

AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

Security alchemy: How AWS uses math to prove security

Brigid Johnson (she/her)

GM of IAM Access Analyzer
AWS

Neha Rungta (she/her)

Director in AWS Identity
AWS

Let's start with a story

Once upon a time in AWS...

we were in search for something magical

something that could tell us with certainty

if security configurations were right.



Security alchemy: The rubric



Higher security



The apothecary



Automated reasoning & comprehensive analysis



Transfigurations



Brewing potions



Security charms in AWS

Higher security



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Growing security controls



Network

Control access to inbound and outbound traffic



Access

Control which identities can access actions and resources under specific conditions



Data encryption

Layer of security to encrypt data at rest and in transit

Security control examples



Service control policies (SCPs)



Identity-based policies



Resource-based policies



Endpoint policies



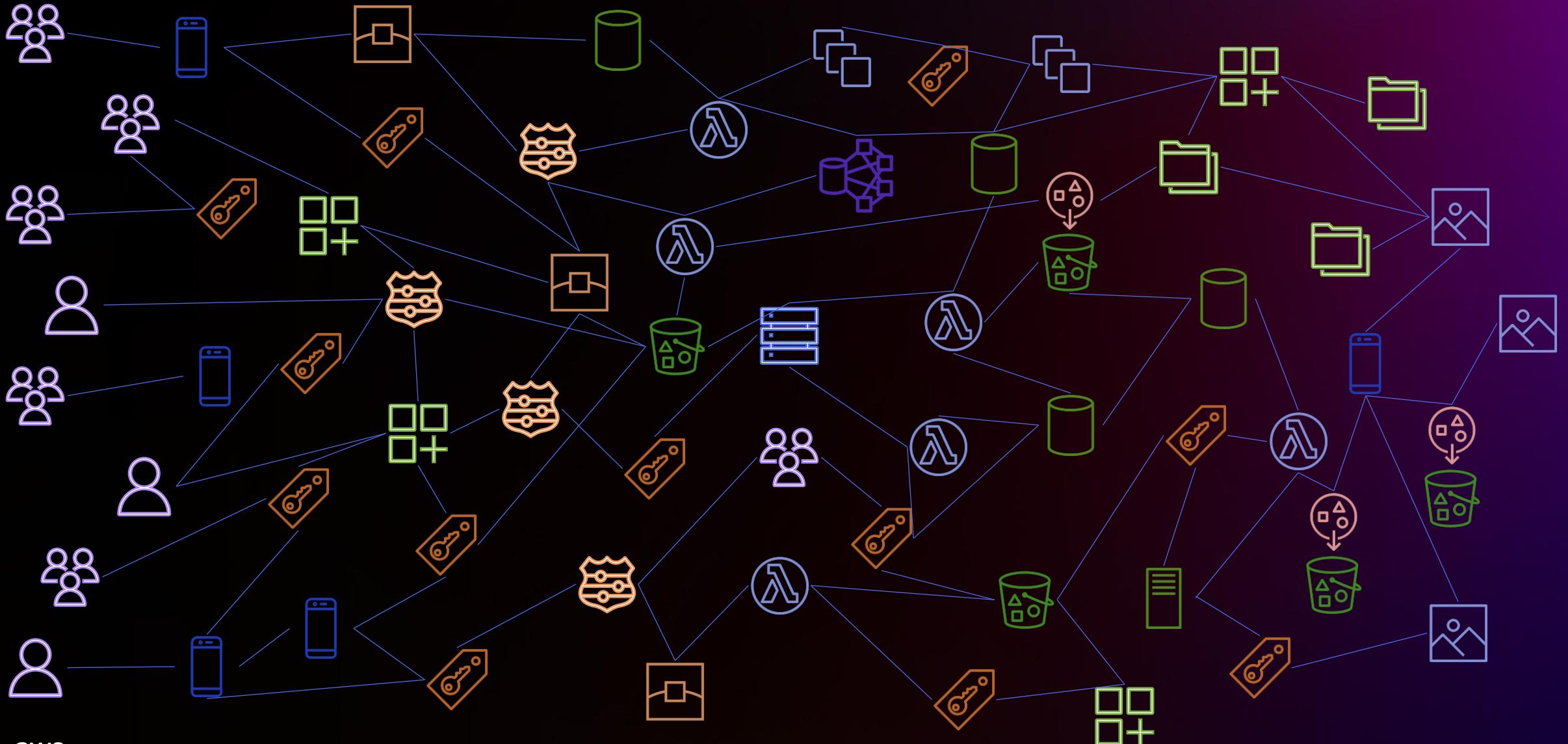
Block public access (BPA) with Amazon S3



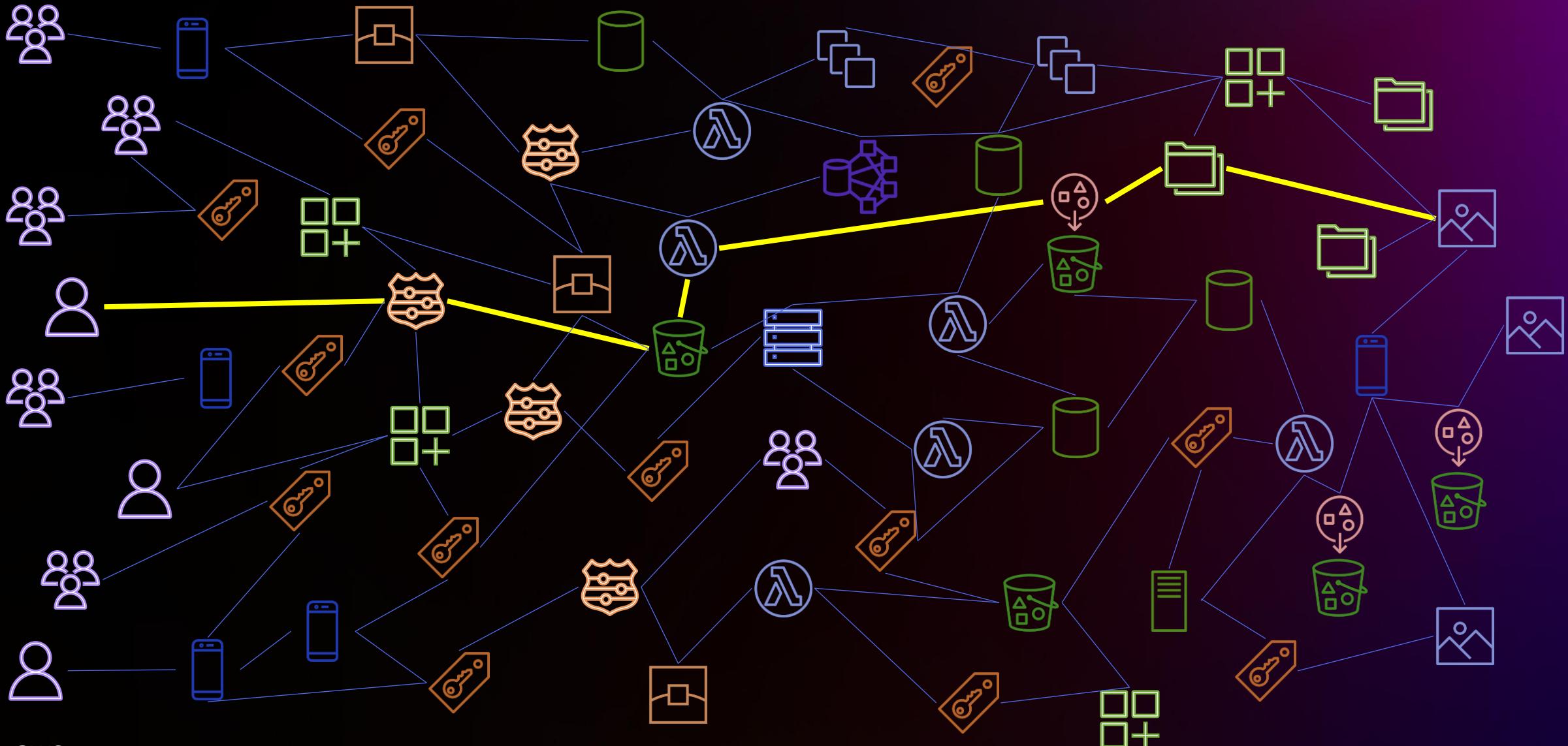
Service-specific controls such as AWS Resource Access Manager



