

# Detecting Credential Compromise in **AWS**

| Will Bengtson  
| @\_\_muscles


**NETFLIX**

# -> whoami

Senior Security Software Engineer on Netflix's Security Tools and Operations Team

Netflix is a microservice ecosystem running 100% in AWS. We try and like to build cool things:

- [Least Privilege: Security Gain Without Developer Pain](#)
- [Application DoS in Microservice Architectures](#)
- [Best Practice for Managing Security Operations on AWS](#)

 @\_\_muscles

 <https://github.com/willbengtson>

# This is not a machine learning talk

Why use machine learning when things can be much more simple



# What is the scope of this talk?

Detection of compromised AWS instance credentials (STS credentials) outside of your environment

STS - The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users)<sup>1</sup>

<sup>1</sup> <https://docs.aws.amazon.com/STS/latest/APIReference/Welcome.html>

**WHAT'S THE PROBLEM?**

**WHO IS DOING THIS WELL?**

**WHY IS THIS SO HARD?**

**WHAT TOOLS ARE THERE?**



# CloudTrail

CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services.<sup>1</sup>

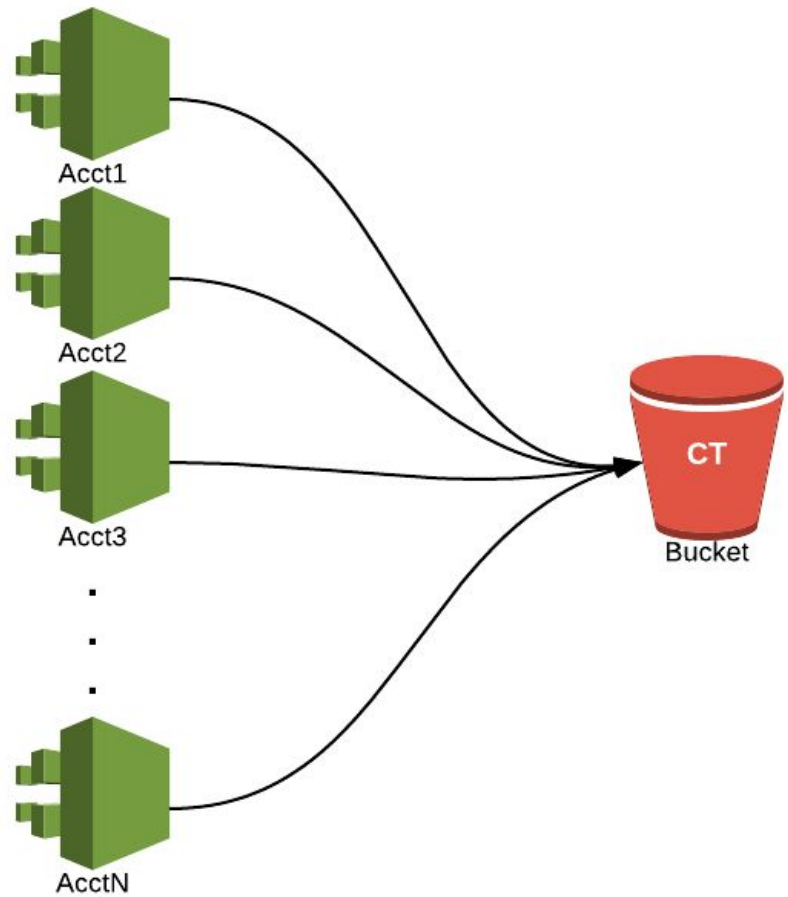
- Accessible via console
- Deliverable via S3 or CloudWatch Logs
  - AccountID\_CloudTrail\_RegionName\_YYYYMMDDTHHmmZ?UniqueString.FileNameFormat
- Up to 15 or 20 minutes delayed

<sup>1</sup> <https://aws.amazon.com/cloudtrail/>

```

{
  "Records": [
    {
      "eventVersion": "1.0",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
      },
      "eventTime": "2014-03-06T21:22:54Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "StartInstances",
      "awsRegion": "us-east-2",
      "sourceIPAddress": "205.251.233.176",
      "userAgent": "ec2-api-tools 1.6.12.2",
      "requestParameters": {
        "instancesSet": {
          "items": [
            {
              "instanceId": "i-ebeaf9e2"
            }
          ]
        }
      },
      "responseElements": {
      }
    }
  ]
}

