Ultimate Guide to Incident Response **in AWS**

Table of Contents

Introduction

Investigating a security compromise in AWS can be a dizzying prospect. There are (as of writing) **over 200** services in AWS. To make matters worse, different services log in different formats to different locations. Some will write to CloudTrail, some to CloudWatch. Others display logs directly in a custom console, or an S3 bucket.

Below, we've tried to cover the AWS services you are most likely to encounter during security incidents in AWS. We've also included pointers on where to go for more information on investigating and recovering from incidents in the various services.

EC2 Incident Response

If you've identified a potentially compromised EC2 instance, there are a number of immediate actions you can take:

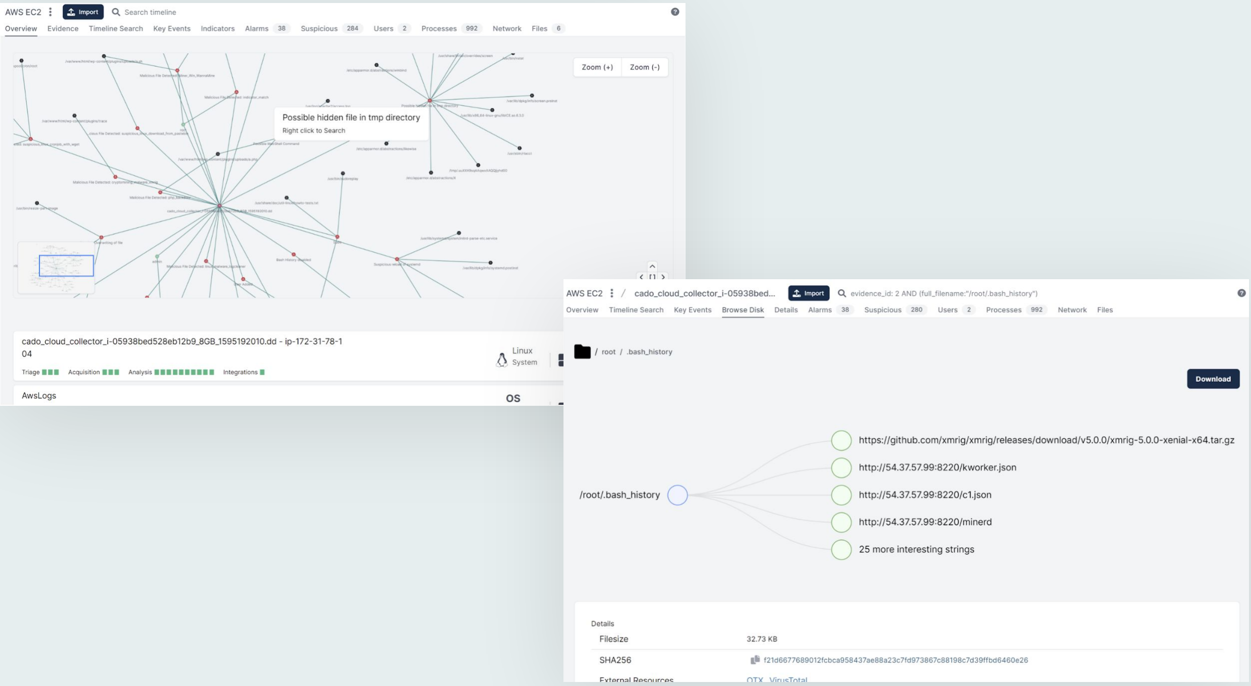
- To limit the possibility of data theft, change the security group to one that doesn't allow any outbound internet access.
- Identify if there was an Instance Profile attached to the EC2. If there was, check CloudTrail logs to see if it may have been abused to access other resources in AWS.
- Take a snapshot of the EC2, to enable forensic analysis later on.

Community Resources

SANS has published a Whitepaper titled "[Digital Forensic Analysis of Amazon Linux EC2 Instances](#)". A number of tools were released at [Blackhat 2016](#) for AWS. Whilst a little dated now, there are useful tools in the ThreatResponse [Github repository](#) for preserving forensic artifacts from EC2 instances, as well as isolating them and associated IAM credentials.

