aws RE:INFORCE

aws RE:INFORCE

SEP315

An open-source adventure in the cloud, containers, and incident response

Nathan Case
Security Geek
AWS

Andrew Krug
Staff Security Engineer
Mozilla

aws RE:INFORCE

SEP315

Additional Facilitators

Gene Wood
Senior Staff Security Engineer
Mozilla

Guillaume Destuynder
Senior Staff Security Engineer
Mozilla

Related breakouts

Day of Week, Month Day

Session Title

Time – Time | Location

Day of Week, Month Day

Session Title

Time – Time | Location

Day of Week, Month Day

Session Title

Time – Time | Location

Agenda – 120 minute workshop

Intro / Why OSS? (5-minutes)

Anatomy of a Healthy Security Operation (5-minutes)

Security Simulation Scenario Review (5-minutes)

Exploration and Alert Authoring (7-minutes)

Investigation (7-minutes)

Workshop Section 1

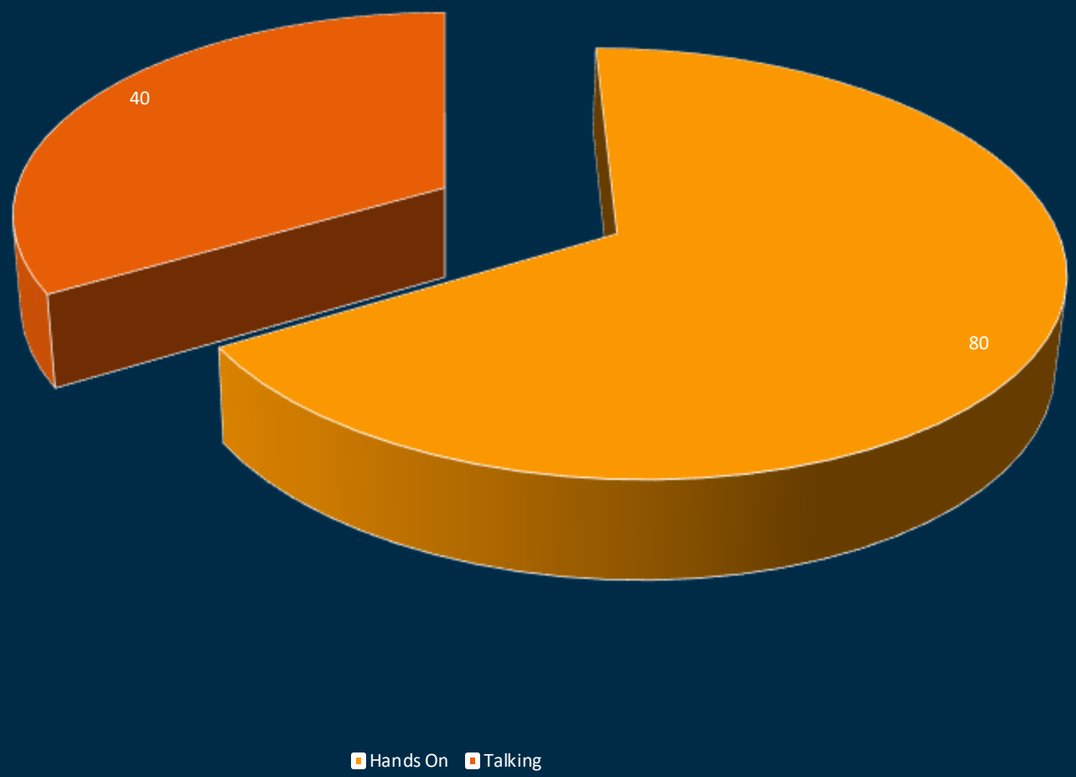
Agenda – 120 minute workshop

Responding to your findings aka Incident Response (10-minutes)

Workshop Section 2

Wrap up / Retrospective (2-minutes)

A workshop unlike any other



Introduction / Why OSS

aws RE:INFORCE

Workshop Introduction

What are we going to do?

Play through a security simulation in teams of two.

Use tools that Mozilla is releasing to the public in order to detect and respond to threats.

How are we going to do it?

Lecture format followed by open Q&A / Lab Time at intervals.

Ask questions! Everything is in scope so ask! (You have 4 facilitators)

What are our goals?

Learn to tell stories using a SIEM and your sleuthing skills to get to the truth!

Ultimately try to remove the threat from your environment.

Have an awesome time practicing the act of incident response:

“Doing your best job on your worst day.”

aws

Why Open Source

Keep the web a global public resource open and accessible to all.

For us, open means **transparent**, as in open source – you're not locked in to what the original creator did.

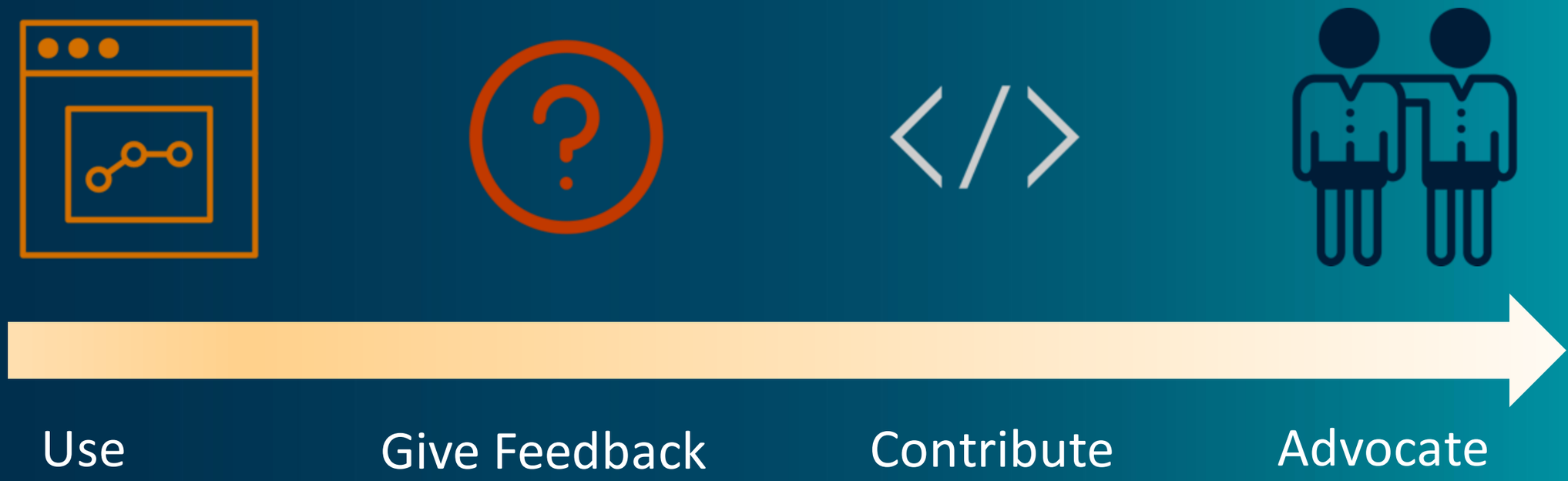
And in our case "**open**" also means distributed decision making.

Mitchell Baker

Chair, Mozilla

aws RE:INFORCE

Contribution is not just CODE!



Different Types of Participation

We don't compete on security.

Dan Kaminsky

aws RE:INFORCE

Anatomy of a Healthy Security Operation

aws RE:INFORCE

The make of the security team

There's no wrong way to "do security” – except not doing it.

- Embedded security engineers
- Dedicated security operations team
- Distributed security (engineers that *also* do security)
- ...the model needs to work for your business

Scope

Focus on the threat management and incident response

Intelligence gathering

Proactive defense

Incident handling

Tying it all together

So what's a SIEM?

