What is happening in your AWS account?

An intro to logging, processing and responding to activity in your AWS environment

About me

- Infrastructure security lead @ Bitstamp
- Lately dealing a lot with AWS
- In the past Product security lead and prior to that software engineer for multiple years

Today’s talk

- One of the fundamentals of security is knowing what is actually going on
- Content:
  - 1. getting the coverage you need - log sources and what can they tell us
  - 2. collection - where to put them and what will help you
  - 3. respond when something happens - how and tools

(Part 1 - 1/5) The baseline - CloudTrail

- By default it logs what you do with your AWS infrastructure: Console, CLI, SDKs and APIs (management events).
- Also: data events and insight events.
- It produces *events* with additional data about what happened - time, event name, related arns, some request data like IP, user-agent, and user, assumed role etc.
- Event selectors (reduce n of events you do not care about)
- Management events enabled by default, only stored 90 days, no persistence.
- Gotchas:
  - Can include sensitive information -> make sure you secure them (content of bucket, lambda params).
  - There is no trace id :(
  - PassRole :'(Auditing IAM PassRole: A Problematic Privilege Escalation Permission (Ermetic)
  - Not all services and everything is covered: supported, unsupported
  - Parameters and crucial data might be missing - example on the next page
  - Organizational trail -> very handy
  - Duplicate events -> $$$:(

Trails	You can deliver one copy of your ongoing management events to your Amazon Simple Storage Service (S3) bucket for free by creating trails. Limits may apply.

(Part 1 - 2/5) The baseline - CloudTrail - examples

Data events Info

Data events show information about the resource operations performed on or within a resource. Additional charges apply

Advanced event selectors are enabled
Use the following fields for fine-grained control over the data events captured by your trail.

Switch to basic event selectors

▾ Data event: S3

Remove

Data event type
Choose the source of data events to log.

S3
- S3 ✓
- Lambda
- DynamoDB
- S3 Outposts
- Managed Blockchain
- S3 Object Lambda
- Lake Formation
- EC2 instance connect endpoint
- Cognito Identity Pools
- Kendra Ranking
- SageMaker metrics experiment trial component
- EBS direct APIs
- S3 Access Point
- DynamoDB Streams
- CloudTrail
- GuardDuty detector
- FinSpace
- SageMaker feature store

Add data event type

Advanced event selectors Info

Log or exclude events from specific resources.

Field	Operator	Value
readOnly	equals	false
AND + Condition		
resources.ARN	does not equal	arn:aws:s3:::somebucketidontwannalog (Browse)

+ Field + Condition

▾ JSON view

```
[
  {
    "name": "",
    "fieldSelectors": [
      {
        "field": "eventCategory",
        "equals": [
          "Data"
        ]
      },
      {
        "field": "resources.type",
        "equals": [
          "AWS::S3::Object"
        ]
      },
      {
        "field": "readOnly",
        "equals": [
          "false"
        ]
      },
      {
        "field": "resources.ARN",
        "notEquals": [
          "arn:aws:s3:::somebucketidontwannalog"
        ]
      }
    ]
  }
]
```

Parameters can appear in the request without being logged by CloudTrail

CloudTrail events aren't exact representations of the actions they log; sometimes certain request parameters just aren't logged for certain actions. Take a look at this CloudTrail event for elasticbeanstalk:AssociateEnvironmentOperationsRole (some details redacted):

