Lift and Adrift

Understanding threats in an AWS environment

About Me

Jason Plummer

Senior Security Consultant @ Security Compass

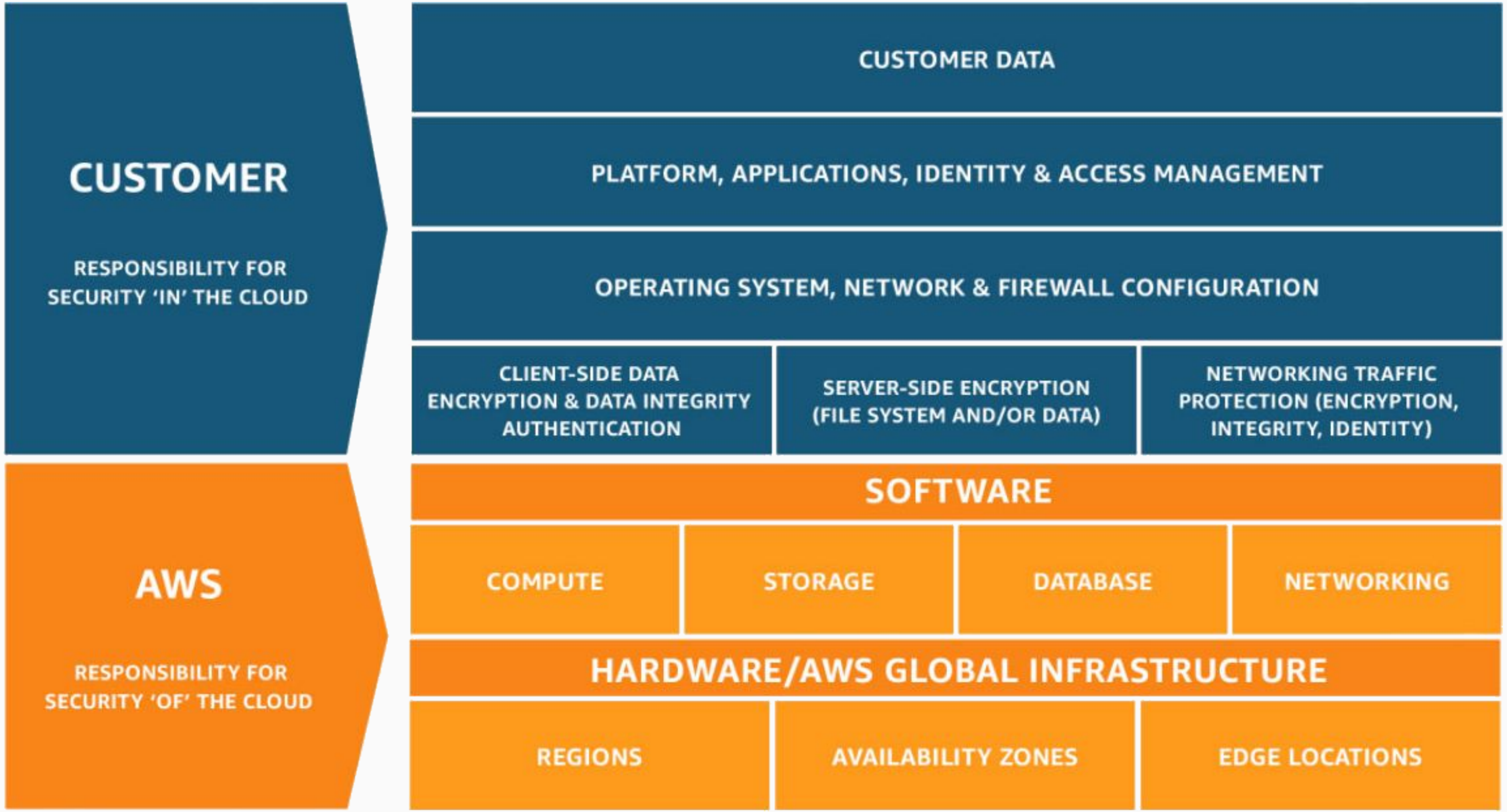
- AWS Certified Security - Specialty (+ others)
- Graduated from UOIT (now OnTechU)
- Have done some “cloudy” things

Agenda

- Introduction
- Threats to your AWS environment
    - Services “working as intended”
    - Permissions “a feature not a bug”
- Preventing the threats

Securing AWS

- AWS being secure does not equate to your data being secure
- Security is collaborative effort between AWS and the customer
  - Shared responsibility model
- Collaboration consists of two distinct components
  - Security **‘of’** AWS
  - Security **‘in’** AWS



CUSTOMER

RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD

CUSTOMER DATA

PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT

OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION

CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION

SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)

NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)

AWS

RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD

SOFTWARE

COMPUTE

STORAGE

DATABASE

NETWORKING

HARDWARE/AWS GLOBAL INFRASTRUCTURE

REGIONS

AVAILABILITY ZONES

EDGE LOCATIONS

Shared Responsibility Model - https://aws.amazon.com/compliance/shared-responsibility-model/

Summary

- Security responsibilities shared
  - AWS
  - Customers
- Security controls
  - AWS provides them but generally cannot predict appropriate usage
  - Customers must properly implement available controls to protect data

Services “working as intended”

Services “working as intended”

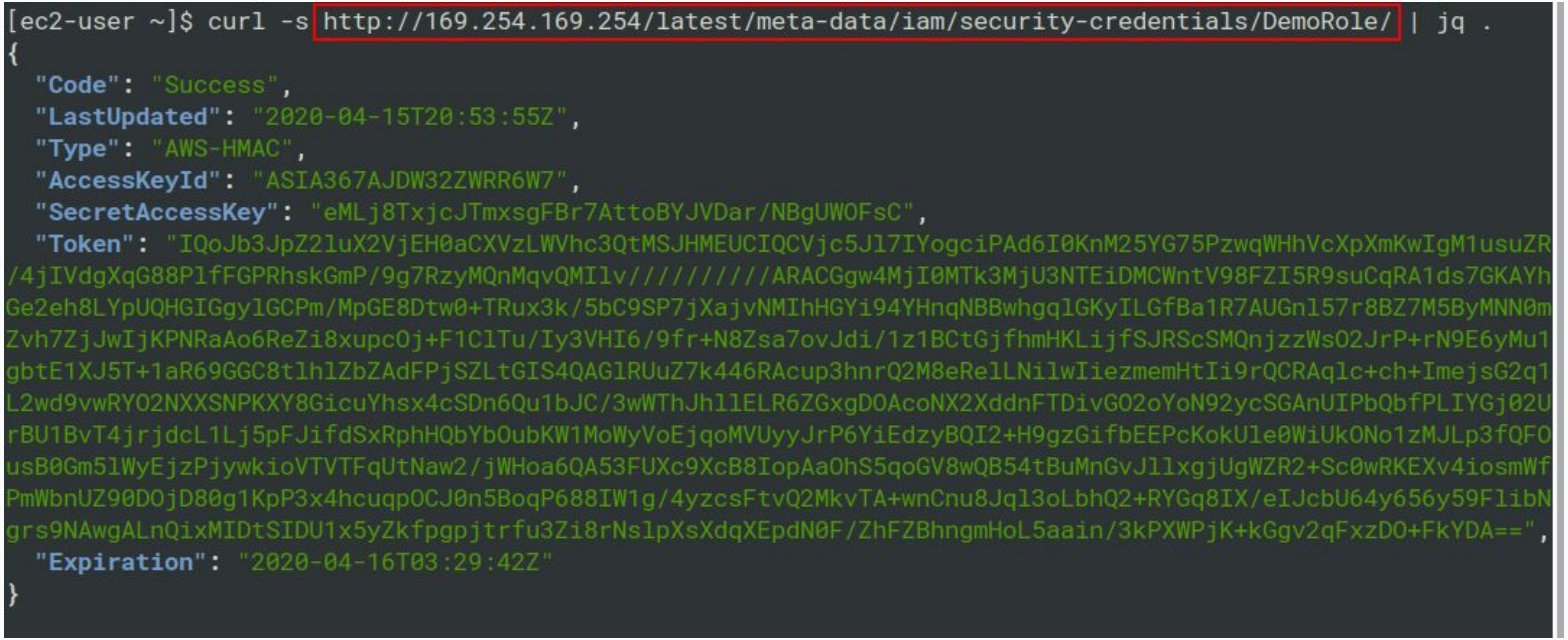
- Leaked credentials via Instance Metadata Service (IMDS)
- Subdomain takeovers via <multiple options>
- Secrets in storage via Elastic Block Storage (EBS)
- Local privilege escalation via Systems Manager (SSM)

