AWS Cloud Security

With Cloud Custodian

Kapil Thangavelu
Overview

- Intro Cloud Custodian
- Cloud Security Strategies
- Enabling Audit Trails
- IAM Tools
- Resource Security
- S3 Access Control
- Network Security
- Encryption at REST
Cloud Custodian

Custodian is an open source rules engine for fleet management in AWS.

YAML DSL for policies based on querying resources or subscribe to events, apply filters, take actions.

Outputs to Amazon S3, Amazon CloudWatch Logs, Amazon CloudWatch Metrics

```yaml
- name: require-rds-encrypt-and-non-public
  resource: rds
  mode:
    - type: cloudtrail
      events:
        - CreateDBInstance
  filters:
    - or:
      - Encrypted: false
      - PubliclyAvailable: true
  actions:
    - type: delete
      skip-snapshot: true
```
Drives Behavior Change

Notifies users in real-time as they do something wrong.

Drives Compliance

Security/Access Control, Encryption, Backups, etc.

Drives Cost Savings

Off-hours, Garbage Collection of unused and underutilized resources.
Policies as Code

Policy as Code

- Versioning (Rollbacks & Audit)
- Peer Review
- Abstraction and encapsulation
- Code Reuse
- Automation

Policy as Code Pipeline

Continuous Integration
- Drone
  - custodian validate mypolicy.yml
  - custodian run --dryrun mypolicy.yml
- LGTM or Github Reviews
  - Multi-person signoff

Continuous Delivery
- git pull & custodian run policy.yml
Cloud Custodian

Resource type policies (ec2 instance, ami, auto scale group, bucket, elb, etc).

Filter resources

Invoke actions on filtered set

Output resource json to s3, metrics to cloudwatch

Vocabularies of actions, and filters for policy construction.

- name: ebs-copy-instance-tags
  resource: ebs
  filters:
    - type: value
      key: "Attachments[0].Device"
      value: not-null
  actions:
    - type: copy-instance-tags
      tags:
        - App
        - Env
        - Owner
        - Name
Multi Step Workflows

“Poorly tagged instances, should be stopped in 1 day, and then terminated in 3”

- mark-for-op

- marked-for-op

Chain together multiple policies.

- name: ec2-tag-compliance-mark
  resource: ec2
  description: |
  Find all non-compliant tag instances for stoppage in 1 days.
  mode:
    type: config-rule
  filters:
    - "tag:maid_status": absent
    - or:
      - "tag:App": absent
      - "tag:Env": absent
      - "tag:Owner": absent
  actions:
    - type: mark-for-op
      op: stop
      days: 1

- name: ec2-tag-compliance-stop
  resource: ec2
  description: |
  Stop poorly tagged and schedule Terminate.
  mode:
    type: periodic
  schedule: rate(1 day)
  filters:
    - type: marked-for-op
      op: stop
    - or:
      - "tag:App": absent
      - "tag:Env": absent
      - "tag:Owner": absent
  actions:
    - stop
    - type: mark-for-op
      op: terminate
      days: 2
Rich Filtering

JMESPath Expressions

Value Type (size, cidr_size, age, expiration, etc)

Operators (>=, <=, list in/not-in, regex, etc)

Arbitrary nesting with and, or, not blocks to combine filters.

Values from file/url/s3.

```json
# This filter is to take out EMR instances
- type: event
  key: "detail.userIdentity.invokedBy.serviceName"
  value: "elasticmapreduce.amazonaws.com"
  op: not-equal

# Find all old non prod instances
- not:
  - "tag:AppEnv": prod
- or:
  - type: instance-age
    days: 30
    op: gte
  - type: image-age
    days: 30
    op: gte

# Find all tag values not in external json file
- type: value
  key: "tag:AccountId"
  op: not-in
  values_from:
    expr: "accounts.*.accountNumber"
    url: s3://mybucket/mykey
```
Policy Execution (lambda, realtime, etc)

Policies can isomorphically be run in various execution modes.

