AWS Incident Response Cheat Sheet

About Cado Security

Cado Security is the provider of the first cloud forensics and incident response platform. The platform leverages the scale and speed of the cloud to automate the end-to-end incident response process – from data capture and processing to investigation and response.

Introduction

With the rapid migration to the cloud, it's becoming increasingly difficult to keep track of all of the different data sources, commands, and tools available from each Cloud Service Provider (CSP). This cheat sheet was designed to provide security professionals with an overview of key best practices, data sources and tools that they can have at their disposal when responding to an incident in an AWS environment.

Cloud Security Incident Domains

There are three domains in an AWS environment that fall under the customers' responsibility to secure:

Service domain incidents impact AWS accounts, IAM permissions, resource metadata, billing, and other areas. They are addressed using AWS API mechanisms and can be caused by configuration or resource permission issues.

Infrastructure domain incidents involve data or network-related activities within Amazon EC2 instances, such as processes, data, and VPC traffic. Responding to these incidents requires the ability to acquire data for forensic analysis, interact with the instance's operating system, and utilize AWS APIs. Forensic analysis and investigations can be conducted using AWS APIs and DFIR tools within a dedicated EC2 instance.

Application domain incidents occur in application code or software deployed to services or infrastructure. They should be addressed in cloud threat detection and response playbooks and can be managed using cloud tools with automated acquisition, recovery, and deployment. Similar response strategies as the infrastructure domain may be applicable.

Useful Open Source Tools

AWS_IR is a Python CLI tool used to automate initial response actions

Margarita Shotgun is used to dump memory from systems

SANS Investigative Forensic Toolkit is an all-in-one forensic toolkit

Diffy is a tool for identifying changes or differences in systems

Automatic Forensics Orchestrator collects full snapshots of EC2 systems

OSQuery is an endpoint detection and response tool

SOF-ELK is an analytics platform focused on the needs of computer forensics and investigation teams

Prowler is a multi-purpose toolkit

kube-forensics is used to dump the running pod and all its containers

Invictus-AWS automatically enumerates and acquires relevant data

Useful List Commands

List AWS regions
aws ec2 describe-regions

List information about Cloudwatch alarms
aws cloudwatch describe-alarms | jq -r '.MetricAlarms[ ] | .AlarmName+" "+.Namespace+"

List EC2 instances with ID, Type, Name
aws ec2 describe-instances | jq -r '.Reservations[].Instances[]|.InstanceId+" "+.InstanceType+" "+(.Tags[] | select(.Key == "Name").Value)'

List security groups
aws ec2 describe-security-groups | jq -r '.SecurityGroups[]|.GroupId+" "+.GroupName'

List Subnets for a VPC
aws ec2 describe-subnets --filter Name=vpc-id,Values=<Your_VPC_ID> | jq -r '.Subnets[]|.SubnetId+" "+.CidrBlock+" "+(.Tags[]|select(.Key=="Name").Value)'

Lists the Logs that are available in a region
aws logs describe-log-groups --region <region>

Important Log Types

CloudTrail: Tenant audit logs

CloudTrail Insights: API usage outside of baselines

CloudWatch Logs: Forwarded logs from applications and endpoints

GuardDuty: Anomaly detection within CloudTrail

VPC flow logs: NetFlow logs from your VPCs

S3 Server access: Logs from web-based storage access

Route 53: DNS Resolver Logs

Load Balancer Logs: Logs requests sent to your Load Balancer

Accessing Logs

Access logs directly via AWS web console

Store and search logs via S3 and Athena

Access logs via real-time API call via event hub

Exporting Logs

Via web console: This is very limited. Log files can only be downloaded one at a time which is a slow process. Further, sorting of files is unavailable once there are more than 999.

Via CLI: Export logs using the aws cp and aws sync commands.

- **aws cp command:** aws s3 cp s3:// <log_bucket_here>/AWSLogs . --recursive
- the aws cp command will copy the logs to the current working directory, including the contents of subfolders. To copy to another location replace the '.' with the desired location.
- **aws sync command:** aws s3 sync s3:// <log_bucket_here>/AWSLogs . --recursive
- aws sync commands work exactly the same as aws cp commands, but if there is already a copy of the logs at the destination, it will only download new log events and update the log file.

AWS Native Tools

Amazon Detective
Log collection and analytics powered by machine learning

Amazon CloudWatch
Visualization of real-time logs in automated dashboards

AWS Security Hub
A centralized hub for monitoring security alerts

AWS SSM
Allows the execution of tools to gather forensic data and take actions on compromised systems

AWS Snapshots

Snapshots in AWS are a crucial feature when it comes to digital forensics. They are point-in-time copies of your Amazon instances. These snapshots serve as backups and can be used for various forensic investigations.

