2024
DUBLIN

2 - 4 October, 2024 / Dublin, Ireland

DREDGE – AN OPEN-SOURCE CLOUD DFIR KIT

Santiago Abastante

Solidarity Labs, Argentina

sabastante@solidaritylabs.io

ABSTRACT

Cloud incident response can be daunting, requiring a plethora of tools and skills, and while most cloud-based startups can't allocate budget for preventive controls, there is less space for them to understand what to do if they are hacked.

That's why I created Dredge, an open-source framework designed to streamline cloud incident investigations, by allowing cloud engineers and incident responders to execute non-trivial response tasks effortlessly, irrespective of their familiarity with specific cloud platforms or incident response tactics.

The main idea is to empower engineers to respond to attacks no matter what preparation they have had, taking advantage of most of the out-of-the box security features cloud providers offer but not everybody is aware of – such as being able to retrieve a forensic image from a running server or getting logs that they didn't know they had.

The following are some key features that differentiate Dredge from existing tooling:

- Python-based CLI.
- Retrieve logs seamlessly from *GitHub*, *Kubernetes*, *AWS*, *GCP* or *Azure*.
- Take action: whether it's blocking an IP in an *AWS* tenant, disabling an AccessKey, isolating an EC2 instance, or strategically extracting crucial post-compromise user data.
- Identify tactical misconfigurations that can be exploited by an attacker.
- Create an attack timeline based on IOCs.
- Analyse retrieved data effortlessly within a terminal, utilizing built-in capabilities from *VirusTotal* and *Shodan*.
- Cloud incident response guidelines for companies to embrace and build their playbooks.

TECHNICAL CONCEPTS

AWS technical concepts

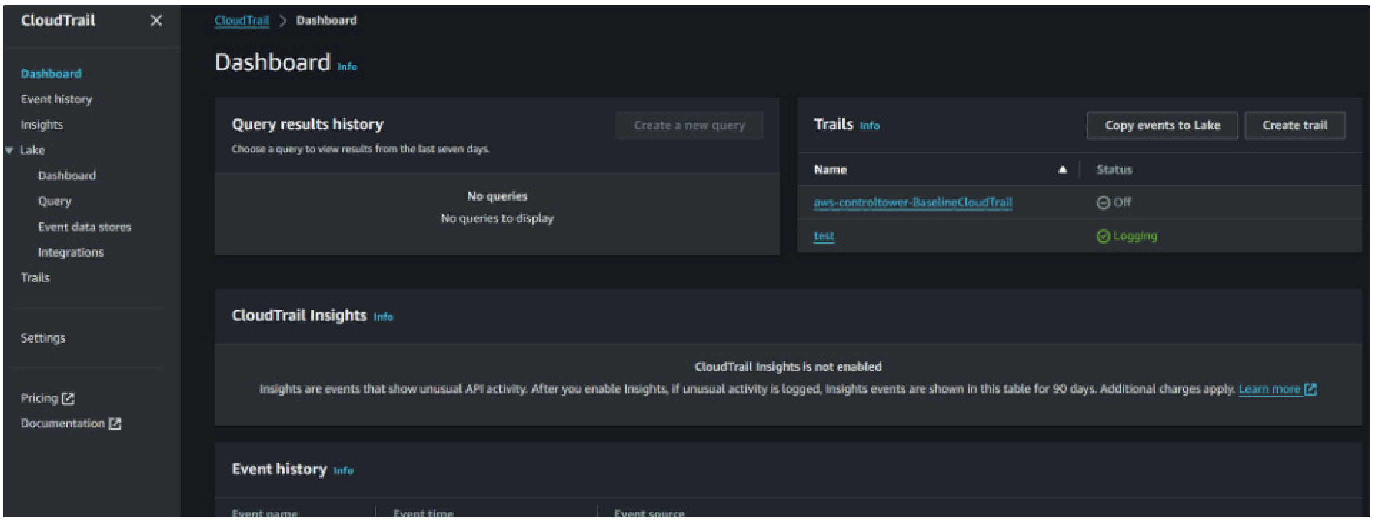
CloudTrail logs

Amazon CloudTrail is an *AWS* service that allows monitoring, logging, and retaining actions performed in the account or organization. *CloudTrail* provides details of executed API calls, including the identity of the actor, the timestamp, the source IP address, the request parameters, and the response.

In *AWS*, an API call is an invocation of a function or command using the AWS Management Console, the CLI, or the AWS SDKs (Software Development Kits, such as Boto3). API stands for Application Programming Interface and allows applications to interact with each other.

When you interact with an *AWS* service, such as creating an EC2 server or uploading a file to an S3 bucket, you are essentially making an API call. *CloudTrail* logs those API calls.

CloudTrail is not enabled by default, but there are some alternatives that we will see in the following sections.



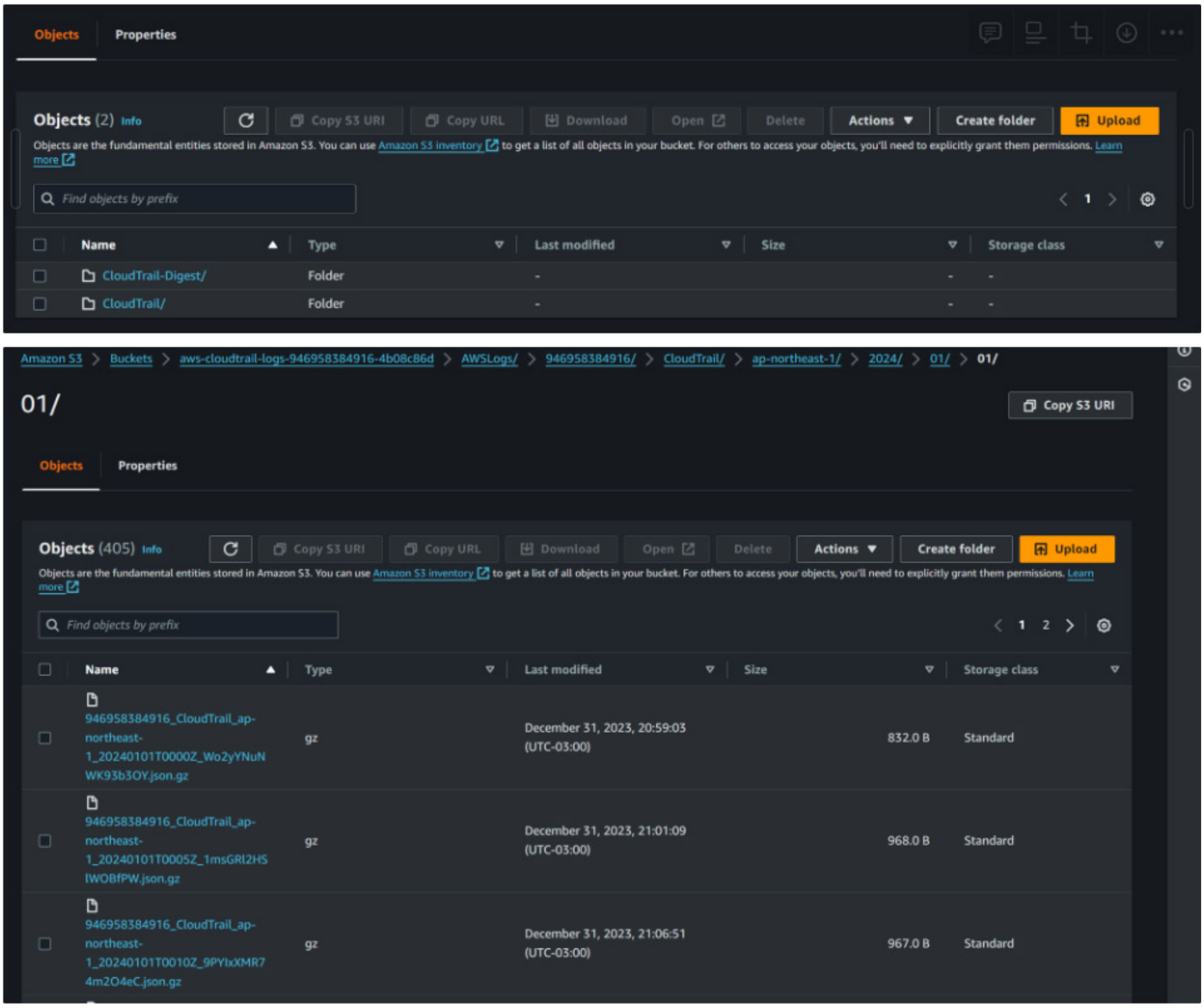
Choose a query to view results from the last seven days.

No queries
No queries to display

Name	Status
aws-controltower-BaselineCloudTrail	Off
test	Logging

CloudTrail Insights is not enabled
Insights are events that show unusual API activity. After you enable Insights, if unusual activity is logged, Insights events are shown in this table for 90 days. Additional charges apply. Learn more

Figure 1: CloudTrail console.



Objects (2) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
CloudTrail-Digest/	Folder	-	-	-
CloudTrail/	Folder	-	-	-

Amazon S3 > Buckets > aws-cloudtrail-logs-946958384916-4b08c86d > AWSLogs/ > 946958384916/ > CloudTrail/ > ap-northeast-1/ > 2024/ > 01/ > 01/

01/

Objects (405) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
946958384916_CloudTrail_ap-northeast-1_20240101T0000Z_Wo2yYNuNWK93b3OY.json.gz	gz	December 31, 2023, 20:59:03 (UTC-03:00)	832.0 B	Standard
946958384916_CloudTrail_ap-northeast-1_20240101T0005Z_1msGRI2HSlWOBfPW.json.gz	gz	December 31, 2023, 21:01:09 (UTC-03:00)	968.0 B	Standard
946958384916_CloudTrail_ap-northeast-1_20240101T0010Z_9PYIxXMR74m2O4eC.json.gz	gz	December 31, 2023, 21:06:51 (UTC-03:00)	967.0 B	Standard

Figure 2: CloudTrail logs in S3 bucket.