Raw | Copy | Extern | EnlighterJS

```
1.  {
2.      "eventVersion": "1.05",
3.      "userIdentity": {
4.          "type": "IAMUser",
5.          "principalId": <principalId>,
6.          "arn": "arn:aws:iam::123456789012:user/<user>",
7.          "accountId": "123456789012",
8.          "accessKeyId": <accessKeyId>,
9.          "userName": <user>
10.     },
11.     "eventTime": "2020-12-14T14:39:53Z",
12.     "eventSource": "elasticbeanstalk.amazonaws.com",
13.     "eventName": "AssociateEnvironmentOperationsRole",
14.     "awsRegion": "us-east-1",
15.     "sourceIPAddress": "xxx.xxx.xxx.xxx",
16.     "userAgent": "aws-cli/1.18.185 Python/3.9.0+ Linux/4.4.0-19041-
    Microsoft botocore/1.19.25",
17.     "requestParameters": null,
18.     "responseElements": null,
19.     "requestID": <GUID>,
20.     "eventID": <GUID>,
21.     "eventType": "AwsApiCall",
22.     "recipientAccountId": "123456789012"
23. }
```

The action request includes the parameters "EnvironmentName" and "OperationsRole" but **Cloud Trail has not logged them**. It's therefore impossible to tell which role was passed from the CloudTrail log.

(Part 1 - 3/5) More coverage

Log sources

- VPC flow logs
- Route53 resolver query logs
- Config resource change events
- CloudFront access logs
- WAF firewall logs
- StepFunctions workflow logs
- RDS database activity streams
- Lambda application logs
- …

