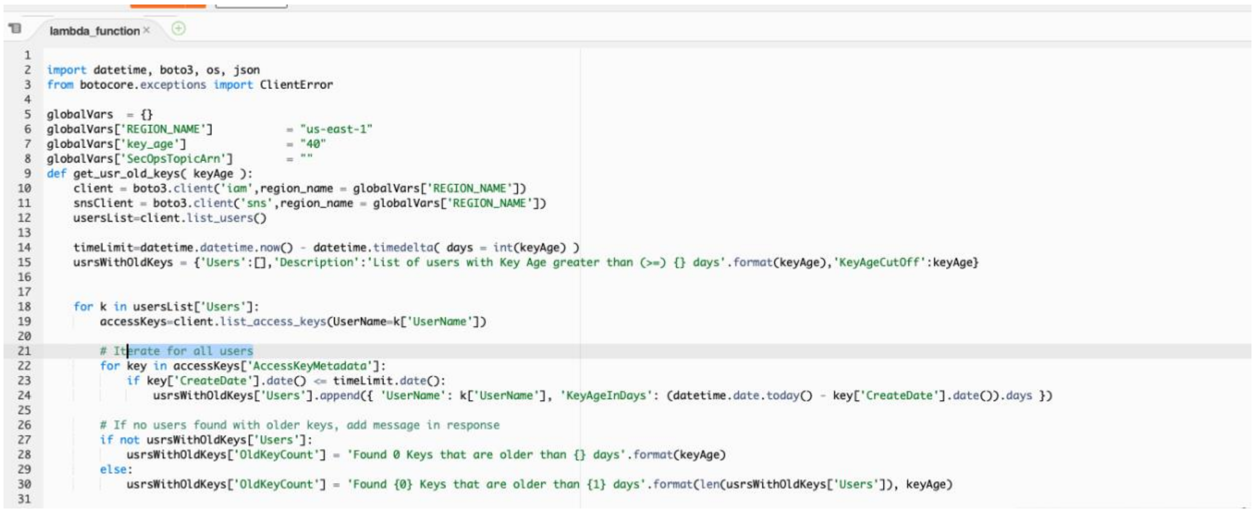
Automatically identify users with old Access/Secret Keys.

If company has multiple IAM user and it is difficult track all users key, to identify old and unused access keys and remove vulnerability and cannot do it manually if many IAM users were there. Using Lambda and CloudWatch which will scan in the account to look for old and unused keys, so that we can rotate and delete the unused keys. Configure SNS to notify SOC team.

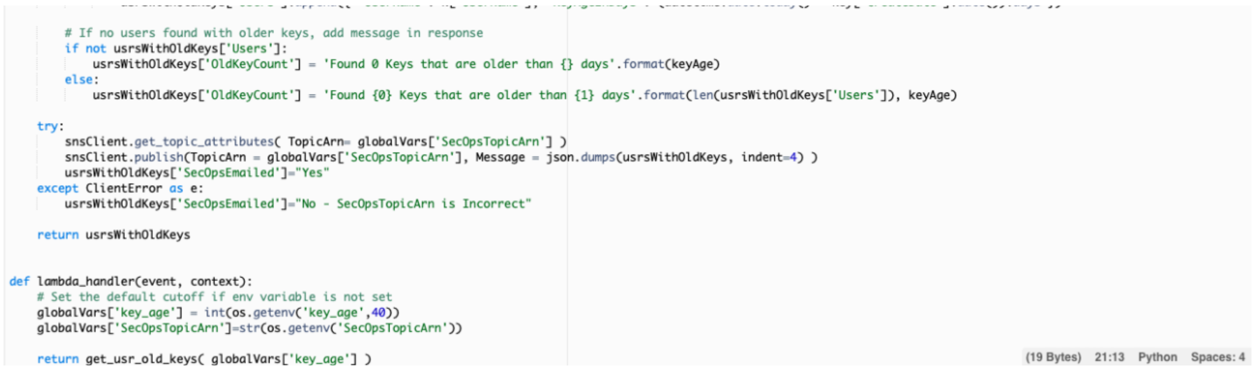
Requirements:

1) IAM ROLE: LAMBDA SERVICE WITH permission IAMReadOnlyAccess and AmazonSNSFullAccess permission.
2) SNS Topic ARN need to create.