```



# First Iteration

# First Iteration

## Requirements

- Know all IPs in environment (multiple accounts) for the last hour
- Compare each IP found in CloudTrail to list of IPs
  - If we had the IP at the time of log keep going
  - If we DID NOT have the IP at the time of the log, ALERT

# AWS Limitations

- Pagination
- Rate Limiting

**New Approach**

# New Approach

- Use an understanding of how AWS works to our advantage
- Make a strong but reasonable assumption
- Profit

From 0 to full coverage in around 6 hours



**HOW DOES AWS WORK?**



Amazon  
EC2



IAM



Meta-data  
Service



AWS SDK

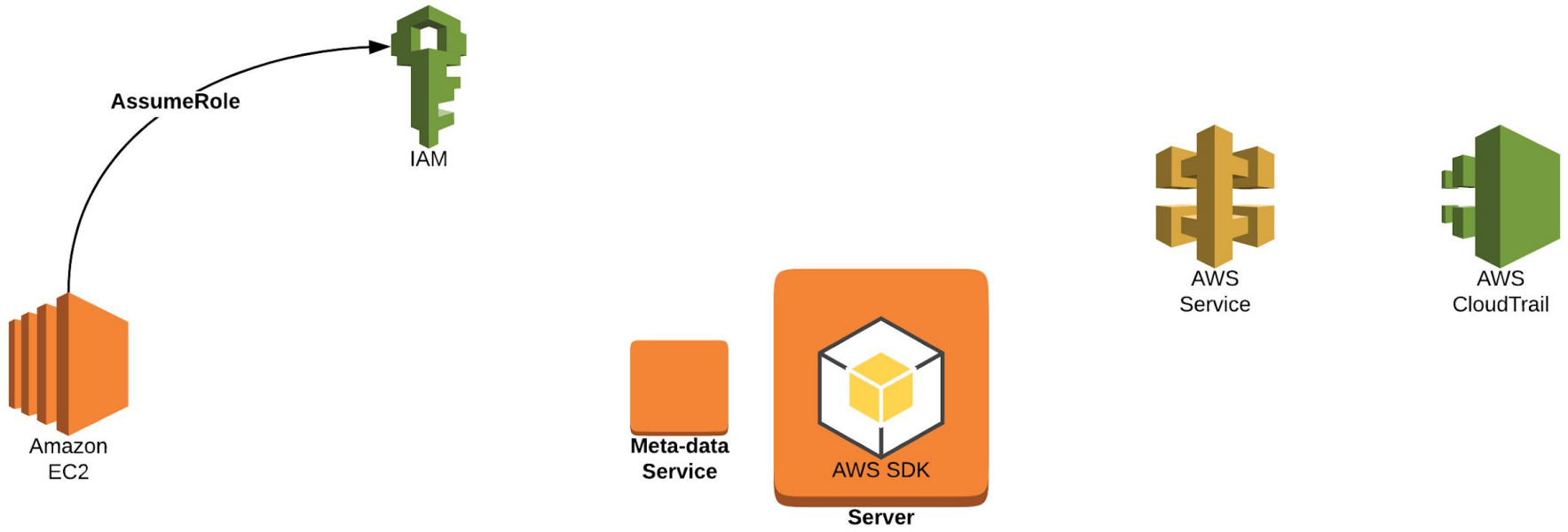
Server

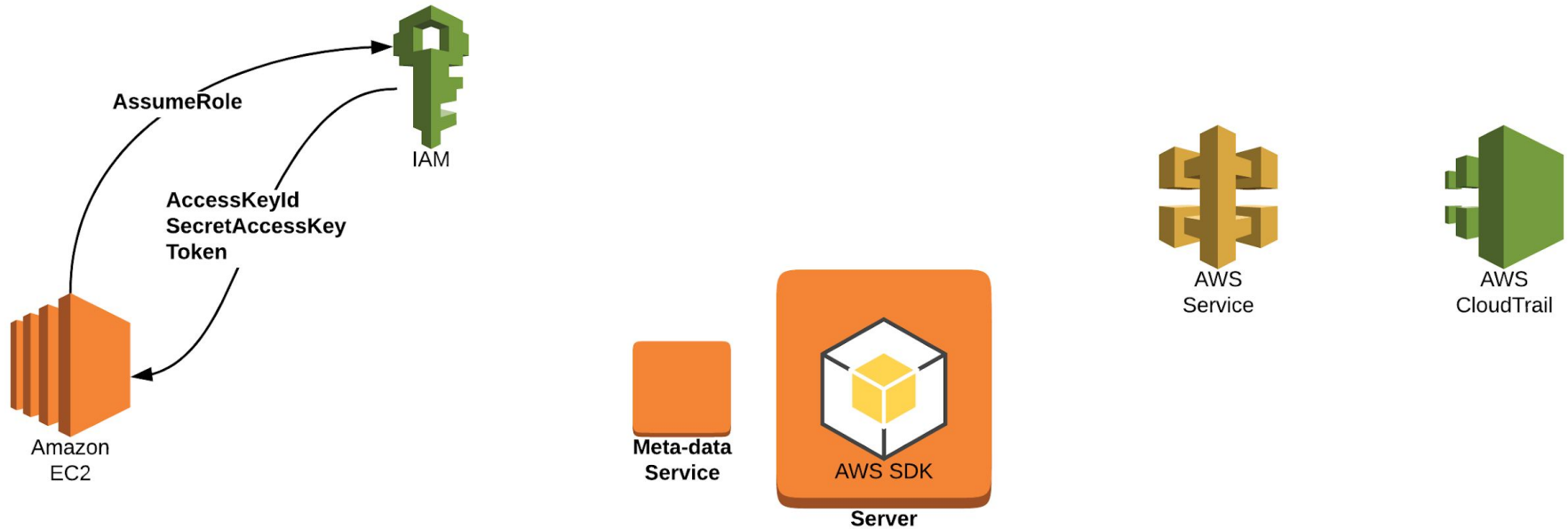


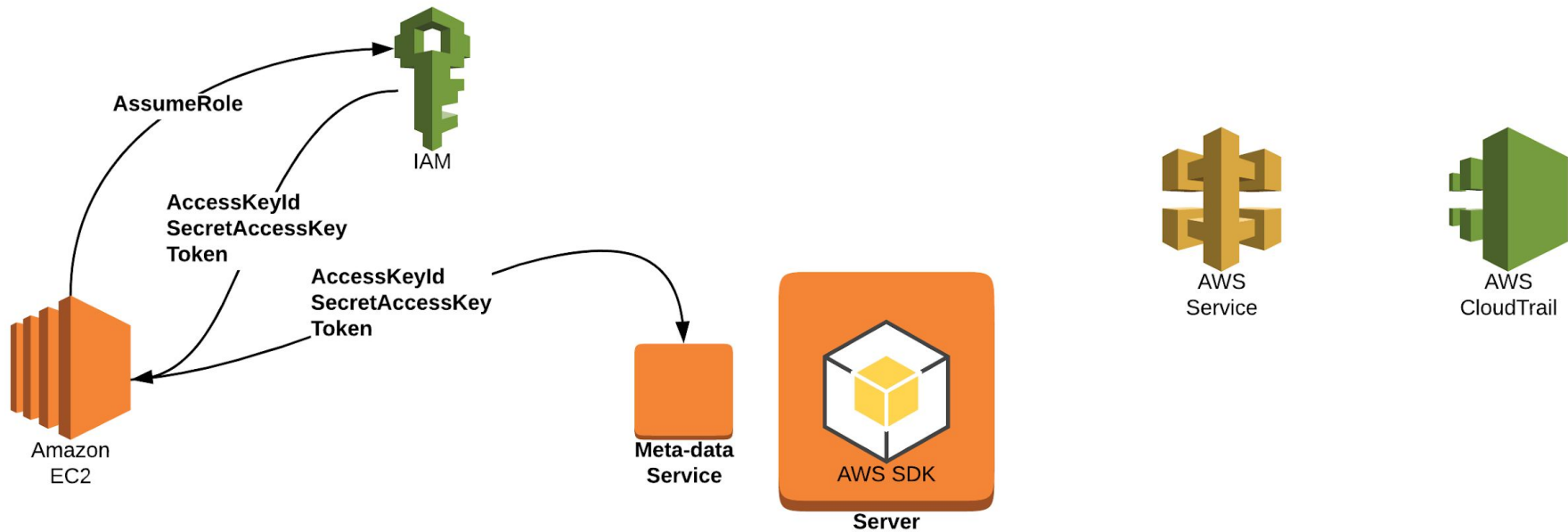
AWS  
Service

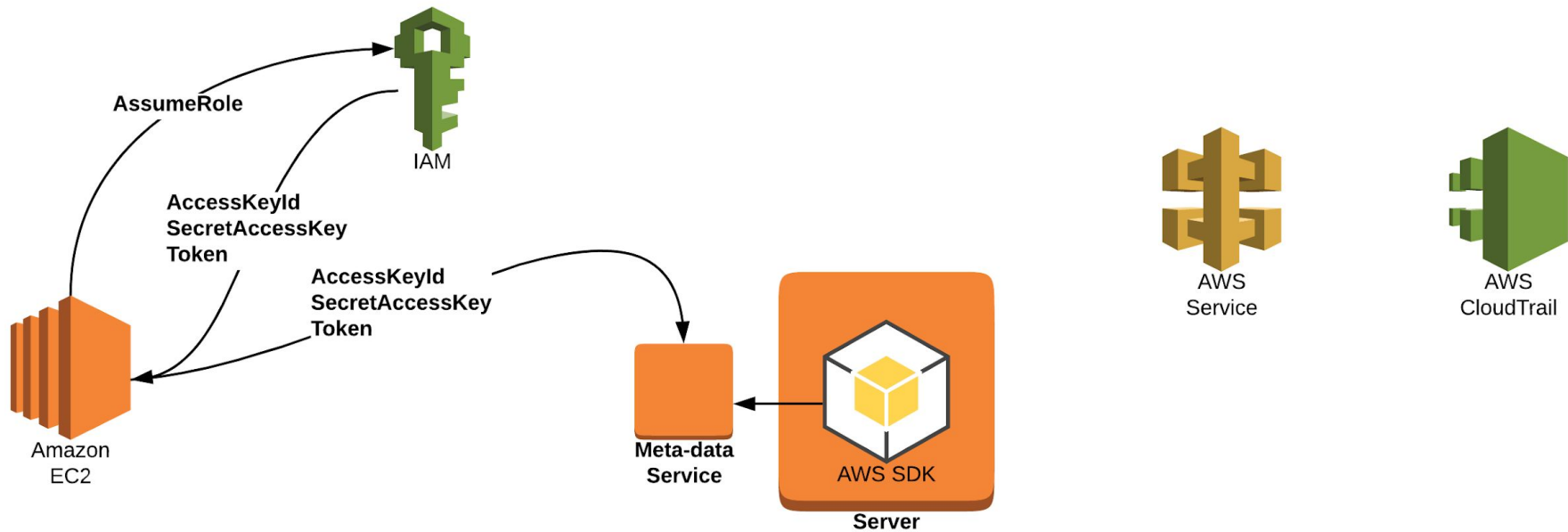


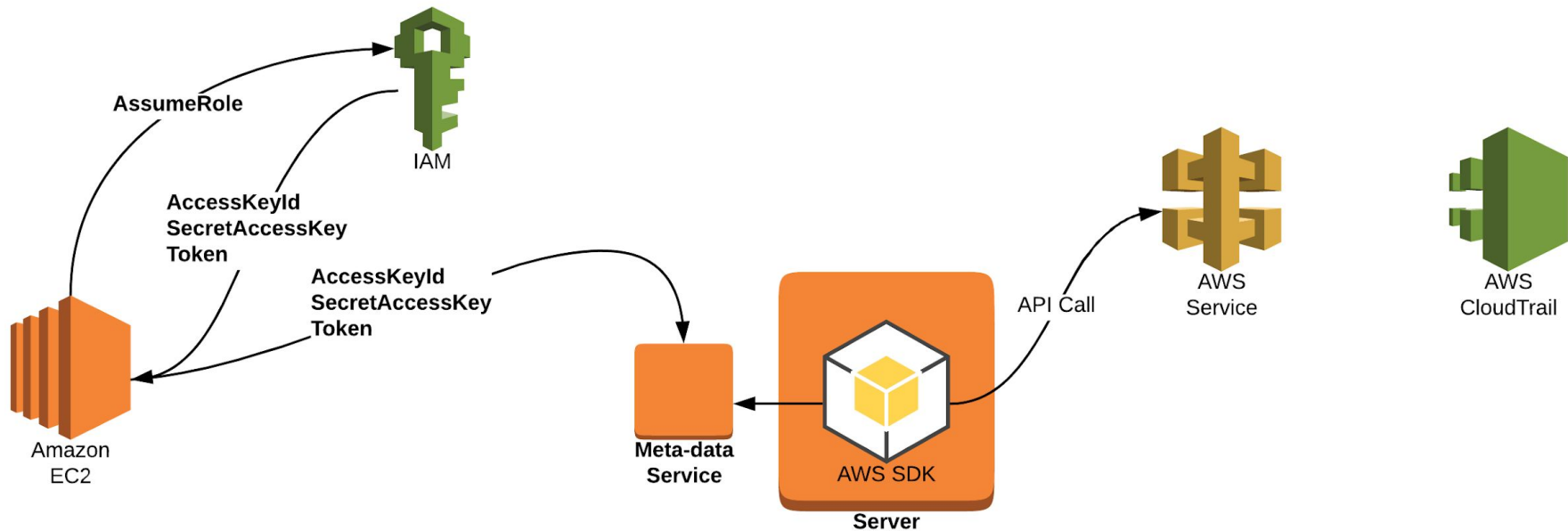
AWS  
CloudTrail

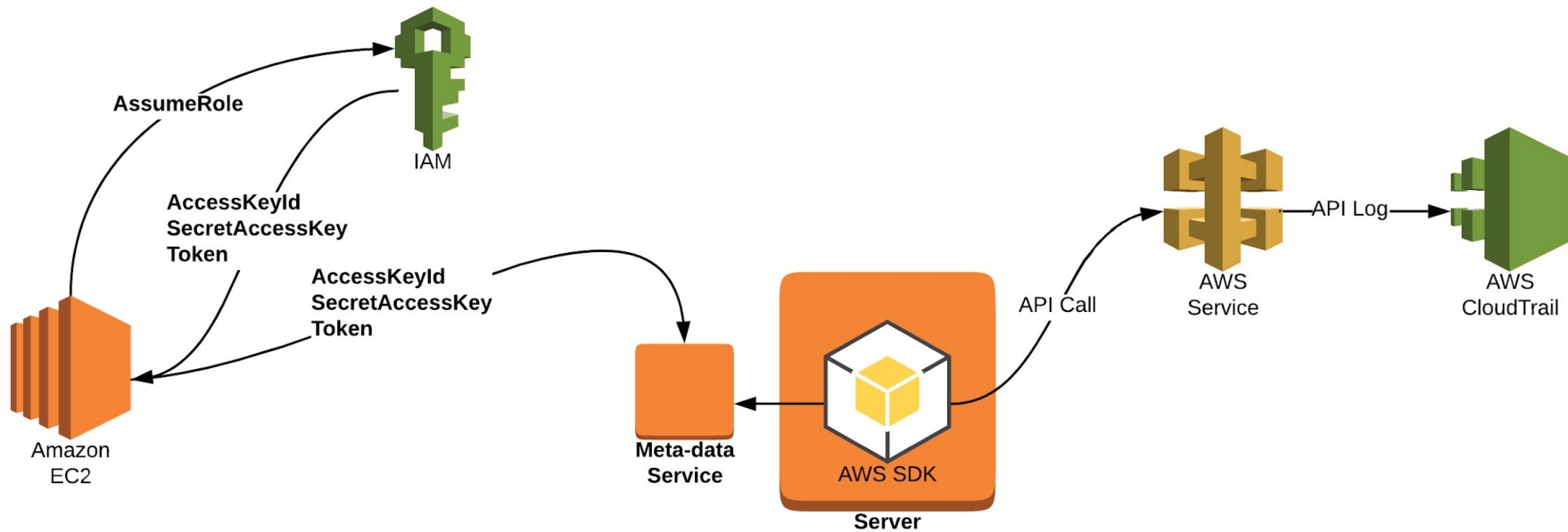






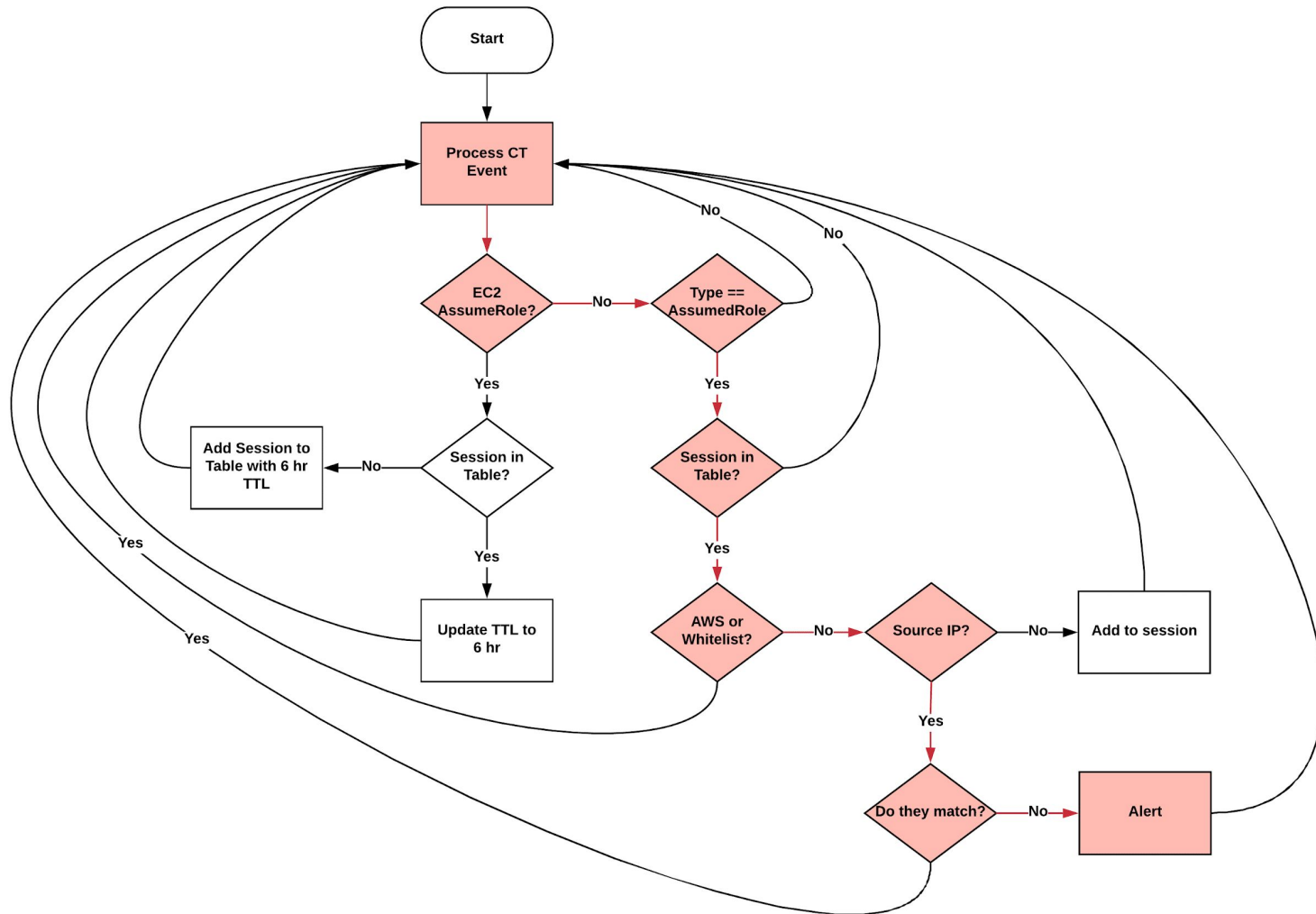


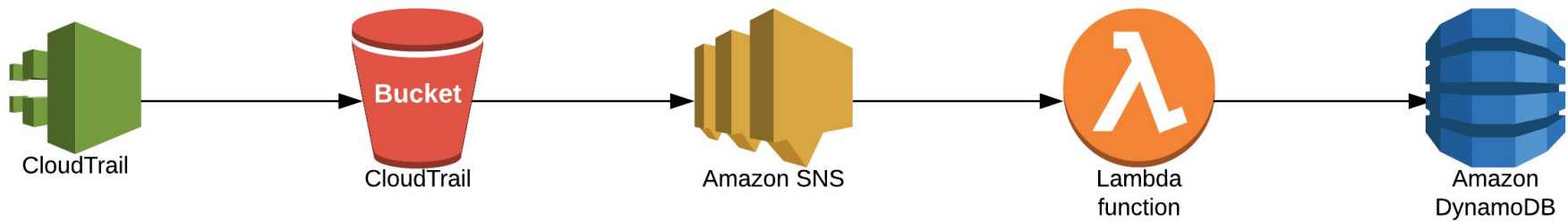






**STRONG ASSUMPTION**





identifier	source_ip	arn	ttl_value