SIEM: Security Information Event Management

- Log records (event data)
- Alerts
- Event data correlation
- Incident management
- Reports, dashboards, compliance
- OIC storage (indicators of compromise)

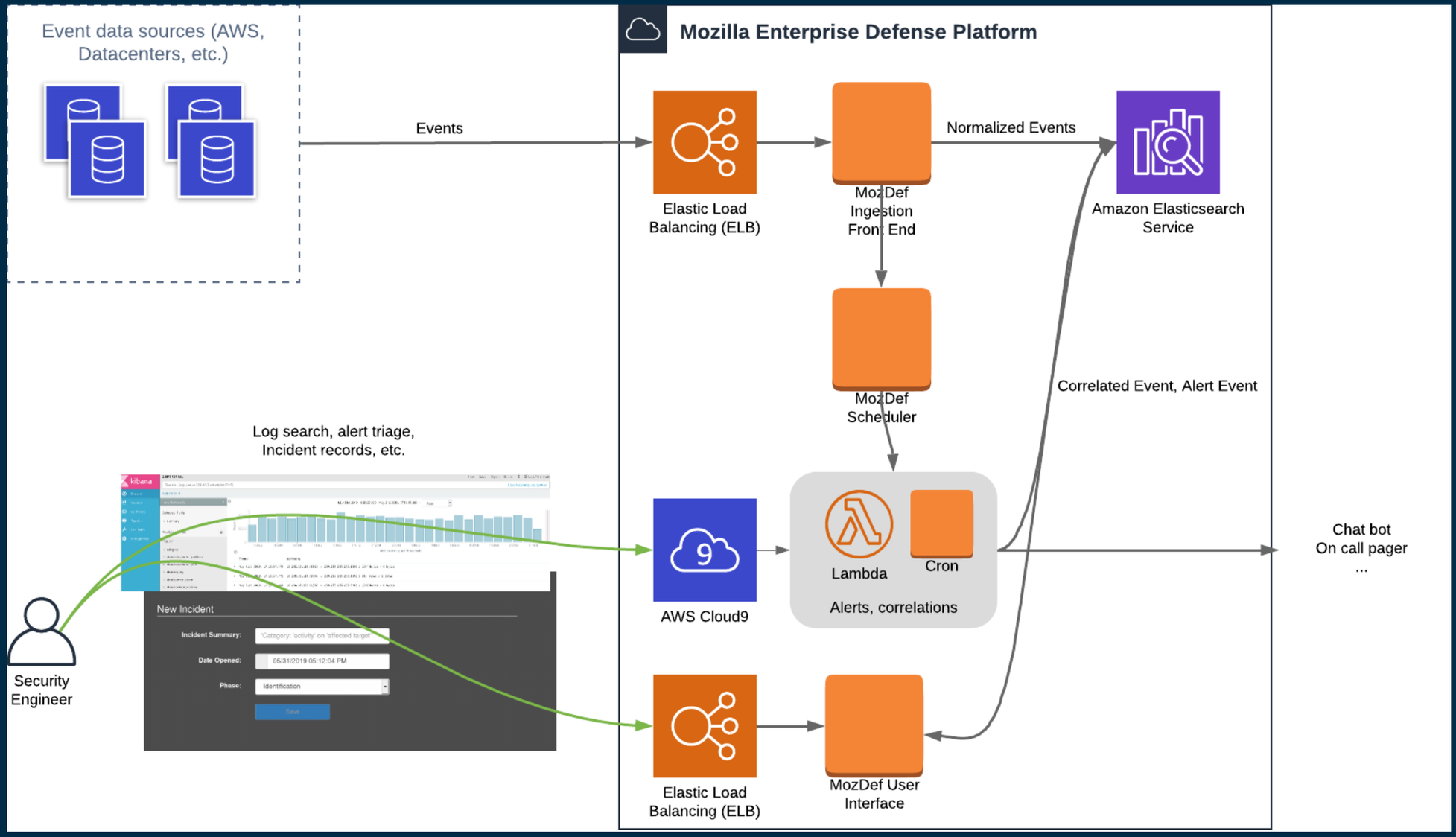
Mozilla Enterprise Defense Platform is an open-source SIEM

aws

Some things you'll want from your SIEM

- Know what you have *(instances, accounts, data ==> logs)*
- Understand your business *(what's most at risk?)*
- Be able to respond when there's a problem
- Understand your weaknesses *(most common incident type)*

Typical event pipeline



A note on Normalized Event Data

Event data comes from various sources in various formats

For alerting and correlation to be as easy and reliable as possible, typical data fields need to be normalized, e.g.:

Original event format	Normalized event
Ip: 1.2.3.4	sourceIpAddress: 1.2.3.4
User: banzai	details.username: banzai
...	...

See https://mozdef.readthedocs.io/en/latest/usage.html for details about the Mozilla Enterprise Defense Platform normalized event format

Typical incident workflow

A few months ago you wrote an alert matching on a Lambda function talking to a specific set of known-bad IP addresses...

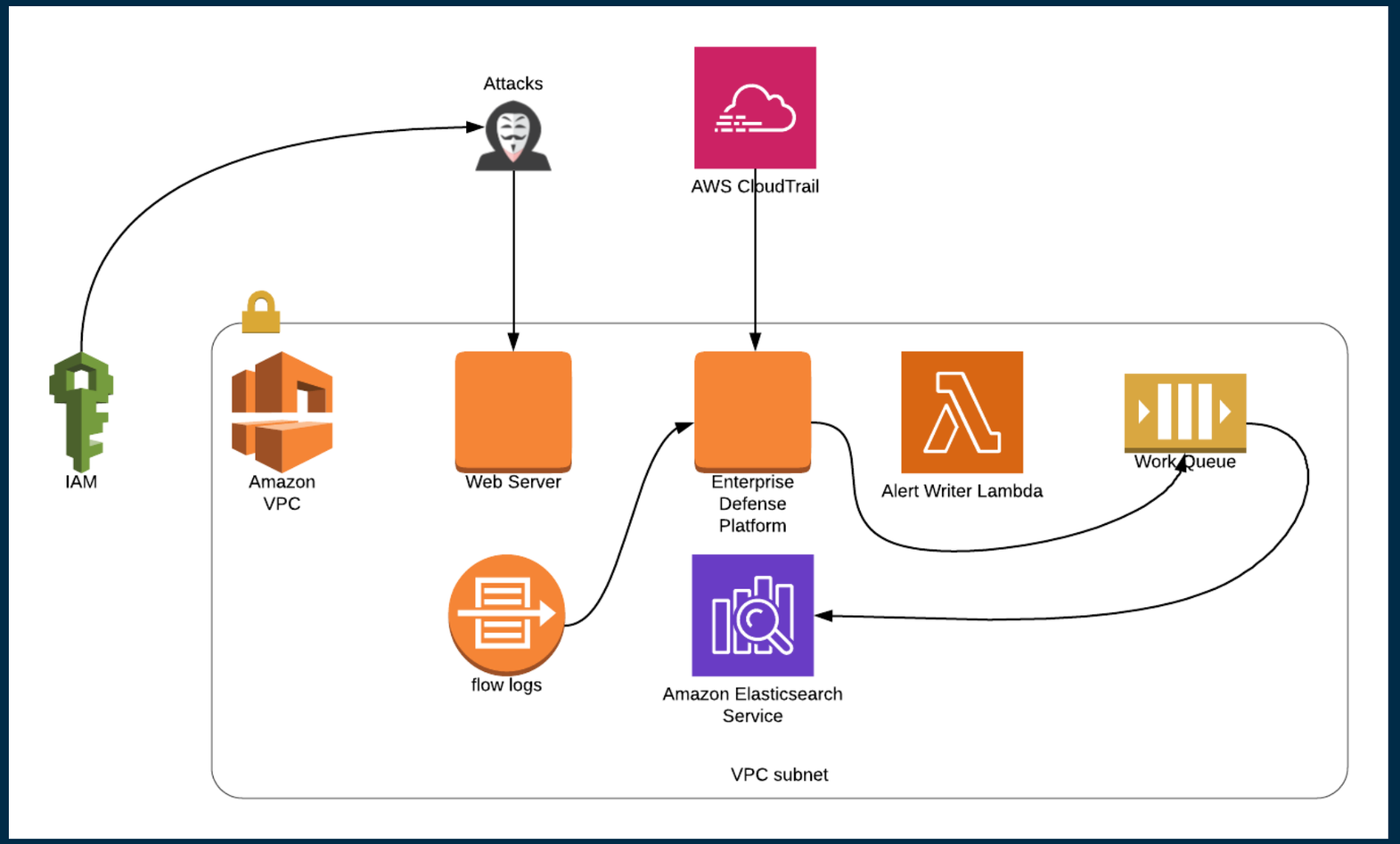
1. Identification / Open incident / Assemble team
2. Containment
3. Eradication
4. Recovery
5. Lesson learned
6. Close the incident

