



HACKTHEBOX

🎃 HackTheBoo 2024 - Hybrid Unifier 🎃

Hello CTF player 😊 !

This is the "hybrid unifier" challenge. I have created a flask API for you to alpha test.

You do not need to be concerned about privacy as your session is secured with End-to-End (E2E) encryption in both server and client sides.

As an alpha tester, I don't want to make your life harder so here is some documentation of how to use the API ... and who knows? If your testing is indeed Alpha and you find no bugs, there is a coveted flag for you as a gift 😊.

High Level Protocol Description

For E2E encryption, a shared session key is established via the Diffie Hellman Key Exchange Protocol between you (the client) and the server.

This session key is used as an AES symmetric key for encrypting the rest of the communication.

Endpoints

For the following endpoints, only the `POST` method is allowed.

- `/api/request-session-parameters`

You can use this endpoint to obtain the Diffie Hellman parameters. You will need them for initializing the secure session.

- `/api/init-session`

You can use this endpoint to establish a secure session with the server. You receive the server's Diffie Hellman public key and you send the server yours.

- `/api/request-challenge`

You can use this endpoint to request an encrypted challenge from the server. It is encrypted with the shared session key. This challenge is required for authentication to interact with the `/api/dashboard` endpoint.

- `/api/dashboard`

You can use this endpoint to send the action you want to the server. The available actions at the moment are `flag` and `about`. Keep in mind that you need to send an encrypted packet as the communication is E2E encrypted.

The `/api/request-challenge` and `/api/dashboard` endpoints can be accessed only after a secure session is established. For this, you need to first interact with the `/api/init-session` endpoint.

Interacting with the API programmatically

To interact with the API endpoints programmatically, we recommend using the `requests` Python library. You can send json data in a POST request as shown below:

```
URL = 'http://localhost:1337'
r = requests.post(f'{URL}/a-cool-endpoint', json={'key1': 'json', 'key2': 'data'})
```

To read the response content, you can use the `content` attribute:

```
print(r.content)
```

Good luck with your testing and I hope you come back with a leet flag 🚩!