

aws RE:INFORCE



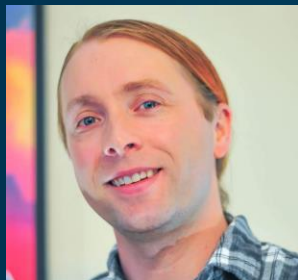
SEP 3 15

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

Nathan Case
Security Geek
AWS

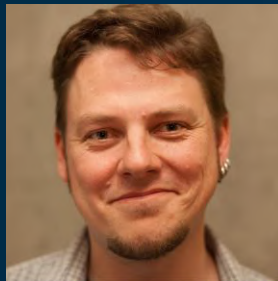


Andrew Krug
Staff Security Engineer
Mozilla

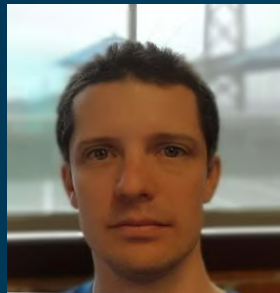


SEP 3 15

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

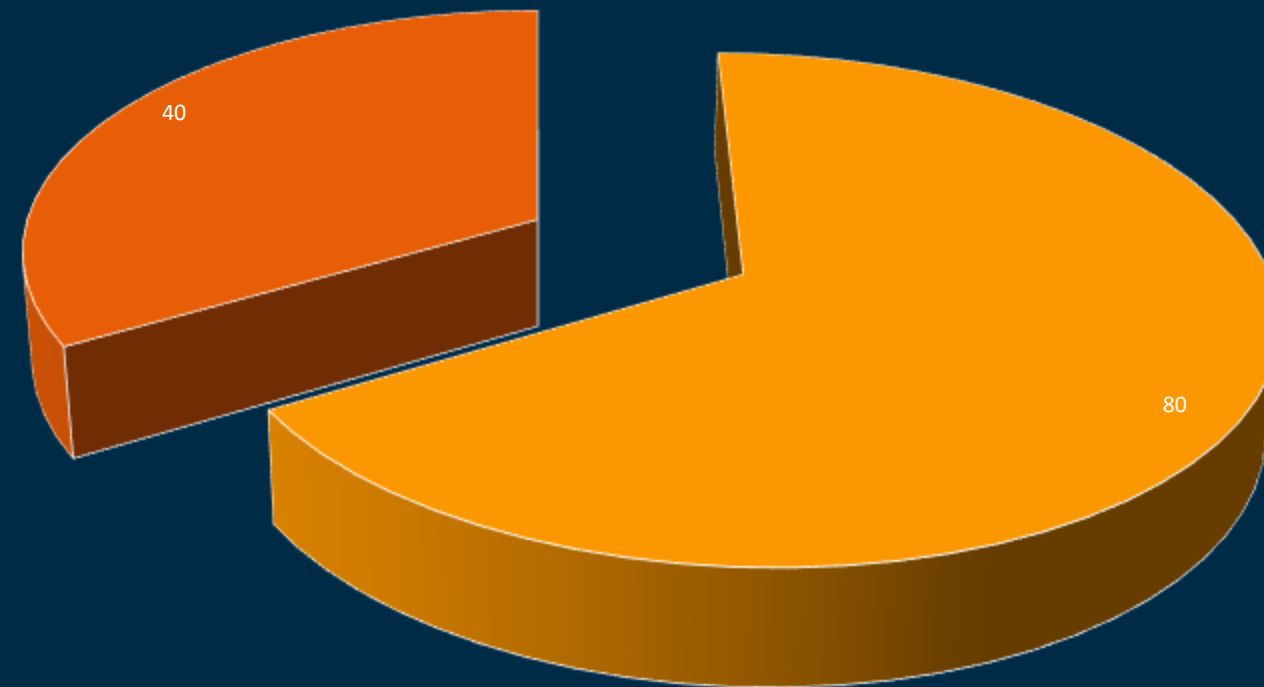
Agenda – 120 minute workshop

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

Workshop Section 2

Wrap up / Retrospective (2-minutes)

A workshop unlike any other



■ Hands On ■ Talking

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.”

Why Open Source

moz://a

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

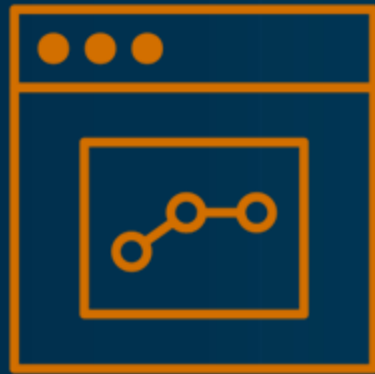
For us, open means **transparent**, as in open source – you're not locked in to what the original creator did.

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



Mitchell Baker
Chair, Mozilla

Contribution is not just CODE!



Use

Give Feedback

Contribute

Advocate

Different Types of Participation

We don't compete on security.