Record all actions as you progress

Security Simulation Overview

aws RE:INFORCE

Simulation Environment



Expected IOCs – Use the tools to find them

Excessive Describe for a Production Service

SSH Traffic from the Outside

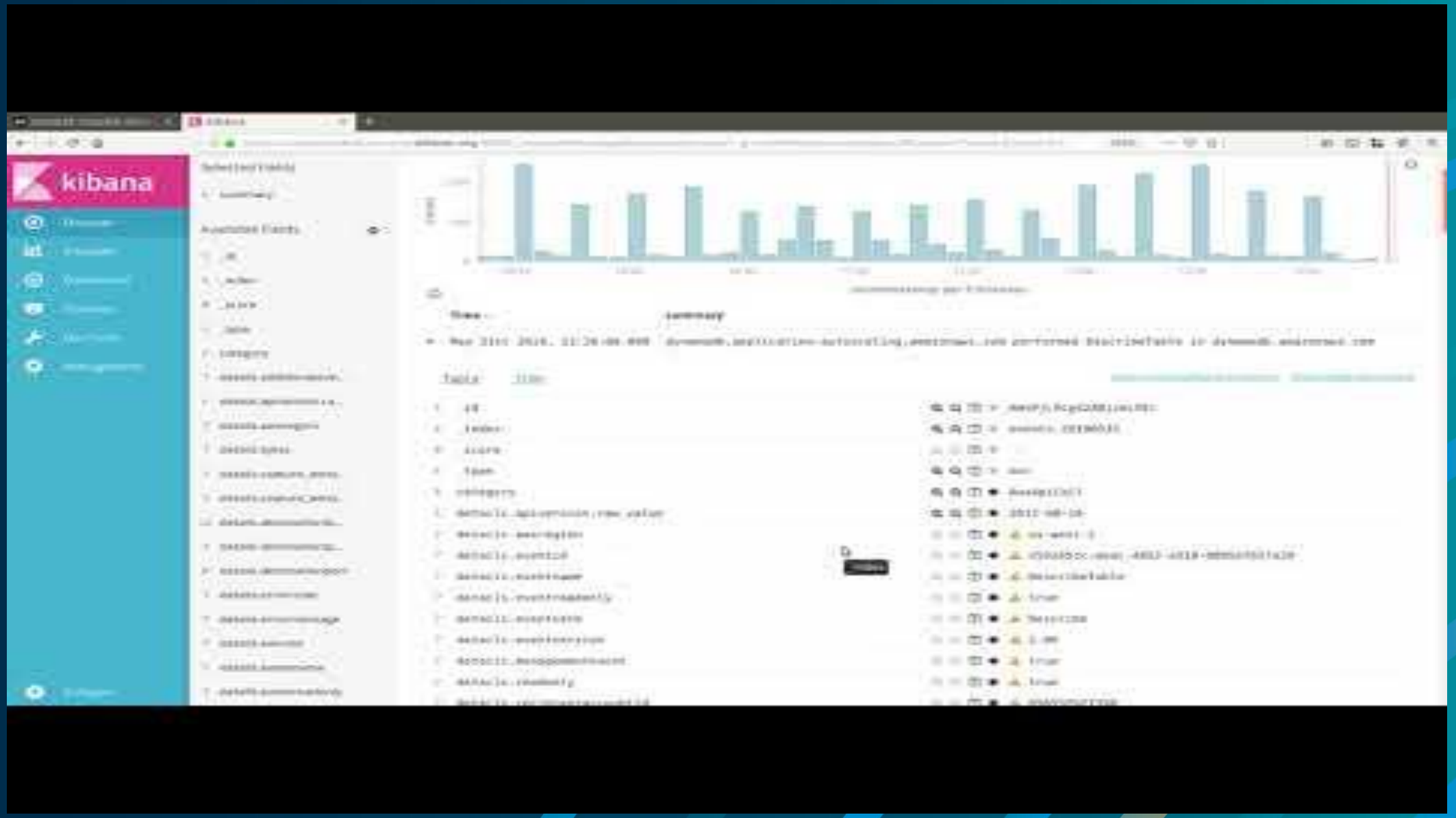
New S3 bucket suddenly made public

API calls from two or more IPs for the same principle

We will provide full solutions at the end of the workshop!

Exploring your environment with Kibana

aws RE:INFORCE



aws RE:INFORCE

Mozdef Event Structure : Overview


```
{
  "category": "AwsApiCall",
  "processid": "8113",
  "receivedtimestamp": "2019-05-31T12:58:55.651911+00:00",
  "severity": "INFO",
  "utctimestamp": "2019-05-31T12:52:44+00:00",
  "processname": "uwsgi",
  "timestamp": "2019-05-31T12:52:44+00:00",
  "hostname": "iam.amazonaws.com",
  "mozdefhostname": "mozdef6.private.mdc1.mozilla.com",
  "summary": "203.0.113.177 performed CreateAccessKey in iam.amazonaws.com",
  "source": "cloudtrail",
  "details": {},
  "plugins": [
    "lower_keys",
    "cloudtrail",
    "ipFixup",
    "geoip"
  ],
  "type": "cloudtrail",
  "tags": []
}
```

Mozdef Event Structure : Details


```
{
  "eventversion": "1.05",
  "eventid": "99f5e75f-938a-44d5-8ce7-de22386de59a",
  "responseelements": {
    "accesskey": {
      "status": "Active",
      "username": "johndoe",
      "createdate": "May 31, 2019 12:52:44 PM",
      "accesskeyid": "AKIATOEMIJ2CVIWLR2GX"
    }
  },
  "sourceipaddress": "203.0.113.177",
  "eventverb": "Create",
  "requestparameters": {
    "username": "johndoe"
  },
  "awsregion": "us-east-1",
  "eventname": "CreateAccessKey",
  "eventreadonly": false,
  "sourceipv4address": "203.0.113.177",
  "sourceipgeolocation": {
    "city": "Buffalo",
    "region_code": "NY",
    "time_zone": "America/New_York",
    "dma_code": 514,
    "metro_code": "Buffalo, NY",
    "country_name": "United States",
    "postal_code": "14217",
    "longitude": -78.8769,
    "country_code": "US",
    "latitude": 42.9719,
    "continent": "NA"
  },
  "useridentity": {
    "username": "janedoe",
    "principalid": "AIDAIXG56MO4VJVDB4CX2",
    "accesskeyid": "ASIATOEMIJ2C6UD5TYOJ",
    "invokedby": "signin.amazonaws.com",
    "sessioncontext": {
      "attributes": {
        "creationdate": "2019-05-31T12:49:32Z",
        "mfaauthenticated": "true"
      }
    },
    "type": "IAMUser",
    "arn": "arn:aws:iam::123456789123:user/janedoe",
    "accountid": "123456789123"
  },
  "useragent": "signin.amazonaws.com",
  "recipientaccountid": "123456789123",
  "requestid": "fb933566-83a2-11e9-b4d0-adbaaa3e035a"
}
```

Authoring Alerts

aws RE:INFORCE

Two types of alerts. No waiting.

aws RE:INFORCE

Why alert?

- You can't respond if you can't detect.
- There are lots of patterns / IOCs

IOC: Indicator of Compromise

Alert Type 1: Simple Alerts

Respond to the presence of an event that contains an action.

Example: Someone made a specific api call. `iam:CreateUser`

```
# How many minutes back in time would you like to search?
search_query = SearchQuery(minutes=15)


# What would you like to search for?
search_query.add_must([
    TermMatch('source', 'cloudtrail'),
    TermMatch('details.eventname', "CreateUser")
])


self.filtersManual(search_query)
self.searchEventsSimple()
self.walkEvents()
```

Alert Type 2: Aggregation Alert

- Respond to the presence of multiple events across a sliding window
- Example: Someone did n of x things over x services in a time series.

