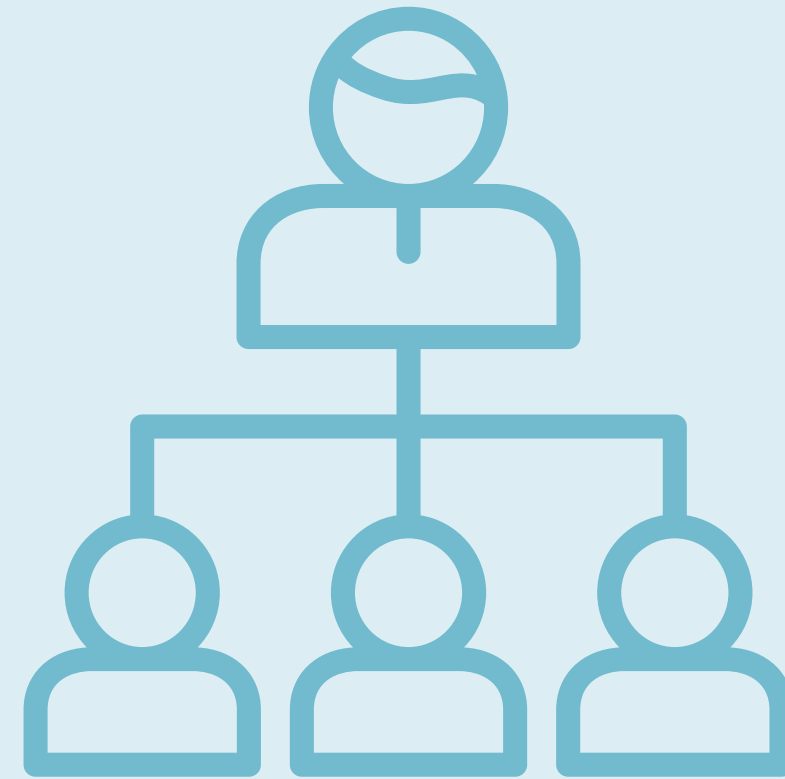


ATTACKING ACCESS CONTROL MODELS IN MODERN WEB APPLICATIONS



AGENDA OF THE TALK

01 Introduction to Access Control Models

02 Common Mistakes Found in Access Control models

03 Our Methodology

04 Right Mindset and Right Toolset


05 Our statistics

06 Q/A





\$WHOAMI

- Imran parray
- Founder |  Snapsec
- Independent Cybersecurity Researcher
- Bug Bounty hunter
- Spends a lot of time writing bash, Python, Automation, and tons of articles on snapsec.co/blog

WHAT ARE ACCESS CONTROL MODELS.

Access control models are used to control the access of users to resources and how they interact with them. Nobody in an organization should have free rein to access any resource. Access control is the combination of policies and technologies that decide which authenticated users may access which resources.

Data/Services/Endpoints





TYPES OF ACCESS CONTROL MODELS

- 01 **Mandatory Access Control**
- 02 **Discretionary Access Control**
- 03 **Role-Based Access Control**
- 04 **Privileged Access Management**

ACCESS CONTROLS ARE EVERYWHERE.

	Owner/admin	Member	Guest
Send messages and upload files	✓	✓	✓
Join any public channel	✓	✓	
Delete your own messages	✓	✓	✓
Create a channel*	✓	✓	
Create a private channel*	✓	✓	Multi-channel guest
Convert a channel to private**	✓		
Manage channels with Slack Connect†	✓	■	
Archive a channel**	✓	✓	
Rename a channel**	✓		
Delete a channel	✓		
Set channel retention	★		
Set private channel retention	✓	■	
Set posting permissions	✓	✓	

Slack

Permissions Summary

The following permissions can be assigned in a space:

Category	Permission
All	<p>View gives you permission to access the content in this space, and see it in the space directory and other places like the dashboard.</p> <p>Delete own gives you permission to delete any pages, blogs, attachments and comments you've created in this space (regardless of whether other users have subsequently edited the content).</p>
Pages	<p>Add page gives you permission to create new pages and edit existing pages in this space (assuming the page is not <i>restricted</i> for editing).</p> <p>Delete page gives you permission to delete any page in the space.</p>
Blog	<p>Add blog gives you permission to create new blog posts and edit existing blog posts in this space (assuming the blog post is not restricted for editing).</p> <p>Delete page gives you permission to delete any blog post in the space. Delete permission is also required to <i>move</i> a page or blog to a different space.</p>
Attachments	<p>Add attachment gives you permission to upload (attach) <i>files</i> to pages and blog posts in this space, and to edit attached files using the Companion app.</p>

Confluence Products

▼ 👤 SHARED

Data analyst	0
Designer	0
Editor	0
Copywriter	0
Admin	0

Typeform

Basic permissions

Basic permissions are lower risk and can be enabled by develop

Permission Name	Fields	API
Obtain basic user information	name, avatar, description	Batch obtain user information
Obtain department information	id, name, chat_id, status	Obtain department details , Batch obtain department details

Larksuite

Access Control

Read...	To learn...
Role-Based Access Control	About the concept of role-based access control and how it applies in Auth0.
Authorization Policies	About the concept of authorization policies and how they apply in Auth0.
Rules for Authorization Policies	How rules apply to authorization policies and Auth0's role-based access control (RBAC) system.
Sample Use Cases: Role-Based Access Control	How to implement roles-based authorization (RBAC) in different scenarios and explore how to use rules with RBAC.
Sample Use Cases: Actions with Authorization	How to use actions with roles-based access control (RBAC). For use with our Authorization Core feature set.
Sample Use Cases: Rules with Authorization	How to use rules with roles-based access control (RBAC). For use with our Authorization Core feature set.
Authorization Core vs. Authorization Extension	About the differences between Auth0's core RBAC release and the Authorization Extension.
Configure Core RBAC	How to configure Auth0 Core Authorization features for role-based access control (RBAC) of your APIs.

Auth0

User roles permissions matrix

Below is a table of the the permissions each user may have access to based on their role:

	Owner	Administrator	Developer	Billing Manager	Support
API Credentials	x	x	x		
Billing History & Settings	x	x		x	
Dev Tools	x	x	x		
IoT: Console Access	x	x	x		x
IoT: Manage SIMs, Fleets &	x	x	x		

Twilio

4 TYPES OF MISTAKES

1

**PERMISSIONS
AREN'T
IMPLEMENTED
PROPERLY**

2

**PERMISSION X
OVERRIDE
PERMISSION Y**

3

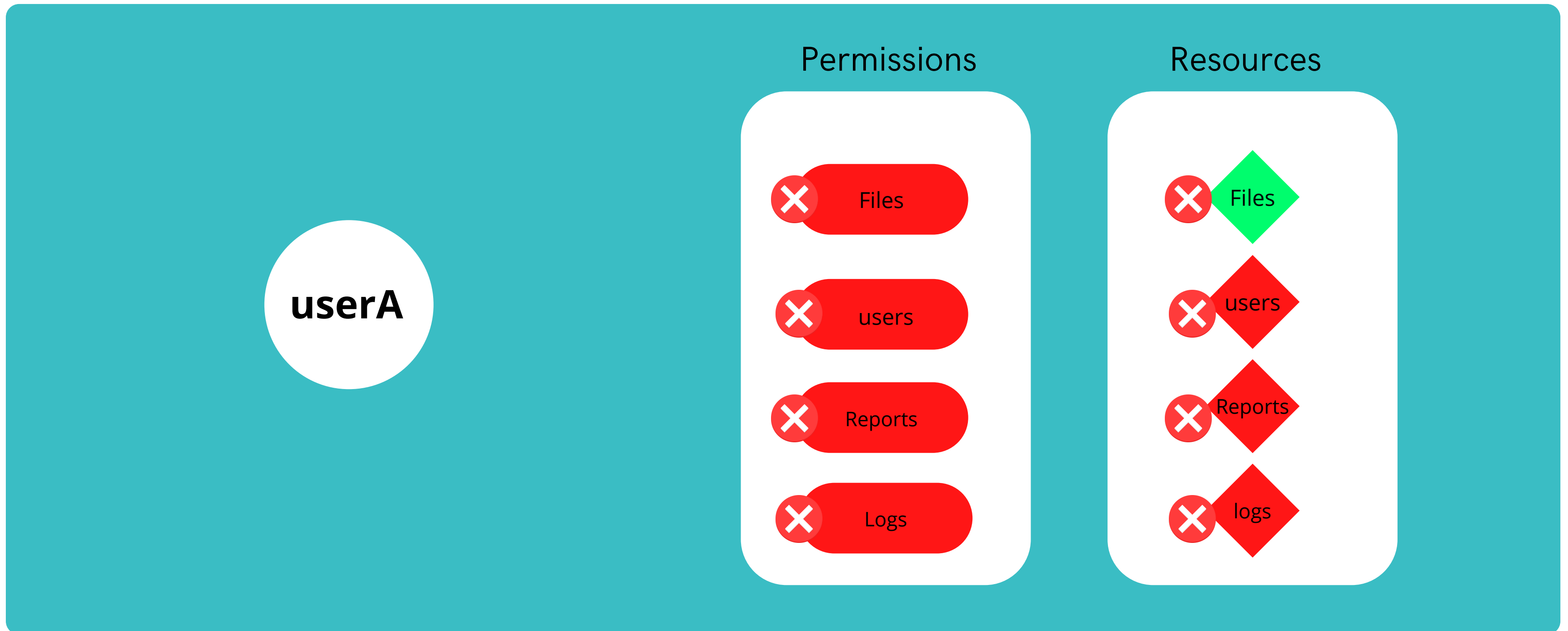
**SET OF
PERMISSION
OVERRIDE
PERMISSION X**