Cado Security Resources

We've published a video tutorial on how to investigate a compromised EC2 Instance on [YouTube](#). You can use [Cado Response](#) to import potentially compromised EC2 systems in a single click for investigation. However, if you've set up an API to drive an automated response framework, you can automatically capture data immediately following detection to reduce the Mean Time to Respond (MTTR).



cado_cloud_collector_i-05938bed528eb12b9_8GB_1595192010.dd - ip-172-31-78-1
04

Triage ▮▮▮ Acquisition ▮▮▮ Analysis ▮▮▮▮▮▮▮▮▮▮ Integrations ▮

AwsLogs

evidence_id: 2 AND (full_filename:"/root/.bash_history")

/ root / .bash_history

/root/.bash_history

- https://github.com/xmrig/xmrig/releases/download/v5.0.0/xmrig-5.0.0-xenial-x64.tar.gz
- http://54.37.57.99:8220/kworker.json
- http://54.37.57.99:8220/c1.json
- http://54.37.57.99:8220/minerd
- 25 more interesting strings

Details

Filesize	32.73 KB
SHA256	f21d6677689012fcbca958437ae88a23c7fd973867c88198c7d39ffbd6460e26
External Resources	OTX VirusTotal

Example analysis of a compromised AWS EC2 System in Cado Response

Official AWS Resources

AWS provides a number of experimental solutions to help isolate, preserve and analyze compromised EC2 systems. A few key ones to play with include:

- "Solution for AWS Cloud for Incident Response in EC2 instances"
  This is a CloudFormation deployment to quarantine EC2 systems via SSM commands on the host themselves, perform security group changes, and snapshot EBS volumes.

- "Automated Incident Response with SSM"
  Another solution that uses SSM that can also quarantine EC2 systems, but is based on the outcome of GuardDuty events.

- "Automated Incident Response and Forensics Framework"
  A set of Security Hub actions to acquire data from EC2 systems.

- "Automated Forensics Orchestrator for Amazon EC2"
  A more recent CloudFormation deployment to acquire data from EC2 systems then points you to the free SANS SIFT Linux distribution for command line analysis at the raw disk level.

- "EC2 Auto Clean Room Forensics"
  A CloudFormation deployment that will run the open-source fls tool to dump file timestamps from files found on a compromised EC2 system.

EKS Incident Response

If you’ve identified a potentially compromised container in EKS, there are two potential ways forward:

- If the container is running on an underlying EC2, then refer to the suggested steps above for immediate actions.
- If the container is running on Fargate, then collect any data required for later analysis **before** subsequently suspending it.