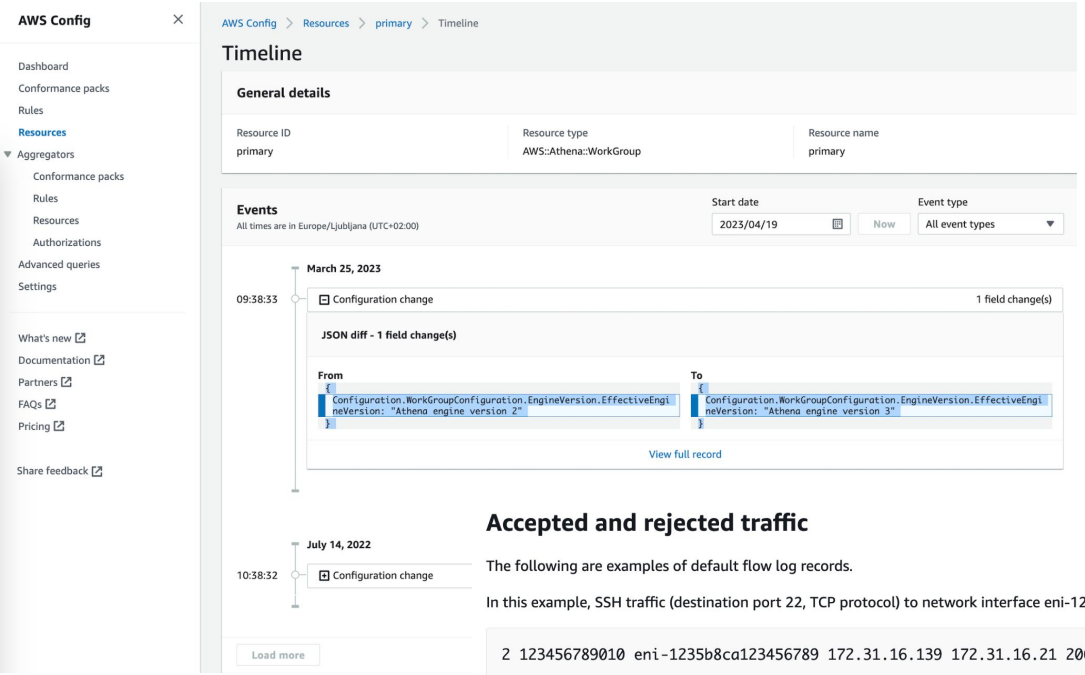
Services

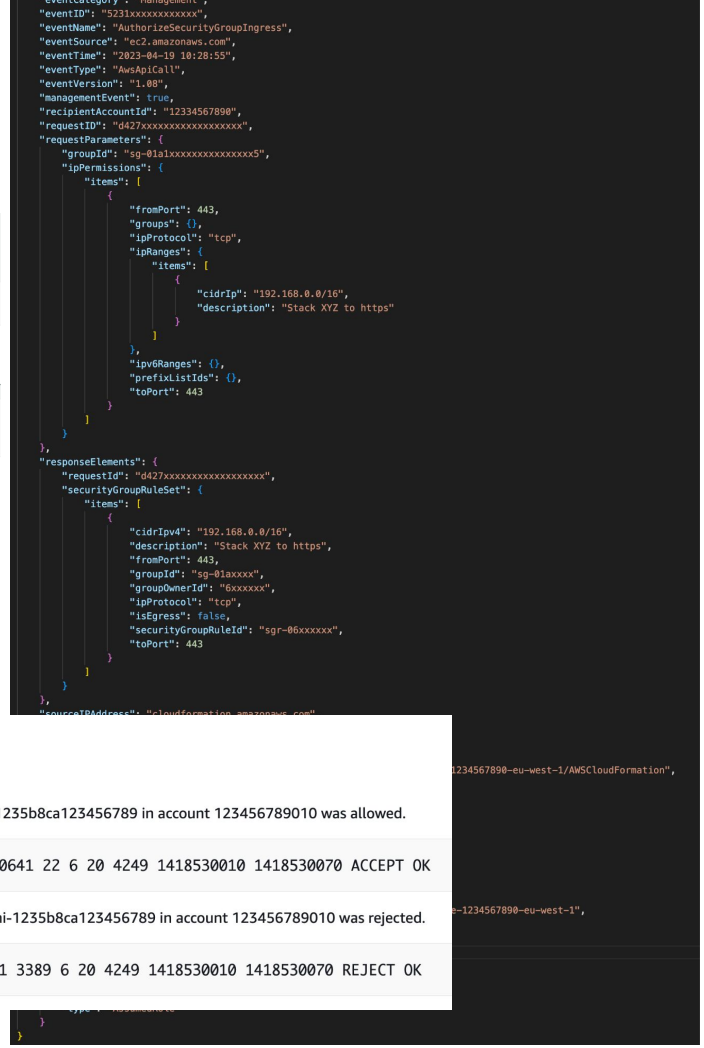
- AWS Config -> see diffs, very nice, query past and current state
- GuardDuty (builds its own lake of logs)
- Security Hub
- Detective

How to enable logging on every AWS service in existence circa 2021

https://github.com/matthewdfuller/aws-guides/blob/main/aws-logging-services.csv

(Part 1 - 4/5) Examples 1





Accepted and rejected traffic

The following are examples of default flow log records.

In this example, SSH traffic (destination port 22, TCP protocol) to network interface eni-1235b8ca123456789 in account 123456789010 was allowed.

```
2 123456789010 eni-1235b8ca123456789 172.31.16.139 172.31.16.21 20641 22 6 20 4249 1418530010 1418530070 ACCEPT OK
```

In this example, RDP traffic (destination port 3389, TCP protocol) to network interface eni-1235b8ca123456789 in account 123456789010 was rejected.

```
2 123456789010 eni-1235b8ca123456789 172.31.9.69 172.31.9.12 49761 3389 6 20 4249 1418530010 1418530070 REJECT OK
```

Trojan:EC2/DGADomainRequest.B

Finding ID: 94c3[redacted]

Feedback

High EC2 instance i-0a[redacted]3 is querying algorithmically generated domains. Such domains are commonly used by malware and could be an indication of a compromised EC2 instance. Info

Investigate with Detective

Overview

Severity	HIGH
Region	eu-central-1
Count	102
Account ID	[redacted]
Resource ID	i-0a8[redacted]
Created at	03-13-2023 15:02:35 (a month ago)
Updated at	04-19-2023 10:14:09 (2 hours ago)

Malware scan (not enabled)

Scan status	Enable Malware Protection

Resource affected

Resource role	TARGET
Resource type	Instance

Instance details

Instance ID	i-0a8[redacted]
Instance type	t3.medium
Instance state	running
Availability zone	eu-central-1a
Image ID	ami-2[redacted]
Image description	Amazon Linux 2 AMI 2[redacted]
Launch time	03-10-2023 13:50:56