Dan Kaminsky



Anatomy of a Healthy Security Operation

aws RE:INFORCE

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

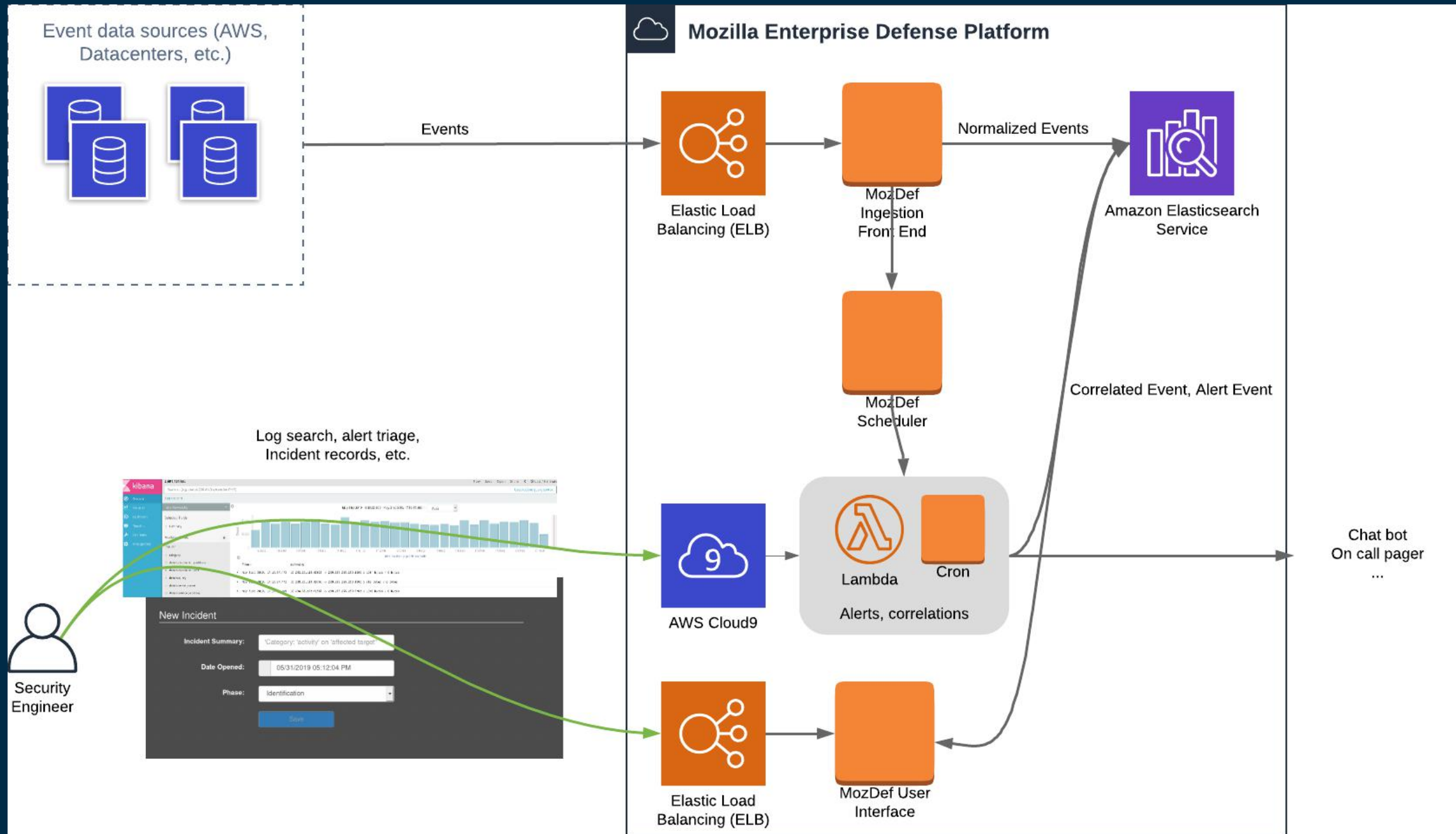
- Log records (event data)
- Alerts
- Event data correlation
- Incident management
- Reports, dashboards, compliance
- OIC storage (indicators of compromise)

Mozilla Enterprise Defense Platform is an open-source SIEM

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

Ip: 1.2.3.4

User: banzai

...

Normalized event

sourceIpAddress: 1.2.3.4

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...

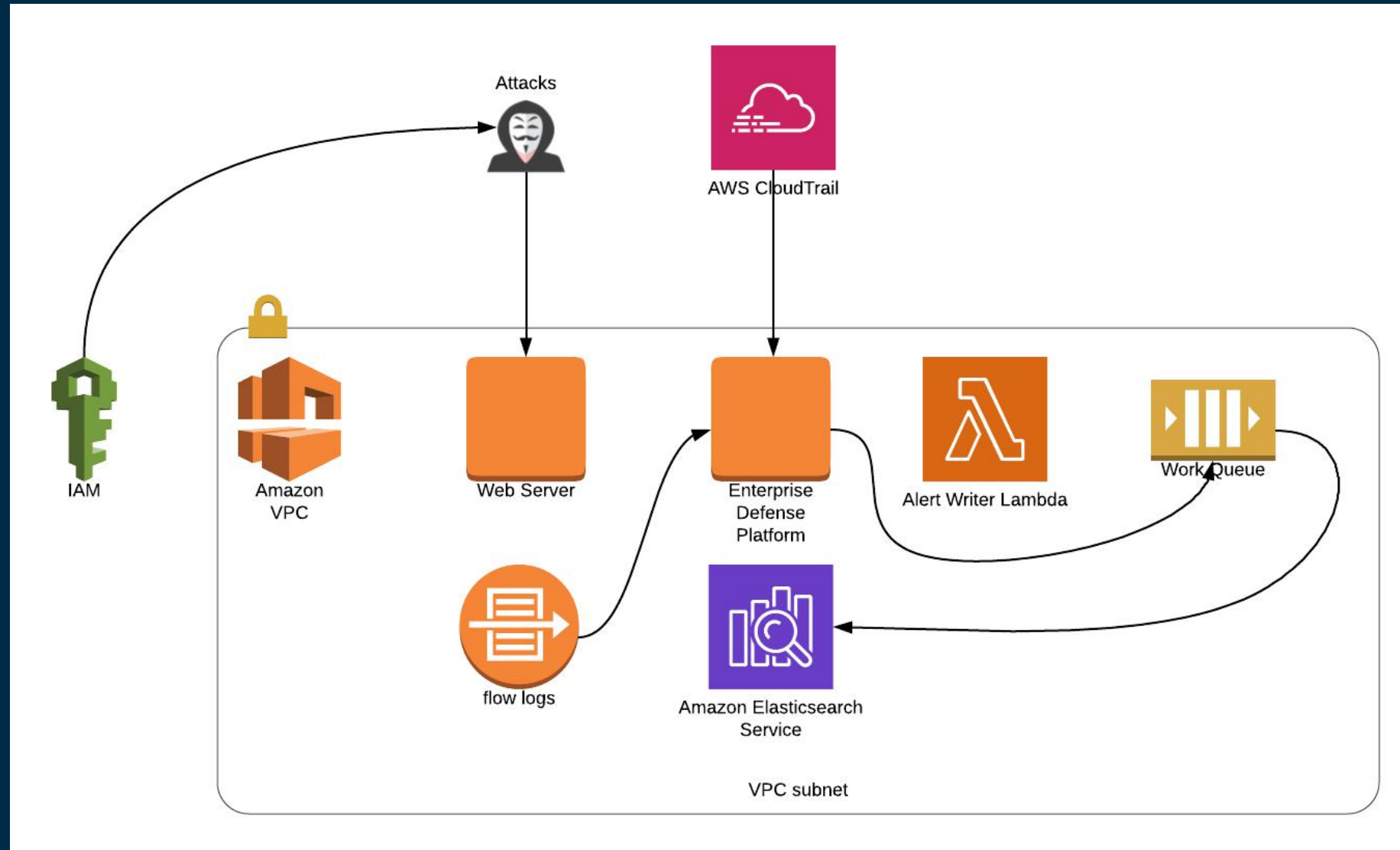
1. Identification / Open incident / Assemble team
2. Containment
3. Eradication
4. Recovery
5. Lesson learned
6. Close the incident