Community Resources

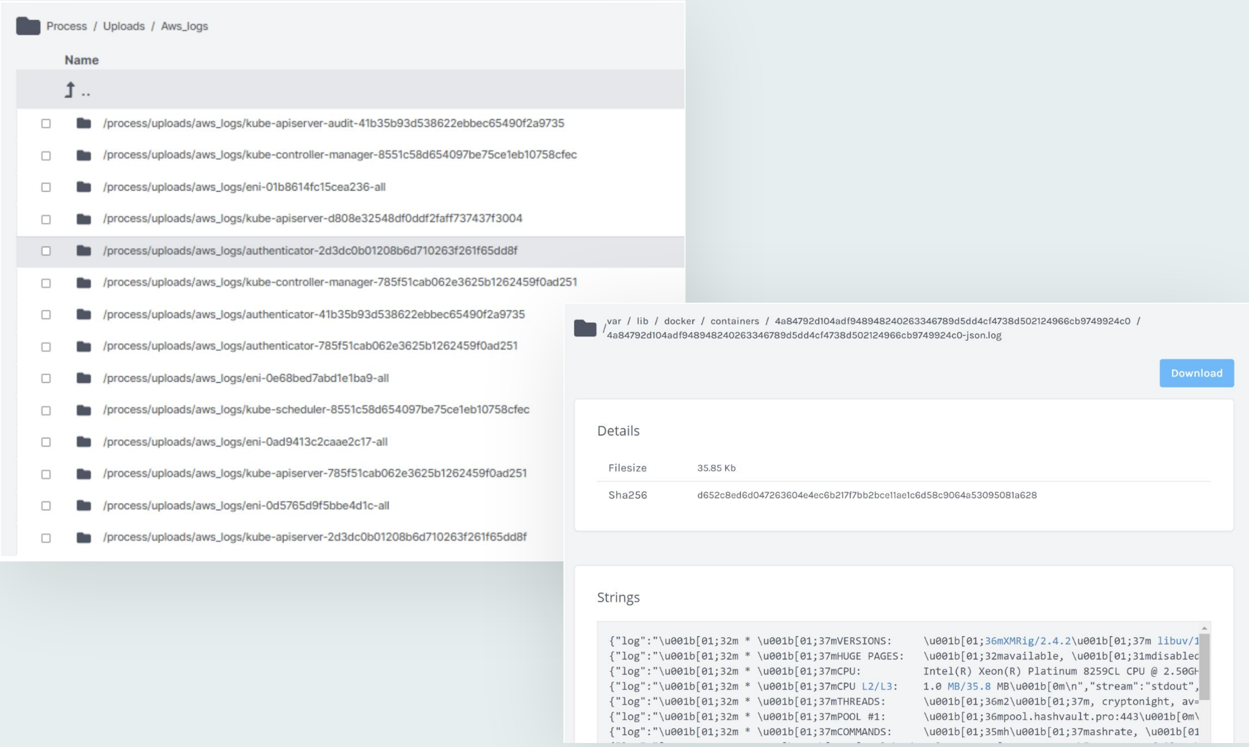
kube-forensics allows a cluster administrator to dump the current state of a running pod and all its containers so that security professionals can perform off-line forensic analysis.

Cado Security Resources

We previously published a playbook dedicated to investigating compromises in EKS environments. Check out the GitHub repository with sample data taken from a compromised EKS system, and an associated talk on how to analyze it.

Cado Response can analyze compromised EKS systems by collecting:

- Logs in S3 from the AWS side (e.g. kubernetes authentication).
- A full copy of a container from within the container itself.
- A full copy of the underlying EC2 that the container is running on, which also includes a number of logs from the host operating system as well as versioned files from the container due to how the overlay2 filesystem works.



Process / Uploads / Aws_logs

Name

↥ ..

- □ /process/uploads/aws_logs/kube-apiserver-audit-41b35b93d538622ebbec65490f2a9735
- □ /process/uploads/aws_logs/kube-controller-manager-8551c58d654097be75ce1eb10758cfec
- □ /process/uploads/aws_logs/eni-01b8614fc15cea236-all
- □ /process/uploads/aws_logs/kube-apiserver-d808e32548df0ddf2faff737437f3004
- □ /process/uploads/aws_logs/authenticator-2d3dc0b01208b6d710263f261f65dd8f
- □ /process/uploads/aws_logs/kube-controller-manager-785f51cab062e3625b1262459f0ad251
- □ /process/uploads/aws_logs/authenticator-41b35b93d538622ebbec65490f2a9735
- □ /process/uploads/aws_logs/authenticator-785f51cab062e3625b1262459f0ad251
- □ /process/uploads/aws_logs/eni-0e68bed7abd1e1ba9-all
- □ /process/uploads/aws_logs/kube-scheduler-8551c58d654097be75ce1eb10758cfec
- □ /process/uploads/aws_logs/eni-0ad9413c2caae2c17-all
- □ /process/uploads/aws_logs/kube-apiserver-785f51cab062e3625b1262459f0ad251
- □ /process/uploads/aws_logs/eni-0d5765d9f5bbe4d1c-all
- □ /process/uploads/aws_logs/kube-apiserver-2d3dc0b01208b6d710263f261f65dd8f

/ var / lib / docker / containers / 4a84792d104adf948948240263346789d5dd4cf4738d502124966cb9749924c0 / 4a84792d104adf948948240263346789d5dd4cf4738d502124966cb9749924c0-json.log

