



Discord

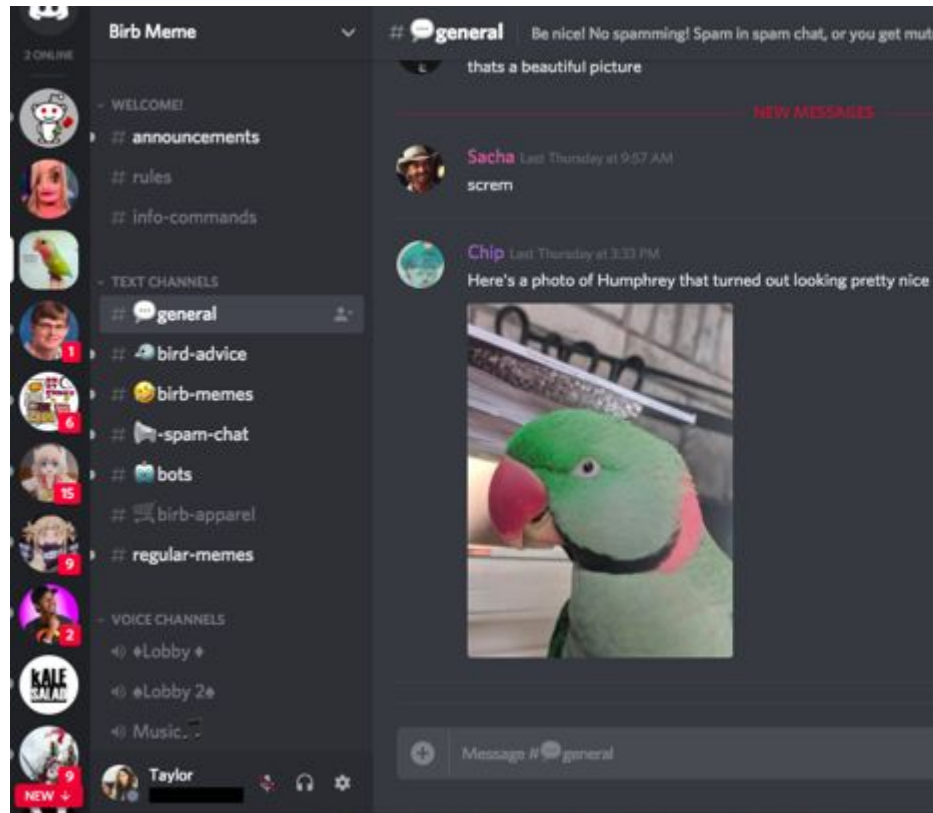
<https://discord.gg>

Meeting flag goes on board hereabouts

Discord Intro

- Text and voice chat
- Social media

- Comprehensive API
- Electron & Websockets



Challenges: 3 Categories

Bots & API

Recon

Social Engineering

Bots & API

- Bots & API: Mistakes bot creators make, Discord API
 - Bot commands and implementation
 - Discord API wrappers (discord.py) and docs
 - Webhooks

 - Electron Inspector (Control-Shift-I)
 - Websockets
 - Minified JS

Recon

- Recon: Hide and seek!
 - Where can we hide a flag in a Discord server?
 - Check every corner, click every button, inspect every asset
 - All solvable without writing code
 - Only one challenge, Recon 7, involves the bot

Social Engineering

- Social Engineering: How to corroborate lies
 - Assumptions about features
 - Impersonation and information leakage
 - Bot use and abuse

More Challenge Info

- Self-botting: NOT REQUIRED
 - Against Discord ToS
 - Not immediately bannable, but...
 - Very lax and case-dependent enforcement, RAI
- Developer Mode
 - Copy ID's (also called snowflakes) from UI
 - Settings -> Appearance -> Advanced -> Developer Mode

More Challenge Info

- Platform
 - All challenges possible with Desktop client
 - Almost all possible with Web, nearly all with Web
 - Mobile (eg, an iPad with the Discord app), good luck...
- **Important: Flag format is different**
 - Alphanumeric with underscores only
 - Example: `flag_discord_challenge_info`

now do them.

Bots & API 1: Source

- Bot prefix is !
 - Bot prefixes are not standard
 - Many different libraries for wrapping the same API
- Try !help, then try !source
 - Many Discord bots are open-source.
 - Much easier to find vulnerabilities
- Servers are called guilds internally
 - No guild -> Message is a DM (Direct Message)
- Solution: DM the bot “flag pls”

Social Engineering 1: Half-Life 3 Confirmed

- Desktop only
- “Now Playing”
- Any application can be added as a “game” and renamed
- User Settings -> Game Activity -> Add it! -> “Half-Life 3”

Bots & API 2: Auto Delete

- Sends the flag and then immediately deletes it
- Open developer console using Control-Shift-I (i, not L)
- Same developer tools as Chrome
- Discord UI is just a web page

- Create DOM breakpoint (demo live)
- Pre-load the source so it's less likely to crash

Social Engineering 2: Impersonation 1

- You have Manage Roles permission
- Discord's permissions groups
- Each role has permissions, a color, can be hoisted
- Each user can have multiple roles

- Permissions add (except when they don't)
- Color indicates role hierarchy, but...