4

DESIGN FLAWS

1 PERMISSIONS AREN'T IMPLEMENTED PROPERLY

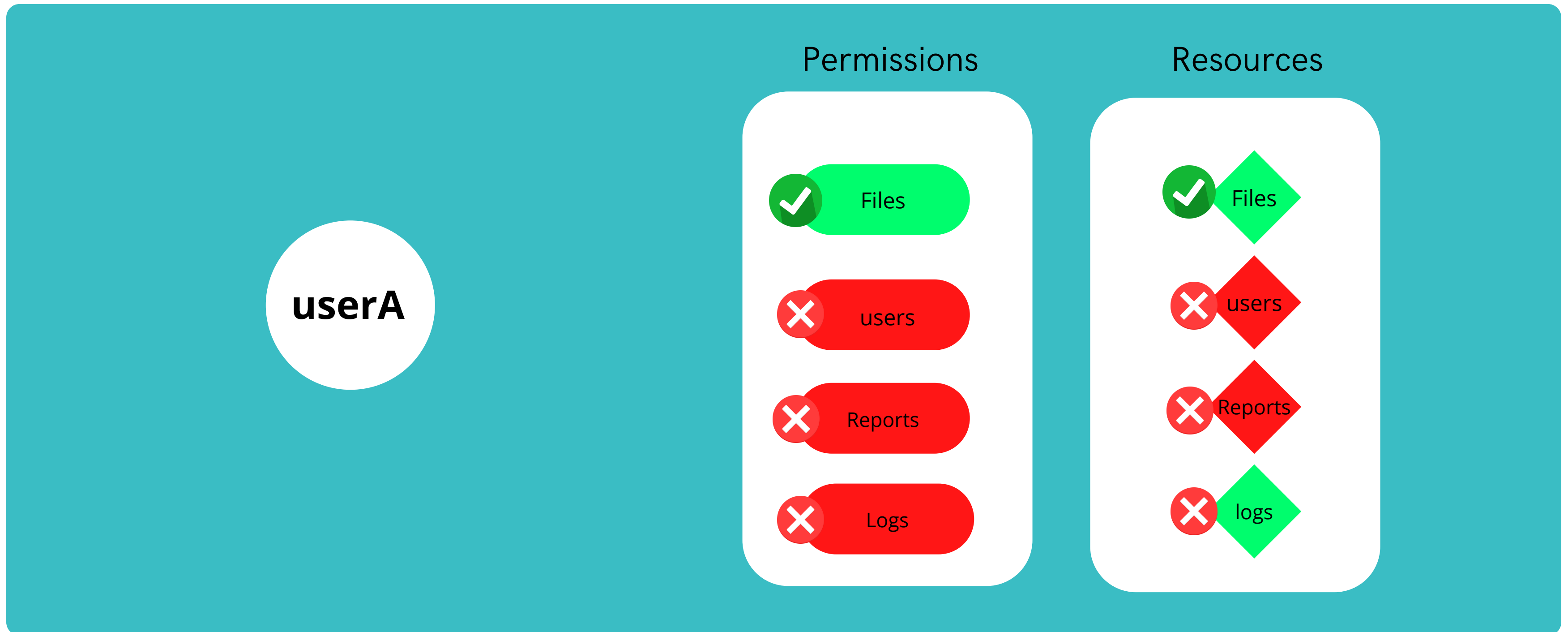
user can access /api/files without file permission



2

PERMISSION X OVERRIDE PERMISSION Y

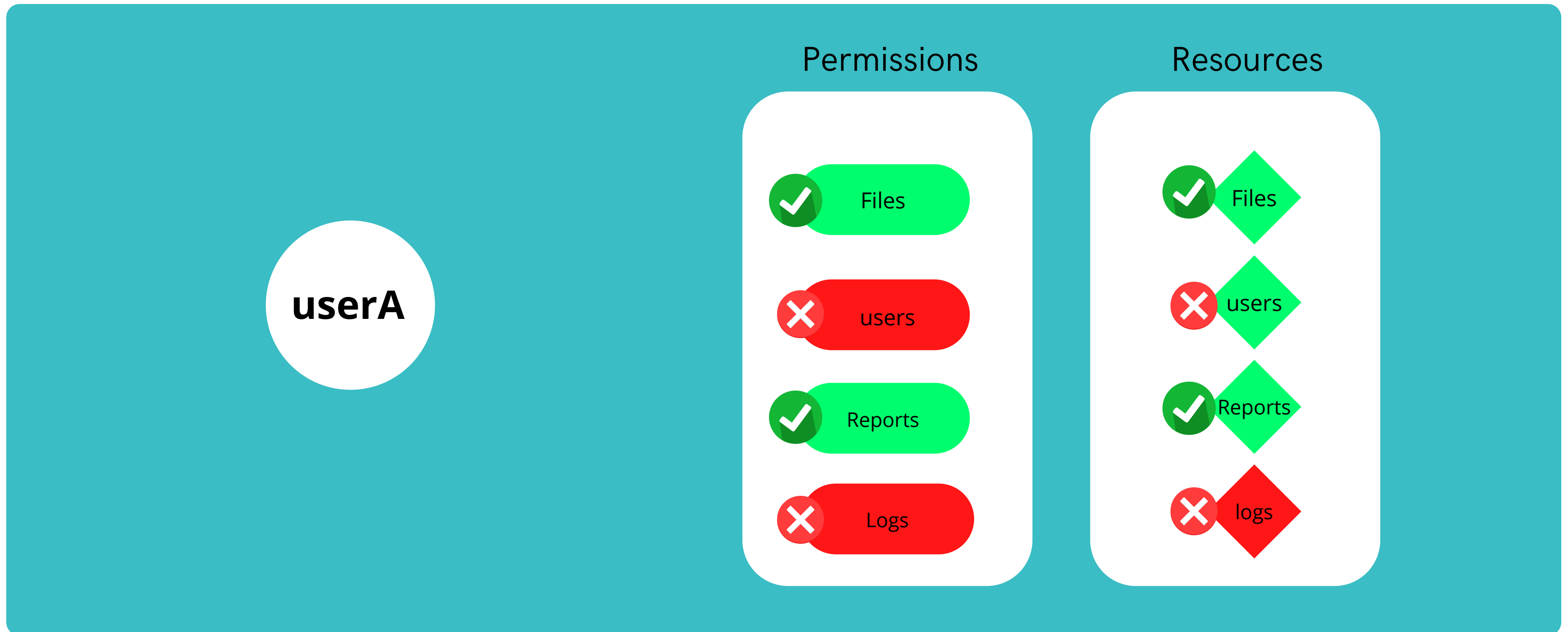
when the userA is give Files permission he can access logs as well



