Detecting Multi-Step IAM Attacks in AWS Environments via Model Checking

Ilia Shevrin[1], Oded Margalit[2]

[1]Citi
[2]Ben-Gurion University

Background

Cloud adoption is on the rise, more data is stored in the cloud

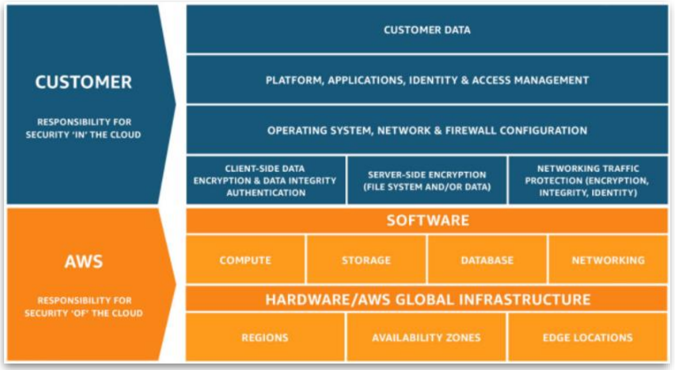
Security posture of cloud applications is a growing concern

AWS introduced the shared responsibility model:

AWS is responsible for **infrastructure and hardware**

The customer is responsible for **application, data and IAM**

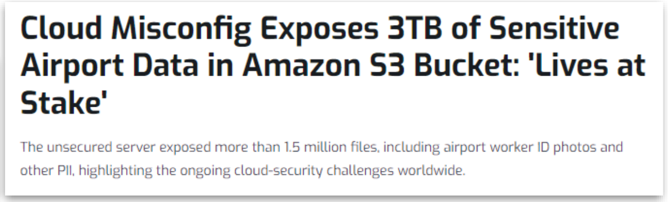
But IAM is notoriously hard to master due to its complexity



CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

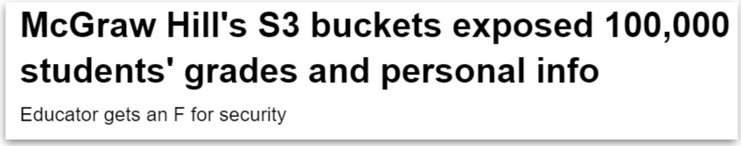
Image from

https://docs.aws.amazon.com/whitepapers/latest/security-overview-of-amazon-codeguru-reviewer/the-shared-responsibility-model.html



Cloud Misconfig Exposes 3TB of Sensitive Airport Data in Amazon S3 Bucket: 'Lives at Stake'

The unsecured server exposed more than 1.5 million files, including airport worker ID photos and other PII, highlighting the ongoing cloud-security challenges worldwide.



McGraw Hill's S3 buckets exposed 100,000 students' grades and personal info

Educator gets an F for security

IAM Misconfiguration Example

Alice

```
"Effect ": "allow",
"Action ": "*Role*",
"Resource ": "*"

"Effect ": "deny",
"Action ": "*Role*",
"Resource ": "Alice"
```

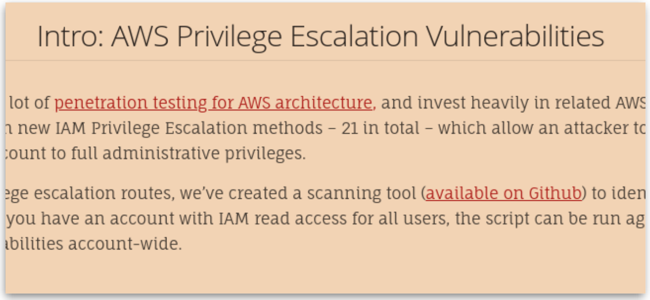
bobs-bucket

```
"Effect ": "deny",
"Action ": "*",
"Resource ": "bobs-bucket/*",
"Principal ": "Alice"
```

```
1.  CreateRole Carol

1.  PutRolePolicy Carol
    {“effect”:”allow”,
     “action”:”*”,
     “resource”:”*”}

1.  AssumeRole Carol

1.  GetObject bobs-bucket/*
```

Existing AWS IAM Security Tools

Rhino Security Labs identified 20+ AWS IAM privilege escalation techniques and released Pacu - an open source tool that scans policies and detects potential usage of these techniques



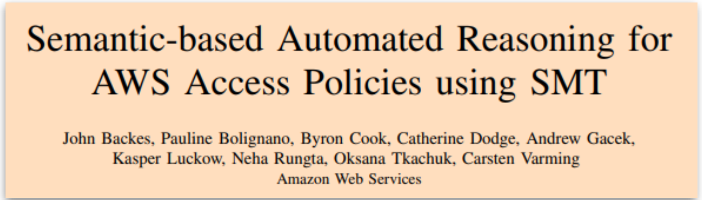
Intro: AWS Privilege Escalation Vulnerabilities

lot of penetration testing for AWS architecture, and invest heavily in related AWS
n new IAM Privilege Escalation methods – 21 in total – which allow an attacker to
count to full administrative privileges.

ege escalation routes, we've created a scanning tool (available on Github) to iden
you have an account with IAM read access for all users, the script can be run ag
abilities account-wide.