As you can see, the logs are stored within a folder structure inside the bucket that follows the following scheme:

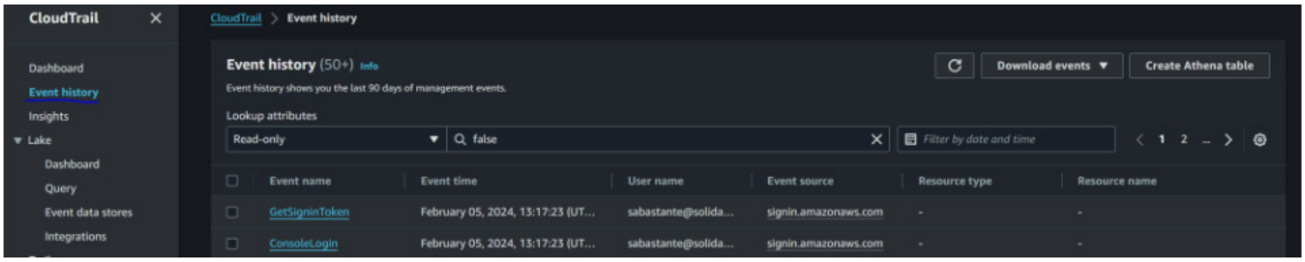
AWSLogs/**{account number}**/CloudTrail/**{region}**/**{year}**/**{month}**/**{day}**/log.json.gz

To view these logs, we have several alternatives:

1. Download them
2. Integrate them with a third-party solution like a SIEM
3. Implement a structure with *AWS* services (Glue + Athena to be covered later)
4. Export and view them in *CloudWatch* (to be covered later)

Event history logs

Event history also stores *AWS* API logs, but is simpler to use; we can view the account logs directly from the *AWS* console:



Event history (50+) Info

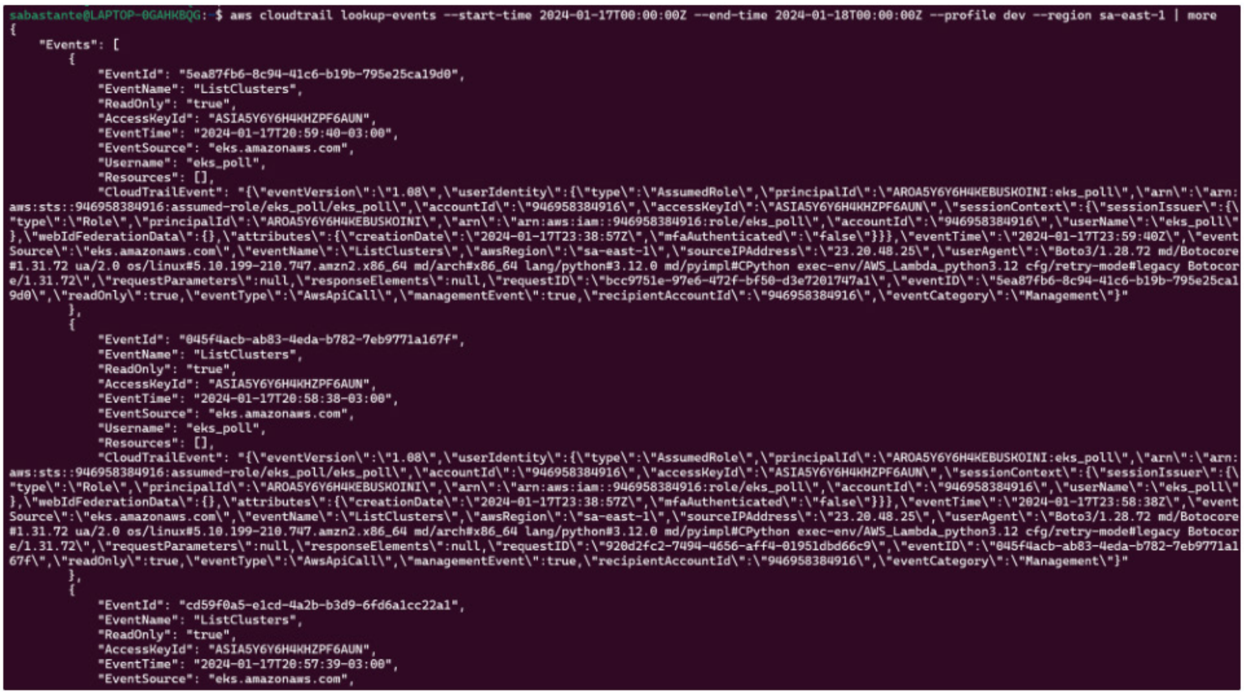
Event history shows you the last 90 days of management events.

Lookup attributes: Read-only, false

Event name	Event time	User name	Event source	Resource type	Resource name
GetSigninToken	February 05, 2024, 13:17:23 (UT...	sabastante@solida...	signin.amazonaws.com	-	-
ConsoleLogin	February 05, 2024, 13:17:23 (UT...	sabastante@solida...	signin.amazonaws.com	-	-

Figure 3: Event history allows us to view account logs from the AWS console.

The complexity begins when we need to visualize logs from multiple different accounts, for which we might want to use the API.



```
sabastante@LAPTOP-0GAHKBQG:~$ aws cloudtrail lookup-events --start-time 2024-01-17T00:00:00Z --end-time 2024-01-18T00:00:00Z --profile dev --region sa-east-1 | more
{
    "Events": [
        {
            "EventId": "5ea87fb6-8c94-41c6-b19b-795e25ca19d0",
            "EventName": "ListClusters",
            "ReadOnly": "true",
            "AccessKeyId": "ASIA5Y6Y6H4KHZPF6AUN",
            "EventTime": "2024-01-17T20:59:40-03:00",
            "EventSource": "eks.amazonaws.com",
            "Username": "eks_poll",
            "Resources": [],
            "CloudTrailEvent": "{\"eventVersion\":\"1.08\",\"userIdentity\":{\"type\":\"AssumedRole\",\"principalId\":\"AROA5Y6Y6H4KEBUSKOINI:eks_poll\",\"arn\":\"arn:
aws:sts::946958384916:assumed-role/eks_poll/eks_poll\",\"accountId\":\"946958384916\",\"accessKeyId\":\"ASIA5Y6Y6H4KHZPF6AUN\",\"sessionContext\":{\"sessionIssuer\":{\
"type\":\"Role\",\"principalId\":\"AROA5Y6Y6H4KEBUSKOINI\",\"arn\":\"arn:aws:iam::946958384916:role/eks_poll\",\"accountId\":\"946958384916\",\"userName\":\"eks_poll\"
},\"webIdFederationData\":{},\"attributes\":{\"creationDate\":\"2024-01-17T23:38:57Z\",\"mfaAuthenticated\":\"false\"}}},\"eventTime\":\"2024-01-17T23:59:40Z\",\"event
Source\":\"eks.amazonaws.com\",\"eventName\":\"ListClusters\",\"awsRegion\":\"sa-east-1\",\"sourceIPAddress\":\"23.20.48.25\",\"userAgent\":\"Boto3/1.28.72 md/Botocore
#1.31.72 ua/2.0 os/linux#5.10.199-210.747.amzn2.x86_64 md/arch#x86_64 lang/python#3.12.0 md/pyimpl#CPython exec-env/AWS_Lambda_python3.12 cfg/retry-mode#legacy Botocor
e/1.31.72\",\"requestParameters\":null,\"responseElements\":null,\"requestID\":\"bcc9751e-97e6-472f-bf50-d3e7201747a1\",\"eventID\":\"5ea87fb6-8c94-41c6-b19b-795e25ca1
9d0\",\"readOnly\":true,\"eventType\":\"AwsApiCall\",\"managementEvent\":true,\"recipientAccountId\":\"946958384916\",\"eventCategory\":\"Management\"}"
        },
        {
            "EventId": "045f4acb-ab83-4eda-b782-7eb9771a167f",
            "EventName": "ListClusters",
            "ReadOnly": "true",
            "AccessKeyId": "ASIA5Y6Y6H4KHZPF6AUN",
            "EventTime": "2024-01-17T20:58:38-03:00",
            "EventSource": "eks.amazonaws.com",
            "Username": "eks_poll",
            "Resources": [],
            "CloudTrailEvent": "{\"eventVersion\":\"1.08\",\"userIdentity\":{\"type\":\"AssumedRole\",\"principalId\":\"AROA5Y6Y6H4KEBUSKOINI:eks_poll\",\"arn\":\"arn:
aws:sts::946958384916:assumed-role/eks_poll/eks_poll\",\"accountId\":\"946958384916\",\"accessKeyId\":\"ASIA5Y6Y6H4KHZPF6AUN\",\"sessionContext\":{\"sessionIssuer\":{\
"type\":\"Role\",\"principalId\":\"AROA5Y6Y6H4KEBUSKOINI\",\"arn\":\"arn:aws:iam::946958384916:role/eks_poll\",\"accountId\":\"946958384916\",\"userName\":\"eks_poll\"
},\"webIdFederationData\":{},\"attributes\":{\"creationDate\":\"2024-01-17T23:38:57Z\",\"mfaAuthenticated\":\"false\"}}},\"eventTime\":\"2024-01-17T23:58:38Z\",\"event
Source\":\"eks.amazonaws.com\",\"eventName\":\"ListClusters\",\"awsRegion\":\"sa-east-1\",\"sourceIPAddress\":\"23.20.48.25\",\"userAgent\":\"Boto3/1.28.72 md/Botocore
#1.31.72 ua/2.0 os/linux#5.10.199-210.747.amzn2.x86_64 md/arch#x86_64 lang/python#3.12.0 md/pyimpl#CPython exec-env/AWS_Lambda_python3.12 cfg/retry-mode#legacy Botocor
e/1.31.72\",\"requestParameters\":null,\"responseElements\":null,\"requestID\":\"920d2fc2-7494-4656-aff4-01951dbd66c9\",\"eventID\":\"045f4acb-ab83-4eda-b782-7eb9771a1
67f\",\"readOnly\":true,\"eventType\":\"AwsApiCall\",\"managementEvent\":true,\"recipientAccountId\":\"946958384916\",\"eventCategory\":\"Management\"}"
        },
        {
            "EventId": "cd59f0a5-e1cd-4a2b-b3d9-6fd6a1cc22a1",
            "EventName": "ListClusters",
            "ReadOnly": "true",
            "AccessKeyId": "ASIA5Y6Y6H4KHZPF6AUN",
            "EventTime": "2024-01-17T20:57:39-03:00",
            "EventSource": "eks.amazonaws.com",
```

Figure 4: Getting API logs using AWS event history API.