3

SET OF PERMISSION OVERRIDE PERMISSION X

when user is given Files and Reports permission he can access users as well



DESIGN FLAWS - LEADS ACCESS CONTROL ISSUES

Permissions



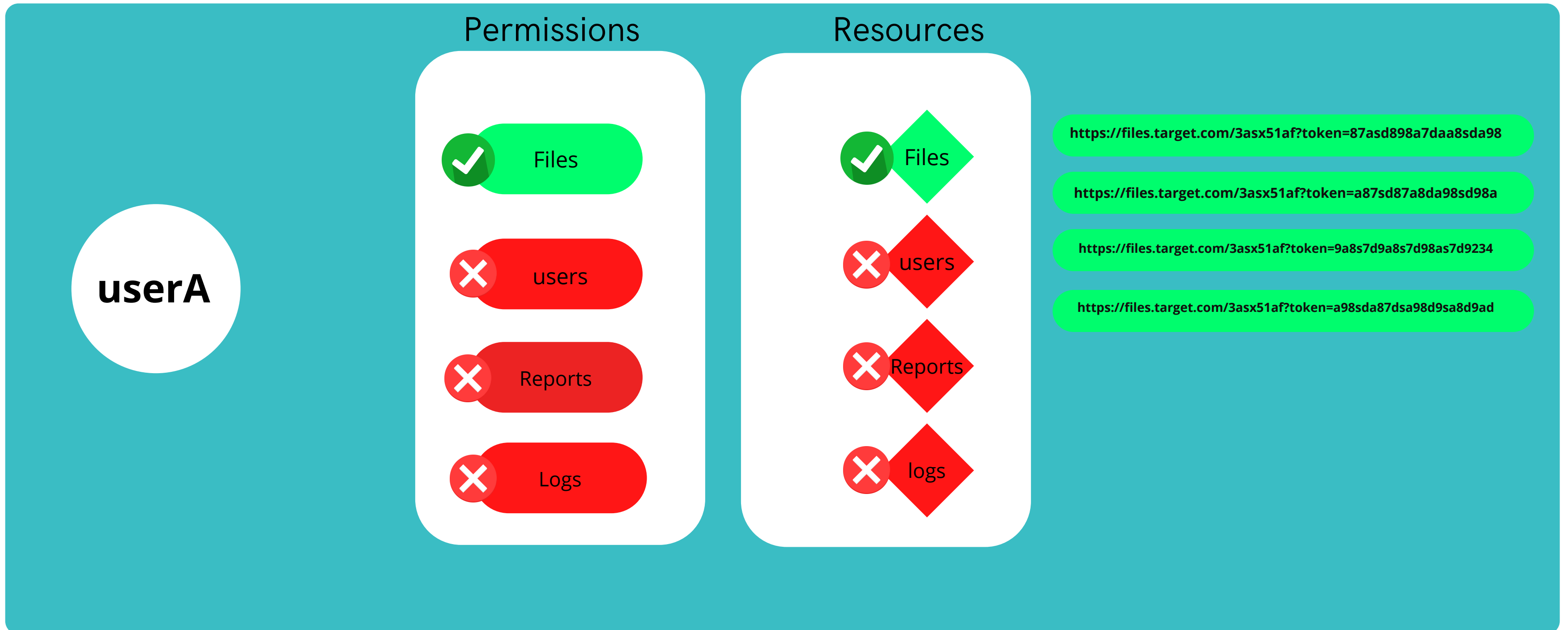
Resources



4

DESIGN FLAWS - LEADS ACCESS CONTROL ISSUES

Example:



1

DESIGN FLAWS - LEADS ACCESS CONTROL ISSUES

Example:

<https://files.target.com/3asx51af?token=a87sd87a8da98sd98a>

<https://files.target.com/3asx51af?token=9a8s7d9a8s7d98as7d9234>

<https://files.target.com/3asx51af?token=a98sda87dsa98d9sa8d9ad>

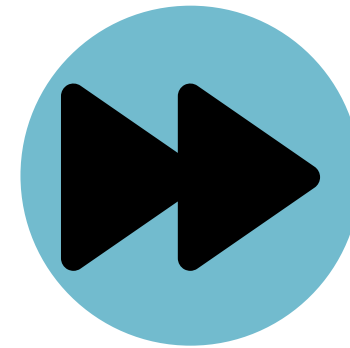
userA

<https://files.target.com/3asx51af?token=87asd898a7daa8sda98>

a9s7d9a8sd98as09d8sa09d809sa
8d0a9s8d09as8d09sa8d9sa80d98
sa09d8a09sd8as09d8as098d0s98
d0as98d9as8d098sa0d98as09d80
asd97ad9hdas9h832hr83fh87gf87
gf843gf8gf8a9s7d9a8sd98as09d8s
a09d809sa8d0a9s8d09as8d09sa8
d9sa80d98sa09d8a09sd8as09d8a
s098d0s98d0as98d9as8d098sa0d
98as09d80asd97ad9hdas9h832hr
83fh87gf87gf843gf8gf8a9s7d9a8s
d98as09sasdadasf87gf843gf8gf8

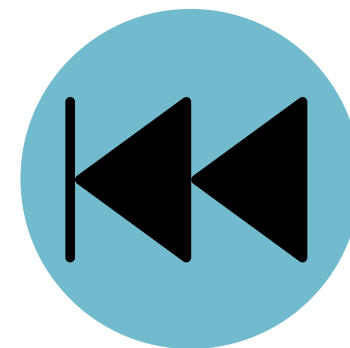
Updated content of the resource

OUR METHODOLOGY



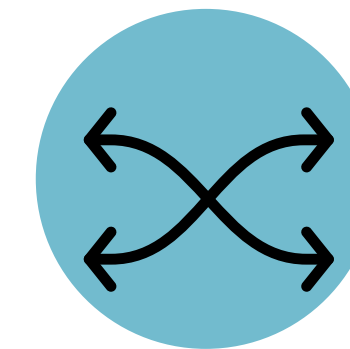
FORWARD APPROACH

Give one permission and target all other permission



BACKWARD APPROACH

Give all permissions and target one permission



MIXED APPROACH

Give some and target some



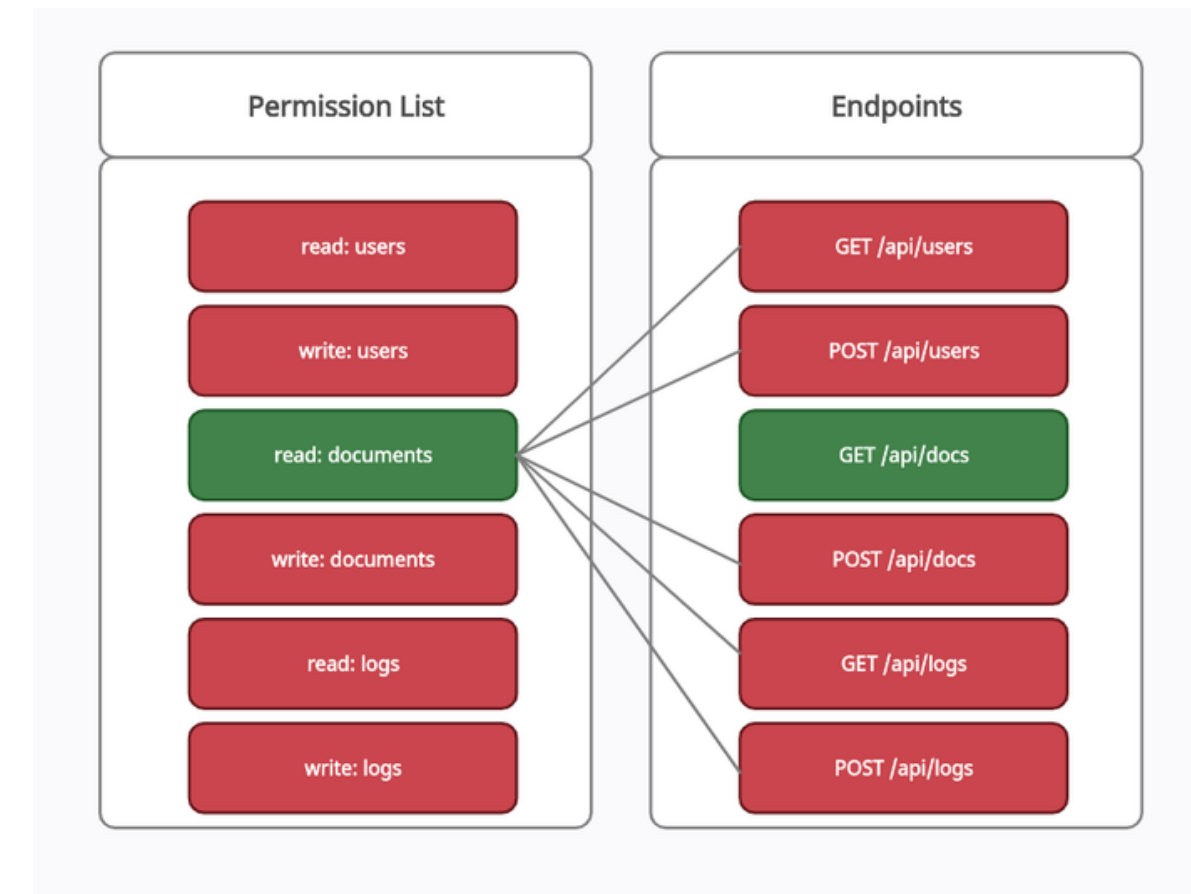
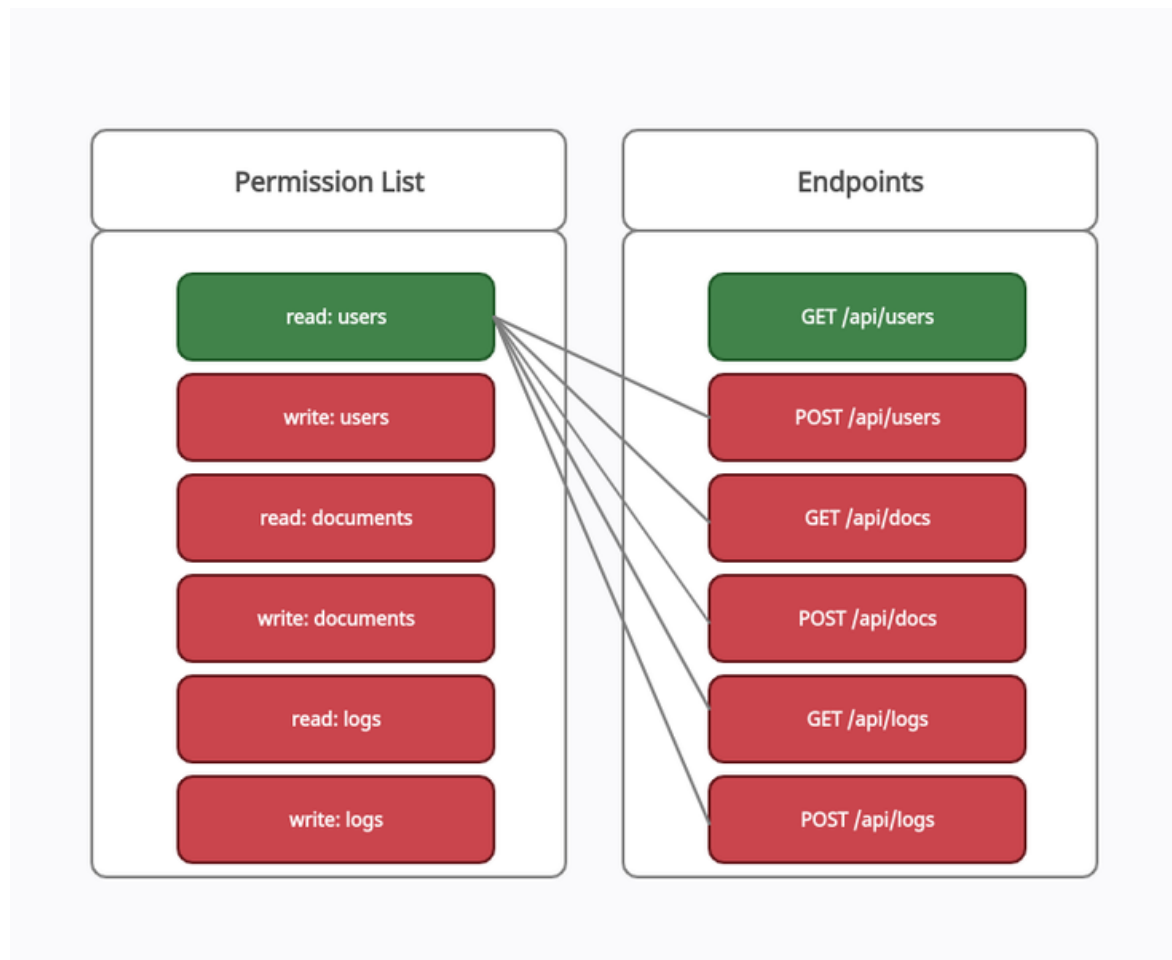
EXAMPLE:

