Mediterra software

System portfolio

Control Driver

Technologies: JavaScript, Emby Server, home automation embedded platform

Duration: 6 months

Implemented driver for closed home automation embedded platform. Main features - media browsing and remote control of Emby Server (formerly Media Browser). We implemented the driver within pure JavaScript environment as well as HTTP and websocket protocol on top of raw TCP sockets. We also applied Emby Server client side API with authorization and live remote session handling.

Features different strategies for media browsing. Supports media browsing using Emby Server views hierarchy and iTunes-like queries.

Web dashboard

Technologies: Python, Django, Google Maps API, BeagleBone Black

Duration: 2 months

The goal of the project is to track the objects in a real time on a map and provides the info on the web dashboard. Trucks and tanks act like as objects with installed BeagleBone Board (BBB) and Janus modem (all logic is custom and implemented on our side due to complex nature of plugging it in) as a communicator.

Server receives data from BBB and provide users with geolocation info about objects and its status at the same time choosing the best route for the truck. On the second stage web dashboard is enhanced with the prediction analysis using ETL and machine learning techniques and advanced routing system.

Sandbox-like solution on Windows Server 2008 R2 platform

Technologies: Visual Studio, WDM, KMDF, WinDbg, VirtualKD

Duration: 6 months

Transparently installed and run software in redirected place, instead of Program Files. It provides good isolation of installed software packages. Component is implemented as kernel mode minifilter driver.

Transparent per-user redirection of some keys tree within registry. Component is implemented as kernel mode registry filter driver.

Per-user mutex name mangling. It allows different users to have more than one running instance of protected by named mutex applications. Use user-mode DLL injection and hooking.

Virtual Windows Registry

Technologies: EasyHook, C++, C#, windows API, WinDbg, ntsd, PE format, code injection

Duration: 1 year

Implemented a system allowing arbitrary application to run in a sandbox to protect system windows registry from any modification and harm. It also easily deploy the app with a portable «offline registry file».

The project involved code injection, hooking windows API functions, debugging 3rdparty applications in assembly language, debugging COM multithreaded multiple apartment code, Wow64 support, IAT (Import address table) modification, PE format understanding.

Mediterra
software

System portfolio

Cloud Anti-Malware Software

Technologies: C++11, Parallel Patterns Library, Sciter, windows API, SQLite, Microsoft Unit Testing Framework, JSON, HTTP, HTML, CSS, XML

Duration: 9 months

Development of an anti-malware application that scans and cleans your computer automatically, prevents malefactors from redirecting users to fake websites, removes browser hijacker and get rid of browser hijack.

The project involved multithreading, COM technology, Scheduler WinAPI, Cryptography WinAPI, network communication by http protocol, HTML/CSS/TiScript technologies for GUI implementation.

File Migration Network Program

Technologies: C++, boost, Ubuntu, network file system (NFS, CIFS), Eclipse

Duration: 6 months

Worked on a program to migrate files and folders from one network file system to another. Program supports network file systems such as NFS, CIFS, and multi-stage processes of estimation, copying and finishing and supports huge storage amounts like petabytes (1000s of Terabytes), including various versions of NetApp data storage devices.

System portfolio

Secure Thin Client

Technologies: C++/C, WinDDK/IFS

Duration: 6 months

Client/server application that allows to rollback all the changes made after the specified point in time (feature similar to Windows Restore function). After start working (creating restore point) it handles all writes to HDD and makes them temporary. All the applications (including Windows itself) sees all the changes in usual form (as they are made persistent), but as soon as user selects «Rollback» then all data (including Windows binary and settings files) are gotten restored to the original state.

Additionally application handles all user inputs (USB ports, floppy drives, external hard drives and etc.) and if some of them are denied by application settings, then it makes them inaccessible. Client application communicates with server part in order to control all user actions and report them to the centralized server.

HDMI Monitor Status

Technologies: C#, SetupApi, WinDDK

Duration: 2 months

Development of application that allows to detect status of monitors connected to PC through HDMI interface. Each PC has several HDMI ports and monitors connected to them (to all or only to some). Main task of the application is to detect what monitor is connected to which HDMI port (monitor identification is done using hardware serial number). For each connected monitor it is necessary to detect its state (turned on/ turned off), status (works well or in failure state), gather information and provide access to it using C# assembly to external tools. Additionally it was required to map specific .NET Screen object to each working monitor with correct support of multi-screen desktops.