Modes
- Poll
- Cloud Trail
- Periodic
- Config Rule

Custodian will automatically provision event sources and lambda functions.

- name: require-rds-encrypt-and-non-public
  resource: rds
  mode:
    - type: cloudtrail
      events:
        - CreateDBInstance
      filters:
        - or:
          - Encrypted: false
            - PubliclyAvailable: true
      actions:
        - type: delete
          skip-snapshot: true
Interactive Discovery

$ custodian schema ec2
ec2:
  actions: [auto-tag-user, autorecover-alarm, invoke-lambda, mark, mark-for-op, modify-security-groups, normalize-tag, notify, put-metric, remove-tag, rename-tag, resize, set-instance-profile, snapshot, start, stop, tag, tag-trim, terminate, unmark, untag]
  filters: [and, default-vpc, ebs, ephemeral, event, health-event, image, image-age, instance-age, instance-uptime, marked-for-op, metrics, network-location, not, offhour, onhour, or, security-group, singleton, state-age, subnet, tag-count, Value]
Interactive Discovery

$ custodian schema ec2.filters.ebs
Help
----
EC2 instances with EBS backed volume
Filters EC2 instances with EBS backed storage devices (non ephemeral)

:Example:

```yaml
code-block:: yaml
policies:
  - name: ec2-encrypted-ebs-volumes
    resource: ec2
    filters:
      - type: ebs
        key: encrypted
        value: true
```

Schema
------
Custodian Capabilities

Imagination is the limit

- Encryption
- Backups
- Garbage Collection
- Unused
- Offhours
- Tag compliance
- SG compliance

109 resources
524 unique actions
376 unique filters
Aws Security w/ Custodian
Cloud Security Strategies

Small Accounts

- Billing Transparency (+Budgets API)
- Blast Radius Protection
- Reduce Data Visibility
- Simpler IAM Configuration
- Dx/Ux usability
- Rate limit transparency

IAM

- Focus Least privilege
- Cross Account
- Think like a red team around each api.

Cloud Controls as either
Preventive - Access Control (IAM)
Detective - Discovery and remediation post creation.
Security Foundations - Audit Trails

- Cloud Trail
- AWS Config
- VPC Flow Logs
- S3 Data Events
- ELB/ALB Access Logs
Ensuring Audit Trail - Cloud Trail

Cloud Trail - Control Plane API calls for services.

```yaml
policies:
  - name: aws-cloudtrail-not-enabled
    resource: account
    region: us-east-1
    tags:
      - level: high
    description: |
      Scans for accounts which do not have CloudTrails enabled.
    filters:
      - type: check-cloudtrail
        global-events: true
        multi-region: true
        running: true
        kms: true
        file-digest: true
```
Ensuring Audit Trail - Cloud Config

Config provides for periodic polling and archiving of current and historical revisions of resources (that it supports).

```yaml
policies:
  - name: aws-config-not-enabled
    resource: account
    region: us-east-1
    tags:
      - level: high
    description: |
      Policy scans for accounts which do not have CloudTrails enabled.

filters:
  - type: check-config
    all-resources: true
    global-resources: true
    running: true
```
Ensuring Audit Trail - S3 Data Events

CloudTrail for Data Plane events on S3 (Read, Writes)

```yaml
policies:
- name: s3-audit
  resource: account
  actions:
    - type: enable-data-events
      data-trail:
        name: s3-trail
        # collect logs on all regions bucket to this region.
        multi-region: us-east-1
        create: true
        s3-bucket: my-s3-logs
        s3-prefix: S3Logs
```
Ensuring Audit Trail - ALB/ELB/Cloudfront Logs

Capture access logs from all albs

policies:
- name: appelb
  resource: vpc
  filters:
    - type: value
      key: Attributes."access_logs.s3.enabled"
      value: False
  actions:
    - type: set-s3-logging
      bucket: my-lb-logs
      prefix: "Logs/{AccountId}/{LoadBalancerName}"
Ensuring Audit Trail - VPC Flow Logs

Flow logs give basic metadata on connections between ips within a vpc.

```
policies:
  - name: flow-log-not-enabled
    resource: vpc
    filters:
      - type: flow-log
        state: absent
    actions:
      - <<: *notify_action
        subject: "Missing VPC Flow Logs"
```
tools/c7n_logexporter - Log Exporter

Export cloudwatch log at scale on historical basis across many accounts.