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "ec2.amazonaws.com"
  },
  "eventTime": "2018-04-03T23:52:43Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "AssumeRole",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "ec2.amazonaws.com",
  "userAgent": "ec2.amazonaws.com",
  "requestParameters": {
    "roleSessionName": "i-00000000000002131",
    "roleArn": "arn:aws:iam::123456789012:role/myCoolRole"
  },
  "responseElements": {
    "credentials": {
      "sessionToken": "FQoDYblahblahblah",
      "accessKeyId": "ASIAXXXXXXXXXXXXXXXXXX",
      "expiration": "Apr 4, 2018 6:19:07 AM"
    }
  },
  "requestID": "f5884380-640e-4655-86b2-3b0701268fac",
  "eventID": "0b81ef6b-ea80-431e-8037-3ca7f2fbb338",
  "resources": [
    {
      "ARN": "arn:aws:iam::123456789012:role/myCoolRole",
      "accountId": "123456789012",
      "type": "AWS::IAM::Role"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012",
  "sharedEventID": "adab4f62-c082-4f08-84cd-073ccdfa7bee"
}
```

identifier	source_ip	arn	ttl_value
i-000000000000002131		arn:aws:iam::123456789012:assumed-role/myCoolRole	1531904179.955654

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAXXXXXXXXXXXXXXXXXX:i-00000000000002131",
    "arn": "arn:aws:sts::123456789012:assumed-role/myCoolRole/i-00000000000002131",
    "accountId": "123456789012",
    "accessKeyId": "ASIAXXXXXXXXXXXXXXXXXX",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-04-03T23:54:03Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAXXXXXXXXXXXXXXXXXX",
        "arn": "arn:aws:iam::123456789012:role/myCoolRole",
        "accountId": "123456789012",
        "userName": "myCoolRole"
      }
    }
  },
  "eventTime": "2018-04-03T23:54:06Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "DescribeInstances",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "52.95.255.121",
  "userAgent": "Boto/2.48.0 Python/2.7.12 Linux/4.4.0-98-generic",
  "requestParameters": {
  },
  "responseElements": null,
  "requestID": "fff57ab1-105d-4c18-9216-a702a603f388",
  "eventID": "7a779948-d108-4ea0-9519-ca6494fec9d4",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

