CAL POLY

CSC 321: Introduction to Computer Security

Module 5: Access Control

Bret Hartman

Department of Computer Science and Software Engineering

California Polytechnic State University

E-mail: bahartma@calpoly.edu

A special thanks to Dr. Bruce DeBruhl and Dr. Phoenix (Dongfeng) Fang, the authors of most of this material

Quiz 4 available
Assignment 4 Hashing and Passwords due

Capital One

Responsible Disclosure (Shared) <responsibledisclosure@capitalone.com>

[External Sender] Leaked s3 data

[redacted]

Wed, Jul 17, 2019 at 1:25 AM

To: "responsibledisclosure@capitalone.com" <responsibledisclosure@capitalone.com>

Hello there,

There appears to be some leaked s3 data of yours in someone's github / gist:

https://gist.github.com[redacted]

Let me know if you want help tracking them down.

Thanks,

[redacted]

The tip that alerted Capital One to its data breach.

[redacted]

Paige "erratic" Thompson, in an undated photo posted to her Slack channel.

VPC = AWS virtual private cloud
Aurora = AWS database
SSM = AWS systems manager

#netcrave

14 | 5 | Never give up on your dreams

```
total 485G
drwxr-xr-x   7 erratic root  4.0K Jun 27 15:31 .
-rw-r--r--   1 erratic users  55K Jun 27 00:00 42lines.net.tar.xz
drwxr-xr-x  12 root    root  4.0K May 29 09:26 ..
drwxr-xr-x 669 erratic users  36K Jun 27 18:23 ISRM-WAF-Role
-rw-r--r--   1 erratic users  28G Jun 27 18:55 ISRM-WAF-Role.tar.xz
-rw-r--r--   1 erratic users  35G Jun 27 15:31 Rotate_Access_key.tar.xz
-rw-r--r--   1 erratic users  25G Jun 27 10:08 apperian.tar.xz
-rw-r--r--   1 erratic users  264 Jun 27 00:00 apperian2.tar.xz
-rw-r--r--   1 erratic users  12K Jun 27 00:00 astem.tar.xz
-rw-r--r--   1 erratic users  28G Jun 27 09:46 cicd-instance.tar.xz
drwxr-xr-x  67 erratic users 4.0K Jun 27 18:50 code_deploy_role
-rw-r--r--   1 erratic users  59G Jun 27 18:55 code_deploy_role.tar.xz
drwxr-xr-x  39 erratic users  12K Jun 27 15:24 ec2_s3_role
-rw-r--r--   1 erratic users  76G Jun 27 18:55 ec2_s3_role.tar.xz
-rw-r--r--   1 erratic users 9.8G Jun 27 13:16 ecs.tar.xz
-rw-r--r--   1 erratic users 2.3G Jun 27 03:26 ford.tar.xz
-rw-r--r--   1 erratic users 224M Jun 27 00:06 fuckup.tar.xz
-rw-r--r--   1 erratic users  38G Jun 27 15:28 globalgarner.tar.xz
-rw-r--r--   1 erratic users  408 Jun 27 00:00 hslonboarding-prod-backup1.tar.xz
-rw-r--r--   1 root    root  8.0G Jun  3 23:11 identiphy.img
-rw-r--r--   1 erratic users 1.4M Jun 27 00:00 identiphy.tar.xz
-rw-r--r--   1 erratic users 204K Jun 27 00:00 infobloxcto.tar.xz
-rw-r--r--   1 erratic users  13G Jun 27 03:15 iwcodeacademy.tar.xz
2:56 PM -rw-r--r--   1 erratic users 408M Jun 27 00:54 s3_logrotate_role.tar.xz
-rw-r--r--   1 erratic users 356M Jun 27 04:45 safesocial.tar.xz
-rw-r--r--   1 erratic users 4.5G Jun 27 04:10 service_devops.tar.xz
-rw-r--r--   1 erratic users  11G Jun 27 07:29 starofservice.tar.xz
drwxr-xr-x   9 erratic users 4.0K Jun 27 17:57 unicredit
```

