



# AsterFiester

34 Followers About



## \$3133.70 Google Dialogflow IDOR Vulnerability



AsterFiester Just now · 4 min read

Hey, Amazing Hackers, am Raidh Here, Hope, you all are doing well.

I am back again with my 2nd write up on Dialogflow IDOR vulnerability which is interesting to find. So, without wasting any time and lets begin the read.



*haven't you read the previous writeup then please check out.*

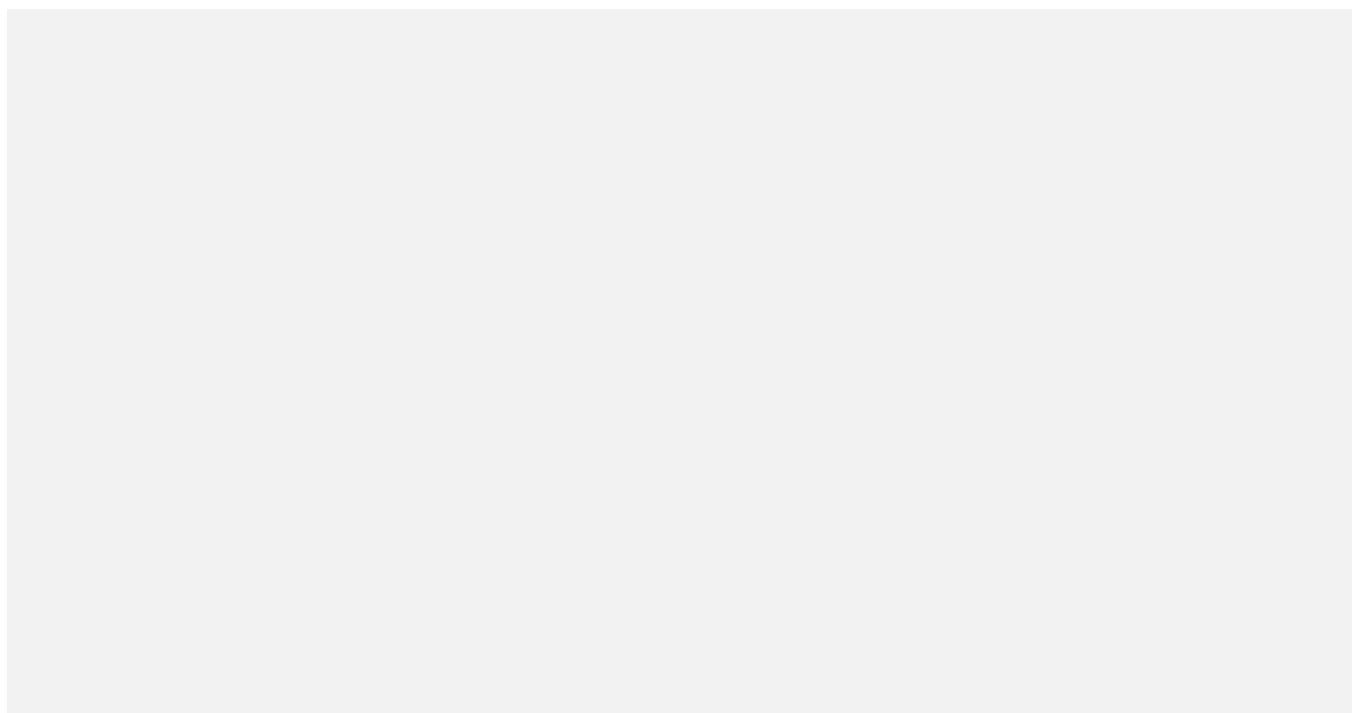
### The Finding

After finding the previous vulnerability I didn't give up. So, I started searching for more bugs and I understand the app is more vulnerable with broken access issues.

The dialogflow have 2 different versions of applications. Dialogflow Essentials and Dialogflow CX. Actually Dialogflow CX is an upgraded version of Essentials with lots of feature. So, I started searching for broken access control issues.

I always test IDOR manually because using automated tools won't make any sense for changing methods adding custom headers and more.