GitHub technical concepts

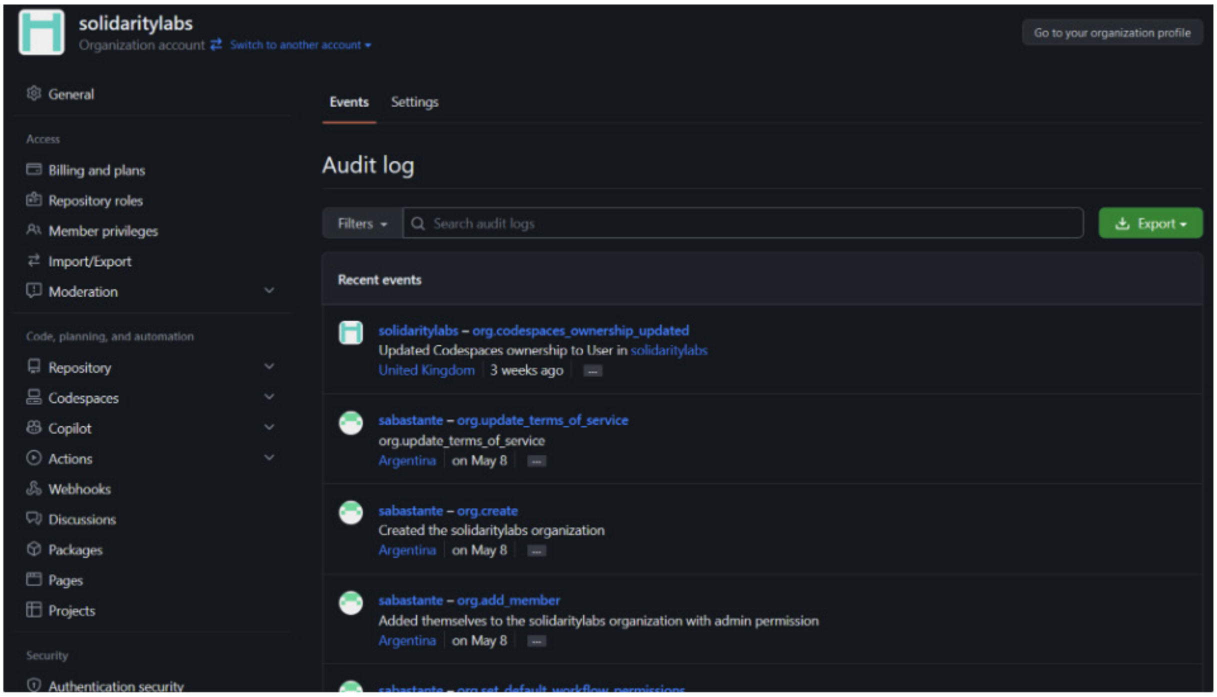
Obtaining *GitHub* logs can present several challenges. First and foremost is the issue of access permissions, as repository owners control who can view and retrieve logs. Moreover, *GitHub*'s rate-limiting policies can also hinder extensive log retrieval, making it crucial to use pagination and efficient querying.

Additionally, Audit Log Git events REST API and export capabilities are generally only available for *GitHub Enterprise Cloud* customers – the requirement for a premium subscription to access the *GitHub* API for log retrieval adds a cost barrier for some users.

It's also important to note that there are differences between the data accessible through the web management interface and via the API, as certain log details may only be available through direct API access.

GitHub provides several types of logs to help users monitor and analyse various aspects of their repositories and activity. Some of the common *GitHub* log types include:

1. **Audit logs**: these logs track actions taken within an organization, helping to monitor and audit user and system activities.
2. **Access logs**: access logs record who has accessed a repository or organization, providing information on who viewed or interacted with the content.
3. **Error logs**: error logs contain information about errors and issues that occur within the *GitHub* platform, aiding in troubleshooting and issue resolution.
4. **Commit logs**: these logs document changes made to a repository, including details about commits, branches and pull requests.
5. **Deployment logs**: deployment logs track the status and history of deployments, which is essential for managing the deployment process.
6. **Workflow run logs**: *GitHub* actions and workflows produce logs that capture the details of workflow runs, including build and test results.
7. **Issue and pull request logs**: these logs provide information about issues and pull requests, including comments, status changes, and assignments.
8. **Security logs**: security logs offer insights into security-related events, such as vulnerability scanning and alerts.
9. **Traffic logs**: traffic logs record traffic data for a repository, helping to understand the popularity and usage of the repository's content.



solidaritylabs
Organization account ⇄ Switch to another account ▾

Go to your organization profile

Audit log

Recent events

solidaritylabs – org.codespaces_ownership_updated
Updated Codespaces ownership to User in solidaritylabs
United Kingdom | 3 weeks ago | …

sabastante – org.update_terms_of_service
org.update_terms_of_service
Argentina | on May 8 | …

sabastante – org.create
Created the solidaritylabs organization
Argentina | on May 8 | …

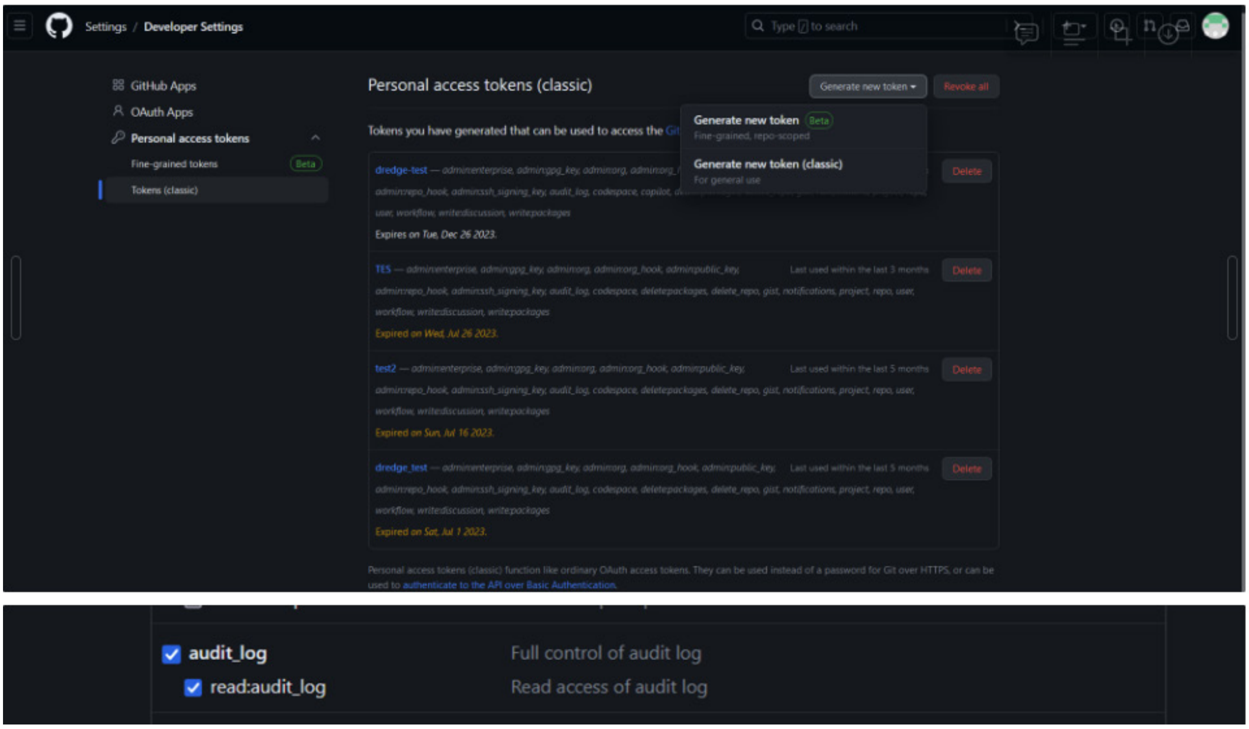
sabastante – org.add_member
Added themselves to the solidaritylabs organization with admin permission
Argentina | on May 8 | …

sabastante – org.set_default_workflow_permissions

Figure 5: Getting GitHub logs from management plane.

Access token creation

To get audit logs from the API for an organization, we need to create a user access token that requires `read:audit_log` permissions. `Admin:org` is not needed, despite what the documentation says.



Personal access tokens (classic)

Tokens you have generated that can be used to access the Gi

dredge-test — admin:enterprise, admin:gpg_key, admin:org, admin:org_
admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot,
user, workflow, write:discussion, write:packages
Expires on Tue, Dec 26 2023.

TES — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages
Last used within the last 3 months
Expired on Wed, Jul 26 2023.

test2 — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages
Last used within the last 5 months
Expired on Sun, Jul 16 2023.

dredge_test — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages
Last used within the last 5 months
Expired on Sat, Jul 1 2023.

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

☑ audit_log Full control of audit log
☑ read:audit_log Read access of audit log

Figure 6: Getting audit logs.

By default, *GitHub* does not display the source IP address for events in your organization's audit log.

GitHub displays an IP address for each event in the organization audit log that meets the following criteria:

- The actor is an organization member or owner
- The target is either an organization-owned repository that is private or internal, or an organization resource that is not a repository, such as a project.

Query logs from the API

1. Example for enterprises:

```
curl --include -H "Authorization: Bearer {TOKEN}" \ --request GET "https://api.github.com/
enterprises/{your-enterprise}/audit-log??include=all&per_page=50"
```

2. Example for organizations:

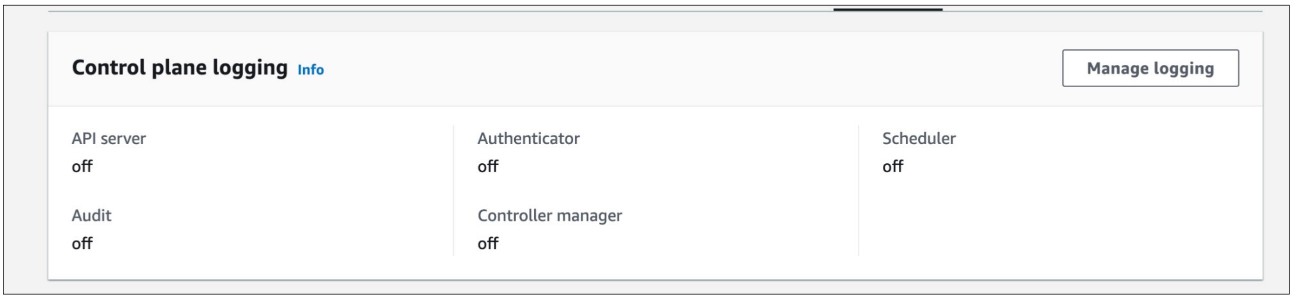
```
curl --include -H "Authorization: Bearer {TOKEN}" \ --request GET "https://api.github.com/
orgs/{your-organization}/audit-log??include=all&per_page=50"
```

The '?include=all' is important to get every request, like Clones for example, which are not shown in the web interface.

Kubernetes technical concepts

In a *Kubernetes* environment, logs refer to the recorded events and messages generated by various components and applications running within the cluster. *Kubernetes* provides a centralized logging mechanism that allows you to collect, store, and analyse logs from different sources, including pods, containers, and control plane components. Logs are essential for understanding the behaviour and performance of your applications, diagnosing issues, and monitoring the health of your *Kubernetes* infrastructure.