Social Engineering 2: Impersonation 1 continued

- Colors only indicate the highest role someone has
- Multiple roles can have the same name
- Multiple roles can have the same color

- Solution: Make an Admin role that's the same color.
- Get the color from the dev console, or a screenshot

Bots & API 3: Webhooks 1

- Webhooks are URL's that go directly to Discord
- Manage Webhooks channel permission (show live)
- POSTing information to the URL sends a message
- No authentication besides the ID and token in the URL
- Very easy to use compared to a bot
- No hosting required in some cases

Bots & API 3: Webhooks 1

- <https://discordapp.com/developers/docs/resources/webhook>
- Solution: Visit the webhook URL in a browser.

Webhook Object

Used to represent a webhook.

Webhook Structure

FIELD	TYPE	DESCRIPTION
id	snowflake	the id of the webhook
guild_id?	snowflake	the guild id this webhook is for
channel_id	snowflake	the channel id this webhook is for
user?	user object	the user this webhook was created by (not re
name	?string	the default name of the webhook
avatar	?string	the default avatar of the webhook
token	string	the secure token of the webhook

Example Webhook

```
...fecd1726de135cbe28a41f8b2f777c372
```

```
...ee11f3"
```

```
...  
...  
...}
```

Get Webhook with Token

```
GET /webhooks/{webhook.id}/{webhook.token}
```

Same as above, except this call does not require authentication and returns no user in the webhook object.

Social Engineering 3: Impersonation 2

- Webhooks continued!
- Default username and avatar
- Defaults can be overwritten!
- JSON format:


```
{  
  "content": "Hello world!",  
  "username": "kuilin"  
}
```

EDIT WEBHOOK

NAME

CHANNEL

WEBHOOK ICON
We recommend an image of at least 256x256



Minimum Size: 128x128

WEBHOOK URL

[Need help with setup?](#)

Bots & API 4: Webhooks 2

- ID-based mentioning
- Messages are 100% text on Discord, with markup
- Inline “objects” (called mentions) via snowflake numbers

- #sigpwny-ctf == <#509273019701657610>
- @kuilin == <@168176809152610304>

- Can you think of two more?

Bots & API 4: Webhooks 2

- Channel ID exposed in the webhook info
- Can mention anywhere - even in a DM or unrelated server
- Parsing is done client-side
- Implications??
- Solution: Send `<# channel ID from webhook >` anywhere

Social Engineering 8: Weak Link

- Skipped the rest of the Impersonation sequence, explore on your own
- This server has a “user” that clicks on everything
- If you send a link in the chat, they will click on it
- Use that to expose their Useragent and IP address
- You don't need a web server: Grabify: <http://grabify.link>
- Solution: Send a link you control, look at server logs

Bots & API 5: Webhooks 3

- Webhook messages look like bot user messages, and can run commands sometimes!
- Bots can see if a message's author is a bot or not
- Different permissions than a user using the bot
- !say is exceptionally vulnerable...
 - !say /promote @kuilin
 - !say @everyone

Bots & API 5: Webhooks 3

- Solution: Use the webhook to send a !sendflagto
- Mention yourself using <@ your ID >

```
{  
  "content": "!sendflagto <@168176809152610304>"  
}
```
- Most very popular bots (Tatsumaki, MEE6) disallow this

Recon 1-5

- Spoilers! Not really much to explain

Recon 1-5 Solutions

- Spoilers! Not really much to explain
- Recon 1: Bot's real name. View bot profile to see it.
 - Real Discord names are independent of server, nicknames are server-specific
 - Server mods can change your nickname, but not your real name
- Recon 2: Bot's profile picture, hidden in the corner
 - Use Inspect Element (or any web proxy... Fiddler?) to view image
 - Images are square, but Discord crops them to circles
- Recon 3: Server emoji names can have information
- Recon 4: Pingable role, push @ to bring up the menu
- Recon 5: Server Settings is viewable to anyone with ≥ 1 Manage permission
 - Even something completely unrelated like Manage Emojis

Advanced tricks

- Super hard: Bots & API 9-10
 - Reverse engineering... minified Javascript!
 - Using the Inspector
 - Don't be afraid to Google weird syntax
- Super hard: Social Engineering 10
 - No further hints on this one :P
- Others: Ask me for hints on the Discord server