Supposed this is our target App



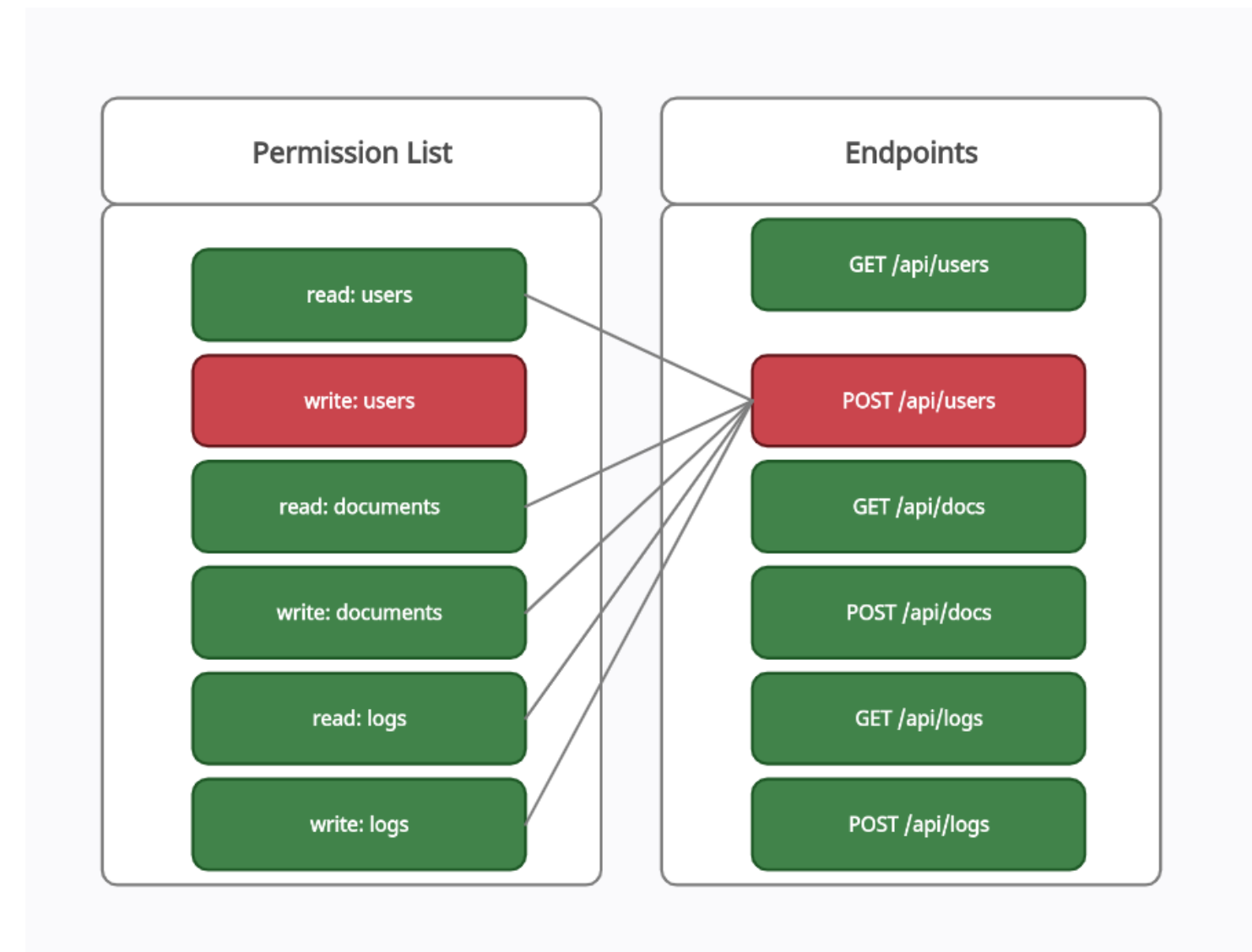
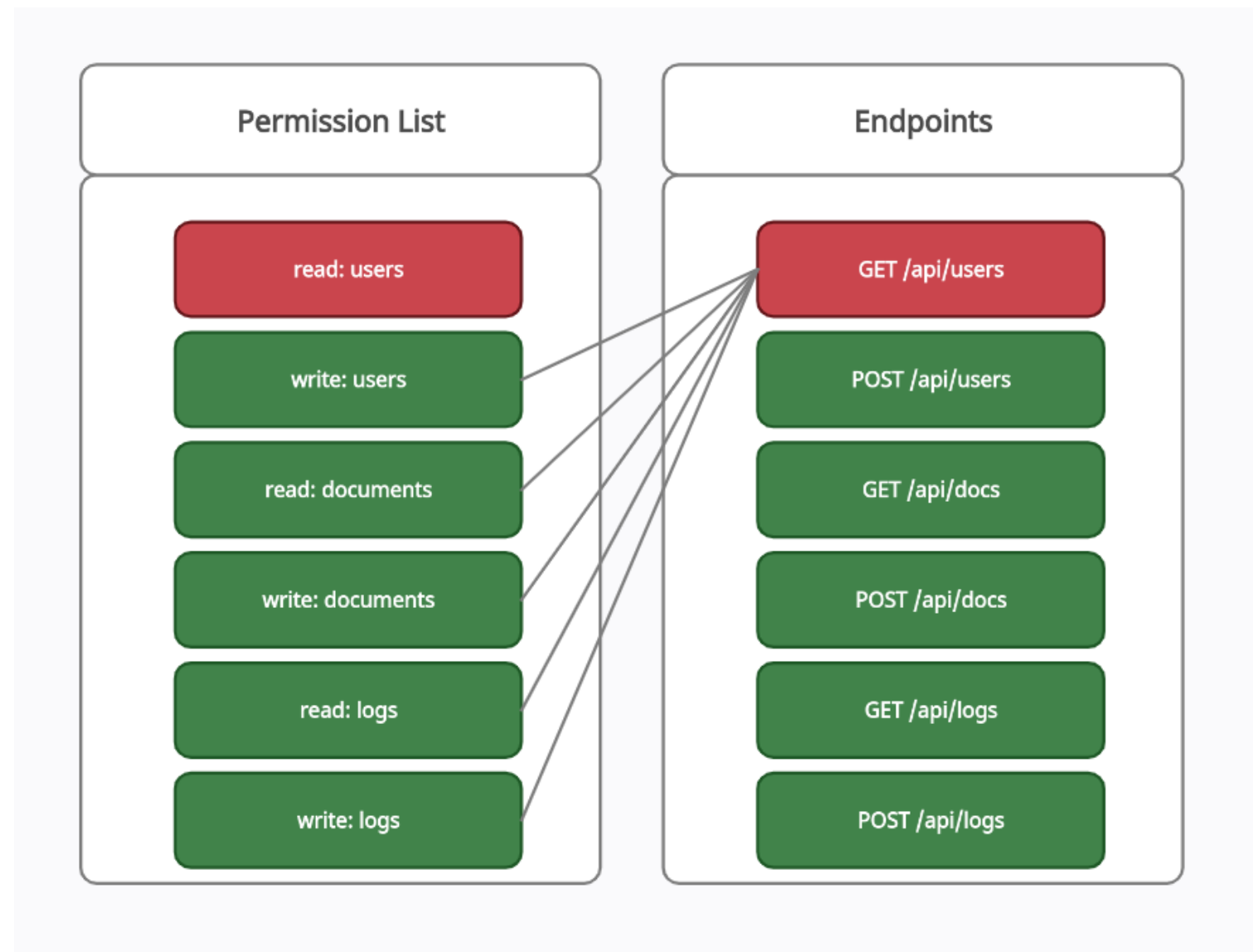
FORWARD APPROACH

Give one permission and target all other permission



BACKWORD APPROACH

Give all permissions and target one permission



THE RIGHT MINDSET

FEW MORE TIPS

Translating Endpoints

Modern Api's designing encourages **Consistency** and a **hierarchical relational** Approach.