Details

Filesize	35.85 Kb
Sha256	d652c8ed6d047263604e4ec6b217f7bb2bce11ae1c6d58c9064a53095081a628

Strings

```
{"log":"\u001b[01;32m * \u001b[01;37mVERSIONS:     \u001b[01;36mXMRig/2.4.2\u001b[01;37m libuv/1
{"log":"\u001b[01;32m * \u001b[01;37mHUGE PAGES:   \u001b[01;32mavailable, \u001b[01;31mdisablec
{"log":"\u001b[01;32m * \u001b[01;37mCPU:          Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GH
{"log":"\u001b[01;32m * \u001b[01;37mCPU L2/L3:    1.0 MB/35.8 MB\u001b[0m\n","stream":"stdout",
{"log":"\u001b[01;32m * \u001b[01;37mTHREADS:      \u001b[01;36m2\u001b[01;37m, cryptonight, av=
{"log":"\u001b[01;32m * \u001b[01;37mPOOL #1:      \u001b[01;36mpool.hashvault.pro:443\u001b[0m\
{"log":"\u001b[01;32m * \u001b[01;37mCOMMANDS:     \u001b[01;35mh\u001b[01;37mashrate, \u001b[01
```

Example analysis of a compromised AWS EKS System in Cado Response

Official AWS Resources

AWS provides advice on incident response and forensics in their EKS Best Practices documentation on Github. They also recently released new GuardDuty detections for EKS.

ECS Fargate Incident Response

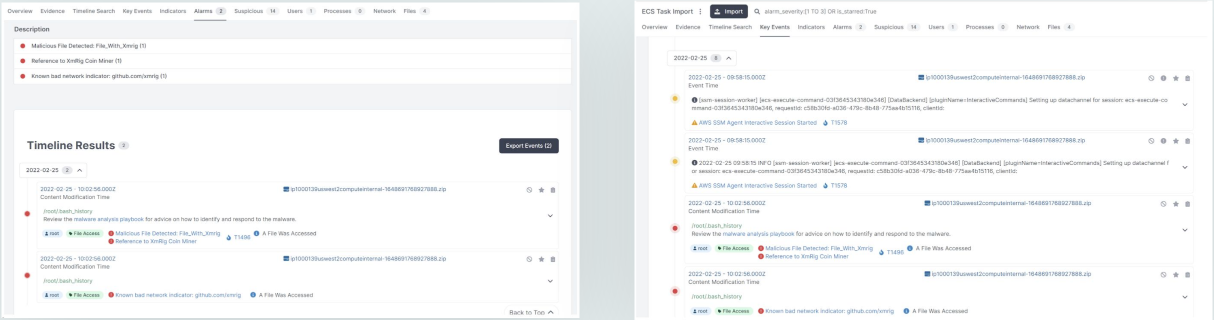
While the visibility provided by built-in CSP tools (e.g. AWS CloudWatch and CloudTrail) is important, these data sources alone are not sufficient to perform an in-depth investigation. Leveraging third-party incident and threat intelligence capabilities prove vital to gain a deeper level of visibility across container assets. In this context, the useful data to include as part of your investigation are the system logs and files from within the container, the containers running processes and active network connections, the container host system and container runtime logs (if accessible), the container host memory (if accessible) and the AWS VPC flow logs for the VPC the container is attached to.If data collection wasn't baked into the container declaration before the need to investigate arose, you need to rely on data you can actively interrogate out of the container.

Or if you’re using ECS on Fargate, taking an image of the running container or the container host isn’t an option as the underlying AWS infrastructure is shared with no access to the end customers. In this case, the /proc directory gives us a snapshot of the volatile state of the container, much like a memory dump. Using this ‘snapshot’ we can build a basic picture of what is going on in the container at that time, such as running processes, active network connections, open files etc. This data provides a foundation of what you need to correlate with other data sources such as firewall logs, network subnet flows etc during an investigation to understand the actions carried out by an attacker.