Access paths of systems



Determine if there is a path



The apothecary



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Your shopping list

Universal statements



Universal statements
in comprehensive
analysis section

AWS controls



AWS security controls

Custom settings



Security configurations
and resource metadata



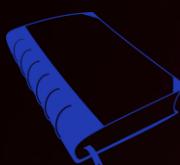
Transformer potion



Solver potion



Compute

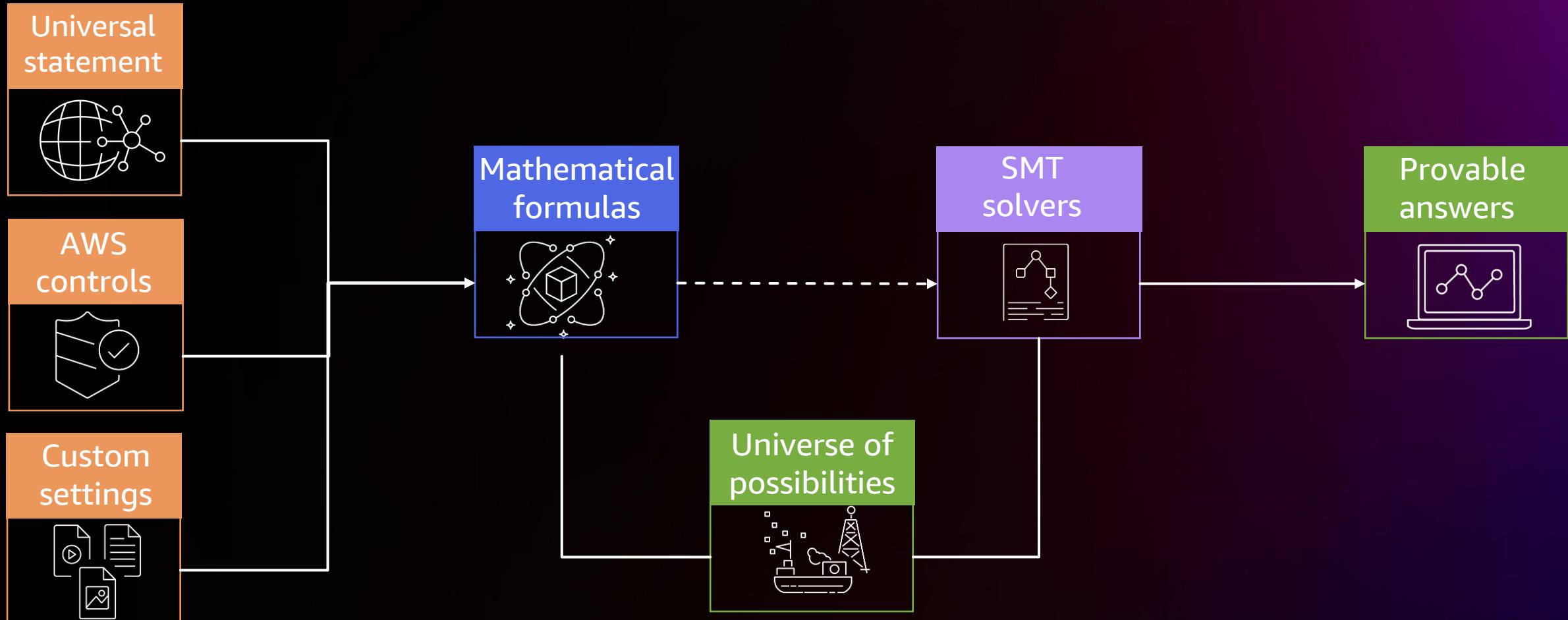


Automated reasoning
potions textbook

Automated reasoning and comprehensive analysis

Introduction to automated reasoning

Collect → Transform → Solve → Comprehensive



Comprehensive as a science



Muggle

● Question

● Configurations

● Comprehensive



Scientist

● Property

● All possible paths

● Proof of one path



Security
alchemist

Universal
statement



Universe of
possibilities



Provable
answers



Universal statements

A statement that always holds

A statement you can prove

If it doesn't hold, there is a reason



Universal statement examples

“All my data at rest is encrypted”

“There is no public access to my S3 buckets”

“No credentials are logged”

“There are no deadlocks in my code”

AWS controls (the knobs)

Fine-grained security controls that have a defined behavior

Deny statement
in policies



Access restricted for the action, resource-specified, and conditions met

Assume role
policy



Cross-account access allowed to set of accounts specified in **Principal** element—and no others

Block Public
Access (BPA)



Restricts public access for all new buckets

Customer security configurations

Established security controls based on customer intent

Resource policy
on an S3 Access
Point



Allow access to only **Principal** elements
within **organization-witches**—and no
others