RhinoSecurityLabs / pacu

AWS developed Zelkova that mathematically verifies properties in IAM policies - for example checking if a bucket is public

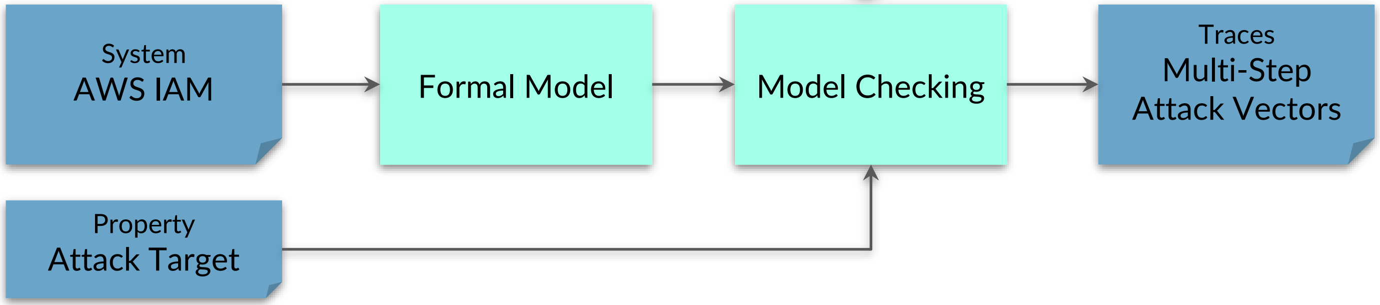


Semantic-based Automated Reasoning for AWS Access Policies using SMT

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

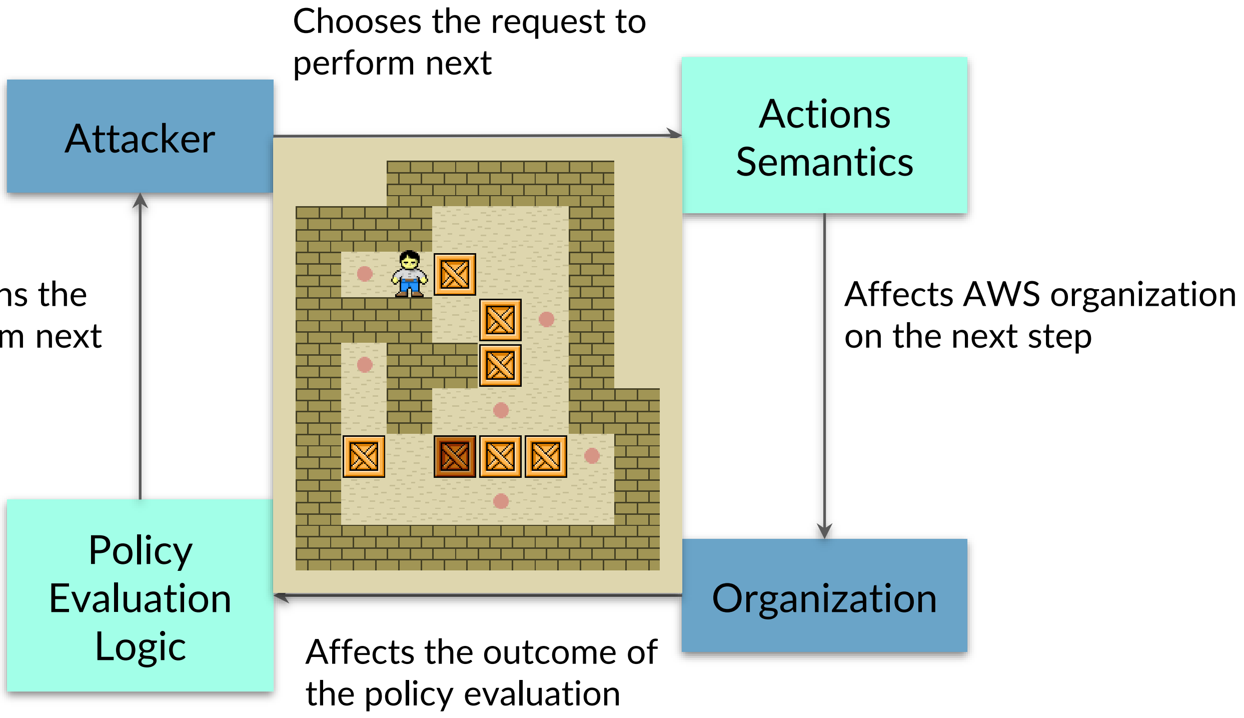
A Model Checking Approach

Known technique to check whether a model of a system satisfies certain properties