Cado Security Resources

We’ve published a number of materials on investigating compromises in ECS including:

- AWS ECS: Fully Managed but Frustrating to Investigate
- Top 9 Best Practices for AWS ECS Security
- Cado Response in Action: Investigating ECS Fargate



Description

- Malicious File Detected: File_With_Xmrig (1)
- Reference to XmRig Coin Miner (1)
- Known bad network indicator: github.com/xmrig (1)

Timeline Results 2

2022-02-25 2

2022-02-25 - 10:02:56.000Z ip1000139uswest2computeinternal-1648691768927888.zip
Content Modification Time
/root/.bash_history
Review the malware analysis playbook for advice on how to identify and respond to the malware.
root File Access Malicious File Detected: File_With_Xmrig Reference to XmRig Coin Miner T1496 A File Was Accessed

2022-02-25 - 10:02:56.000Z ip1000139uswest2computeinternal-1648691768927888.zip
Content Modification Time
/root/.bash_history
root File Access Known bad network indicator: github.com/xmrig A File Was Accessed

alarm_severity:[1 TO 3] OR is_starred:True

2022-02-25 8

2022-02-25 - 09:58:15.000Z ip1000139uswest2computeinternal-1648691768927888.zip
Event Time
[ssm-session-worker] [ecs-execute-command-03f3645343180e346] [DataBackend] [pluginName=InteractiveCommands] Setting up datachannel for session: ecs-execute-command-03f3645343180e346, requestId: c58b30fd-a036-479c-8b48-775aa4b15116, clientId:
AWS SSM Agent Interactive Session Started T1578

2022-02-25 - 09:58:15.000Z ip1000139uswest2computeinternal-1648691768927888.zip
Event Time
2022-02-25 09:58:15 INFO [ssm-session-worker] [ecs-execute-command-03f3645343180e346] [DataBackend] [pluginName=InteractiveCommands] Setting up datachannel for session: ecs-execute-command-03f3645343180e346, requestId: c58b30fd-a036-479c-8b48-775aa4b15116, clientId:
AWS SSM Agent Interactive Session Started T1578

2022-02-25 - 10:02:56.000Z ip1000139uswest2computeinternal-1648691768927888.zip
Content Modification Time
/root/.bash_history
Review the malware analysis playbook for advice on how to identify and respond to the malware.
root File Access Malicious File Detected: File_With_Xmrig Reference to XmRig Coin Miner T1496 A File Was Accessed

2022-02-25 - 10:02:56.000Z ip1000139uswest2computeinternal-1648691768927888.zip
Content Modification Time
/root/.bash_history
root File Access Known bad network indicator: github.com/xmrig A File Was Accessed

Example analysis of a compromised AWS ECS System in Cado Response

Lambda Incident Response

The main challenge when investigating a potentially compromised serverless system is working out just what data to collect. Whilst you can't typically access the underlying container on which Lambda runs (without hooking into libraries at deploy time to collect the writable /tmp folder at run time) you can grab:

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

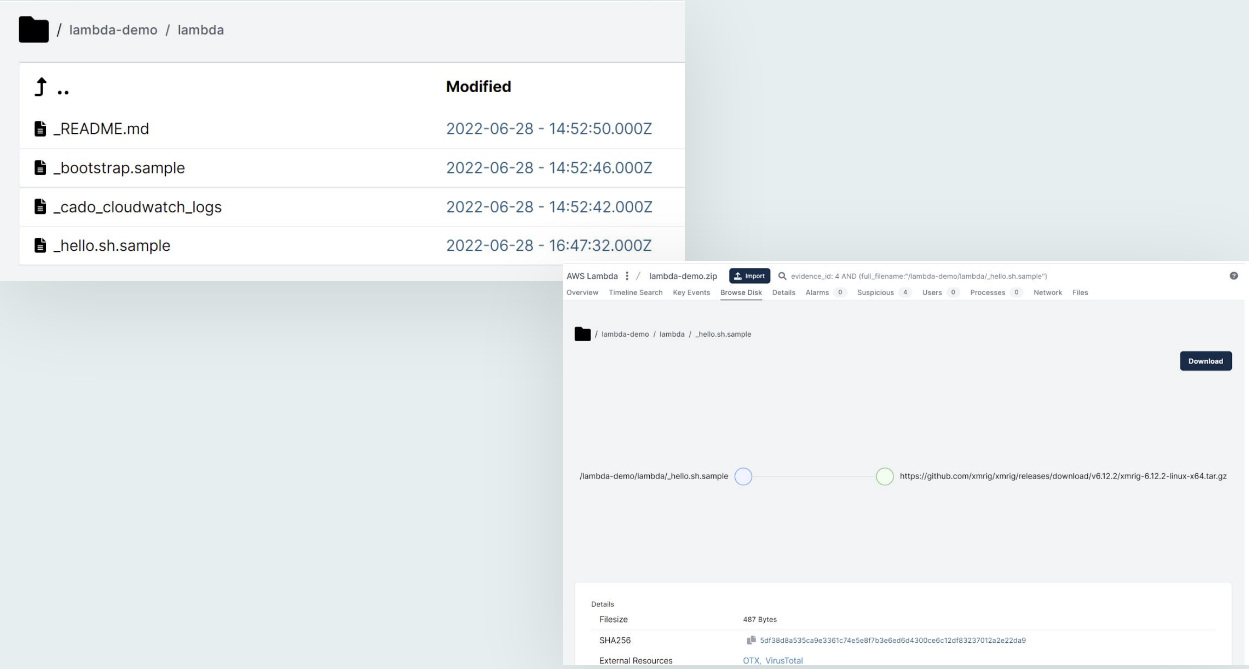
Community Resources

Whilst more focused on investigating logs for errors, this post on StackOverflow is the best advice we've seen for practical advice on how to search through logs from Lambda.

Cado Security Resources

We've published a number of materials on investigating compromises in ECS including:

- Cado Security Extends Support To Serverless Environments
- Cado Discovers Denonia: The First Malware Specifically Targeting Lambda



/ lambda-demo / lambda

..	Modified
_README.md	2022-06-28 - 14:52:50.000Z
_bootstrap.sample	2022-06-28 - 14:52:46.000Z
_cado_cloudwatch_logs	2022-06-28 - 14:52:42.000Z
_hello.sh.sample	2022-06-28 - 16:47:32.000Z

evidence_id: 4 AND (full_filename:"/lambda-demo/lambda/_hello.sh.sample")

/ lambda-demo / lambda / _hello.sh.sample

/lambda-demo/lambda/_hello.sh.sample ○ ─── ○ https://github.com/xmrig/xmrig/releases/download/v6.12.2/xmrig-6.12.2-linux-x64.tar.gz

Details

Filesize	487 Bytes
SHA256	5df38d8a535ca9e3361c74e5e8f7b3e6ed6d4300ce6c12df83237012a2e22da9
External Resources	OTX, VirusTotal

Example analysis of a compromised AWS Lambda function in Cado Response

Official AWS Resources

AWS have published a number of guides which may be useful when investigating a suspect Lambda function:

- [Accessing Amazon CloudWatch logs for AWS Lambda]()
- [Security in AWS Lambda]()

Investigating Other Resources

Below we've listed some resources for responding to compromised systems in other AWS environments.

ECR

Amazon Elastic Container Registry is a service to host Docker container images, much like DockerHub. Though rare, attackers can backdoor either popular public containers or even private ones in accounts they have compromised. In the worst case, this can lead to deploying backdoored images.

- Image Scanning in ECR (AWS)

S3

Thanks to improvements in the default settings, data loss incidents caused by accidentally setting private data public in S3 are less common than they used to be. If enabled, access logs can help inform you what data was accessed (limiting the scope of mandatory data-breach notifications) and CloudTrail logs can be used identify when data was set public (again limiting the scope if notifications).

- Logging and monitoring in Amazon S3 (AWS)
- Understanding and Preventing S3 Leaks (Security Boulevard)
- Misconfigured Amazon S3 Buckets Continue to be a Launchpad for Malicious Code (Microsoft)

IAM

AWS IAM underpins all other services, enabling and controlling access. It is likely that any non-trivial investigation in AWS will involve IAM.