While testing the application, I saw an interesting option called Test Cases.



But the Test Cases are empty and I tried to import them but it didn't work! :(

So, I started searching about the feature but I got nothing. So finally, I decided to read the documentation. I know reading documentation is like breathing in the space :) but I don't have any option, Hackers!

Finally, I found something interesting. There is an amazing feature to build a prebuild agent. you guys might be confused about **Agent**. It is just like a project to create everything, that's all.

Lets make a pre build agent.



There are so many pre build agents available and I just selected the first one.



After creating the prebuild agent. I looked for Test Cases.



Yes, It was there. Whooooo hooooo..



I selected on of the Test Case and clicked Delete.

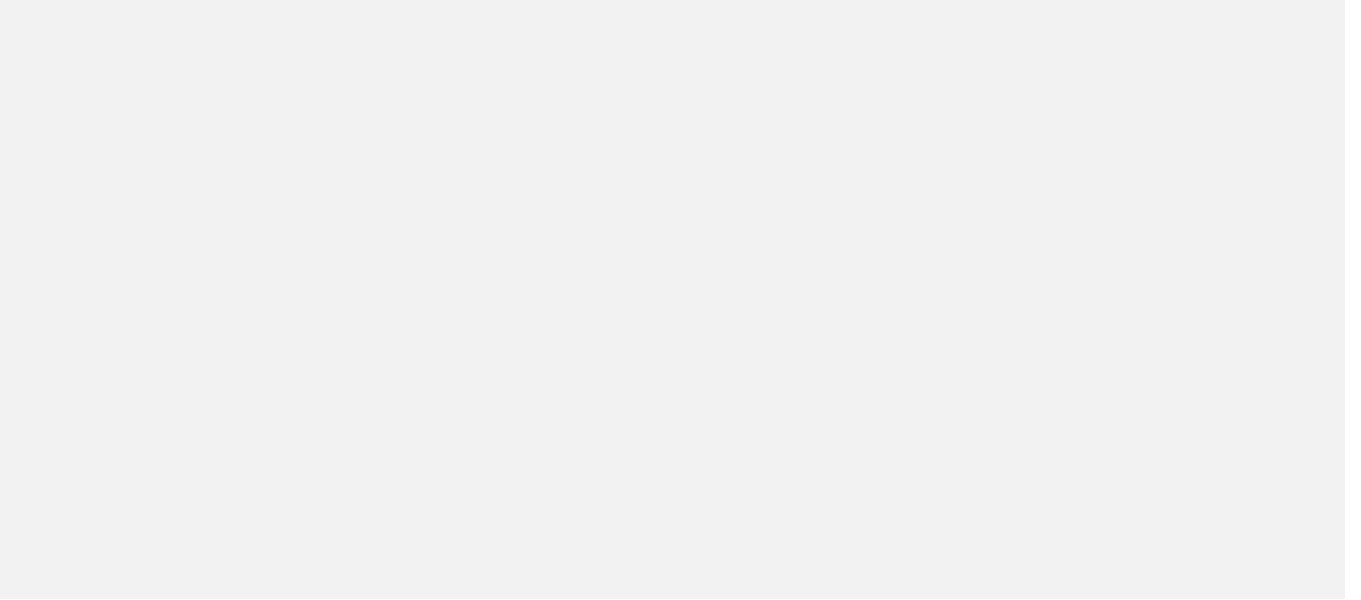
At the mean time, I turned on my interceptor and the request was like this...

```
{“names”:[“projects/agent-nameas-nqsq/locations/us-central1/agents/3fbab09c-  
d034-4f35-bb1f-1ab0837e8806/testCases/9a1c14e8-3dee-43a6-8e58-  
10d4a1fb8a70”]}
```

It have two different id's one agent id and the testCases id. It is hard to guess and hard to do bruteforcing. So, I thought, this will be a dead end.