Instance Metadata Service

- Instance metadata service (IMDS) accessible from EC2 instances

```
[ec2-user ~]$ curl http://169.254.169.254/
```

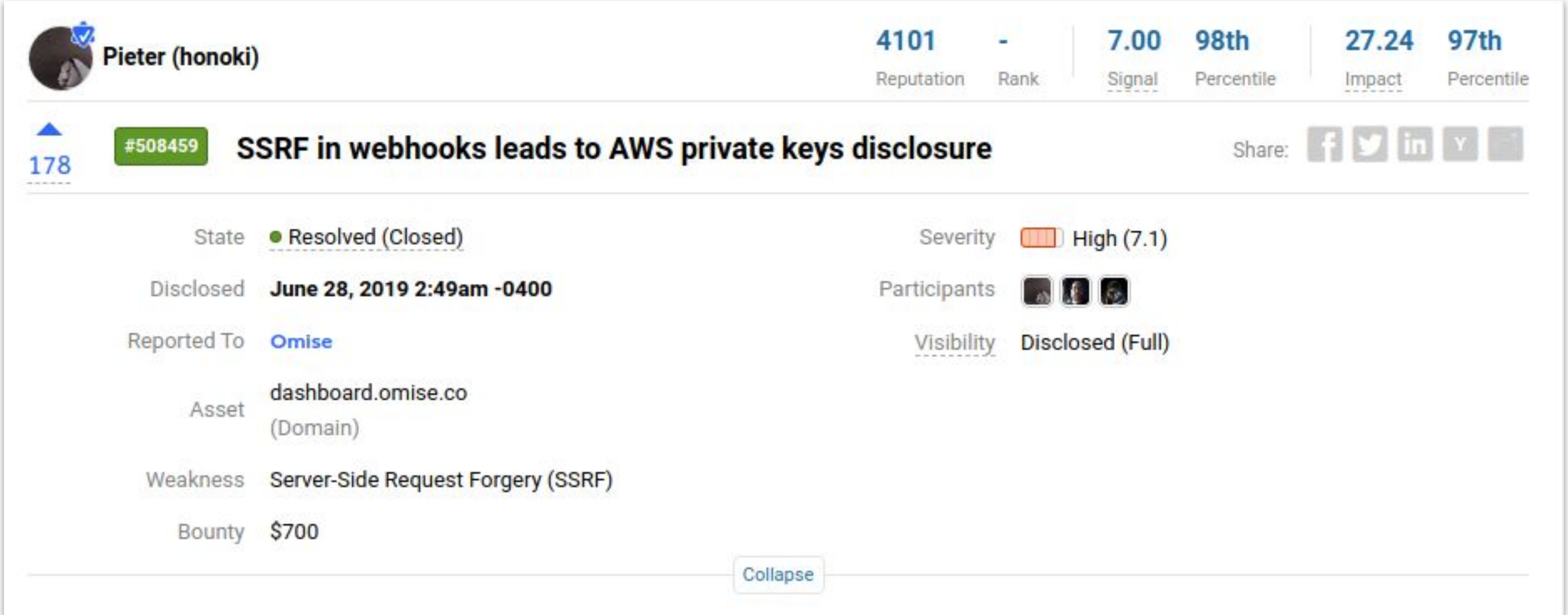
- Allows access to sensitive information
  - Metadata such as temporary credentials or configuration details
  - User data such as startup scripts and commands
- Requires requests to be made from the instance



```
[ec2-user ~]$ curl -s http://169.254.169.254/latest/meta-data/iam/security-credentials/DemoRole/ | jq .
{
  "Code": "Success",
  "LastUpdated": "2020-04-15T20:53:55Z",
  "Type": "AWS-HMAC",
  "AccessKeyId": "ASIA367AJDW32ZWRR6W7",
  "SecretAccessKey": "eMLj8TxjcJTmxsgFBr7AttoBYJVDar/NBgUWOFsC",
  "Token": "IQoJb3JpZ2luX2VjEH0aCXVzLWVhc3QtMSJHMEUCIQCVjc5Jl7IYogciPAd6I0KnM25YG75PzwqWHhVcXpXmKwIgM1usuZR
/4jIVdgXqG88PlfFGPRhskGmP/9g7RzyMQnMqvQMIlv//////////ARACGgw4MjI0MTk3MjU3NTEiDMCWntV98FZI5R9suCqRA1ds7GKAYh
Ge2eh8LYpUQHGIGgylGCPm/MpGE8Dtw0+TRux3k/5bC9SP7jXajvNMIhHGYi94YHnqNBBwhgqlGKyILGfBa1R7AUGnl57r8BZ7M5ByMNN0m
Zvh7ZjJwIjKPNRaAo6ReZi8xupcOj+F1ClTu/Iy3VHI6/9fr+N8Zsa7ovJdi/1z1BCtGjfhmHKLijfSJRScSMQnjzzWsO2JrP+rN9E6yMu1
gbtE1XJ5T+1aR69GGC8tlhlZbZAdFPjSZLtGIS4QAGlRUuZ7k446RAcup3hnrQ2M8eRelLNilwIiezmemHtIi9rQCRAqlc+ch+ImejsG2q1
L2wd9vwRYO2NXXSNPKXY8GicuYhsx4cSDn6Qu1bJC/3wWThJhllELR6ZGxgDOAcoNX2XddnFTDivGO2oYoN92ycSGAnUIPbQbfPLIYGj02U
rBU1BvT4jrjdcL1Lj5pFJifdSxRphHQbYbOubKW1MoWyVoEjqoMVUyyJrP6YiEdzyBQI2+H9gzGifbEEPcKokUle0WiUkONo1zMJLp3fQFO
usB0Gm5lWyEjzPjywkioVTVTFqUtNaw2/jWHoa6QA53FUXc9XcB8IopAaOhS5qoGV8wQB54tBuMnGvJllxgjUgWZR2+Sc0wRKEXv4iosmWf
PmWbnUZ90DOjD80g1KpP3x4hcuqpOCJ0n5BoqP688IW1g/4yzcsFtvQ2MkvTA+wnCnu8Jql3oLbhQ2+RYGq8IX/eIJcbU64y656y59FlibN
grs9NAwgALnQixMIDtSIDU1x5yZkfpgpjtrfu3Zi8rNslpXsXdqXEpdN0F/ZhFZBhngmHoL5aain/3kPXWPjK+kGgv2qFxzDO+FkYDA==",
  "Expiration": "2020-04-16T03:29:42Z"
}
```

