

Forum: [Hack Forums](#)

Thread: [Bypassing Windows 11 \(fully patched\)](#)

jfk0699 - December 17, 2025, 11:32 am

OK so I have written a very simple and basic C program (console app) that makes use of strcpy routine and want to exploit it by feeding it more data by way of the command line. Here is the program below and what I want is for it to execute a payload that opens a hello world message box or better opens up calc.exe.

Unfortunately I've tried passing more 1000 bytes of input characters at the command line but the program just gracefully shuts down no errors no messages no nothing. I'm using Visual Studio 2019 Community as my compiler. I understand Windows 11 makes use of several security techniques such as DEP, ASLR, Stack Canaries and etc. I want to be able to bypass these measures and execute my payload. The payload works as I've tested it that is the message box and calc.exe both open and display inside a separate C program but do not open via a stack overflow.

Any pointers would be greatly appreciated.

```
Code int main(int argc, char** argv)
{
    int age;

    char buffer[500];

    strcpy(buffer, argv[1]);
    printf("strcpy gives %s\n", buffer);

    while (1)
    {
        printf("Enter your age: ");
        scanf_s("%d", &age); // No size parameter needed for integers
        printf("You are %d years old.\n", age);
    }
    return 0;
}
```

Terry A. Davis - December 17, 2025, 5:54 pm

Well because that's not quite how this works.

You are on the right path for sure thinking that a stack overflow is the way you want to go, but you're going to need a way to bypass all the things that make code execution upon a stack overflow impossible, which.... I honestly think don't readily exist without a huge pay cheque coming your way, though happy to be shown wrong, I just don't think that's quite how this works in Windows these days.

jfk0699 - December 17, 2025, 6:59 pm

Thank for the reply post. I think also that I'm on the right track. Perhaps a 0 day exploit might be a possible solution. Now if I can find one that does not cost an arm and a leg. I also do know of some forums on the dark web that might be helpful. There has to be readily available 0 day exploits that

will do the trick for my needs.

Thanks again.

jfK0699 - January 12, 2026, 8:47 am

OK so I've decided to take a fresh new approach and disable all security measures namely DEP, ASLR and Stack Canaries when compiling. So now I've run into an insolvable problem that I've spent over 1 week on. That is the encoding of the shell code for bytes such as 0x81 does not have a defined mapping under Windows-1252 or UTF-8 encoding schemes. Any ideas on how to get around this hurdle ?

jfK0699 - January 14, 2026, 1:42 pm

(Jan 12, 2026 09:47 AM)jfK0699 Wrote: [\[CITE\]](#) OK so I've decided to take a fresh new approach and disable all security measures namely DEP, ASLR and Stack Canaries when compiling. So now I've run into an insolvable problem that I've spent over 1 week on. That is the encoding of the shell code for bytes such as 0x81 does not have a defined mapping under Windows-1252 or UTF-8 encoding schemes. Any ideas on how to get around this hurdle ?

OK for anyone interested it involves encoding the undefined bytes using another character map other than Window-1252 where this byte is undefined. So I used latin1 code page