Incident Response on AWS

Dave Walker, Specialist Solutions Architect, Security and Compliance

01/08/19

Agenda

- "Expect to be Hacked"
- "Be Prepared"
- "That looks Weird..."
- Initial Response Actions
- "Decisions, Decisions..."
- Preserving Evidence, by Service
- Analysis
- Training your People
- Conclusions

aws

"Expect to be Hacked"

Start Here

https://d1.awsstatic.com/whitepapers/aws_security_incident_response.pdf

"Be Prepared..."

IR Principles

- Establish Goals
- Respond using available (pre-built) tools and services
- Know what you have and what you need
- Do things that scale
- Use redeployment mechanisms
- Iteratively automate the mundane
- Learn and improve your process

aws

“Be Prepared”

- Be Contactable by AWS
- Build an Incident Response Handbook
- Train your People
- Know who to Contact, about What and When

- CloudFormation Everything
- Maintain “Intellectual Property Holding”
  - Templates, golden AMIs, Database snapshots

aws

Be Contactable by AWS

- Ensure each AWS account has appropriate email addresses
    - Set them to point at mailing lists, not individuals
- Ditto phone numbers
    - Set them to point at (PABX-style) hunt groups, not individuals

aws

Segment your Environments

- Limit compromise scopes
- Keep transaction state in as few places as possible
- Back state up to different accounts, frequently

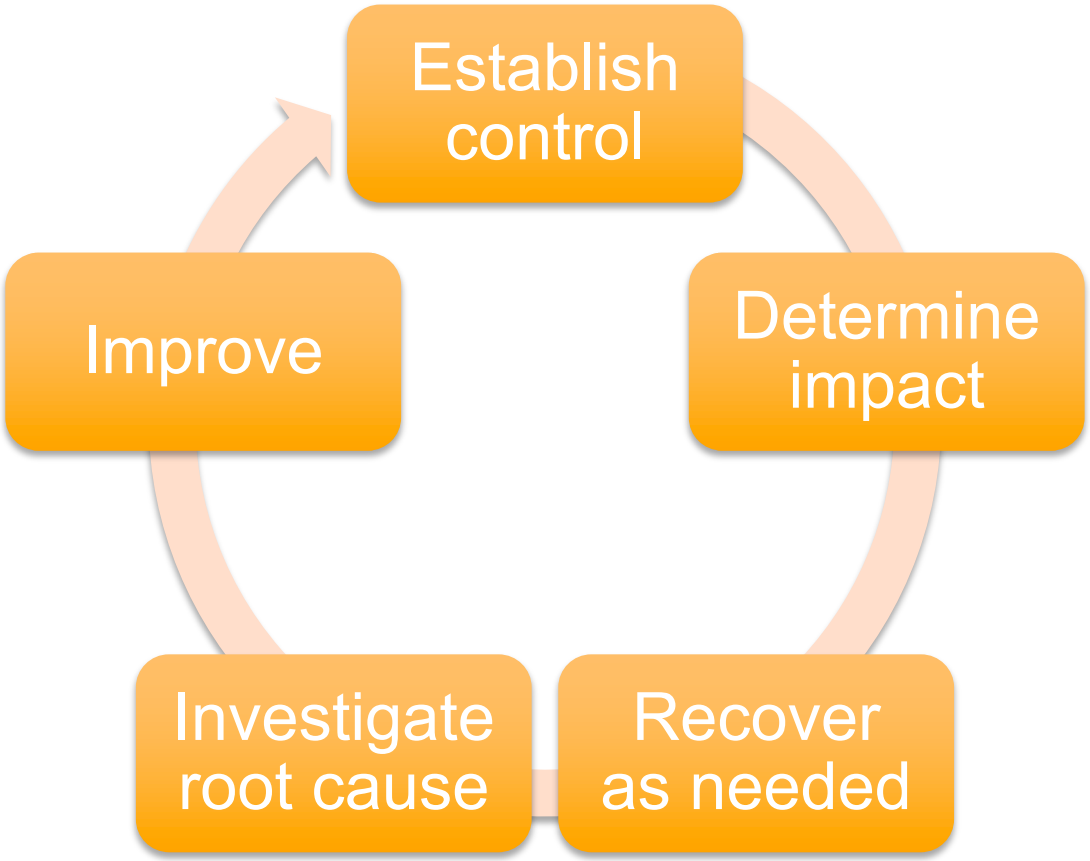
aws

Build an Incident Response Runbook

- The clue's in the name – it's a Book, Printed on Paper
  - …or, at least, a lever-arch file
  - …but still have centralised electronic copy under version control
- Where to start?
  - Set Policy
  - Eg: do you know what you're going to do, should you see each of the Findings which can be raised by Amazon GuardDuty?
    - https://docs.aws.amazon.com/guardduty/latest/ug/guarddut y_finding-types-active.html

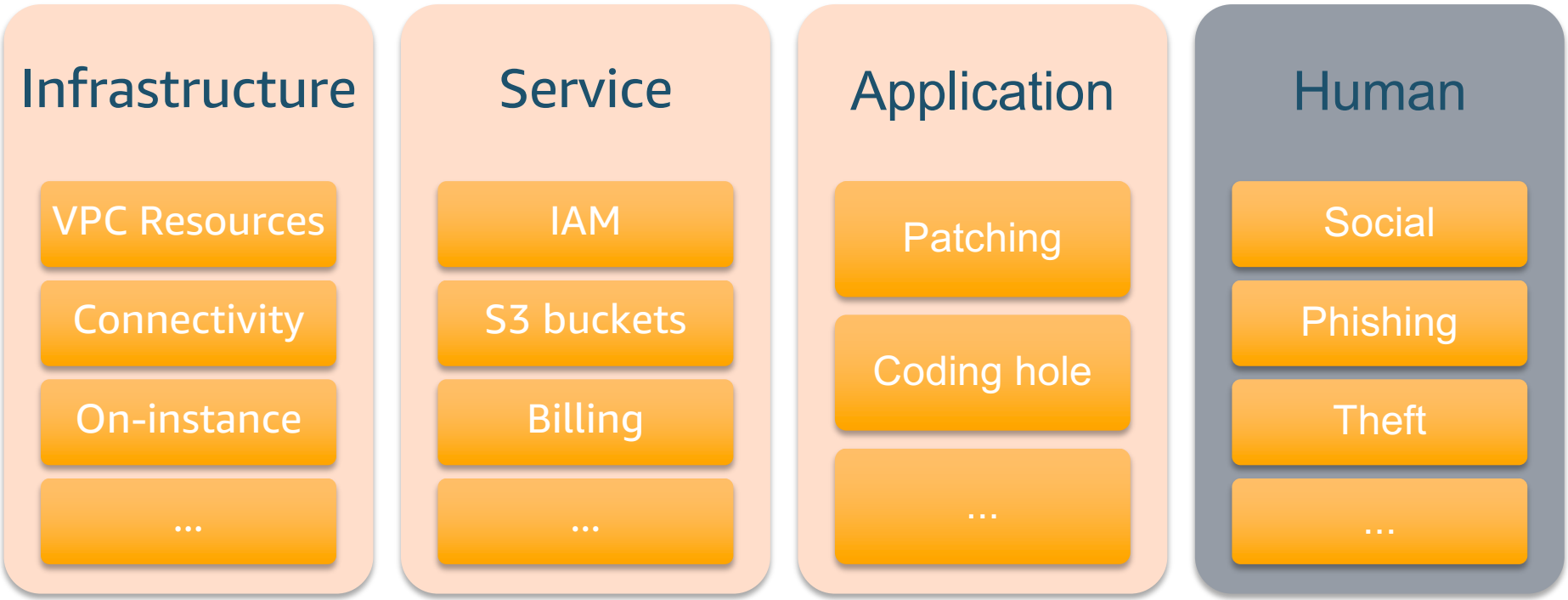
aws

IR Lifecycle



Understanding Your Attack Surface

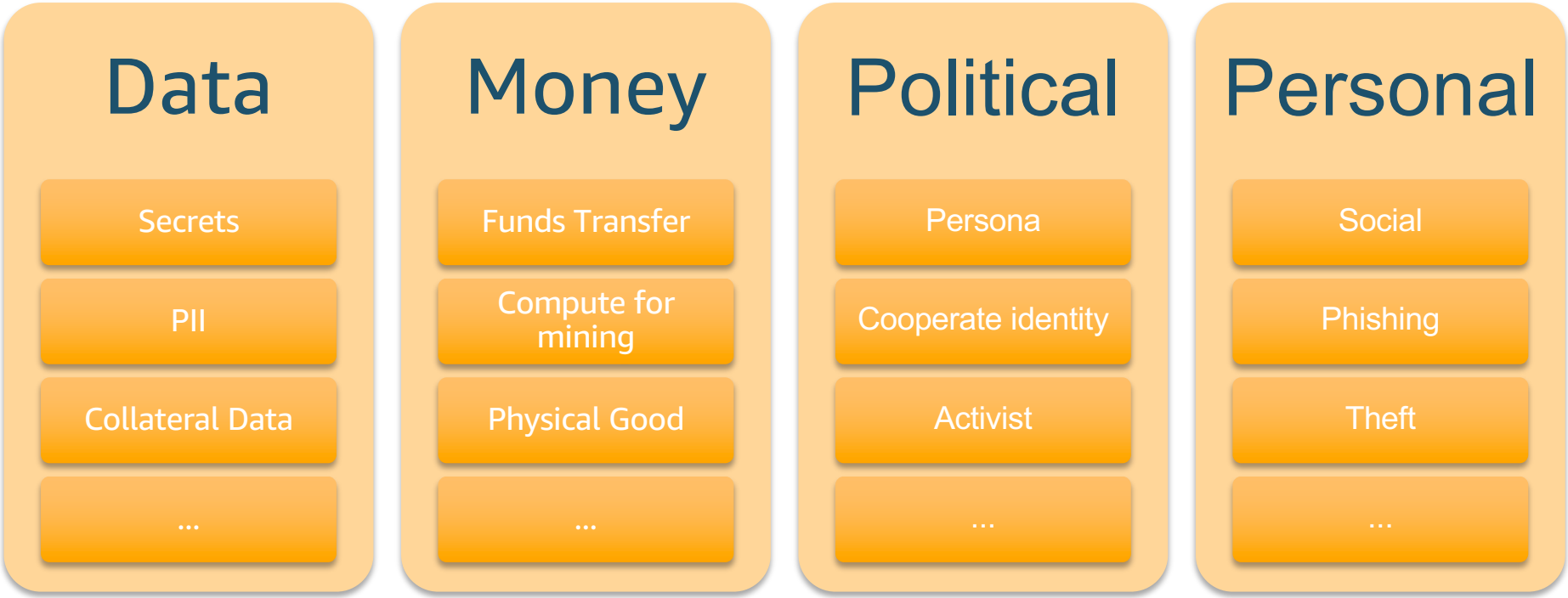
Incident Response Domains



aws

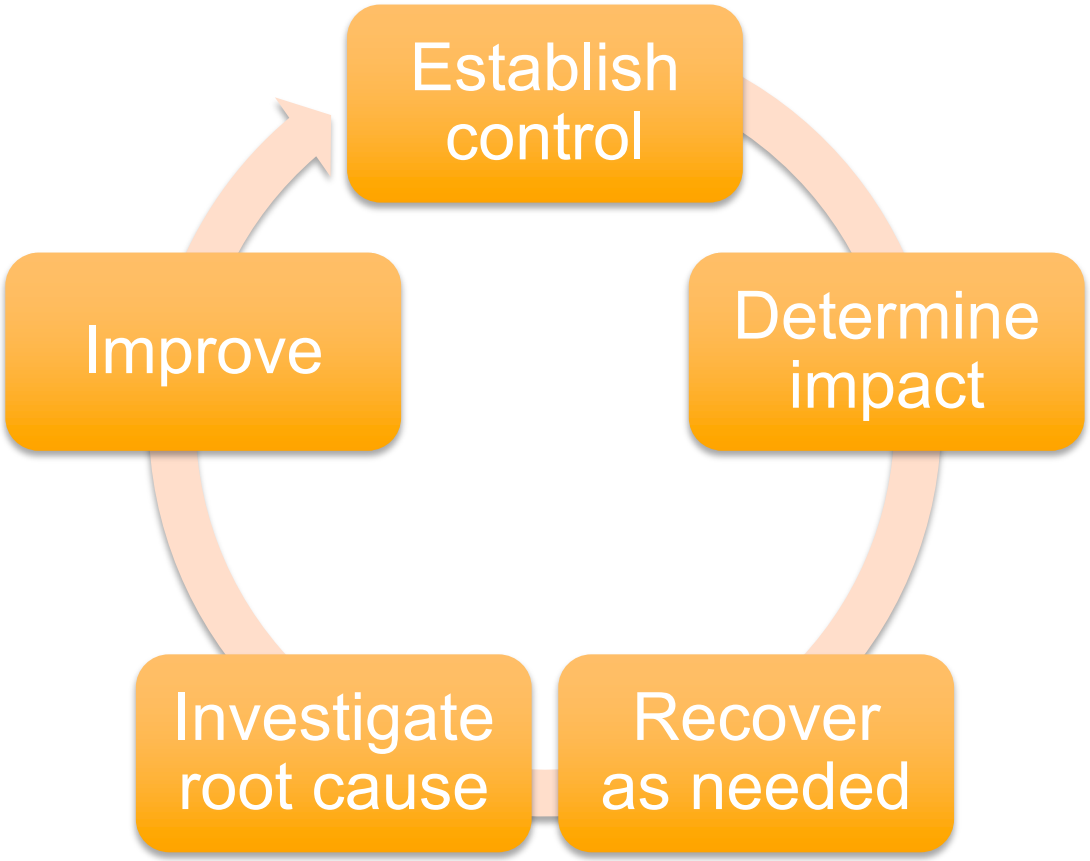
Understanding your critical assets

Working backwards from your “customer”