GET /api/users/111/folders/222/files/333



This endpoint Lets me access File 333 inside Folder 222 of user 111

GET /api/users/111/reports/1123



This endpoint Lets me access File 333 inside Folder 222 of user 111

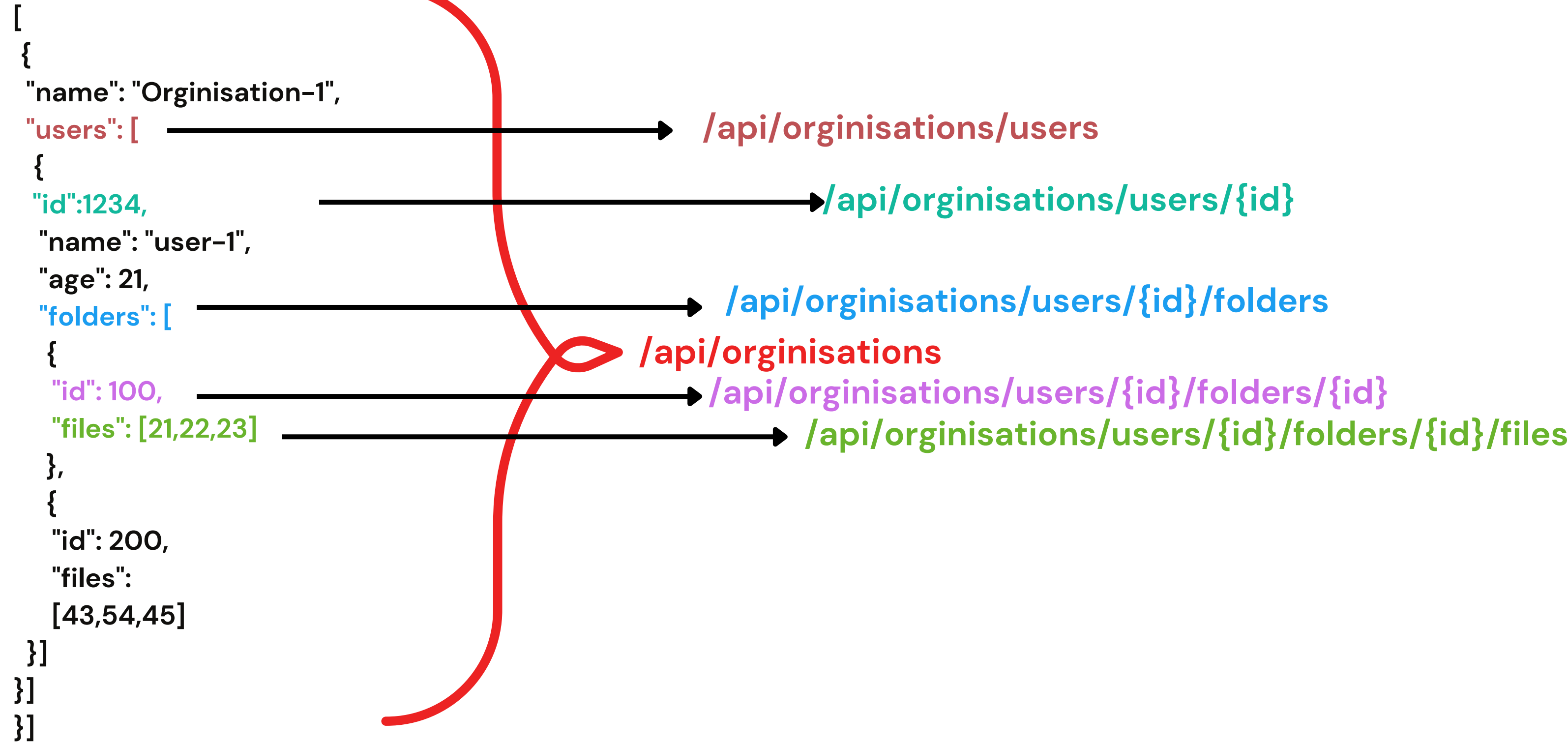
How can I access the files of an user 33 in folder 21



GET /api/users/33/folder/21/files

Translating Responses

HTTP RESPONSE



Guessing Endpoints

Modern Api's designing encourages Consistency and a **hierarchical relational** Approach.