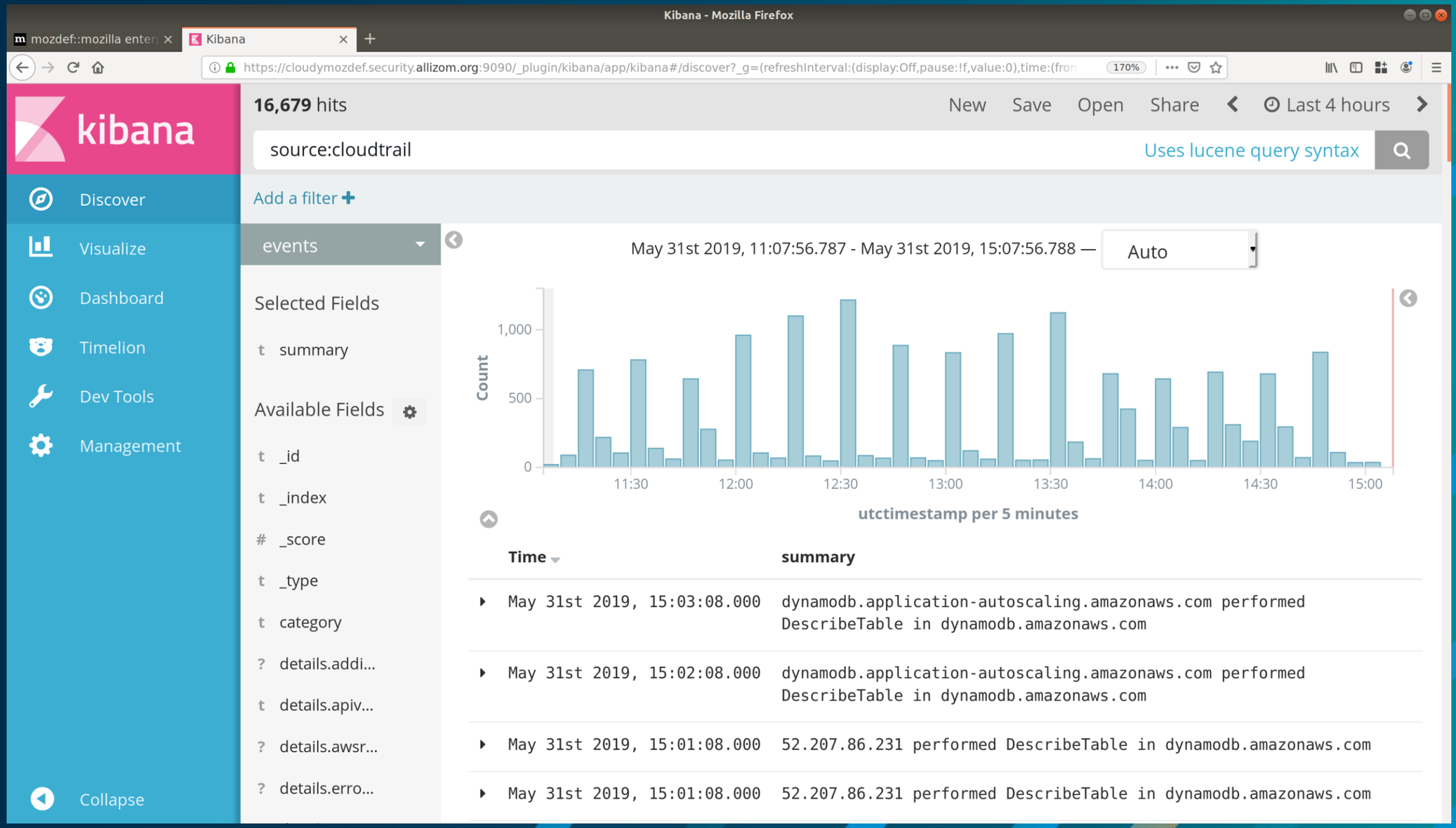
```
# Create a query to look back the last 20 minutes
search_query = SearchQuery(minutes=20)


# Add search terms to our query
search_query.add_must([
    TermMatch('source', 'cloudtrail'),
    TermMatch('details.eventverb', 'Describe'),
    ExistsMatch('details.source')
])


self.filtersManual(search_query)
# We aggregate on details.hostname which is the AWS service name
self.searchEventsAggregated('details.source', samplesLimit=2)
self.walkAggregations(threshold=50)
```