1. **Container logs**: these logs contain the output and error messages generated by individual containers running within a pod. They provide insights into the application's behaviour, including status, events, and any issues encountered by the container.
2. **Kube-apiserver logs**: the kube-apiserver logs record activities and events related to the *Kubernetes* API server. These logs are crucial for monitoring API requests, authentication, authorization, and any errors or warnings related to the API server's functionality.
3. **Kube-controller-manager logs**: the kube-controller-manager logs capture information about the *Kubernetes* controller manager. These logs provide details on various controllers, including node controller, replication controller, endpoint controller, and others. They help monitor the behaviour and health of controller processes.
4. **Kube-scheduler logs**: the kube-scheduler logs contain information about the *Kubernetes* scheduler, which assigns pods to nodes based on resource requirements, affinity rules, and other constraints. These logs provide insights into the scheduling decisions made by the scheduler.
5. **Kube-proxy logs**: the kube-proxy logs capture events and activities related to the *Kubernetes* network proxy running on each node. These logs provide details on network routing, load balancing, and any errors or warnings related to the proxy's operation.
6. **Ingress controller logs**: if you are using an Ingress controller for managing external access to your cluster, the logs from the Ingress controller provide insights into the routing, load balancing, and SSL termination processes for incoming traffic.
7. **Application logs**: these logs are specific to your applications running within *Kubernetes* pods. They capture application-specific events, errors, and informational messages. Application logs are vital for monitoring application behaviour, diagnosing issues, and troubleshooting application-level problems.



API server
off

Authenticator
off

Scheduler
off

Audit
off

Controller manager
off

Figure 7: EKS logging configuration.

DREDGE

Dredge is a tool designed to identify and respond quickly to attacks in cloud environments, particularly when one is not adequately prepared.

With Dredge, you can quickly gather logs from cloud providers and SaaS services such as *AWS*, *Azure*, *GitHub*, etc. It is intended to abstract forensic analysis from the specific technical knowledge of cloud environments, allowing for a rapid response in the event of an attack.

It is also equipped with a set of detection rules that enable the analysis of collected events in search of TTPs (tactics, techniques and procedures) or IOCs (Indicators of Compromise) in a practical and user-friendly manner.

Features

Key features

- Python-based CLI
- Retrieve logs seamlessly from *GitHub*, *Kubernetes*, *AWS*, *GCP* or *Azure*.
- Take action: whether it's blocking an IP in an *AWS* tenant, disabling an AccessKey, isolating an EC2 instance, or strategically extracting crucial post-compromise user data.
- Identify tactical misconfigurations that can be exploited by an attacker.
- Create an attack timeline based on IOCs.
- Analyse retrieved data effortlessly within a terminal, utilizing built-in capabilities from *VirusTotal* and *Shodan*.
- Cloud incident response guidelines for companies to embrace and build their playbooks.

Log collection features

- *AWS* - EventHistory
- *AWS* - GuardDuty
- *AWS* - CloudTrail (S3)
- *AWS* - VPC Flow logs (S3)
- *AWS* - Load Balancer (S3)
- *AWS* - WAF logs (S3)
- *GitHub* - Audit logs
- *Kubernetes* - Logs
- *Kubernetes* - Pod logs

Cloud status features

- *AWS* - IAM user list
- *AWS* - IAM access keys
- *AWS* - Lambda functions
- *AWS* - EC2 instance data
- *AWS* - RDS
- *AWS* - EKS
- *AWS* - S3 buckets, public buckets and public objects
- *GCP* - API logs
- *GitHub* - Audit logs

Threat hunting features

- IOCs search
- Custom rules creation
- *Shodan* integration
- *VirusTotal* integration
- *AWS* API call timeline creation
- *AWS* threat hunt (IP, IAM User or Access Key Id)
- Dangerous *AWS* API calls hunt

Incident response features

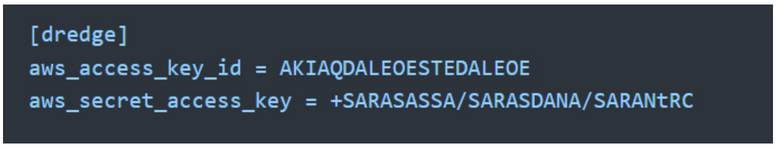
- *AWS* - Delete IAM user
- *AWS* - Disable AccessKey
- *AWS* - Remove logging profile

- *AWS* - Network isolate EC2 instance
- *AWS* - Get forensic image from EC2 instance volume
- *AWS* - Get Lambda env vars
- *AWS* - Make a bucket private
- *AWS* - Make an object private

Setup

Installation

1. Clone the repo
2. Install python3 requirements
   a. `pip3 install -r requirements.txt`
3. Use cloud provider credentials



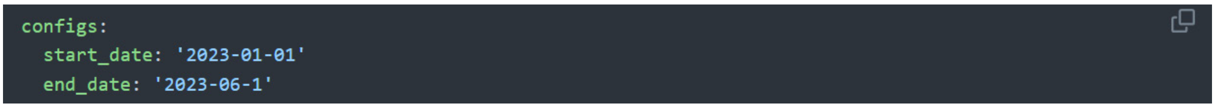
```
[dredge]
aws_access_key_id = AKIAQDALEOESTEDALEOE
aws_secret_access_key = +SARASASSA/SARASDANA/SARANtRC
```

Figure 8: Example using AWS credentials.

4. Start
   a. `python3 dredge.py --help`

Setting up config file

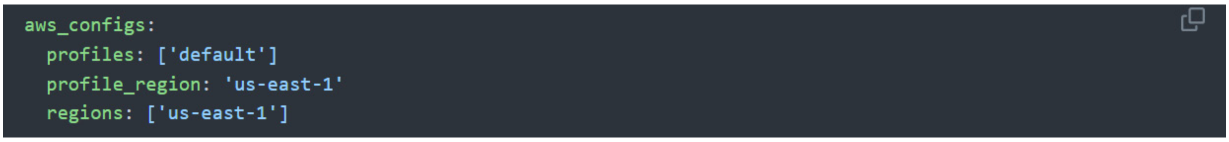
1. Specify dates, keeping in mind that EventHistory logs can take a long time to retrieve. Try to be specific.



```
configs:
  start_date: '2023-01-01'
  end_date: '2023-06-1'
```

Figure 9: Specify dates.

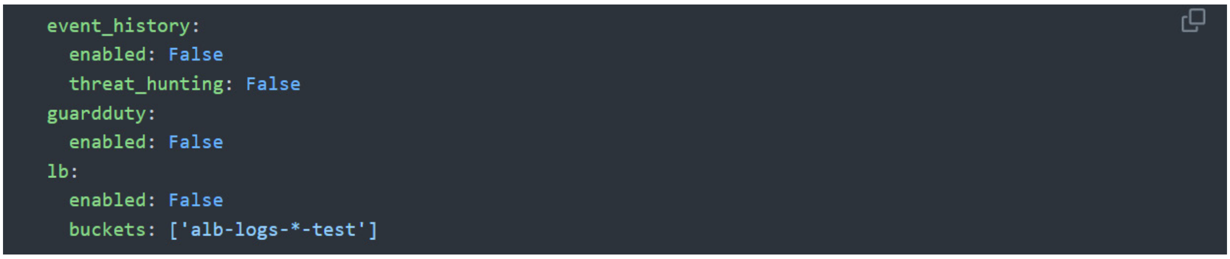
2. Define the *AWS* configs:
   a. Profile_region is the profile for *AWS* authentication.
   b. Regions are those needed for log retrieval if a multi-region strategy is in place.
   c. You can specify multiple profiles to get logs from different accounts.



```
aws_configs:
  profiles: ['default']
  profile_region: 'us-east-1'
  regions: ['us-east-1']
```

Figure 10: Define the AWS configs.

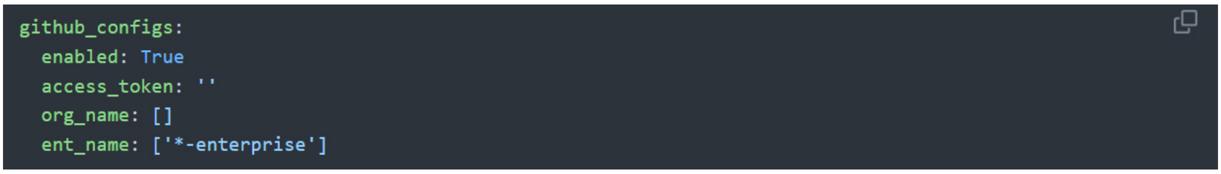
3. Configure the config file.
   a. Set 'enabled: True' for the log sources you want to analyse.
   b. For logs stored in S3 buckets (LB | WAF | VPC | CLOUDTRAIL) you must specify the bucket name.



```
  event_history:
    enabled: False
    threat_hunting: False
  guardduty:
    enabled: False
  lb:
    enabled: False
    buckets: ['alb-logs-*-test']
```

Figure 11: Configure the config file.

