Bug Bounty: Broken API Authorization

Omkar Bhagwat (th3_hidd3n_mist)
Nov 12 · 3 min read

Hey everyone, I'd like to share how I found a simple API authorization bug in a private program, which affected thousands of sub-domains and allowed me to exploit a plethora of unprotected functionality without user interaction, from account deletion to takeovers and leaking limited information(Full name, e-mails ids and employer).

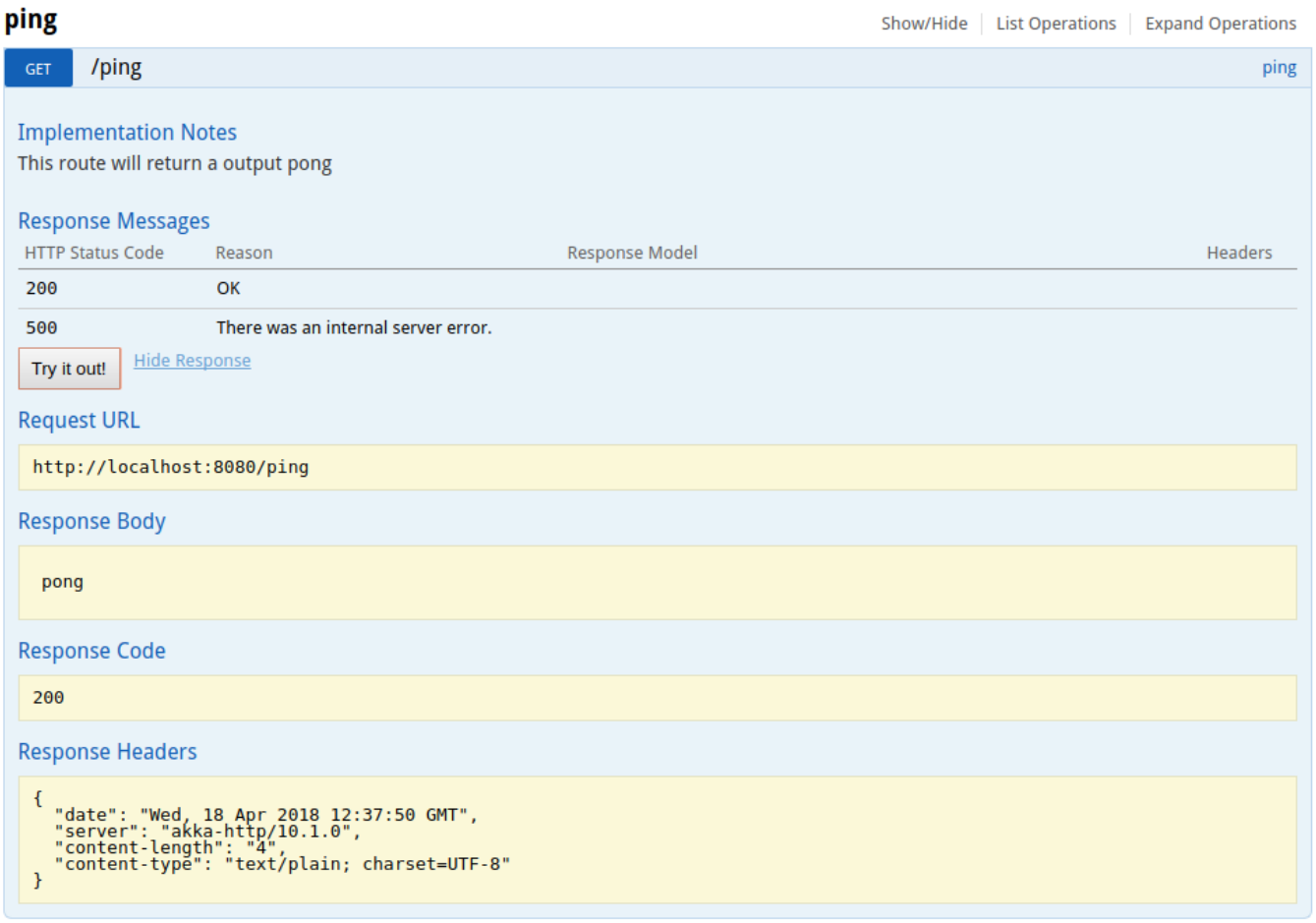
Tl;dr: The server wasn't checking if the authorization bearer token belonged to a regular user or a poweruser.