IAM policies
with conditions



Deny access to **Principal** elements outside of
organization-witches

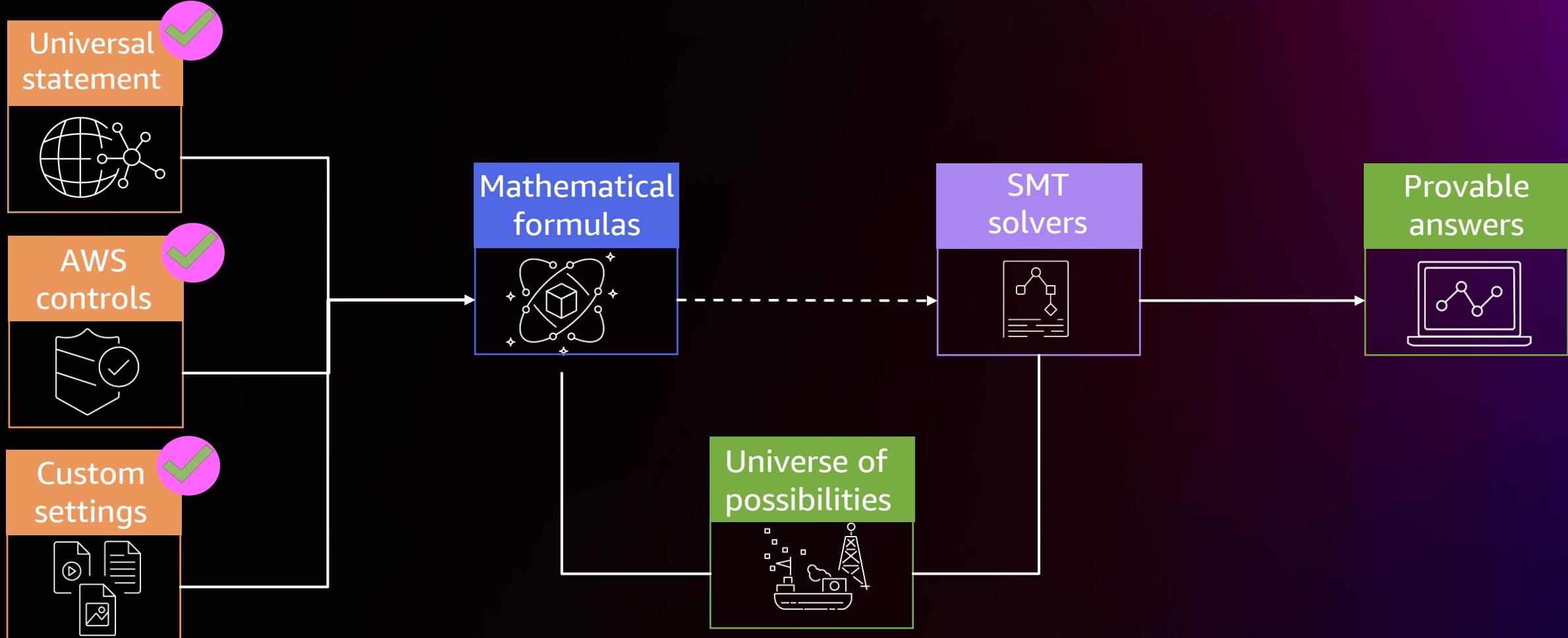
Block Public
Access (BPA)



Restricts public access for all buckets in the
account **broom-dance**, regardless of the policy
attached to the bucket

Ready to transform

Collect → Transform → Solve → Comprehensive



Transfigurations



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Encoding AWS security controls into math

Semantic-based Automated Reasoning for AWS Access Policies using SMT

John Backes, Pauline Bolignano, Byron Cook, Catherine Dodge, Andrew Gacek,
Kasper Luckow, Neha Rungta, Oksana Tkachuk, Carsten Varming
Amazon Web Services

Abstract—Cloud computing provides on-demand access to IT resources via the Internet. Permissions for these resources are defined by expressive access control policies. This paper presents a formalization of the Amazon Web Services (AWS) policy language and a corresponding analysis tool, called ZELKOVA, for verifying policy properties. ZELKOVA encodes the semantics of policies into SMT, compares behaviors, and verifies properties. It provides users a sound mechanism to detect misconfigurations of their policies. ZELKOVA solves a PSPACE-complete problem and is invoked many millions of times daily.

I. INTRODUCTION

Cloud computing provides on-demand access to IT resources via the Internet. The convenience of accessing resources in the cloud is made secure by user-specified *access control policies*. An access control policy is an expressive specification of what resources can be accessed, by whom, and under what conditions. Policy configuration is often

In this paper, we present the development and application of ZELKOVA, a policy analysis tool designed to reason about the semantics of AWS access control policies. ZELKOVA translates policies and properties into Satisfiability Modulo Theories (SMT) formulas and uses SMT solvers to check the validity of the properties. We use off-the-shelf solvers and an in-house extension of Z3 called Z3AUTOMATA.

ZELKOVA reasons about all possible permissions allowed by a policy in order to verify properties. For example, ZELKOVA can answer the questions “Is this resource accessible by a particular user?” and “Can an arbitrary user write to this resource?”. The property to be verified is specified in the policy language itself, eliminating the need for a different specification or formalism for properties. In addition, ZELKOVA provides many built-in checks for common properties.

The SMT encoding uses the theory of strings, regular



From muggle to math

My bucket **magic-potions** is only **accessible** inside **vpc-broom** or **vpc-wand**

```
(declare-fun my-property () Bool)
(assert
  (= my-property
     (=> allowed
          (or (= awsSourceVpc "vpc-broom")
              (= awsSourceVpc "vpc-wand")))))
```

From policy to math

```
{  
    "Effect": "Allow",  
    "Principal": {  
        "AWS": "11112222333"  
    },  
    "Action": "s3:GetObject",  
    "Resource": "arn:aws:s3:::magic-potions/*"  
},  
{  
    "Effect": "Deny",  
    "Principal": "*",  
    "Action": "*",  
    "Resource": "*",  
    "Condition": {  
        "NOT":  
            "StringNotEquals": {  
                "aws:SourceVpc": ["vpc-broom", "vpc-wand"]  
            }  
    }  
}
```

At least one allow matches, and no denies match

AND

```
(declare-fun principal () String)  
(declare-fun action () String)  
(declare-fun resource () String)  
(declare-fun awsSourceVpc () String)
```

```
(declare-fun s1-match () Bool)  
(assert  
(= s1-match  
( and  
(= principal "11112222333")  
(= action "s3:GetObject")  
(str.in_re resource  
(re.++  
(str.to_re "arn:aws:s3:::magic-potions/")  
re.all))))
```

OR

```
(declare-fun s2-match () Bool)  
(assert  
(= s2-match  
not( or(= awsSourceVpc "vpc-broom")  
(= awsSourceVpc "vpc-wand")))))
```

```
(declare-fun allowed () Bool)  
(assert  
(= allowed  
(and s1-match  
(not s2-match))))
```

Universe of possibilities

Model of all potential scenarios

Coverage is complete

Includes future scenarios



Universe of possibilities example

```
(declare-fun principal () String)
(declare-fun action () String)
(declare-fun resource () String)
(declare-fun awsSourceVpc () String)
(declare-fun s1-match () Bool)
(assert
 (= s1-match
    (and
     (= principal "111122223333")
     (= action "s3:GetObject")
     (str.in_re resource
      (re.++
       (str.to_re "arn:aws:s3:::magic-potions/")
       re.all)))))

(declare-fun s2-match () Bool)
(assert
 (= s2-match
    (not (or (= awsSourceVpc "vpc-broom")
              (= awsSourceVpc "vpc-wand")))))

(declare-fun allowed () Bool)
(assert
 (= allowed
    (and s1-match
         (not s2-match))))
```

000000000000 calling s3:AbortMultiPartUpload from vpc-0000000?