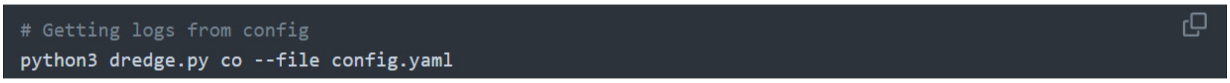
4. For *GitHub* logs, you need to specify:
    a. Organization or enterprise name
    b. Access token
    c. Set enabled true



```
github_configs:
  enabled: True
  access_token: ''
  org_name: []
  ent_name: ['*-enterprise']
```

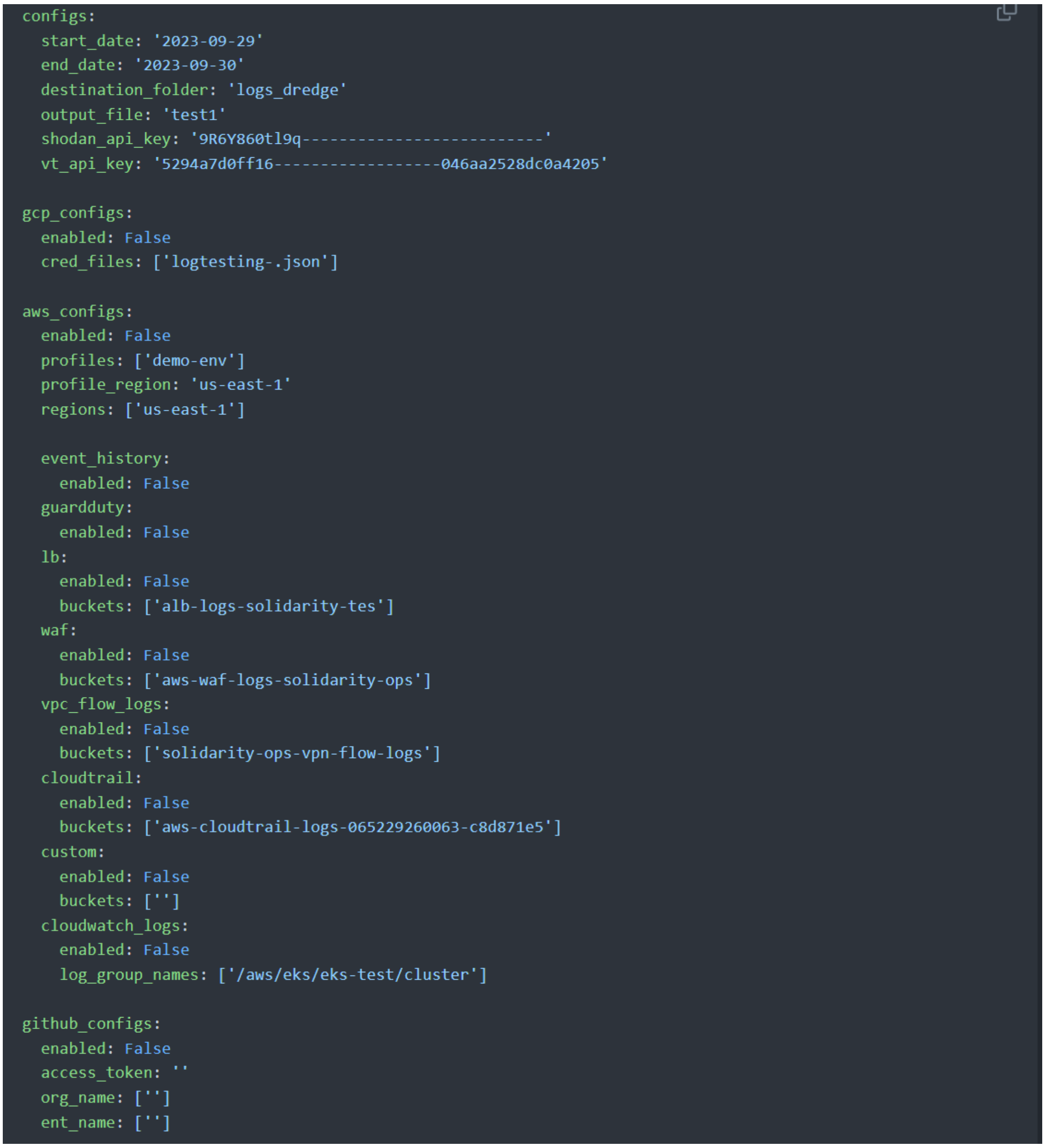
Figure 12: Configure the config file for GitHub logs.

5. Execution



```
# Getting logs from config
python3 dredge.py co --file config.yaml
```

Figure 13: Execution.



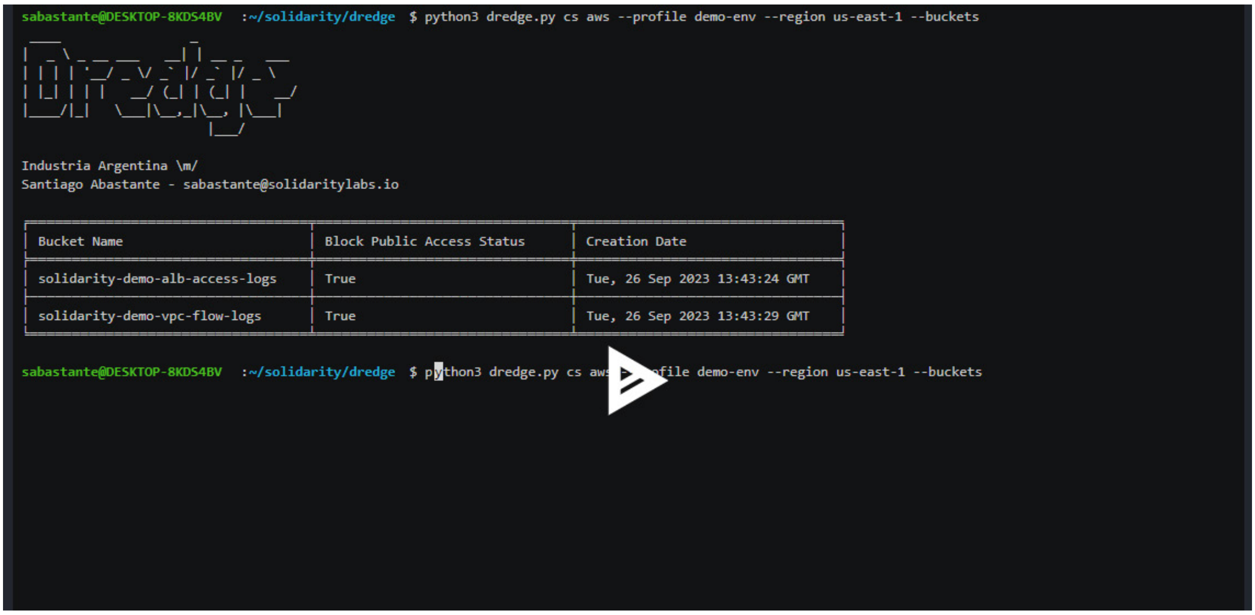
```
configs:
  start_date: '2023-09-29'
  end_date: '2023-09-30'
  destination_folder: 'logs_dredge'
  output_file: 'test1'
  shodan_api_key: '9R6Y860tl9q-------------------------'
  vt_api_key: '5294a7d0ff16------------------046aa2528dc0a4205'

gcp_configs:
  enabled: False
  cred_files: ['logtesting-.json']

aws_configs:
  enabled: False
  profiles: ['demo-env']
  profile_region: 'us-east-1'
  regions: ['us-east-1']

  event_history:
    enabled: False
  guardduty:
    enabled: False
  lb:
    enabled: False
    buckets: ['alb-logs-solidarity-tes']
  waf:
    enabled: False
    buckets: ['aws-waf-logs-solidarity-ops']
  vpc_flow_logs:
    enabled: False
    buckets: ['solidarity-ops-vpn-flow-logs']
  cloudtrail:
    enabled: False
    buckets: ['aws-cloudtrail-logs-065229260063-c8d871e5']
  custom:
    enabled: False
    buckets: ['']
  cloudwatch_logs:
    enabled: False
    log_group_names: ['/aws/eks/eks-test/cluster']

github_configs:
  enabled: False
  access_token: ''
  org_name: ['']
  ent_name: ['']
```

Figure 14: Full config file example.

Usage examples

The following figures illustrate some Dredge usage examples.

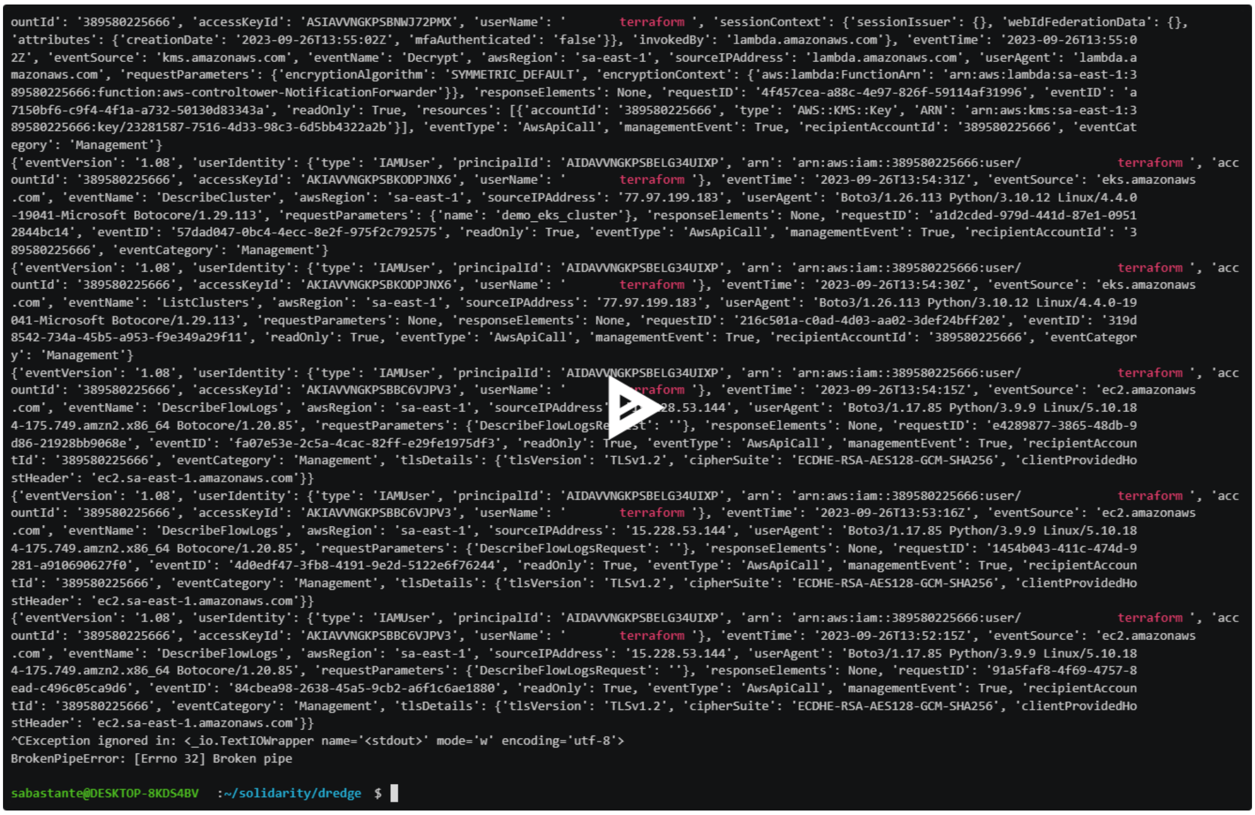
```
# Getting S3 buckets
python3 dredge.py cs aws --profile <demo-env> --region <us-east-1> --buckets
```



```
sabastante@DESKTOP-8KDS4BV   :~/solidarity/dredge  $ python3 dredge.py cs aws --profile demo-env --region us-east-1 --buckets
 ____                _
|  _ \ _ __ ___  __| | __ _  ___
| | | | '__/ _ \/ _` |/ _` |/ _ \
| |_| | | |  __/ (_| | (_| |  __/
|____/|_|  \___|\__,_|\__, |\___|
                      |___/

Industria Argentina \m/
Santiago Abastante - sabastante@solidaritylabs.io

+---------------------------------+-------------------------------+----------------------------------+
| Bucket Name                     | Block Public Access Status    | Creation Date                    |
+=================================+===============================+==================================+
| solidarity-demo-alb-access-logs | True                          | Tue, 26 Sep 2023 13:43:24 GMT    |
+---------------------------------+-------------------------------+----------------------------------+
| solidarity-demo-vpc-flow-logs   | True                          | Tue, 26 Sep 2023 13:43:29 GMT    |
+---------------------------------+-------------------------------+----------------------------------+

sabastante@DESKTOP-8KDS4BV   :~/solidarity/dredge  $ python3 dredge.py cs aws    ofile demo-env --region us-east-1 --buckets
```

Figure 15: Cloud status list S3 buckets.

