mehedishakeel

Nov 2, 2022 · 4 min read · Listen

Improper Access Control — My Third Finding on HackerOne!

Improper Access Control means web application or software functions does not restrict or incorrectly restricts access and usage to any hidden resource or private functionality from an unauthorized actor. I found a bug on a web application through Swagger Api UI where I was able to send emails to anyone from the company mail address, including attachments and also html injections.

What is **Swagger UI**? It's a set of open-source tools built around the OpenAPI Specification that can help you design, build, document and consume REST APIs. The major Swagger tools include: Swagger Editor — browser-based editor where you can write OpenAPI definitions. It's a collection of HTML, JavaScript, and CSS assets that dynamically generate beautiful documentation from a Swagger-compliant API. To learn more about Swagger Api UIs **Here**.

Now let's discuss how I found my third bug and the tools and techniques I used.

It's was the same private program, where I got my second bug **Broken Link Hijacking**. That's why I can't share the company name, but I will try to explain my best so that you can get the exact scenario about finding this bug. I want to mention that the target scope was huge so lets take the example scope domains as

```
*.mehedishakeel.com
```

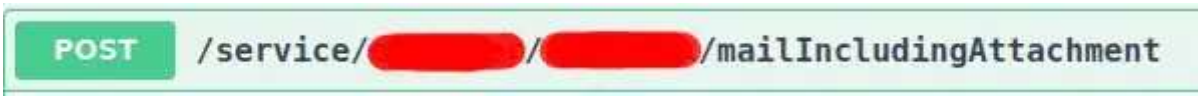
It was my third finding on the previous write up targets so I already had the subdomains which I collected using **subfinder** & **httpx**. Now from all of those subdomain, my target subdomain was

```
api.mehedishakeel.com
```

I used the httpx **-status-code** and **-title -tech-detect** flag during the subdomain enumeration, identifying that this url was using the swagger ui api.

So, I open the url in browser, and get the following page UI:

I start expanding every option on the api, and found that every api call need an authorization tokens or you can say some some sort of keys, except for one option. It doesn't required any sort of keys and token to use the api endpoint.



Expanding this option I found I can make the following request using this endpoint to send emails to anyone including attachments, which doesn't required any tokens authorization or keys:

```
{
  "messageBody": "string",
  "emailTo": "string",
  "attachment": [
    {
      "name": "string",
      "encodedImage": "string"
    }
  ]
}
```

So, first I make a simple request through the ui to send an email, and my request was the following:

```
{
  "messageBody": "this is a test from hackerone",
  "emailTo": "myhandle@wearehackerone.com",
  "attachment": [
    {
      "name": "This is a Test PoC",
      "encodedImage": "testing"
    }
  ]
}
```

As expected, I got **status code 200** in response and then i check my inbox, and found an email coming from the business email of targeted company domain name.

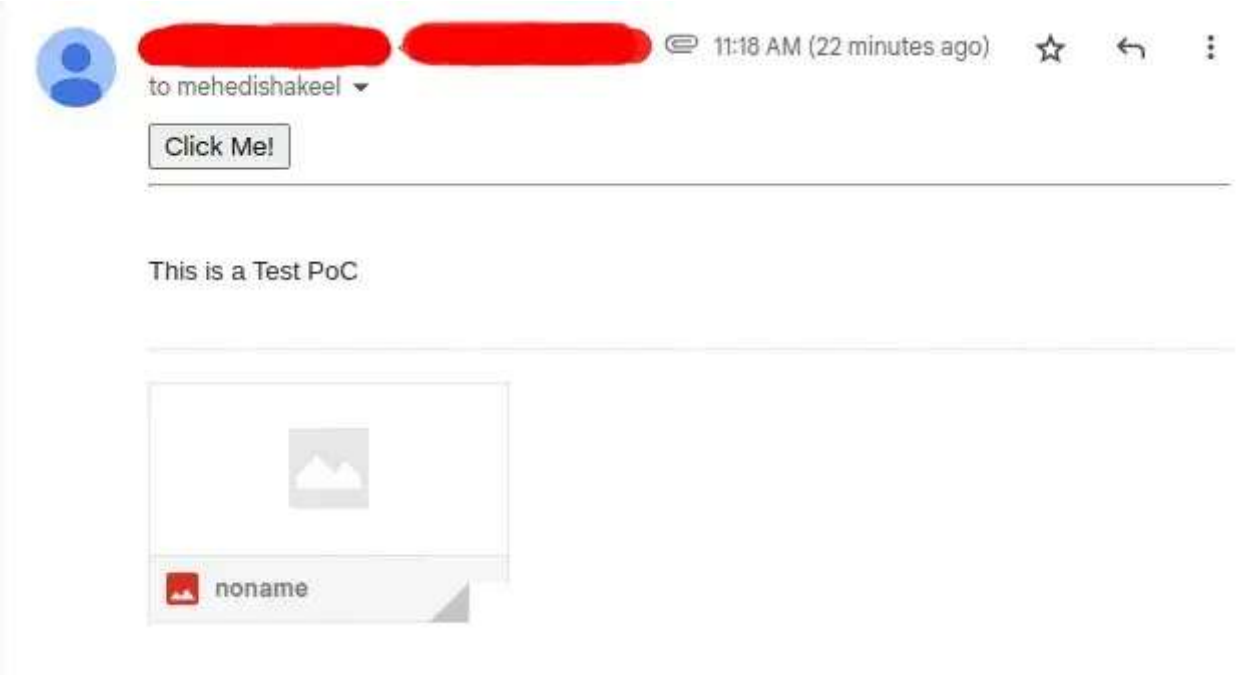
This vulnerability bugs me, so I made more tests. So, I think why not try to inject html code into the message body and see if that works? I make the following changes in my request,

```
{
  "messageBody": "<button>Click Me!</button>",
  "emailTo": "myhandle@wearehackerone.com",
  "attachment": [
    {
      "name": "This is a Test PoC",
      "encodedImage": "testing"
    }
```

```
    ]
}
```

