

Wanderley Caloni

- Location: São Paulo, Brazil
- Email: wanderley.caloni@gmail.com
- Stack: C/C++ (26 yrs), Windows (26 yrs), Debugging (23 yrs), Reverse Engineering (15 yrs), Tech/Art Writing and Knowledge Sharing (15 yrs), Linux (10 yrs).
- Languages: Python, C#, Assembly, PHP, Java, Golang, Pascal.
- Tools: Visual Studio, WinDbg, Git, SSH+Vim.

Work

Sr. Developer at BitForge (2015-current)

- ATM. The old solution was slow and demanded frequent physical support to the ATMs. As the main target of the project was security and resilience, I developed and maintained a new system with a low level protocol using SSL-based protocol written in C++ (Boost.Asio) to allow packet exchange and support screen share even in adverse network bandwidth scenarios, by example, in the middle of Amazonian Forest. The outcome was the exchange of the 20 years old solution, what decreased the frequency of physical support at the ATMs (20k+).
- Firmware. The client needed to support a RTOS solution using a non RTOS in the application layer (Windows). I developed and sustained real time communication between a in house firmware and the Operating System running managed code and Web API by keeping a thin communication layer in a SDK written in C++ that supported real time response independent

from the managed code response. The solution was sold and delivered for the final client.

Tech Lead at **InteliTrader** (2019-2023)

- **InteliMarket**. The main goal was to keep high performance market data traffic, even with low computational resources. I kept the system running in a uptime at 99% and up to date to B3 UMDF protocol last spec. The bug response time daily was about 5 minutes because I organize a time schedule in the team, including backups, and a monitoring system that alerted the team every time an occurrence could be happening, before the final customer even noticed. The outcome was that the reliability of the solution increased and InteliMarket doubled the number of clients subscribing to the solution. I also reduced subscription delays in 50% by removing a core but unnecessary event queue. The entire solution was a mix of C sockets, C++ callbacks and Boost.Asio supporting the multicast protocol.
- **MetaTrader**. Improved order routing performance by 200%+ by keeping a secondary worker thread to persist orders in the database, becoming success case in the local market, being recommended by MetaQuotes in the media to its Brazilian clients. Even without a reliable test environment the solution was proved resilient by writing a massive quantity of unit tests to cover all the cases where the final customer reported a behavior error. With that system the bug occurrences in production fell from 6 to only 1 per year. The entire system was developed in C++ with Boost.

Risk Developer at **EzMarket** (2012-2015)

- EMS. As a experienced trader I was selected to develop the entire Risk System for the order routing system by transforming the client mathematics formulae to high performance and reliable solution in C++ to manage the financial risk of the entire broker with 20k+ active accounts by using POCO Libraries. I also improved the channels internal communication system to be completely configured via XML with routing rules defined by message type, independent from any individual component. The outcome was the exchange of the old solution, what increased total trades and broker total revenue.

Sr. Developer at **SCUA** (2008-2012)

- Disk Cryptography. The main goal was to protect data from users and companies using low level cryptography in Hard Drive and Pen Drives (USB). This was a tricky project because not always the cryptography worked on every hardware. I had a small team of 3 people and a lot of machines using our product. Sometimes the system boot crashed and I had to analyze the MBR information and to debug the boot process in real mode (8086 assembly) to save the information for the users. The outcome was that this was one of the best selling products from SCUA because big companies loved the idea of a customizable cryptography solution free from hackers of Microsoft trying to break and with a team ready to fix any problems. For this solution it was used Assembly, WDK, Lilo source code and Linux for the bootable CD of troubleshooting.
- Application and User Control System (SCUA). Updated the Windows XP solution to support Vista+. In order to do this I needed to adapt the current custom GINA

(Graphical Identification and Authentication) to Credential Provider, allowing other identification methods such as biometric systems to login on Windows. The solution was developed using C++ and native API. The outcome of this project was that old customers were able to update their OSes to Windows Vista, 7 and newer versions without losing the control provided by the SCUA solution. Among technologies used are WinAPI, Windows Services and Device Drivers (WDK). I also fixed a serious bug with thread starvation in server attending to 80k clients by bypassing CTree Database solution to direct access main database in SQL Server.

Security Specialist at **OpenCS** (2005-2008)

- **Sniper.** The main goal of this project was to protect users from system vulnerabilities when accessing online banking on the computer. Brazilian banking system was a target from severals online attacks at that time, and Sniper solution was based on API hook by device drivers, creating a shield to protect users from malicious software running on the computer that detect the bank site access and try to capture the user information to login on the bank. After the first release I was responsible for develop a protection for Sniper itself, because hackers were trying to reverse engineer our solution, so I developed techniques for anti debugging and string obfuscation. A global log system was created to allow us to analyze some bugs happening only in specific sets of hardware and software. The number of computers where Sniper was installed was counted in hundreds of thousands and some issues were happening in all kinds of sets, like a user running Solitaire in a Windows 98. As an outcome, this product was sold

to a big bank from the South of Brazil and was the main cause of the company growing in the first two to three years after the first release. Among the technologies used are Assembly, WDK and Virtual Machines such as VMWare, Parallels, Virtual Box and QEMU. I had also to reverse engineered several trojans in a week basis to discover the API used in the threats in order to develop a new version with the new protection.

Security Developer at **SCUA** (2001-2005)

- Application and User Control System (SCUA). The main goal of this project is to protect the computer from viruses and other threats based on access rules on paths and Windows policies. The access rules are kept in a distributed CTree database and the product needed a Graphical Interface to configure those rules. I developed the desktop management application in C++ Builder by adapting MSVC libraries. It was my first big project with 10k+ lines of code. The outcome was that it became the oficial app to configure rules to SCUA, being used even today for all customers, 20 years after being developed.