Gabriel Ravier

Looking for employment

CONTACT

61449 Steinbach am Taunus

Currently in Germany (can move if necessary)

E-mail: gabravier@gmail.com

Phone: +33 6 36 46 16 43

SKILLS

C and C++ - Expert
Linux – Expert
x86 assembly – Expert
Cybersecurity – Experienced
Reverse Engineering (IDA/Ghidra) – Highly experienced
Amazon Web Services (AWS) – Experienced
Git – Expert
POSIX/Unix Shell - Expert
CMake – Experienced
GNU Make – Expert
PostgreSQL – Experienced
Penetration Testing – Intermediate
TypeScript (Node.js) - Intermediate
Sybase – Intermediate

WORK EXPERIENCE

Commerzbank — Sep 2024 – Aug 2025

Intern (end-of-studies internship)

I worked as a software engineer at Commerzbank (second largest bank in Germany), on the internal software used for handling hedge accounting, which had been previously developed over more than 25 years by several dozen people in the company. I worked on cross-platform (Linux/Windows) C++ code and SQL code using both Sybase and PostgreSQL. I worked with both German and Czech colleagues (with English or German as the working language)

I worked on a full migration of this software stack from using Sybase as its backend database to using PostgreSQL for it, converting swathes of C++, Bash and SQL code to avoid using Sybase-specific extensions and use standard SQL features that PostgreSQL supports, along with implementing connections to PostgreSQL servers using libpq, replacing Sybase's proprietary libraries.

I also reduced the complexity of the build integration of several libraries through vendoring, so they could be integrated more easily with the rest of the codebase, and fixed many warnings the codebase formerly contained, along with memory issues found in C++ code, many of which were found through the integration of valgrind into the project's CI and unit tests.

TrackIt — Jul 2021 – Dec 2021, Apr 2023 – Aug 2023

Intern

I worked as a software engineer at TrackIt, an American company offering services for working with AWS, on their flagship software offerings, TrackIt and Tagbot, primarily through fixing bugs and implementing new features across both the backend and frontends, using Go and JavaScript. I worked with both American and French colleagues (English was typically used as the working language).

Those projects heavily interfaced with the AWS API for the purposes of helping manage one's AWS resources, giving me substantial experience with it.

I also contributed to several other minor projects which involving AWS, which gave me further experience with AWS. Additionally, I helped speed up the resolution of a few severe production problems, and participated in several deployments of the above software, obtaining practical experience with Kubernetes in the process.

I also contributed to TrackIt's security offerings, participating in 1 penetration test and 3 security assessments.

I also worked on software, written in C++, Shell and TeX, which assists in the creation of security assessment reports. Notably, I wrote a program which automatically converts a report from the Prowler security tool into a human-readable PDF, and created a webservice to make this tool accessible, using TypeScript and the AWS Cloud Development Kit (CDK), allowing use of this program from a web-based interface hosted on AWS.

NZUP — Jun 2018 – Jun 2018

Intern

I worked 4 weeks in this company as an intern, working to optimize their Go backend, through conversion of specific bottleneck functions to optimized C code to greatly improve their performance.

PROJECTS

Yalibct (Yet Another Libc Testsuite) — Aug 2022 – Present

This project, which I am currently working on in my spare time, aims to create a testsuite for libc, so as to try to find as many bugs as possible in existing libc implementations. As libc is the most fundamental library present in a system (almost all programs rely on it in some way or another – e.g. it is at the core of every single program running or relying on any code and/or library written in C or C++ or using a runtime written in C or C++ (e.g. almost all programs written in C#, JS, Python, Ruby, Java, PHP, etc.)), bugs in it are particularly important, and this testsuite aims to find as many of those as possible.

As of early 2025, I have, while developing this project, filed more than 73 bug reports (usually along with a patch fixing the issue) with various notable libc implementation, such as glibc, musl, cosmopolitan and dietlibc.

Epitech - AREA — Jan 2023 – Mar 2023

This was a project to write an application alike to IFTTT and/or Zapier, in which users, from either a web interface or a cross-platform Android/iOS mobile application could link Actions and Reactions from various services by creating Triggers.

A trigger defined an action → reaction workflow - such that when an action occurs, a given reaction results. For instance, a trigger could be "when receiving an e-mail in Gmail, a message gets sent to a Discord

channel", or "adding a track to a Spotify playlist sends an email through Outlook".

I wrote a very large majority of the backend code, handling all of the business logic (the frontends only serve as user interfaces that redirect requests to/from the backend server), implementing 15 actions and 16 reactions across 17 services.

I also implemented a formatting system with variable substitution, allowing users to arbitrarily inject data from actions into reacitons in their triggers (for instance, a trigger that, when a new Astronomy Picture Of the Day is published by NASA, sends a message to a Telegram channel, could make the message something like "NASA's picture of the day, '$(title)', has been published at $(url)!" and the backend would replace the placeholders with the title and URL of the picture).