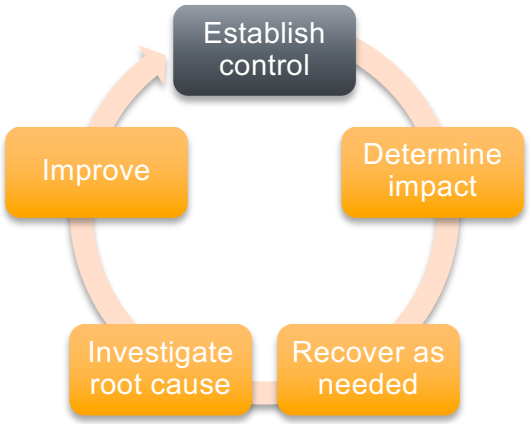


aws

IR Lifecycle



IR Lifecycle

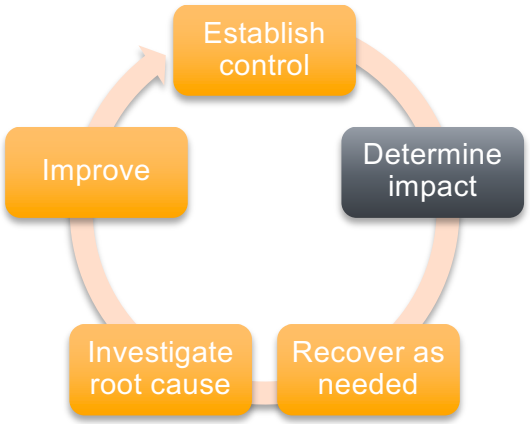


Establish control

- Can I Log into the Console?
- Can I log into the Instance?
- Can I review/copy logs/Cloudtrail?
- Can I copy information to a forensics account?
- Can I rotate credentials?
- Can I review Billing?
- Can I isolate the Instance?

aws

IR Lifecycle

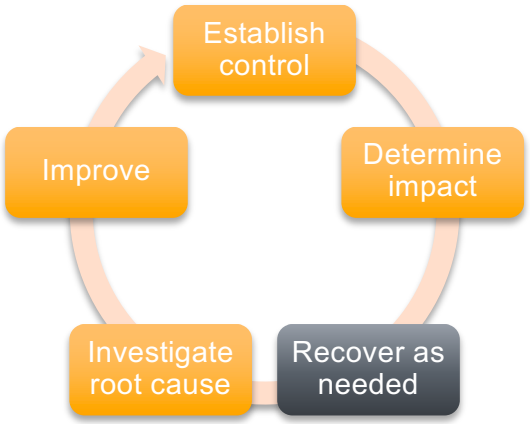


Determine impact

- Review logs/CloudTrail/VPC Flow Logs for changes
- Reviews Account resources
- Review Billing
- Review Access Permissions
- Review Data Loss and targets

aws

IR Lifecycle

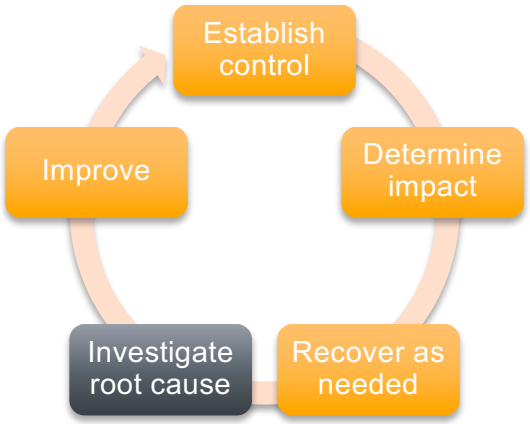


Recover as needed

- Do I remove instances?
- Do I change Security Groups?
- Do I remove IAM / Instance User(s)?
- Do I change credentials?
- Do I recover Security Groups?

aws

IR Lifecycle

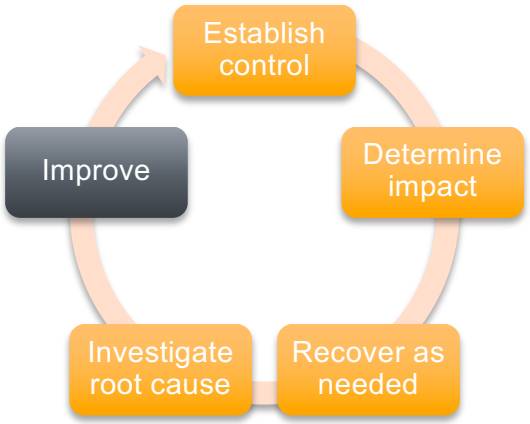


Investigate root cause

- How did this happen?
- Why did this happen?
- Who did it?
- How can we stop it from happening again?

aws

IR Lifecycle



Improve

- Improve our systems and processes
- Iterate.
- Iterate.
- Iterate.
- Iterate.

aws

Your Runbook!

Critical Threats

Infrastructure

- Backdoor:EC2/XORDDOS
-
- ...

Service

- **Stealth:IAMUser/CloudTrailLoggingDisabled**
-
- ...

Application

- **Recon:IAMUser/MaliciousIPCaller**
-
- ...

Informational/Other Threats

Infrastructure

- **UnauthorizedAccess:EC2/RDPBruteForce**
- Recon:EC2/PortProbeUnprotectedPort
- ...

Service

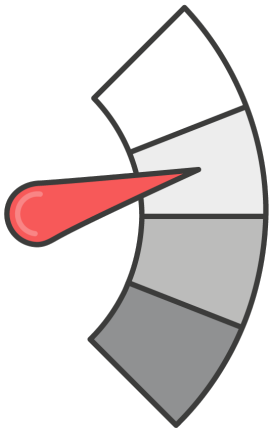
- **Persistence:IAMUser/UserPermissions**
-
- ...

Application

- **Recon:IAMUser/MaliciousIPCaller**
- **Recon:EC2/Portscan**
- ...

aws

Detective Control Feeds



- Logs and Monitors
- Billing Activity
- Threat Intelligence
- AWS Outreach
- Ad Hoc Contact

“If it Moves, Log It…”

- AWS CloudTrail
- AWS Config
- Amazon CloudWatch Logs
- Amazon VPC Flow Logs
- ELB logs
- Amazon API Gateway Logs
- Amazon Redshift Logs
- …

aws

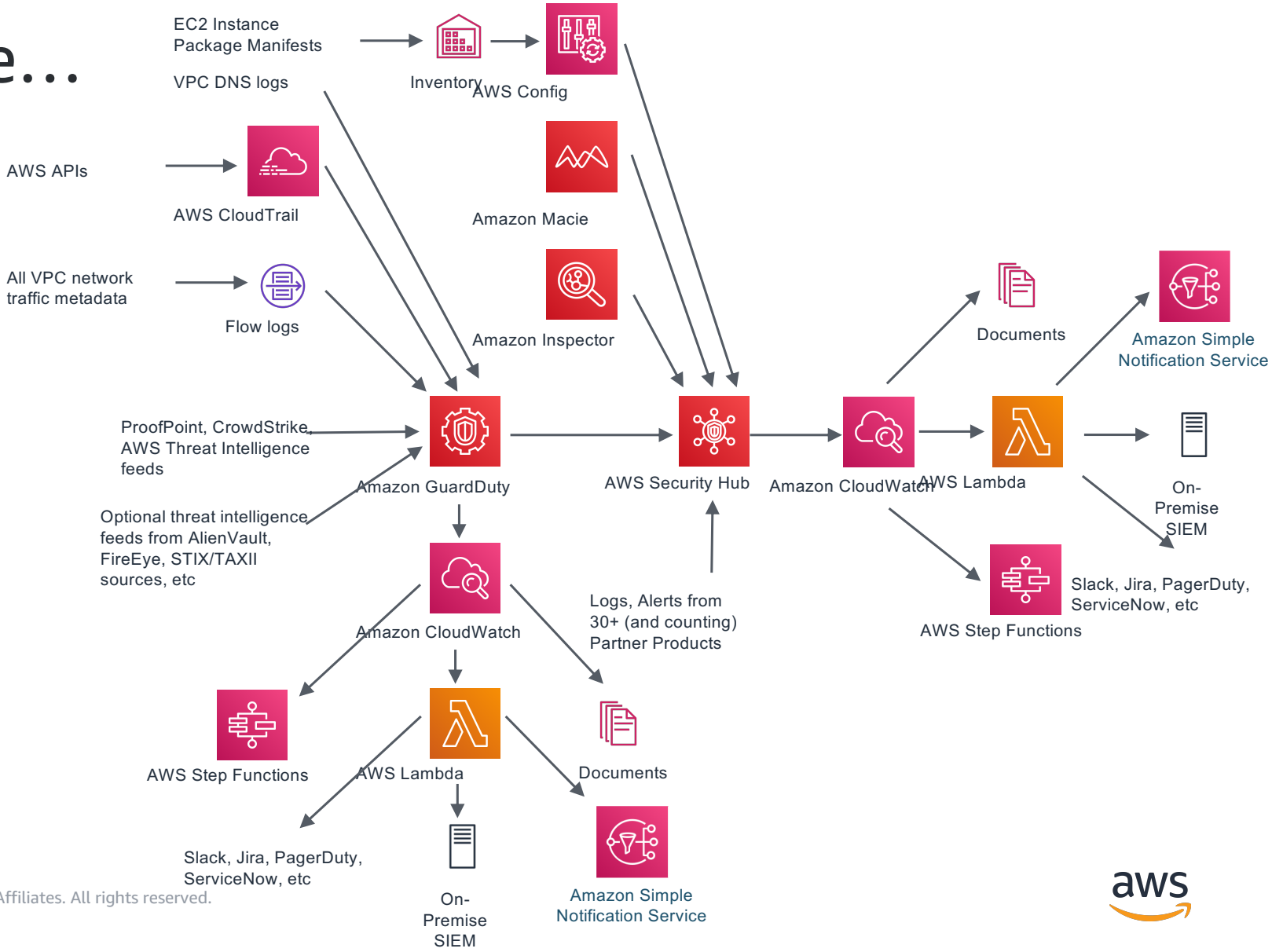
Don't Forget DDoS!

- AWS Shield (Advanced)
- Amazon Route53
- WAF (AWS and / or 3rd party)
- Anti-bot solutions
    - …to deal with credential-stuffing, etc
    - …from folk such as https://datadome.co/

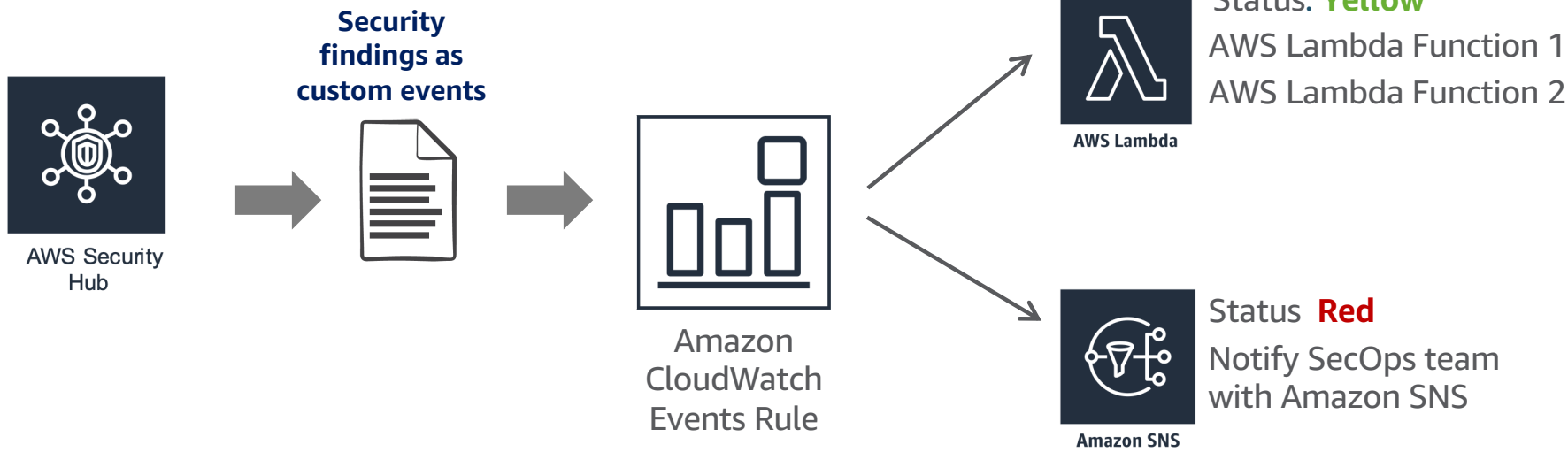
aws

"That Looks Weird…"

