I discovered something fascinating about Discord recently.

Firstly, I created a Python script to send messages rapidly on Discord using different usernames. This was for experimenting on a channel in my personal server, a test space. As seen in the attached image, the result was impressive. However, I later modified the script to send 100 messages consecutively. That's when trouble started: Discord quickly detected abnormal activity and reacted by blocking my script with the error "rate limited, retry after 0.3 seconds" (the delay could vary). It was a smart measure by Discord's creators to prevent spam.

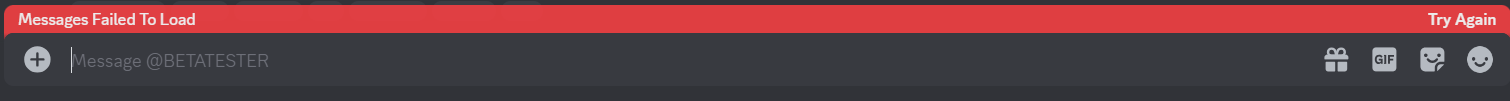
After adjusting my code to respect this delay between each message, the script worked initially. However, as the 100 messages were sent, the slowdown delay accumulated, significantly reducing the execution speed towards the end. I then modified the code to generate random usernames for each message, but this introduced a new issue. Confused, I reverted to the previous version of the code, yet the problem persisted with an accumulation of errors in the Python console: "Rate limited. Retrying after 0.5 seconds." Unfortunately, the script couldn't resume sending messages every 0.5 seconds as planned, likely because the delays had accumulated to several minutes due to my failure to stop the program immediately.

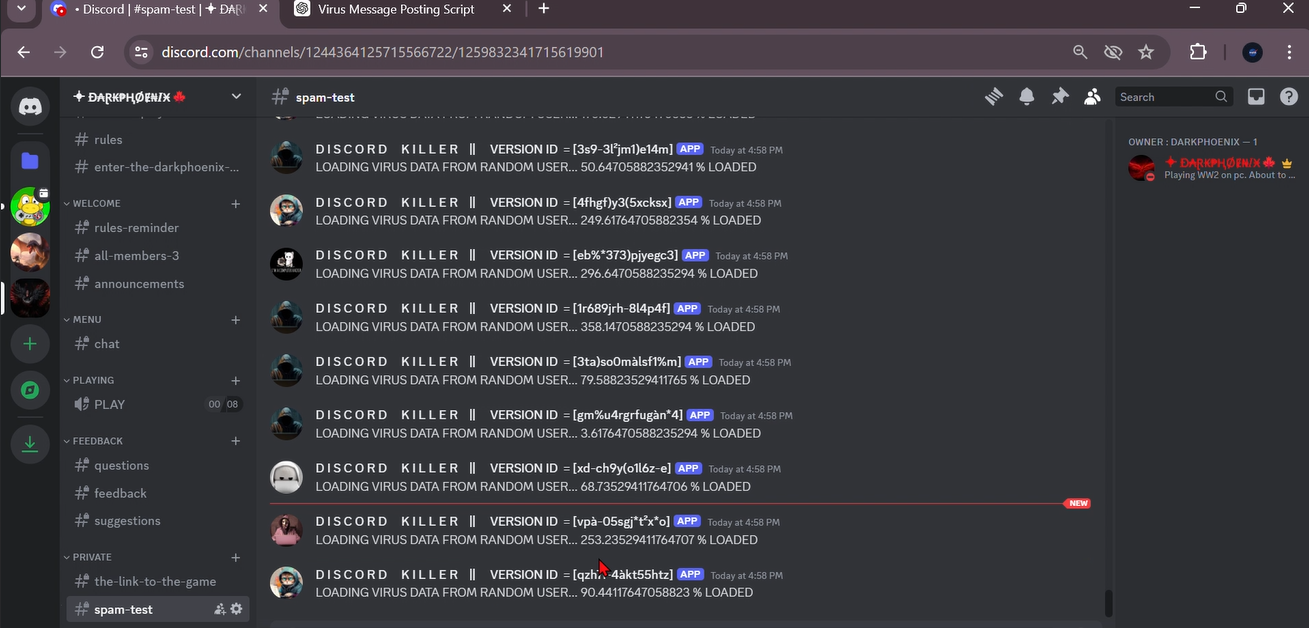
Initially thinking it was a Discord bug, I refreshed the page, but it made no difference. Messages wouldn't load in any chats, and after a few minutes, I realized I couldn't interact with Discord at all: I couldn't change my username, send messages, adjust server channel settings, or even view old messages. I felt disconnected from the network, though all other websites were functioning normally. Was I banned from Discord without warning or notification? It seemed unlikely, yet I was perplexed.

After about 15 minutes of troubleshooting, everything miraculously returned to normal. I reran my script, and this time, it worked as expected, sending the 100 messages without any issues. That's when I understood what had happened: Discord had paused all my activities due to accumulated delays and their security measures. Launching the program multiple times likely triggered this security reaction.