KrebsonSecurity

In-depth security news and investigation

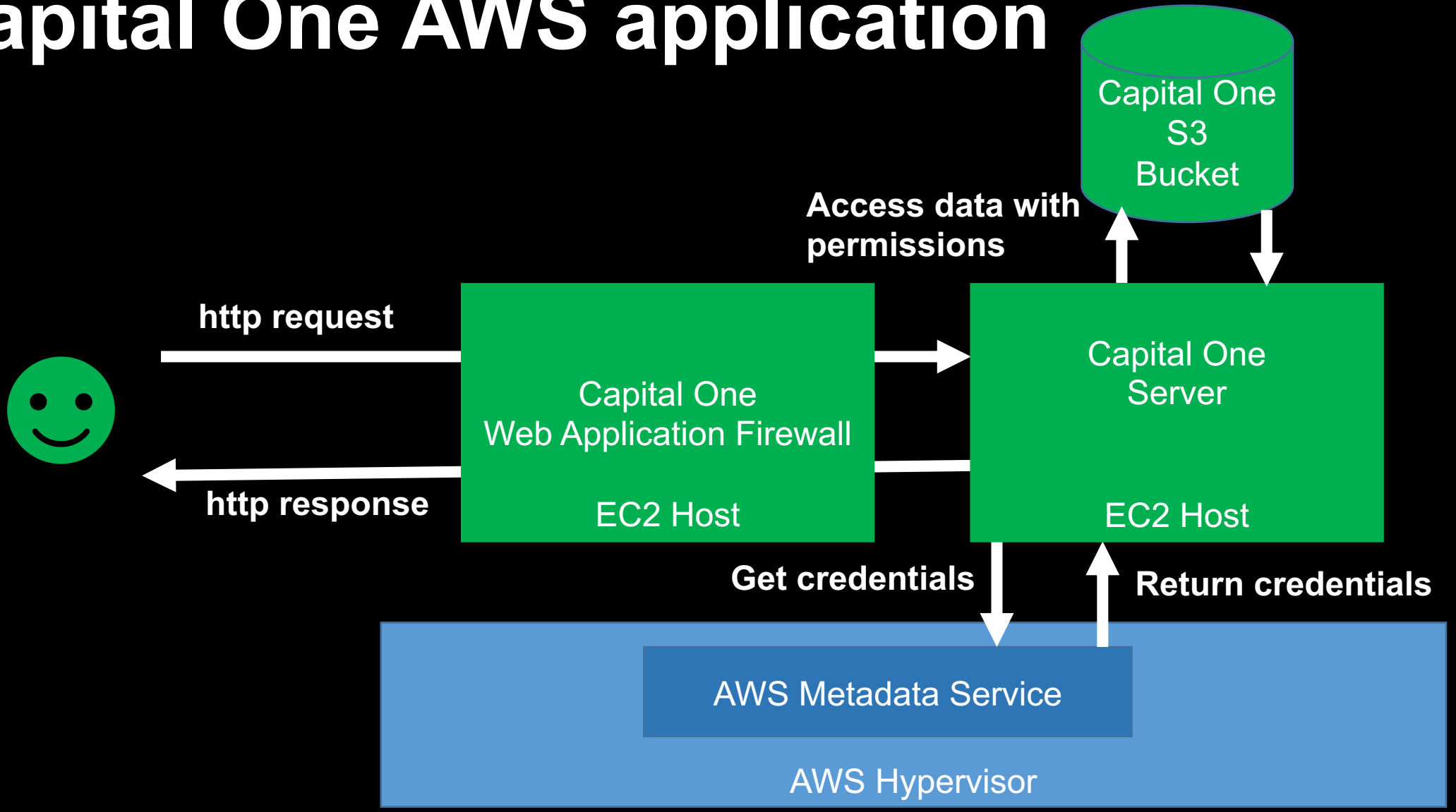
HOME ABOUT THE AUTHOR ADVERTISING/SPEAKING

Capital One Data Theft Impacts 106M People

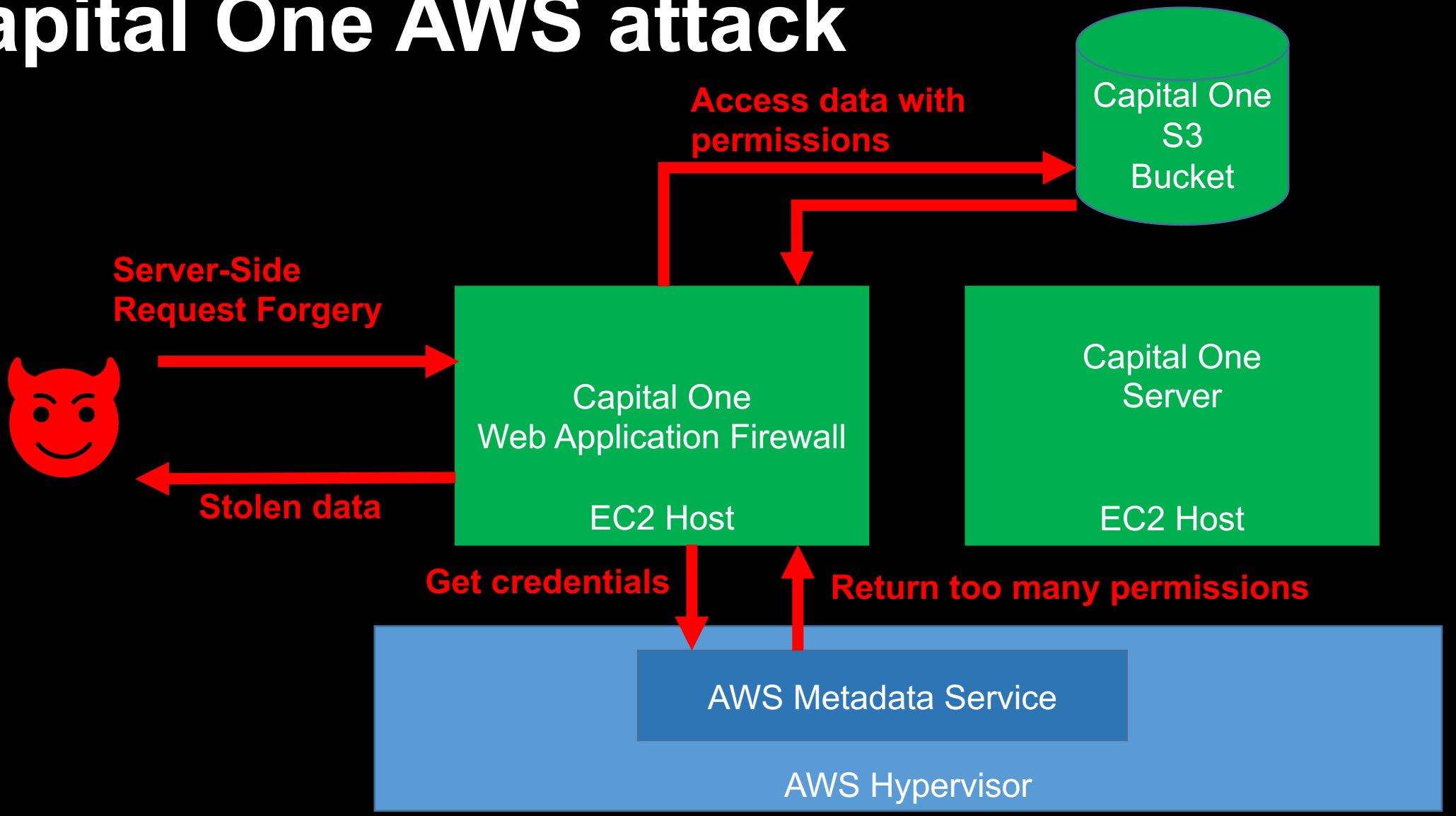
July 30, 2019 257 Comments

Federal prosecutors this week charged a Seattle woman with stealing data from more than 100 million credit applications made with **Capital One Financial Corp**. Incredibly, much of this breach played out publicly over several months on social media and other open online platforms. What follows is a closer look at the accused, and what this incident may mean for consumers and businesses.

Capital One AWS application



Capital One AWS attack



[Preventing The Capital One Breach]

Credit Cards Checking & Savings Auto Business Commercial Learn & Grow

Information on the Capital One Cyber Incident

What happened

On July 19, 2019, we determined that an outside individual gained unauthorized access and obtained certain types of personal information about Capital One credit card customers and individuals who had applied for our credit card products.

RICHARD D. FAIRBANK, CHAIRMAN AND CEO

"While I am grateful that the perpetrator has been caught, I am deeply sorry for what has happened...I sincerely apologize for the understandable worry this incident must be causing those affected and I am committed to making it right."