Record all actions as you progress

Security Simulation Overview



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

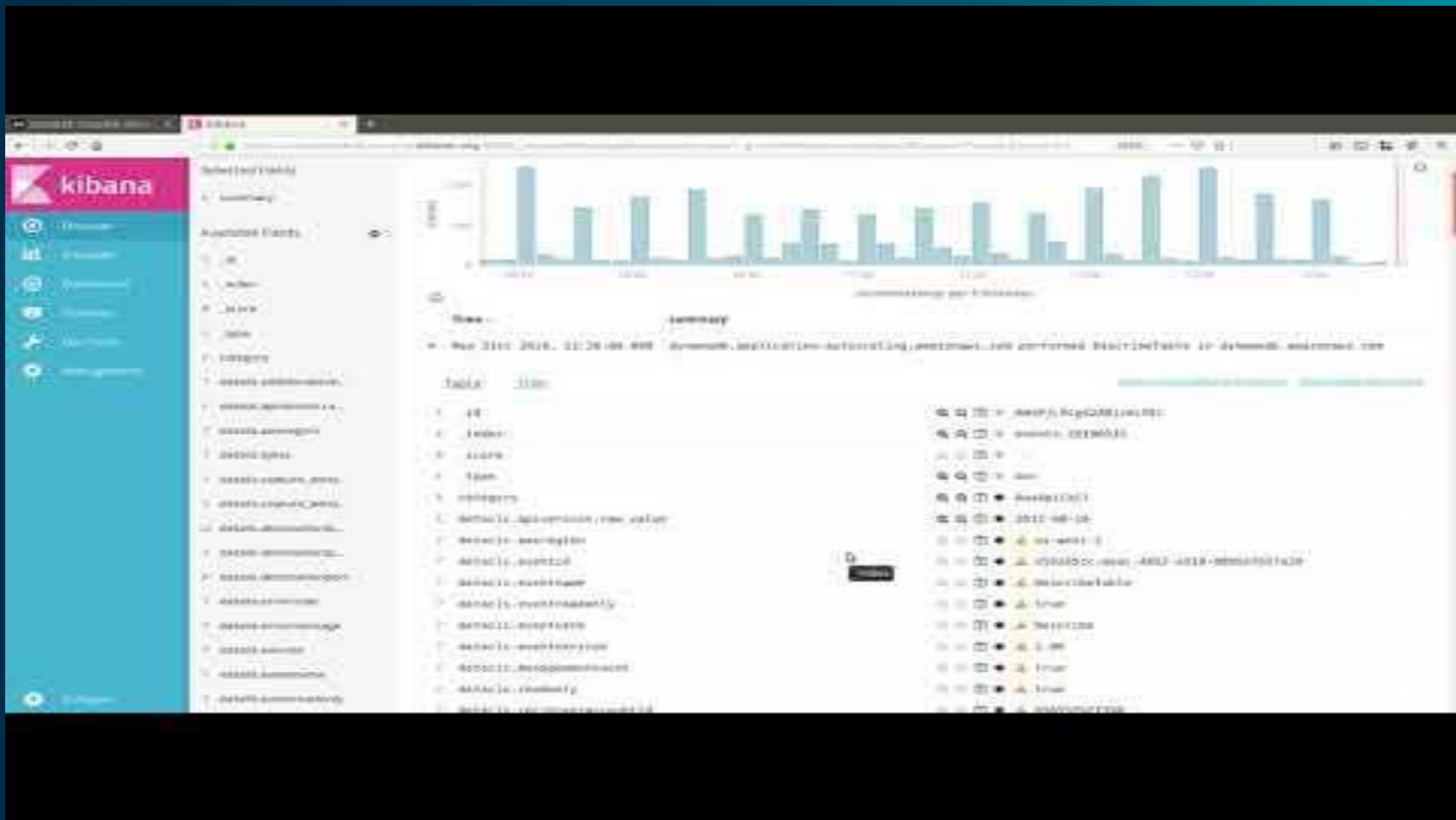
API calls from two or more IPs for the same principle

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

Exploring your environment with Kibana

aws RE:INFORCE

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Mozdef Event Structure : Overview

```
1  {
2    "category": "AwsApiCall",
3    "processid": "8113",
4    "receivedtimestamp": "2019-05-31T12:58:55.651911+00:00",
5    "severity": "INFO",
6    "utctimestamp": "2019-05-31T12:52:44+00:00",
7    "processname": "uwsgi",
8    "timestamp": "2019-05-31T12:52:44+00:00",
9    "hostname": "iam.amazonaws.com",
10   "mozdefhostname": "mozdef6.private.mdc1.mozilla.com",
11   "summary": "203.0.113.177 performed CreateAccessKey in iam.amazonaws.com",
12   "source": "cloudtrail",
13   "details": {},
14   "plugins": [
15     "lower_keys",
16     "cloudtrail",
17     "ipFixup",
18     "geoip"
19   ],
20   "type": "cloudtrail",
21   "tags": []
22 }
```