identifier	source_ip	arn	ttl_value
i-000000000000002131		arn:aws:iam::123456789012:assumed-role/myCoolRole	1531904179.955654



identifier	source_ip	arn	ttl_value
i-000000000000002131	52.95.255.121	arn:aws:iam::123456789012:assumed-role/myCoolRole	1531904179.955654

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAXXXXXXXXXXXXXXXXXX:i-00000000000002131",
    "arn": "arn:aws:sts::123456789012:assumed-role/myCoolRole/i-00000000000002131",
    "accountId": "123456789012",
    "accessKeyId": "ASIAXXXXXXXXXXXXXXXX",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-04-03T23:54:03Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAXXXXXXXXXXXXXXXXXX",
        "arn": "arn:aws:iam::123456789012:role/myCoolRole",
        "accountId": "123456789012",
        "userName": "myCoolRole"
      }
    }
  },
  "eventTime": "2018-04-03T23:54:10Z",
  "eventSource": "ec2.amazonaws.com",
  "eventName": "DescribeVolumes",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "52.95.255.121",
  "userAgent": "Boto/2.48.0 Python/2.7.12 Linux/4.4.0-98-generic",
  "requestParameters": {
  },
  "responseElements": null,
  "requestID": "fff57ab1-105d-4c18-9216-a702a603f388",
  "eventID": "7a779948-d108-4ea0-9519-ca6494fec9d4",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

identifier	source_ip	arn	ttl_value
i-000000000000002131	52.95.255.121 =?= 52.95.255.121	arn:aws:iam::123456789012:assumed-role/myCoolRole	1531904179.955654



identifier	source_ip		ttl_value
i-000000000000002131	52.95.255.121 =?= 52.95.255.121	arn:aws:iam::9012:ass ume	1531904179.955654

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAXXXXXXXXXXXXXXXXXX:i-00000000000002131",
    "arn": "arn:aws:sts::123456789012:assumed-role/myCoolRole/i-00000000000002131",
    "accountId": "123456789012",
    "accessKeyId": "ASIAXXXXXXXXXXXXXXXX",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-04-03T22:04:57Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAXXXXXXXXXXXXXXXXXX",
        "arn": "arn:aws:iam::123456789012:role/myCoolRole",
        "accountId": "123456789012",
        "userName": "myCoolRole"
      }
    }
  },