Live Show Cloud9 IDE for Alerts

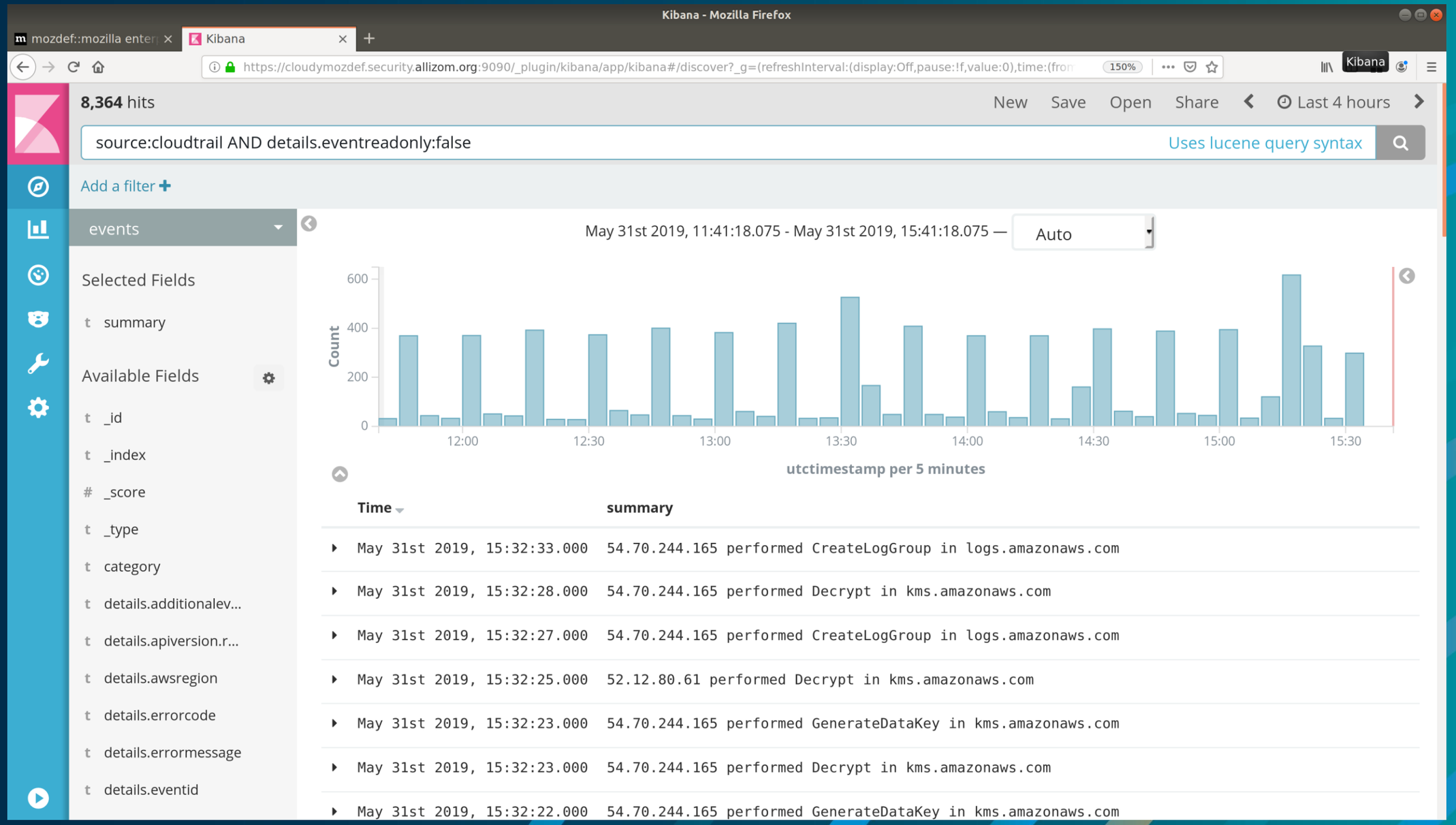
aws RE:INFORCE

Investigating with Kibana

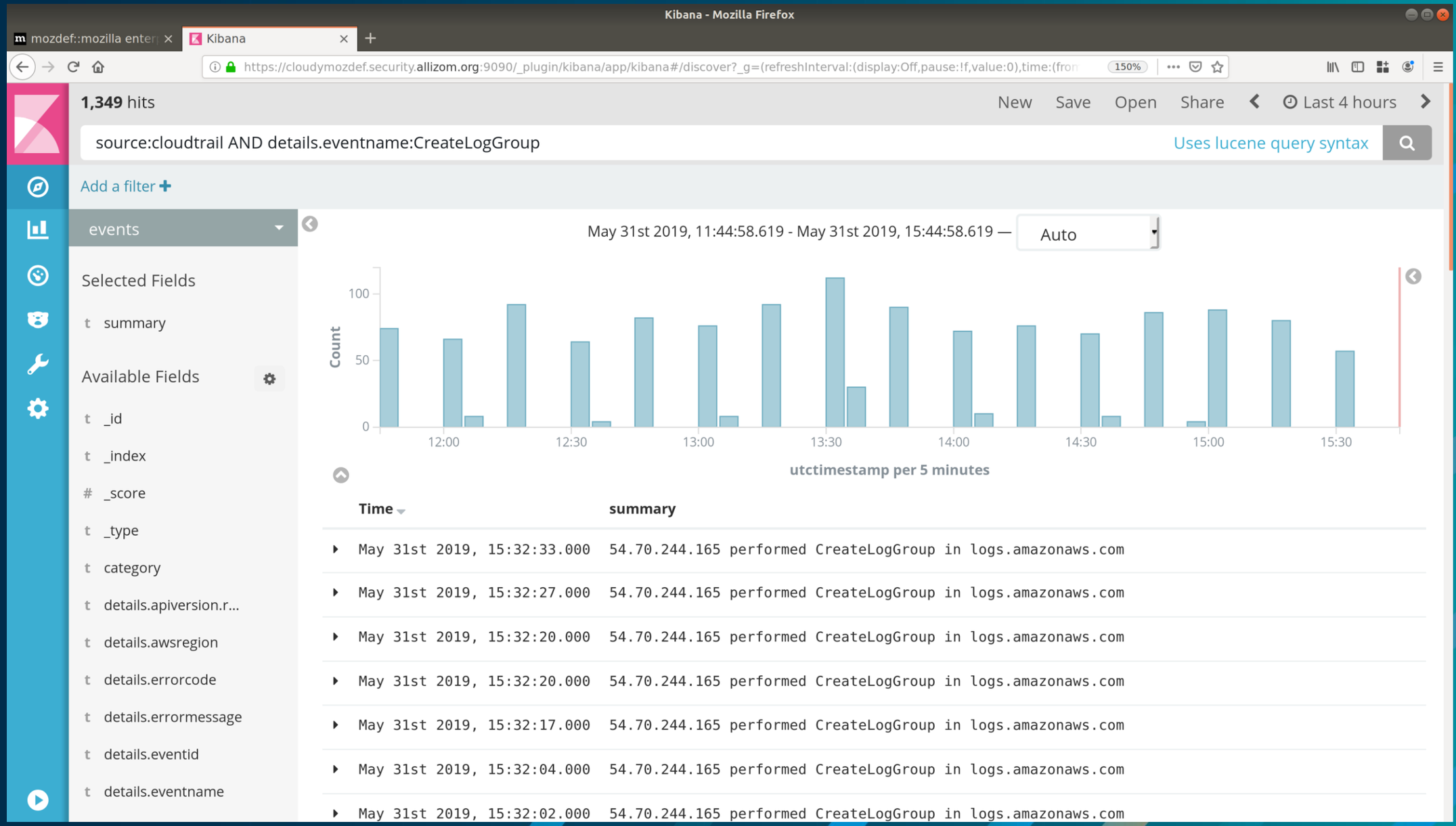
aws RE:INFORCE



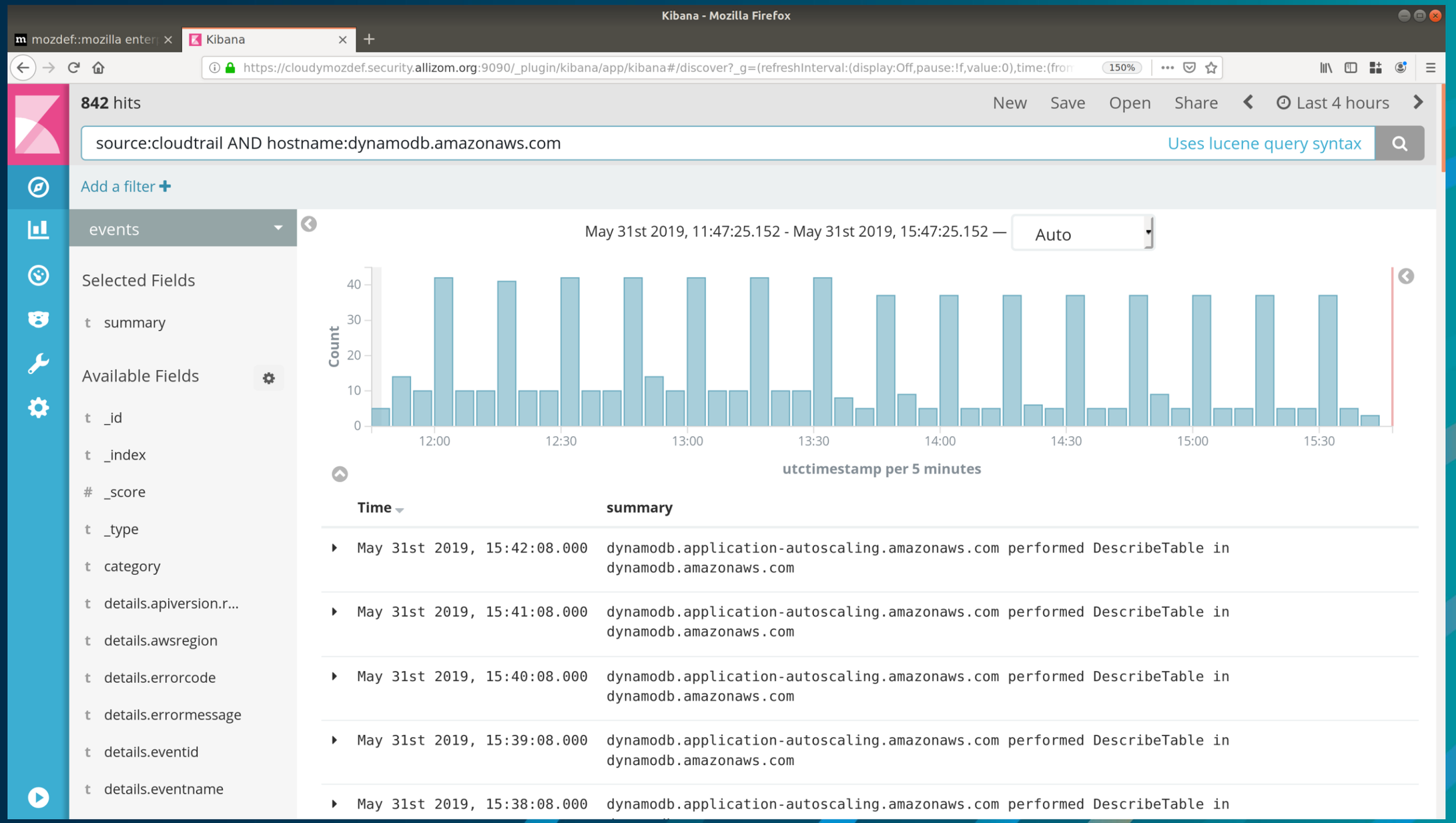
aws RE:INFORCE



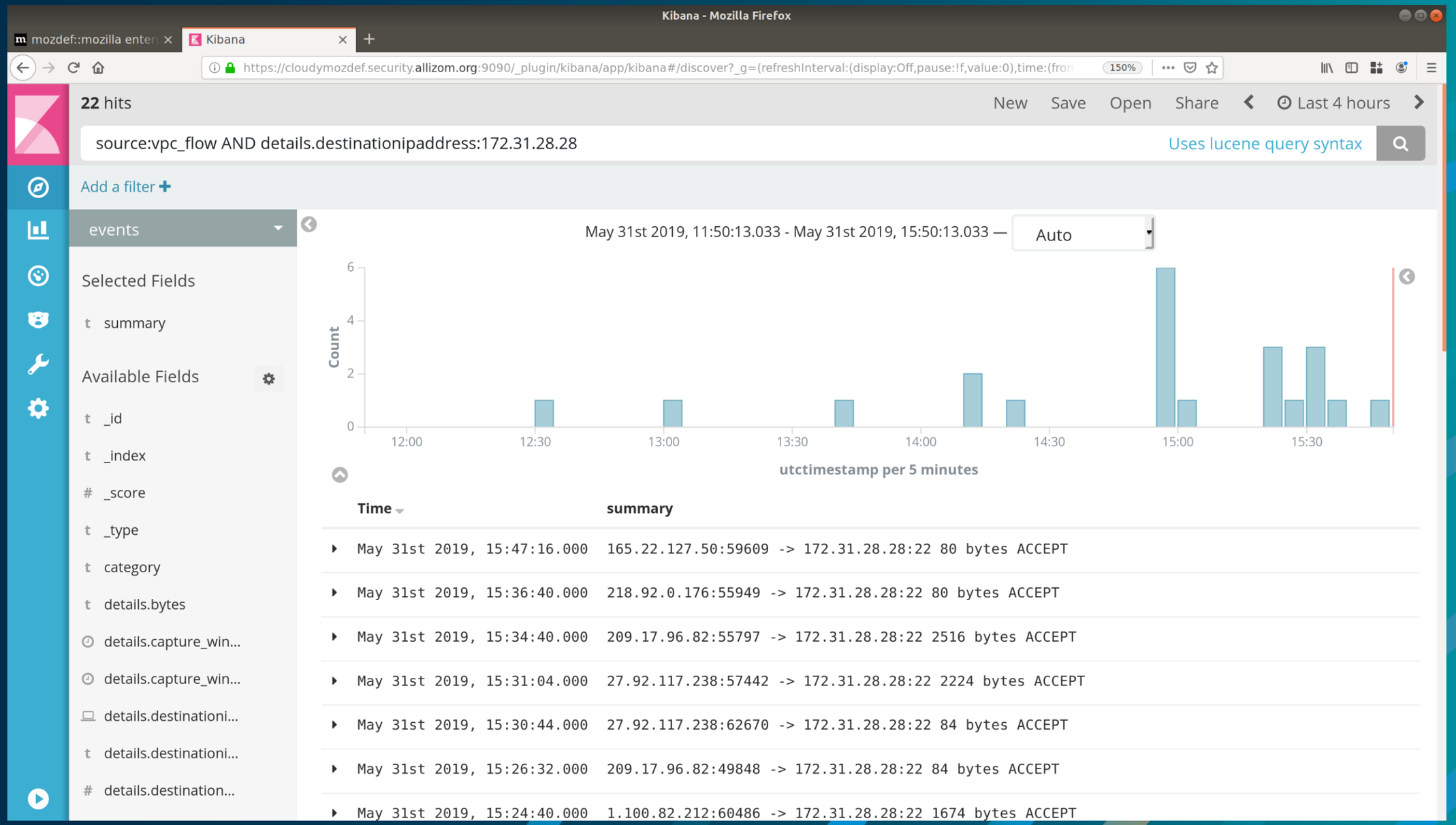
aws RE:INFORCE



aws RE:INFORCE



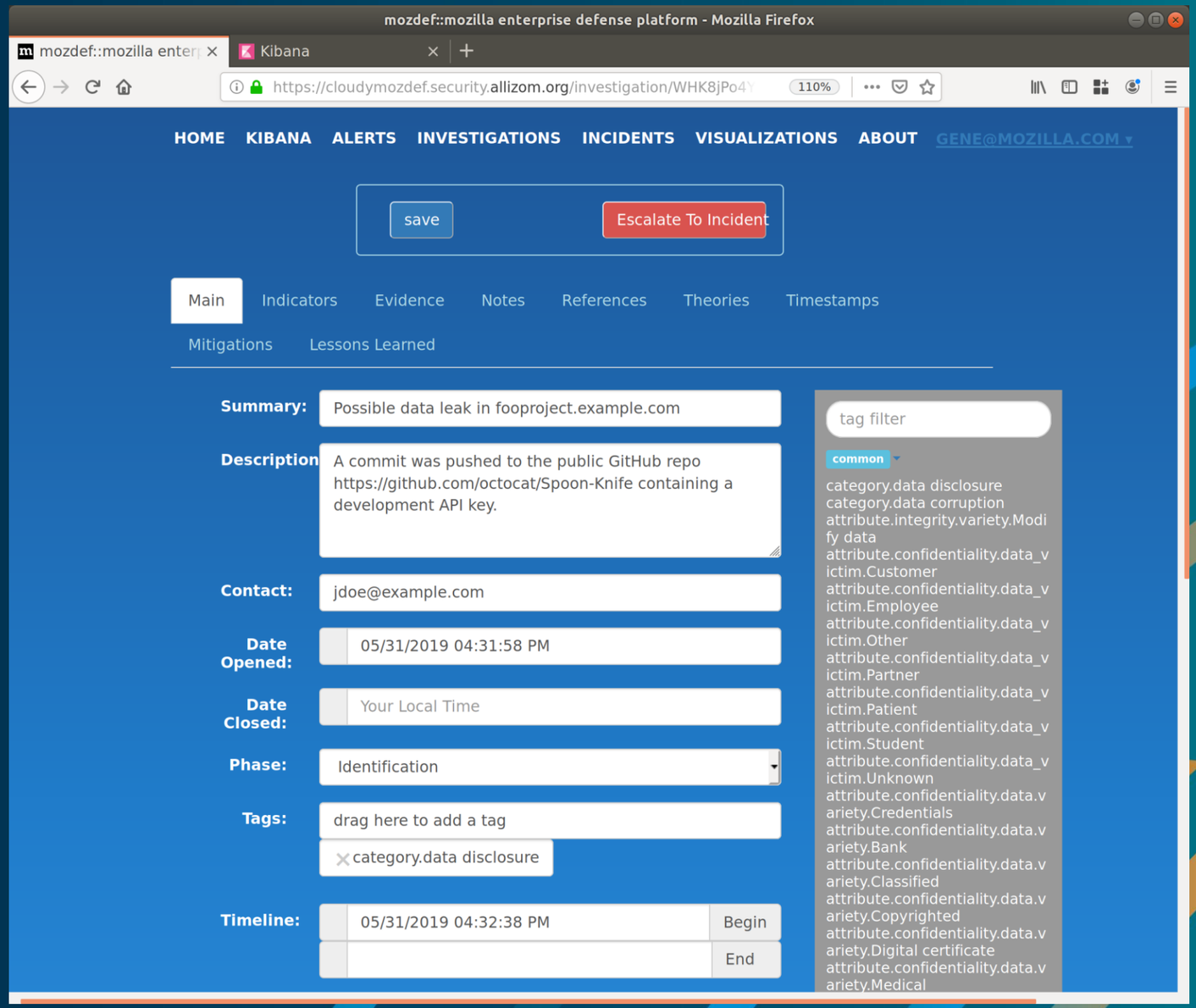