Mozdef Event Structure : Details

```
1  {
2    "eventversion": "1.05",
3    "eventid": "99f5e75f-938a-44d5-8ce7-de22386de59a",
4    "responseelements": {
5      "accesskey": {
6        "status": "Active",
7        "username": "johndoe",
8        "createdate": "May 31, 2019 12:52:44 PM",
9        "accesskeyid": "AKIATOEMIJ2CVIWL2GX"
10     },
11   },
12   "sourceipaddress": "203.0.113.177",
13   "eventverb": "Create",
14   "requestparameters": {
15     "username": "johndoe"
16   },
17   "awsregion": "us-east-1",
18   "eventname": "CreateAccessKey",
19   "eventreadonly": false,
20   "sourceipv4address": "203.0.113.177",
21   "sourceipgeolocation": {
22     "city": "Buffalo",
23     "region_code": "NY",
24     "time_zone": "America/New_York",
25     "dma_code": 514,
26     "metro code": "Buffalo, NY",
27     "country_name": "United States",
28     "postal_code": "14217",
29     "longitude": -78.8769,
30     "country_code": "US",
31     "latitude": 42.9719,
32     "continent": "NA"
33   },
34   "useridentity": {
35     "username": "janedoe",
36     "principalid": "AIDAIXG56M04VJVDB4CX2",
37     "accesskeyid": "ASIATOEMIJ2C6UD5TY0J",
38     "invokedby": "signin.amazonaws.com",
39     "sessioncontext": {
40       "attributes": {
41         "creationdate": "2019-05-31T12:49:32Z",
42         "mfaauthenticated": "true"
43       }
44     },
45     "type": "IAMUser",
46     "arn": "arn:aws:iam::123456789123:user/janedoe",
47     "accountid": "123456789123"
48   },
49   "useragent": "signin.amazonaws.com",
50   "recipientaccountid": "123456789123",
51   "requestid": "fb933566-83a2-11e9-b4d0-adbaaa3e035a"
52 }
```

Authoring Alerts



Two types of alerts. No waiting.

aws RE:INFORCE

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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')
])