000000000000 calling s3:AbortMultiPartUpload from vpc-0000001?

...

111122223333 calling s3:GetObject from vpc-broom?

...

111122223333 calling s3:GetObject from vpc-wand?

...

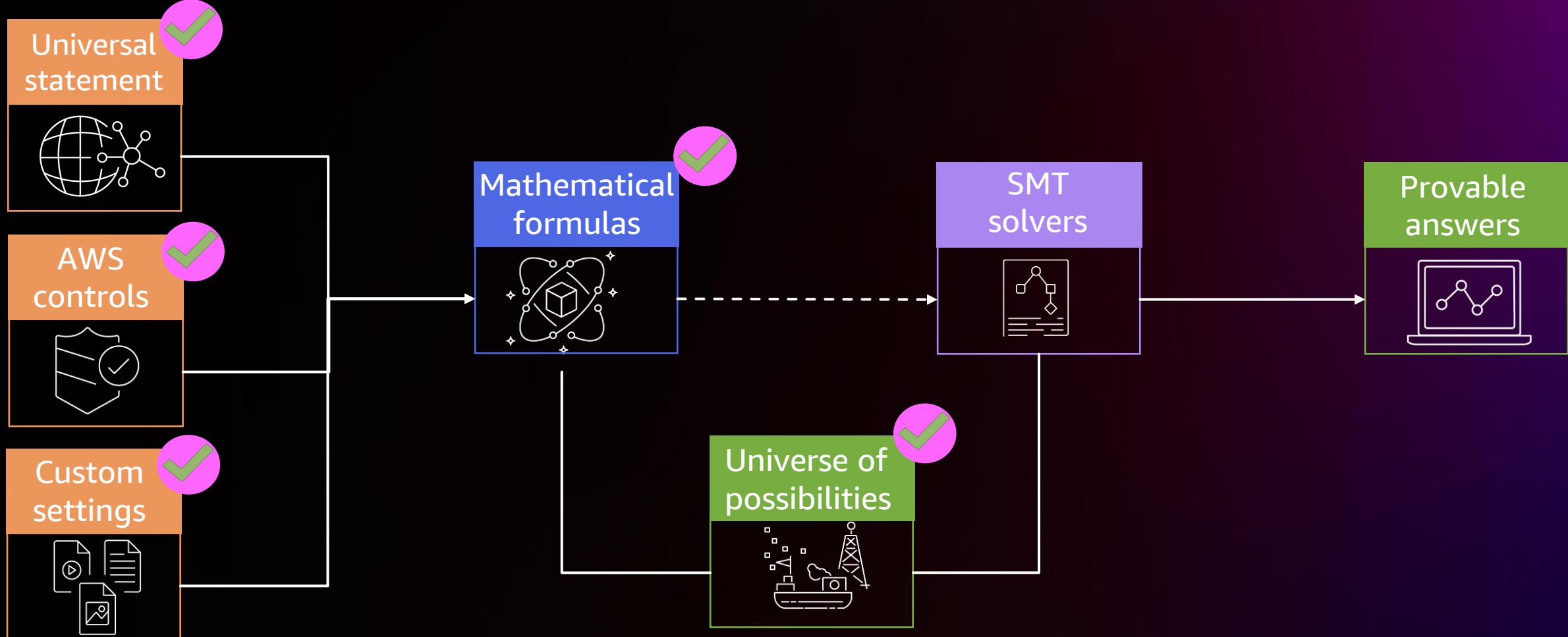
999999999999 calling s3:UpdateJobStatus from vpc-ffffffe?

999999999999 calling s3:UpdateJobStatus from vpc-fffffff?



Ready to solve

Collect → Transform → Solve → Comprehensive



SMT solvers

Theorem prover for satisfiability modulo theories

Solves against constraints

Combines proven algorithms

Optimized for real-world scenarios

Examples:

Z3

CVC4

CVC5



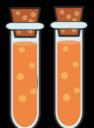
Solve for a path

Configurations

```
(assert  
  (= s1-match  
      (and  
        (= principal "111122223333"))  
    ...))
```

Universal statement

```
(declare-fun my-property () Bool)  
(assert  
  (= my-property  
      (=> allowed  
        (or (= awsSourceVpc "vpc-broom")  
            (= awsSourceVpc "vpc-wand")))))
```



Solver potion

```
(assert  
  (not my-property))
```



Provable answer

```
False  
"No path found"
```

Demo: Solve

Demo!

Question

My bucket `magic-potions` is accessible only inside `vpc-broom` or `vpc-wand`

Path

Is there an access path that is not in `vpc-broom` or `vpc-wand`?