Making Sense…



Use Case: Alert Triage



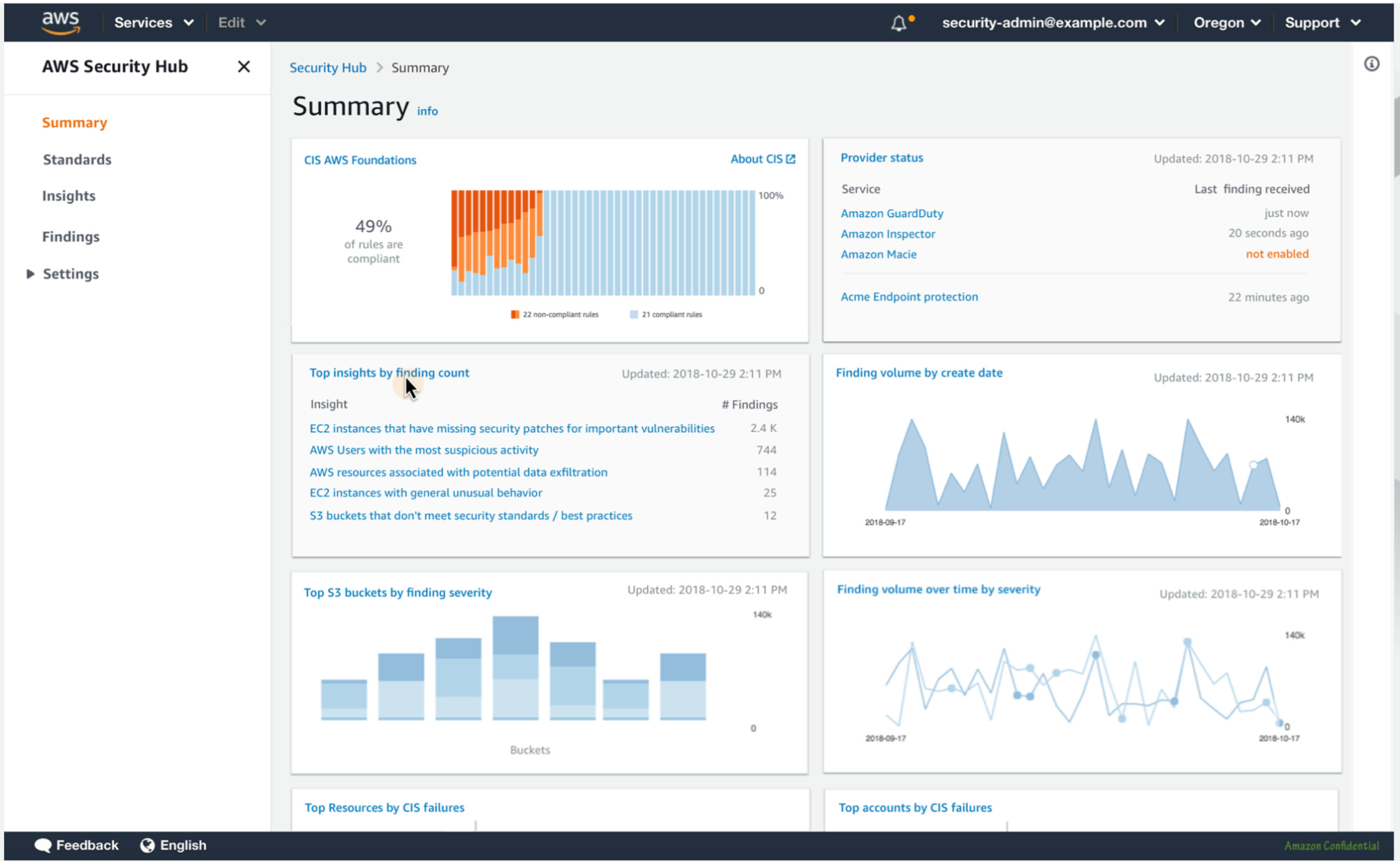
"That Looks Weird…"

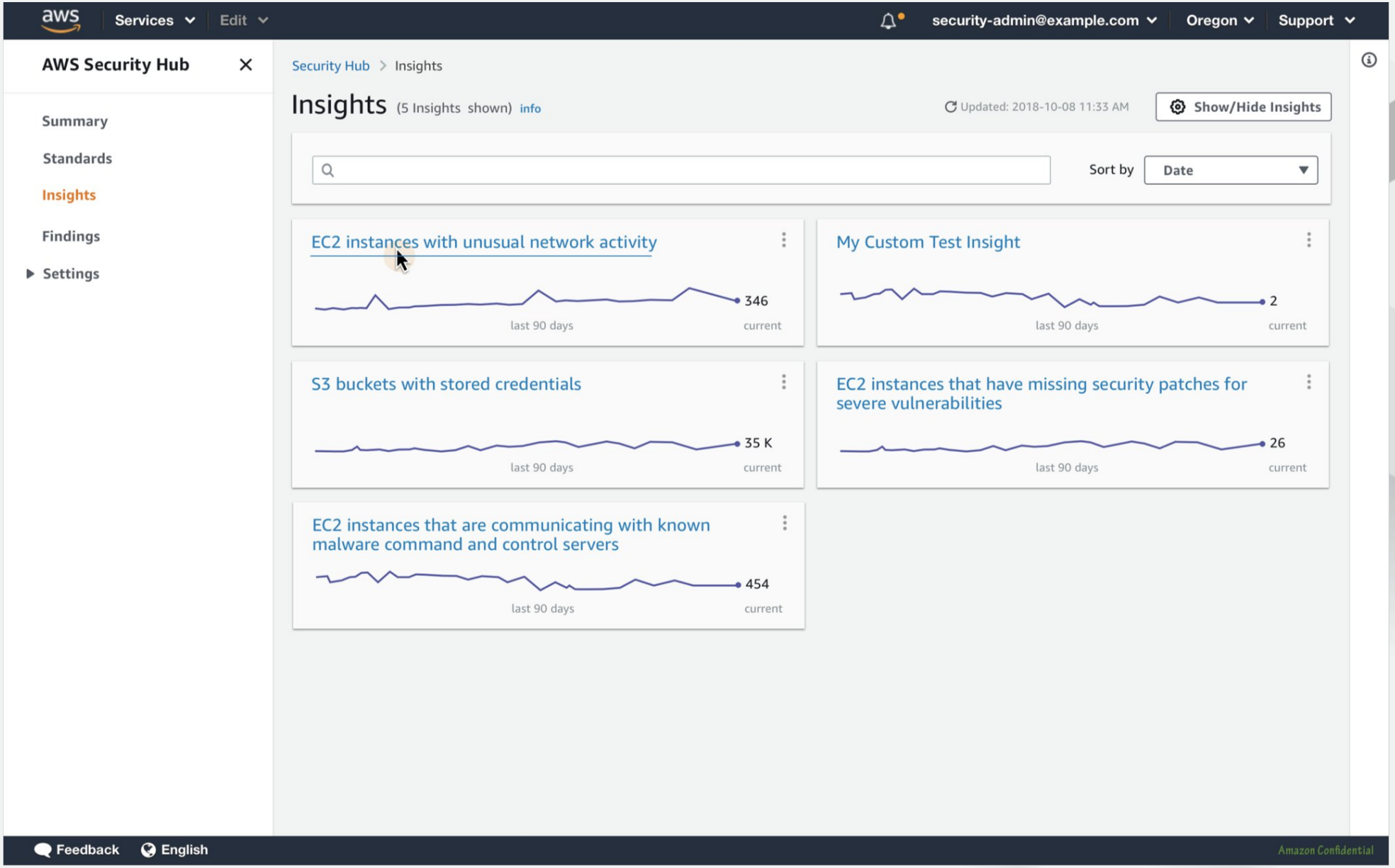
- Triage
- What can we Automate ("SOAR")?
  - See (eg) https://docs.aws.amazon.com/config/latest/developerguide/remediation.html
  - …or products from partners:
    - Turbot, Demisto, Armor…
  - …or https://github.com/cloud-custodian/cloud-custodian
  - …or…