Python Code:



```python

import datetime, boto3, os, json
from botocore.exceptions import ClientError

globalVars  = {}
globalVars['REGION_NAME']           = "us-east-1"
globalVars['key_age']               = "40"
globalVars['SecOpsTopicArn']        = ""
def get_usr_old_keys( keyAge ):
    client = boto3.client('iam',region_name = globalVars['REGION_NAME'])
    snsClient = boto3.client('sns',region_name = globalVars['REGION_NAME'])
    usersList=client.list_users()

    timeLimit=datetime.datetime.now() - datetime.timedelta( days = int(keyAge) )
    usrsWithOldKeys = {'Users':[],'Description':'List of users with Key Age greater than (>=) {} days'.format(keyAge),'KeyAgeCutOff':keyAge}


    for k in usersList['Users']:
        accessKeys=client.list_access_keys(UserName=k['UserName'])

        # Iterate for all users
        for key in accessKeys['AccessKeyMetadata']:
            if key['CreateDate'].date() <= timeLimit.date():
                usrsWithOldKeys['Users'].append({ 'UserName': k['UserName'], 'KeyAgeInDays': (datetime.date.today() - key['CreateDate'].date()).days })

        # If no users found with older keys, add message in response
        if not usrsWithOldKeys['Users']:
            usrsWithOldKeys['OldKeyCount'] = 'Found 0 Keys that are older than {} days'.format(keyAge)
        else:
            usrsWithOldKeys['OldKeyCount'] = 'Found {0} Keys that are older than {1} days'.format(len(usrsWithOldKeys['Users']), keyAge)

```



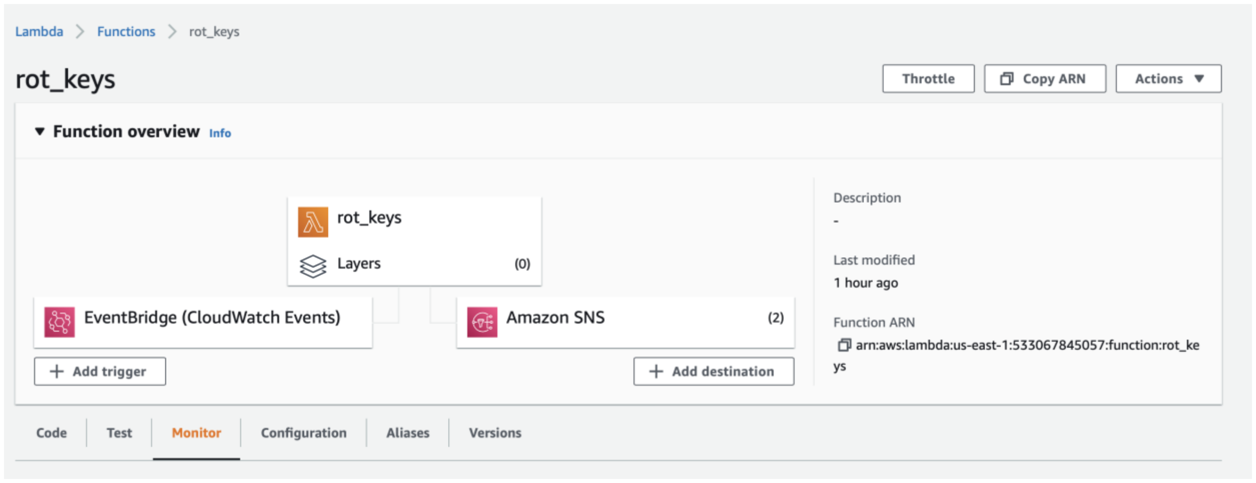
```python
        # If no users found with older keys, add message in response
        if not usrsWithOldKeys['Users']:
            usrsWithOldKeys['OldKeyCount'] = 'Found 0 Keys that are older than {} days'.format(keyAge)
        else:
            usrsWithOldKeys['OldKeyCount'] = 'Found {0} Keys that are older than {1} days'.format(len(usrsWithOldKeys['Users']), keyAge)
```

```python
    try:
        snsClient.get_topic_attributes( TopicArn= globalVars['SecOpsTopicArn'] )
        snsClient.publish(TopicArn = globalVars['SecOpsTopicArn'], Message = json.dumps(usrsWithOldKeys, indent=4) )
        usrsWithOldKeys['SecOpsEmailed']="Yes"
    except ClientError as e:
        usrsWithOldKeys['SecOpsEmailed']="No - SecOpsTopicArn is Incorrect"

    return usrsWithOldKeys


def lambda_handler(event, context):
    # Set the default cutoff if env variable is not set
    globalVars['key_age'] = int(os.getenv('key_age',40))
    globalVars['SecOpsTopicArn']=str(os.getenv('SecOpsTopicArn'))

    return get_usr_old_keys( globalVars['key_age'] )
```

- `globalVars['key_age']` - Set the `key_age` to any value you desire, by default it is set to 40 days
- `globalVars['SecOpsTopicArn']` - Update the ARN of your SNS Topic

  Change it in environment variable and deploy lambda function.

