Hi Team
I am a freelance web security auditor and bug bounty hunter. I identify security loopholes and flaws that can lead to the website security compromise or website users security compromise. I have discovered a bug in your website.

Bug: No Proxy Protection Impact: Account Compromise/Click jacking

Description:

CORS Proxy simply takes advantage of Cross-Origin Resource Sharing and this is a feature which was added in HTML 5. With this serves can specify that they want browsers to allow other websites to request resources they host.

CORS Proxy is simply an HTTP Proxy that adds up a header to responses saying “Anyone can request this” or in technical terms it adds Access-Control-Allow-Origin with (*) value and as it can specify headers so it can also strip away headers. Which is the reason it can strip away the XFrame-Options value whether it be set to “DENY” or “SameOrigin” and like this the CSP Policy and CrossOrigin policy can also be bypassed.

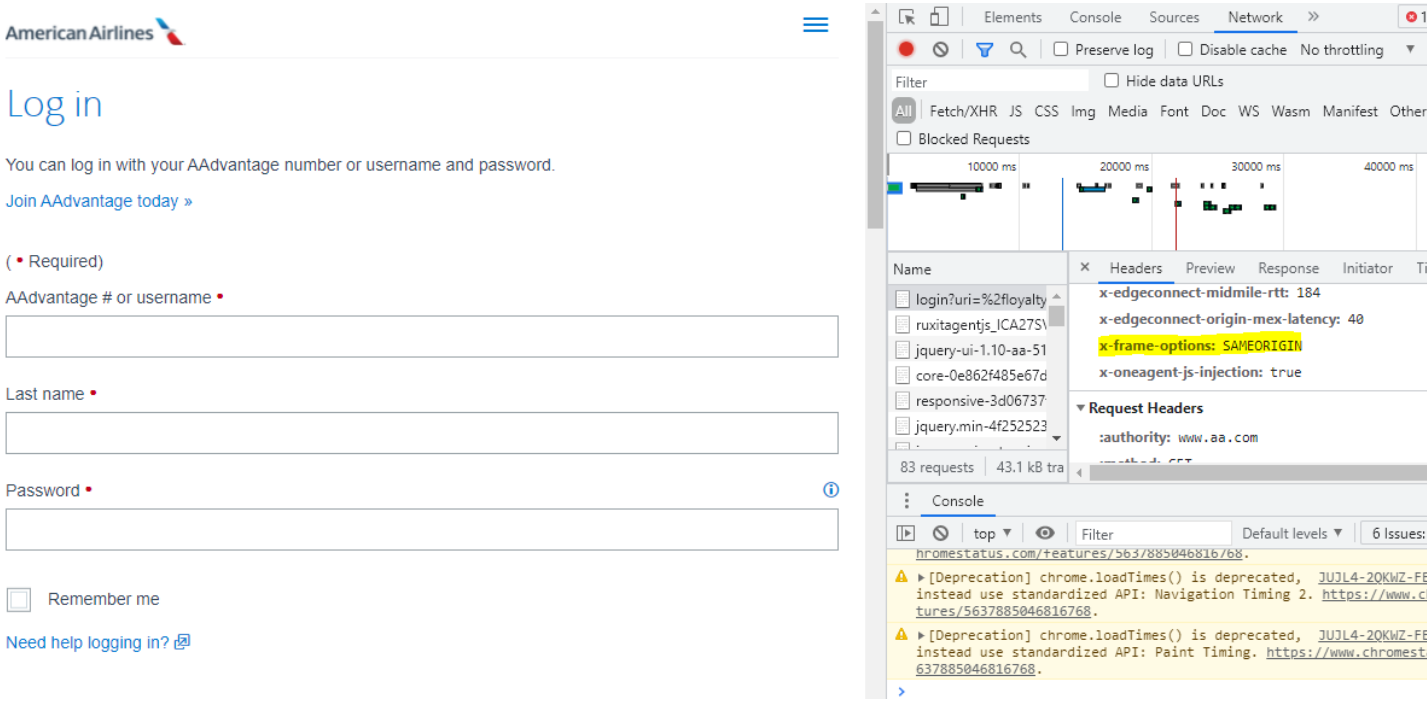
Clickjacking Protection on:
https://www.aa.com/loyalty/login?uri=%2floyalty%2flogin&previousPage=%2fhomePage.do&bookingPathStateId=&marketId=