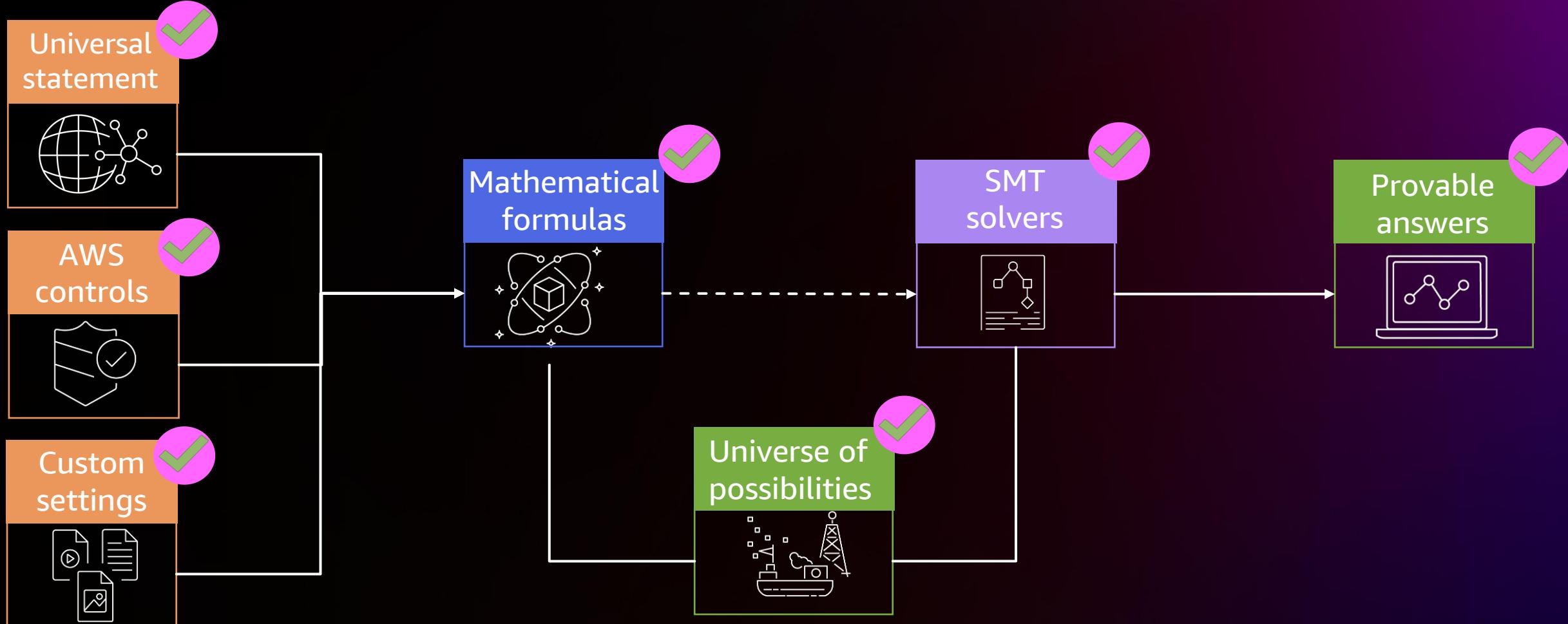
Demo steps

1. Input the universal statement (in math)
2. Input the policy (in math)
3. Use an SMT solver



Automated reasoning

Collect → Transform → Solve → Comprehensive



Brewing comprehensive potions

Advanced automated reasoning



More complex questions



Solving with speed



Solving at scale



More complex questions

“My KMS keys are only accessed from inside my AWS organization”

“My bucket is not shared beyond my account”



Multiple universal statements for one question

“Does Pickles have cross-account access to **apple-bucket**? ”

Universal statement



The resource policy grants public access

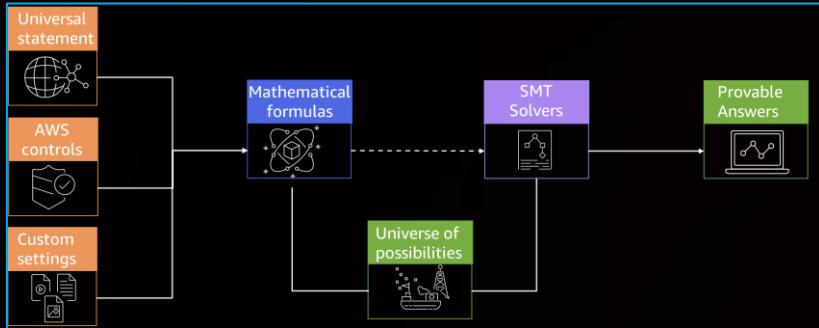
The access point policy grants public access

Pickles has admin access to **apple-bucket**

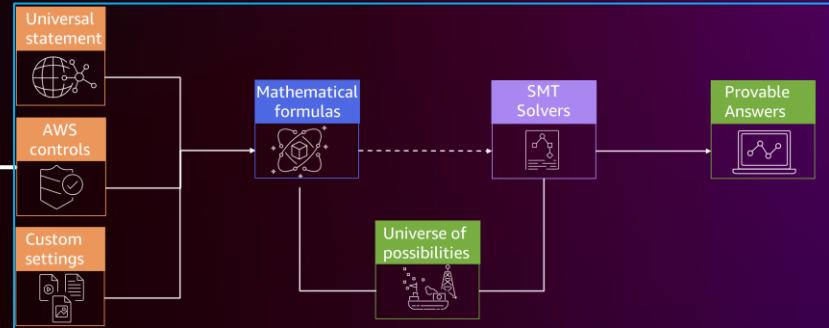
Pickles has eat access to **apple-bucket**

Combining provable answers

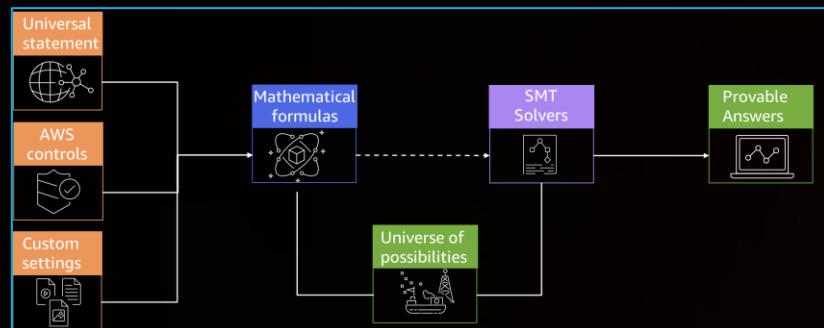
Resource policy grants public access



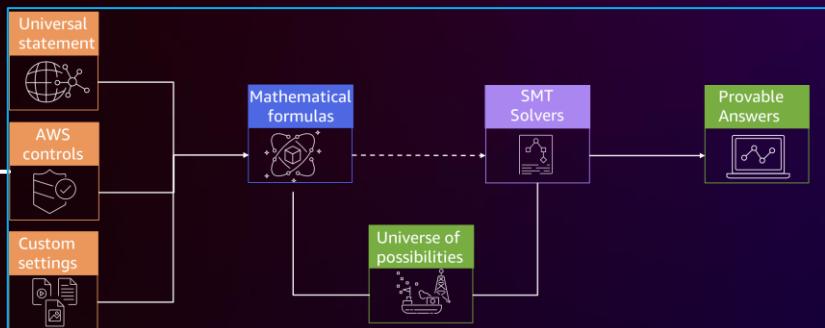
Pickles has admin access to **apple-bucket**