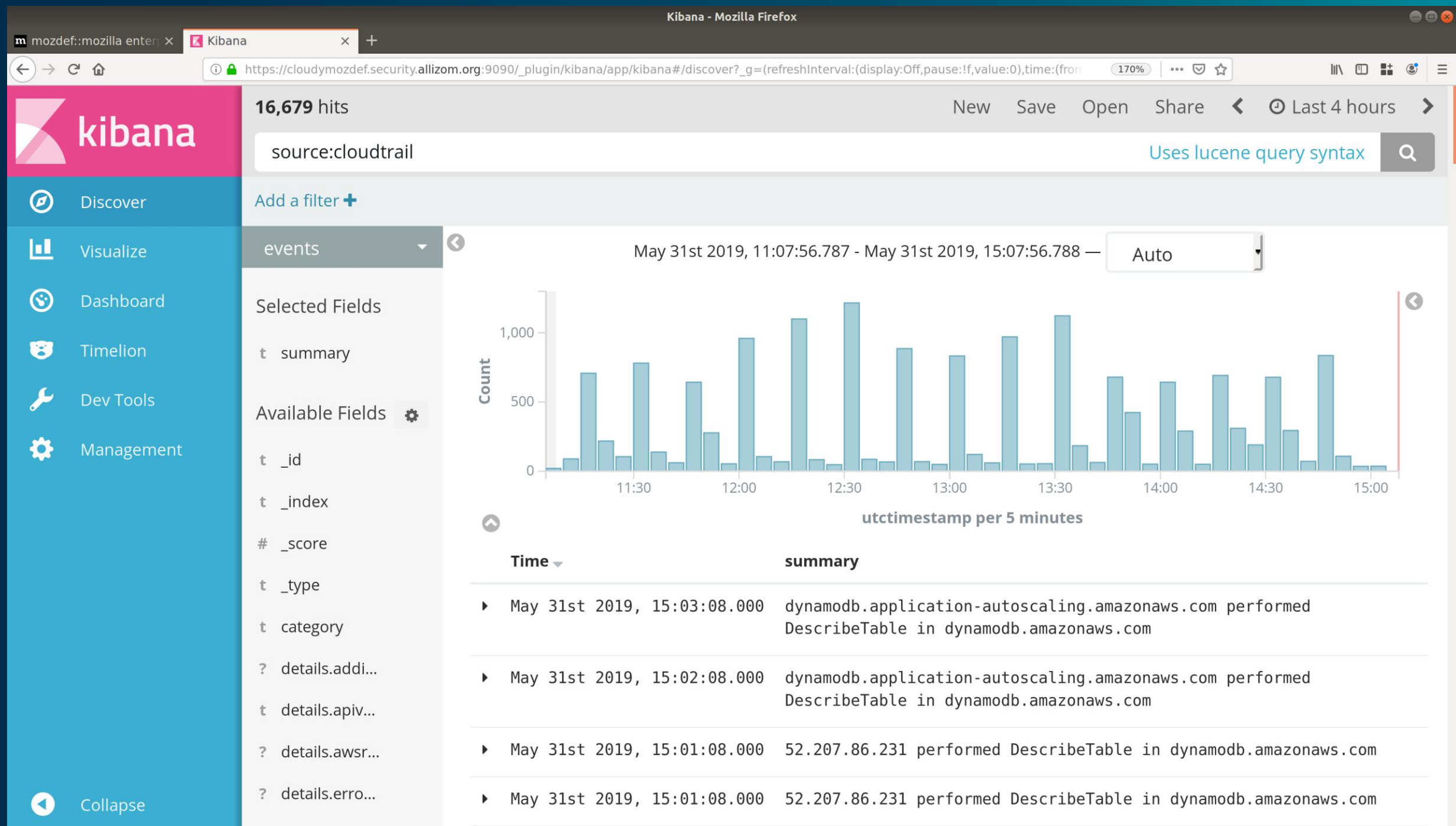
self.filtersManual(search_query)
# We aggregate on details.hostname which is the AWS service name
self.searchEventsAggregated('details.source', samplesLimit=2)
self.walkAggregations(threshold=50)
```

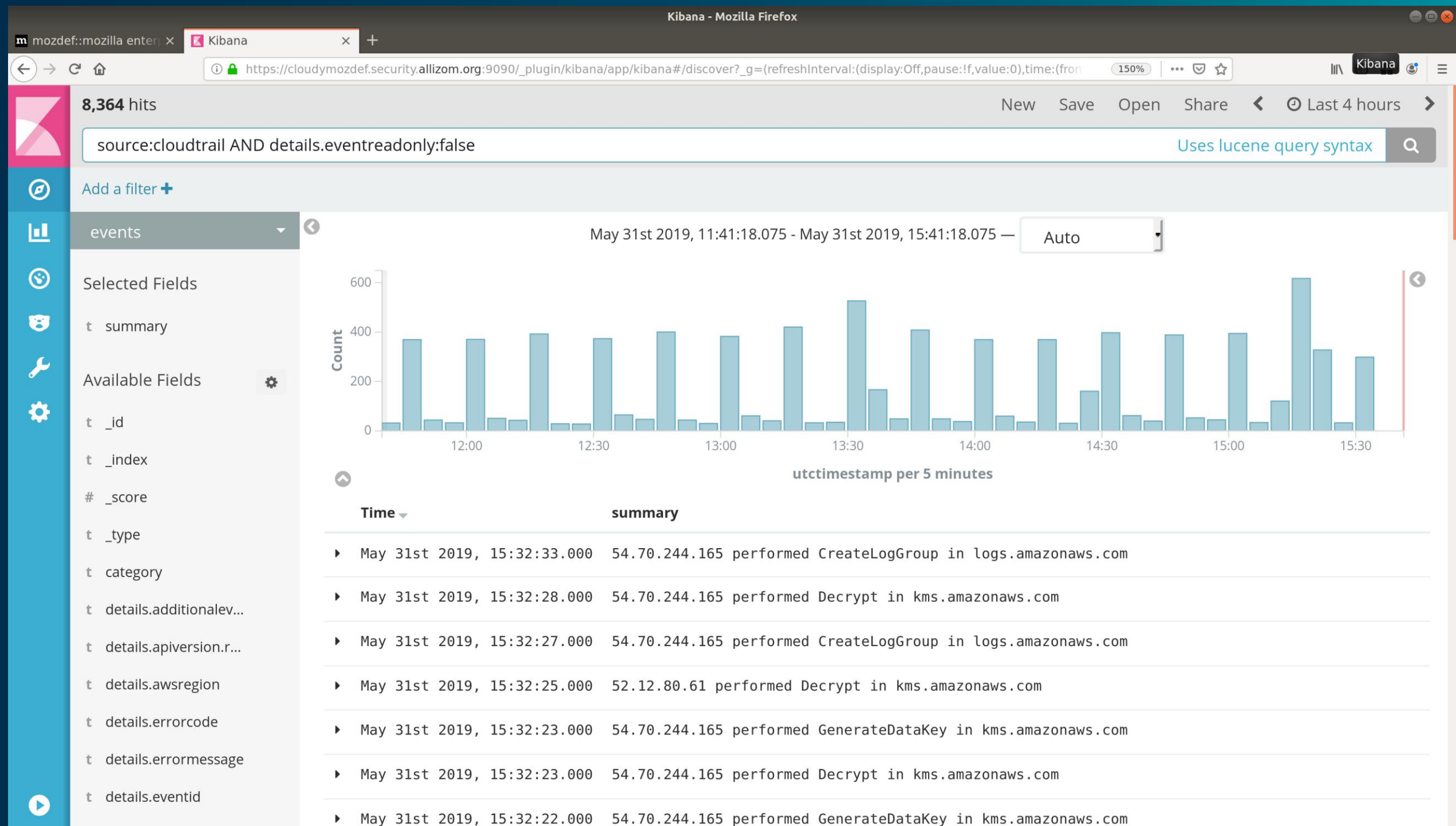
Live Show Cloud9 IDE for Alerts

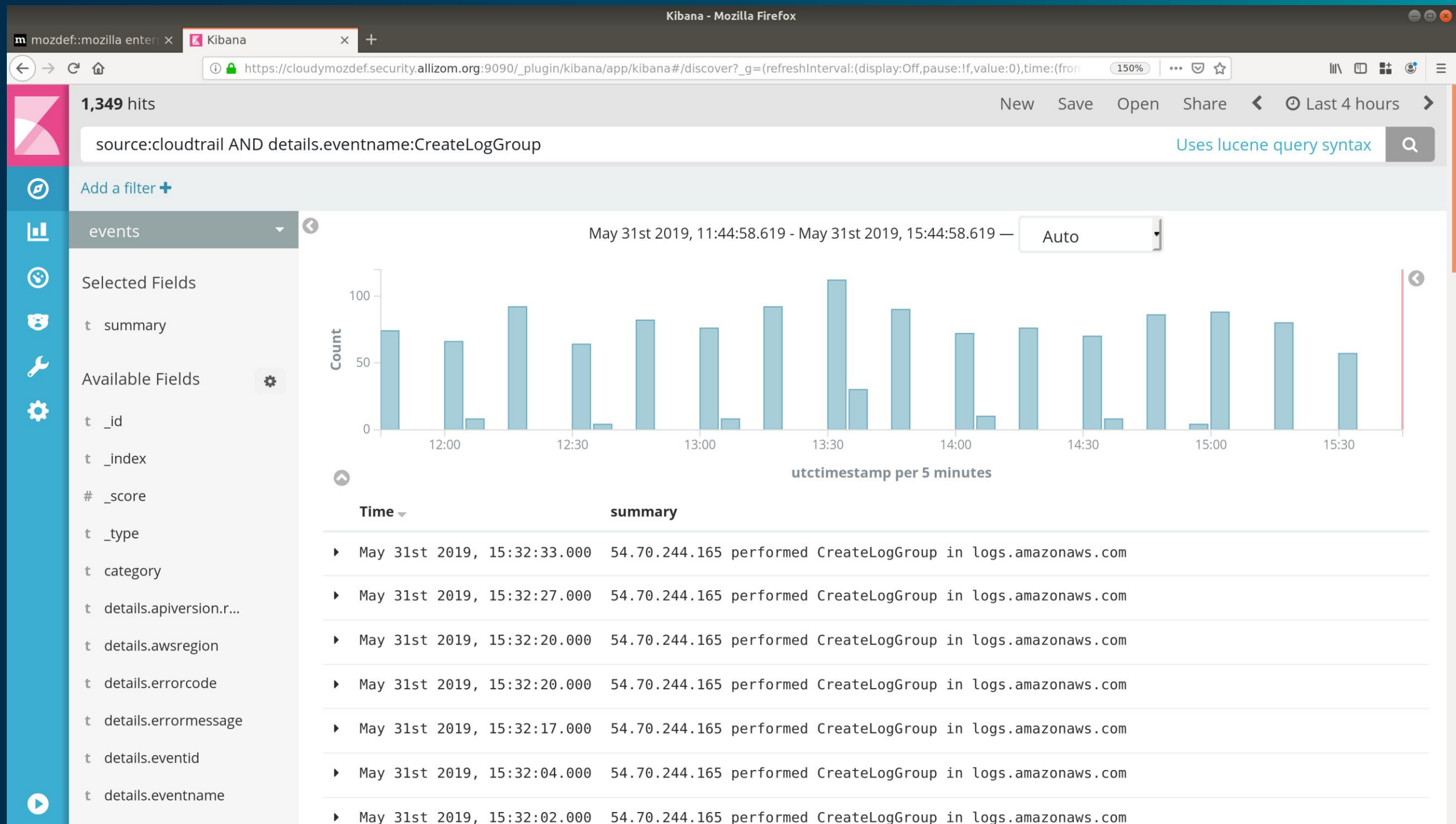
aws RE:INFORCE

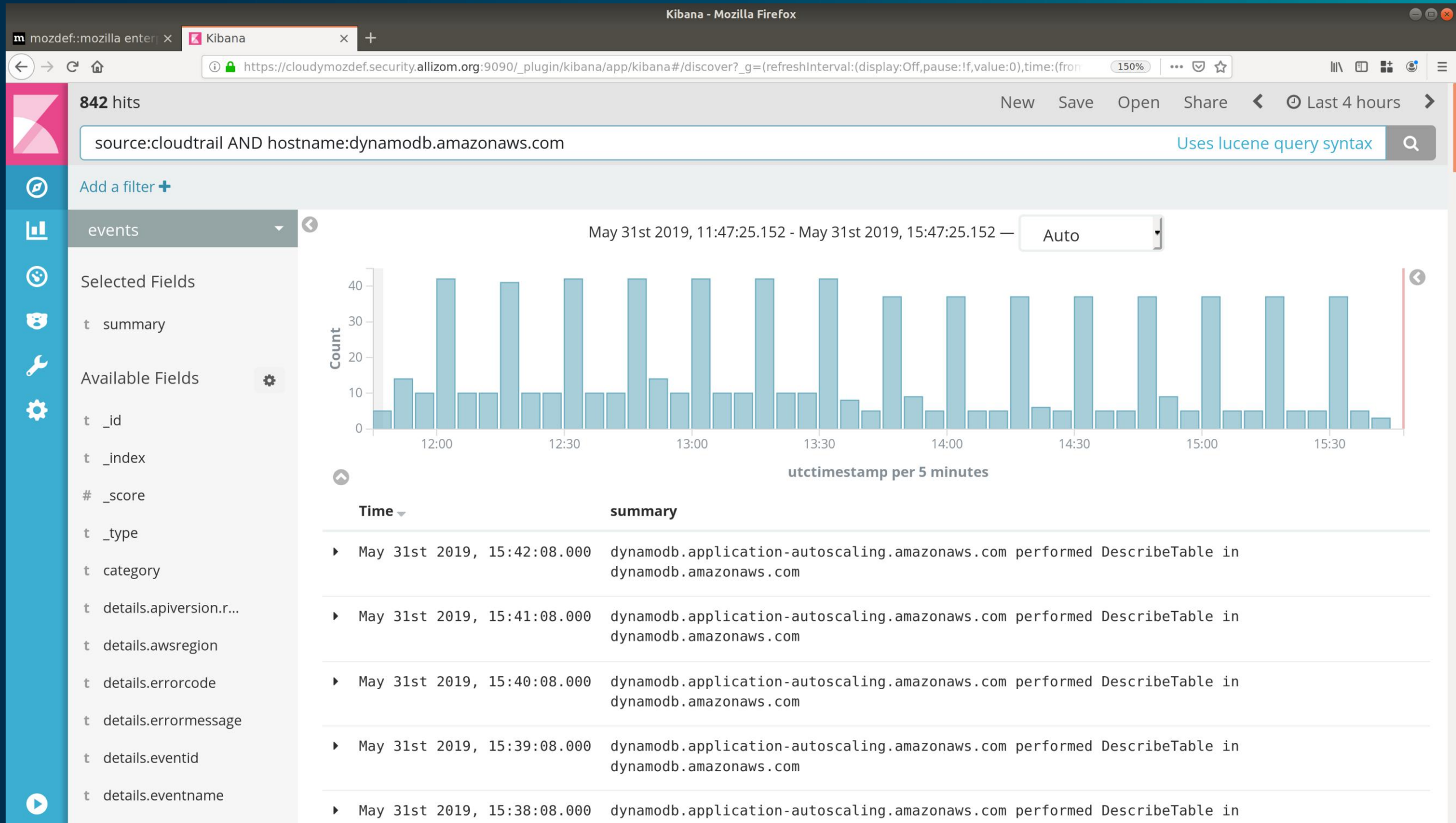
Investigating with Kibana

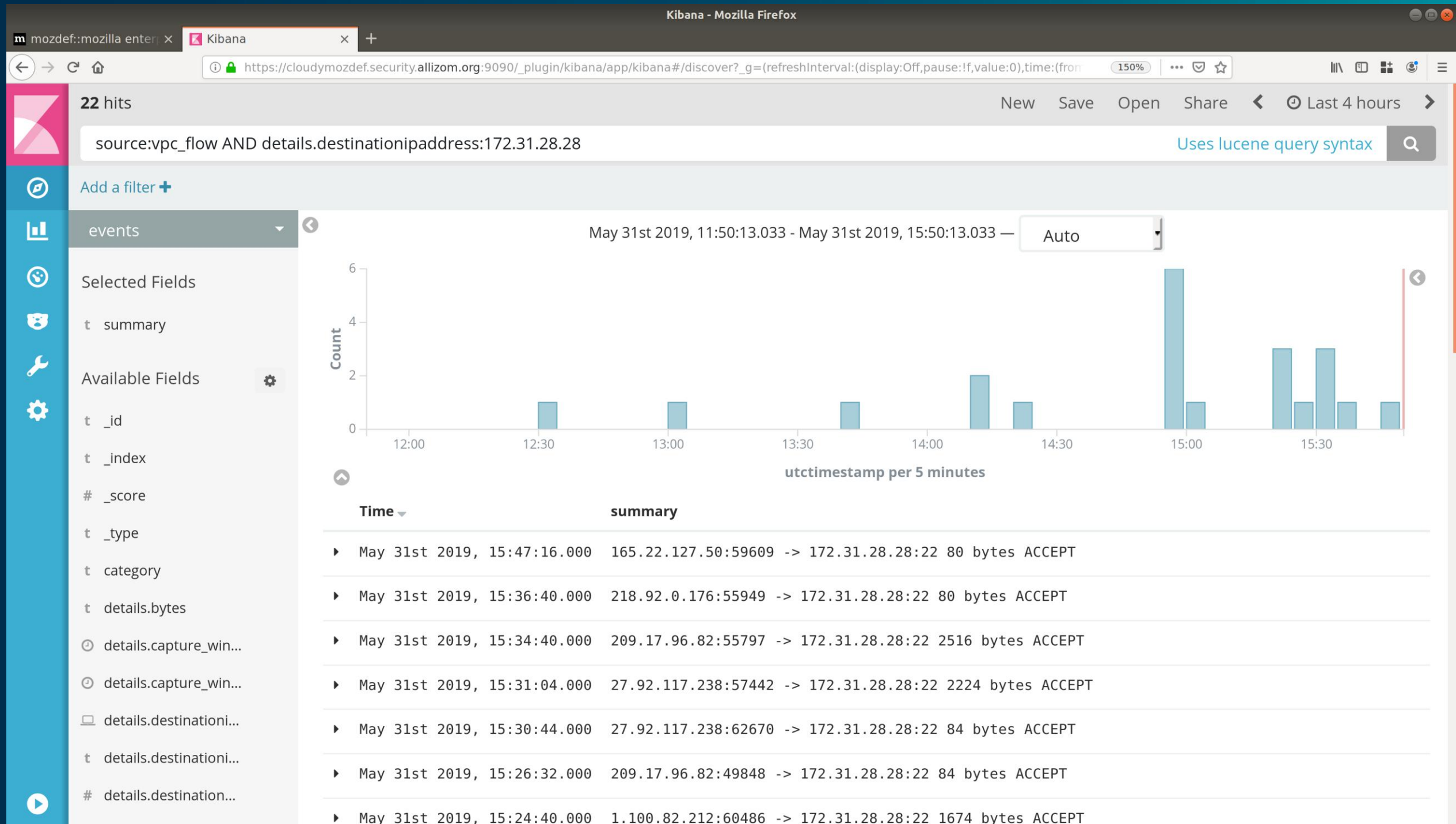
aws RE:INFORCE











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

mozdef::mozilla enterprise defense platform - Mozilla Firefox

mozdef::mozilla enter x Kibana x +

https://cloudmozdef.security.allizom.org/investigation/WHK8jPo4 110%

HOME KIBANA ALERTS INVESTIGATIONS INCIDENTS VISUALIZATIONS ABOUT GENE@MOZILLA.COM

save Escalate To Incident

Main Indicators Evidence Notes References Theories Timestamps Mitigations Lessons Learned

Summary: Possible data leak in fooproject.example.com

Description: A commit was pushed to the public GitHub repo https://github.com/octocat/Spoon-Knife containing a development API key.

Contact: jdoe@example.com