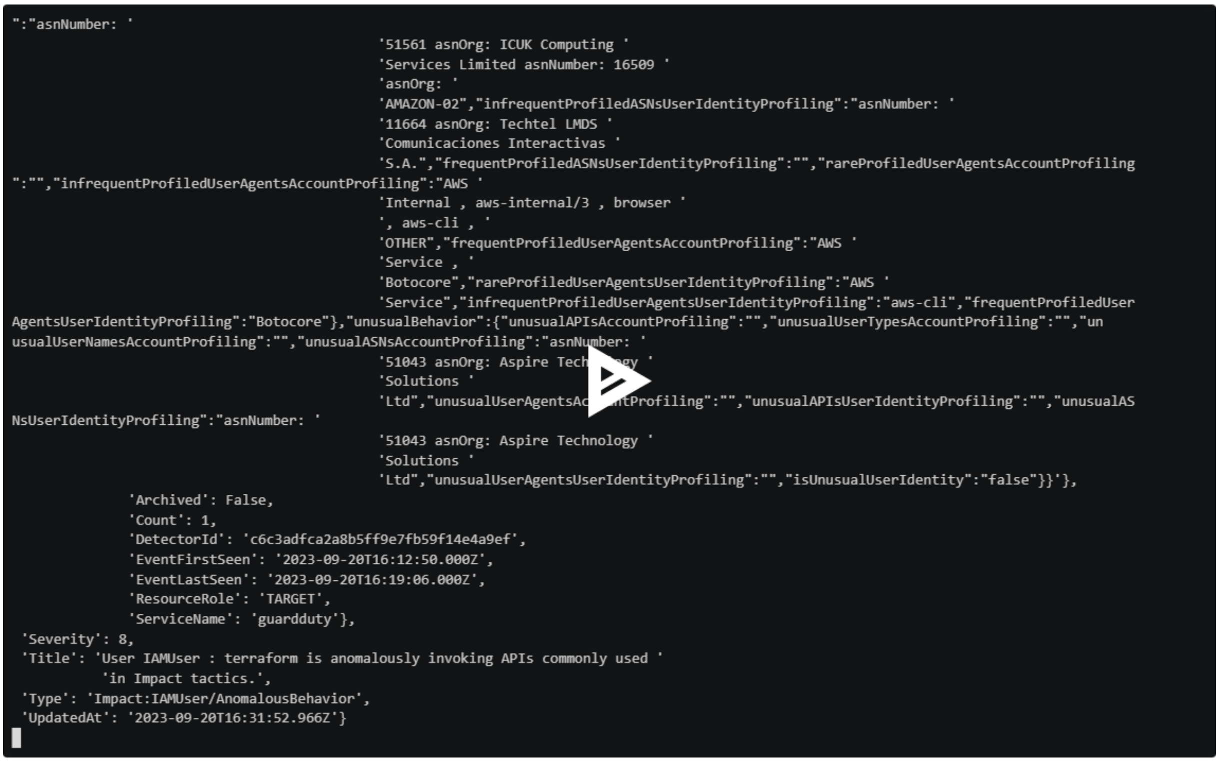
```
# Getting Event History Logs
python3 dredge.py lr aws --profile <demo-env> --region <sa-east-1> --log event_history
```



```
ountId': '389580225666', 'accessKeyId': 'ASIAVVNGKPSBNWJ72PMX', 'userName': '       terraform ', 'sessionContext': {'sessionIssuer': {}, 'webIdFederationData': {},
'attributes': {'creationDate': '2023-09-26T13:55:02Z', 'mfaAuthenticated': 'false'}}, 'invokedBy': 'lambda.amazonaws.com'}, 'eventTime': '2023-09-26T13:55:0
2Z', 'eventSource': 'kms.amazonaws.com', 'eventName': 'Decrypt', 'awsRegion': 'sa-east-1', 'sourceIPAddress': 'lambda.amazonaws.com', 'userAgent': 'lambda.a
mazonaws.com', 'requestParameters': {'encryptionAlgorithm': 'SYMMETRIC_DEFAULT', 'encryptionContext': {'aws:lambda:FunctionArn': 'arn:aws:lambda:sa-east-1:3
89580225666:function:aws-controltower-NotificationForwarder'}}, 'responseElements': None, 'requestID': '4f457cea-a88c-4e97-826f-59114af31996', 'eventID': 'a
7150bf6-c9f4-4f1a-a732-50130d83343a', 'readOnly': True, 'resources': [{'accountId': '389580225666', 'type': 'AWS::KMS::Key', 'ARN': 'arn:aws:kms:sa-east-1:3
89580225666:key/23281587-7516-4d33-98c3-6d5bb4322a2b'}], 'eventType': 'AwsApiCall', 'managementEvent': True, 'recipientAccountId': '389580225666', 'eventCat
egory': 'Management'}
{'eventVersion': '1.08', 'userIdentity': {'type': 'IAMUser', 'principalId': 'AIDAVVNGKPSBELG34UIXP', 'arn': 'arn:aws:iam::389580225666:user/           terraform ', 'acc
ountId': '389580225666', 'accessKeyId': 'AKIAVVNGKPSBKODPJNX6', 'userName': '       terraform '}, 'eventTime': '2023-09-26T13:54:31Z', 'eventSource': 'eks.amazonaws
.com', 'eventName': 'DescribeCluster', 'awsRegion': 'sa-east-1', 'sourceIPAddress': '77.97.199.183', 'userAgent': 'Boto3/1.26.113 Python/3.10.12 Linux/4.4.0
-19041-Microsoft Botocore/1.29.113', 'requestParameters': {'name': 'demo_eks_cluster'}, 'responseElements': None, 'requestID': 'a1d2cded-979d-441d-87e1-0951
2844bc14', 'eventID': '57dad047-0bc4-4ecc-8e2f-975f2c792575', 'readOnly': True, 'eventType': 'AwsApiCall', 'managementEvent': True, 'recipientAccountId': '3
89580225666', 'eventCategory': 'Management'}
{'eventVersion': '1.08', 'userIdentity': {'type': 'IAMUser', 'principalId': 'AIDAVVNGKPSBELG34UIXP', 'arn': 'arn:aws:iam::389580225666:user/           terraform ', 'acc
ountId': '389580225666', 'accessKeyId': 'AKIAVVNGKPSBKODPJNX6', 'userName': '       terraform '}, 'eventTime': '2023-09-26T13:54:30Z', 'eventSource': 'eks.amazonaws
.com', 'eventName': 'ListClusters', 'awsRegion': 'sa-east-1', 'sourceIPAddress': '77.97.199.183', 'userAgent': 'Boto3/1.26.113 Python/3.10.12 Linux/4.4.0-19
041-Microsoft Botocore/1.29.113', 'requestParameters': None, 'responseElements': None, 'requestID': '216c501a-c0ad-4d03-aa02-3def24bff202', 'eventID': '319d
8542-734a-45b5-a953-f9e349a29f11', 'readOnly': True, 'eventType': 'AwsApiCall', 'managementEvent': True, 'recipientAccountId': '389580225666', 'eventCategor
y': 'Management'}
{'eventVersion': '1.08', 'userIdentity': {'type': 'IAMUser', 'principalId': 'AIDAVVNGKPSBELG34UIXP', 'arn': 'arn:aws:iam::389580225666:user/           terraform ', 'acc
ountId': '389580225666', 'accessKeyId': 'AKIAVVNGKPSBBC6VJPV3', 'userName': '        raform '}, 'eventTime': '2023-09-26T13:54:15Z', 'eventSource': 'ec2.amazonaws
.com', 'eventName': 'DescribeFlowLogs', 'awsRegion': 'sa-east-1', 'sourceIPAddress'   28.53.144', 'userAgent': 'Boto3/1.17.85 Python/3.9.9 Linux/5.10.18
4-175.749.amzn2.x86_64 Botocore/1.20.85', 'requestParameters': {'DescribeFlowLogsRe   st': ''}, 'responseElements': None, 'requestID': 'e4289877-3865-48db-9
d86-21928bb9068e', 'eventID': 'fa07e53e-2c5a-4cac-82ff-e29fe1975df3', 'readOnly': True, 'eventType': 'AwsApiCall', 'managementEvent': True, 'recipientAccoun
tId': '389580225666', 'eventCategory': 'Management', 'tlsDetails': {'tlsVersion': 'TLSv1.2', 'cipherSuite': 'ECDHE-RSA-AES128-GCM-SHA256', 'clientProvidedHo
stHeader': 'ec2.sa-east-1.amazonaws.com'}}
{'eventVersion': '1.08', 'userIdentity': {'type': 'IAMUser', 'principalId': 'AIDAVVNGKPSBELG34UIXP', 'arn': 'arn:aws:iam::389580225666:user/           terraform ', 'acc
ountId': '389580225666', 'accessKeyId': 'AKIAVVNGKPSBBC6VJPV3', 'userName': '       terraform '}, 'eventTime': '2023-09-26T13:53:16Z', 'eventSource': 'ec2.amazonaws
.com', 'eventName': 'DescribeFlowLogs', 'awsRegion': 'sa-east-1', 'sourceIPAddress': '15.228.53.144', 'userAgent': 'Boto3/1.17.85 Python/3.9.9 Linux/5.10.18
4-175.749.amzn2.x86_64 Botocore/1.20.85', 'requestParameters': {'DescribeFlowLogsRequest': ''}, 'responseElements': None, 'requestID': '1454b043-411c-474d-9
281-a910690627f0', 'eventID': '4d0edf47-3fb8-4191-9e2d-5122e6f76244', 'readOnly': True, 'eventType': 'AwsApiCall', 'managementEvent': True, 'recipientAccoun
tId': '389580225666', 'eventCategory': 'Management', 'tlsDetails': {'tlsVersion': 'TLSv1.2', 'cipherSuite': 'ECDHE-RSA-AES128-GCM-SHA256', 'clientProvidedHo
stHeader': 'ec2.sa-east-1.amazonaws.com'}}
{'eventVersion': '1.08', 'userIdentity': {'type': 'IAMUser', 'principalId': 'AIDAVVNGKPSBELG34UIXP', 'arn': 'arn:aws:iam::389580225666:user/           terraform ', 'acc
ountId': '389580225666', 'accessKeyId': 'AKIAVVNGKPSBBC6VJPV3', 'userName': '       terraform '}, 'eventTime': '2023-09-26T13:52:15Z', 'eventSource': 'ec2.amazonaws
.com', 'eventName': 'DescribeFlowLogs', 'awsRegion': 'sa-east-1', 'sourceIPAddress': '15.228.53.144', 'userAgent': 'Boto3/1.17.85 Python/3.9.9 Linux/5.10.18
4-175.749.amzn2.x86_64 Botocore/1.20.85', 'requestParameters': {'DescribeFlowLogsRequest': ''}, 'responseElements': None, 'requestID': '91a5faf8-4f69-4757-8
ead-c496c05ca9d6', 'eventID': '84cbea98-2638-45a5-9cb2-a6f1c6ae1880', 'readOnly': True, 'eventType': 'AwsApiCall', 'managementEvent': True, 'recipientAccoun
tId': '389580225666', 'eventCategory': 'Management', 'tlsDetails': {'tlsVersion': 'TLSv1.2', 'cipherSuite': 'ECDHE-RSA-AES128-GCM-SHA256', 'clientProvidedHo
stHeader': 'ec2.sa-east-1.amazonaws.com'}}
^CException ignored in: <_io.TextIOWrapper name='<stdout>' mode='w' encoding='utf-8'>
BrokenPipeError: [Errno 32] Broken pipe

sabastante@DESKTOP-8KDS4BV   :~/solidarity/dredge  $ 
```