Instances make requests

- Server-side request forgery (SSRF)
  - Causes requests to be made from instance
  - Allows access to instance metadata service
- Proxy traffic through an EC2 instance
  - Essentially SSRF by design



Pieter (honoki)

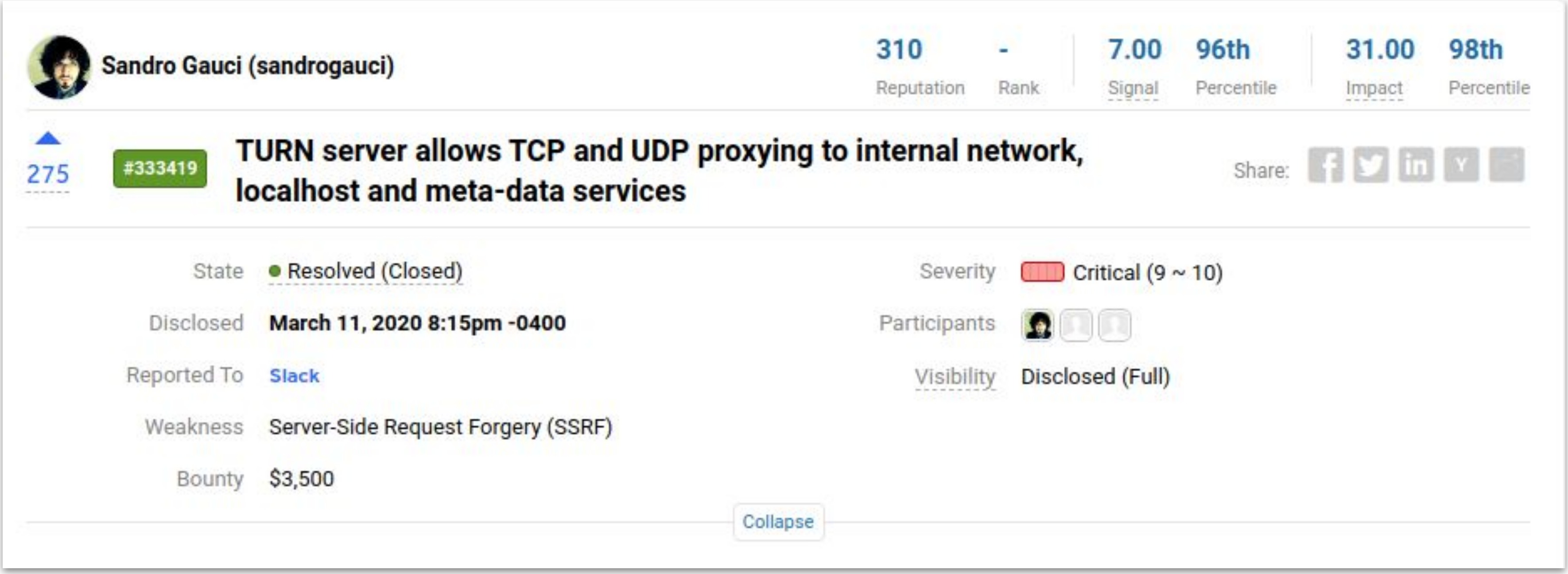
4101	-	7.00	98th	27.24	97th
Reputation	Rank	Signal	Percentile	Impact	Percentile

178

#508459 SSRF in webhooks leads to AWS private keys disclosure

State	● Resolved (Closed)	Severity	High (7.1)
Disclosed	**June 28, 2019 2:49am -0400**	Participants	
Reported To	Omise	Visibility	Disclosed (Full)
Asset	dashboard.omise.co (Domain)		
Weakness	Server-Side Request Forgery (SSRF)		
Bounty	$700		

https://hackerone.com/reports/333419



Sandro Gauci (sandrogauci)

310	-	7.00	96th	31.00	98th
Reputation	Rank	Signal	Percentile	Impact	Percentile

▲ 275

#333419

TURN server allows TCP and UDP proxying to internal network, localhost and meta-data services

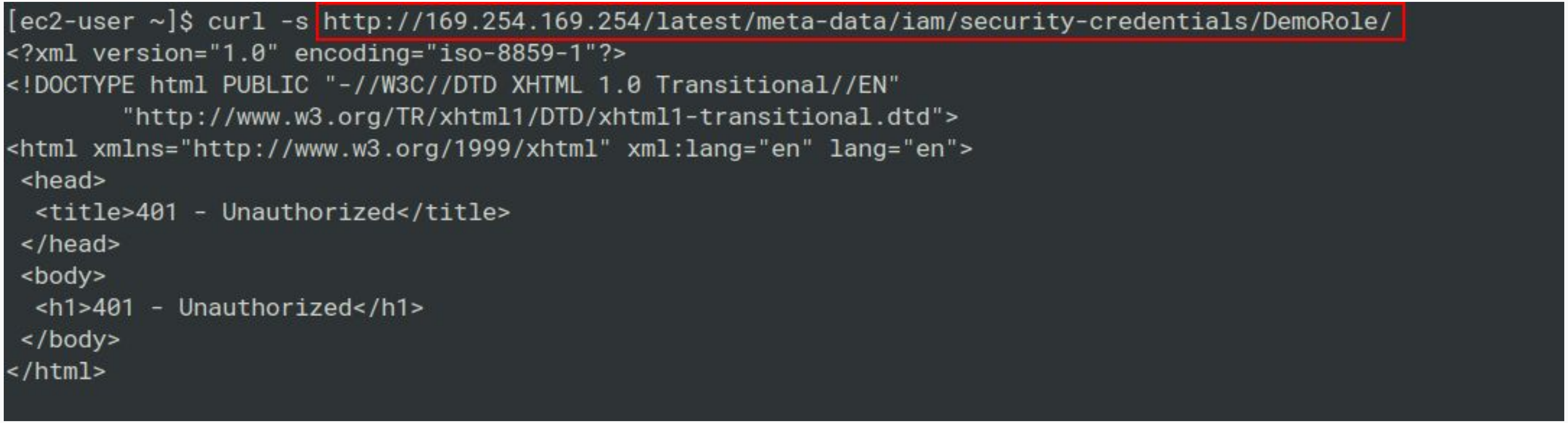
State	● Resolved (Closed)	Severity	🟥 Critical (9 ~ 10)
Disclosed	**March 11, 2020 8:15pm -0400**	Participants	
Reported To	Slack	Visibility	Disclosed (Full)
Weakness	Server-Side Request Forgery (SSRF)		
Bounty	$3,500		

https://hackerone.com/reports/508459

This can be fixed

- Use Instance Metadata Service version 2
  - Introduces a session based approach
  - Requires obtaining a token to use in subsequent requests

