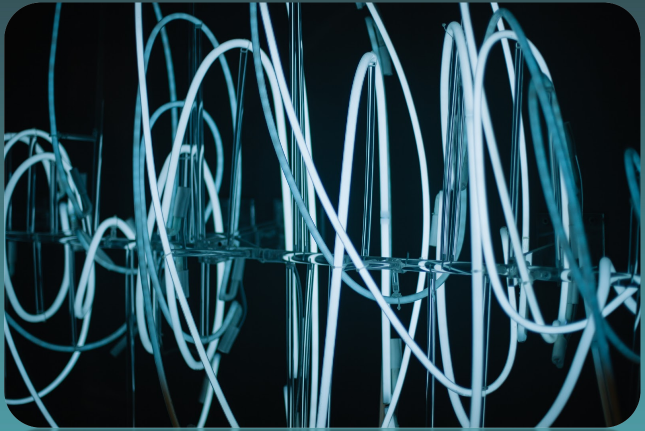
AUTHENTICATION VS. AUTHORIZATION

What's the Difference?

Confusion between the two terms

Authentication

- Asserting Identity
- Proving Identity



Authorization

- Process of allowing access
- Predicated upon proven identity
- Depends upon authentication
- If you don’t authenticate, you cannot authorize

WHAT IS AUTHENTICATION?

Authentication – What is it?

Authentication is the process of asserting what your identity is and then proving your identity.

Example in real life:

- You hear a knock at the door, you ask "who is it?"
- Person responds with the name of your old college friend, but the voice doesn't sound right
- You look through the peephole to ensure it really is your college friend
- You find out it is, but they have a cold, hence the reason their voice doesn't sound right

Authentication – What is it?

What happened here?

- When knocked, they *asserted* identity
- They *provided proof* as well – the sound of their voice
- Failed to prove identity of their voice because of the cold
- You sought other proof by looking through peephole
- You saw your friend, you let them in

In applications, when we have to authenticate, we assert our identity with a user id /username and supply a password

The 4 Factors Of Authentication

Something You Know

- Stipulates a unique piece of information that only one exact user should know
- Knowledge of a password, passcode, or answer to a secret question
- Previous account activity that only a specific user would know (bank transactions, emails, etc.)

Something You Are

- Biometric data that is unique to a single individual
- Can including iris scans, palm prints, fingerprints, etc
- Cannot be changed

Something You Have

- Ownership of a hardware token or device
- Often has a key tied to certain systems or apps
- Token can be used as a form of positive identification

Somewhere You Are

- Can be a contributing factor, not a stand-alone factor like the above
- May use something you know in conjunction with somewhere you are to have stronger authentication
- When authenticating a system, the geographic location of the user (based on IP address) is used to determine whether the login attempt (based on past login attempt locations) is authentic or fraudulent

WHAT IS AUTHORIZATION

Authorization – What is it?

Authorization is the process of allowing access, predicated upon your proven identity. Depends on authentication.

Banking App Example of Authorization

- Authenticate by providing your username and password
- By knowing those, you have asserted your identification and proved it
- App allows you in (you are authenticated)
- You are only *authorized* to see your own data from your authentication – not that of anyone else, someone set those parameters for you

If you don’t authenticate, you cannot really authorize

Horizontal Authorization vs. Vertical Authorization

Horizontal Authorization

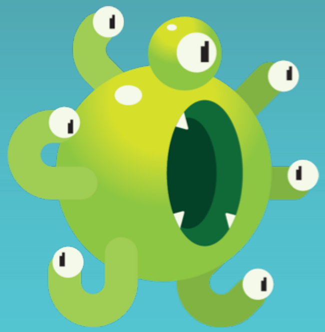
- Dictates what *access to data a user has*. Banking example: I can see my data, but not my neighbors. My banker is authorized to see both of our data.

Vertical Authorization

- Dictates what a *user can do*. Banking example: While I’m allowed to view my banking data, I cannot create a new business account. I need to go to my banker, provide information, and they open the account. My banker has privileged access, more authorization and is allowed to open an account in my name.

Authorization by Role Structure

- Somewhat older system
- Authorization by role structure is a protocol where authorization is determined based on roles in an organization
- Profiles are made and set with default privileges (based on roles), giving the intended user a strict set of restrictions and limitations in a system
- Convenient, in that it requires less upfront coding to be implemented
- Also known as Role-Based Access Control (RBAC)



Authorization by Permission Based Structure

- While RBAC is a strict system based on the user’s role, permission-based authorization system allows more fine-tuned control over access to data
- Users are given roles with different limitations (privileges/permissions) than with RBAC systems
- Users are directly given a myriad of specific permissions to access certain data and/or carry out certain actions
- Requires more time for planning, coding and implementation

Authorization by Permission Based Structure

Example: Consider a bank clerk

The bank clerk’s manager is going out of town and she needs someone to open the safe on Monday. The normal “Bank Clerk” role does not have authorization to do this. You must be a “Branch Manager” to open the safe.

The branch manager can grant the role of “branch manager” to the clerk in order to open the safe, based on the RBAC structure. However, when this happens, the bank clerk can also see all her colleagues salaries, discipline history and other access the manager has, that the clerk *should not* have access to. This is not ideal.

If they had a fine grained permission based authorization model, the branch manager could grant the clerk permission to only open the safe, and drill it down to that specific day!

In Summary

Authentication shows the system who you are

Authorization shows the application what the authenticated user is allowed to do

Different ways to do both, but they are very different from each other

About Cypress Data Defense:

Our goal is to help organizations secure their IT development and operations using a pragmatic, risk-based approach. The diverse background of our founders allows us to apply security controls to governance, networks, and applications across the enterprise.

Contact us to learn more!

https://www.cypressdatadefense.com/contact/

Email: info@cypressdatadefense.com

Phone Number: 720.588.8133

CYPRESS
DATA DEFENSE