$ c7n-log-exporter
Usage: c7n-log-exporter [OPTIONS] COMMAND [ARGS]...

c7n cloudwatch log group exporter

Options:
--help  Show this message and exit.

Commands:
 access Check iam permissions for log export access...
 export export a given log group to s3
 run run export across accounts and log groups...
 size size of exported records for a given day.
 status report current export state status
 sync sync last recorded export to actual Use...
 validate validate config file
IAM

Principles

- Least Privilege

- User Credential Rotation (Use STS/Instance Roles wherever possible)

- User MFA

Starting Reference on AWS Policies

AWS CIS Benchmark

Other opensource tools

Netflix (Aardvark & RepoKid)

https://github.com/Netflix-Skunkworks/aardvark

Least Privilege IAM Roles via AccessAdvisor

Zalando’s FullStop

https://github.com/zalando-stups/fullstop

CloudTrail Analytics
IAM

MFA

- name: user-missing-mfa
  resource: iam-user
  region: us-east-1
  filters:
    - type: credential
      key: password_enabled
    - type: mfa-device
      value: empty
  actions:
    - <<<: *notify_action
      subject: "Enable MFA on your AWS user"

Password Policy

- name: iam-password-policy
  resource: account
  region: us-east-1
  filters:
    - or:
      - type: password-policy
        key: MaxPasswordAge
        value: 30
        op: gte
      - type: password-policy
        key: MinimumPasswordLength
        value: 8
        op: lte
Detect Root Logins

- name: root-login-detected
  resource: account
  description: |
      Notifies on root user logins
  mode:
    type: cloudtrail
    role: arn:aws:iam::{account_id}:role/Eg
    events:
      - ConsoleLogin
  filters:
    - type: event
      key: "detail.userIdentity.type"
      value: Root
  actions:
    - <<: *notify_action
      subject: "AWS Root Console Login Alert"

Detect Lack of Root Hardware MFA

- name: root-virtual-mfa
  resource: account
  description: |
      Notifies on root user without hw mfa
  filters:
    - type: iam-summary
      key: AccountMFAEnabled
      value: true
    - type: has-virtual-mfa
      value: true
  actions:
    - <<: *notify_action
      subject: "AWS Root Missing MFA"
Resources w/ Embedded IAM

- SNS
- SQS
- Glacier
- Lambda
- ECR
- KMS
- S3*
Resources w/ Embedded IAM

ECR

vars:
  accounts_url: &accounts_url s3://mybucket/myaccounts.json

policies:
  - name: ecr-cross-account
    resource: ecr
    filters:
      - type: cross-account
        whitelist_from:
          expr: "accounts.*.accountNumber"
          url: *accounts_url
    actions:
      - <<- *notify_action
        subject: "[custodian] ECR available externally"
      - type: remove-statements
        statement_ids: matched

Lambda

vars:
  accounts_url: &accounts_url s3://mybucket/myaccounts.json

policies:
  - name: lambda-cross-account
    resource: lambda
    filters:
      - type: cross-account
        whitelist_from:
          expr: "accounts.*.accountNumber"
          url: *accounts_url
    actions:
      - <<- *notify_action
        subject: "[custodian] Lambda available externally"
      - type: remove-statements
        statement_ids: matched
S3 Access Control

- Bucket & Object ACL
- Bucket Policy
- Logging
- PreSigned URL
- VPC Endpoint Restrictions
S3 - Global ACL

Legacy Primitive

Global external access

Limit or Yank Object ACL from IAM

- name: s3-global-acl
  resource: s3
  region: us-east-1
  filters:
    - type: global_grants
      allow_website: true
  actions:
    - <=: *notify_action
      subject: "s3 buckets shared externally"
    - delete-global-grants
S3 Bucket Policy

Allows fine grain grants of access with a number of conditions.