Suddenly my spidey sense started tingling. Just kidding. Lol

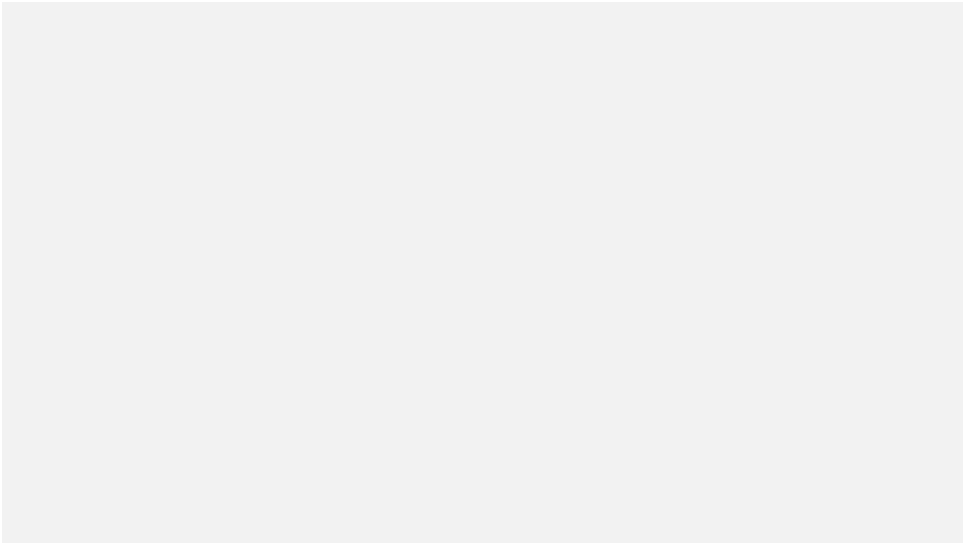
I start thinking, why wouldn't you create a same prebuild agent in the victims account?. And I created another account and imported the same prebuild agent and started testing?



So In the victim account, I deleted the same test case and intercept the request. I saw the same test case id which is inside the attacker account. I was like wow, this is something interesting.

I just sent the both request to burp comparer and looked into what are the main differences in both request?

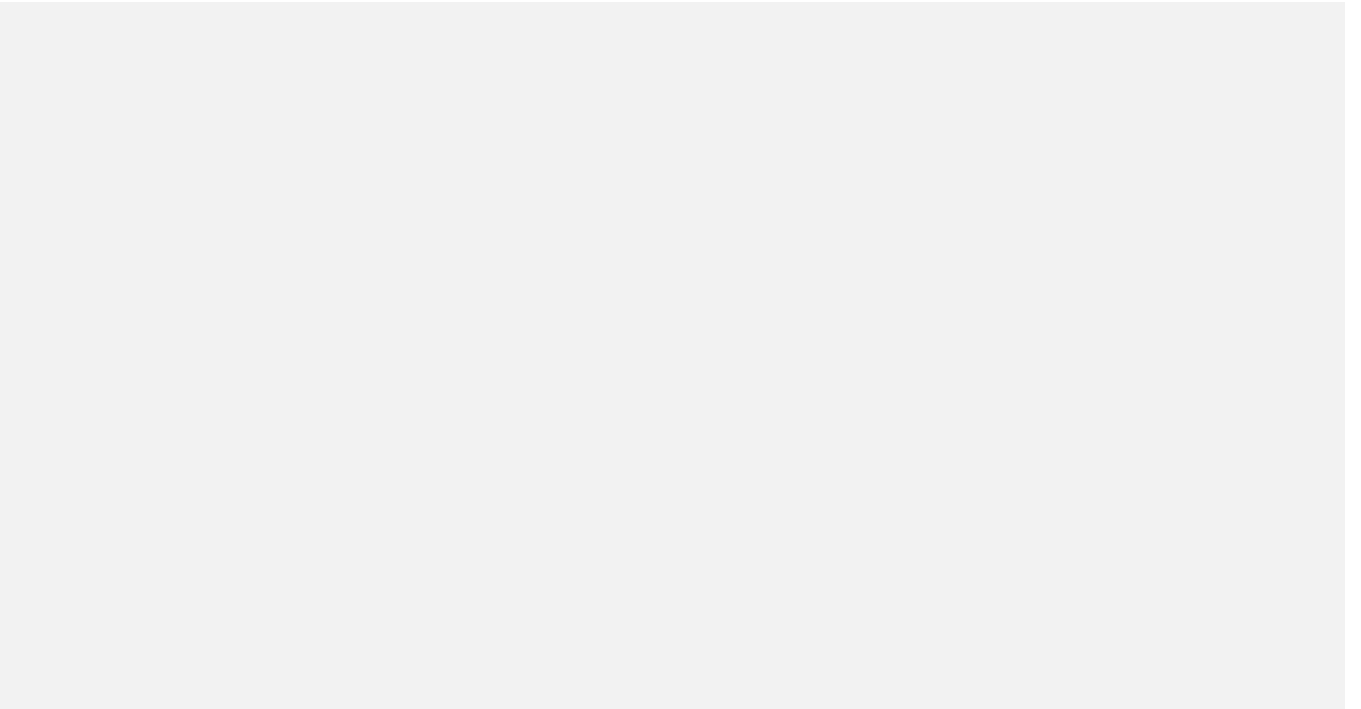
It's the **agents/3fbab09c-d034-4f35-bb1f-1ab0837e8806** id that's the main difference in the both requests and I just replace the attacker agent id to victims agent id and It successfully deleted the test case in the victim account. but it is not guessable and no possibility to bruteforce, It might be more complicated.