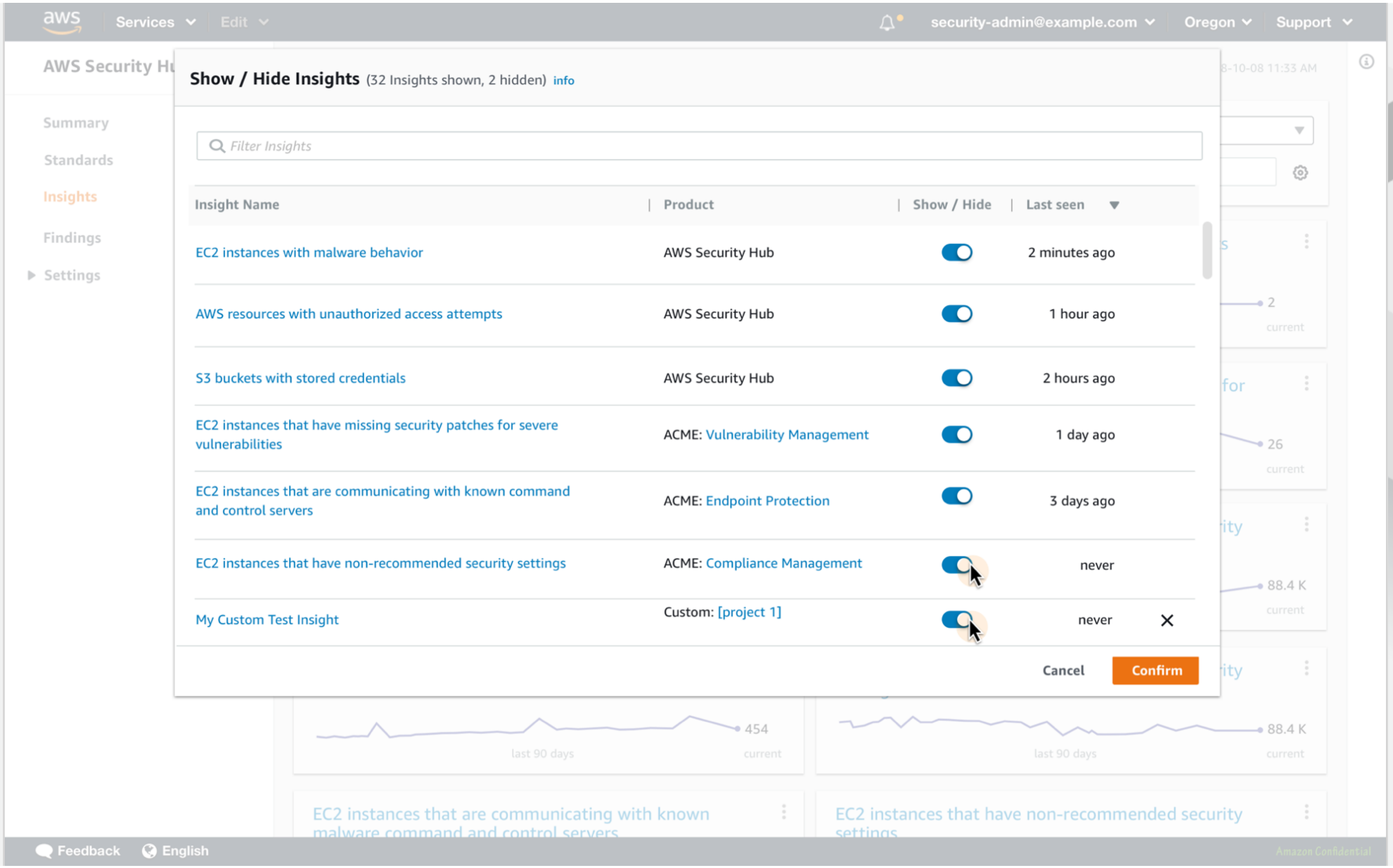
aws

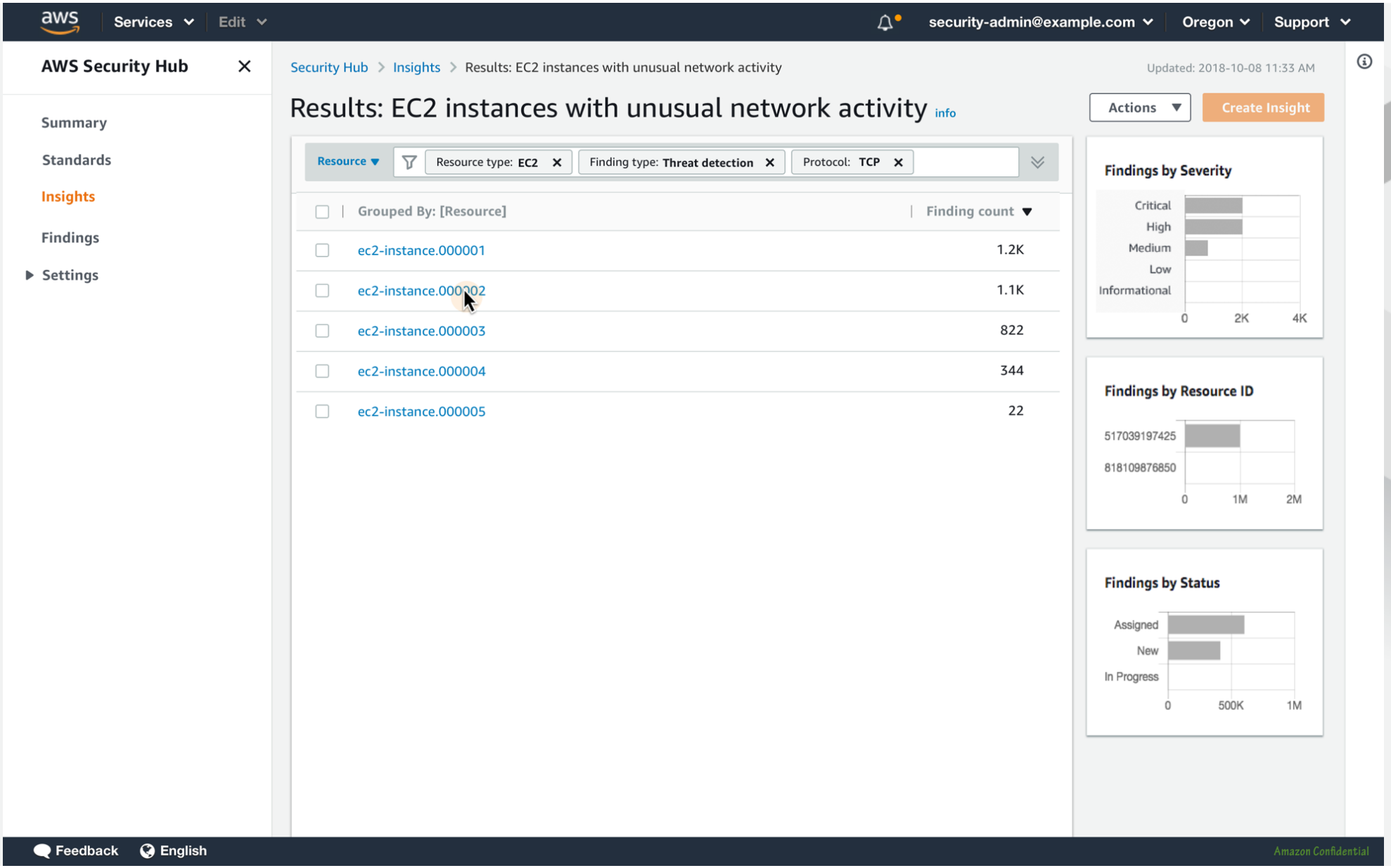
Alert Triage

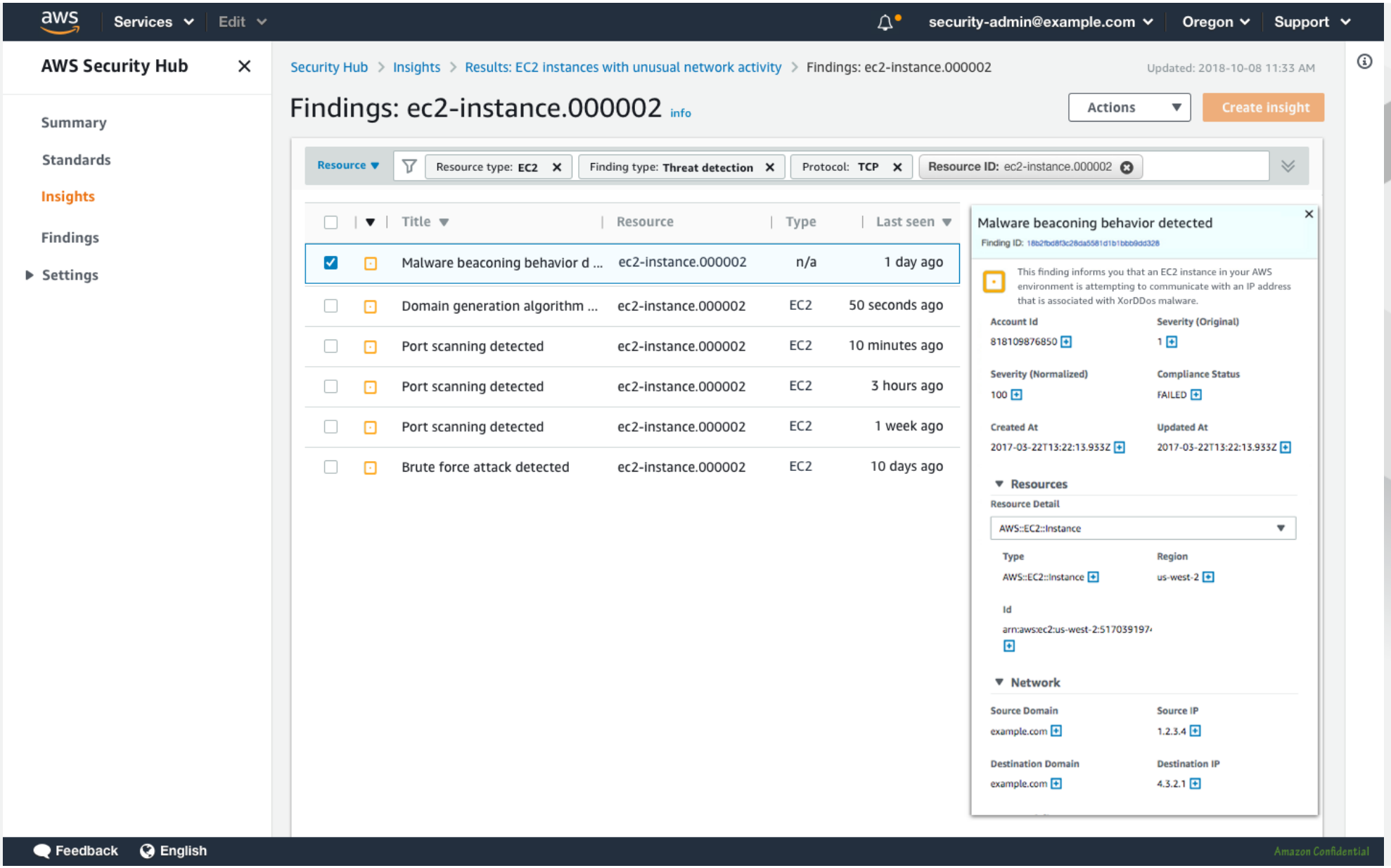
- AWS Security Finding Format ("ASFF")
  - https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-findings-format.html


aws
Services
Edit
security-admin@example.com
Oregon
Support
AWS Security Hub
Summary
Standards
Insights
Findings
Settings
Security Hub > Summary
Summary info
CIS AWS Foundations
About CIS
49%
of rules are compliant
100%
0
22 non-compliant rules
21 compliant rules
Provider status
Updated: 2018-10-29 2:11 PM
Service
Last finding received
Amazon GuardDuty
just now
Amazon Inspector
20 seconds ago
Amazon Macie
not enabled
Acme Endpoint protection
22 minutes ago
Top insights by finding count
Updated: 2018-10-29 2:11 PM
Insight
Findings
EC2 instances that have missing security patches for important vulnerabilities
2.4 K
AWS Users with the most suspicious activity
744
AWS resources associated with potential data exfiltration
114
EC2 instances with general unusual behavior
25
S3 buckets that don't meet security standards / best practices
12
Finding volume by create date
Updated: 2018-10-29 2:11 PM
140k
0
2018-09-17
2018-10-17
Top S3 buckets by finding severity
Updated: 2018-10-29 2:11 PM
140k
0
Buckets
Finding volume over time by severity
Updated: 2018-10-29 2:11 PM
140k
0
2018-09-17
2018-10-17
Top Resources by CIS failures
Top accounts by CIS failures
Feedback
English
Amazon Confidential


aws
Services
Edit
security-admin@example.com
Oregon
Support
AWS Security Hub
Summary
Standards
Insights
Findings
Settings
Security Hub > Insights
Insights (5 Insights shown) info
Updated: 2018-10-08 11:33 AM
Show/Hide Insights
Sort by
Date
EC2 instances with unusual network activity
346
last 90 days
current
My Custom Test Insight
2
last 90 days
current
S3 buckets with stored credentials
35 K
last 90 days
current
EC2 instances that have missing security patches for severe vulnerabilities
26
last 90 days
current
EC2 instances that are communicating with known malware command and control servers
454
last 90 days
current
Feedback
English
Amazon Confidential


aws
Services
Edit
security-admin@example.com
Oregon
Support
AWS Security Hu
Summary
Standards
Insights
Findings
Settings
Show / Hide Insights (32 Insights shown, 2 hidden) info
Filter Insights
Insight Name
Product
Show / Hide
Last seen
EC2 instances with malware behavior
AWS Security Hub
2 minutes ago
AWS resources with unauthorized access attempts
AWS Security Hub
1 hour ago
S3 buckets with stored credentials
AWS Security Hub
2 hours ago
EC2 instances that have missing security patches for severe vulnerabilities
ACME: Vulnerability Management
1 day ago
EC2 instances that are communicating with known command and control servers
ACME: Endpoint Protection
3 days ago
EC2 instances that have non-recommended security settings
ACME: Compliance Management
never
My Custom Test Insight
Custom: [project 1]
never
Cancel
Confirm
454
last 90 days
current
88.4 K
last 90 days
current
EC2 instances that are communicating with known malware command and control servers
EC2 instances that have non-recommended security settings
Feedback
English
Amazon Confidential


aws
Services
Edit
security-admin@example.com
Oregon
Support
AWS Security Hub
Summary
Standards
Insights
Findings
Settings
Security Hub > Insights > Results: EC2 instances with unusual network activity
Updated: 2018-10-08 11:33 AM
Results: EC2 instances with unusual network activity info
Actions
Create Insight
Resource
Resource type: EC2
Finding type: Threat detection
Protocol: TCP
Grouped By: [Resource]
Finding count
ec2-instance.000001 1.2K
ec2-instance.000002 1.1K
ec2-instance.000003 822
ec2-instance.000004 344
ec2-instance.000005 22
Findings by Severity
Critical
High
Medium
Low
Informational
0
2K
4K
Findings by Resource ID
517039197425
818109876850
0
1M
2M
Findings by Status
Assigned
New
In Progress
0
500K
1M
Feedback
English
Amazon Confidential


aws
Services
Edit
security-admin@example.com
Oregon
Support
AWS Security Hub
Summary
Standards
Insights
Findings
Settings
Security Hub > Insights > Results: EC2 instances with unusual network activity > Findings: ec2-instance.000002
Updated: 2018-10-08 11:33 AM
Findings: ec2-instance.000002 info
Actions
Create insight
Resource
Resource type: EC2
Finding type: Threat detection
Protocol: TCP
Resource ID: ec2-instance.000002
Title
Resource
Type
Last seen
Malware beaconing behavior d ... ec2-instance.000002 n/a 1 day ago
Domain generation algorithm ... ec2-instance.000002 EC2 50 seconds ago
Port scanning detected ec2-instance.000002 EC2 10 minutes ago
Port scanning detected ec2-instance.000002 EC2 3 hours ago
Port scanning detected ec2-instance.000002 EC2 1 week ago
Brute force attack detected ec2-instance.000002 EC2 10 days ago
Malware beaconing behavior detected
Finding ID:
This finding informs you that an EC2 instance in your AWS environment is attempting to communicate with an IP address that is associated with XorDDos malware.
Account Id
818109876850
Severity (Original)
1
Severity (Normalized)
100
Compliance Status
FAILED
Created At
2017-03-22T13:22:13.933Z
Updated At
2017-03-22T13:22:13.933Z
Resources
Resource Detail
AWS::EC2::Instance
Type
AWS::EC2::Instance
Region
us-west-2
Id
Network
Source Domain
example.com
Source IP
1.2.3.4
Destination Domain
example.com
Destination IP
4.3.2.1
Feedback
English
Amazon Confidential

"Decisions, Decisions..."