Access point policy grants public access

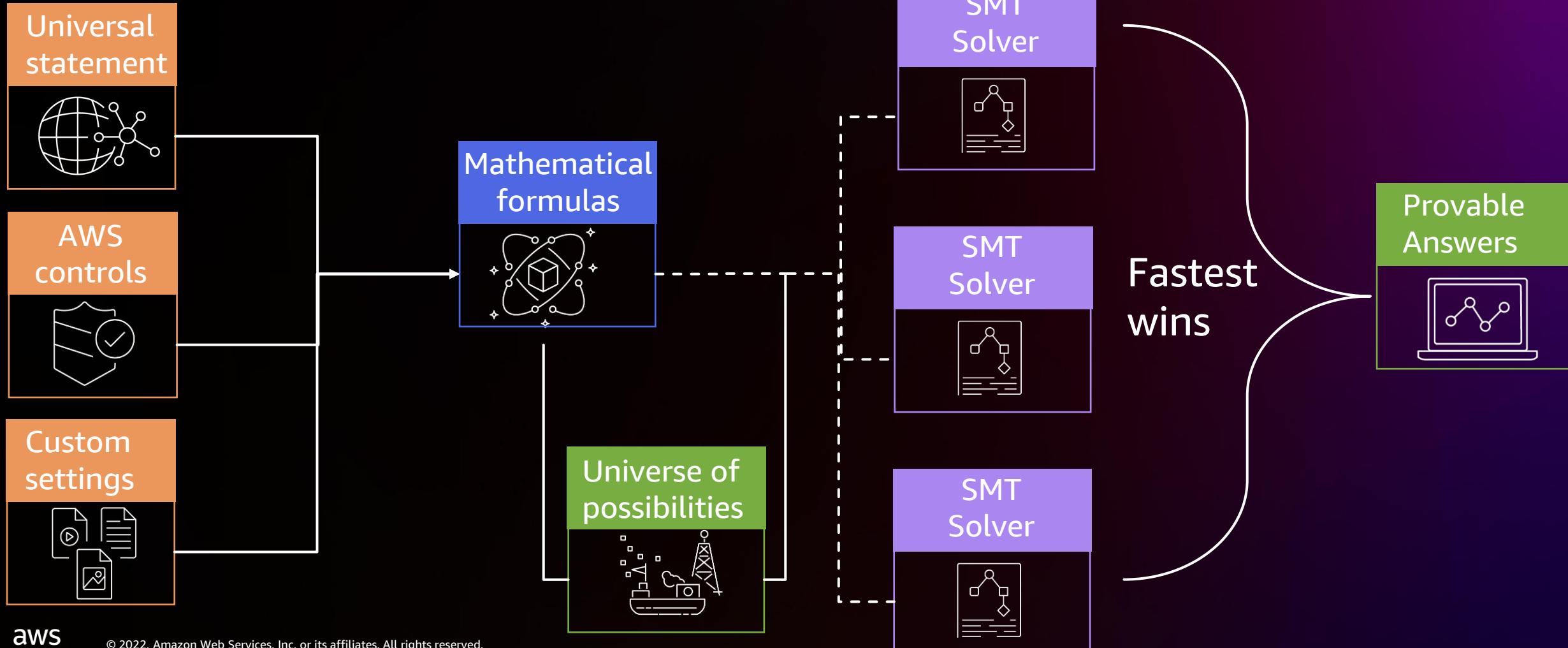


Pickles has eat access to **apple-bucket**



Solving with speed

Collect → Transform → Solve → Comprehensive



Solving at scale

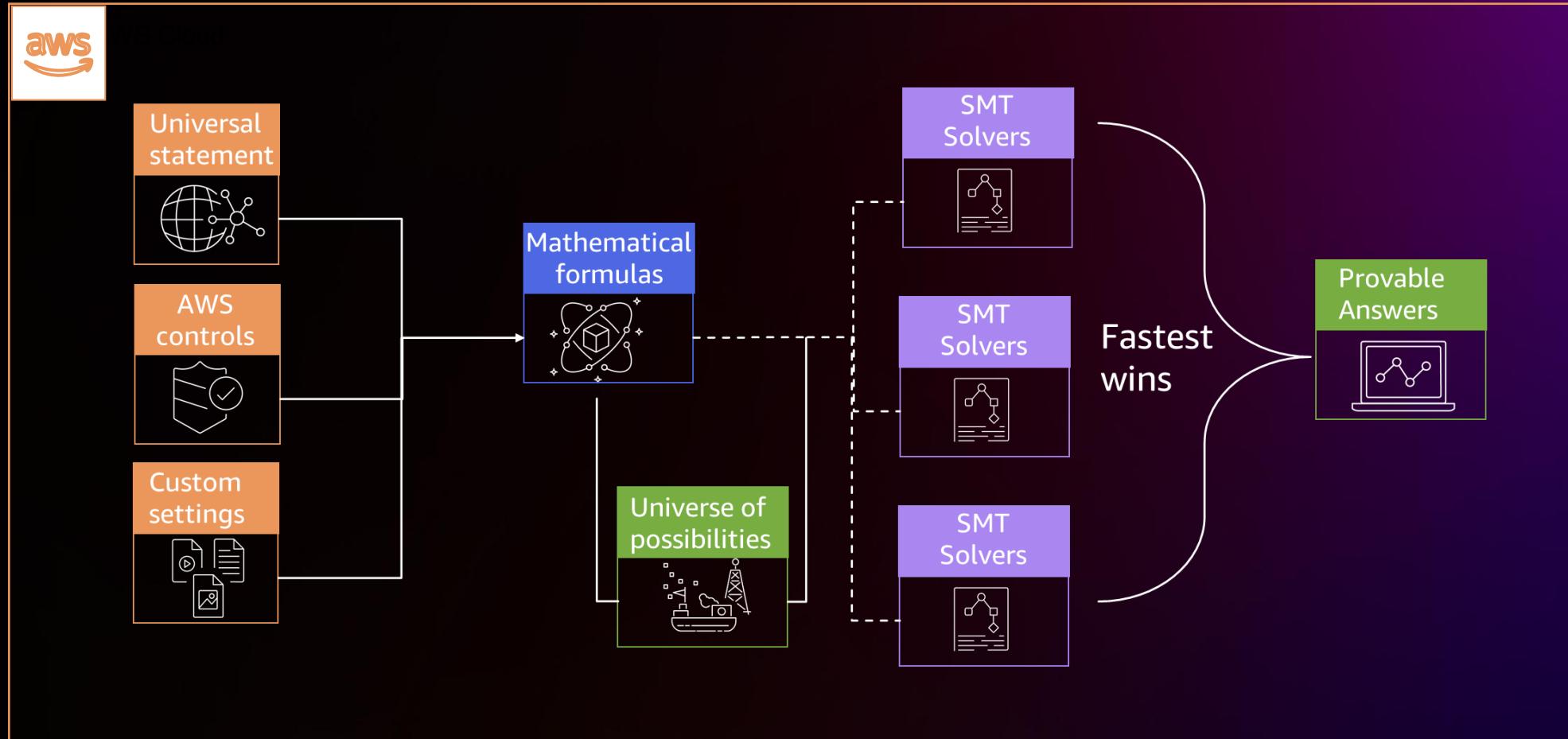
AWS runs over a billion SMT queries a day and growing



Compute



Blog link



Security charms in AWS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Managed automated reasoning



IAM Access Analyzer



VPC Network Access Analyzer



Block Public Access



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



“Show me public access”

Use IAM Access Analyzer to review public findings for 14 resource types

- ★ **Enable** IAM Access Analyzer in your account or organization
- ★ Continuously **monitors and reviews** access controls across 14 resource types
- ★ Uses **automated reasoning** to determine public or cross-account access
- ★ **Generates findings** for you to review and determine if they match your intent
- ★ **Verify** public and cross-account access by previewing access as you modify resource policies

Review public access findings

Demo!