aws RE:INFORCE



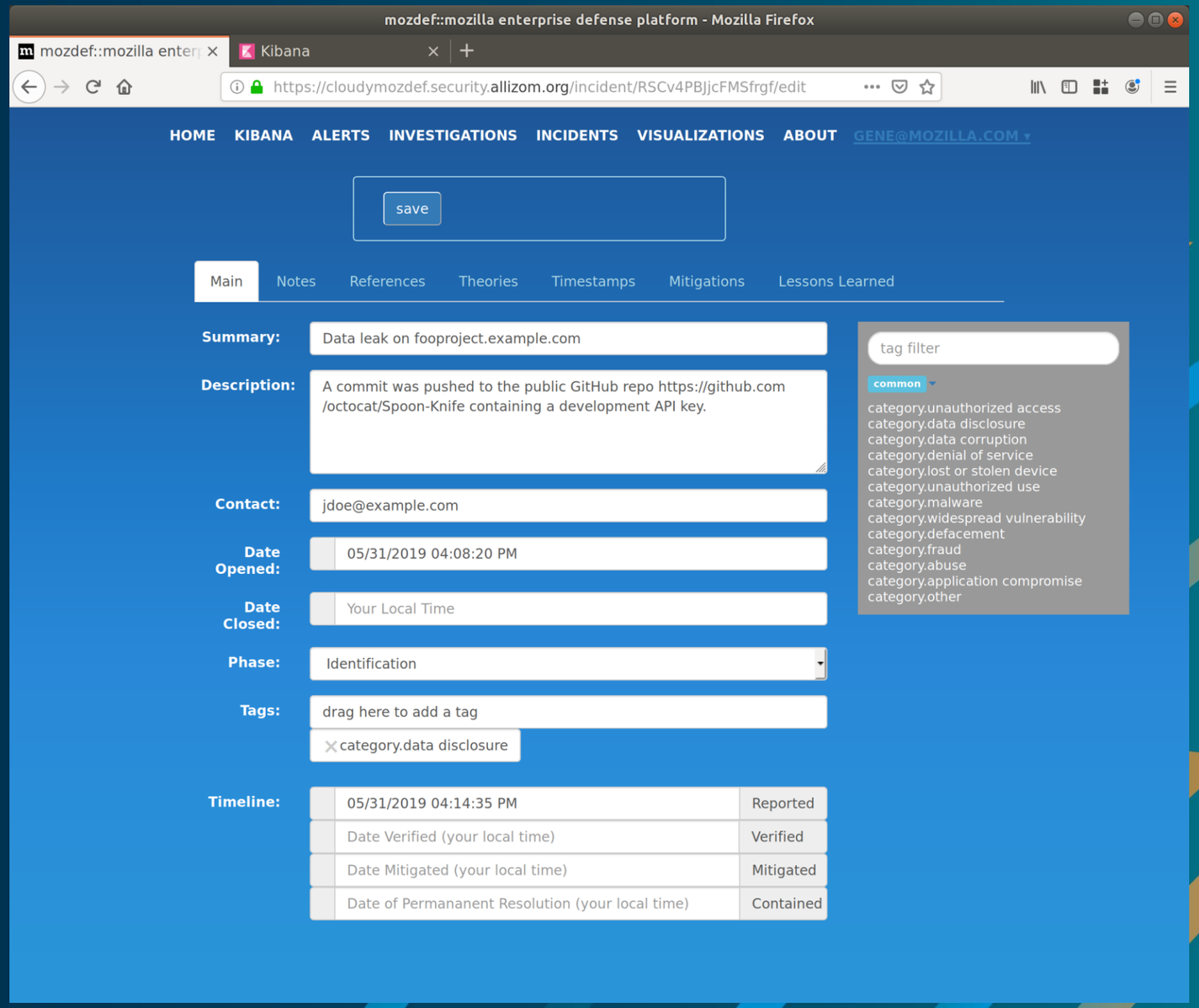
aws RE:INFORCE

Initial Investigation

- Gather data based on the initial indicator you received
  - IP Address, AWS IAM username, time period, AWS service
- Determine who the responsible actor is
  - What IP / IAM User / IAM Role is acting
  - Is this benign
- Escalate from Investigation to Incident