SECURITY / POLICY / TECH

Seattle hacker gets probation for $250M Capital One data breach

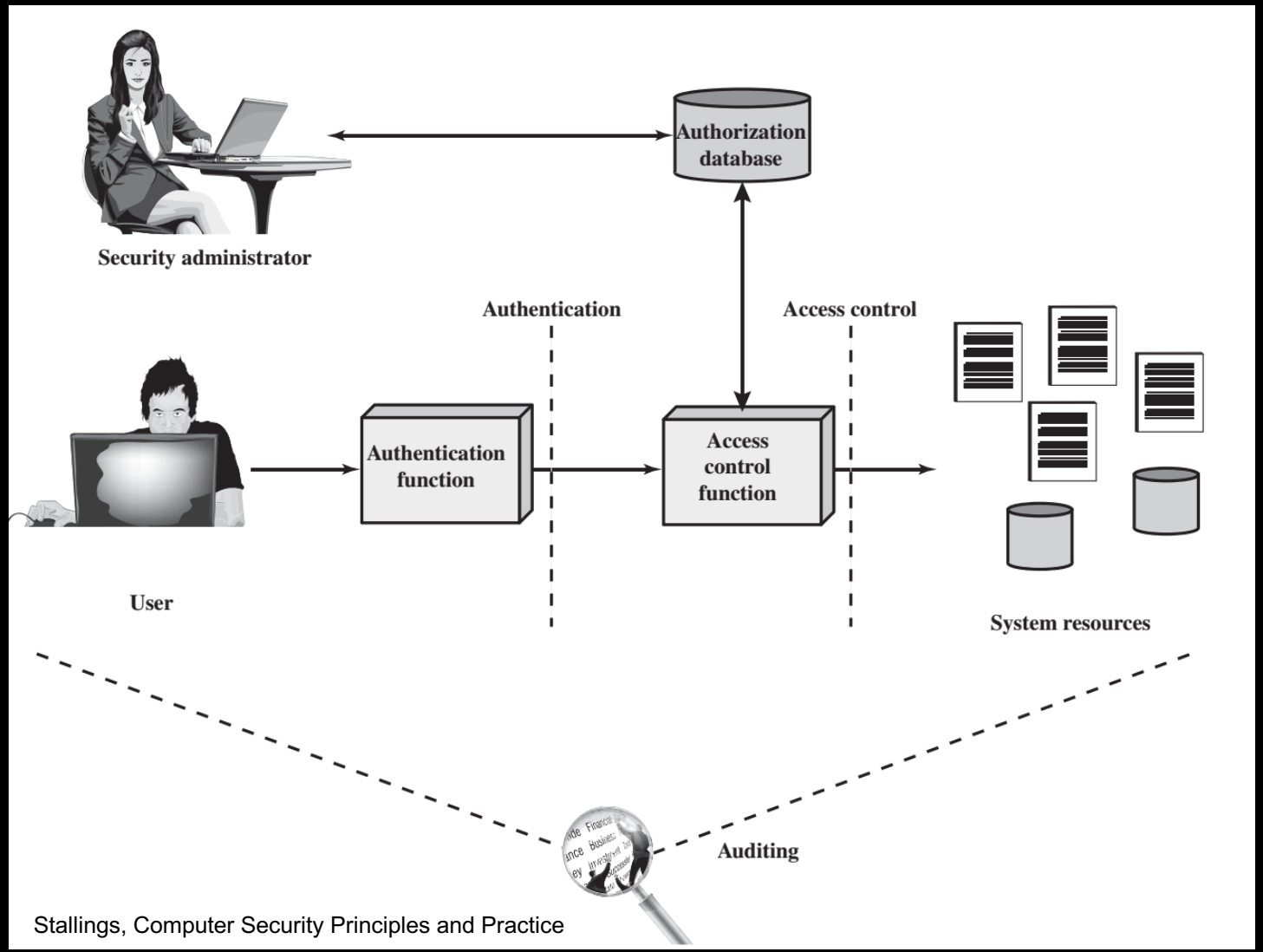
Photo by Amelia Holowaty Krales / The Verge

/ Software engineer Paige Thompson was given time served plus five years of probation for the 2019 incident

By CORIN FAIFE / @corintxt

Oct 5, 2022, 9:41 AM PDT | 4 Comments / 4 New

Authentication, Authorization, and Accounting



Stallings, Computer Security Principles and Practice

Managing this framework is known as *Identity and Access Management* (IAM)

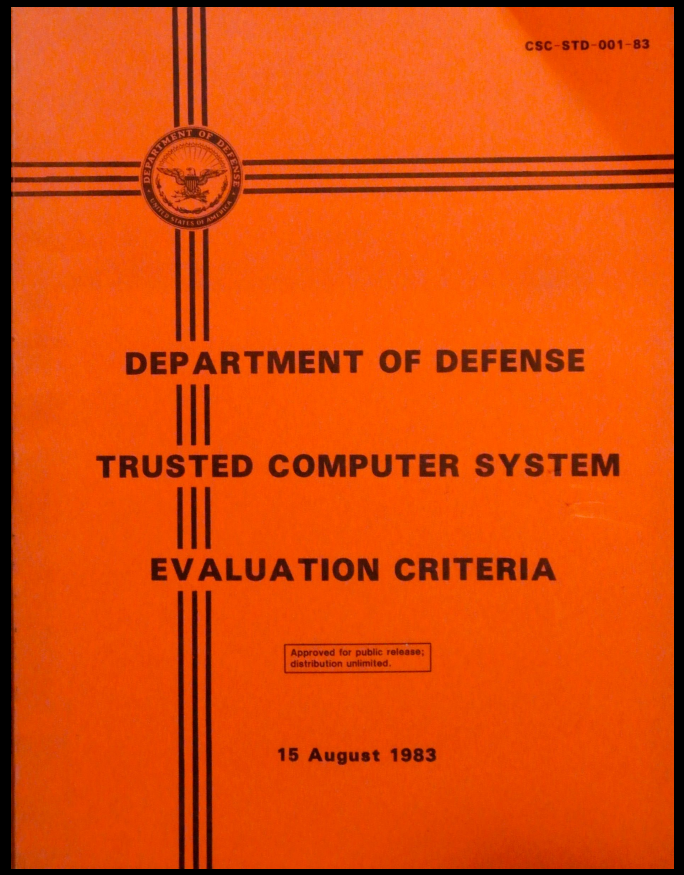
Principle of least privilege

- Only grant the minimal amount of access privileges
  - Minimize the protection domain
  - Each subject has minimal rights
- How is this done practically?
- It is really difficult
  - People want to have all access rights
  - Breaches often occur where too much access has been granted

Trusted Computing Base (TCB)

- The TCB is the set of all hardware, firmware, and software critical to its security
  - Enforces the system security policy

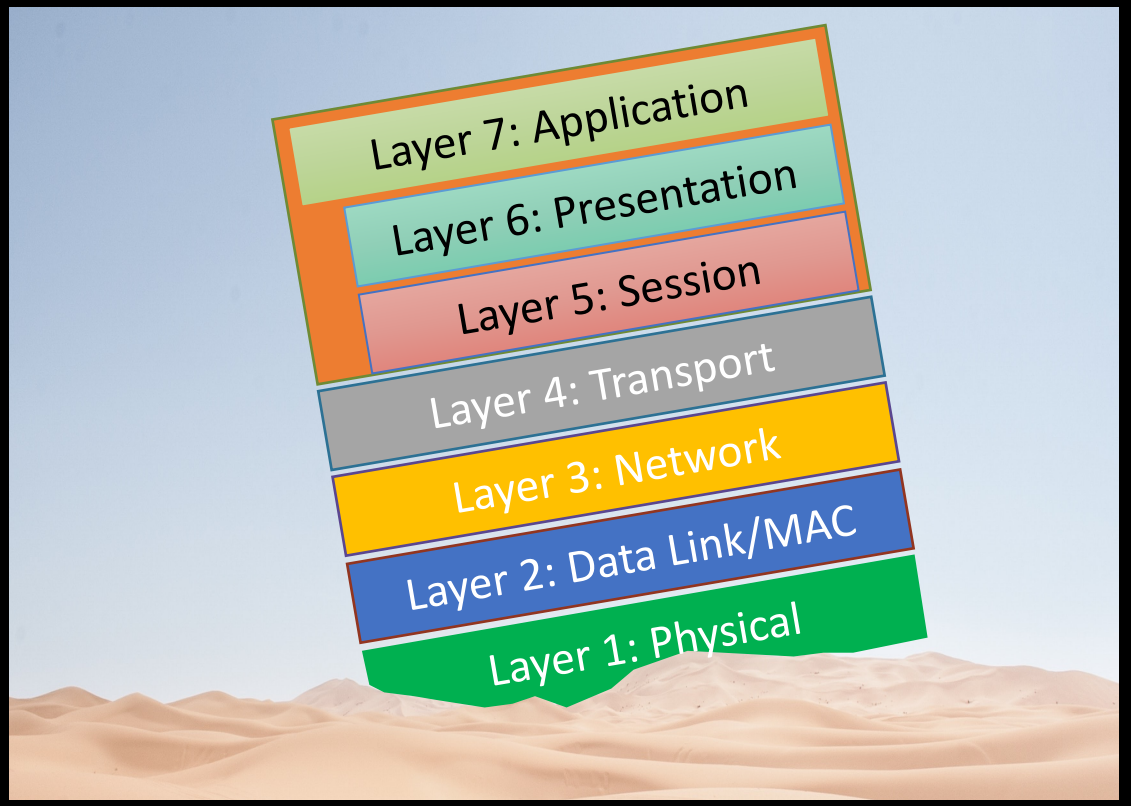
- The TCB should be:
  - Tamperproof
  - Not bypassable
  - Verifiable
  - Simple



Access control across the architecture

- Browsers – Same Origin Policy
- Databases and applications
- Containers
- Virtual Machine isolation
- Operating system
  - Linux, macOS, iOS, Android, Windows
- Processors
  - Intel, Arm
- Networks
  - Segmentation, Virtual Private Networks, Virtual Local Area Networks

- All have different access control policies, and they are usually full of errors!