Test a page for clickjacking/framing vulnerability

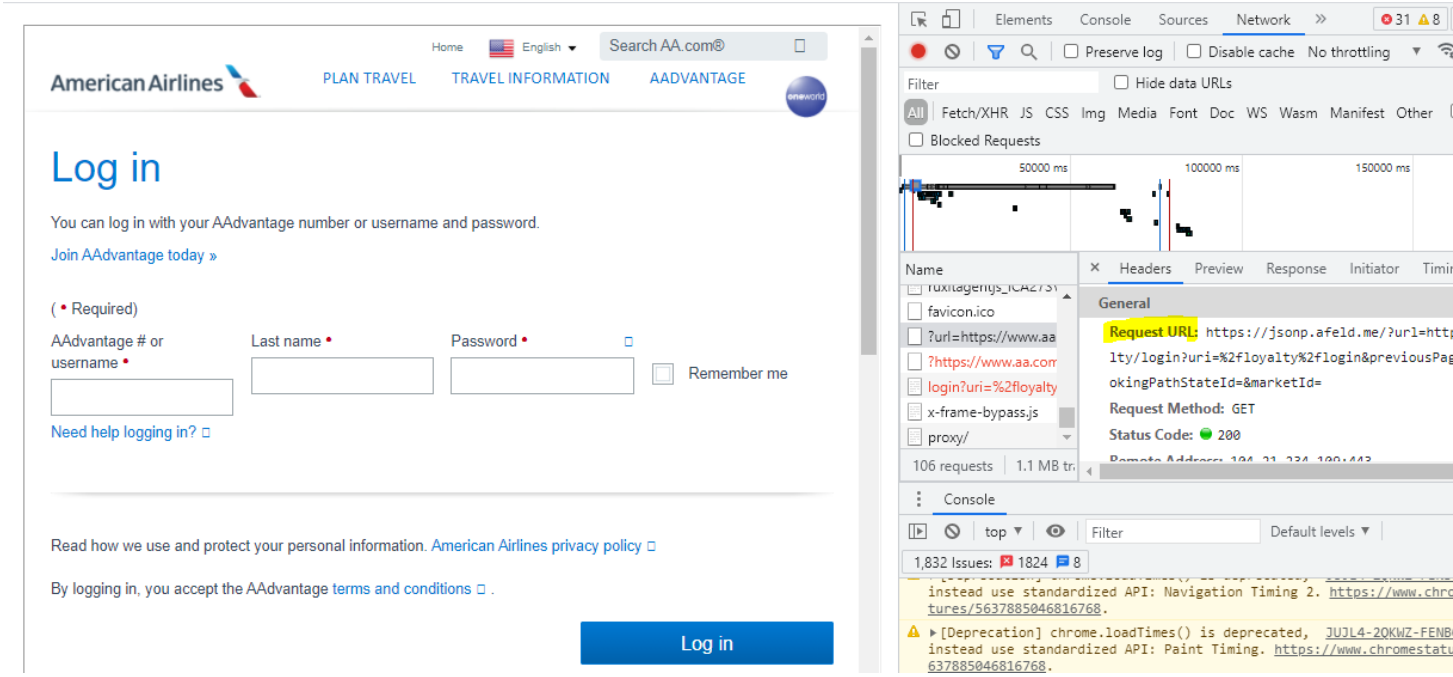
Enter the URL to frame:

https://www.aa.com/loyalty/lo Test it!

Current Fix deployed for clickjacking (X-Frame-Options: `sameOrigin`)



Screenshot Proof: (x-frame-options:SameOrigin bypassed)



Exploit Code + Proof Of Concept:

You can access the exploit code from here : https://github.com/niutech/x-frame-bypass
You would need to place the website link in the iframe provided in the index.html file while simultaneously hosting both the exploit files that are index.html and x-frame-bypass.js.

Impacts: The site can also be opened in an iframe after the user has logged it making it hard for the user to avoid phishing. A user can be tricked into entering his credentials in what he may think are the placeholder for the original website details. And thus his credentials would be sent to the attacker as shown in POC. User's account can be compromised using above POC. Attacker can get full access to the account.

Remediation:

You must enable proxy protection for your website.

This can be easily verified by visiting this link which would open your website through an un authorized proxy.

http://younow-cors-header.herokuapp.com/index.php?q=https://builtworlds.com/login/

Also you can visit the below links for ffsng.com which has implemented by provided fix to detect the vpn and redirect users to a vpn detected page.

http://younow-cors-header.herokuapp.com/index.php?q=https://ffsng.com

Or you can visit the second link which for godaddy.com requested through the same proxy which straight breaks the context of the proxy once detected and replace the link with the help of document.location.replace.href() function.

http://younow-corsheader.herokuapp.com/index.php?q=https://sso.godaddy.com/?realm=idp&app=dashboard.api&path= %2fvh-login-redirect&marketid=en-PK

Reference:

https://hackerone.com/reports/85624 https://hackerone.com/reports/7264

Looking forward to hearing from you on this. Also hoping to receive a bug bounty reward for reporting this.

Best Regards

Possible Fix (You would need to have the below code added to your .htaccess file and rewrite the websites access criteria. This fix also allows for white listing any trusted third party proxy).

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteCond %{HTTP:VIA} !^$ [OR]
RewriteCond %{HTTP:FORWARDED} !^$ [OR]
RewriteCond %{HTTP:USERAGENT_VIA} !^$ [OR]
RewriteCond %{HTTP:X_FORWARDED_FOR} !^$ [OR]
RewriteCond %{HTTP:PROXY_CONNECTION} !^$ [OR]
RewriteCond %{HTTP:XPROXY_CONNECTION} !^$ [OR]
RewriteCond %{HTTP:HTTP_PC_REMOTE_ADDR} !^$ [OR]
RewriteCond %{HTTP:XROXY_CONNECTION} !^$ [OR]
RewriteCond %{HTTP:X-FORWARDED-FOR} !^$ [OR]
RewriteCond %{HTTP:HTTP_CLIENT_IP} !^$ [OR]
RewriteCond %{HTTP:FORWARDED-FOR} !^$ [OR]     RewriteCond %{HTTP:X-FORWARDED} !^$

RewriteRule ^(.*)$ - [F]
</IfModule>
```