DevOps Shack

PROWLER

Click Here To Enrol To Batch-5 | DevOps & Cloud DevOps

Prowler is an open-source security tool designed for cloud environments, specifically for assessing the security posture of AWS (Amazon Web Services), Azure, and GCP (Google Cloud Platform) environments. Prowler is often used in DevOps processes to ensure that cloud resources are configured securely and to identify potential vulnerabilities.

Here is a detailed explanation of Prowler with examples:

What is Prowler?

Prowler is a security tool that automates security best practices on cloud platforms. It is written in bash and follows the CIS (Center for Internet Security) benchmarks. CIS benchmarks provide guidance for securely configuring various technologies, and Prowler helps to automate the checks against these benchmarks in cloud environments.

Features and Capabilities:

1. **Automated Checks:**
    - Prowler automates the process of checking cloud configurations against predefined security benchmarks. This includes checking for compliance with industry standards and best practices.
2. **CIS Benchmark Compliance:**

- Prowler follows the CIS benchmarks for AWS, Azure, and GCP. These benchmarks provide guidelines for securing cloud environments and are widely recognized in the industry.

3. **Multiple Cloud Platforms:**
    - Prowler supports AWS, Azure, and GCP, making it versatile for organizations using different cloud providers.
4. **Continuous Security Monitoring:**
    - Prowler can be integrated into continuous integration/continuous deployment (CI/CD) pipelines or used in regular security assessments to ensure ongoing security monitoring.

Installation & usage

First Create an AWS user with below permissions & Policies

To create an AWS user with the specified policy and permissions, you can follow these steps. I'll provide instructions using the AWS Management Console:

1. **Sign in to the AWS Management Console:**
    - Go to AWS Management Console.
    - Sign in with your AWS account credentials.
2. **Navigate to the IAM Console:**
    - In the AWS Management Console, navigate to the IAM (Identity and Access Management) service.
3. **Create a New User:**
    - In the IAM dashboard, click on "Users" in the left navigation pane.
    - Click on the "Add user" button.
4. **Configure User Details:**
    - Enter a username for the new user.
    - Choose the access type. For programmatic access, select "Programmatic access."
    - Click "Next: Permissions."
5. **Attach Policies:**
    - In the "Set permissions" section, choose "Attach existing policies directly."

- Search for and attach the policies: "SecurityAudit" and "ViewOnlyAccess."

6. **Add Inline Policy:**

   - In the "Set permissions" section, click on "Attach policies" and then "Attach policies directly."
   - Choose "Add inline policy."
   - Enter a name for the policy and click on the "JSON" tab.
   - Paste the below JSON policy document into the editor. Certainly! Below is the provided JSON policy formatted in Markdown:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "account:Get*",
        "appstream:Describe*",
        "appstream:List*",
        "backup:List*",
        "cloudtrail:GetInsightSelectors",
        "codeartifact:List*",
        "codebuild:BatchGet*",
        "dlm:Get*",
        "drs:Describe*",
        "ds:Get*",
        "ds:Describe*",
        "ds:List*",
        "ec2:GetEbsEncryptionByDefault",
        "ecr:Describe*",
        "ecr:GetRegistryScanningConfiguration",
        "elasticfilesystem:DescribeBackupPolicy",
        "glue:GetConnections",
        "glue:GetSecurityConfiguration*",
        "glue:SearchTables",
        "lambda:GetFunction*",
        "logs:FilterLogEvents",
        "macie2:GetMacieSession",
        "s3:GetAccountPublicAccessBlock",
        "shield:DescribeProtection",
        "shield:GetSubscriptionState",
        "securityhub:BatchImportFindings",
        "securityhub:GetFindings",
        "ssm:GetDocument",
        "ssm-incidents:List*",
        "support:Describe*",
        "tag:GetTagKeys",
        "wellarchitected:List*"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowMoreReadForProwler"
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET"
      ],
        "Resource": [
          "arn:aws:apigateway:*::/restapis/*",
          "arn:aws:apigateway:*::/apis/*"
        ]
      }
    ]
}
```

7. **Review and Create:**

   - Review the configurations and policies attached.
   - Click "Next: Tags" if you want to add tags (optional).
   - Click "Next: Review."

8. **Review the Configuration:**

   - Review the user details, attached policies, and inline policy.
   - Click "Create user."

9. **Download Access Credentials:**

   - Once the user is created, you'll see a confirmation screen.
   - Download the user's access key ID and secret access key. This information is crucial for programmatic access.

10. **Record User Details:**

- Record the user's details, including the access key ID, secret access key, and user ARN.

The user is now created with the specified policies and permissions. Ensure that you securely manage the access key credentials, and provide the necessary details to the user for programmatic access.

Install AWS CLI:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
sudo apt install unzip
unzip awscliv2.zip
sudo ./aws/install
aws configure
```

Install Python:

```
sudo apt update
sudo apt install python3.9
python3.9 --version
```

Install Python Virtual Environment:

```
sudo apt update
sudo apt install python3.9-venv python3.9-distutils
wget https://bootstrap.pypa.io/get-pip.py
sudo python3.9 get-pip.py
pip3.9 --version  # OR pip --version
```

Install CFFI:

```
pip3.9 install cffi

# Check whichever package is uninstalled, install it as shown in video
```

Install Prowler:

```
pip install prowler
prowler -v
```

Usage Examples:

1. **List Categories and Compliance:**

```
2. prowler aws --list-categories
   prowler aws --list-compliance
```

3. **Run Prowler Scan:**

```
prowler aws
```

4. **List AWS Services and Run Specific Services:**

```
5. prowler aws --list-services
   prowler aws --services s3 ec2
```

6. **Exclude Specific Checks:**

```
prowler aws --excluded-checks s3_bucket_public_access
```

7. **Specify AWS Profile and Region:**

```
prowler aws --profile custom-profile -r us-east-1
```

Make sure to replace "custom-profile" with your AWS profile name and adjust the region as needed.

Additional Notes:

- Ensure that your AWS CLI configuration (access key, secret key, default region) is properly set up before running Prowler.

- The commands provided assume a Linux environment. For other operating systems, the installation steps might be slightly different.
- The Prowler commands may vary based on the version of Prowler installed. Check the Prowler documentation for the most accurate and up-to-date information.

Remember to review the Prowler documentation for any additional or updated information: Prowler GitHub