Access control policy

- Determines what access rights (authorizations) a subject has for a set of objects
- It answers questions like
  - Do you have the right to access the grading program?
  - Do I have the right to change your grades?
  - Do you have the right to access the cyber lab?
- Foundation for enforcement of confidentiality and integrity policies

Access control models

- Mandatory Access Control (MAC) – Based on data sensitivity
- Discretionary Access Control (DAC) – Based on owner decision
- Role-Based Access Control (RBAC) – Based on user role
- Attribute-Based Access Control (ABAC) – Based on user / data attributes
- Expressiveness: ABAC > RBAC > DAC; MAC is completely different
- Lots of approaches
  - Bell-LaPadula – secrecy focused
  - Clark-Wilson – integrity focused
  - SELinux – least privilege
  - Why so many?

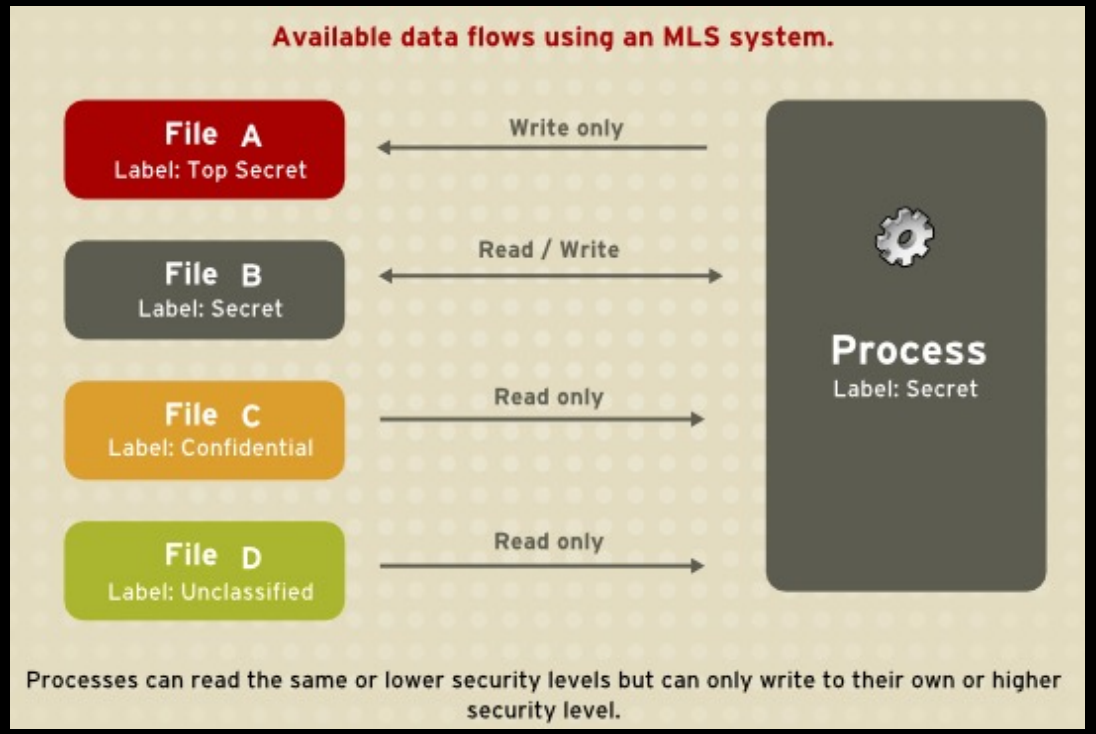
Subjects, objects, and access rights

- Subject – entity capable of accessing object, such as:
  - Owner
  - Group
  - World
- Object – resource to which access is controlled via access rights such as:
  - Read
  - Write
  - Execute
  - Delete
  - Create
  - Search

Mandatory Access Control (MAC)

- Access based on comparing security labels of objects to security clearances of subjects
- Examples: Multilevel security (MLS), hardware ring architecture
- Flexibility problems?

RedHat Security Enhanced Linux (SELinux) Bell-LaPadula model documentation



Discretionary Access Control (DAC)

- Access based on identity of requestor and on access rules stating what requestors are allowed to do

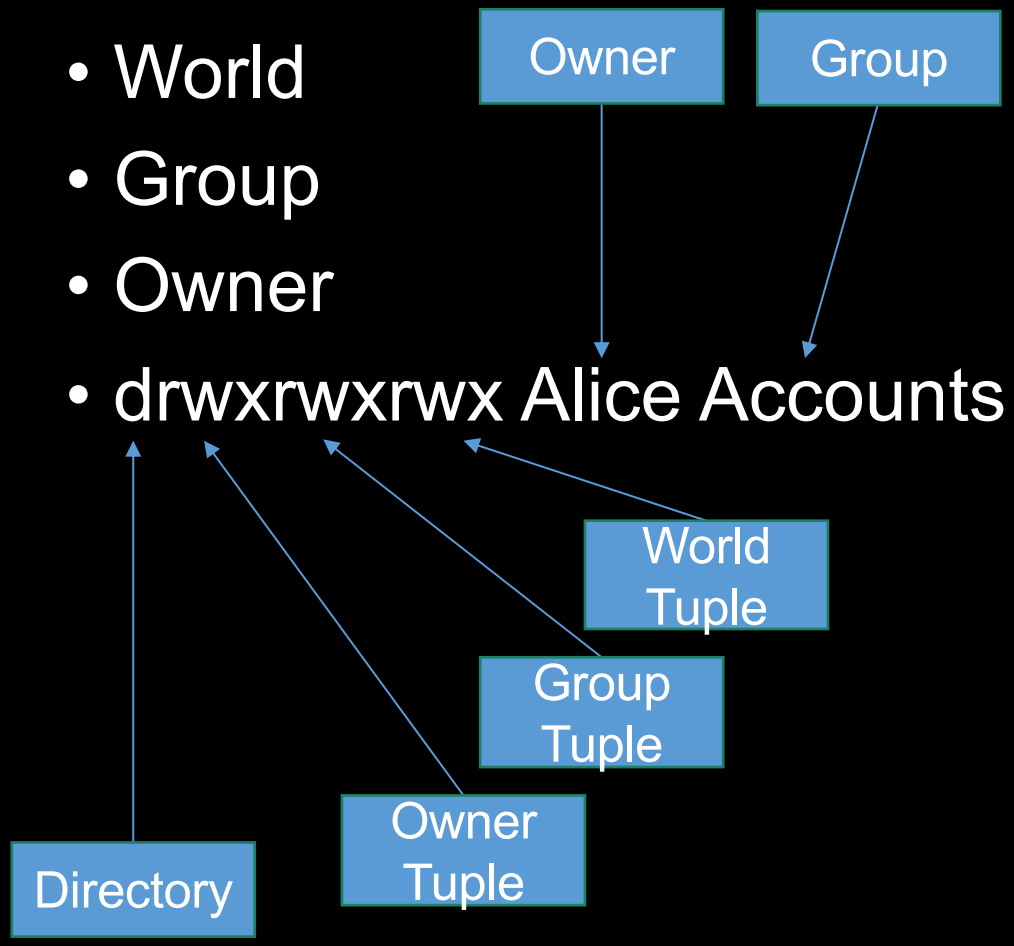
		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Stallings, Computer Security Principles and Practice