Was already suggested in the context of network vulnerabilities by Ritchey & Ammann in 2000

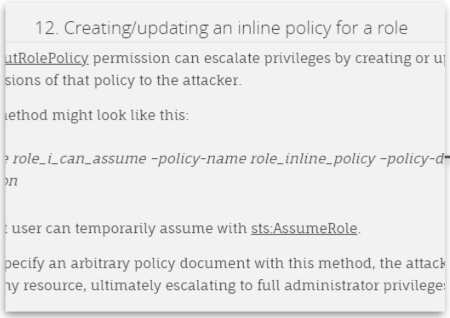
AWS IAM Model


Chooses the request to perform next
Attacker
Actions Semantics
Impacts what actions the attacker can perform next
Affects AWS organization on the next step
Policy Evaluation Logic
Organization
Affects the outcome of the policy evaluation

Actions Semantics Encoding

Recognize only *means* actions - affect IAM directly or indirectly (in total around 60 AWS actions)

Recognize a privilege escalation prone action



12. Creating/updating an inline policy for a role

utRolePolicy permission can escalate privileges by creating or u
sions of that policy to the attacker.

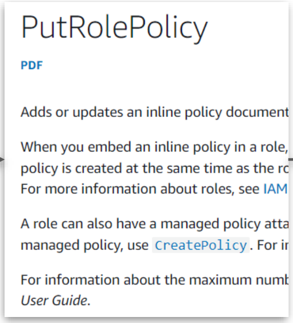
ıethod might look like this:

e role_i_can_assume –policy-name role_inline_policy –policy-d
on

: user can temporarily assume with sts:AssumeRole.

pecify an arbitrary policy document with this method, the attack
ny resource, ultimately escalating to full administrator privilege

Understand AWS documentation and translate into an action semantics formula



PutRolePolicy

PDF

Adds or updates an inline policy document

When you embed an inline policy in a role,
policy is created at the same time as the ro
For more information about roles, see IAM

A role can also have a managed policy atta
managed policy, use `CreatePolicy`. For ir

For information about the maximum numb
User Guide.

```
action = PutRolePolicy implies
  forall account in accounts:
    forall role in account.IAMRoles:
      ((resourceAccount = account.id and
      resourceName = role.name) implies
            role.MaximumPermissions' = true) and
      ((resourceAccount != account.id or
      resourceName != role.name) implies
            role.MaximumPermissions' =
            role.MaximumPermissions)
```

Rhino Security Labs website at
https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/

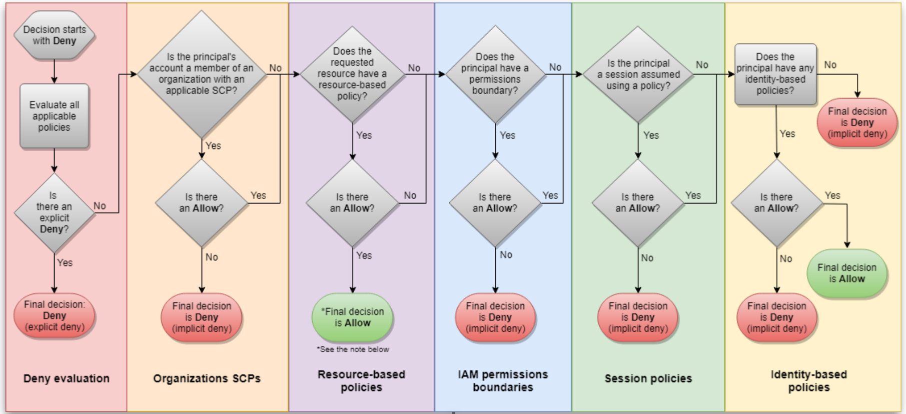
AWS documentation at
https://docs.aws.amazon.com/IAM/latest/APIReference/API_PutRolePolicy.html

Policy Evaluation Logic Encoding

```
"Effect ": "allow",
"Action ": "*Role*",
"Resource ": "*"

"Effect ": "deny",
"Action ": "*Role*",
"Resource ": "Alice"
```