aws RE:INFORCE



aws RE:INFORCE

Incident Response with Kibana

- Determine what the attacker has done by tracking their actions in events
- Identify what actions need to be taken as a result to contain the breach
- Use data in Kibana to determine the systems that need to be addressed to eradicate the threat

"Doing your best on your worst day."

Responding to a Threat aka Incident Response

aws RE:INFORCE

High-Level Playbook

Adversary or intern

Your environment

CloudWatch Events

Step Function

Lambda responder

Responding to Findings: Remediation

Responding to Findings: Automation Example

- Lambda Function:
  - Removes instance from current Security Group(s) and adds to one with all ingress and egress blocked
  - Snapshots EBS volume(s)
  - Alerts Security Team

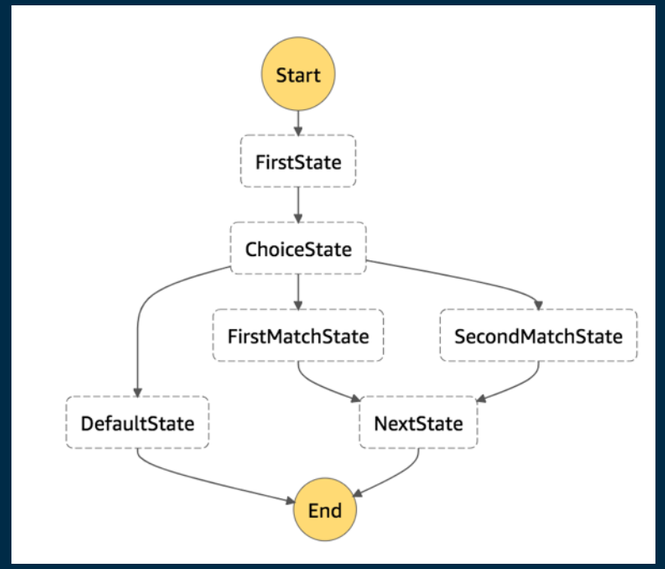
- SSM Document:
  - Forensics can begin
    - Network Capture
    - Memory Dump
    - Process review
    - Internal Tools

Responding to Findings: Automation Example

- Step Function:

AWS Step Functions lets you coordinate multiple AWS services into serverless workflows so you can build and update apps quickly. Using Step Functions, you can design and run workflows that stitch together services such as AWS Lambda and Amazon ECS into feature-rich applications. Workflows are made up of a series of steps, with the output of one step acting as input into the next. Application development is simpler and more intuitive using Step Functions, because it translates your workflow into a state machine diagram that is easy to understand, easy to explain to others, and easy to change. You can monitor each step of execution as it happens, which means you can identify and fix problems quickly. Step Functions automatically triggers and tracks each step, and retries when there are errors, so your application executes in order and as expected.

AWS Step functions

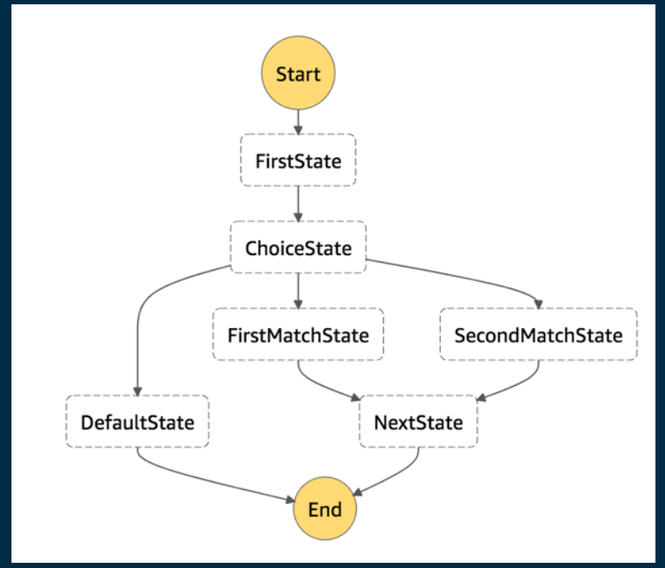


Responding to Findings: Automation Example

- Step Function:

AWS Step Functions lets you coordinate multiple AWS services into serverless workflows so you can build and update apps quickly. Using Step Functions, you can design and run workflows that stitch together services such as AWS Lambda and Amazon ECS into feature-rich applications. Workflows are made up of a series of steps, with the output of one step acting as input into the next. Application development is simpler and more intuitive using Step Functions, because it translates your workflow into a state machine diagram that is easy to understand, easy to explain to others, and easy to change. You can monitor each step of execution as it happens, which means you can identify and fix problems quickly. Step Functions automatically triggers and tracks each step, and retries when there are errors, so your application executes in order and as expected.

AWS Step functions



Remediation - CryptoCurrency:EC2/BitcoinTool.B!DNS


```
[
 {
  "schemaVersion": "2.0",
  "accountId":”0123456789",
  "region": "us-west-2",
  "partition": "aws",
  "id": ”[GUID]",
  "arn": "arn:aws:guardduty:us-west-2:01234567890:detector/[GUID]/finding/[Finding GUID]",
  "type": "CryptoCurrency:EC2/BitcoinTool.B!DNS",
  "resource": {
   "resourceType": "Instance",
   "instanceDetails": {
    "instanceId": "i-99999999",
    "instanceType": "p2.xlarge",
    "launchTime": "2017-12-20T23
    "platform": null,
    "productCodes": [
     {
      "productCodeId": "Generate
      "productCodeType": "Generat
     }
```


Finding: [“type”]= “**CryptoCurrency:EC2/BitcoinTool.B!DNS”**
Instance: [“instanceDetails”][“instanceId”] = “i-99999999”

Remediation - CryptoCurrency:EC2/BitcoinTool.B!DNS

Problem description

CryptoCurrency:EC2/BitcoinTool.B!DNS has been found in GuardDuty under this mean that we have an account or machine that has been compromised.

This finding informs you that an EC2 instance in your AWS environment is querying a domain name that is associated with Bitco in-related activity. Bitcoin is a worldwide cryptocurrency and digital payment system. Besides being created as a reward for Bitcoin mining, bitcoin can be exchanged for other currencies, products, and services. Unless you use this EC2 instance to mine or manage cryptocurrency or your EC2 instance is involved in blockchain activity, yo ur EC2 instance might be compromised.

Data to gather for troubleshooting

Account User ID, Role or Profile that was accessed
Instance ID , Subnet ID, VPC ID
Connectivity to other systems
Review of CloudTrail and VPC Flows to and around the specified instance.

Steps to troubleshoot and fix

1.Notify IR Team On call.
2.Run Automate instance quarantine
3.Role credentials associated with the above identity
4.Snapshot instance and VPC Flow logs to forensics account
5.Validate that new ASG created instance is working correctly

Urgency category

Critical

Escalation path:

Unable to fix, escalate to these individuals or gro
1.Someone, email and phone number
2.Someone Else, email phone number
3.Distribution List
4....
5....

Finding: ["type"]= "**CryptoCurrency:EC2/BitcoinTool.B!DNS"**
Instance: ["instanceDetails"]["instanceId"] = "i-99999999"

Remediation - CryptoCurrency:EC2/BitcoinTool.B!DNS

Problem description

CryptoCurrency:EC2/BitcoinTool.B!DNS has been found in GuardDuty under this mean that we have an account or machine that has been compromised.