GET /api/users/111/folders/222/files/333

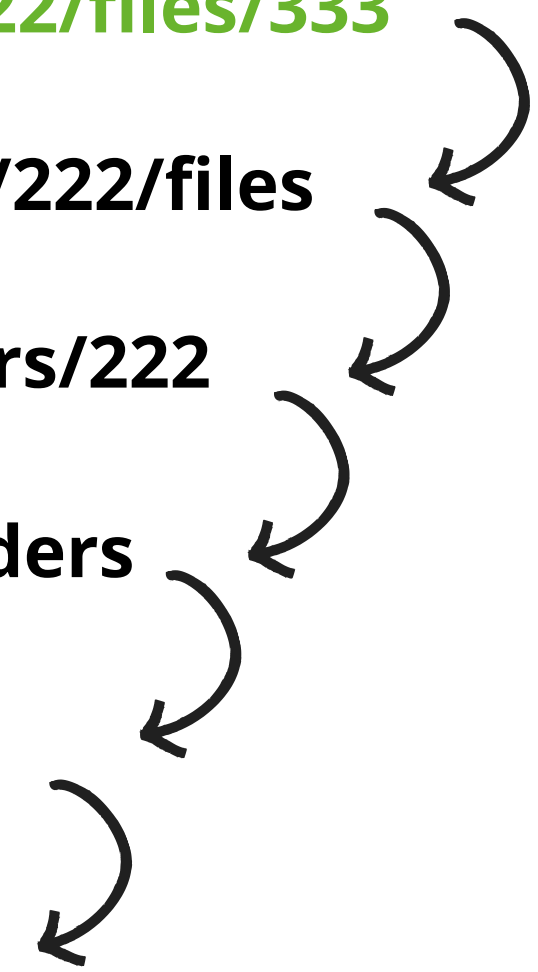
GET /api/users/111/folders/222/files

GET /api/users/111/folders/222

GET /api/users/111/folders

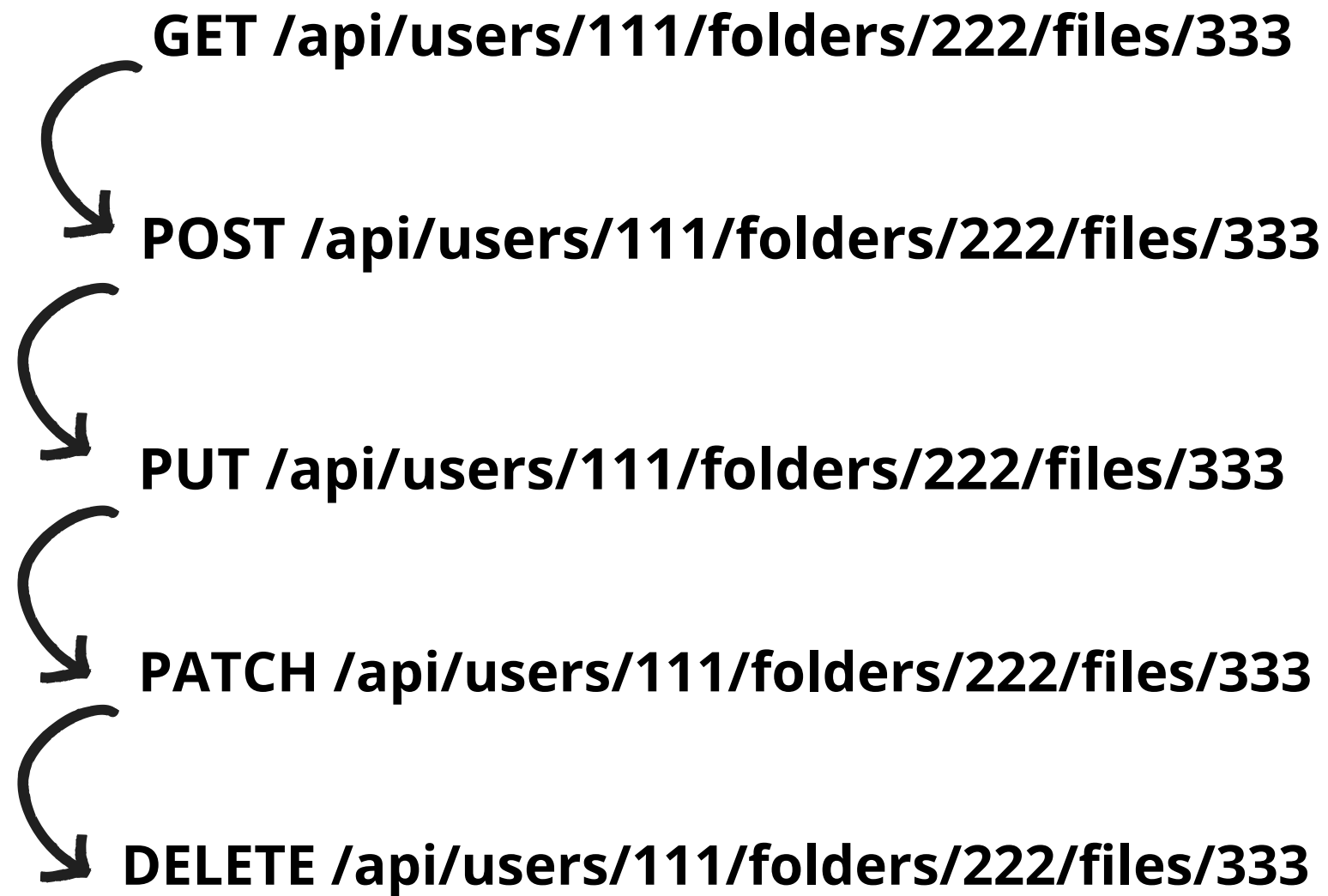
GET /api/users/111

GET /api/users



Guessing Methods

Modern Api's designing encourages **Consistency** and a hierarchical relational Approach.



THE RIGHT TOOLSET

LESS IS MORE

AUTHORIZE

Automatic authorization enforcement detection extension for Burp Suite

The screenshot displays the Burp Suite interface with the 'Authorize' tab selected. The main window is divided into two panes. The left pane shows a list of intercepted requests with columns for 'URL' and 'Authorization Enforcement Status'. The right pane contains configuration options for the extension.

URL	Authorization Enforcement Status
https://github.com:443/Quit/en/Autorize	Authorization enforced!!! (please configure enforcement detector)
https://github.com:443/Quit/en/Autorize	Authorization enforced!!! (please configure enforcement detector)
https://github.com:443/Quit/en/Autorize/show_partial:partial=recently_touched_branches_list	Authorization enforced!!! (please configure enforcement detector)
https://github.com:443/Quit/en/Autorize/issues/counts	Authorization bypass:
https://github.com:443/_sockets	Authorization enforced!!! (please configure enforcement detector)
https://www.google-analytics.com:443/collect	Authorization bypass:
https://www.google-analytics.com:443/collect?v=1&_v=j30&a=390061675&t=pageview&_s=1&dl=https%3A%2F%2Fgithub.com%2FQuit/en%2FAutori...	Authorization bypass:
https://collector.githubapp.com:443/github/page_view?dimensions(page)=https%3A%2F%2Fgithub.com%2FQuit/en%2FAutorize&dimensions(title)=Q...	Authorization bypass:
https://github.com:443/_stats	Authorization bypass:
https://fbcdn-video-d-a.akamaihd.net:443/hvideo-ak-xpa1/v/t42.1790-2/10950765_10155225512495112_67071319_n.mp4?rl=549&vabr=305&oh=726ae3fd5...	Authorization bypass:
https://github.com:443/Quit/en/Autorize	Authorization enforced:
https://github.com:443/Quit/en/Autorize/show_partial:partial=recently_touched_branches_list	Authorization enforced!!! (please configure enforcement detector)
https://github.com:443/Quit/en/Autorize/issues/counts	Authorization bypass:
https://github.com:443/_sockets	Authorization enforced!!! (please configure enforcement detector)
https://www.google-analytics.com:443/collect	Authorization bypass:
https://www.google-analytics.com:443/collect?v=1&_v=j30&a=1052251930&t=pageview&_s=1&dl=https%3A%2F%2Fgithub.com%2FQuit/en%2FAuto...	Authorization bypass:
https://0-edge-chat.facebook.com:443/pull/channel=p_1164700792&seq=7&partition=-2&clientid=418e75d7&cb=fzom&idle=6&cap=8&uid=1164700792&...	Authorization enforced:
https://collector.githubapp.com:443/github/page_view?dimensions(page)=https%3A%2F%2Fgithub.com%2FQuit/en%2FAutorize&dimensions(title)=Q...	Authorization bypass:
https://github.com:443/_stats	Authorization bypass:

Configuration Panel:

- Authorization checks: **Intercept is on** (green button)
- Ignore 304/204 status code responses:
- Prevent 304 Not Modified status code:
- Auto Scroll:
- Clear List: (button)
- Cookie: test=test

Enforcement Detector / Interception Filters:

- Type: Content-Length. (constant Content-Length number of enforced response)
- Content: 33067
- Add filter: (button)
- Filter List: Content-Length: 33067
- Remove filter: (button)

<https://portswigger.net/bappstore/f9bbac8c4acf4ae4d7dc92a991af2f>

AUTO-REPEATER

This extension automatically repeats requests, with replacement rules and response diffing.

The screenshot shows the Burp Suite interface with the AutoRepeater extension settings. The main window displays a list of requests and responses. A 'Deactivate AutoRepeater' dialog is open, showing a table of replacement rules. The 'Log Filter' tab is active, and a rule is defined with the following settings: Boolean Operator: And, Match Original Or Modified: Original, Match Type: String In Request, Match Relationship: Does Not Match, Match Condition: socket.io. An 'Edit Filter' dialog is also open, showing the same settings for the selected rule.

#	Method	URL	Orig. S...	Status	Orig. R...	Resp. ...	Resp. ...
9	GET	https://juice-shop.herokuapp.com:443/?debug=1	503	503	710	710	0
8	GET	https://juice-shop.herokuapp.com:443/?debug=1	503	503	710	710	0
7	GET	https://juice-shop.herokuapp.com:443/?debug=1	503	503	710	710	0
3	GET	https://juice-shop.herokuapp.com:443/?debug=1	503	503	710	710	0
2	GET	https://juice-shop.herokuapp.com:443/?debug=1	503	503	710	710	0
1	GET	https://juice-shop.herokuapp.com:443/?debug=1	503	503	710	710	0

Log Filter	Log Highlighter
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

Add	Enabled	Boolean Operator	Original Or Modified	Match Type	Match Relationship	Match Condition
<input type="button" value="Add"/>	<input checked="" type="checkbox"/>	And	Original	String In Request	Does Not Match	socket.io

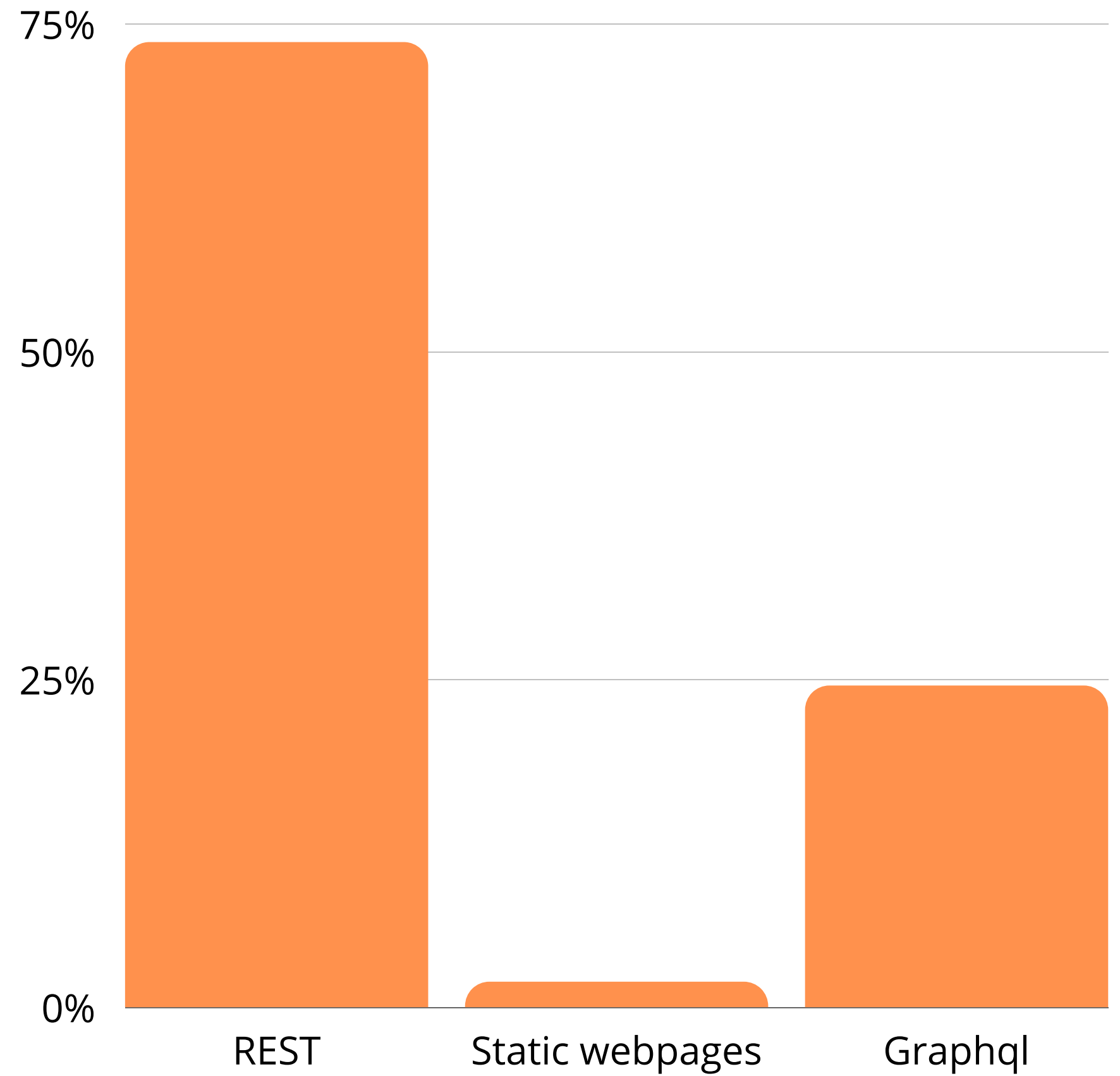
Boolean Operator: And
Match Original Or Modified: Original
Match Type: String In Request
Match Relationship: Does Not Match
Match Condition: socket.io