  "eventTime": "2018-04-03T23:55:41Z",
  "eventSource": "sts.amazonaws.com",
  "eventName": "GetCallerIdentity",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "67.178.52.232",
  "userAgent": "aws-cli/1.14.45 Python/2.7.12 Linux/4.4.0-127-generic botocore/1.8.49",
  "requestParameters": {
  },
  "responseElements": null,
  "requestID": "0be425d4-ee38-4923-bf0f-32ddcc5f1f66",
  "eventID": "e18d0e02-44b5-4457-a98e-18a4eda1d91f",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

identifier	source_ip	arn	ttl_value
i-000000000000002131	52.95.255.121 =?= 67.178.52.232	arn:aws:iam::123456789012:assumed-role/myCoolRole	1531904179.955654



identifier	source	arn	ttl_value
i-000000000000002131	52.95.257	456789012:ass Role	1531904179.955654

# Edge Cases

There are a few edge cases to this approach that you may want/need to account for in order to prevent false positives. The edge cases are as follows:

- AWS will make calls on your behalf using your credentials if certain API calls are made
  - `sourceIPAddress: <service>.amazonaws.com`
- You have an AWS VPC Endpoint(s) for certain AWS Services
  - `sourceIPAddress: 192.168.0.22`
- You attach a new ENI or associate a new address to your instance
  - `sourceIPAddress: something new if external subnet`



# Avoiding Detection

## Server Side Request Forgery (SSRF)

- Use the same method that you pulled credentials to make the API calls

```
https://ec2.amazonaws.com/?Action=AssociateAddress&InstanceId=i-1234567890abcdef0&PublicIp=192.0.2.1&AUTHPARAMS
```

## Popped Box

- Attacker can execute commands on the system directly

# Final Thoughts

- Understand how AWS works and CloudTrail to make your life easier
- Understand what is logged in CloudTrail
  - Trailblazer is now OSS
    - AWS API Enumeration for Cloudtrail Intelligence / Attack Platform
    - <https://github.com/willbengtson/trailblazer-aws>
- AWS Credential Compromise Detection OSS
  - One way to detect credential compromise - Reference Architecture/Code
  - <https://github.com/Netflix-Skunkworks/aws-credential-compromise-detection>

**Thank you!**

@\_\_muscles

**NETFLIX**