Question

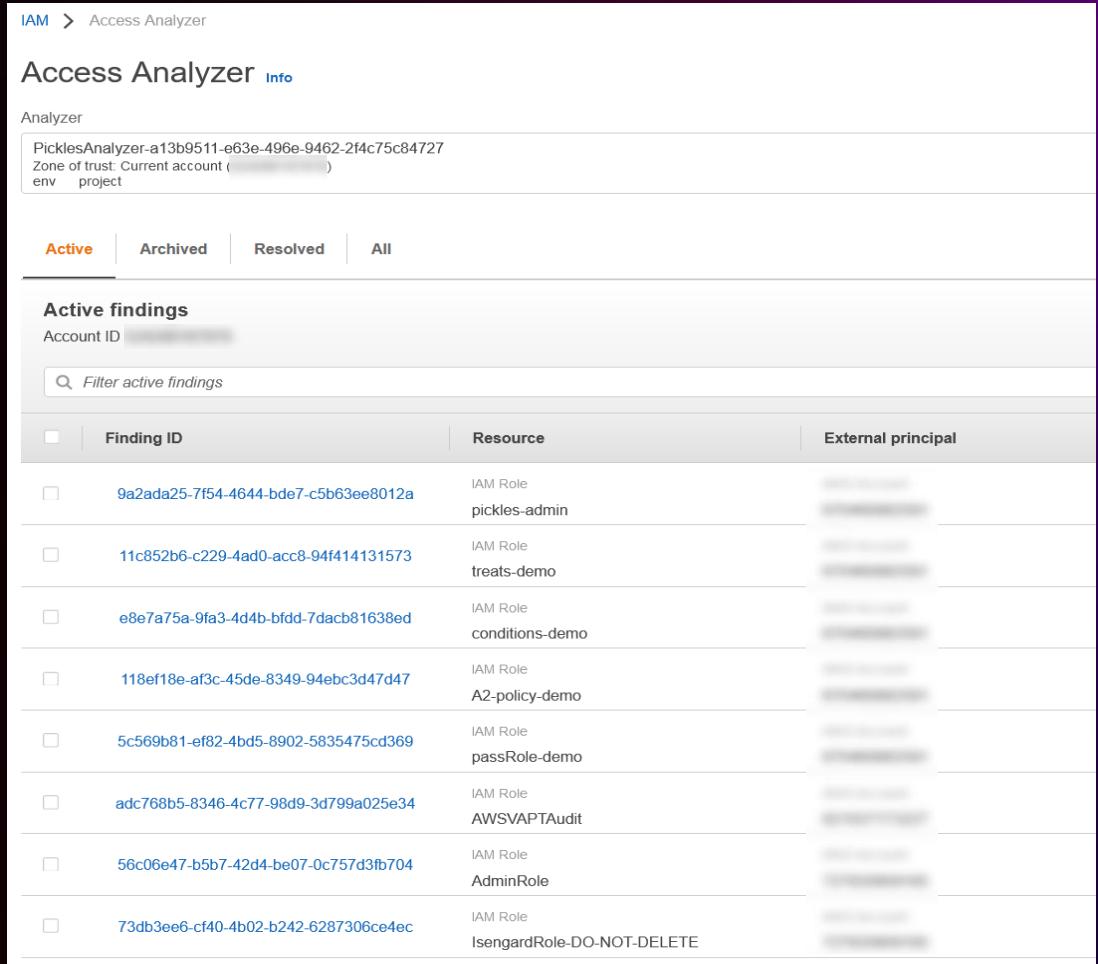
Which of my buckets are public?

Path

For each bucket, is there a path to public?

Demo steps

1. Prep: Enable IAM Access Analyzer
2. Review public findings
3. Investigate if public is intended



The screenshot shows the AWS IAM Access Analyzer interface. At the top, it displays the Analyzer configuration: "Analyzer" name is "PicklesAnalyzer-a13b9511-e63e-496e-9462-2f4c75c84727", "Zone of trust" is "Current account ([REDACTED])", and "env" and "project" are listed. Below this, there are tabs for "Active" (which is selected), "Archived", "Resolved", and "All". The main section is titled "Active findings" and shows a table of findings. The columns are "Finding ID", "Resource", and "External principal". There are eight entries in the table:

Finding ID	Resource	External principal
9a2ada25-7f54-4644-bde7-c5b63ee8012a	IAM Role	pickles-admin
11c852b6-c229-4ad0-acc8-94f414131573	IAM Role	treats-demo
e8e7a75a-9fa3-4d4b-bfdd-7dacb81638ed	IAM Role	conditions-demo
118ef18e-af3c-45de-8349-94ebc3d47d47	IAM Role	A2-policy-demo
5c569b81-ef82-4bd5-8902-5835475cd369	IAM Role	passRole-demo
adc768b5-8346-4c77-98d9-3d799a025e34	IAM Role	AWSVAPTAudit
56c06e47-b5b7-42d4-be07-0c757d3fb704	IAM Role	AdminRole
73db3ee6-cf40-4b02-b242-6287306ce4ec	IAM Role	IsengardRole-DO-NOT-DELETE



“Show me the future”

Demo!

Use IAM Access Analyzer to see how policy updates change the future

Verify public and cross-account access by previewing access as you modify resource policies

Question

Does this policy update resolve or grant public access?

Path

Is there a path to public for this policy?