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" \
     -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`

[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v \
     http://169.254.169.254/latest/meta-data/
```



```
[ec2-user ~]$ curl -s http://169.254.169.254/latest/meta-data/iam/security-credentials/DemoRole/
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title>401 - Unauthorized</title>
 </head>
 <body>
  <h1>401 - Unauthorized</h1>
 </body>
</html>
```

Additional defense

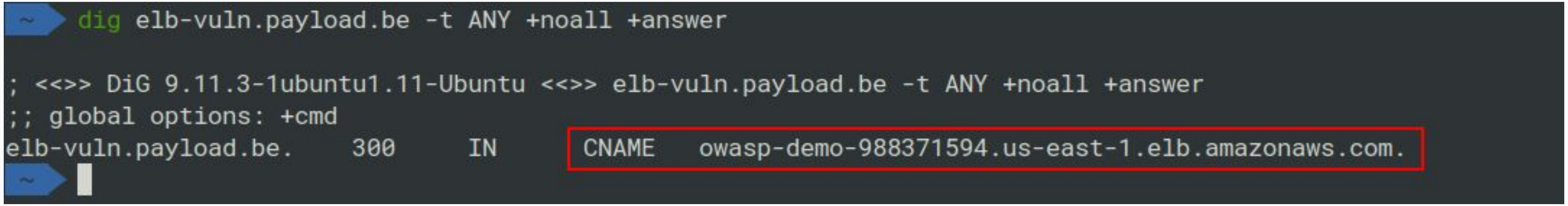
- Audit applications for weaknesses
  - Attack typically requires SSRF vulnerability
- Limit scope of instance profiles
  - Restricted permissions mitigate post exploit impact

Subdomain takeovers

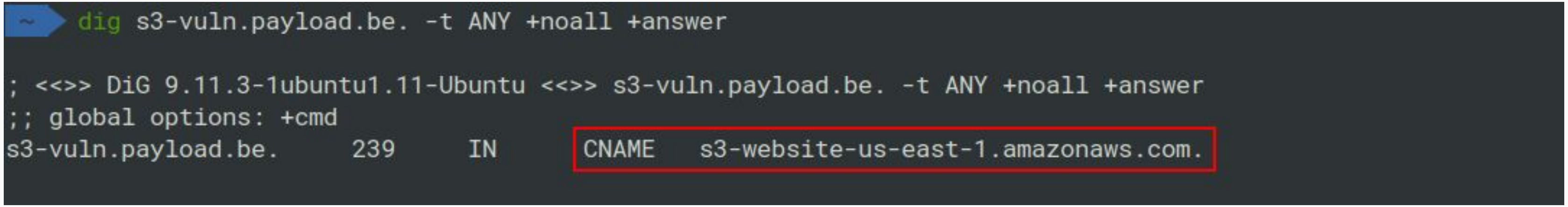
- Misconfigured DNS records exploited
  - Company creates new DNS record for subdomain that points to new service
  - Service no longer used by Company, but they don’t remove DNS record for subdomain
  - Attacker signs up for service and claims subdomain as theirs with no verification
  - Attacker can now leverage Company’s original subdomain to serve arbitrary content
- CNAME records typically used to map subdomain to service

AWS makes heavy usage of DNS

- Several AWS services generate domains for user resources
  - Custom domains can be used to access these resources
  - Commonly done using CNAME records
- Resources generally have predefined format
  - `d111111abcdef8.cloudfront.net`
  - `elb-123456789.us-east-1.elb.amazonaws.com`
  - `bucketname.s3-website-us-east-1.amazonaws.com`



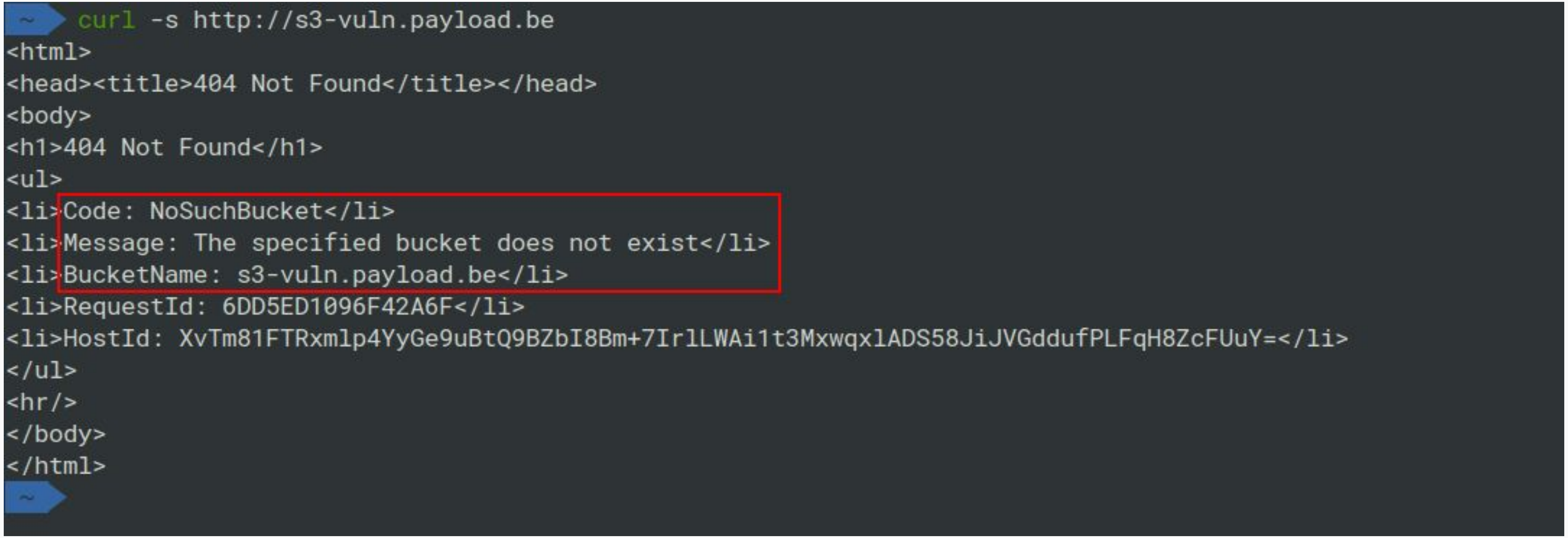
```
~  dig elb-vuln.payload.be -t ANY +noall +answer

; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> elb-vuln.payload.be -t ANY +noall +answer
;; global options: +cmd
elb-vuln.payload.be.    300     IN      CNAME   owasp-demo-988371594.us-east-1.elb.amazonaws.com.
~
```



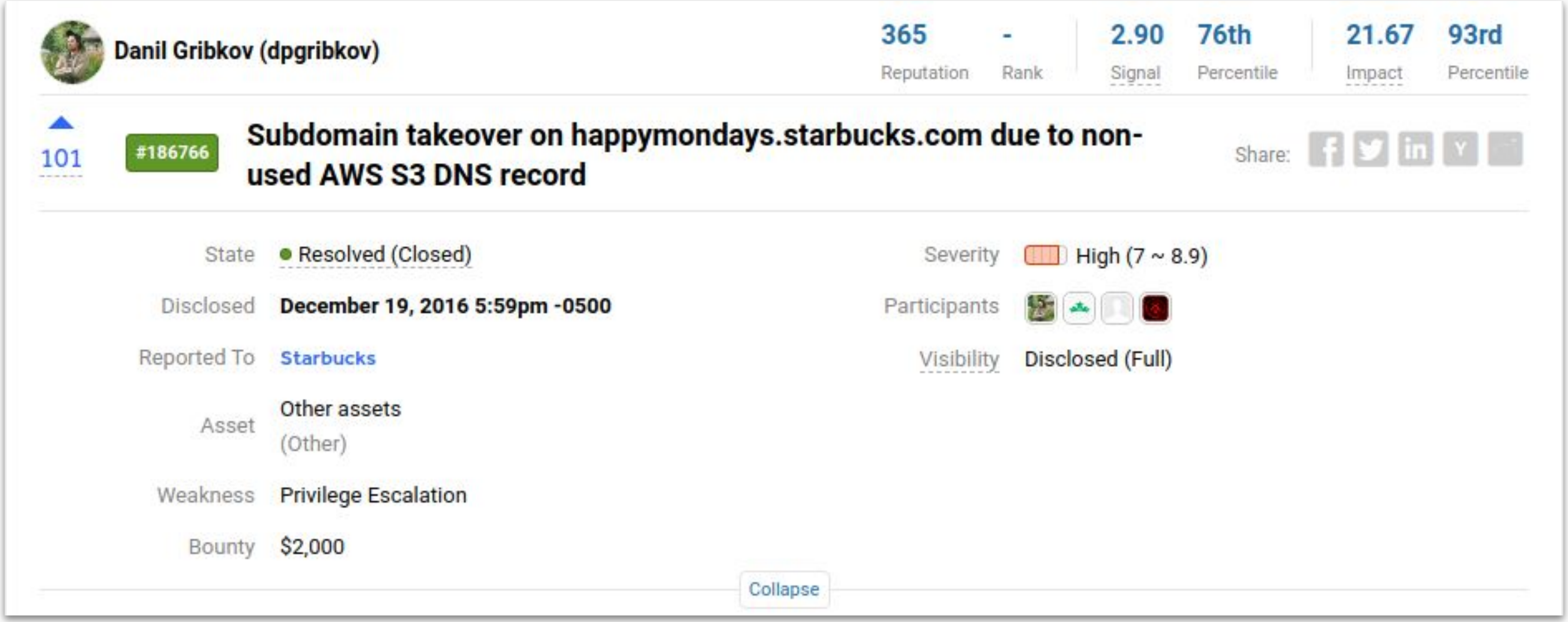
```
~  dig s3-vuln.payload.be. -t ANY +noall +answer

; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> s3-vuln.payload.be. -t ANY +noall +answer
;; global options: +cmd
s3-vuln.payload.be.     239     IN      CNAME   s3-website-us-east-1.amazonaws.com.
```

Dangling domains taken over

- Results in CNAME to AWS domains
  - `*.amazonaws.com`
  - `*.cloudfront.net`
- Resource domains are not restricted
  - Requires valid AWS account to create
  - Created resources are only regionally unique