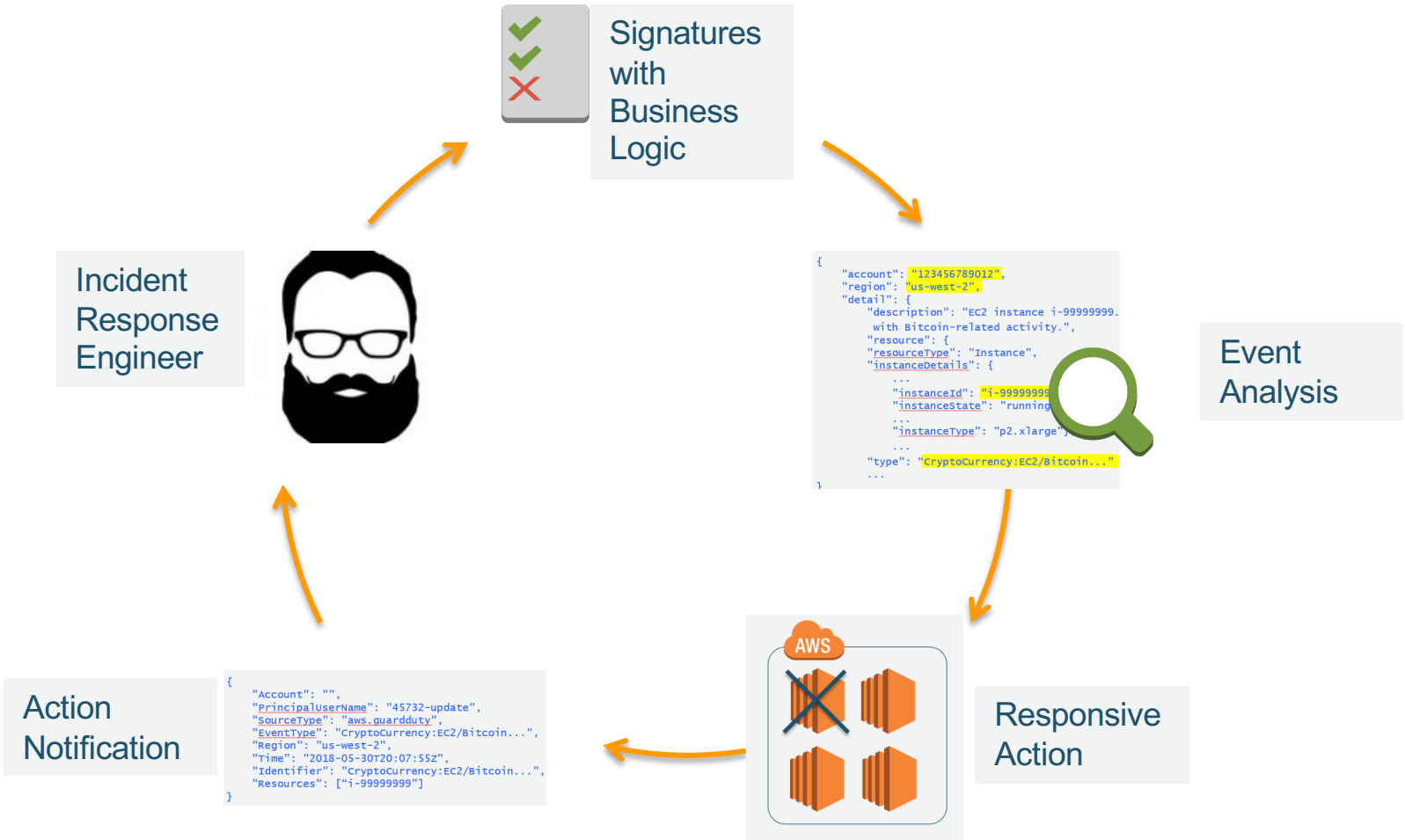
“Decisions, Decisions…”

- “Manage and Mitigate” or “Pursue and Prosecute”?
  - First closes open doors, but destroys evidence
  - Second (in traditional environments) is a lot more hassle
- Can we do both?
  - Maybe – but let's tackle the first one, first…

aws

Initial Response Actions ("Manage and Mitigate")

Introducing AERO

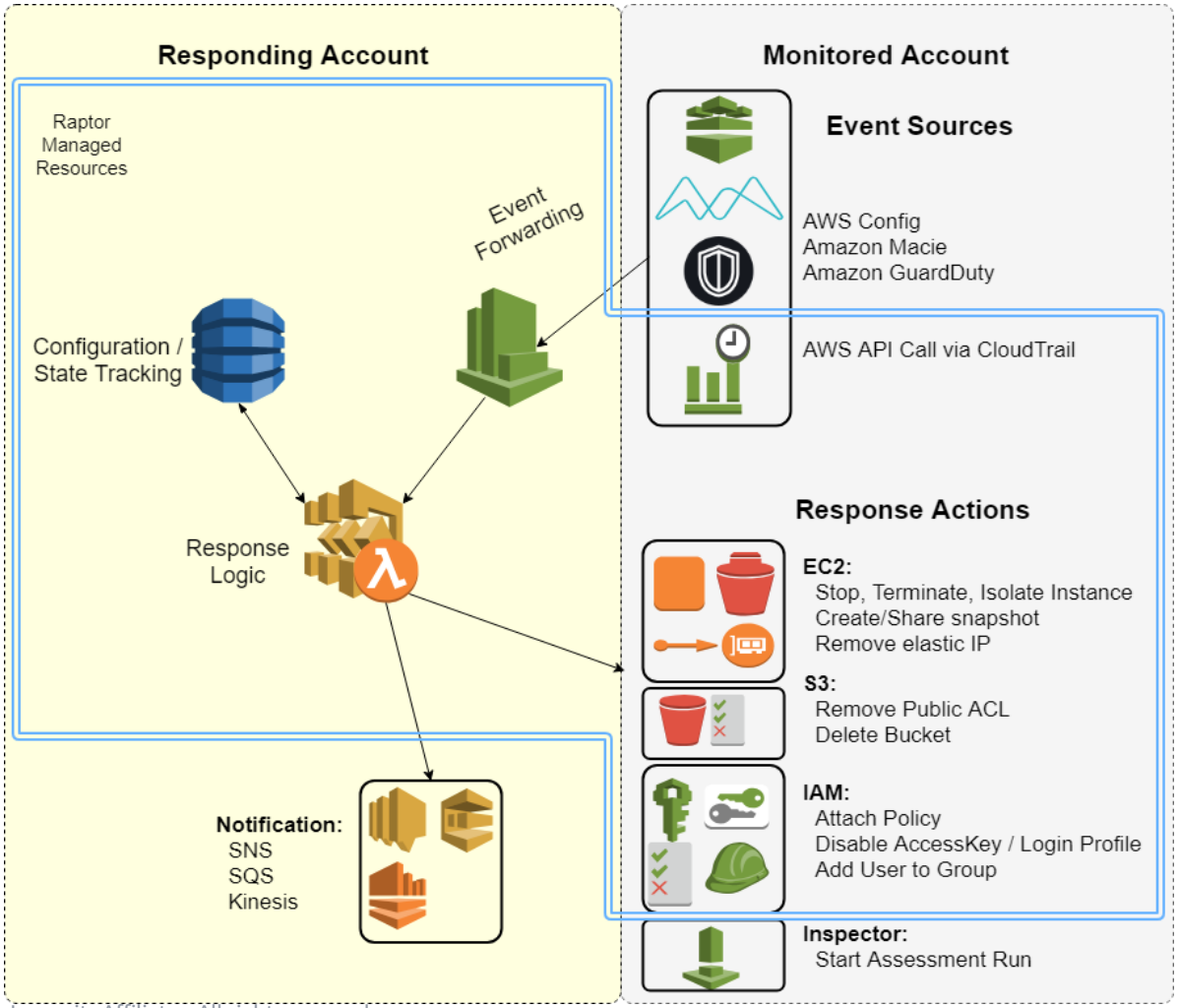


AERO Prerequisites

- An account with appropriate blast radius security controls to deploy and manage AERO.
- Approval to use AWS Services: CloudFormation, CloudWatch, DynamoDB, Lambda, S3, SNS, and Step Functions.
- AWS Services enabled and configured: CloudTrail, Config, GuardDuty, and (optionally) Macie.
- Python to perform initial install and generate artifacts.

aws

AERO Architecture



AERO Example

Scenario: IAM Access Key was used to provision EC2 instances with the intent of mining crypto-currency.

Detection: GuardDuty detects that an EC2 instance in your AWS environment is querying a domain name that is associated with Bitcoin-related activity.

Our desired response is to terminate the instances and notify an email distribution list for follow-up.

aws

AERO Example: CloudWatch Events

In the AERO Member Account, GuardDuty generates Finding and sends to CloudWatch Events. CloudWatch Events forwards to CloudWatch Event Bus in AERO Master Account.

```
{
    "account": "123456789012",
    "region": "us-west-2",
    "detail": {
        "description": "EC2 instance i-99999999 is querying a domain name that is associated
         with Bitcoin-related activity.",
        "resource": {
        "resourceType": "Instance",
        "instanceDetails": {
            ...
            "instanceId": "i-99999999",
            "instanceState": "running",
            ...
            "instanceType": "p2.xlarge”},
            ...
        "title": "Bitcoin-related domain name queried by EC2 instance i-99999999.",
        ...
}
```


aws

AERO Example: Lambda Trigger

In the AERO Master Account, CloudWatch Events triggers the Response Handler Lambda function to analyze the event by processing signature logic for conditional evaluation.

```
{
    "account": "123456789012",
    "region": "us-west-2",
    "detail": {
        "description": "EC2 instance i-99999999...
         with Bitcoin-related activity.",
        "resource": {
        "resourceType": "Instance",
        "instanceDetails": {
            ...
            "instanceId": "i-99999999",
            "instanceState": "running",
            ...
            "instanceType": "p2.xlarge"},
            ...
        “type”: “CryptoCurrency:EC2/Bitcoin...”
        ...
}
```

```
"account": "123456789012",
"region": "us-west-2",
"instanceId": "i-99999999",
“type”: “CryptoCurrency:EC2/Bitcoin...”
```

aws

AERO Example: Response Handler

In the AERO Master Account, the Response Handler Lambda function analyzes the event by processing conditional logic to determine responsive Action, which in this case is to terminate the instance(s).

```
"account": "123456789012",
"region": "us-west-2",
"instanceId": "i-99999999",
“type": “CryptoCurrency:EC2/Bitcoin...”
```

```
TerminateInstanceTest:
  cloudwatch.event:
    - name: guardduty
    - identifier: “CryptoCurrency:EC2/Bitcoin...”
    - actions:
      - "ec2:TerminateInstance"
    - onlyif:
      - and:
        - region: 'us-west-2'
        - or:
          - account: 123456789012
          - account: 123456789013
```


aws

AERO Example: Response Handler

In the AERO Master Account, if the event matches a signature the Response Handler Lambda function initiates a StepFunction execution of the Response Action State Machine.

```
TerminateInstanceTest:
  cloudwatch.event:
    - name: guardduty
    - identifier: “CryptoCurrency:EC2/Bitcoin...”
    - actions:
      - "ec2:TerminateInstance"
    - onlyif:
      - and:
        - region: 'us-west-2'
        - or:
          - account: 123456789012
          - account: 123456789013
```

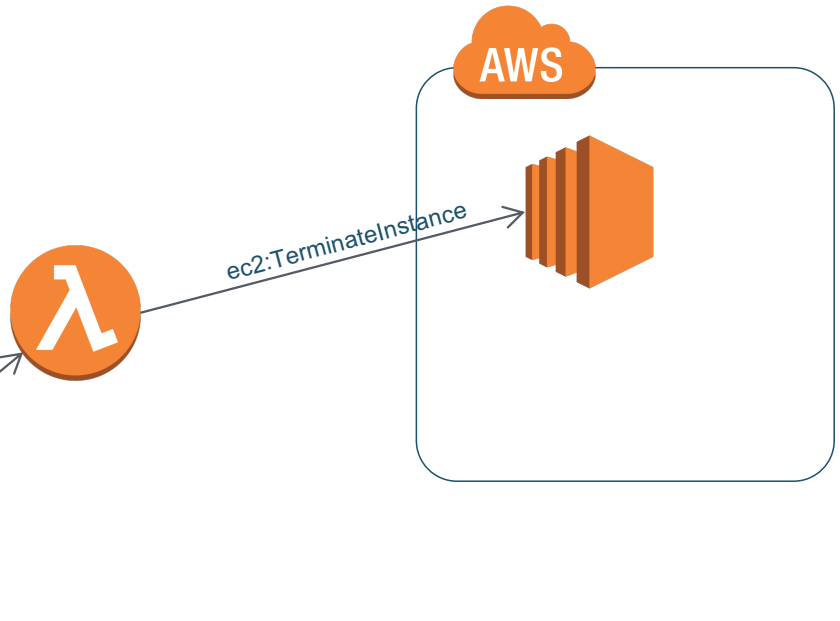
```
{
  "Account": “123456789012",
  "SnsNotification": true,
  "ec2": {
    "RemoveEip": false,
    "ApplySecurityGroup": false,
    "SecurityGroupName": null,
    "instanceId": [“i-99999999”],
    "region": “us-west-2”,
    "Snapshot": {
      "ShareSnap": null
    },
    "StopInstance": false,
    "CreateSnapshot": false,
    "TerminateInstance": true
  },
  "sns": {...}
}
```


aws

AERO Example: Response Handler

In the AEROMaster Account, the Response Action State Machine invokes a Lambda Response function to assume a role in the Member AWS Account to take the responsive action of terminating the EC2 instance(s).

```
{
  "Account": “123456789012",
  "SnsNotification": true,
  "ec2": {
    "RemoveEip": false,
    "ApplySecurityGroup": false,
    "SecurityGroupName": null,
    "instanceId": [“i-99999999”],
    "region": “us-west-2”,
    "Snapshot": {
      "ShareSnap": null
    },
    "StopInstance": false,
    "CreateSnapshot": false,
    "TerminateInstance": true
  },
  "sns": {...}
}
```



