

Gabriel Ravier

Recherche un emploi

| | | |
|---------------------------|--|--------------------------------------|
| CONTACT | 61449 Steinbach am Taunus | E-mail: gabravier@gmail.com |
| | Actuellement en Allemagne (peut déménager si nécessaire) | Phone: +33 6 36 46 16 43 |
| SKILLS | C et C++ - Expert Linux – Expert Assembleur x86 – Expert Cybersecurité – Experimenté Reverse Engineering (IDA/Ghidra) – Très expérimenté Amazon Web Services (AWS) – Expérimenté Git – Expert POSIX/Unix Shell - Expert CMake – Expérimenté GNU Make – Expert PostgreSQL – Expérimenté Penetration Testing – Intermédiaire TypeScript (Node.js) – Intermédiaire Sybase - Intermédiaire | |
| EXPERIENCE PROFESSIONELLE | Commerzbank | 2024-09 – 2025-08 |
| | Stagiaire (Stage de fin d’études) | |
| | J'ai travaillé comme ingénieur logiciel à Commerzbank, seconde plus grande banque d’Allemagne, sur le logiciel interne pour la gestion du hedge accounting, qui avait été précédemment développé pendant plus de 25 ans par plusieurs dizaines de personnes dans l'entreprise. J'ai travaillé sur du code C++ et SQL multiplateforme (Linux/Windows) utilisant Sybase et PostgreSQL J’ai travaillé avec des collègues allemands et tchèques (avec l’anglais ou l’allemand comme langue de travail). | |
| | J'ai participé à la migration complète de ce logiciel de Sybase comme base de données back-end à PostgreSQL. J'ai converti des pans entiers de code C++, Bash et SQL afin d'éviter l'utilisation d'extensions spécifiques à Sybase et d'utiliser les fonctionnalités SQL standard qui fonctionnent avec PostgreSQL. J'ai également implémenté des connexions aux serveurs PostgreSQL avec libpq à la place des bibliothèques propriétaires de Sybase. | |
| | J'ai également réduit la complexité de l’intégration au système de build de plusieurs libraries, afin qu'elles puissent être intégrées plus facilement au reste du code. J’ai aussi corrigé de nombreux warnings que la base de code contenait auparavant, ainsi que des problèmes de mémoire trouvés dans le code C++, dont beaucoup ont été trouvés grâce à l'intégration de valgrind dans le CI et les tests unitaires du projet. | |
| | TrackIt | 2021-07 – 2021-12, 2023-04 - 2023-08 |
| | Stagiaire | |
| | J'ai travaillé comme ingénieur logiciel sur les logiciels phares de TrackIt, TrackIt et Tagbot, principalement en corrigeant des bugs et en implémentant de nouvelles fonctionnalités, tant en back-end qu'en front-end, en utilisant le Go et le JavaScript. J’ai travaillé avec des collègues états-uniens et français (l’anglais étant généralement utilisé comme langue de travail). | |
| | Ces projets utilisaient étroitement l'API AWS pour faciliter la gestion des ressources AWS, ce qui m'a permis d'acquérir une solide expérience avec celle-ci. J'ai également contribué à plusieurs autres projets mineurs impliquant AWS, ce qui m'a permis d'approfondir mon expérience avec ce dernier. J'ai aussi contribué à accélérer la résolution de quelques problèmes de production graves et participé à plusieurs déploiements des logiciels mentionnés ci-dessus, acquérant ainsi une expérience pratique de Kubernetes. J'ai également contribué aux offres de sécurité de TrackIt, en participant à un test d'intrusion et à trois évaluations de sécurité. | |
| | J'ai aussi travaillé sur des logiciels, écrits en C++, Shell et TeX, qui facilitent la création de rapports d'évaluation de sécurité. Notamment, j'ai écrit un programme qui convertit automatiquement un rapport de l'outil de sécurité Prowler en un PDF lisible par l'homme, et j’ai créé un service Web pour rendre cet outil accessible, en utilisant TypeScript et le kit de développement cloud AWS (CDK), permettant l'utilisation de ce programme à partir d'une interface Web hébergée sur AWS. | |
| | NZUP | 2018-06 |
| | Stagiaire | |
| | J’ai travaillé 4 semaines dans cette entreprise sur l'optimisation de leur backend Go, grâce à la conversion de fonctions bottleneck en code C optimisé pour améliorer considérablement leurs performances, ainsi que des modifications architecturales/de la chaîne d'outils à plus grande échelle pour rendre le code backend généralement plus rapide. | |
| PROJETS | Yalibct (Yet Another Libc Testsuite) | 2022-08 - Présent |
| | Ce projet, sur lequel je travaille actuellement pendant mon temps libre, vise à créer une testsuite pour la libc, afin d'essayer de trouver autant de bugs que possible dans les implémentations existantes de la libc. Comme la libc est la bibliothèque la plus fondamentale présente dans un système (presque tous les programmes en dépendent d'une manière ou d'une autre – par exemple, elle est au cœur de tout programme exécutant ou s'appuyant sur tout code et/ou library écrit en C ou C++ ou utilisant un runtime écrit en C ou C++ (par exemple presque tous les programmes écrits en C#, JS, Python, Ruby, Java, PHP, etc.)), les bugs qu'il contient sont particulièrement importants et cette testsuite vise à en trouver le plus grand nombre possible . | |
| | Depuis le début de ce projet, j’ai, lors du développement de ce projet, déposé plus de 73 bug reports (généralement accompagnés d'un patch corrigeant le problème) auprès de diverses implémentations notables de libc, telles que glibc, musl, cosmopolitan et dietlibc. | |