STATISTICS

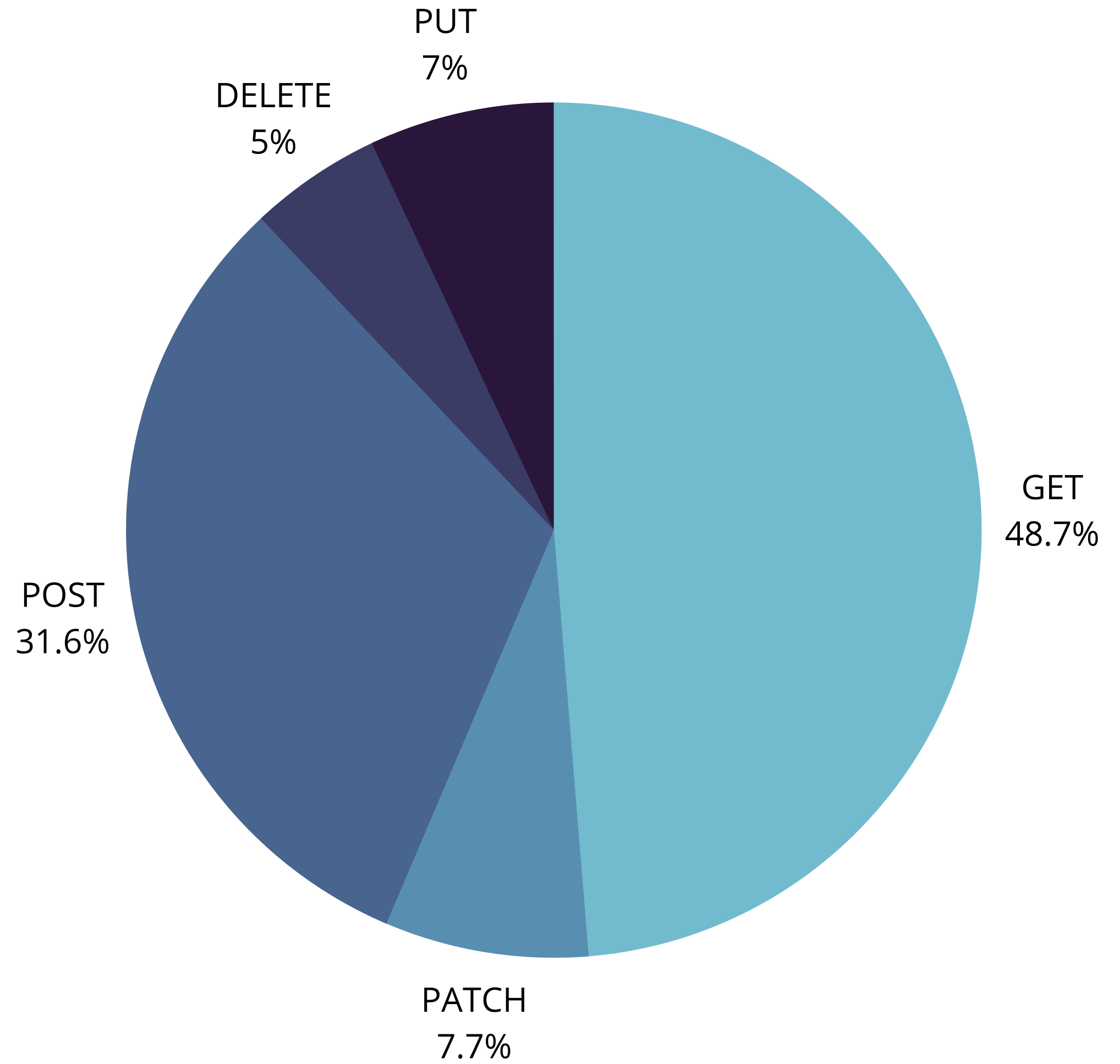
WHAT HAVE WE LEARNED AFTER FINDING
AND REPORTING **~800** ACCESS CONTROL
ISSUES TO **~100** COMPANIES.