- Logging IAM and AWS STS API calls with AWS CloudTrail (AWS)
- AWS, IAM Your Father (Part II - Defensive) (AllThingsDFIR)

CloudTrail

Whilst AWS logs to a number of places, CloudTrail is the key platform to become familiar with first.

- Investigating CloudTrail Logs (Medium)
- AWS CloudTrail —Searching Event logs in S3, Athena and Cloudwatch

Training

There are a number of training courses for responding to incidents in AWS, both free and paid:

- AWS Incident Response Playbooks Workshop (AWS)
- Incident Response with AWS Console and CLI (WellArchitetedLabs)
- EC2 DFIR Workshop (Forensicate.cloud)
- Enterprise Cloud Forensics and Incident Response (SANS)

Legal Requirements during Incident Response

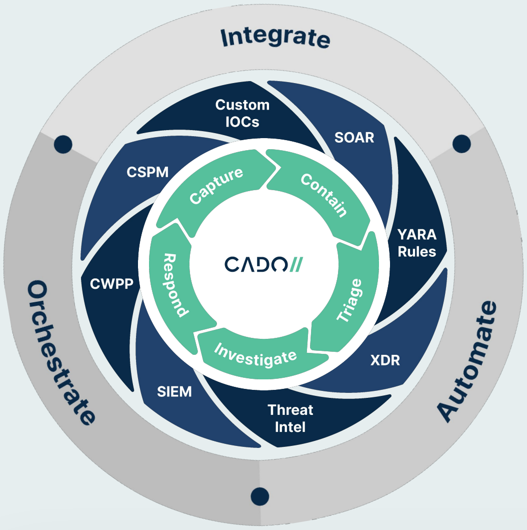
- General Data Protection Regulation in AWS (AWS)

Automating Incident Response

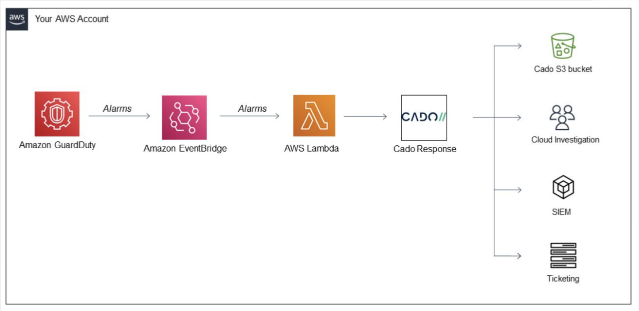
Automating the collection of incident evidence immediately following detection helps ensure security events are appropriately handled before they are at risk of escalating. The lack of automation coupled with alert fatigue often means things are missed and what may seem like a low-severity detection, may actually be connected to something far more malicious. Leveraging automation to remove many of the complexities and manual steps in kicking off a more thorough investigation means security teams can dive deep more often and better protect their environment. By automating evidence collection, analysts save days and in some cases, even weeks during an investigation.

Cado Security Resources

You can also view our tutorials on how to set up integrations with SOAR platforms such as Splunk SOAR and Tines online.



We have also published some example code on our Github to forensically process systems with Cado Response that have been detected as compromised by GuardDuty:



Official AWS Resources

AWS provides a number of tools which may be useful when automating response to security incidents:

- EventBridge is a serverless event bus that makes it easy to connect applications together with data from your own applications, integrated Software-as-a-Service (SaaS) applications, and AWS services.
- Lambda can automatically run code in response to events, making it a great choice for event-driven applications.
- Incident Manager is a new capability of AWS Systems Manager to track incidents in AWS as they progress.
- SSM allows you to execute isolation commands directly on EC2 systems from the inside.
- Security Hub - Aggregates a number of different AWS security tools like **Amazon Detective** together.
- Standard AWS APIs allow you to change a security group to quarantine a system, for example.

AWS provides a workshop to help you build your own semi-automated Play Books. They also provide documentation, a video and a white-paper which can help pad out incident response plans.

For More Information

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