Ie. Grant access to a user within a folder.

```json
vars:
  accounts_url: &aurl s3://mybucket/myaccounts.json

policies:
- name: s3-cross-account
  resource: s3
  region: us-east-1
  filters:
    - type: cross-account
      whitelist_from:
        expr: "accounts.*.accountNumber"
        url: *aurl
  actions:
    - <<: *notify_action
      subject: "s3 buckets available externally"
    - type: remove-statements
      statement_ids: matched
```
S3 PreSigned URL

Application credentials used to sign a url.

ATM, only KMS SSE can stop.

Exists within Azure and GCP Storage as well.

```python
$ python
Python 2.7.13 (default, Apr 4 2017, 08:47:57)
[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.38)]
on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import boto3
>>> client = boto3.client('s3')
>>> url = client.generate_presigned_url('head_object',
{'Bucket': 'custodian-log-archive', 'Key':
'build/example.tgz'}, ExpiresIn=3600*14)
>>> print(url)
https://custodian-log-archive.s3.amazonaws.com/build/example.tgz?AWSAccessKeyId=AKIAIFX6HPKAM5F7JVAA&Expires=1507690282&Signature=22jxjAzJdNUjY4xax9%2F%2FAhMLn50%3D
>>> 
```
Network Security - Public Endpoints

EC2 with role behind ELB

policies:
- name: public-elb-exposed-instance-roles
  resource: elb
  filters:
    - Scheme: internet-facing
    - type: instance
      key: "IamInstanceProfile.Arn"
      value: present

EC2 Instance directly on internet

policies:
- name: public-elb-exposed-instance-roles
  resource: ec2
  filters:
    - PublicDnsName: not-null
Network Security

Security Group

policies:
- name: odd-groups
  resource: security-group
  filters:
  - "tag:Location": "Public"
  - type: ingress
    OnlyPorts: [80, 443, 22]
  actions:
  - remove-permissions

Security group 2

policies:
- name: odd-groups-2
  resource: security-group
  filters:
  - type: egress
    Cidr:
      op: in
      value_type: cidr
      value: "10.0.0.0/8"
  actions:
  - remove-permissions
Network Security - DDOS

CloudFront

- name: activate-cloudfront-shield
  resource: distribution
  region: us-east-1
  filters:
    - type: shield-enabled
      state: false
  actions:
    - type: set-shield
      state: true

ELB

- name: activate-elb-shield
  resource: elb
  filters:
    - type: value
      key: Scheme
      value: internal
      op: not-equal
    - type: shield-enabled
      state: false
  actions:
    - type: set-shield
      state: true
Network Security - WAF

CloudFront

```
policies:
  - name: activate-cloudfront-waf
    resource: distribution
    region: us-east-1
    filters:
      - Status: Deployed
      - WebACLId: empty
    actions:
      - type: set-waf
        web-acl: global-default
        state: true
```

Application Load Balancer

```
policies:
  - name: activate-alb-waf
    resource: app-elb
    filters:
      - type: value
        key: Scheme
        value: internal
        op: not-equal
      - type: waf-enabled
        state: false
    actions:
      - type: set-waf
        web-acl: regional-default
        state: true
```
Network Control - Lambda

Lambdas by default have unfettered internet access and potentially roles that allow them access to

```
policies:
  - name: lambda-vpc-attachment
    resource: lambda
    filters:
      - VpcConfig: absent
    actions:
      - <<- *notify_action
        subject: "[custodian] Lambda non vpc attached"
      - delete
```
Encryption at REST

- Protects against very specific threat models
- Provides ancillary protection around accidental snapshot sharing (image, redshift, ebs, rds).

```yaml
policies:
- name: asg-unencrypted
  resource: asg
  filters:
    - type: not-encrypted
      exclude_images: true

- name: kinesis-unencrypted
  resource: kinesis
  filters:
    - KeyId: absent

- name: ec2-encrypted
  resource: ec2
  filters:
    - type: ebs
      key: Encrypted
      value: false
      skip-devices: ["dev/xvda", "/dev/sda1"]
```
tools/c7n_org - Multi Account Execution

Run and report custodian policies across multiple accounts.

