Shellcode & Process Injection

16-20 NOV 20

Tasking

I was tasked with getting exposure to writing and deploying shellcode/PIC.

PlayingWShellcode

First, I created a project that could successfully run arbitrary shellcode. Of course, this included VirtualAllocing the memory, copying in the code, VirtualProtecting it for execution, and calling the function. I spent a good amount of time doing this properly and handling any errors with *excessive* verbosity. As such, I was able to create some dummy shellcode samples—and plug-and-play ones from the internet (after vetting, of course)—and debug/repair them so they could be run on my box. Now I needed to deploy it somewhere.

HookTest

I dove into the internals of hooking with a flavor of process injection. I wrote a wrapper for MessageBoxA that would add in my own hardcoded values. Then I patched my current process, overwriting the pointer to the real MessageBoxA with my wrapper, which would transparently call the real MessageBoxA with modified values.

So how far could I take this? I did the same thing to VirtualAlloc and also had the wrapper return the pointer to the allocated memory—I could modify permissions before allocation (and then arbitrarily write code to blocks I see it intends to execute) or whatever I wanted and the process would keep chugging along. But this is only really cool if I can hook a *remote* process's call to VirtualAlloc. And this is where I hit a wall. I could successfully load in kernel32.dll, overwrite the VirtualAlloc address, and have the code executed…or attempted. I have some ideas on what this is, mostly ensuring the remote process can access everything needed.

TLDR

1. Wrote/Generated/Loaded arbitrary shellcode
2. Manually hooked function calls
3. Extended to patch remote processes
4. Remote processes are currently failing to execute my code, fixes in mind

Future Plans

- Spend more time on this (if allocated)
    - Fix remote process injection
    - After finishing CryptoAPI intro (30 NOV-4 DEC), perhaps hook crypto calls to steal keys…
- Create library for performing this injection
    - Integrate known-good open-source tools for streamlining