jamk.fi

Web application security

Week 06

Timo Lehosvuo, M3426
TTV18S1

Harjoitustyö
Web application security, Heikki Salo, Joni Ahonen
28.3.2021
Tekniikan ala

1. Other

CVE-2019-19226: DSL-2680 Broken Access Control - Enable/Disable MAC address filter (https://github.com/x0e1f/DSL-2680-multiple-vulnerabilities/blob/master/CVE-2019-19226.md)

This vulnerability makes it possible for the attacker to enable or disable MAC address filtering on the DSL-2680 wireless router by uploading a crafted POST request to "Forms/WlanMAcFilter_1" without being authenticated on the admin interface. This could potentially let through or block wanted/unwanted traffic on the network. This can be fixed by implementing access control mechanisms, deny access to functionalities by default and disabling web server directory listing. Example command: curl -d "WLAN_FltActive=0" -X POST "http://192.168.1.1/Forms/WlanMacFilter_1".

CVE-2019-19224: DSL-2680 Broken Access Control - Download Router Configuration (https://github.com/x0e1f/DSL-2680-multiple-vulnerabilities/blob/master/CVE-2019-19224.md)

This vulnerability makes it possible for the attacked to download the routers current configuration file by submitting a "rom-0" GET request without being authenticated on the admin interface. This could potentially allow hacker to better exploit the router since it knows it configurations and potentially hack the whole network. Fix to this vulnerability is to implement access control mechanisms, deny access to functionalities by default and blocking the download of files. Example command: *curl http://192.168.1.1/rom-0 --output configFile.*

CVE-2019-19223: DSL-2680 Broken Access Control - Reboot Router
(https://github.com/x0e1f/DSL-2680-multiple-vulnerabilities/blob/master/CVE-2019-19223.md)

This vulnerability makes it possible for the attacker to reboot the router by submitting a “reboot.html” GET request without being authenticated on the admin interface. This could potentially shutdown the whole network if there is no other way for the network traffic to travel. Fix to this vulnerability is to implement access control mechanism and deny access to functionalities by default. Example command: *curl http://192.168.1.1/reboot.html.*

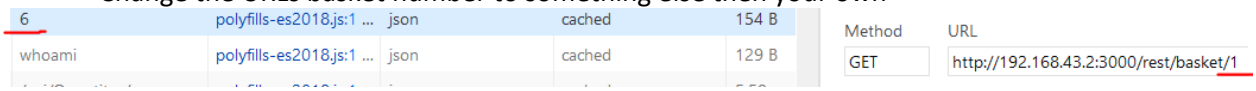
2. Issue report

2.1 Attacker is able to view another user’s shopping basket

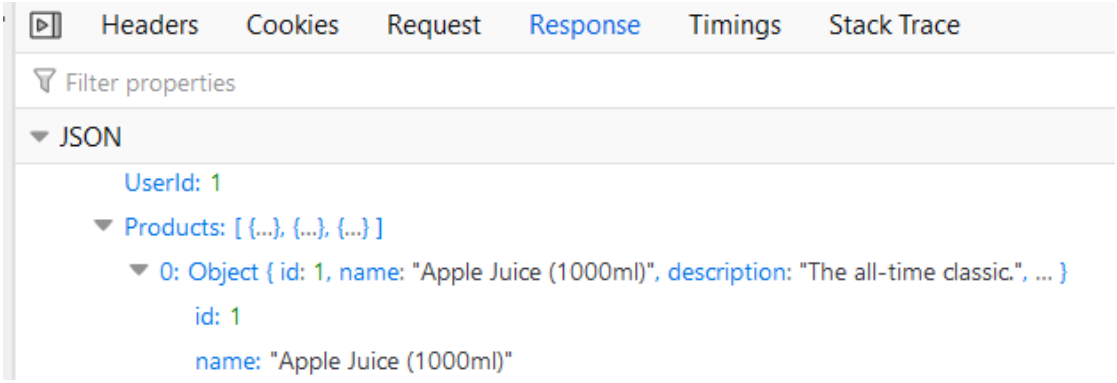
Description: Attacker is able to modify the request that contains shopping basket information and change the basket id to view another user’s basket

Steps to produce:

- Login to juice shop
- Start to inspect the network traffic with “Inspect element”
- Click “Your basket”
- Find the *GET* request with your *userid*
- Right click it and select “Edit and Resend”
- Change the URLs basket number to something else then your own



- Find the new GET request
- You can now look what is on the other persons basket from the response tab



Mitigation:

- Deny the access to functionalities by default
- Use access control lists and role-based authentication mechanisms
- Hide or encrypt the traffic so it's not plaintext

3 Issue report

3.1 Attacker is able give feedback using another user's name

Description: Attacker is able to modify the POST request and change payload to give feedback as another user

Steps to produce:

- Login to juice shop
- Go to "Customer feedback" from the side menu (top left corner)
- Start to inspect the network traffic with "Inspect element"
- Write a feedback and send it
- Find your own POST request and click "Edit and Resend"
- Modify the values to match another user and send it

Method URL

POST http://192.168.43.2:3000/api/Feedbacks/

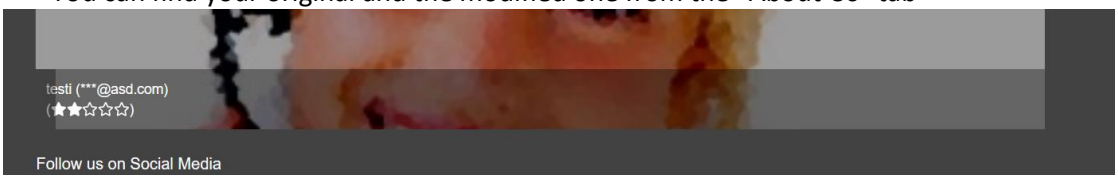
Request Headers:

```
Host: 192.168.43.2:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: application/json, text/plain, */*
Accept-Language: fi-FI,fi;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.43.2:3000/
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6
Content-Type: application/json
```

Request Body:

```
{"UserId":1,"captchaId":6,"captcha":"18","comment":"testitesti (admin@juice-sh.op)","rating":5}
```

- You can find your original and the modified one from the “About Us” tab



Mitigation:

- Deny the access to functionalities by default
- Use access control lists and role-based authentication mechanisms
- Hide or encrypt the traffic so it’s not plaintext

Score Board

References

- https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html
- https://pwning.owasp-juice.shop/part2/broken-access-control.html#post-some-feedback-in-another-users-name
- https://pwning.owasp-juice.shop/part2/broken-access-control.html#view-another-users-shopping-basket
- https://github.com/x0e1f/DSL-2680-multiple-vulnerabilities/blob/master/CVE-2019-19223.md
- https://github.com/x0e1f/DSL-2680-multiple-vulnerabilities/blob/master/CVE-2019-19224.md
- https://github.com/x0e1f/DSL-2680-multiple-vulnerabilities/blob/master/CVE-2019-19226.md