Figure 16: Log collection get Event History logs.

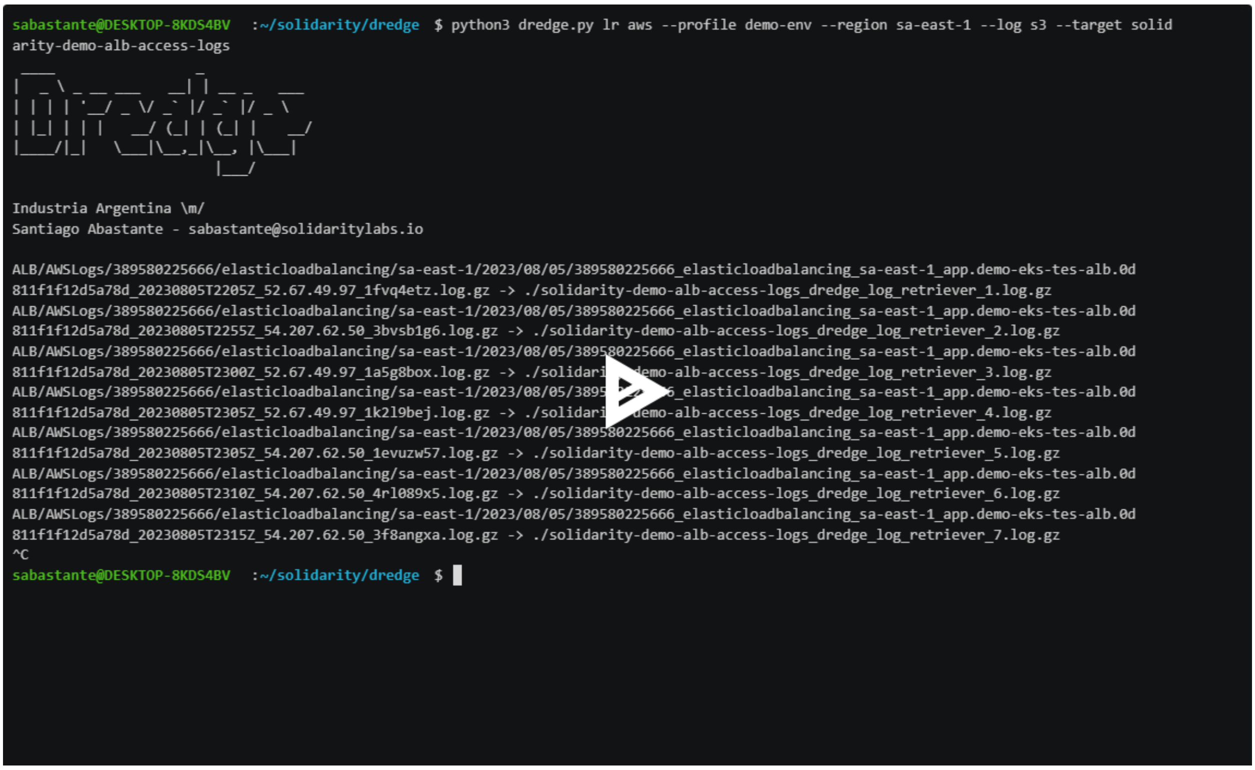
```
# Getting Guardduty Logs
python3 dredge.py lr aws --profile <demo-env> --region <sa-east-1> --log guardduty
```



```
":"asnNumber: '
                                                 '51561 asnOrg: ICUK Computing '
                                                 'Services Limited asnNumber: 16509 '
                                                 'asnOrg: '
                                                 'AMAZON-02","infrequentProfiledASNsUserIdentityProfiling":"asnNumber: '
                                                 '11664 asnOrg: Techtel LMDS '
                                                 'Comunicaciones Interactivas '
                                                 'S.A.","frequentProfiledASNsUserIdentityProfiling":"","rareProfiledUserAgentsAccountProfiling
":"","infrequentProfiledUserAgentsAccountProfiling":"AWS '
                                                 'Internal , aws-internal/3 , browser '
                                                 ', aws-cli , '
                                                 'OTHER","frequentProfiledUserAgentsAccountProfiling":"AWS '
                                                 'Service , '
                                                 'Botocore","rareProfiledUserAgentsUserIdentityProfiling":"AWS '
                                                 'Service","infrequentProfiledUserAgentsUserIdentityProfiling":"aws-cli","frequentProfiledUser
AgentsUserIdentityProfiling":"Botocore"},"unusualBehavior":{"unusualAPIsAccountProfiling":"","unusualUserTypesAccountProfiling":"","un
usualUserNamesAccountProfiling":"","unusualASNsAccountProfiling":"asnNumber: '
                                                 '51043 asnOrg: Aspire Technology '
                                                 'Solutions '
                                                 'Ltd","unusualUserAgentsAccountProfiling":"","unusualAPIsUserIdentityProfiling":"","unusualAS
NsUserIdentityProfiling":"asnNumber: '
                                                 '51043 asnOrg: Aspire Technology '
                                                 'Solutions '
                                                 'Ltd","unusualUserAgentsUserIdentityProfiling":"","isUnusualUserIdentity":"false"}}'},
              'Archived': False,
              'Count': 1,
              'DetectorId': 'c6c3adfca2a8b5ff9e7fb59f14e4a9ef',
              'EventFirstSeen': '2023-09-20T16:12:50.000Z',
              'EventLastSeen': '2023-09-20T16:19:06.000Z',
              'ResourceRole': 'TARGET',
              'ServiceName': 'guardduty'},
 'Severity': 8,
 'Title': 'User IAMUser : terraform is anomalously invoking APIs commonly used '
          'in Impact tactics.',
 'Type': 'Impact:IAMUser/AnomalousBehavior',
 'UpdatedAt': '2023-09-20T16:31:52.966Z'}
```

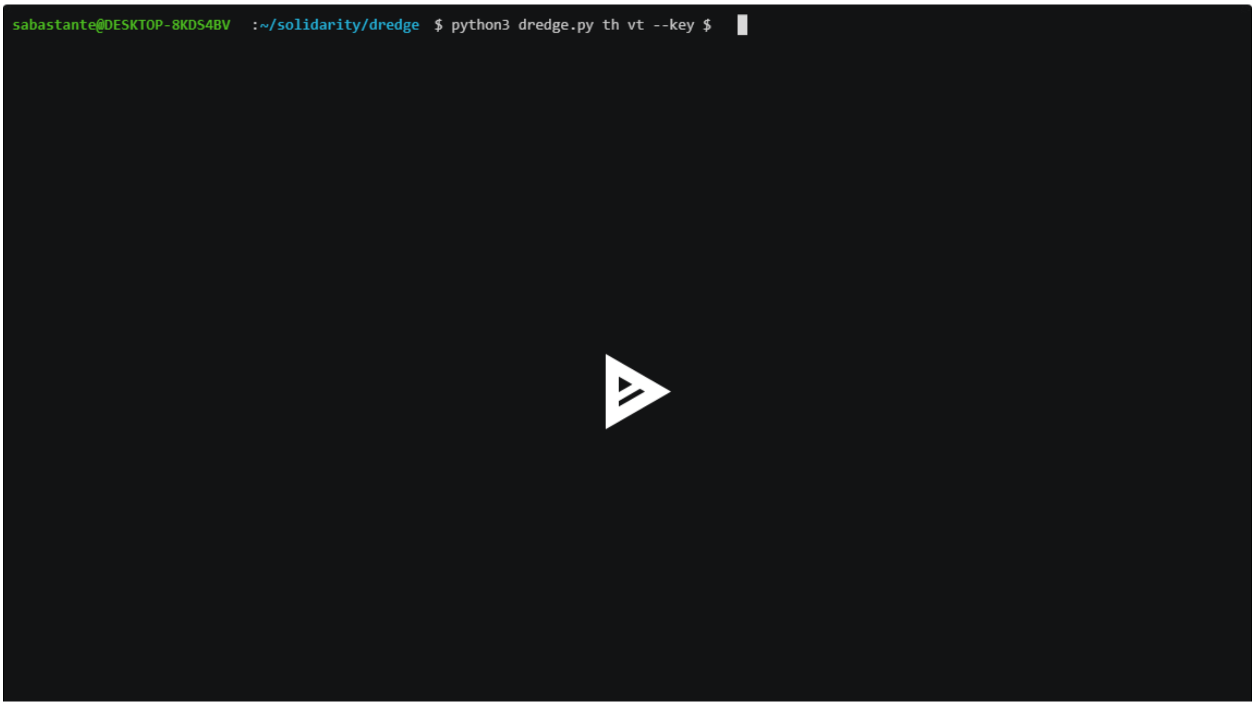
Figure 17: Get GuardDuty events.

```
python3 dredge.py lr aws --profile <demo-env> --region <sa-east-1> --log s3 --target
<solidarity-demo-alb-access-logs>
```