This finding informs you that an EC2 instance in your AWS environment is querying a domain name that is associated with Bitcoin-related activity. Bitcoin is a worldwide cryptocurrency and digital payment system. Besides being created as a reward for Bitcoin mining, bitcoin can be exchanged for other currencies, products, and services. Unless you use this EC2 instance to mine or manage cryptocurrency or your EC2 instance is involved in blockchain activity, your EC2 instance might be compromised.

Data to gather for troubleshooting

Account User ID, Role or Profile that was accessed

Instance ID , Subnet ID, VPC ID

Connectivity to other systems

Review of CloudTrail and VPC Flows to and around the specified instance.

Steps to troubleshoot and fix

1.Notify IR Team On call.

2.Run Automate instance quarantine

3.Role credentials associated with the above identity

4.Snapshot instance and VPC Flow logs to forensics account

5.Validate that new ASG created instance is working correctly

Urgency category

Critical

Escalation path:

Unable to fix, escalate to these individuals or groups in this o

1.Someone, email and phone number

2.Someone Else, email phone number

3.Distribution List

4....

5....

Steps to troubleshoot and fix

1.Notify IR Team On call.

2.Run Automate instance quarantine

3.Role credentials associated with the above identity

4.Snapshot instance and VPC Flow logs to forensics account

5.Validate that new ASG created instance is working correctly

Remediation - CryptoCurrency:EC2/BitcoinTool.B!DNS

Problem description

CryptoCurrency:EC2/BitcoinTool.B!DNS has been found in GuardDuty under this mean that we have an account or machine that has been compromised.

This finding informs you that an EC2 instance in your AWS environment is querying a domain name that is associated with Bitcoin-related activity. Bitcoin is a worldwide cryptocurrency and digital payment system. Besides being created as a reward for Bitcoin mining, bitcoin can be exchanged for other currencies, products, and services. Unless you use this EC2 instance to mine or manage cryptocurrency or your EC2 instance is in

Data to gather for troubleshooting

Account User ID, Role or Profile that was accessed
Instance ID , Subnet ID, VPC ID
Connectivity to other systems
Review of CloudTrail and VPC Flows to and around th

Steps to troubleshoot and fix

1.Notify IR Team On call.
2.Run Automate instance quarantine
3.Role credentials associated with the above identity
4.Snapshot instance and VPC Flow logs to forensics a
5.Validate that new ASG created instance is working

Urgency category

Critical

Escalation path:

Unable to fix, escalate to these individuals or groups
1.Someone, email and phone number
2.Someone Else, email phone number
3.Distribution List
4....
5....

Items to Code:

1. Cloud Watch Filter to trap a finding from GuardDuty, with:
   ["type"]= "**CryptoCurrency:EC2/BitcoinTool.B!DNS"**
2. Step Functions Start
   a. SNS Fires to notify Ops of an issue
   b. Lambda Function is fired to run SSM
      i. Finished and a Lambda Function is fired to quarantine the instance
   c. Lambda Function is fired to Snap Shot the instance
   d. Step Function checks responses
3. Lambda is fired to Stop and destroy the instance.

aws

Remediation - CryptoCurrency:EC2/BitcoinTool.B!DNS

Problem description

CryptoCurrency:EC2/BitcoinTool.B!DNS has been found in GuardDuty under this mean that we have an account or machine that has been compromised.

This finding informs you that an EC2 instance in your AWS environment is querying a domain name that is associated with Bitcoin-related activity. Bitcoin is a worldwide cryptocurrency and digital payment system. Besides being created as a reward for Bitcoin mining, bitcoin can be exchanged for other currencies, products, and services. Unless you use this EC2 instance to mine or manage cryptocurrency or your EC2 instance is involved in blockchain activity, your EC2 instance might be compromised.

Data to gather for troubleshooting

Account User ID, Role or Profile that was accessed

Instance ID , Subnet ID, VPC ID

Connectivity to other systems

Review of CloudTrail and VPC Flows to and around the specified instance.

Steps to troubleshoot and fix

1.Notify IR Team On call.

2.Run Automate instance quarantine

3.Role credentials associated with the above identity

4.Snapshot instance and VPC Flow logs to forensics account

5.Validate that new ASG created instance is working correctly

Urgency category

Critical

Escalation path:

Unable to fix, escalate to these individuals or groups

1.Someone, email and phone number

2.Someone Else, email phone number

3.Distribution List

4....

5....

Items to Code:

1. Actual Coding to occur later in the talk.

aws

Remediation - CryptoCurrency:EC2/BitcoinTool.B!DNS

Problem description

CryptoCurrency:EC2/BitcoinTool.B!DNS has been found in GuardDuty under this mean that we have an account or machine that has been compromised.

This finding informs you that an EC2 instance in your AWS environment is querying a domain name that is associated with Bitcoin-related activity. Bitcoin is a worldwide cryptocurrency and digital payment system. Besides being created as a reward for Bitcoin mining, bitcoin can be exchanged for other currencies, products, and services. Unless you use this EC2 instance to mine or manage cryptocurrency or your EC2 instance is involved in blockchain activity, your EC2 instance might be compromised.

Data to gather for troubleshooting

Account User ID, Role or Profile that was accessed

Instance ID , Subnet ID, VPC ID

Connectivity to other systems

Review of CloudTrail and VPC Flows to and around the specified instance.

Steps to troubleshoot and fix

1.Notify IR Team On call.

2.Run Automate instance quarantine

3.Role credentials associated with the above ide

4.Snapshot instance and VPC Flow logs to foren

5.Validate that new ASG created instance is wo

Urgency category

Critical

Escalation path:

Unable to fix, escalate to these individuals or gr

1.Someone, email and phone number

2.Someone Else, email phone number

3.Distribution List

4....

5....

Escalation path:

Unable to fix, escalate to these individuals or groups in this order:

1.Someone, email and phone number

2.Someone Else, email phone number

3.Distribution List

4....

5....

Postmortem - CryptoCurrency:EC2/BitcoinTool.B!DNS

Problem description

CryptoCurrency:EC2/BitcoinTool.B!DNS has been found in GuardDuty under this mean that we have an account or machine that has been compromised.

John, our lead developer added his AWS Key and Secret key to his most recent git post. This was found by someone and then sold to a Crypto Mining company in another country. We had bad threat detection and the account was utilized for a couple of days before we found out.

-or-

John had his laptop stolen and didn't encrypt his hard drive. Because he kept every thing in his local Git Repo his user was compromised.

Postmortem

Utilize good development practices. Adding static variables that contain access keys to a git, causes long term issues for a cloud account.

- Utilize git-secrets
  - Attend a workshop at re:invent discussing use of open source dev tools
- Limit blast radius
  - Enjoy one of the multi account session at re:Invent

The loss of corporate resources that were unencrypted.

- Encrypt hard Drives going forward
- Limit account activities of humans for threat detection
- Limit account access of people in production and test environmen

Members of the Postmortem Team:

Developers
Operations
Security Operations
Management ?
Leadership Level ?

Postmortem - CryptoCurrency:EC2/BitcoinTool.B!DNS

Problem description

CryptoCurrency:EC2/BitcoinTool.B!DNS has been found in GuardDuty under this mean that we have an account or machine that has been compromised.

John, our lead developer added his AWS Key and Secret key to his most recent git post. This was found by someone and then sold to a Crypto Mining company in another country. We had bad threat detection and the account was utilized for a couple of days before we found out.

-or-

John had his laptop stolen and didn't encrypt his hard drive. Because he kept every thing in his local Git Repo his user was compromised.

Postmortem

Utilize good development practices. Adding static variables that contain access keys to a git, causes long term issues for a cloud account.

- Utilize git-secrets
  - Attend a workshop at re:invent discussing use of open source dev tools
- Limit blast radius
  - Enjoy one of the multi account session at re:Invent

The loss of corporate resources that were unencrypted.

- Encrypt hard Drives going forward
- Limit account activities of humans for threat detection
- Limit account access of people in production and test environments

Aws_labs repos.
https://github.com/awslabs

Wrap up / Retro

aws RE:INFORCE

Retro

How to reach us

Twitter: @andrewkrug @kangsterizer

Email: akrug@mozilla.com gene@mozilla.com kang@mozilla.com

Want to get in the MozDef Beta? - take a quick survey

https://www.surveygizmo.com/s3/5040959/7ef0ac201fb2

Nathan Case
Security Geek
AWS

Andrew Krug
Staff Security Engineer
Mozilla

Gene Wood
Senior Staff Security Engineer
Mozilla

Guillaume Destuynder
Senior Staff Security Engineer
Mozilla

Thank You!

Don’t forget to do your surveys!

aws RE:INFORCE