Example: Linux access control list (ACL)

- World
- Group
- Owner
- drwxrwxrwx Alice Accounts



Bookkeeping v1 – Discretionary Access Control (DAC)

	Operating System	Accounts Program	Accounting Data	Audit Trail
Sam	rwx	rwx	rw	r
Alice	x	x	rw	-
Bob	rx	r	r	r

- Sam = Sysadmin
- Alice = Manager
- Bob = Auditor

But Sam and Alice shouldn't have write access to accounting data

Bookkeeping v2 – Individual Based DAC

	Operating System	Accounts Program	Accounting Data	Audit Trail
Sam	rwx	rwx	r	r
Alice	x	x	r	-
Accounts Program	rx	r	rw	w
Bob	rx	r	r	r

- How is confidentiality handled?
- How is integrity handled?
- Scaling problems?

Role-Based Access Control (RBAC)

- Access based on roles that users have and on rules stating what accesses are allowed to users in given roles
- Scales when adding new users
- Problems when adding roles?

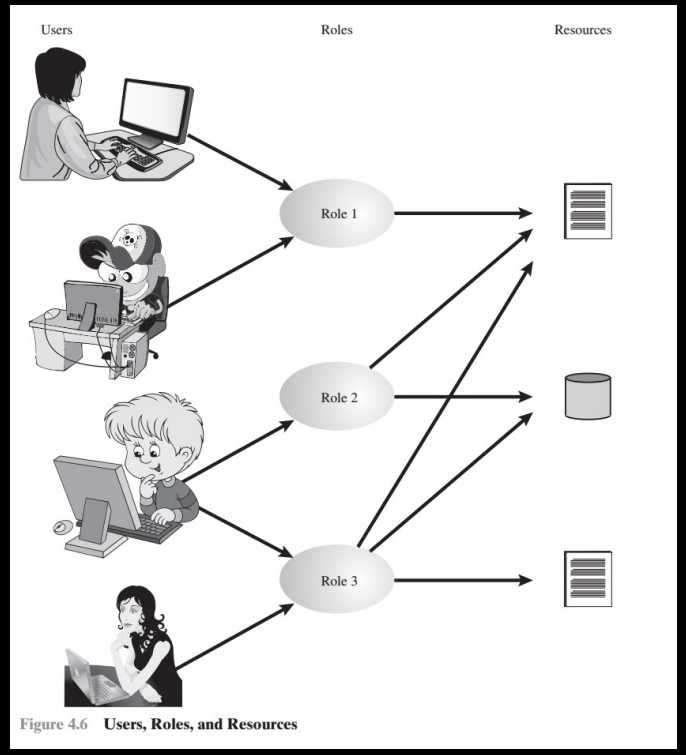


Figure 4.6 **Users, Roles, and Resources**

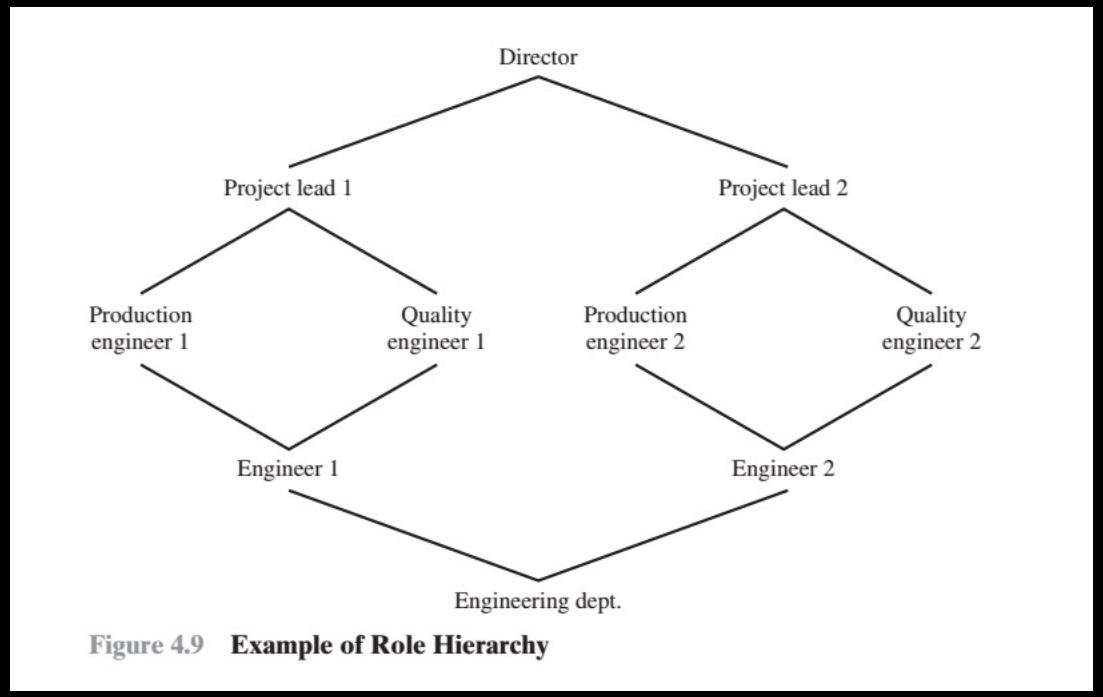


Figure 4.9 **Example of Role Hierarchy**

Bookkeeping v3 – RBAC

	Operating System	Accounts Program	Accounting Data	Audit Trail
Head sysadmin	rwx	rwx	r	r
Sys-admin	rwx	r	-	r
Manager	x	rx	r	-
Accounts Program	rx	r	rw	w
Auditor	rx	r	r	r
Teller	x	x	-	-

Attribute-Based Access Control (ABAC)

- Access based on attributes of user, resource to be accessed, and current environmental conditions
- Works well for attribute-value pairs
- New attributes and values may be added on the fly
- Can define complex access policies computed dynamically
  - For example, attending physician may access a patient's records
- Performance problems?