IAM instance profile

ARN	arn:aws:iam::1[redacted]5...
ID	[redacted]

Instance tags

Owner	infrasec
AWS :cloudformation:logical-id	[redacted]spiderfoot[redacted]
AWS :cloudformation:stack-name	[redacted]piderfoot
Name	production-spiderfoot
AWS :cloudformation:stack-id	arn:aws:cloudformation:eu-cen[redacted]-lac...
Repository	[redacted]
Stack name	[redacted]
Context config	default
Purpose	production

Network interfaces

Network interface 0 (eni-077c9460b29ee9282)

Network interface ID	eni-077[redacted]
Private dns name	ip-10-0[redacted]mpute.internal
Private IP address	10.0.10[redacted]
Subnet ID	subnet-08[redacted]
VPC ID	vpc-0e2[redacted]

Private IP addresses

Private dns name	[redacted].central-1.compute.internal
Private IP address	10.0.10[redacted]

Security groups

Group ID	sg-0c[redacted]
Group name	la[redacted]i...

Action

Action type	DNS_REQUEST
Protocol	0
Blocked	false

Actor

Domain	st[redacted]xd.onion

Additional information

Archived	false

Security Hub > Security standards > AWS Foundational Security Best Practices v1.0.0 > KMS.3

Last updated: 13 hours ago Regions: All Linked Regions

AWS KMS keys should not be deleted unintentionally

[KMS.3] This control checks whether AWS Key Management Service (KMS) customer managed keys (CMK) are scheduled for deletion. The control fails if a KMS CMK is scheduled for deletion. Remediation instructions

Control status | Standards and requirements

Control status

Security Hub calculates the enablement and compliance status for controls every 24 hours. When you change the enablement status of a control, it applies only to the current account and Region.

Disable Control

Status	Severity
Failed (Updated 13 hours ago)	Critical

All checks | Failed | Unknown | Passed | Suppressed

[redacted]

All checks (100)

Actions | Workflow status | Download | 1 ...

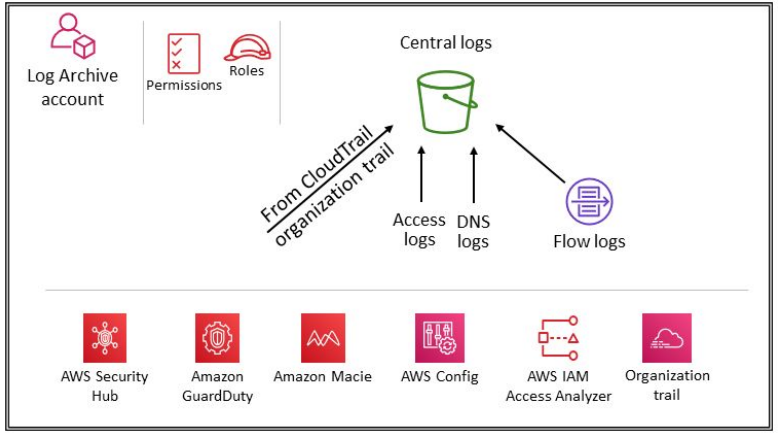
Add filters

Compliance Status	Workflow	Account	Region	Resource	Investigate	Updated	Finding .json
FAILED	NEW	[redacted]		KMS Key [redacted]		10 hours ago	
FAILED	NEW			KMS Key [redacted]		10 hours ago	
FAILED	SUPPRESSED			KMS Key [redacted]		11 hours ago	