```
~ curl -s http://s3-vuln.payload.be
<html>
<head><title>404 Not Found</title></head>
<body>
<h1>404 Not Found</h1>
<ul>
<li>Code: NoSuchBucket</li>
<li>Message: The specified bucket does not exist</li>
<li>BucketName: s3-vuln.payload.be</li>
<li>RequestId: 6DD5ED1096F42A6F</li>
<li>HostId: XvTm81FTRxmlp4YyGe9uBtQ9BZbI8Bm+7IrlLWAi1t3MxwqxlADS58JiJVGddufPLFqH8ZcFUuY=</li>
</ul>
<hr/>
</body>
</html>
~
```



Danil Gribkov (dpgribkov)

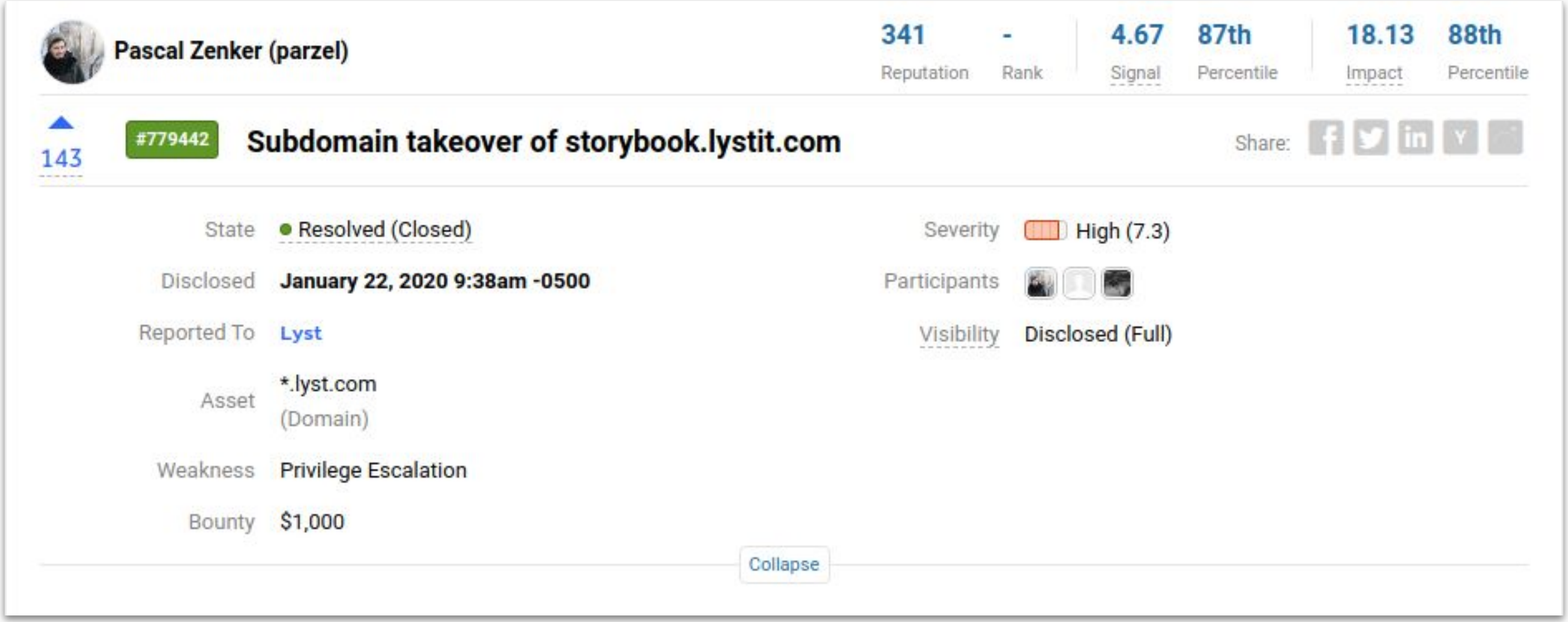
365	-	2.90	76th	21.67	93rd
Reputation	Rank	Signal	Percentile	Impact	Percentile

101

#186766

Subdomain takeover on happymondays.starbucks.com due to non-used AWS S3 DNS record

State	● Resolved (Closed)	Severity	High (7 ~ 8.9)
Disclosed	December 19, 2016 5:59pm -0500	Participants	
Reported To	Starbucks	Visibility	Disclosed (Full)
Asset	Other assets (Other)		
Weakness	Privilege Escalation		
Bounty	$2,000		



Pascal Zenker (parzel)

341	-	4.67	87th	18.13	88th
Reputation	Rank	Signal	Percentile	Impact	Percentile

143

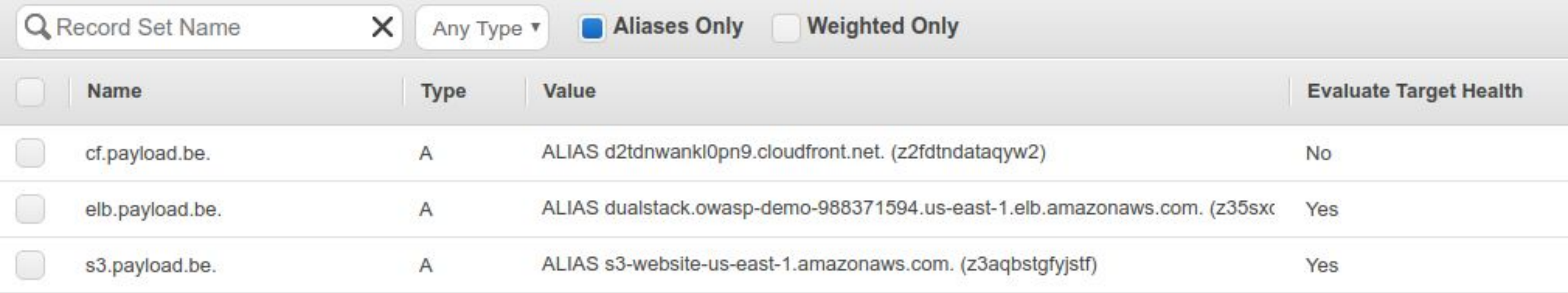
#779442 **Subdomain takeover of storybook.lystit.com**

Share:

State	● Resolved (Closed)
Disclosed	**January 22, 2020 9:38am -0500**
Reported To	Lyst
Asset	*.lyst.com (Domain)
Weakness	Privilege Escalation
Bounty	$1,000
Severity	High (7.3)
Participants	
Visibility	Disclosed (Full)

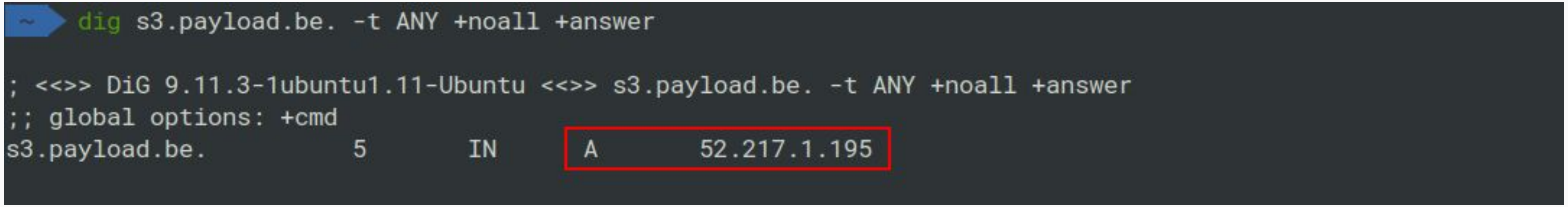
Get the hang of it

- Amazon provides some guard rails
  - Random component to domains generated by most resource types
  - Internal lock on resources when releasing them back into availability pool
- Route53 has an ALIAS target option
  - Replaces CNAME with a dynamically computed A record
  - Allows health check of target for certain resource types
  - Not perfect, especially for S3



Record Set Name ✕ | Any Type ▾ | ☑ Aliases Only | ☐ Weighted Only

☐	Name	Type	Value	Evaluate Target Health
☐	cf.payload.be.	A	ALIAS d2tdnwankl0pn9.cloudfront.net. (z2fdtndataqyw2)	No
☐	elb.payload.be.	A	ALIAS dualstack.owasp-demo-988371594.us-east-1.elb.amazonaws.com. (z35sxc	Yes
☐	s3.payload.be.	A	ALIAS s3-website-us-east-1.amazonaws.com. (z3aqbstgfyjstf)	Yes

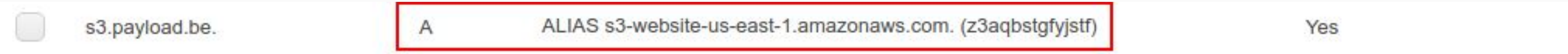