Date Opened: 05/31/2019 04:31:58 PM

Date Closed: Your Local Time

Phase: Identification

Tags: drag here to add a tag category.data disclosure

Timeline: 05/31/2019 04:32:38 PM Begin End

tag filter

common

category.data disclosure
category.data corruption
attribute.integrity.variety.Modify data
attribute.confidentiality.data_victim.Customer
attribute.confidentiality.data_victim.Employee
attribute.confidentiality.data_victim.Other
attribute.confidentiality.data_victim.Partner
attribute.confidentiality.data_victim.Patient
attribute.confidentiality.data_victim.Student
attribute.confidentiality.data_victim.Unknown
attribute.confidentiality.data_variety.Credentials
attribute.confidentiality.data_variety.Bank
attribute.confidentiality.data_variety.Classified
attribute.confidentiality.data_variety.Copyrighted
attribute.confidentiality.data_variety.Digital certificate
attribute.confidentiality.data_variety.Medical

mozdef::mozilla enterprise defense platform - Mozilla Firefox

mozdef::mozilla enter x Kibana

https://cloudymozdef.security.allizom.org/incident/RSCv4PBjJcFMSfrgf/edit

HOME KIBANA ALERTS INVESTIGATIONS INCIDENTS VISUALIZATIONS ABOUT GENE@MOZILLA.COM