(Part 1 - 5/5) Examples 2

(Part 2 - 1/1) Collect logs

- AWS Prescriptive Guidance / Security Reference Architecture proposes separate **Log Archive** account with dedicated S3 buckets. Also Control Tower configures it for you out of the box.
- **Central location** helps you making sure your **policies** ensuring **access**, **integrity**, **durability**, and dealing with **sensitivity** are controlled: log integrity setup, access to view and modify is locked, backups (CIA) ..
- **Security Lake** takes similar approach - manages for you, but is heavily WIP and you will need to write extensions for sources it does not support. Dumps parquet -> Athena -> very nice
- **CloudTrail Lake** - only for CT, Config and external integration, SQL-like query language
- Processing - S3 will have delay, can be minutes, where response needs to be ASAP, you want to hook up either to SQS, Kinesis Firehose (these two are usually supported on log sources, but then you need to manage S3 store yourself - with **CloudTrail duplicate trail is charged extra**).
- Use external tools (there is loads and often they are $$$) - “SIEM”): Elastic, Panther, Logtail, Matano (OSS, AWS native), Splunk, DataDog…

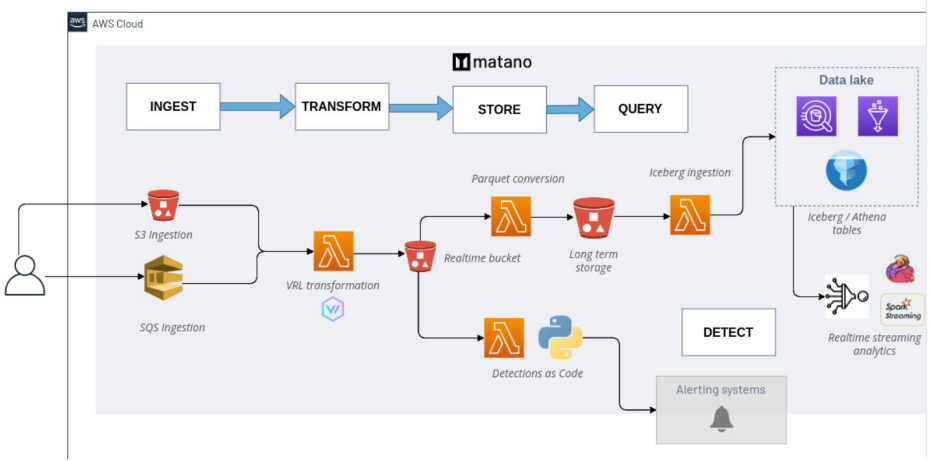


Security Lake collects data from custom sources in addition to the following AWS se

- AWS CloudTrail management and data events (S3, Lambda)
- Amazon Route 53 resolver query logs
- AWS Security Hub findings
- Amazon Virtual Private Cloud (Amazon VPC) flow logs

(Part 3 - 1/2) Process and respond

- Basic way to automatically respond: lambda hook to source SQS, KF or S3 event **BUT**
- it will be hard to correlate events this way or do some **deduplication** and **aggregation** -> tools like Matano can help you with that (or your favorite SIEM)
- Detective will help you drill down



https://www.streamalert.io/architecture.html

https://www.matano.dev/

(Part 3 - 2/2) Example

Detect failed attempts to export AWS EC2 instance in AWS CloudTrail logs.

```
def detect(record):
  return (
    record.deepget("event.action") == "CreateInstanceExportTask"
    and record.deepget("event.provider") == "ec2.amazonaws.com"
    and record.deepget("event.outcome") == "failure"
  )
```

Detect Brute Force Logins by IP across all configured log sources (e.g. Okta, AWS, GWorkspace)

detect.py

```
def detect(r):
    return (
        "authentication" in r.deepget("event.category", [])
        and r.deepget("event.outcome") == "failure"
    )


def title(r):
    return f"Multiple failed logins from {r.deepget('user.full_name')} - {r.deepget('source.ip')}"


def dedupe(r):
    return r.deepget("source.ip")
```

detection.yml

```
---
tables:
  - aws_cloudtrail
  - okta_system
  - o365_audit
alert:
  severity: medium
  threshold: 5
  deduplication_window_minutes: 15
  destinations:
    - slack_my_team
```

Questions?

Thank you!