Encode formulas similarly to Zelkova

```
allowStatements =
  action in “*Role*” and
  resourceType = Role

denyStatements =
  action in “*Role*” and
  resourceName = “Alice” and
  resourceType = Role
```



Decision starts with Deny
Evaluate all applicable policies
Is there an explicit Deny?
Yes
No
Final decision: Deny (explicit deny)
Deny evaluation
Is the principal's account a member of an organization with an applicable SCP?
No
Yes
Is there an Allow?
Yes
No
Final decision is Deny (implicit deny)
Organizations SCPs
Does the requested resource have a resource-based policy?
No
Yes
Is there an Allow?
No
Yes
*Final decision is Allow
*See the note below
Resource-based policies
Does the principal have a permissions boundary?
No
Yes
Is there an Allow?
Yes
No
Final decision is Deny (implicit deny)
IAM permissions boundaries
Is the principal a session assumed using a policy?
No
Yes
Is there an Allow?
Yes
No
Final decision is Deny (implicit deny)
Session policies
Does the principal have any identity-based policies?
No
Final decision is Deny (implicit deny)
Yes
Is there an Allow?
Yes
Final decision is Allow
No
Final decision is Deny (implicit deny)
Identity-based policies

```
ServiceControlPoliciesAllow and
(ResourceBasedPoliciesAllow or
(IdentityBasedPoliciesAllow and
PermissionsBoundariesAllow)) and

not ServiceControlPoliciesDeny and
not ResourceBasedPoliciesDeny and
not PermissionsBoundariesDeny and
not IdentityBasedPoliciesDeny
```

Implementation and Evaluation

We implemented the model checking process using Java + Z3 SAT Solver API. We used a bounded model checking algorithm (BMC) + an exhaustive version

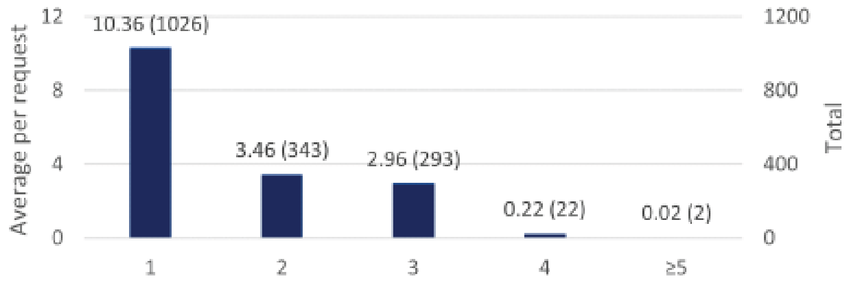
We use a large pre-production AWS organization with ~100 accounts, with an average of ~200 IAM resources in each account

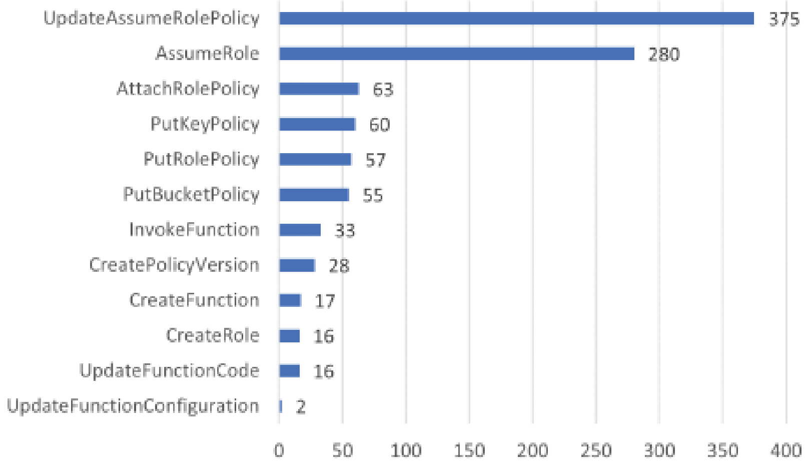
We ask how does the model checking process manage to detect existing misconfigurations in real world AWS environments

Gathered 141 different requests from security engineers, testing who can get access to data resources such as S3 buckets or SQS queues

Evaluation Results

Attack vectors by length	Actions performed in all the attack vectors





- ❑ Security engineers were satisfied with the results and fixed a lot of sneaky over-permissive policies
- ❑ Additional performance evaluation showed that the approach detects IAM attacks of up to 5 steps, in accounts with hundreds of resources, in under a minute

Thank you!

Questions?

iliashevrin@mail.tau.ac.il