```
~ dig s3.payload.be. -t ANY +noall +answer

; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> s3.payload.be. -t ANY +noall +answer
;; global options: +cmd
s3.payload.be.          5       IN      A       52.217.1.195
```

S3 website with custom domain that is still at risk of domain takeover

There’s a quirk

- Health checks for S3 targets don’t work as expected
    - Endpoint always “healthy” due to virtual host routing
    - Even if bucket doesn’t exist, the subdomain will be resolvable



□	s3.payload.be.	A	ALIAS s3-website-us-east-1.amazonaws.com. (z3aqbstgfyjstf)	Yes

Improve the architecture

- Don’t dangle your domains!
  - Manage subdomains within Route53
  - Refine process for creating and removing resources (automation)
- Review best practices for recommended architectures
  - CloudFront distributions instead of S3 websites

Sharing secrets in storage

- Block storage
  - Provides raw access to disks (real of virtual) allowing bit-by-bit copy
  - Used by Amazon Elastic Block Storage (EBS)
- Object storage
  - Exposes files at higher level by abstracting actual filesystem
  - Used by Amazon S3

Scenario

- Building custom Amazon Machine Image (AMI) for public
    - Image is manually built from within the instance
    - Snapshots created once desired configuration attained
    - AMI generated from snapshot and made publicly accessible

Accidental sharing

- EBS-backed AMIs include snapshot of original volume
  - Previously deleted files may still exist on volume
- Contents of these files may be sensitive
  - Secrets used during provisioning
  - Internal configurations not intended for sharing
- Running a recovery tool may recover this data

It takes effort to fix

- Avoid widespread sharing of snapshots and AMIs where possible
- Develop clean pipeline for AMI generation for public images
  - Implement automation for AMIs intended to be shared
  - Ensure sensitive data is never present on volume

Local privilege escalation

- AWS Systems Manager (SSM)
  - Allows remote administration of hosts from central portal
  - Similar to tanium it allows remote command execution
- Evolution of AWS Systems Manager
  - Initially limited to command execution and inventory management
  - Features now include interactive shell and port forwarding

Scenario

- Developer uses SSM for interactive shell on host
    - Developer uses a language for which an AWS SDK exists
    - SSM interactive sessions configured to run as limited user
    - Host has associated instance profile with the following permissions
        - ssm:RunCommand
        - ssm:SendCommand

User might as well be root

- Users with access to a shell on the host can run AWS commands
  - AWS SDK will get credentials from IMDS to run commands
  - Permissions will allow user to send the host arbitrary commands
  - Commands will be executed as root user

Permissions have consequences

- Ensure appropriate permissions
  - Avoid temptation to simply apply SSMFullAccess managed policy
  - Permit individual actions only when necessary
  - Rarely ever a reason for hosts to have SSM:SendCommand

Permissions are “feature not bug”

Identity and Access Management (IAM)

- IAM entities referred to as principals
  - Users, Groups, Roles
  - AWS services or accounts
- Security policies are used to grant entities permissions or resources
  - Can relate to specific actions and/or resources
  - Can either allow or deny particular action/resource combinations
  - Can either be Identity Based or Resource Based

Security Policies (Identity Based)

- Policies consist of a collection of statements
    - Effect: defines whether statement should grant or restrict permissions (`Allow` or `Deny`)
    - Actions: collection of API service and operations referenced (`s3:*`)
    - Resources: specific resources referenced by Amazon Resource Name (`"*"`)
- Attached to IAM entities granting permissions

Dangerous Permissions

- Certain permissions allow direct modification of privileges
- Changing associated policies
  - Example: `iam:Attach(User|Group|Role)Policy`
  - Reason: allows additional policies (permissions) to be associated with iam entities
- Changing access to existing entities
  - Example: `iam:Create(AccessKey|LoginProfile)`
  - Reason: allows additional persons access to those IAM entities (and their permissions)