save

Main Notes References Theories Timestamps Mitigations Lessons Learned

Summary: Data leak on fooproject.example.com

Description: A commit was pushed to the public GitHub repo https://github.com/octocat/Spoon-Knife containing a development API key.

Contact: jdoe@example.com

Date Opened: 05/31/2019 04:08:20 PM

Date Closed: Your Local Time

Phase: Identification

Tags: drag here to add a tag
category.data disclosure

Timeline:

| | |
|--|-----------|
| 05/31/2019 04:14:35 PM | Reported |
| Date Verified (your local time) | Verified |
| Date Mitigated (your local time) | Mitigated |
| Date of Permanent Resolution (your local time) | Contained |

tag filter

common

category.unauthorized access
category.data disclosure
category.data corruption
category.denial of service
category.lost or stolen device
category.unauthorized use
category.malware
category.widespread vulnerability
category.defacement
category.fraud
category.abuse
category.application compromise
category.other

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



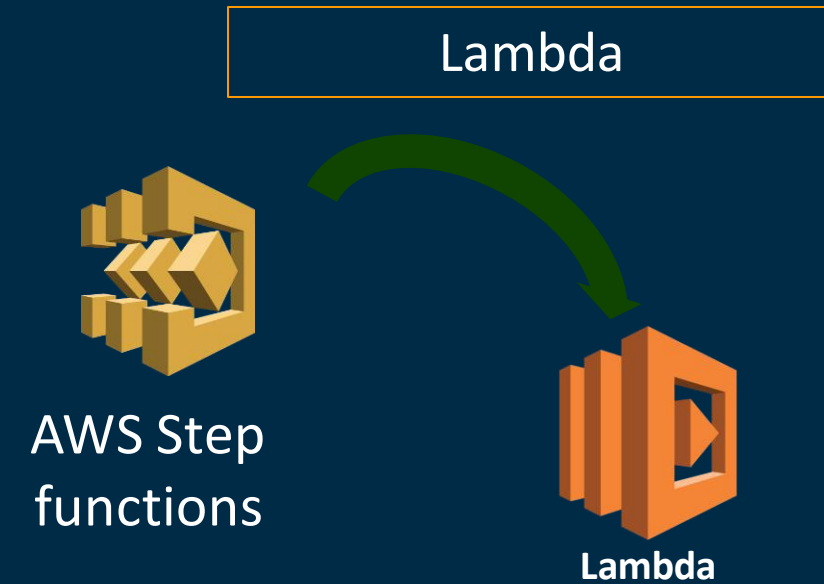
Responding to Findings: Remediation