To test this theory, I ran the script one last time, and as expected, the console displayed the delay timeout errors again. I immediately stopped the code to prevent further delay accumulation. In conclusion, my script temporarily blocked my Discord account due to these accumulated delays.

The lesson learned is to reduce the number of messages sent at once. Decreasing from 100 to 10 or 5 messages can prevent these issues and preserve the integrity of your Discord account.





import aiohttp

import asyncio

import random

import time

class Setup:

    def \_\_init\_\_(self) -> None:

        self.chars = 'abcdefghijklmnopqrstuvwxyz134567890%ù²(-à)\*'

        self.webhook\_url = 'https://discord.com/api/webhooks/1259832964989325352/Krej5cxAF4AKxrP1GAHX4Lir46ITGZm7uVwRUD-rZR1kptKZbN6fre5Kix4yiUsQvYXz'

        self.usernames = []

        self.contents = []

        self.avatar\_urls = []

        self.generate\_usernames\_list()

        self.generate\_contents\_list()

        self.generate\_avatar\_urls\_list()

    def generate\_usernames\_list(self):

        for \_ in range(100):

            username = ''.join(random.choice(self.chars) for \_ in range(15))

            username = 'D I S C O R D     K I L L E R    ||     VERSION ID  = [' + str(username) + ']'

            self.usernames.append(username)

    def generate\_contents\_list(self):

        for i in range(100):

            content = "LOADING VIRUS DATA FROM RANDOM USER..." + " " + str(i \* (123 / 34)) + " % LOADED"  # random factor

            self.contents.append(content)

    def generate\_avatar\_urls\_list(self):

        any\_avatar\_url = "https://www.radiofrance.fr/s3/cruiser-production/2020/03/7495829e-7534-4ada-8632-7ee75844e531/870x489\_gettyimages-513088279.jpg"

        self.avatar\_urls.append(any\_avatar\_url)

        any\_avatar\_url = "https://media.teachprivacy.com/wp-content/uploads/2023/07/28143141/Hacker-Funny-AdobeStock\_574954979-scaled.jpeg"

        self.avatar\_urls.append(any\_avatar\_url)

        any\_avatar\_url = "https://as1.ftcdn.net/v2/jpg/05/80/66/52/1000\_F\_580665262\_SbO69iupamlna60JTP3QEUn2I1S9Myi3.jpg"

        self.avatar\_urls.append(any\_avatar\_url)

        any\_avatar\_url = "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTOSbhqsghVrKH5620pPh1wvOw7Fngy4TXi-w&s"

        self.avatar\_urls.append(any\_avatar\_url)

        any\_avatar\_url = "https://t3.ftcdn.net/jpg/05/74/95/50/360\_F\_574955028\_Y2qisbczKSUUAMkvC4eAIomVPWga22Ix.jpg"

        self.avatar\_urls.append(any\_avatar\_url)

        any\_avatar\_url = "https://nerdshizzle.com/wp-content/uploads/2021/01/im-a-hacker.jpg"

        self.avatar\_urls.append(any\_avatar\_url)

    def create\_random\_payload(self):

        payload = {

            "username": self.usernames[random.randint(0, len(self.usernames) - 1)],

            "content": self.contents[random.randint(0, len(self.contents) - 1)],

            "avatar\_url": self.avatar\_urls[random.randint(0, len(self.avatar\_urls) - 1)]

        }

        return payload, self.webhook\_url

class Sender:

    def \_\_init\_\_(self, webhook\_url, payload):

        self.webhook\_url = webhook\_url

        self.payload = payload

    async def send\_webhook\_message(self):

        async with aiohttp.ClientSession() as session:

            while True:

                async with session.post(self.webhook\_url, json=self.payload) as response:

                    if response.status == 204:

                        print(f"MESSAGE = {self.payload['content']} || SENT SUCCESSFULLY.")

                        return

                    elif response.status == 429:  # Rate limited

                        data = await response.json()

                        retry\_after = data.get("retry\_after", 0.5)

                        print(f"Rate limited. Retrying after {retry\_after} seconds.")

                        await asyncio.sleep(retry\_after)

                    else:

                        print(f"Failed to send message. Status code: {response.status}")

                        print(await response.text())

                        return

async def main():

    keep\_data = False

    number\_of\_messages = 3

    setup = Setup()

    tasks = []

    if not keep\_data:

        for \_ in range(number\_of\_messages):

            payload, webhook = setup.create\_random\_payload()

            sender = Sender(webhook, payload)

            tasks.append(sender.send\_webhook\_message())

    else:

        payload, webhook = setup.create\_random\_payload()

        sender = Sender(webhook, payload)

        for \_ in range(number\_of\_messages):

            tasks.append(sender.send\_webhook\_message())

    await asyncio.gather(\*tasks)

if \_\_name\_\_ == "\_\_main\_\_":

    asyncio.run(main())