Create a snapshot:

```
aws ec2 create-snapshot
--volume-id <volume_id>
--description “Snapshot
created”
aws ec2 create-volume
--availability-zone <DFIR_zone>
--snapshot-id <snapshot_id>
```

Download Snapshot (Coldsnap):

```
coldsnap --region <region>
download <snapshot_id>
image.dd
```

Mount a Snapshot:

```
aws ec2 attach-volume
--volume-id <volume_id>
--instance-id <DFIR_instance>
--device </dev/sdX>
```

Cloudwatch Logs

Cloudwatch automatically collects logs and allows you to query them in real time. Go to the cloudwatch console > select insight > logs > then choose your log groups and set your time constraints. Use the following queries to quickly identify suspicious activity:

IAM Logs

List all IAM access denied attempts

```
filter errorCode like
/Unauthorized|Denied|Forbidd
en/ | fields awsRegion,
userIdentity.arn, eventSource,
eventName, sourceIPAddress,
userAgent
```

List the actions an access key has performed

```
filter userIdentity.accessKeyId
=”<Access_Key>” | fields
awsRegion, eventSource,
eventName, sourceIPAddress,
userAgent
```

List IAM actions performed by a specified IP address

```
filter sourceIPAddress =
“192.0.2.1” | fields awsRegion,
userIdentity.arn, eventSource,
eventName, sourceIPAddress,
userAgent
```

List all IAM user and role creation events

```
filter eventName=”CreateUser”
or eventName = “CreateRole” |
fields
requestParameters.userName,
requestParameters.roleName,
responseElements.user.arn,
responseElements.role.arn,
sourceIPAddress, eventTime,
errorCode
```

List all ‘ListBucket’ events (this can reveal if an attacker is trying to access your buckets)

```
filter eventName =”ListBuckets”
| fields awsRegion, eventSource,
eventName, sourceIPAddress,
userAgent
```

VPC FLow Logs

List reject requests by IP address

```
filter action=”REJECT” | stats
count(*) as numRejections by
srcAddr | sort numRejections
desc
```

List reject requests originating from inside your VPC

```
filter action=”REJECT” and
srcAddr like /^10\./ | stats
count(*) as numRejections by
srcAddr | sort numRejections
desc
```

List requests originating from a specific ip address

```
filter srcAddr = “192.0.2.1” |
fields @timestamp, interfaceId,
dstAddr, dstPort, action
```

List outgoing requests from a specific IP address by number of requests

```
filter srcAddr = “10.1.1.1” | stats
count(*) as numConnections by
dstAddr | sort numConnections
desc
```

IAM

Use these commands to quickly view all roles, users, and groups:

list all roles in JSON format
aws iam list-roles

list all users
aws iam list-users --output table --query 'Users[*].UserName'

list all groups
aws iam list-groups --output table --query 'Groups[*].GroupName'

Use these commands to quickly block and unblock compromised roles, users, groups, or access keys:

block role
aws iam put-role-policy --role-name <ROLE> --policy-name DenyAll --policy-document '{ "Statement": [{ "Effect": "Deny", "Action": "*", "Resource": "*" }] }

remove block
aws iam delete-role-policy --role-name <ROLE> --policy-name DenyAll

block user
aws iam put-user-policy --user-name <USER> --policy-name DenyAll --policy-document '{ "Statement": [{ "Effect": "Deny", "Action": "*", "Resource": "*" }] }

remove block
aws iam delete-user-policy --user-name <USER> --policy-name DenyAll

block group
aws iam put-group-policy --group-name <GROUP> --policy-name DenyAll --policy-document '{ "Statement": [{ "Effect": "Deny", "Action": "*", "Resource": "*" }] }

remove block
aws iam delete-group-policy --group-name <GROUP> --policy-name DenyAll

disable an access key
aws iam update-access-key --access-key-id <Access_Key> --status Inactive --user-name user

Responding to EC2 Incidents

Step 1: Prevent data theft by changing the security group to one that disallows outbound traffic

Step 2: Check if there was an Instance Profile attached. CloudTrail logs can be used to see if it was abused to access other resources.

Step 3: Snapshot the Instance to allow for forensic investigation.

Responding to EKS Incidents

Identify if the compromised system is running on an EC2 instance or fargate.

For EC2: Follow the steps outlined in the "Responding to EC2 Incidents" section above.

For Fargate: In this case, it is important to collect all forensic data before the system is shutdown.

Responding to Lambda Incidents

These can be particularly frustrating to investigate as the underlying container can not be accessed. You can however collect:

- ➔ The code of the Lambda function
- ➔ Any environment variables it is set to use
- ➔ Any logs in CloudWatch and CloudTrail

If you'd like to learn more about what Cado Security is doing to help advance investigations and incident response in the cloud, request a demo today.

Further reading and resources:

CloudTail cheat sheet by invictus

SANS posters and cheat sheets

AWS CloudTrail Searching Event logs in S3, Athena and Cloudwatch

Digital Forensic Analysis of Amazon Linux EC2 Instances

AWS ECS: Fully Managed but Frustrating to Investigate

AWS, IAM Your Father (Part II - Defensive)

Logging and monitoring in Amazon S3