AERO Example: Response Handler

In the AERO Master Account, the Response Action State Machine concludes by triggering Logging Handler Lambda Functions to log and notify.

```
{
  "Account": “123456789012",
  "SnsNotification": true,
  "ec2": {
    "RemoveEip": false,
    "ApplySecurityGroup": false,
    "SecurityGroupName": null,
    "instanceId": [“i-99999999”],
    "region": “us-west-2”,
    "Snapshot": {
      "ShareSnap": null
    },
    "StopInstance": false,
    "CreateSnapshot": false,
    "TerminateInstance": true
  },
  "sns": {...}
}
```

```
{
    "Account": "",
    "PrincipalUserName": "45732-update",
    "SourceType": "aws.guardduty",
    "EventType": "CryptoCurrency:EC2/Bitcoin...",
    "Region": "us-west-2",
    "Time": "2018-05-30T20:07:55Z",
    "Identifier": "CryptoCurrency:EC2/Bitcoin...",
    "Resources": [“i-99999999”]
}
```


aws

Initial Response Actions ("Pursue and Prosecute")

“Decisions, Decisions…”

- “Manage and Mitigate” or “Pursue and Prosecute”?
  - First closes open doors, but destroys evidence
  - Second (in traditional environments) is a lot more hassle
- Can we do both?

aws

Get the Right People Involved

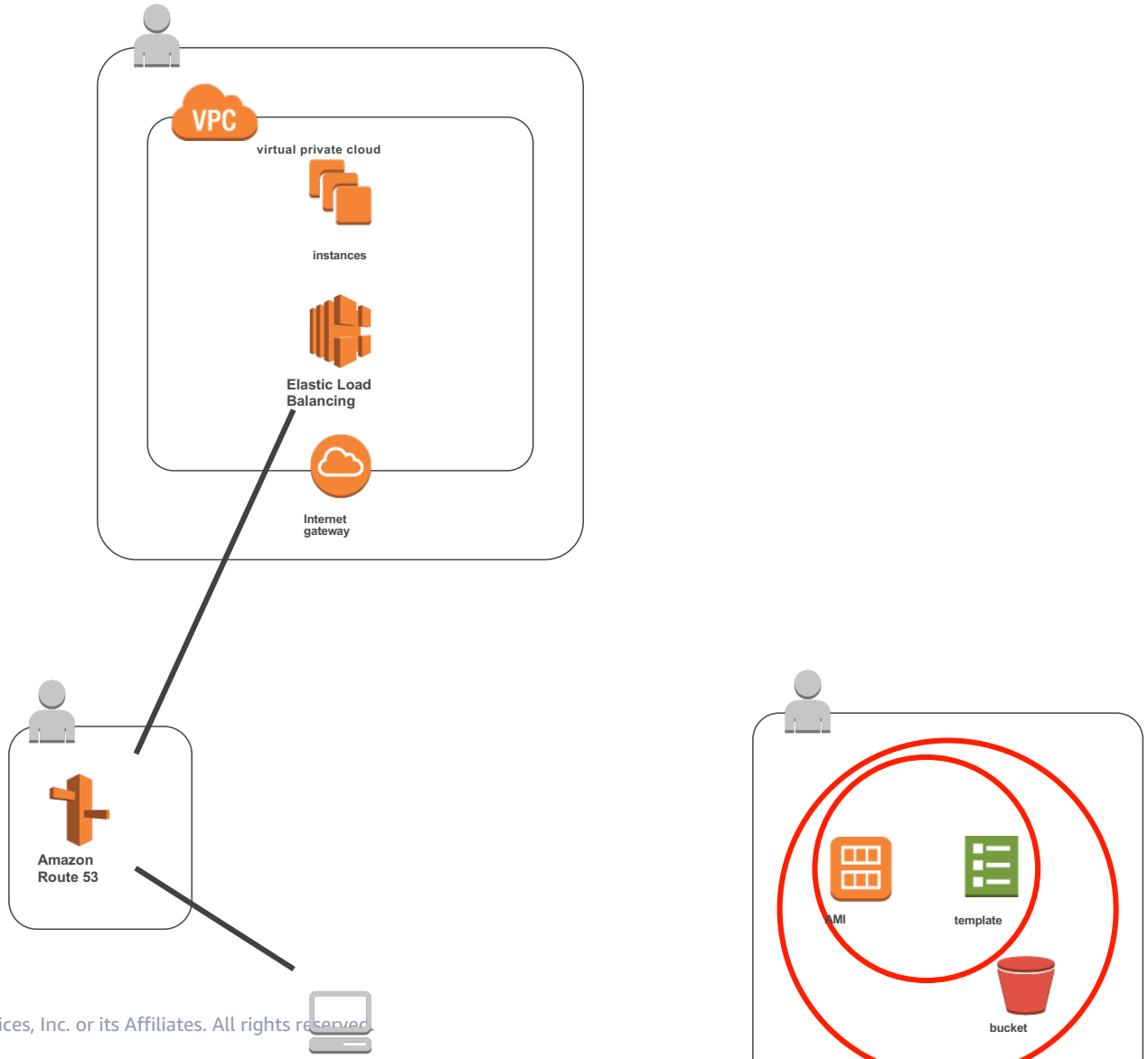
- AWS (ticket or TAM)
- Auditors, Regulators
- Forensic Investigators (QIRAs, CESG-recommended firms – see eg https://www.governmentcomputing.com/central-government/news/newscesg-announces-cyber-incident-response-providers)?
- Police?
- Ensure all contact details for the above are in your Runbook, and current…

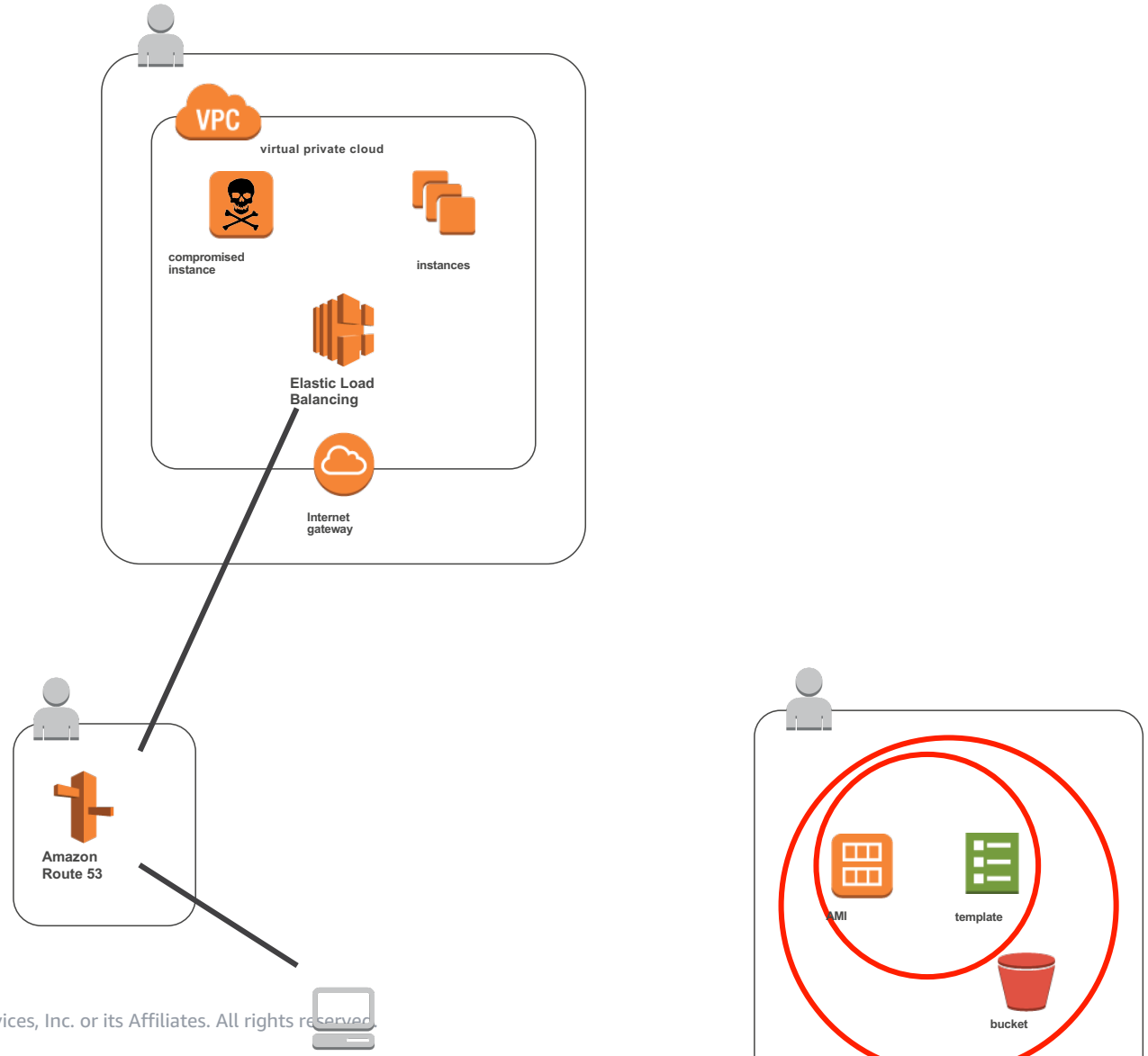
aws

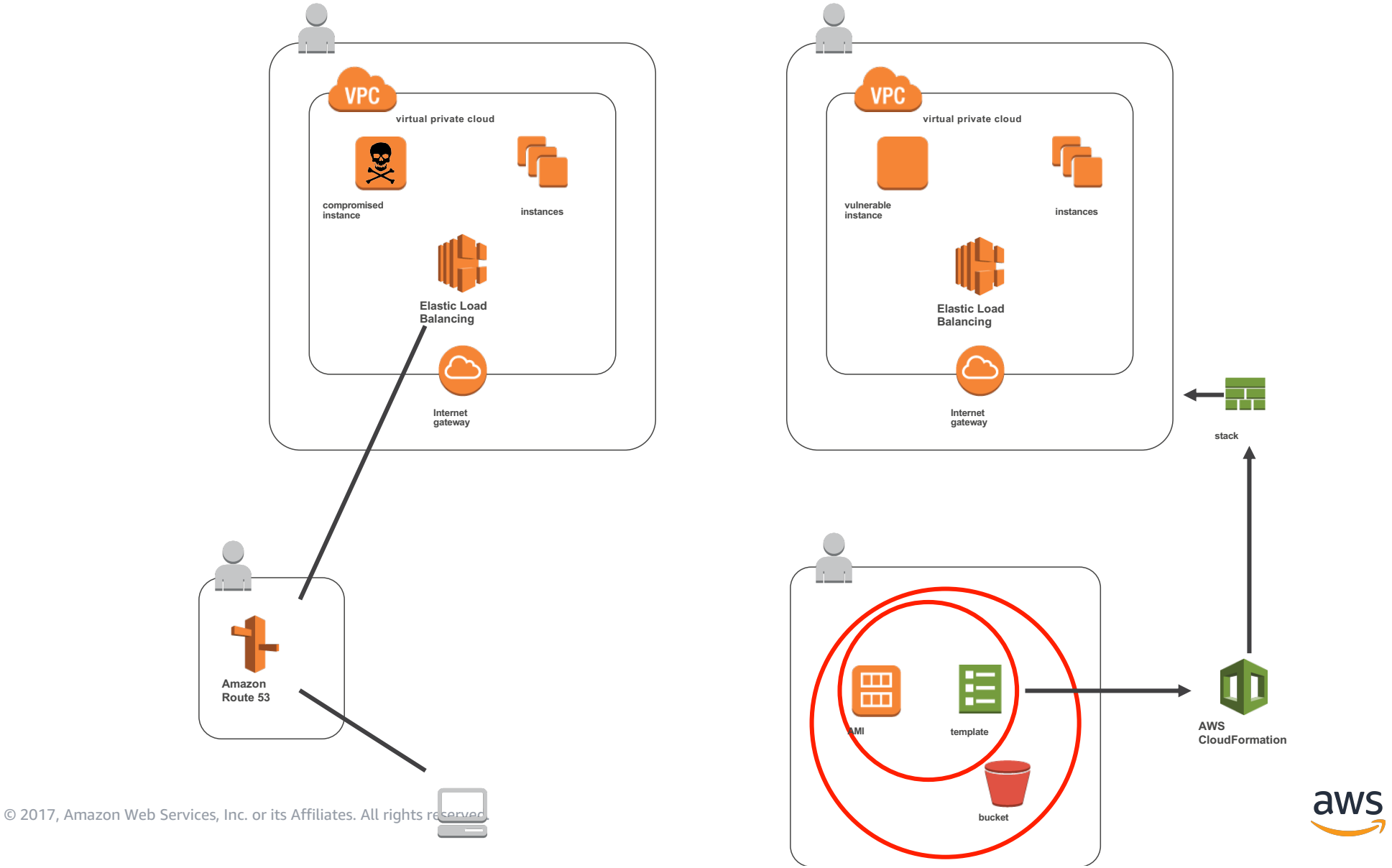
“Decisions, Decisions…”