Bonus run arbitrary scripts

```bash
$ c7n-org run-script -s ~/waf-deploy \
   -c accounts/custodian.yml -r us-east-1 \n   -t "partition:us" \n   -t "type:dev" \n   aws cloudformation deploy \n       --stack-name default-global-waf \n       --template-file=deploy/waf-default.yml
```

```bash
$ c7n-org
Usage: c7n-org [OPTIONS] COMMAND [ARGS]...

custodian organization multi-account runner.

Options:
--help  Show this message and exit.

Commands:
report  report on a cross account policy execution.
run     run a custodian policy across accounts
run-script  run an aws script across accounts
```
tools/c7n_traildb - Cloud Trail Indexing

Question - Who’s abusing the api?

c7n-traildb - Load Cloudtrail hourly on daily segment per account/region into s3 sqlite files (namesake adroll traildb).

c7n-trailts - Ingest time series data on aggregate queries into influxdb.

c7n-trailes - Ingest traildb into elasticsearch

$ c7n-traildb --help
usage: c7n-traildb [-h] --bucket BUCKET [--prefix PREFIX] --account ACCOUNT
         [--user USER] [--event EVENT] [--source SOURCE]
         [--not-source NOT_SOURCE] [--day DAY] [--month MONTH]
         [--tmpdir TMPDIR] [--region REGION] [--output OUTPUT]
         [--profile PROFILE] [--assume ASSUME_ROLE]
         [--field {userIdentity,responseParameters,responseElements}]

optional arguments:
  -h, --help            show this help message and exit
  --bucket BUCKET
  --prefix PREFIX
  --account ACCOUNT
  --user USER
  --event EVENT
  --source SOURCE
  --not-source NOT_SOURCE
  --day DAY
  --month MONTH
  --tmpdir TMPDIR
  --region REGION
  --output OUTPUT
  --profile PROFILE
  --assume ASSUME_ROLE
  --field {userIdentity,responseParameters,responseElements}
tools/c7n_salactus - Scale out S3 Scanning

Goal - Examine every object in s3 across many accounts.

Stateless asg working against a redis queue.

Custom visitors (acl, encryption ootb)

Support for S3 Inventory

$ c7n-salactus --help
   Salactus, eater of s3 buckets

Commands:

- accounts: Report on stats by account
- buckets: Report on stats by bucket
- failures: Show any unexpected failures
- inspect-bucket: Show all information on a bucket
- inspect-partitions: Discover the partitions on a bucket
- inspect-queue: Show contents of a queue
- queues: Report on progress by queues
- reset: Delete all persistent cluster state
- run: Run on a set of accounts and buckets
- save: Save the current state to a json file
- validate: Validate a configuration file
- watch: Watch scan rates across the cluster
- workers: Show information on salactus workers
Resource metrics indexing

Let’s visualize/alert on resource metrics across accounts.

```
- name: ingress-cloudfront
  resource: distribution
  filters:
    - <<: *metrics_query
      name: Requests
    - <<: *metrics_query
      name: BytesDownloaded
    - <<: *metrics_query
      name: BytesUploaded
    - <<: *metrics_query
      name: TotalErrorRate
    - <<: *metrics_query
      name: TotalErrorRate
  statistics: Average
  actions:
    - <<: *metrics_index
  metrics:
    - Requests
    - BytesDownloaded
    - TotalErrorRate
    - BytesUploaded
```
Roadmap

Short term

- Better Documentation :-)  
- Hackathon/Sprint @ Reinvent (Tuesday)  
- Differing resource changes  
- Additional Cloud Providers
Community

Capital One Opensource - http://developer.capitalone.com

Gitter https://gitter.im/capitalone/cloud-custodian

Github https://github.com/capitalone/cloud-custodian

Mailing List - https://groups.google.com/forum/#!forum/cloud-custodian
Any Questions