Lab Oriented Project

On

Automotive Cyber Security and Aspice

Nitin Jaswal

2110990970

Supervised By

Praveen Kumar Reddy

Department of Computer Science and Engineering,

Chitkara University, Punjab

CHITKARA UNIVERSITY

Automotive Cybersecurity

My Project is an Embedded Systems project, which revolves around the Automotive Cybersecurity.

These are the techstack I have been using in my project.

- C/C++ - For writing the Programming Logic.
- OpenSSL - for encryption and decryption of data.
- IPC Binder – To maintain the Interprocess Communication among various processes.
- Gmock and Gtest – For writing the functional Unit Tests.
- Yocto – It provides Build System to create custom OS for different SOC.
- Cmake – For the Build and automation to generate the build files.

OpenSSL

CHITKARA UNIVERSITY

OpenSSL is a free, open-source command line tool that provides an open source implementation of the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. It is used to generate private keys, manage certificates, and equip client applications with encryption and decryption.

OpenSSL is widely used by Internet servers, including the majority of HTTPS websites. It is compatible with both Windows and Linux distributions.

OpenSSL provides a set of cryptographic functions, including

- Hash functions
- Symmetric and asymmetric encryption
- CA

OpenSSL can be used as a cryptographic library in your own applications.

IPC Binder

CHITKARA UNIVERSITY

IPC Binder is the main Inter-Process Communication (IPC) system in Android. It allows applications to communicate with each other. For example, Android services are built on top of Binder.

Binder IPC allows the application framework to cross process boundaries and call into the Android system services code. This enables high-level framework APIs to interact with Android system services.

Binder IPC Framework enables a remote invocation of the methods in other processes. A client process can communicate to another server process and can run the methods in other process as it is done locally and can get the required data from the server process.

Binder has a maximum transaction size of 1MB. For example, TransactionTooLargeException is thrown when applications try to send bigger than 1MB messages using Binder transaction.

Binder was used in Android instead of an existing interprocess mechanism because in the mid 2000s, there was no reliable standardized fast in-kernel IPC in Linux. The traditional SysV IPC was mostly deprecated due to its age and fell out of use.

Gmock and Gtest

Gmock and Gtest are both testing frameworks for C++. Gmock is a mocking framework, while Gtest is a unit testing framework.

Gmock allows you to create fake objects in order to remove external dependencies for effective testing. This is useful for testing code that relies on other objects, such as databases or web services.

Gtest is a unit testing framework that allows you to write unit tests for your C++ code. Unit tests are small, self-contained tests that test a specific function or unit of code.

Gmock and Gtest can be used together to test code that conforms to the SOLID principles for object-oriented software design.

Yocto

CHITKARA UNIVERSITY

Yocto is a Linux Foundation collaborative open source project that provides tools, templates, and methods for building Linux systems from scratch. The goal of the project is to produce tools and processes that enable the creation of Linux distributions for embedded and IoT software.

Yocto is a set of tools, templates, and methods that are separate from the reference distribution (Poky) and the OpenEmbedded build system. Poky contains some metadata, OpenEmbedded core, and Bitbake.

Yocto automates the complete build process. It provides tools, metadata, and a build framework to create the custom Linux distro for your embedded and IoT devices.

Some popular alternatives and competitors to Yocto include: Ubuntu, Debian, Docker, Buildroot, AWS CloudFormation

Cmake

CHITKARA UNIVERSITY

CMake is a free, open-source, cross-platform tool for software development. It's used for build automation, testing, packaging, and installation of software. CMake uses compiler and platform independent configuration files to generate native build tool files specific to your compiler and platform.

CMake is not a build system itself; it generates another system's build files. It's designed to be used in conjunction with the native build environment.

CMake was designed for C++, but can be used to build software written in other languages, such as C, Java, and Rust.

The CMake Tools extension integrates Visual Studio Code and CMake to make it easy to configure, build, and debug your C++ project.

You can learn more about CMake from the CMake documentation. The CMake tutorial is a good starting point.

MACsec

- **MKAD (MKA Daemon)** is a user-space daemon responsible for implementing the MACsec Key Agreement (MKA) protocol, as defined in the IEEE 802.1X-2010 standard. MKA is a crucial component of MACsec (Media Access Control Security), which provides secure communication at Layer 2 of the OSI model by encrypting and authenticating Ethernet frames. MKAD facilitates the dynamic exchange and management of cryptographic keys between MACsec-capable peers using the EAPOL-MKA (Extensible Authentication Protocol over LAN - MACsec Key Agreement) protocol. It establishes and maintains Security Associations (SAs), negotiates Secure Channels (SCs), and distributes session keys required for MACsec encryption and decryption.

- In our project, MKAD was integrated into an embedded Linux environment to enable end-to-end secure communication between two systems over a physical Ethernet link. We configured and validated MKAD to perform secure peer authentication, real-time key negotiation, and automated secure channel setup. Additionally, we verified the encrypted communication using packet captures and confirmed the integrity and confidentiality of the transmitted data. MKAD's integration demonstrates a practical and secure implementation of Layer 2 encryption, making it highly suitable for securing communication in automotive, industrial, and enterprise embedded systems.

Thank You