Not even trying to cross an airtight hatchway: Calling a function in your own process by synthesizing a function pointer

devblogs.microsoft.com/oldnewthing/20221201-46

December 1, 2022

Raymond Chen

A security vulnerability report arrived that went roughly like this:

> I have found a security vulnerability in the `CONTOSO.DLL` dynamic-link library. [Long description of methodology omitted, including discussion of dead ends and failed attempts. The short version is "I looked for code that calls `printf` with a format string that is generated at runtime rather than a hard-coded string. That code is subject to a format string attack."]
>
> Attached is a proof of concept.
>
> ```
> int main()
> {
>     // vulnerable function is at offset 0x12345
>     auto p = (LPBYTE)LoadLibrary("contoso.dll") + 0x12345;
>     auto fn = (void(*)(char const*, int))p;
>
>     // Call the function with a %n format string
>     fn("%n", 42);
> }
> ```
>
> I am requesting a bounty for this report.
>
> Note that this is the first security vulnerability I have found and submitted. I acknowledge that my understanding is incomplete. Please provide additional advice and assistance to help me become a better security researcher. I look forward to your reply.

This is like calling the natural gas utility company's emergency number to report a major gas leak in your house. The gas company sends a technician over, and they can't find any leak. They ask how you came to suspect that there's a gas leak, and you tell them, "Oh, I didn't smell anything.[1] I called you because I'm hoping to learn more about how to recognize the smell of gas. Do you have any tips?"

Yes, the tip is that if you don’t know how to recognize the smell of gas, you can use existing educational materials to learn how to recognize the smell of gas. Don’t call the emergency line to learn what gas smells like.[2] The emergency line is not intended to be used as a source of training data. There are other places to learn more about the smell of gas.

In this case, no security boundary has been crossed. The “vulnerable” code is loaded into the attacker’s process, and the attacker is calling it directly. Attackers who want to attack their own processes don’t need the help of `contoso.dll`.

For example, they could have gone directly to the C runtime library.

```
int main()
{
    // vulnerable function is called "printf"
    auto p = GetProcAddress(LoadLibrary("ucrtbase.dll"), "printf");
    auto fn = (void(*)(char const*, ...))p;

    // Call the function with a %n format string
    fn("%n", 42);
}
```

which simplifies to

```
int main()
{
    // Call printf with a %n format string
    printf("%n", 42);
}
```

The internal function they found in `contoso.dll` is a passthrough to `printf`. It is called only with known format strings which match the rest of the `printf` parameters. The string is not hard-coded because the format string is looked up at runtime to match the user’s preferred language. There is no way to get this DLL to pass an untrusted format string to `printf`, at least not through the function under attack.

Besides, if you are interested in doing dangerous things by calling functions in a way that cannot be externally triggered, then `printf` is a particularly complicated to do it. Much easier is to find a function that ends with something like

```
    mov     qword ptr [rdx], rcx
    ... other instructions that you can stage mitigations for[3] ...
    ret
```

Put the desired value in `ecx` and the desired target address in `edx`, and call that function! No need to drag `printf` into it.

And if you’re still learning about searching for security vulnerabilities, please don’t send in reports until you’ve learned the part about exploitability. Thanks.

[1] Yes, natural gas is odorless. The smell is added by gas companies.

[2] It is common for parents at my children's Chinese-language school to socialize in the cafeteria while the students are attending their lessons. There was a time a few years ago where one of the parents thought they smelled gas. They asked others to check it out, and opinions were mixed. Some people agreed that they smelled gas, but others thought it was something else. Eventually, the source of the odor was identified: Somebody had brought durian fruit as a snack and was eating it in the cafeteria.

[3] For example, I found this sequence:

```
    mov     dword ptr [rdx],ecx
    mov     rbx,qword ptr [rsp+88h]
    mov     eax,ebp
    add     rsp,40h
    pop     r15
    pop     r14
    pop     r13
    pop     r12
    pop     rdi
    pop     rsi
    pop     rbp
    ret
```

You can stage a call to this by pre-pushing the registers and return address onto the stack and pre-subtracting `40h` from `rsp`, then calling the function.

Hey look, in the same DLL, I found this instruction sequence:

```
    mov     dword ptr [rax],ecx
    ret
```

That will work great.

Raymond Chen

Follow