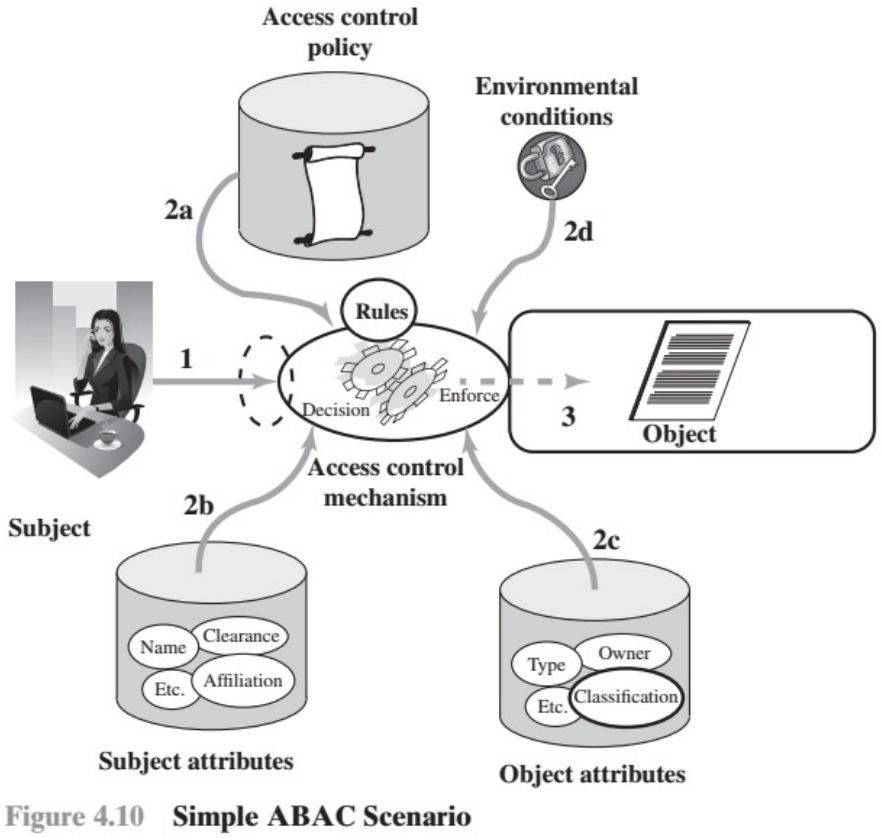


Figure 4.10 Simple ABAC Scenario

Stallings, Computer Security Principles and Practice

Simple ABAC example

User	Role
John	Adult
Mary	Juvenile
Sue	Child

Role	Movie Permissions
Adult	view R, view PG-13, view G
Juvenile	view PG-13, view, view G
Child	view G

Movie Rating	Users Allowed Access
R	Age 17 and older
PG-13	Age 13 and older
G	Everyone

```
R1:can_access(u, m, e) ←
    (Age(u) ≥ 17 ∧ Rating(m) ∈ {R, PG-13, G}) ∨
    (Age(u) ≥ 13 ∧ Age(u) < 17 ∧ Rating(m) ∈ {PG-13, G}) ∨
    (Age(u) < 13 ∧ Rating(m) ∈ {G})
```

#enterprise-group-exercises

For your enterprise group, select an access control policy for your users, describe how it will be used, and justify why it's the best choice. Provide your summary in the #enterprise-group-exercises slack channel

1. Healthcare
2. Government
3. Retail
4. Banking
5. Utility
6. Information Technology
7. University
8. Manufacturing

What we discussed

- Capital One breach
- Principle of least privilege and TCB
- Access control across the architecture
- Access control models
  - MAC
  - DAC
  - RBAC
  - ABAC
- Expressiveness: ABAC > RBAC > DAC; MAC is completely different

What's next

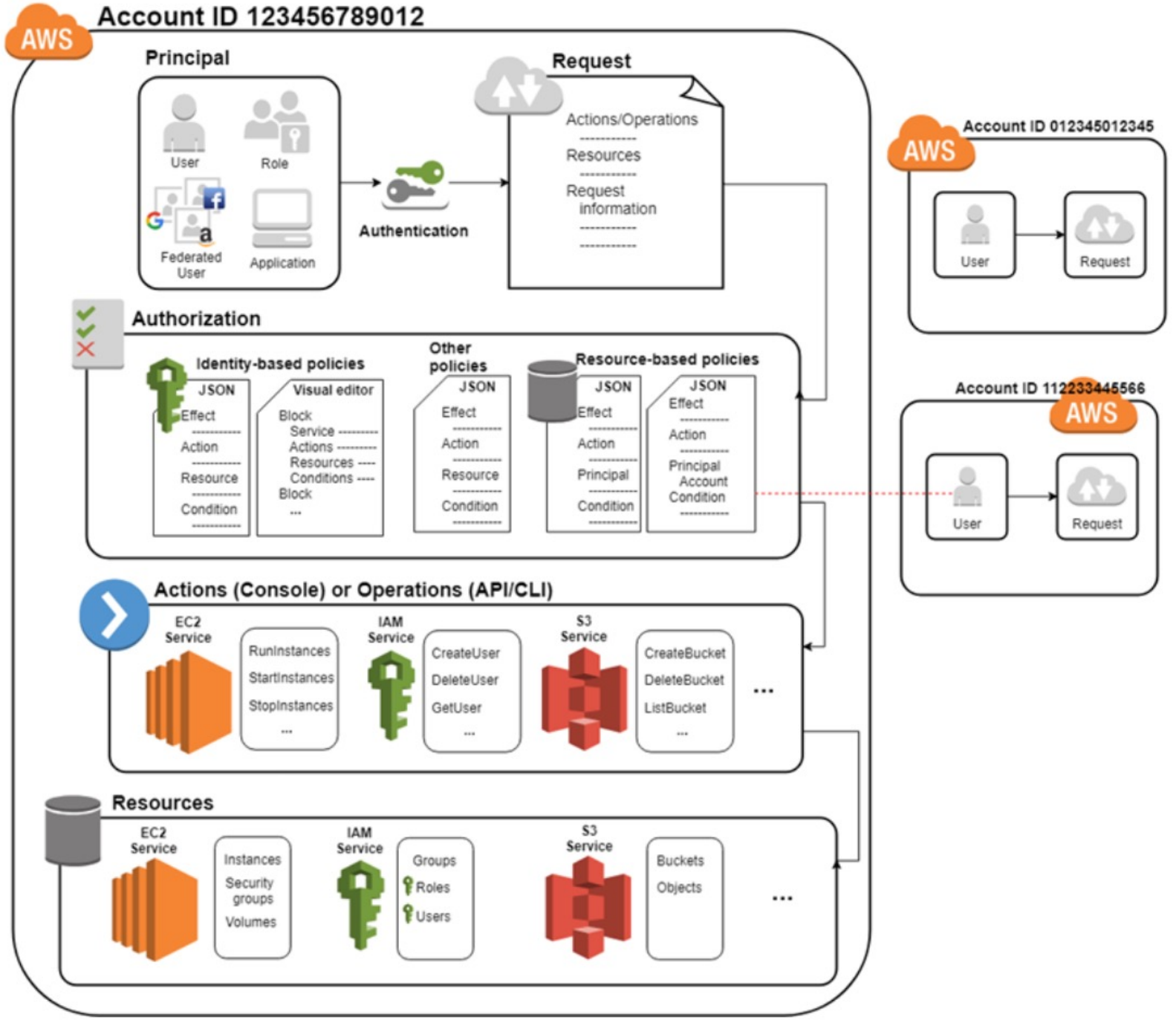
- Module 5 Access Control (continued)
- Mastery extension project overview
- Readings
  - Anderson, Chapter 4, especially 4.2 Operating System Access Controls,
  - Access Control Models: Review of Types and Use-Cases
  - What is Amazon Web Services Identity and Access Management?
    - How IAM works
    - Users in AWS
    - Permissions and Policies in IAM
    - What is ABAC?
- You should be working on lab 5 Access Control due next Tuesday
- #breach-of-the-week – participate on slack!
- Office hours Thurs 11:00am-12:00pm in 192-333 or M/W/F on zoom

Breach of the Week!