Lambda functions:

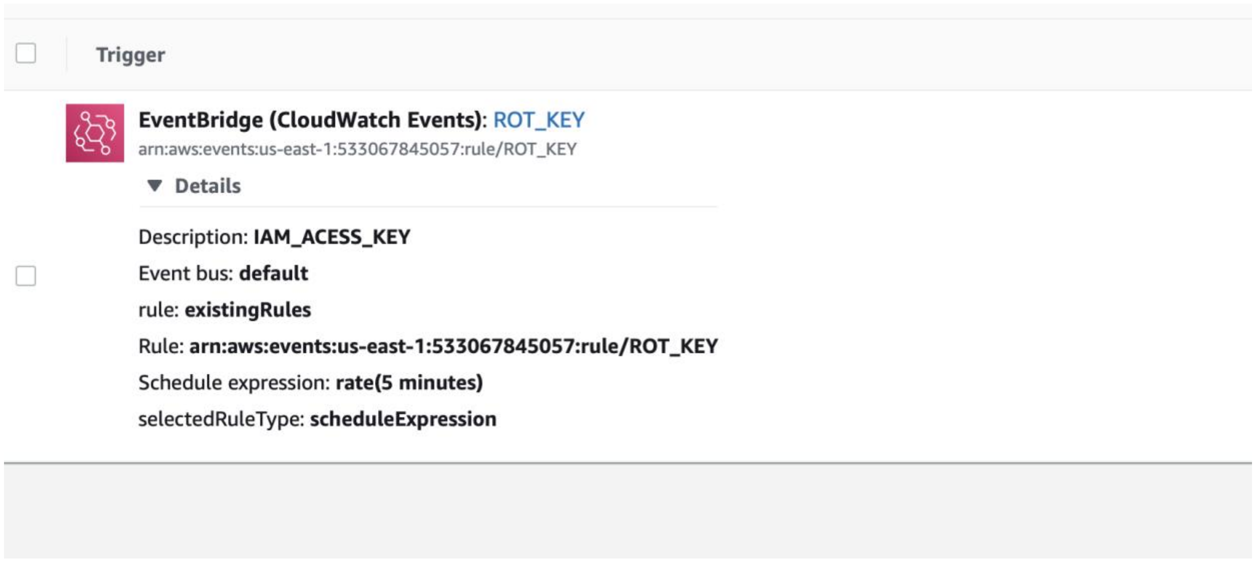


Description
-

Last modified
1 hour ago

Function ARN
arn:aws:lambda:us-east-1:533067845057:function:rot_keys

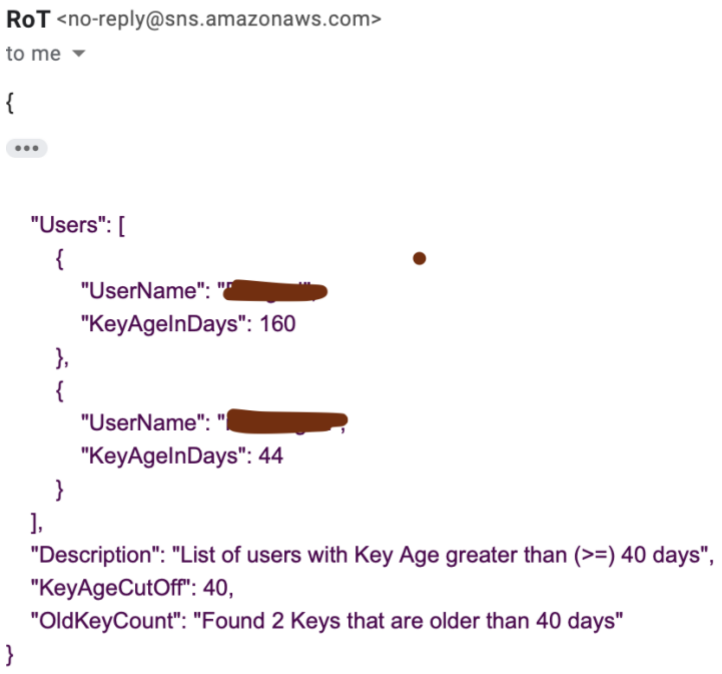
Cloud WATCH events rule: (Trigger)



EventBridge (CloudWatch Events): ROT_KEY
arn:aws:events:us-east-1:533067845057:rule/ROT_KEY

Description: **IAM_ACESS_KEY**
Event bus: **default**
rule: **existingRules**
Rule: **arn:aws:events:us-east-1:533067845057:rule/ROT_KEY**
Schedule expression: **rate(5 minutes)**
selectedRuleType: **scheduleExpression**

SNS notification to email:



RoT <no-reply@sns.amazonaws.com>
to me

```
{
•••

  "Users": [
    {
      "UserName": "[redacted]
      "KeyAgeInDays": 160
    },
    {
      "UserName": "[redacted]
      "KeyAgeInDays": 44
    }
  ],
  "Description": "List of users with Key Age greater than (>=) 40 days",
  "KeyAgeCutOff": 40,
  "OldKeyCount": "Found 2 Keys that are older than 40 days"
}
```

Rotate access keys

After you have identified old keys, You should follow these steps to rotate the keys

1. Create a second access key in addition to the one in use.
2. Update all your applications to use the new access key and validate that the applications are working.
3. Change the state of the previous access key to inactive.
4. Validate that your applications are still working as expected.
5. Delete the inactive access key.