During the planning phase of the project, I experimented with other technologies, such as C++ with cpp-httplib, Apache Cassandra, Python with Flask, MongoDB, PHP, MySQL, Ruby on Rails, SQLite, or TypeORM, before settling on TypeScript with Fastify, Prisma and PostgreSQL.

Epitech - Arcade — Mar 2022 – Apr 2022

This was a project to write a complicated "arcade-like" program that allows one to play various old games using a variety of different graphics library, all through a common interface. We decided to make it possible to use 4 different graphics libraries (including one that works in text mode), SFML, OpenGL, SDL and ncurses, and remake two games, Pac-Man and Nibbler, which we both reproduced accurately.
The common interface was created by me and shared with all the other groups working on the same project at our school. As the interface separated games, graphics libraries and core executable, it was possible to use anyone's core executable with a game from another group and a graphics library from yet another group and have them all work together without issues.

I primarily worked on:
- The Pac-Man game implementation
- The core executable
- The ncurses graphics implementation
- Some parts of the SFML and SDL graphics implementation
- Sound handling across all graphics implementations
- Implementing a small C++ wrapper library for dynamic library handling (dlopen, dlclose, etc.)

Video of the project: https://www.youtube.com/watch?v=yMk1y3bvGwQ

Epitech – Hydra and Chisel — Mar 2022 – Apr 2022, Nov 2022 – Dec 2022

These were two CTF projects where we were tasked with finding flags on various remote systems, with a total of 15 challenges (per project) hosted on the TryHackMe platform, specially created for the purpose of this project.

The challenges were focused around web-hosted services in various languages (i.e. most of the time web-based but with backends in e.g. Python/Node/PHP/Wordpress/etc, sometimes involving other web-accessible services like FTP or NFS) in which one had to find a vulnerability in order to get user-level access to a machine, before then having to find another vulnerability within the system to escalate from user-level privileges to root/administrator-level privileges (e.g. by exploiting poorly secured configs of certain system services, reverse-engineering some administrative process, sometimes even further having to escape a container of some kind or having to penetrate yet another web service).

A global leaderboard was present where all participating students across Epitech were ranked by the amount of challenges they finished and the speed at which they did so - I finished first in the Strasbourg campus in both projects.

Epitech – Binary Security/Binsecu — May 2021– May 2021

This was a CTF project where we were tasked with finding flags on various remote systems which were specially created for the purpose, through finding vulnerabilities/exploits to get them to reveal the secret flags needed to fulfill the project, with the various challenges present on each system getting more and more elaborate as they went on.

In this CTF, most of the challenges involved having user access to a server and having to find a vulnerability in a custom-made SUID program (this practically systematically involved reverse-engineering the program binary to inspect how it behaves and find where a vulnerability might reside) to get root access to the machine.

A global leaderboard was present where all participating students across Epitech were ranked by the amount of challenges they finished and the speed at which they did so - I finished first across all Epitech campuses in the whole year.

CSE2 — Jun 2018 – Oct 2020

A project to make an open source exact equivalent of the executable used in the indie game Cave Story, in order to facilitate modding of the game. A lot of skills in reverse engineering, C and C++ were necessary to work on this. I was one of the biggest contributors on the project, which was essentially fully completed by the end of 2020.
Video of the project: https://www.youtube.com/watch?v=VYRTTl5pk90

EDUCATION

Computer Software Engineering — Sep 2020 – Sep 2025

EPITECH – European Institute of Information Technology

I spent five years studying various programming subjects at Epitech. Epitech's teaching approach is project-based – there were far more projects to work on than classes. A majority of projects were focused around studying the C and C++ programming languages, primarily under a Linux environment but also quite often writing cross-platform (Linux/Windows) code, for various different purposes (command-line utilities, shells, video games, networking, concurrency, compilers, etc.)

Other subjects taught included Cybersecurity (several projects were focused around CTF or OSINT challenges), Haskell, Web Development, DevOps, and more. I also learned various project-management skills over the course of my education at Epitech.

Computer Software Engineering — Sep 2023 – Jun 2024

DCU – Dublin City University

I spent a year studying here as an exchange student from Epitech, studying various subjects such as compiler construction, OOP (with Java), full-stack web applications (both with Python+Django and with Java+Spring), machine learning (with scikit-learn) and DevOps (e.g. through data processing using POSIX utilities and other tools such as jq or R)

Primary/Middle/High School — Oct 2011 – Jun 2020

Lycée Français Victor Hugo – Frankfurt, Germany

French international school in Germany. I obtained a Sciences Baccalaureat (i.e. a “Bac S” or “Bac scientifique”) with mathematics specialization and top marks (i.e. “mention très bien”)

INTERESTS

Programming in general, I spend a lot of time contributing to projects I like. I’ve continually studied and routinely used C and C++, along with assembly (on several different processors), since the age of 13, on various hobby, school and professional projects. I am also part of a number of online communities dedicated to programming.

I am interested in cyber-security, having obtained some experience in a few school-wide CTF and OSINT competitions, and one of my internships.

LANGUAGE SKILLS

I speak close to perfect English and have obtained a 965 on a TOEIC test. My mother tongue is French.