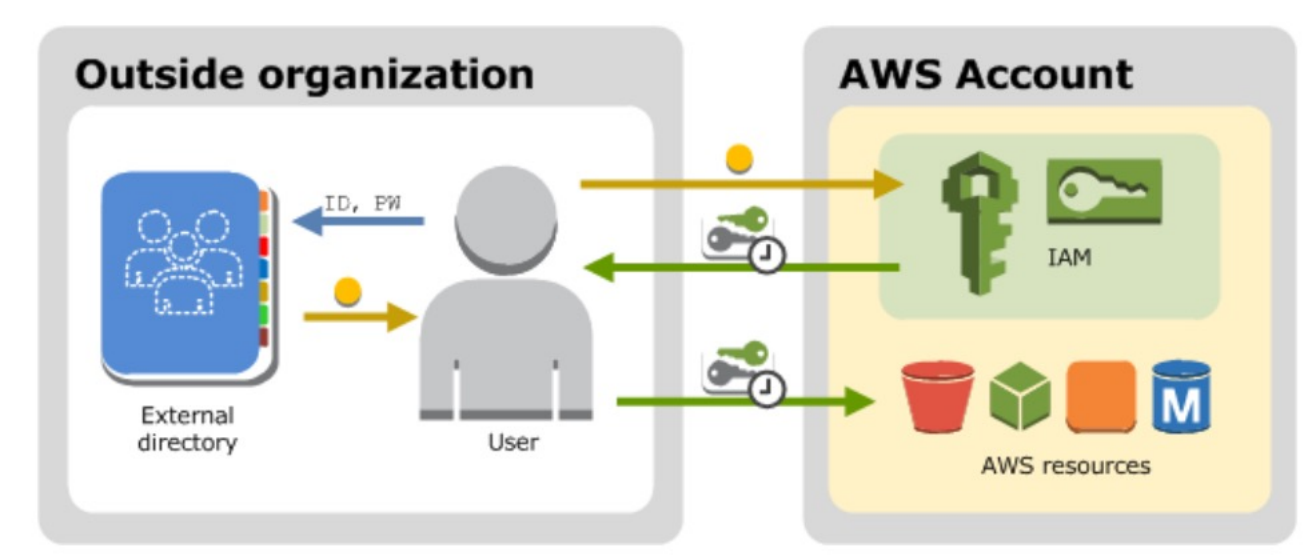
Example: Amazon Web Services Identity and Access Management

- Shared access to AWS account
- Granular permissions – DAC, RBAC, ABAC
- Secure access to AWS resources for applications that run on Amazon EC2
- Multi-factor authentication (MFA)
- Identity federation
- Identity information for assurance
- Eventually Consistent
- All in a convenient 1000 page user guide 

How IAM works



Users in AWS



- Identities can be users, groups, roles, or applications
- Security Assertion Markup Language (SAML 2.0) single sign-on (SSO)
- OpenID Connect (OIDC) compatible identity provider

Security Assertion Markup Language (SAML)

- XML-based exchange of authentication and authorization information between security domains
- Contains
  - Assertions about a principal (subject)
  - Identity provider
  - Digital signature to verify source of assertions
- User for cross-domain single-sign on (SSO)
  - Distributes tokens to multiple domains
  - For example, login to Calpoly then access Canvas

SAML 2.0 example

Digital signature

Subject name

Subject confirmation methods

- Bearer
- Holder-of-key
- Sender-vouches

```
<saml:Assertion
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ID="_d71a3a8e9fcc45c9e9d248ef7049393fc8f04e5f75"
   Version="2.0"
   IssueInstant="2004-12-05T09:22:05Z">
   <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
   <ds:Signature
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
   <saml:Subject>
     <saml:NameID
       Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
       3f7b3dcf-1674-4ecd-92c8-1544f346baf8
     </saml:NameID>
     <saml:SubjectConfirmation
      Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
       <saml:SubjectConfirmationData
         InResponseTo="aaf23196-1773-2113-474a-fe114412ab72"
         Recipient="https://sp.example.com/SAML2/SSO/POST"
         NotOnOrAfter="2004-12-05T09:27:05Z"/>
     </saml:SubjectConfirmation>
   </saml:Subject>
```

Wikipedia

SAML 2.0 example (continued)

Validity

Context

Attribute name

Attribute values

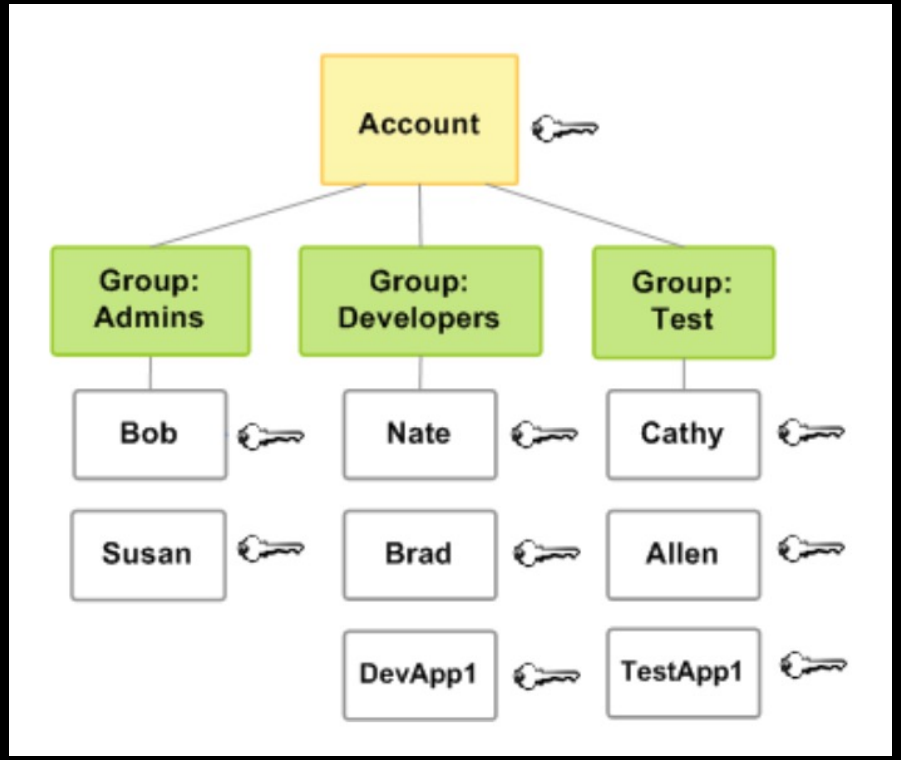
```
<saml:Conditions
  NotBefore="2004-12-05T09:17:05Z"
  NotOnOrAfter="2004-12-05T09:27:05Z">
  <saml:AudienceRestriction>
    <saml:Audience>https://sp.example.com/SAML2</saml:Audience>
  </saml:AudienceRestriction>
</saml:Conditions>
<saml:AuthnStatement
  AuthnInstant="2004-12-05T09:22:00Z"
  SessionIndex="b07b804c-7c29-ea16-7300-4f3d6f7928ac">
  <saml:AuthnContext>
    <saml:AuthnContextClassRef>
      urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
    </saml:AuthnContextClassRef>
  </saml:AuthnContext>
</saml:AuthnStatement>
<saml:AttributeStatement>
  <saml:Attribute
    xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
    x500:Encoding="LDAP"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
    Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
    FriendlyName="eduPersonAffiliation">
    <saml:AttributeValue
      xsi:type="xs:string">member</saml:AttributeValue>
    <saml:AttributeValue
      xsi:type="xs:string">staff</saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
```

