I am a windows system-level & low-level developer.

I am programming in both kernel and user modes.

I have a very deep knowledge of Windows internals and undocumented features, and I have a vast experience of working with both. Such knowledge gives me a great freedom of action. Windows for me is not a black box. I can find creative, effective and smart solutions. I have very deep knowledge and vast experience in asynchronous programming, multithreading, synchronization, managing object's lifetime and access (reference counting, rundown protection.)

In kernel mode I wrote generic/separate (non Pnp) drivers, WDM (Pnp) drivers (virtual bus driver (FDO) and filter drivers (FiDO)) primary for storage, for the file system and input (keyboard/mouse) stack, legacy filters and minifilters, filtering registry, calls and objects operations, process, threads, image notifications. I have an in-depth knowledge of kernel development, specifically kernel <-> user mode communication, I know all aspects of IRP processing, kernel objects, memory dump analysis and remote/live debugging, kernel networking (over TDI interface), DPC, APC and much more.

In user mode I have an excellent knowledge of:

- WIN API (including so called 'native' API),
- processes and threads, DLLs, initialization
- synchronization (of course, this and many other related topics to the kernel mode that I listed above),
- IPC (shared memory, LPC, pipes, mailslots, events, semaphores, mutants,...)
- windows services,
- boot execution apps,
- system registry,
- file systems (especially NTFS (streams, EA, internal structures)),
- memory management,
- Windows cryptography (both legacy and CNG), certificates, etc
- authentication and authorizations,
- credential providers,
- security support providers/authentication packages

- security (tokens, security descriptors, labels, integrity levels),
- networking – I wrote code for client and high-load servers too (based on IOCP (KQUEUE object in kernel)),
- client <-> server communications,
- COM,
- RPC,
- Windows shell, shell extensions
- GUI,
- exception handling (SEH/VEH),
- WIN API and interface hooking,
- I have a perfect knowledge of PE format and of PDB format,

and much more than I can list here.

I have a vast experience with debugging (I have my own toolkits for this, including my own private debugger),

I have experience in reverse engineering (however I prefer debugging and analyzing code under a live debugger than working with a static code analyzer like IDA Pro).

I have the ability to research and discover “why” some WinAPIs "fail",

I can effectively debug cross-process calls,

I can debug system processes, including protected ones (yes, my own debugger can do this),

I can view kernel memory and objects at run-time,

and much more.

My main working language is C++ (CL.EXE compiler(in MSVS)). I use x86/x64 assembler when needed, and I know both quite well. I have a deep knowledge of compiling/linking process and can resolve related issues, like undefined/unresolved symbols, name mangling, calling conversions, etc. I use SDK and WDK. I use IDL when needed for RPC and COM interfaces, or for communications with JavaScript from the C++ code (I can implement IDispatch by typelib help). I have a great experience in this. I also have some basic knowledge of JavaScript and HTML.

I am always focused on code quality and its effectiveness, and have very high motivation.

Some (very partial) samples of my code can be found at: https://github.com/rbmm/

Some projects that I was involved in:

• professional debugger and system tools (in SysInternals style)

• virtual, encrypted, usbstor disk (WDM interface, full PNP)

• virtual smart card reader and smart card implementation (Identity Device (Microsoft Generic Profile) (WDM, PNP)

• MFA credential provider (for protectimus)

• Implementation of smart card (certificate) logon on workstations

• Windows logon with virtual smart cards

• injecting dll from a kernel driver into user mode processes (including with CIG policy, despite dll not signed)

• class library for asynchronous I/O

• class library for UI (like small MFC/ATL classes, but not less functional)

• work on special, high payload, windows servers

• replace windows start button image and system menu (part of StartMenuX project)

...

Since summer of 2016 I have an account on stackoverflow.com (posting basically in [winapi] tag) - http://stackoverflow.com/users/6401656/rbmm?tab=profile