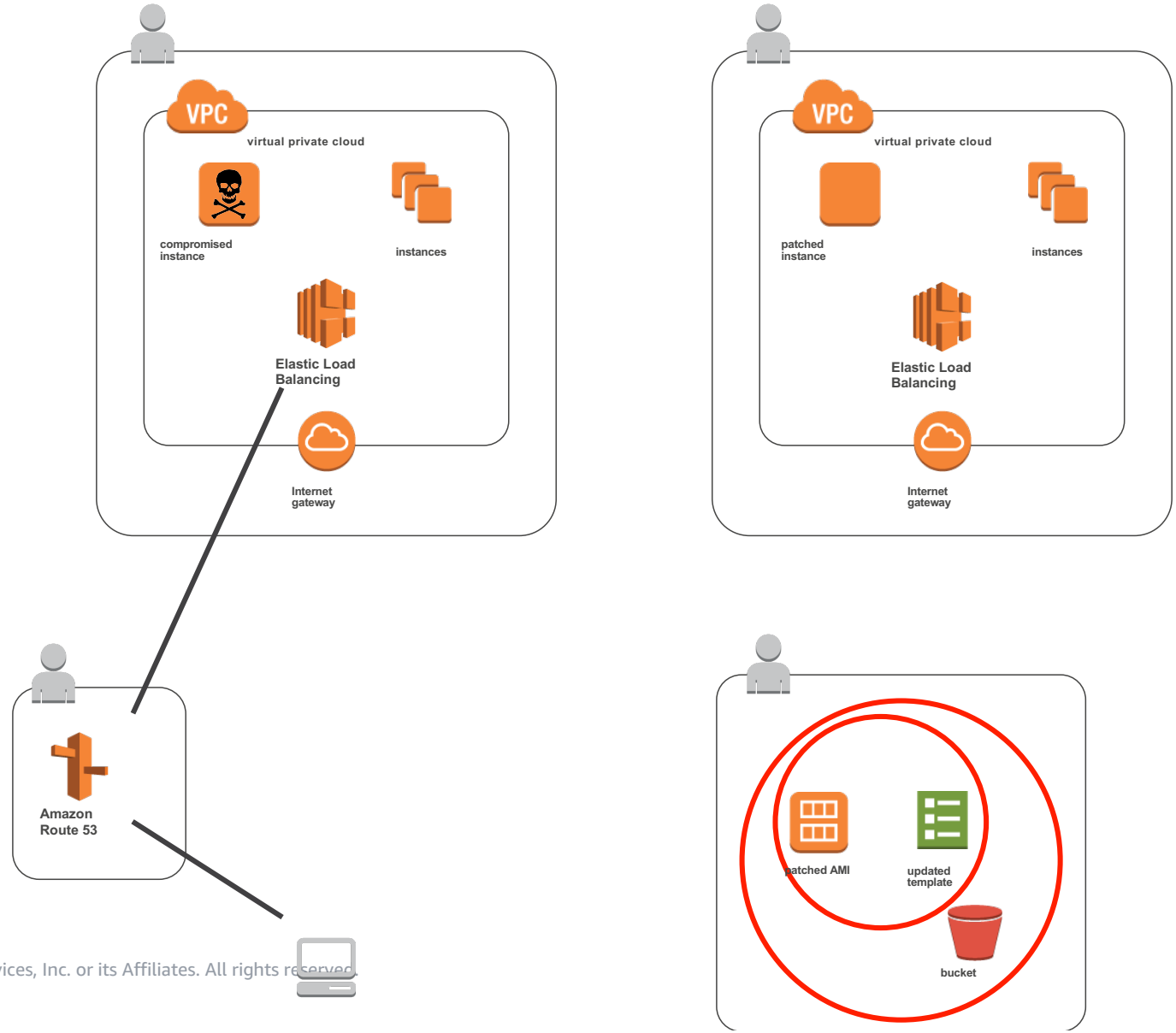
- Replicate
- Repair
- Redirect
- Ringfence

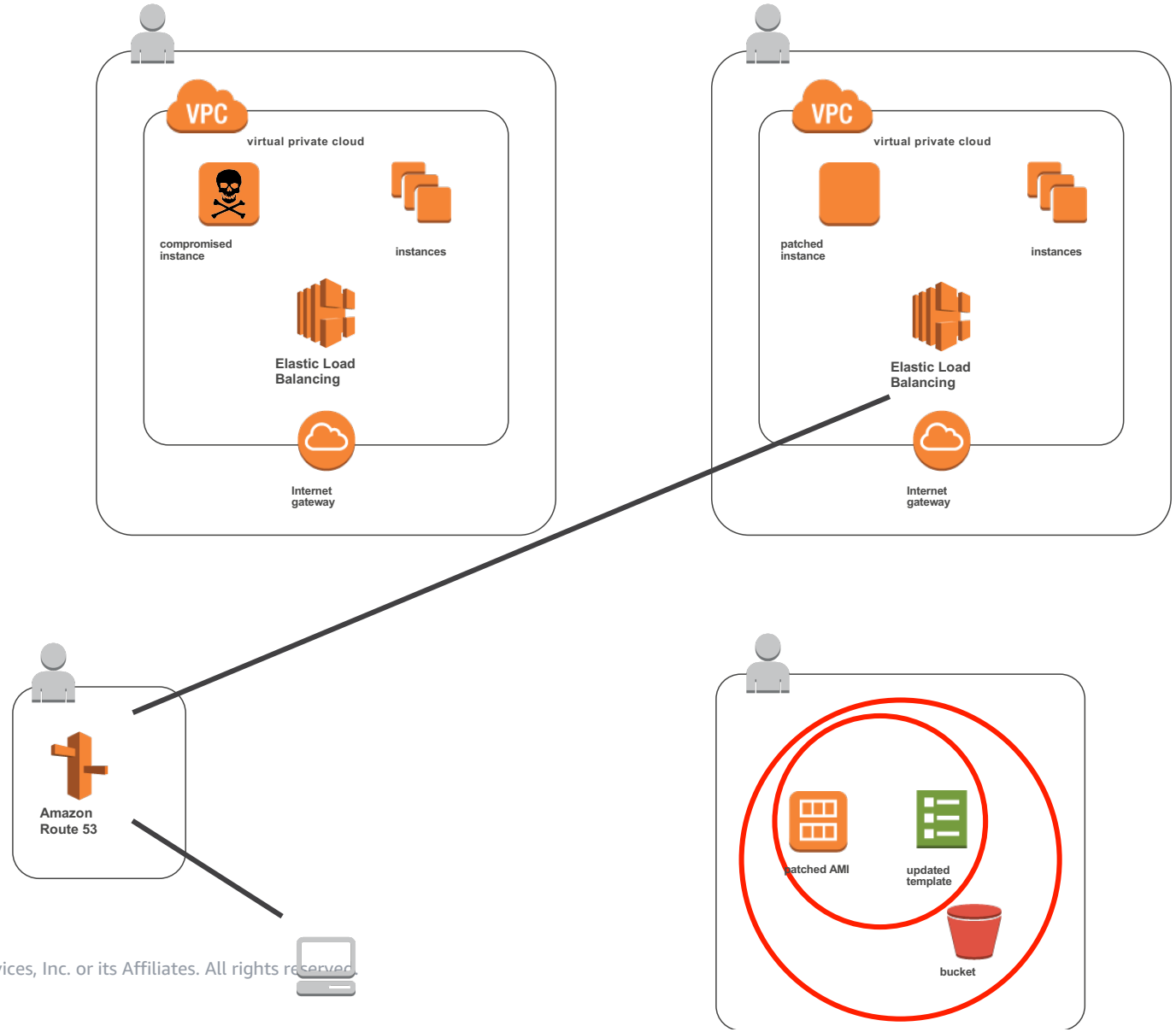
aws

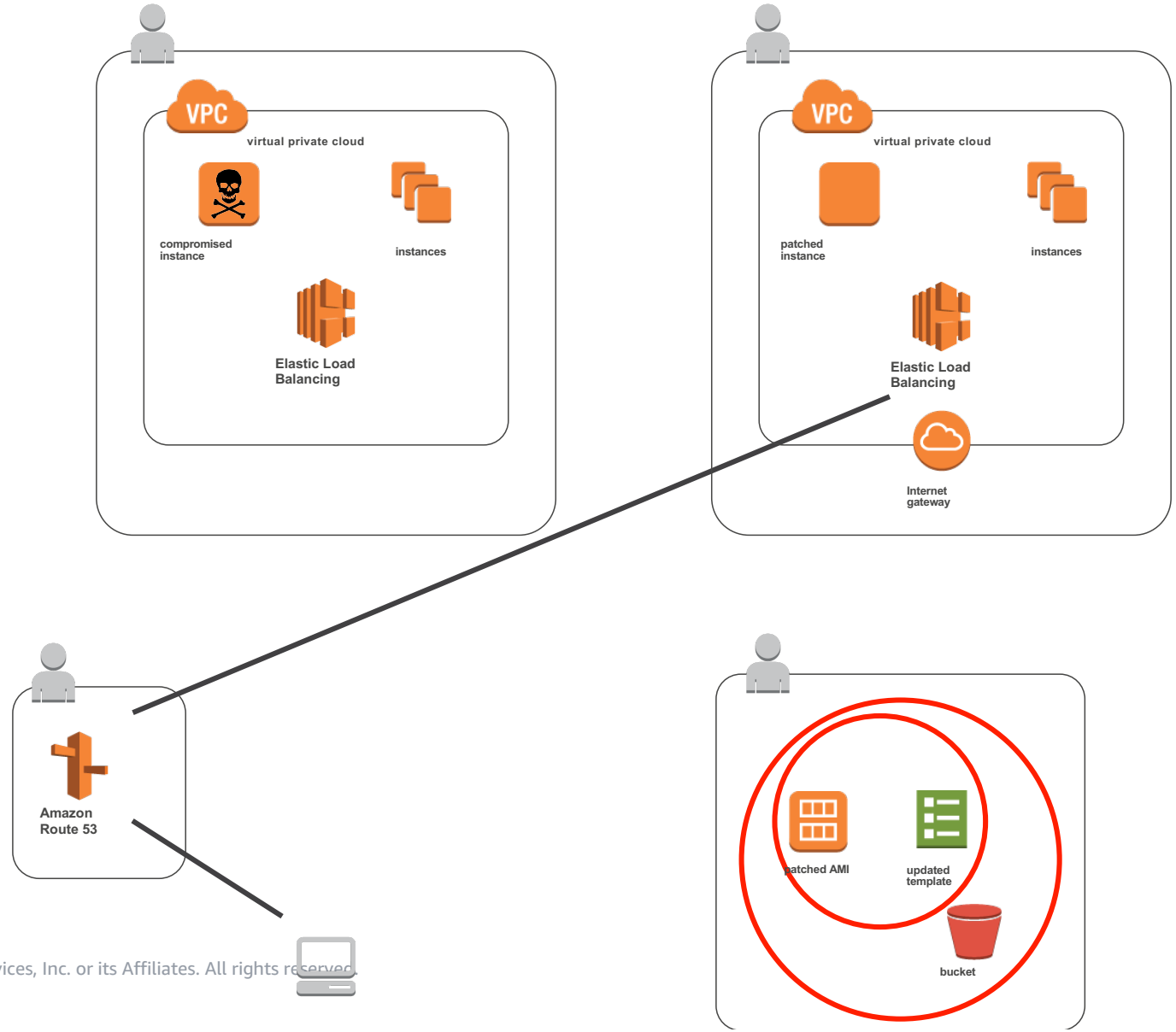












Immutability

- Drop an AWS Service Control Policy on compromised accounts
- Deny:
    - Writes to S3 buckets
    - Ditto EBS volumes
    - Ditto other state-holding assets
    - Deletes of S3 buckets, EBS volumes, other state-holding assets
- But, need to copy Service configs before change…
    - (How?)

aws

Preserving Evidence

Preserving Evidence

- Primary and Secondary Evidence
- Chain of Custody and Separation of Duty
- RAM?
- Disk?
- Object?

aws

Primary and Secondary Evidence

- Primary Evidence:
  - Physical media seized in evidence
  - RAM contents and storage of systems
  - State in assets in a compromised AWS account

- Secondary Evidence:
  - Any copies of the above
  - Output of processing evidence (*needs to be reproducible*)
  - Information derived from the processing output

aws

RAM?

- LiME Forensics Kernel Module to dump memory! (*common advice*)
- …and https://github.com/ThreatResponse/margaritashotgun , at scale

- Pros:
  - You get RAM contents *(+ LiME)*
- Cons:
  - Now you have LiME on primary evidence
  - NOISY! Generates log records
    - …and network traffic
    - …and alters package manifests
    - …and overwrites blocks in the free space pool

aws

Disk?