Demo steps

1. Update bucket policy to remediate public access
2. Review resolved and new findings

The screenshot shows the AWS IAM Access Analyzer interface. At the top, there is a button labeled "+ Add new statement". Below it, the JSON code is displayed: "JSON Ln 7, Col 14". A status bar indicates: Security: 0, Errors: 0, Warnings: 0, and Suggestions: 0. A section titled "Preview external access - optional" includes a note: "Preview and validate Access Analyzer findings for external access to your resource". Below this is the "Analyzer" section, which shows the ARN of the analyzer: "TreatAnalyzer-49e075a3-ae64-429e-9db3-810db324f208" and the zone of trust: "Current account (169026632266)". It also lists "env" and "project" under the zone of trust. At the bottom, there is a navigation bar with tabs: All (highlighted), New, Resolved, Archived, and Existing. Below the tabs is a search bar with the placeholder "Filter all findings". A summary message states: "Existing Public All principals have read access."



“Block public access”

Enable Block Public Access for S3 buckets

-  **Enable** BPA for Amazon S3 in your account on your bucket
-  **Restricts** access to an access point or bucket policy that grants public access
-  **Rejects** calls to attach access point or bucket policy if the specified policy allows public access

Block Public Access demo

Demo!

Universal statement

Block public access to all my buckets

Path

Is there a path to public?

Demo steps

1. Prep: Enable BPA for an account
2. Edit a bucket policy to grant public access
3. Watch the rejection

Block Public Access settings for this account

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies or all. In order to ensure that public access is blocked, turn on one or more of the following settings. You can turn off Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require public access to specific buckets or objects, turn on the specific setting.

Block **all** public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through **new** access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets.
- Block public access to buckets and objects granted through **any** access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through **new** public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies.
- Block public and cross-account access to buckets and objects through **any** public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



“Tell me who can talk to the internet”

Use VPC Network Access Analyzer to review network configurations

- ▶ VPC Network Access Analyzer helps you **identify unintended network access** relative to your security and compliance requirements
- ▶ Helps you **verify** network segmentation, internet accessibility, trusted network paths, and trusted network access
- ▶ Start with four **purpose-built network scopes** or create your own
- ▶ Uses **automated reasoning** to help you understand network access
- ▶ Use VPC Network Access Analyzer in the **console or API**

Internet accessibility demo

Demo!

Question

Which resources can be accessed from internet gateways?

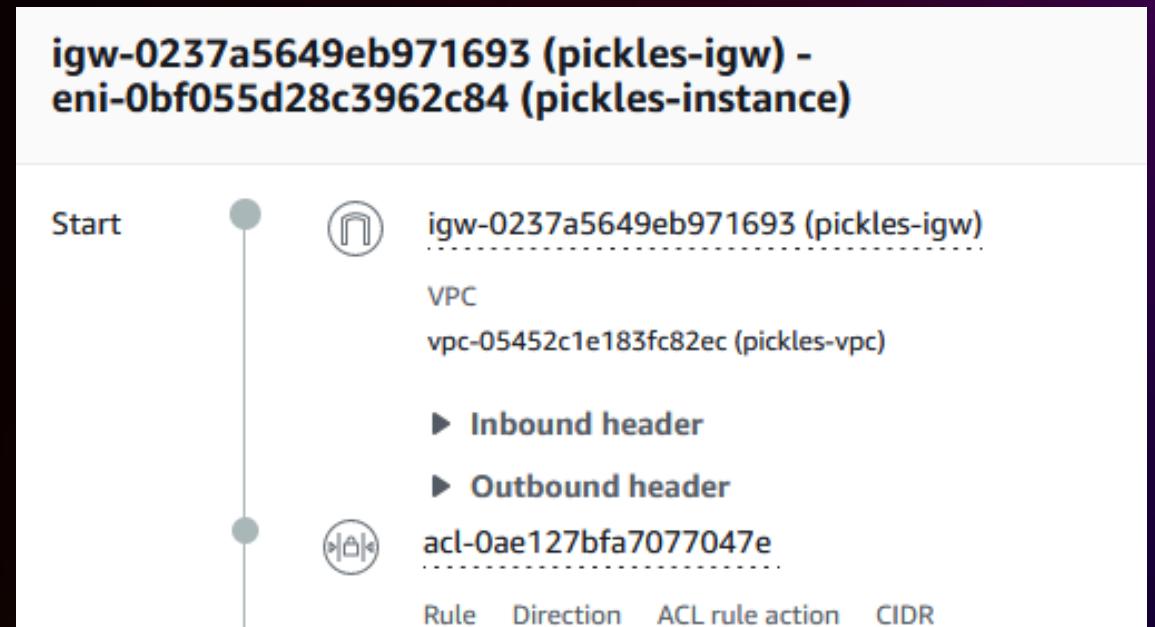
Path

For each network control, prove if there is a path to the internet

Demo steps

1. Navigate to VPC Network Access Analyzer
2. Review internet accessible findings
3. Verify they match your intent

Findings (1/1) Info			
<input type="text"/> Filter findings by resource types or specific resources present in the findings.			
Start	End	Protocol	
(pickles-igw) igw-0237a5649eb971693	(pickles-instance) i-0fa0b46694d6e05bc	TCP	



AWS security charms

Show me public access

Show me cross-account access

Block Public Access

Tell me who can talk to the internet

Tell me who the internet can talk to

What will come next?



Our favorite security alchemy resources

How AWS uses automated reasoning to help you achieve security at scale

AWS Security Blog

How AWS uses automated reasoning to help you achieve security at scale

by Andrew Gacek | on 20 JUN 2018 | in Security, Identity, & Compliance | Permalink | [Comments](#) | [Share](#)

At AWS, we focus on achieving security at scale to diminish risks to your business. Fundamental to this approach is ensuring your policies are configured in a way that helps protect your data, and the Automated Reasoning Group (ARG), an advanced innovation team at AWS, is using [automated reasoning](#) to do it.

Provable Security

How it works

We apply automated reasoning in key service areas such as storage, networking, virtualization, identity, and cryptography. You can work in Amazon CodeGuru, Amazon Simple Storage Service (Amazon S3), AWS Identity and Access Management (IAM), Amazon VPC, and the Amazon VPC Reachability Analyzer.



Dive Deep into IAM Access Analyzer (SEC309)

AWS re:Invent 2019: [NEW LAUNCH!] Dive Deep into IAM Access Analyzer

6.8K views • 2 years ago

AWS Events

AWS Identity and Access Management Access Analyzer is a new capability for

Meet the team | Automated Reasoning | Getting Started | Access

Achieve the highest level of assurance with provable security

AWS Public Sector Summit Online 2021: Achieve the highest level of assurance with provable security

31 views • 1 year ago

AWS Public Sector

Learn how AWS achieves provable security protects at scale. You will hear ingredients of the secret sauce on how automated ...

Provable security | Mathematical logic | Pythagorean theorem | Where is it used? | Foundational... 20 moments



Thank you!

Brigid Johnson
@bjohnso5y

Neha Rungta
@neharungta



Please complete the session
survey in the **mobile app**