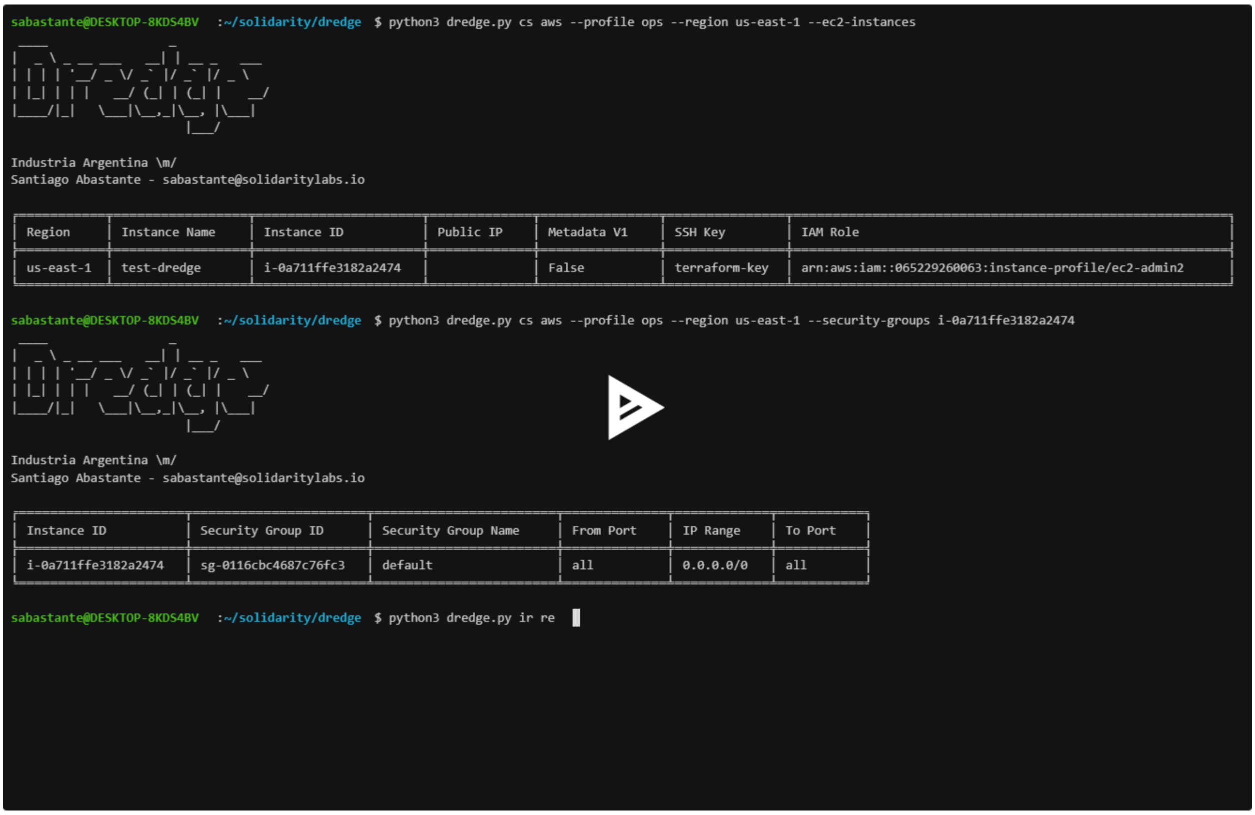
```
sabastante@DESKTOP-8KDS4BV  :~/solidarity/dredge  $ python3 dredge.py lr aws --profile demo-env --region sa-east-1 --log s3 --target solid
arity-demo-alb-access-logs
 ____              _
|  _ \ _ __ ___  __| | __ _  ___
| | | | '__/ _ \/ _` |/ _` |/ _ \
| |_| | | |  __/ (_| | (_| |  __/
|____/|_|  \___|\__,_|\__, |\___|
                      |___/

Industria Argentina \m/
Santiago Abastante - sabastante@solidaritylabs.io

ALB/AWSLogs/389580225666/elasticloadbalancing/sa-east-1/2023/08/05/389580225666_elasticloadbalancing_sa-east-1_app.demo-eks-tes-alb.0d
811f1f12d5a78d_20230805T2205Z_52.67.49.97_1fvq4etz.log.gz -> ./solidarity-demo-alb-access-logs_dredge_log_retriever_1.log.gz
ALB/AWSLogs/389580225666/elasticloadbalancing/sa-east-1/2023/08/05/389580225666_elasticloadbalancing_sa-east-1_app.demo-eks-tes-alb.0d
811f1f12d5a78d_20230805T2255Z_54.207.62.50_3bvsb1g6.log.gz -> ./solidarity-demo-alb-access-logs_dredge_log_retriever_2.log.gz
ALB/AWSLogs/389580225666/elasticloadbalancing/sa-east-1/2023/08/05/389580225666_elasticloadbalancing_sa-east-1_app.demo-eks-tes-alb.0d
811f1f12d5a78d_20230805T2300Z_52.67.49.97_1a5g8box.log.gz -> ./solidarity-demo-alb-access-logs_dredge_log_retriever_3.log.gz
ALB/AWSLogs/389580225666/elasticloadbalancing/sa-east-1/2023/08/05/389580225666_elasticloadbalancing_sa-east-1_app.demo-eks-tes-alb.0d
811f1f12d5a78d_20230805T2305Z_52.67.49.97_1k2l9bej.log.gz -> ./solidarity-demo-alb-access-logs_dredge_log_retriever_4.log.gz
ALB/AWSLogs/389580225666/elasticloadbalancing/sa-east-1/2023/08/05/389580225666_elasticloadbalancing_sa-east-1_app.demo-eks-tes-alb.0d
811f1f12d5a78d_20230805T2305Z_54.207.62.50_1evuzw57.log.gz -> ./solidarity-demo-alb-access-logs_dredge_log_retriever_5.log.gz
ALB/AWSLogs/389580225666/elasticloadbalancing/sa-east-1/2023/08/05/389580225666_elasticloadbalancing_sa-east-1_app.demo-eks-tes-alb.0d
811f1f12d5a78d_20230805T2310Z_54.207.62.50_4rl089x5.log.gz -> ./solidarity-demo-alb-access-logs_dredge_log_retriever_6.log.gz
ALB/AWSLogs/389580225666/elasticloadbalancing/sa-east-1/2023/08/05/389580225666_elasticloadbalancing_sa-east-1_app.demo-eks-tes-alb.0d
811f1f12d5a78d_20230805T2315Z_54.207.62.50_3f8angxa.log.gz -> ./solidarity-demo-alb-access-logs_dredge_log_retriever_7.log.gz
^C
sabastante@DESKTOP-8KDS4BV  :~/solidarity/dredge  $
```

Figure 18: Log retriever get logs from S3 bucket.



```
sabastante@DESKTOP-8KDS4BV  :~/solidarity/dredge  $ python3 dredge.py th vt --key $
```

Figure 19: Threat hunting analyse with VirusTotal.



```
sabastante@DESKTOP-8KDS4BV  :~/solidarity/dredge  $ python3 dredge.py cs aws --profile ops --region us-east-1 --ec2-instances
 ____               _
|  _ \ _ __ ___  __| | __ _  ___
| | | | '__/ _ \/ _` |/ _` |/ _ \
| |_| | | |  __/ (_| | (_| |  __/
|____/|_|  \___|\__,_|\__, |\___|
                      |___/

Industria Argentina \m/
Santiago Abastante - sabastante@solidaritylabs.io

| Region    | Instance Name | Instance ID         | Public IP | Metadata V1 | SSH Key       | IAM Role                                                |
| us-east-1 | test-dredge   | i-0a711ffe3182a2474 |           | False       | terraform-key | arn:aws:iam::065229260063:instance-profile/ec2-admin2   |

sabastante@DESKTOP-8KDS4BV  :~/solidarity/dredge  $ python3 dredge.py cs aws --profile ops --region us-east-1 --security-groups i-0a711ffe3182a2474
 ____               _
|  _ \ _ __ ___  __| | __ _  ___
| | | | '__/ _ \/ _` |/ _` |/ _ \
| |_| | | |  __/ (_| | (_| |  __/
|____/|_|  \___|\__,_|\__, |\___|
                      |___/

Industria Argentina \m/
Santiago Abastante - sabastante@solidaritylabs.io

| Instance ID         | Security Group ID    | Security Group Name | From Port | IP Range  | To Port |
| i-0a711ffe3182a2474 | sg-0116cbc4687c76fc3 | default             | all       | 0.0.0.0/0 | all     |

sabastante@DESKTOP-8KDS4BV  :~/solidarity/dredge  $ python3 dredge.py ir re
```

Figure 20: Incident response network isolate an EC2 instance.

REFERENCES

[1] solidarity-labs / dredge-mvp. https://github.com/solidarity-labs/dredge-mvp?tab=readme-ov-file.

[2] Solidarity Labs. Fantastic Logs (And Where to Find them) - <AWS Cloudtrail> (Part 1). https://www.notion.so/solidaritylabs/Fantastic-Logs-And-Where-to-Find-them-AWS-Cloudtrail-Part-1-0a2f403c4342439f91ac2837e5942935.

[3] Solidarity Labs. Fantastic Logs (And Where to Find them) - <Github>. https://www.notion.so/solidaritylabs/Fantastic-Logs-And-Where-to-Find-them-Github-1297f7872eed4c4da48115583ae85a86.

[4] Amazon. AWS Identity and Access Management Documentation. https://docs.aws.amazon.com/iam/index.html.

[5] Amazon. Amazon Elastic Kubernetes Service Documentation. https://docs.aws.amazon.com/eks/index.html.

[6] Amazon. Amazon Elastic Container Registry Documentation. https://docs.aws.amazon.com/ecr/index.html.

[7] Amazon. Amazon GuardDuty. https://docs.aws.amazon.com/guardduty/index.html.

[8] Amazon. AWS CloudTrail Documentation. https://docs.aws.amazon.com/cloudtrail/index.html.

[9] Amazon. Amazon CloudWatch Documentation. https://docs.aws.amazon.com/cloudwatch/index.html.

[10] Sysdig. Threat Detection Built on Falco. https://sysdig.com/opensource/falco/.

[11] MITRE ATT&CK. Cloud Matrix. https://attack.mitre.org/matrices/enterprise/cloud/.