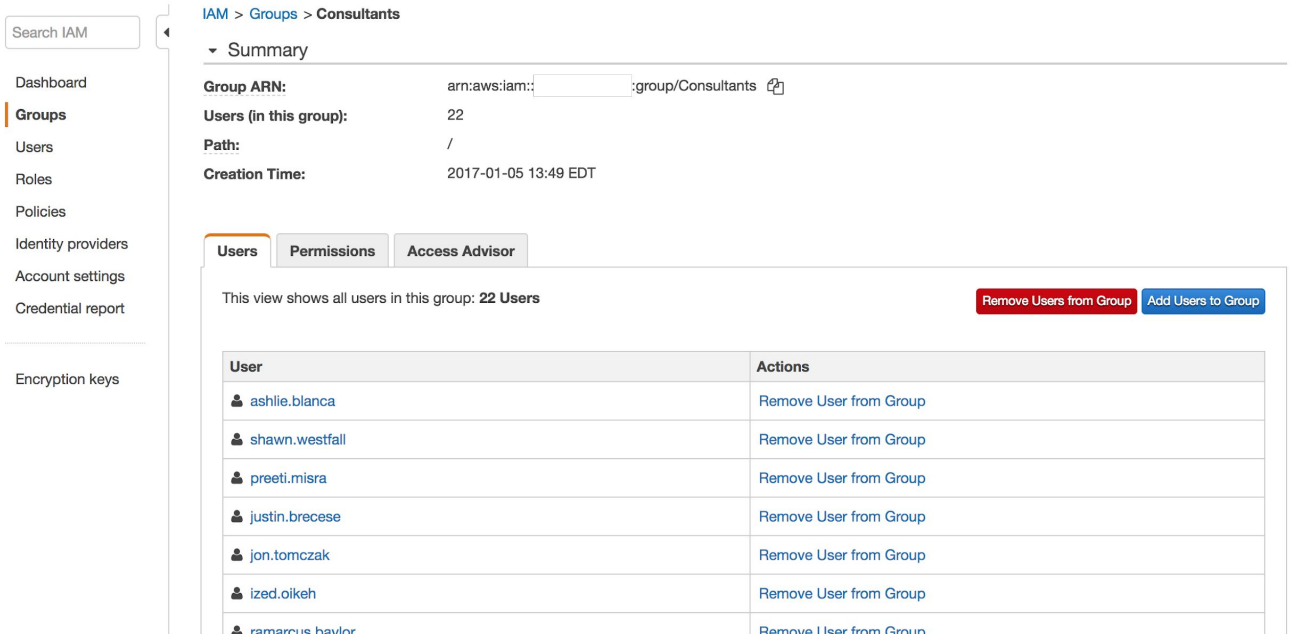
EC2 Metadata Service

- Pretty much always on
- Mileage may vary
- Can include AWS keys
- On “APIPA” address

```
root@ip-10-0-0-101:~# curl http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
2018-03-28
latestroot@ip-10-0-0-10url http://169.254.169.254/1.0/
meta-data
user-dataroot@ip-10-0-0url http://169.254.169.254/1.0/meta-data/
ami-id
ami-launch-index
ami-manifest-path
hostname
instance-id
local-ipv4
public-keys/
reservation-id
security-groupsroot@ip-10-0-0-101:~# curl http://169.254.169.254/1.0/meta-d
```

On IAM

- Identity and Access Management
- Who can log into console, who can have "programmatic access"
- Rarely would a CTF give console
  - Way too easy
  - Too much of a UI pain if limited perms



More on IAM

- JSON policies
  - Dictates what users/groups can, cannot access
- “Roles”
  - Like user permissions, for an AWS service
    - I personally seldom use them
- Makes revocation easy
- Keys can be invalid, grant no rights

AWS Keys and Leakage

```
Date: 2014-04-21 18:46:21
Branch: master
Commit: Removing aws keys

@@ -57,8 +57,8 @@ public class EurekaEVCacheTest extends AbstractEVCacheTest {
             //

             props.setProperty("        datacenter", "cloud");
-            props.setProperty("        awsAccessId", "<aws access id>");
-            props.setProperty("        awsSecretKey", "<aws secret key>");
+            props.setProperty("        awsAccessId", "AKIAJCK2WUHJ2653GNBQ");
+            props.setProperty("        awsSecretKey", "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");
             props.setProperty("        .appinfo.validateInstanceId", "false");

             props.setProperty("        .discovery.us-east-1.availabilityZones", "us-east-1c,us-east-1d,us-east-1e");
```

truffleHog, a popular Git key scanner

Key Leakage

- Popularly leaked on Github, elsewhere
- People get alerts in this case
- Git scanners open source
- **Deleting keys from code isn't enough**
  - Git keeps history, whole point of Git

Demonstrations

- Going through CC and maybe Crypsis's AWS

Questions?

- I've done **a lot** in AWS
- Dad leads intel community AWS sales
- My company does forensics ops in AWS
- If you ever see Niki, repeat AWS Sec intern

if not, work on flaws.cloud