After thinking a lot, I just started dorking google. But it didn't also work and started searching in Dialogflow Essentials for any possibility.



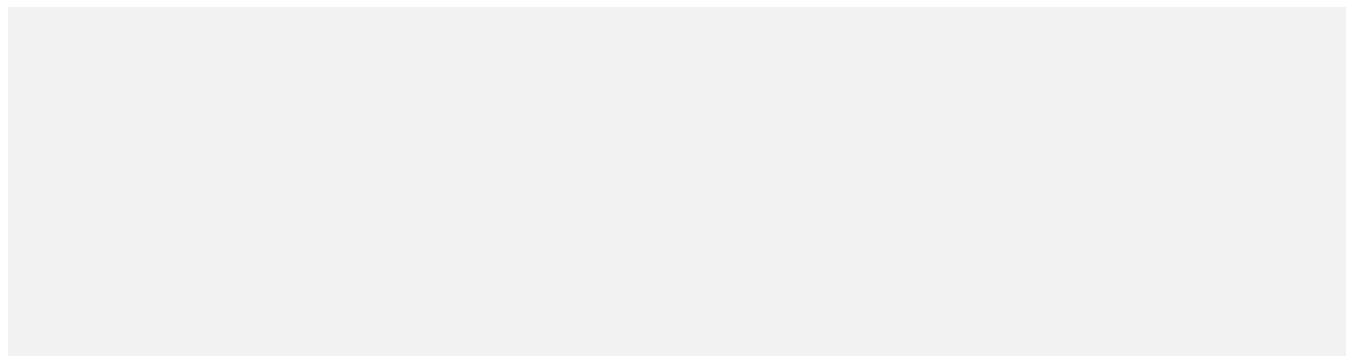
Finally I found something yeeeeee!. To integrate the agent in any website you should need to paste this JavaScript Code.



Or you can share the agent link by enabling web demo.  
<https://bot.dialogflow.com/be29c5bc-b4b7-4d24-b5ba-ec3d3cbe4ba1>

That's the main way to get the agent id. Also we can try different combination of agent id's

Finally Job Done! its time to report the issue. The real impact is, we can select all test cases and able to delete with one click by changing the agent id.



And Got the Reward **\$3133.70**

## TIMELINE

May 27, 2021 12:50AM — Reported

May 27, 2021 04:15AM — Assigned

May 27, 2021 06:45PM — Accepted

Jun 17, 2021 12:20PM — Rewarded \$3133.70

Jul 5, 2021 08:45AM – Fixed

Infosec

Bug Bounty

Cybersecurity

Hacking

Ethical Hacking