Chaining Permissions

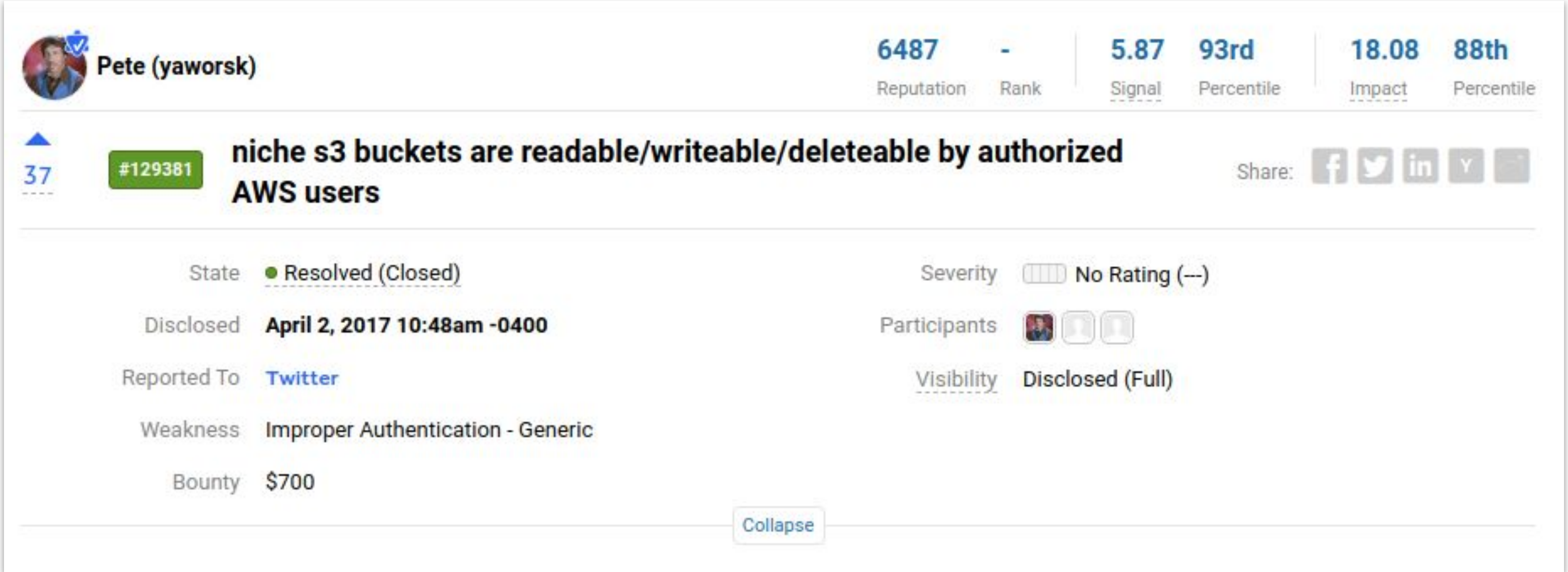
- Leveraging various services can result in additional permission sets
    - User A can only perform actions on Service A
    - Service A can perform actions on Service B
    - User A can then leverage Service A to perform actions on Service B
- Passing unexpected roles (`iam:PassRole`)
    - Users should not be allowed to pass arbitrary roles
    - Often results in escalation of privileges based on additional permissions in role passed

Scenario

- `Effect: Allow, Actions: [ec2:CreateInstance, iam:PassRole], Resource: "*"`
  - Role exists which grants administrative access, but user cannot assume it
  - Role has existing trust relationship with EC2
  - User creates a new EC2 instance and passes that role as the instance profile
  - User logs into new EC2 instance and can now perform administrative actions.

Missing Access Controls

- Several resources allow public access
  - Resources may contain sensitive data unintentionally shared
  - Breach notification littered with such incidents
- Commonly misconfigured services
  - S3
  - ElastiCache
  - ElasticSearch



Pete (yaworsk)

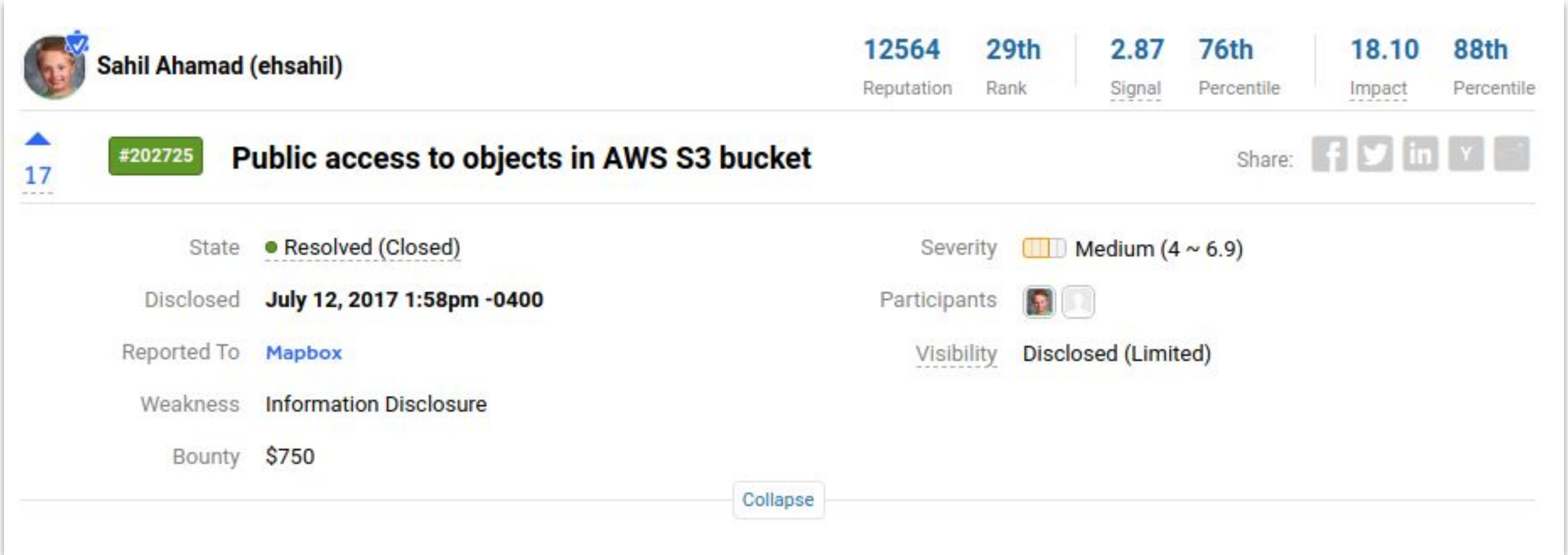
6487	-	5.87	93rd	18.08	88th
Reputation	Rank	Signal	Percentile	Impact	Percentile

37

#129381 **niche s3 buckets are readable/writeable/deleteable by authorized AWS users**

Share:

State	● Resolved (Closed)	Severity	No Rating (---)
Disclosed	**April 2, 2017 10:48am -0400**	Participants	
Reported To	Twitter	Visibility	Disclosed (Full)
Weakness	Improper Authentication - Generic		
Bounty	$700		



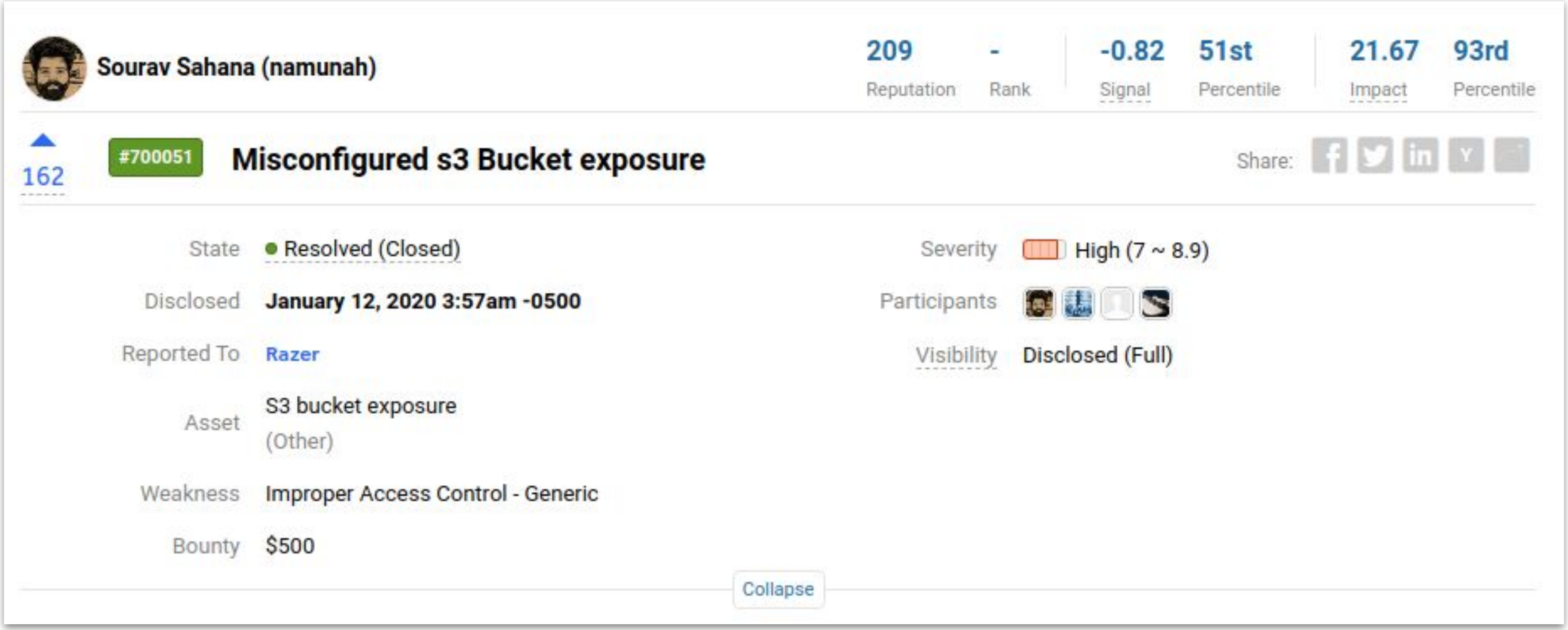
Sahil Ahamad (ehsahil)

12564	29th	2.87	76th	18.10	88th
Reputation	Rank	Signal	Percentile	Impact	Percentile

17

#202725 **Public access to objects in AWS S3 bucket**

State	● Resolved (Closed)	Severity	Medium (4 ~ 6.9)
Disclosed	**July 12, 2017 1:58pm -0400**	Participants	
Reported To	Mapbox	Visibility	Disclosed (Limited)
Weakness	Information Disclosure		
Bounty	$750		



Sourav Sahana (namunah)

209	-	-0.82	51st	21.67	93rd
Reputation	Rank	Signal	Percentile	Impact	Percentile

162

#700051 Misconfigured s3 Bucket exposure

Share:

State	● Resolved (Closed)	Severity	High (7 ~ 8.9)
Disclosed	**January 12, 2020 3:57am -0500**	Participants	
Reported To	Razer	Visibility	Disclosed (Full)
Asset	S3 bucket exposure (Other)		
Weakness	Improper Access Control - Generic		
Bounty	$500		

Preventing Threats

Managing Access

- Centralised authentication
  - Implement some type of SSO and map to predefined roles
  - Easier to manage a single directory with predefined roles
  - Alternative is managing individual users
- Appropriate access controls
  - Restrict access to resources whenever possible
  - Review any resource which is public to determine need

Managing Access

- Effective policies
  - Avoid relying on managed policies as a default
  - Use managed policies as basis then refine action and resource lists
- Dangerous permissions
  - Understand which permissions are dangerous (generally IAM related)
  - Restrict access to these permissions whenever possible
  - Avoid using antipatterns (such as “NotAction”) when possible

Managing Access

- Policy guardrails
  - Service control policies (SCP) and permission boundaries implement restrictions
  - These apply even despite allowed actions permitted by security policies
  - Permissions are the overlap between the security policy and SCP/permission boundary

Continuous Visibility

- CloudTrail audit logs
  - Ensure actions generate audit trails when performed
- Create alerts for sensitive actions
  - Credentials associated with EC2 instance being used from non-AWS IP address
  - Root account activity of any kind
  - Creation or attachment of any policy that grants administrative access

Understanding Services

- Ensure at least a basic understanding of service
  - Potential entry points
  - Service use case
- Researching services
  - Read available documentation
  - Experiment with the service
  - Search bug bounty disclosures for misconfigurations

Security of your AWS environment is a shared responsibility.

Services have security obligations, along with limits on their access controls.

Maintaining a secure environment requires continuous review.