It's a private program, so some information will be redacted and I'll refer to the site as "target.com".

I had a *dirsearch* scan running in the background while skimming through **academy.target.com**, to get an overview of the sites functionality.
I noticed an interesting endpoint like: **academy.target.com/api/docs**
Endpoints like these are a goldmine because they have API documentation and specify the structure of requests and responses.

On browsing to the endpoint, I found the page to be extremely similar to Swagger UI (this site didn't use swagger though). It also had a button simply called "Authenticate", and clicking on it navigated to a login page but it threw a "Account not authorized" message, if I tried logging in.

There were some interesting endpoints like:
/poweruser/add
/poweruser/delete



The page kinda looked like this.

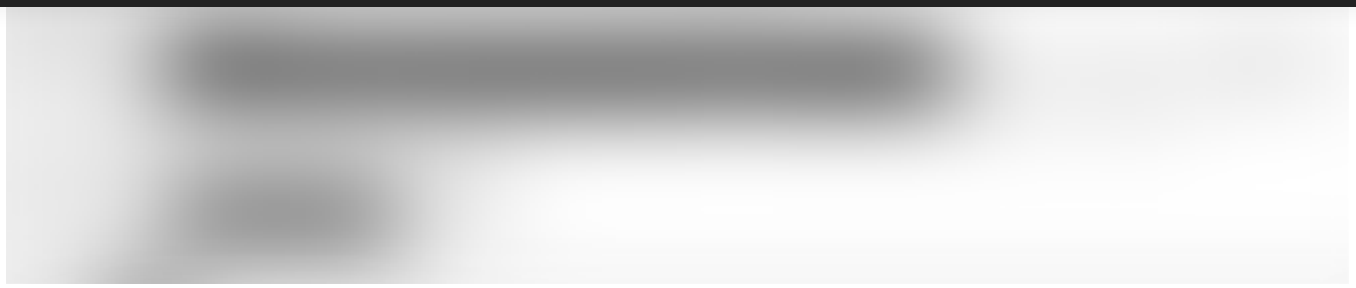
This caught me off guard because it seemed like these endpoints should be reserved for internal/power users use only.
Directly calling the endpoints without any API token or authorization header resulted in:

An unsurprisingly disappointing response

However, I noticed many requests had an authorization bearer token.

I decided to simply copy the header and include it in the calls to the API endpoints I found.
I created another account and tried to change its password, with a **POST** request to **api/user/edit**.

HTTP request to change another users password, this time **with the bearer token**.



Successful response lol.

Voilà! It worked like a charm. Apart from escalating my account to a power-user, I could successfully invoke almost all the other API endpoints.
The documentation detailed the parameters I needed to delete/take over/create new accounts and do some other bad things.

I decided to report the vulnerability directly to the vendor and it turned out they had a private bug bounty program and awarded me a $440 bounty.

Thanks for reading!

. . .

Follow Infosec Write-ups for more such awesome write-ups.

API Bug Bounty Cybersecurity Authorization Bounty Program

New bug bounty hunter, old gamer and anime fan.

Follow

InfoSec Write-ups

A collection of write-ups from the best hackers in the world on topics ranging from bug bounties and CTFs to vulnhub machines, hardware challenges and real life encounters. In a nutshell, we are the largest InfoSec publication on Medium. Maintained by Hackrew

Follow

See responses (1)