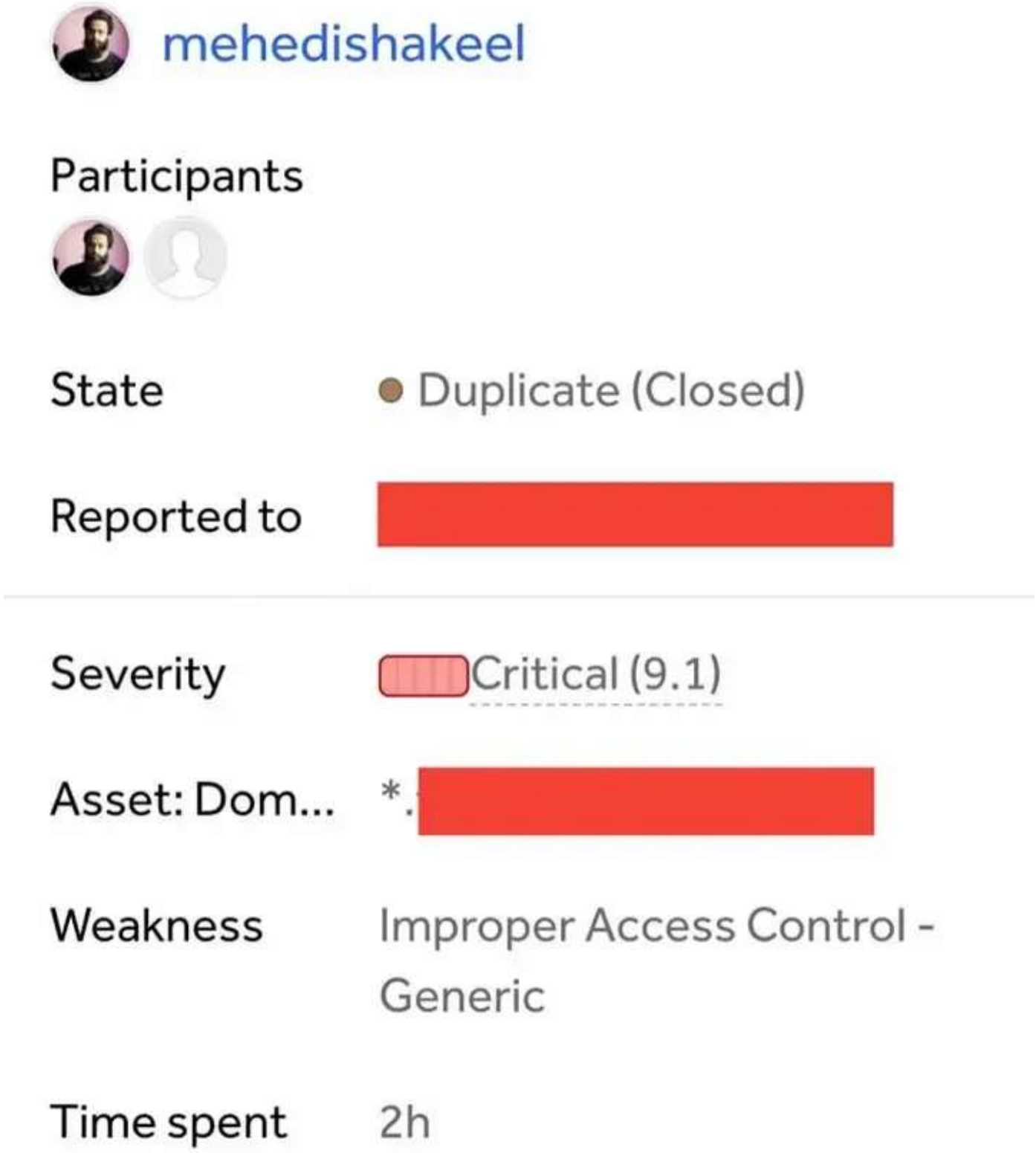
Simple Button Tag as a HTML Injection : **<button>Click Me!</button>**

I got another email with the html injection and a text attachments,



Now it's time submit a detailed report. I prepare a detailed report, guiding every step I took, and make a proper video PoC with a voiceover explaining how the vulnerability works and what are the impact and submit the bug.

Here, my bad luck comes in, as someone else submitted a report with this bug previously. My report was marked as duplicated.



However, I was happy as hunting this bug taught me so much about swagger UI api which I can use in my next findings. That's how I got my third bug on HackerOne. Thank you for reading!

The Gray Area is a collection of great cybersecurity and computer science posts. The best articles are highlighted in a weekly newsletter, sent out every Wednesday. To get updates whenever The Gray Area publishes an article, check out our Twitter page, @TGAonMedium.

This article was slightly modified by TGA Editors for optimized reading.

Hackerone Bug Bounty Bug Bounty Tips Ethical Hacking Cybersecurity