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





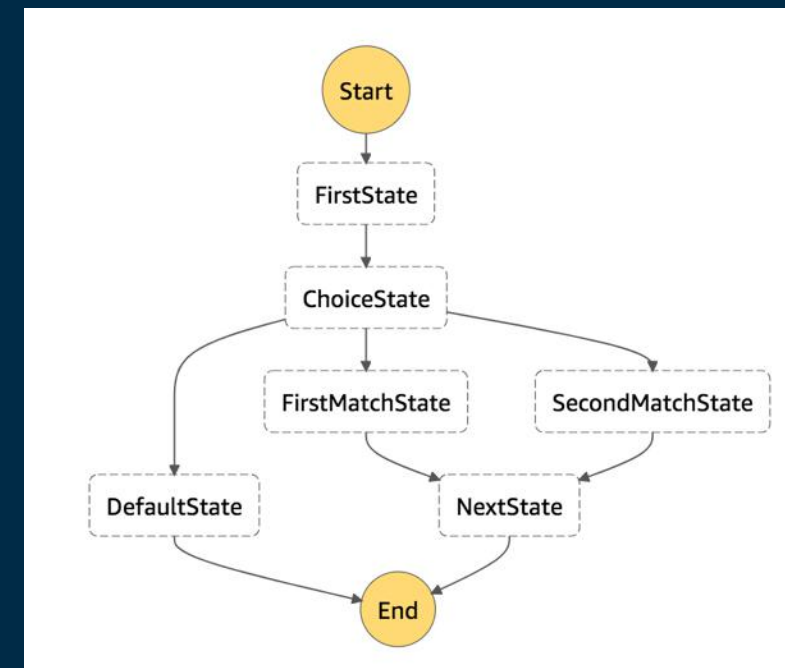
Responding to Findings: Automation Example

- **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





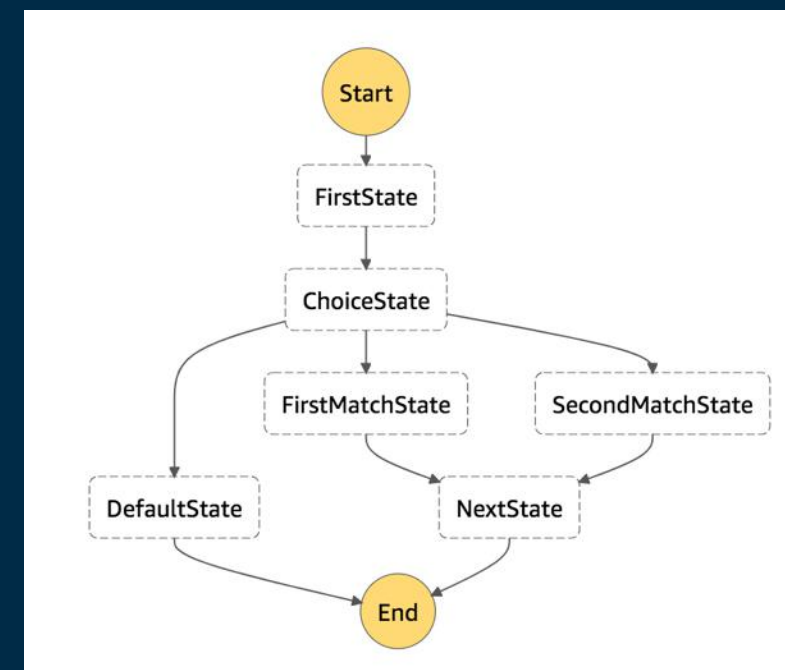
Responding to Findings: Automation Example

- 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": "Generated",
            "productCodeType": "Generated"
          }
        ]
      }
    }
  }
]
```

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

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 order

- 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 the

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 and

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.

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.

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 id
- 4.Snapshot instance and VPC Flow logs to foren
- 5.Validate that new ASG created instance is wor

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 environment

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.surveymzmo.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!