Table of Contents

# Introduction

This guide provides setup guides and usage instructions for configuring Vulnerability Assessment (VA) scanning tools across AWS, Azure, and GCP environments.

# Tools Installation

### Python Version

* Python >3.9.1, <3.13

### Dependency Management

sudo apt update  
sudo apt install -y python3-poetry  
  
python3 -m pip install --upgrade pip  
pip install virtualenv

### AWS CLI

# macOS  
curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"  
sudo installer -pkg AWSCLIV2.pkg -target /  
  
# Linux x86  
sudo yum remove awscli  
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install  
  
# Linux ARM  
sudo yum remove awscli  
curl "https://awscli.amazonaws.com/awscli-exe-linux-aarch64.zip" -o "awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install

### Prowler

git clone https://github.com/prowler-cloud/prowler  
cd prowler  
poetry shell  
poetry install  
poetry run python ./prowler-cli.py --help

### ScoutSuite

git clone https://github.com/nccgroup/ScoutSuite  
cd ScoutSuite  
virtualenv -p python3 venv  
source venv/bin/activate  
pip install -r requirements.txt  
python scout.py --help

### CloudSploit

git clone https://github.com/aquasecurity/cloudsploit.git  
cd cloudsploit  
npm install `aws-sdk@2.1680.0` @azure/identity  
chmod +x ./index.js  
./index.js

# AWS

## Setting up authentication

### Configuring AWS CLI

For general use, the aws configure command is the fastest way to set up your AWS CLI installation. This configure wizard prompts you for each piece of information you need to get started. Unless otherwise specified by using the --profile option, the AWS CLI stores this information in the default profile.

The following example configures a default profile using sample values. Replace them with your own values as described in the following sections.

$ aws configure  
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE  
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY  
Default region name [None]: us-west-2  
Default output format [None]: json

The following example configures a profile named userprod using sample values. Replace them with your own values as described in the following sections.

$ aws configure --profile CSPM\_<Client>\_Role  
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE  
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY  
Default region name [None]: us-west-2  
Default output format [None]: json

### Check that you have set up your credentials

aws configure list [--profile profile-name]  
  
# Example Output  
$ aws configure list  
NAME : VALUE : TYPE : LOCATION  
profile : <not set> : None : None  
access\_key : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ABCD : config\_file : ~/.aws/config  
secret\_key : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ABCD : config\_file : ~/.aws/config  
region : us-west-2 : env : AWS\_DEFAULT\_REGION

### Setting up credentials for cloudsploit

* First copy the example config file

cp config\_example.js config.js

* Add your credentials into the following lines as shown below:

<SNIP>  
aws: {  
 // OPTION 1: If using a credential JSON file, enter the path below  
 // credential\_file: '',  
 // OPTION 2: If using hard-coded credentials, enter them below  
 access\_key: process.env.AWS\_ACCESS\_KEY\_ID || 'AKIA<SNIP>VRZFBL',  
 secret\_access\_key: process.env.AWS\_SECRET\_ACCESS\_KEY || 'KBrw4km<SNIP>fa40JfOL',  
 // session\_token: process.env.AWS\_SESSION\_TOKEN || '',  
 // plugins\_remediate: ['bucketEncryptionInTransit']  
 },  
<SNIP>

## Steps

### Running Scans

# Prowler  
# Default Run  
poetry run python ./prowler-cli.py aws  
  
# Execute checks based on requirements defined in compliance frameworks  
poetry run python ./prowler-cli.py aws --list-compliance  
poetry run python ./prowler-cli.py aws --compliance <compliance\_framework>  
  
# ScoutSuite  
python scout.py aws  
  
# CloudSploit  
./index.js --config=./config.js --cloud=aws --compliance=cis --csv=cloudsploit\_aws.csv --console=table

### Exporting Results

# Prowler  
# Output saved to ./output/ in html, csv and json formats. Recommended to export html.  
  
# ScoutSuite  
# Output saved to ./scoutsuite-report/ in html format with various assets. Recommended to export whole folder  
  
# CloudSploit  
# Output saved to filename under csv field

# Azure

## Setting up authentication

### Setting up credentials for cloudsploit

* First copy the example config file

cp config\_example.js config.js

* Add your credentials into the following lines as shown below:

<SNIP>  
azure: {  
 // OPTION 1: If using a credential JSON file, enter the path below  
 // credential\_file: '/path/to/file.json',  
 // OPTION 2: If using hard-coded credentials, enter them below  
 application\_id: process.env.AZURE\_APPLICATION\_ID || 'b73afeb4<SNIP>-276-8829-338dca105db3', // Client ID  
 key\_value: process.env.AZURE\_KEY\_VALUE || 'dtA8Q~MYW.3nchsu<SNIP>Azn-lA8QovRdelIcuv', // Password  
 directory\_id: process.env.AZURE\_DIRECTORY\_ID || 'bd146fad-35c5<SNIP>b1b2-ee634071b49e', // Tenant ID  
 subscription\_id: process.env.AZURE\_SUBSCRIPTION\_ID || 'fa956b60<SNIP>4016-a330-65b21200374f', // Subscription ID  
 // storage\_connection: process.env.AZURE\_STORAGE\_CONNECTION || '',  
 // blob\_container: process.env.AZURE\_BLOB\_CONTAINER || '',  
 // govcloud: process.env.AZURE\_GOV\_CLOUD || ''  
 },  
<SNIP>

## Steps

### Running Scans

# Prowler  
export AZURE\_CLIENT\_ID="b73afeb4<SNIP>338dca105db3"  
export AZURE\_TENANT\_ID="bd146fad<SNIP>e634071b49e"  
export AZURE\_CLIENT\_SECRET="dtA8Q<SNIP>lIcuv"  
poetry run python ./prowler-cli.py azure --sp-env-auth  
  
# ScoutSuite  
python scout.py azure --tenant <tenant id> --service-principal   
2025-10-06 10:24:19 <SNIP> scout[28193] INFO Launching Scout  
2025-10-06 10:24:19 <SNIP> scout[28193] INFO Authenticating to cloud provider  
Client ID: <application id>  
Client secret: <application secret value>  
  
# CloudSploit  
./index.js --config=./config.js --cloud=azure --compliance=pci --csv=cloudsploit\_azure.csv --console=table

### Exporting Results

# Prowler  
# Output saved to ./output/ in html, csv and json formats. Recommended to export html.  
  
# ScoutSuite  
# Output saved to ./scoutsuite-report/ in html format with various assets. Recommended to export whole folder  
  
# CloudSploit  
# Output saved to filename under csv field

# GCP

## Setting up authentication

### Setting up credentials for cloudsploit

* First copy the example config file

cp config\_example.js config.js

* Extract the relevant fields from the json file and fill in the fields in config.js as seen below

<SNIP>  
google: {  
 // OPTION 1: If using a credential JSON file, enter the path below  
 // credential\_file: process.env.GOOGLE\_APPLICATION\_CREDENTIALS || '/path/to/file.json',  
 // OPTION 2: If using hard-coded credentials, enter them below  
 project: process.env.GOOGLE\_PROJECT\_ID || 'trusty-diorama-474500-h1',  
 client\_email: process.env.GOOGLE\_CLIENT\_EMAIL || 'cspmscanner@trusty-diorama-474500-h1.iam.gserviceaccount.com',  
 private\_key: process.env.GOOGLE\_PRIVATE\_KEY || '-----BEGIN PRIVATE KEY-----<SNIP>\n-----END PRIVATE KEY-----\n'  
 },  
<SNIP>

## Steps

### Running Scans

# Prowler  
poetry run python ./prowler-cli.py gcp --project-ids <project id>  
  
# ScoutSuite  
python scout.py gcp --service-account </PATH/TO/KEY\_FILE.JSON>  
  
# CloudSploit  
./index.js --config=./config.js --cloud=google --compliance=pci --csv=cloudsploit\_gcp.csv --console=table

### Exporting Results

# Prowler  
# Output saved to ./output/ in html, csv and json formats. Recommended to export html.  
  
# ScoutSuite  
# Output saved to ./scoutsuite-report/ in html format with various assets. Recommended to export whole folder  
  
# CloudSploit  
# Output saved to filename under csv field