Wikipedia

Permissions and policies in IAM

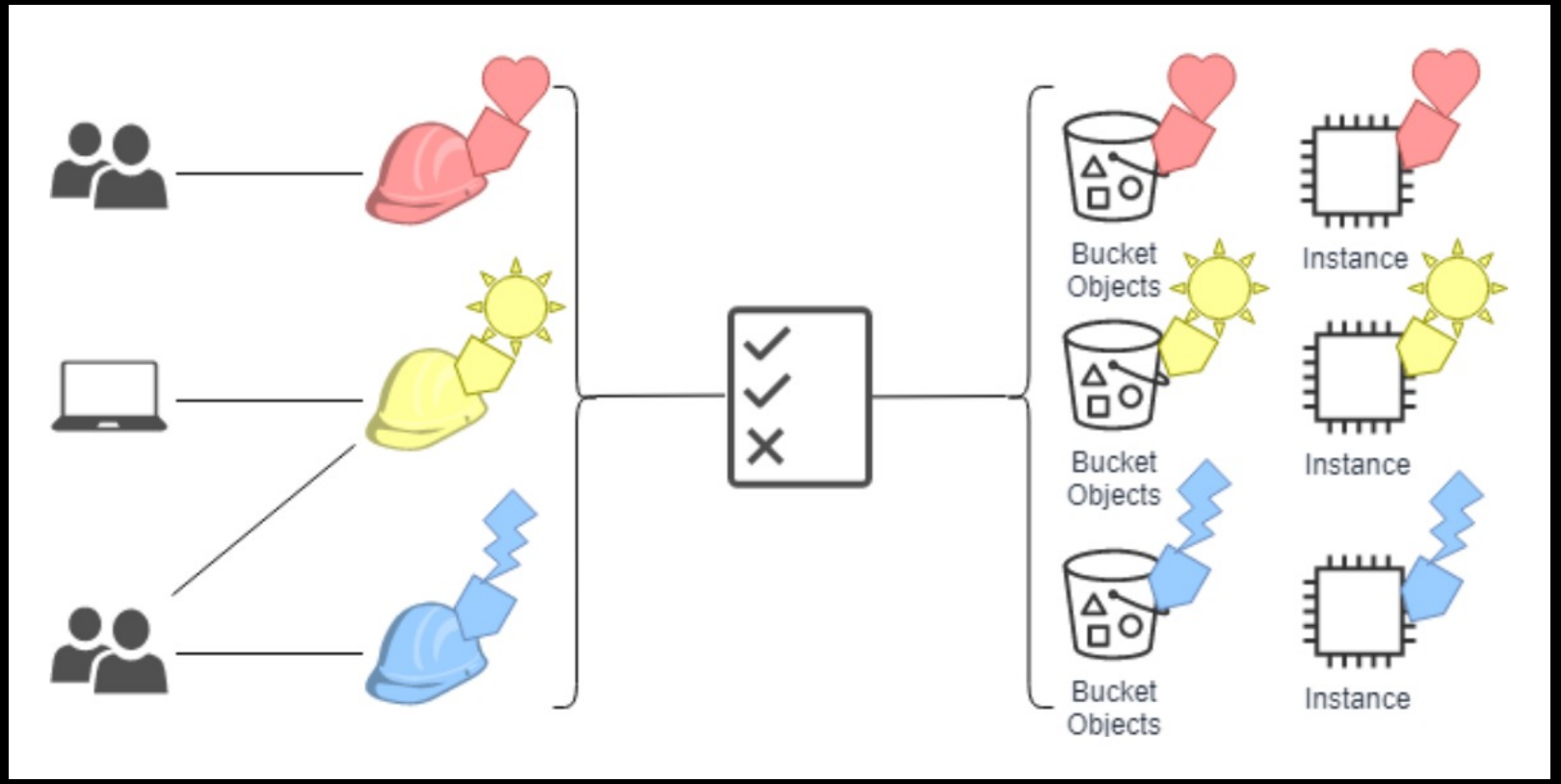
```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "dynamodb:*",
    "Resource": "arn:aws:dynamodb:us-east-2:123456789012:table/Books"
  }
}
```

- Policies can be attached to identities or resources



AWS Attributed-Based Access Control (ABAC)

A tag is a custom attribute label that can be assigned to identities and resources



For example, you can create three roles with the `access-project` tag key. Set the tag value of the first role to `Heart`, the second to `Sun`, and the third to `Lightning`. You can then use a single policy that allows access when the role and the resource are tagged with the same value for `access-project`. For a detailed tutorial that demonstrates how to use

AWS ABAC advantages vs traditional RBAC

- ABAC permissions scale
  - Unnecessary for an administrator to update existing policies to allow access to new resources
- ABAC requires fewer policies
  - Because you don't have to create different policies for different job functions, you create fewer policies
- Using ABAC, teams can change and grow quickly
  - Permissions for new resources are automatically granted based on attributes
- Granular permissions are possible using ABAC
  - You can allow actions on all resources, but only if the resource tag matches the principal's tag
- Use employee attributes from a corporate directory with ABAC
  - Configure your SAML-based or web identity provider to pass session tags to AWS.

Services that work with AWS IAM

Examples of AWS Services

Cryptography and PKI services

Service	Actions	Resource-level permission	Resource-based policies	ABAC	Temporary credentials	Service-linked roles
AWS Certificate Manager (ACM)	Yes	Yes	No	Yes	Yes	Yes
AWS Private Certificate Authority (AWS Private CA)	Yes	Yes	Yes	Yes	Yes	No
AWS CloudHSM	Yes	Yes	No	Yes	Yes	Yes
AWS Key Management Service (AWS KMS)	Yes	Yes	Yes	Yes	Yes	Yes
AWS Signer	Yes	Yes	No	Yes	Yes	No

Access control observations

Access control policy mistakes are everywhere:

- Too many
  - Policies
  - Layers
  - Privileges
  - Errors
  - Inconsistencies
- Resulting in root cause of many security breaches

Access control solutions

- Standards
  - eXtensible Access Control Markup Language (XACML)
  - Open Policy Language (OPA)
- Policy analysis tools
  - Look for errors and inconsistencies
- Run-time monitoring
  - User and Entity Behavior Analytics (UEBA)

All help, but none satisfactory

Expect access control errors to continue!

What we discussed

- AWS IAM
  - How IAM works
  - Users in AWS
  - SAML
  - Permissions and policies
  - ABAC
- Access control observations
- Access control solutions

What's next

- Module 6 Malicious Software
- Readings
  - Anderson Chapter 4 on Access Control (especially 4.4 What goes wrong)
  - Smashing the Stack
  - Format String Vulnerabilities
  - Polymorphic malware
  - Optional: The Geometry of Innocent Flesh on the Bone: Return-into-libc without Function Calls (on the x86)
  - Optional: ROP
  - Optional: Malware
- Quiz 5, Lab 5 Access Control due on Tuesday
- #breach-of-the-week – participate on slack!
- Office hours Thurs 11:00am-12:00pm in 192-333 or M/W/F on zoom