

KYC - Know Your Content for the week. This week's topic - **Cross Origin Resource Sharing (CORS)** !

All the Best !

✓ The second header which we should look out while exploiting CORS is * 1/1

- Access-Control-Allow-Credentials
- Access-Common-Allow-Credentials
- Access-Control-Deny-Credentials
- Access-Common-Deny-Credentials



✓ What do you mean by "Access-control-allow-origin: * " *

1/1

- The vulnerable website will share its resources with everyone else except [attacker.com](#)
- The vulnerable website will share its resources with everyone
- The vulnerable website will share its resources with no one
- None of the above



✓ In the lab "CORS with prefix match" the value of Origin header used was * 1/1

- [hacktify.in.evil.com](#)
- [hacktify.evil.com](#)
- [hacktify.com](#)



[evil.com](#)

✓ What is the severity of CORS where PII is not leaked * 1/1

P3 ✓

P4

P5

None of the above

✓ What do you mean by "Access-control-allow-origin:[attacker.com](#)" * 1/1

The vulnerable website will share its resources with everyone else except [attacker.com](#)

The vulnerable website will share its resources with [attacker.com](#) ✓

The vulnerable website will block [attacker.com](#)

None of the above

✓ Which one of the following is the curl command to check if [example.com](#) 1/1
is vulnerable to CORS or not *

curl [example.com](#) -H Origin: [attacker.com](#)

curl [example.com](#) -I -H Origin: [attacker.com](#) ✓

Both A and B

Both A and B

None of the above

The CORS exploit script is written in * 1/1

HTML

AJAX ✓

JavaScript

PHP

In the lab "CORS with Substring match" the value of Origin header used was * 1/1

hacktify.com

hacktify.co ✓

co.hacktify.co

hacktify.in

CORS comes under which category * 1/1

Broken Access Control

Broken Authentication

Security Misconfiguration ✓

Insufficient Logging and Monitoring

_____ should be avoided in internal networks * 1/1

- Blacklisting
- Wildcards ✓
- Both A and B
- None of the above

In the lab "CORS with Null origin" the value of access-control-allow-origin was * 1/1

- null ✓
- evil.com
- null.evil.com
- null.com

CORS protocol uses _____ to trust web origins * 1/1

- Cookies
- HTTPS
- HTTP Headers ✓

- All of the above

✓ Avoiding whitelisting of _____ to prevent CORS *

1/1

- null ✓
- #
- Origin Header
- None of the above

✓ Which one of the following is NOT an impact of CORS

1/1

- Account Takeover
- Sensitive Data Exposure
- Cookie Stealing
- Database exposure ✓

✓ In the command "curl example.com -I -H Origin: attacker.com" -H is used to *

- Show response with indentation
- Send a custom header ✓
- Return headers of the responses



- Tell curl that CORS is being performed

✓ For which of the following case/s CORS CANNOT be exploited *

1/1

- Access-Control-Allow-Origin : [website.com](#)
- Access-Control-Allow-Origin : *
- Access-Control-Allow-Origin : null
- All of the above



✓ In the command "curl [example.com](#) -I -H Origin: [attacker.com](#)" -I is used to *

1/1

- Show response with indentation
- Send a custom header
- Return headers of the responses
- Tell curl that CORS is being performed



✓ In the CORS exploit script "this.readyState == 4" means *

1/1

- Response is ready to be shown
- Request not initialized
- Server connection established



Request is processing

If a user does not know to use curl he/she can use _____ instead

1/1

gau

waybackurls

grep

Burp Suite

✓

What is the severity of CORS where PII is leaked *

1/1

P2

✓

P3

P4

P5

What was the payload used in the lab "CORS With Arbitrary Origin" *

1/1

Add Original Header with value evil.com

Add X-Frames-Origin Header with value evil.com

Add Origin Header with value evil.com

✓

None of the above



✓ In the lab "CORS with Arbitrary Subdomain" the value of Origin header used was *

- [hacktify.com](#)
- [hacktify.in](#)
- somesubdomain.hacktify
- [somesubdomain.hacktify.in](#)



✓ Which one of the following is not a test case of CORS *

- Access-control-allow-origin:[attacker.com](#)
- Access-control-allow-origin:*
- Access-control-allow-origin:null
- Access-control-allow-origin:#



✓ CORS extends and adds flexibility to *

- Same-Origin Policy
- Cross-Origin Policy
- Content Security Policy



None of the above

✓ In the lab "CORS with suffix match" the value of Origin header used was * 1/1

- [hacktify.com](#)
- [evil.com](#)
- [evil.in](#)
- [evilhacktify.in](#)



✓ The first header which we should look out while exploiting CORS is * 1/1

- Access-Common-Allow-Origin
- Access-Control-Allow-Origin
- Access-Common-Deny-Origin
- Access-Control-Deny-Origin



✓ Which one of the following is a prevention of CORS * 1/1

- Only allow trusted websites
- Proper configuration of cross-domain requests
- Remove Origin Header
- Both A and B



✓ CORS Stands for *

1/1

- Cross Origin Resource Swapping
- Cross Origin Resource Sharing
- Cross Operation Resource Sharing
- Cross Operation Resource Swapping

✓ In the lab "CORS with Escape dot" the value of Origin header used was * 1/1

- hacktify.in
- www.hacktify.in
- www.hacktify.in
- www.in

This form was created inside of VT.

Forms