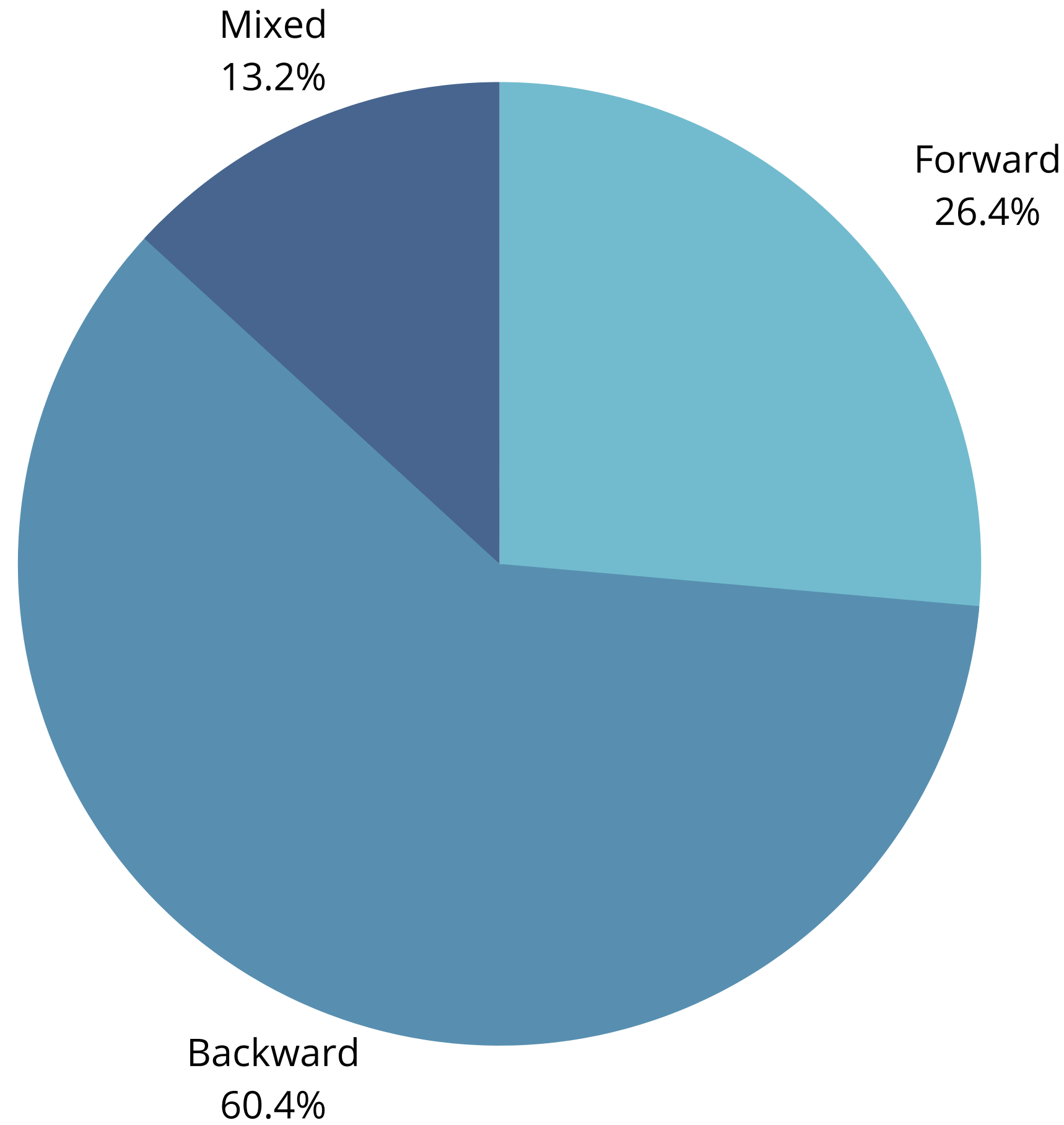
VULNERABLE ENTITIES



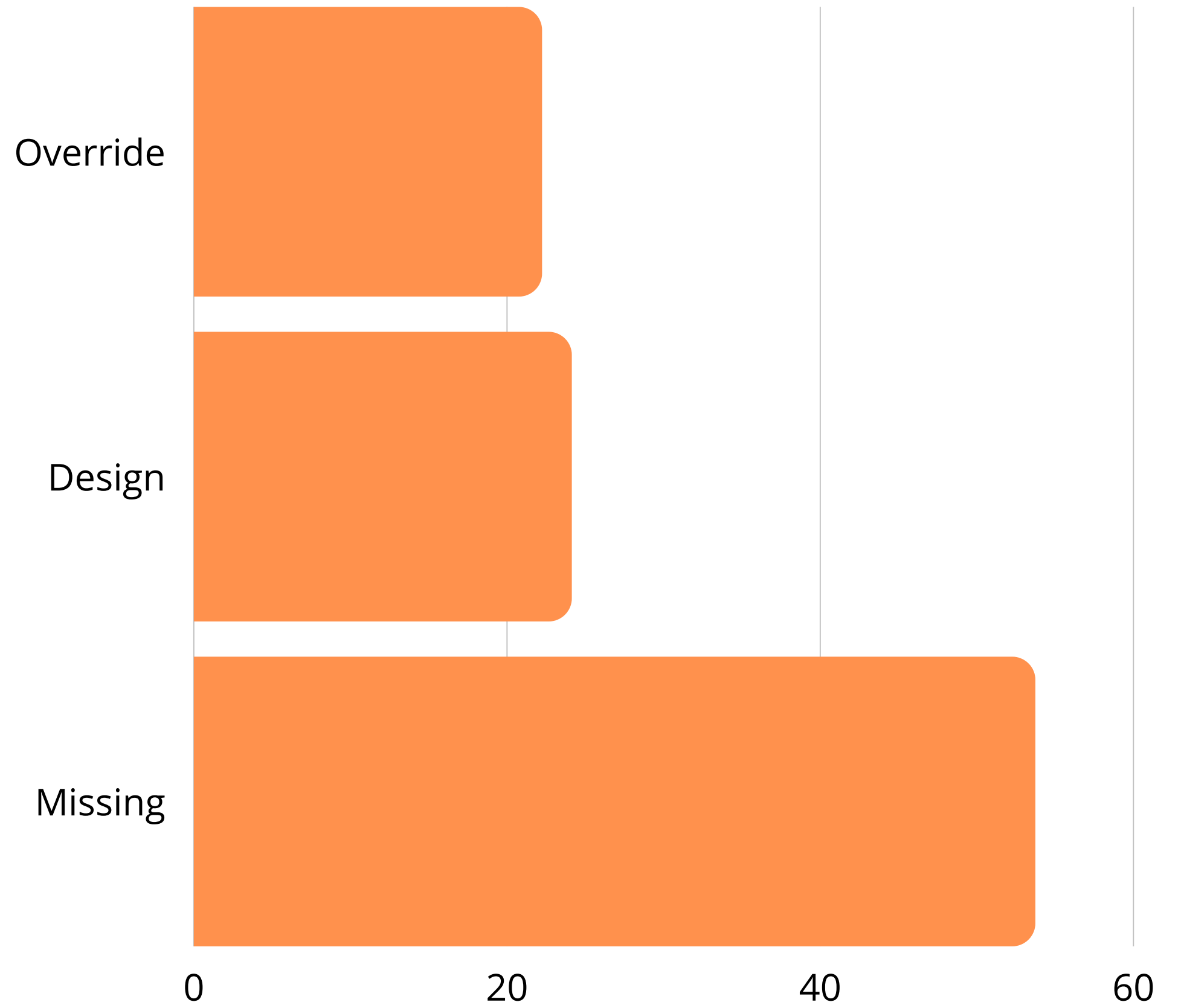
HTTP-METHODS



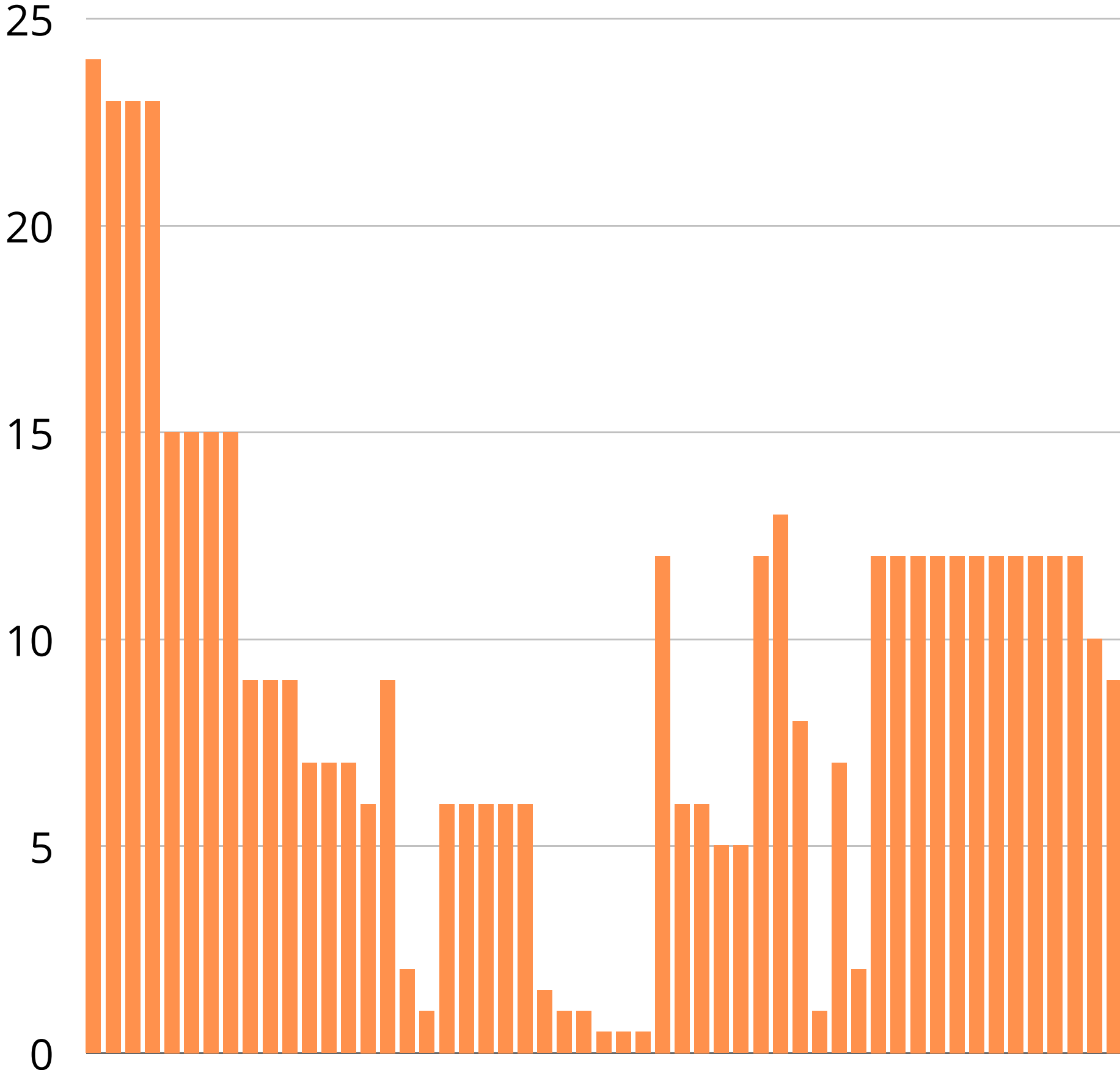
APPROACHES USED



FLAW-TYPES



FIX TIME (MONTHS)




BLOGPOST


<https://snapsec.co/blog/Attacking-Access-Control-Models-In-Modern-Web-Applications/>

The diagram illustrates the relationship between a user, permissions, and API endpoints. A central orange circle labeled 'User' is connected to two columns. The left column, titled 'Permission List', contains six items: 'read: users' (green), 'write: users' (red), 'read: documents' (red), 'write: documents' (green), 'read: logs' (red), and 'write: logs' (green). The right column, titled 'Endpoints', contains six items: 'GET /api/users' (green), 'POST /api/users' (red), 'GET /api/docs' (red), 'POST /api/docs' (green), 'GET /api/logs' (red), and 'POST /api/logs' (green). Lines connect the 'User' to each item in both columns.

Attacking Access Control Models in Modern Web Apps

So far you may have come across various web applications where you were able to invite members with limited access to the information within the organization. Developers are able to...

 Imran Parray
25 Sep 2021



QUESTIONS??