- Amazon Elastic Block Store (EBS)

- Snapshotting is incremental at block level
  - Even when volumes are encrypted
- But, *encryption also gives integrity assurance*
- *(…provided you maintain custody of the key)*
  - Modifying KMS key access may be useful and generates logs
  - Time breakpointing and sync is vital

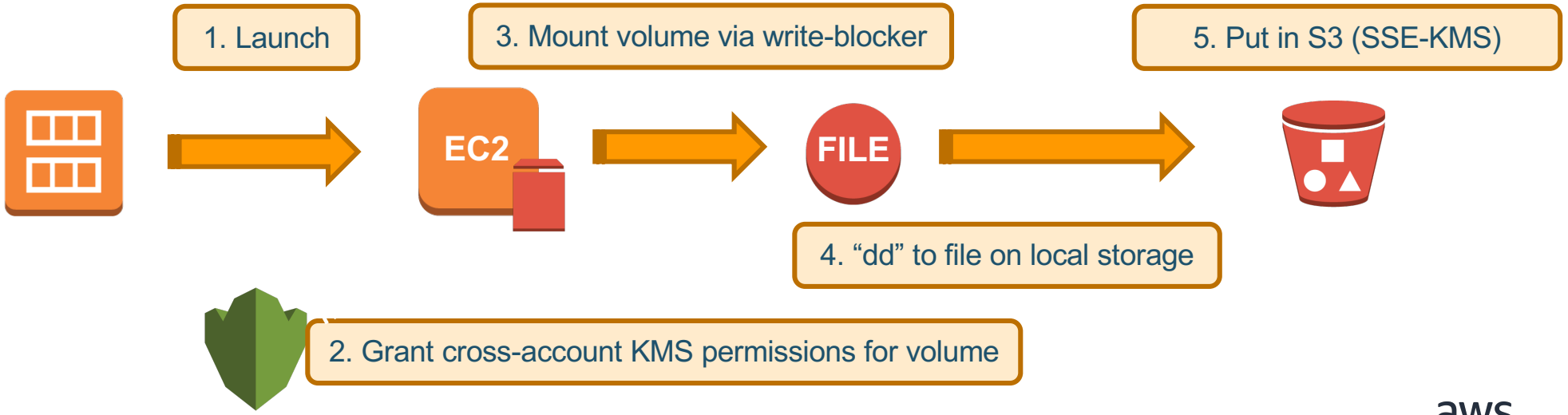
aws

Disk?

Alternatives – Consider a “dding AMI”

Build minimal, open-source, forensic AMI (for forensic EC2)

Pre-install a software write-blocker (eg https://github.com/msuhanov/Linux-write-blocker)



aws

Object?

Can re-use the forensic EC2

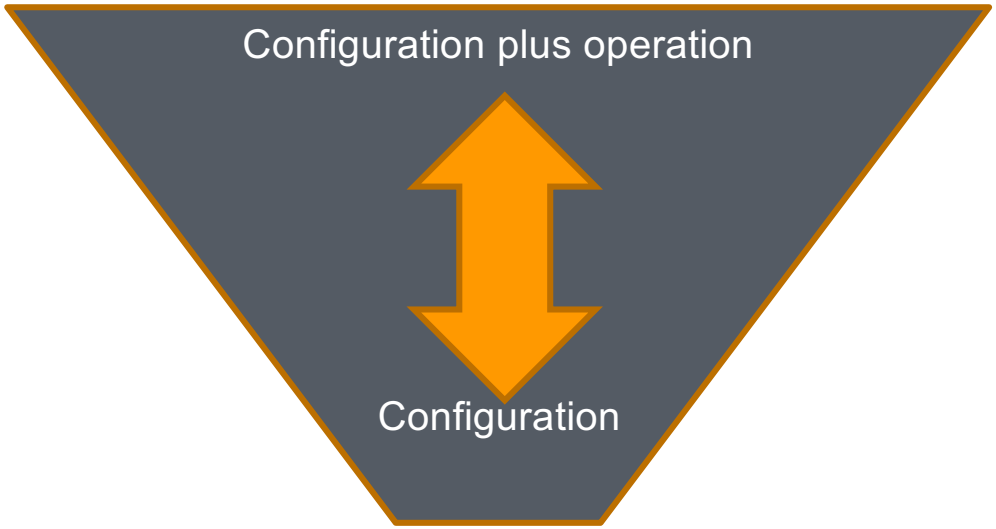
- Decrypt evidence object in the instance
- Re-encrypt and push to the evidence bucket

Preserving Evidence, by Service

Infrastructure services

Container services

Abstracted services



- Source:
- https://d1.awsstatic.com/whitepapers/Security/AWS_Security_Best_Practices.pdf

aws

VPCs and Services which Run In Them

- Create forensic subnets in all VPCs
  - All routes black-holed; all traffic stays within
- Create “isolate” security groups to seal off potentially compromised hosts
- Configure all AMIs to automatically mount ENIs when signalled by events

aws

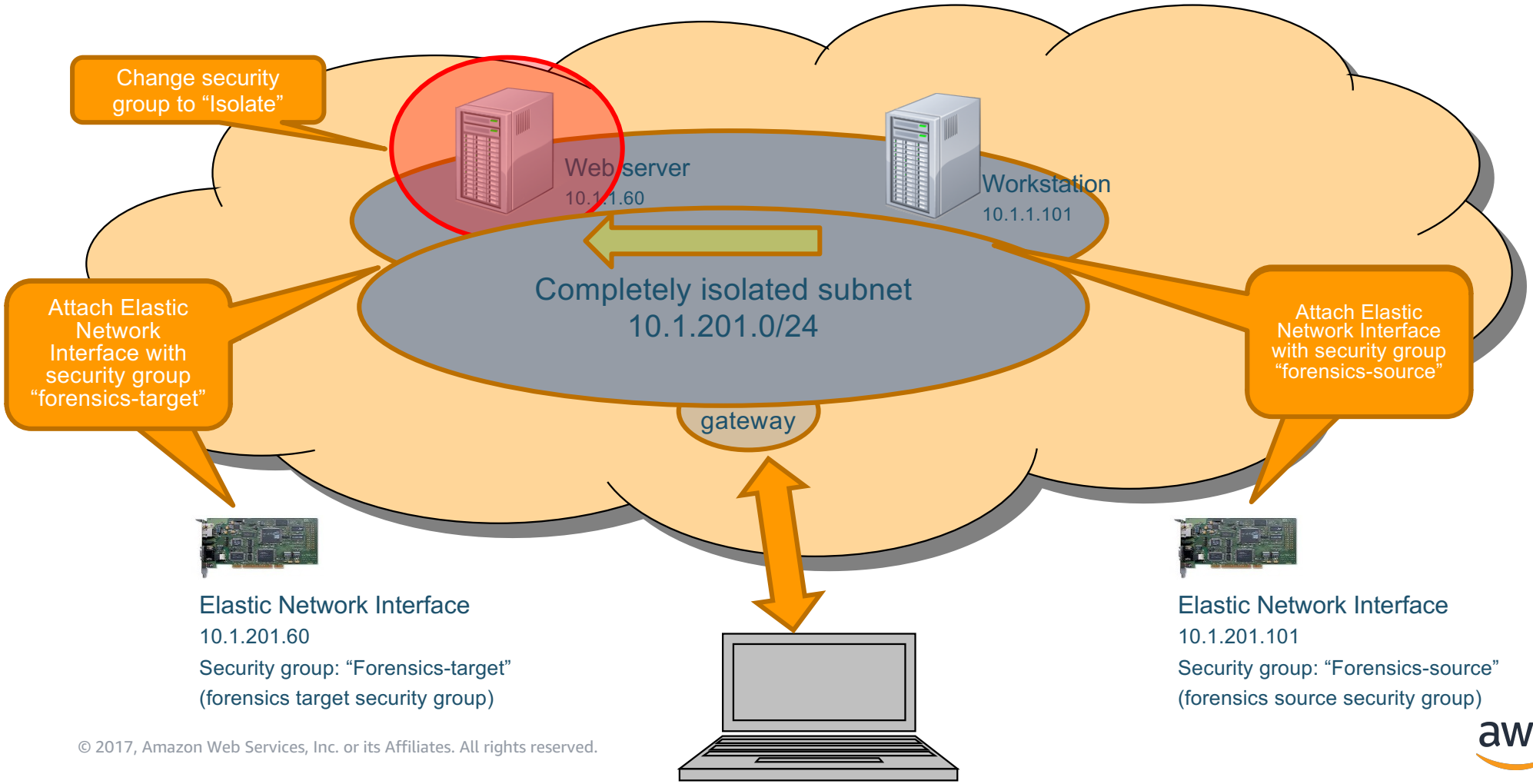
VPCs and Services which Run In Them

- Configure investigation hosts
  - All forensics tools installed and ready to go
  - Protect with deny-all SGs on forensics subnet
  - Set up write blockers, eg https://github.com/msuhanov/Linux-write-blocker

aws

Analysis

Analysis



Elastic Network Interface
10.1.201.60
Security group: “Forensics-target”
(forensics target security group)

Elastic Network Interface
10.1.201.101
Security group: “Forensics-source”
(forensics source security group)

aws

Analysis

EIR
- Mandiant
- EnCase
- FTK
- GRR

Network
- Wireshark
- Moloch

Memory Capture
- Fastdump
- FTK Imager
- LiME

aws

Training your People

Training your People

Security Incident Response Simulations (SIRS) are internal events that provide a structured opportunity to practice your incident response plan during a realistic scenario.

SIRS events are fundamentally about being prepared and iteratively improving your response capabilities.

aws

Working Backwards from Customers

Customers voice the following reasons why they want to perform SIRS:

- **Validate readiness**
- **Develop confidence – Learn from and train staff**
- Generate artifacts for accreditation
- Be agile – Incremental improvement with laser focus
- Become faster and improve tools
- Refine escalation and communication
- Develop comfort with the rare and the creative

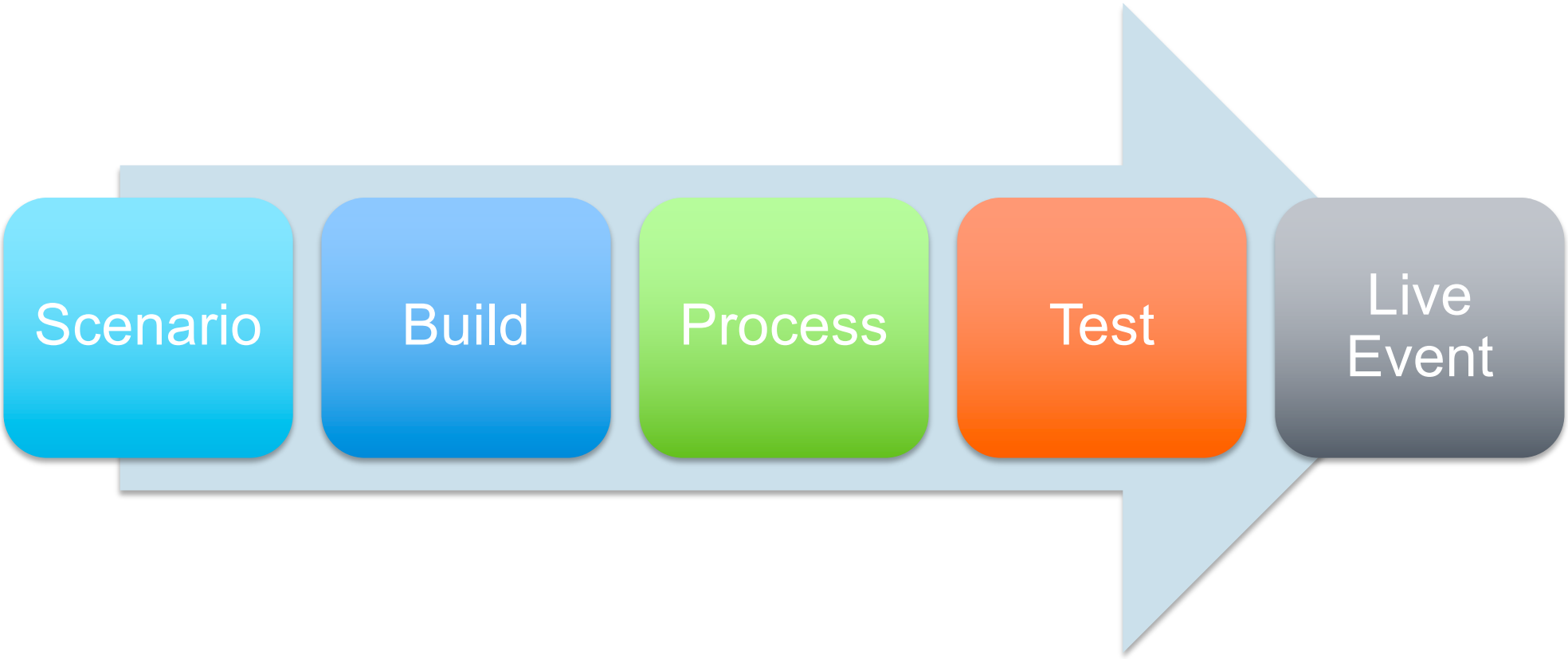
aws

Preparing for a simulation

1. Find an issue of importance.
2. Find ~~skilled security geeks~~, a team that is working in the environment in question.
3. Build a realistic model system.
4. Build and test the scenario elements.
5. Invite other ~~security geeks~~ teams and real people, include security.
6. Run the simulation. Fail.
7. Get better and repeat.

aws

Key Simulation Elements



aws

Conclusion(s)

Conclusion(s)

- "Be Prepared"
- Many currently-recommended practices are free; the rest are cheap
- Configure environments thoughtfully to minimise scope of compromise
- Use rich data emitted by “cloud wrapper” for greater visibility and control, and set up analytics and alarms
- “Snapshot” your vital recovery-candidate data frequently and version it
- Use programmable infrastructure to automate detection (and potentially mitigation)
- Always get the right people involved, at the outset
  - …and that includes AWS

aws

Thank you!