Epitech - AREA

2023-01 – 2023-03

Un projet visant à développer une application similaire à IFTTT et/ou Zapier, permettant aux utilisateurs, depuis une interface web ou une application mobile Android/iOS multi-plateforme, de lier des actions et des réactions de différents services en créant des triggers.

Un trigger définissait un flux action → réaction, de sorte que lorsqu'une action se produit, une réaction donnée se produit. Par exemple, un déclencheur pourrait être “à la réception d'un e-mail dans Gmail, un message est envoyé sur un canal Discord”, ou “l'ajout d'un titre à une playlist Spotify envoie un e-mail via Outlook”.

J'ai écrit la majeure partie du code backend, gérant toute la business logic (les frontends servent uniquement d'interfaces utilisateur redirigeant les requêtes vers/depuis le serveur backend), implémentant 15 actions et 16 réactions sur 17 services.

J'ai également implémenté un système de formatage avec substitution de variables, permettant aux utilisateurs d'injecter arbitrairement des données issues d'actions dans des réactions de leurs déclencheurs (par exemple, un déclencheur qui, lorsqu'une nouvelle photo astronomique du jour est publiée par la NASA, envoie un message à un canal Telegram, pourrait afficher un message du type “La photo du jour de la NASA, '\$(title)', a été publiée à \$(url) !” et le backend remplacerait les placeholders par le titre et l'URL de la photo).

Pendant la phase de planification du projet, j'ai expérimenté avec d'autres technologies, telles que C++ avec cpp-http lib, Apache Cassandra, Python+Flask, MongoDB, PHP, MySQL, Ruby on Rails, SQLite ou TypeORM, pour finalement décider d'utiliser TypeScript avec Fastify, Prisma et PostgreSQL.

Epitech - Arcade

2022-03 – 2022-04

Un projet de programme complexe « arcade-like » permettant de jouer à divers jeux anciens via une interface commune entre jeux et bibliothèques graphiques. Nous avons rendu possible l'utilisation de 4 bibliothèques graphiques différentes (dont une fonctionnant en mode texte), SFML, OpenGL, SDL et ncurses, et reproduit fidèlement les jeux Pac-Man et Nibbler. J'ai conçu l'interface commune, que j'ai partagé avec tous les autres groupes travaillant sur ce même projet dans notre école, séparant les jeux, les bibliothèques graphiques et l'exécutable principal. Il était possible d'utiliser un exécutable principal avec divers jeux et bibliothèques graphiques sans problème (même si ces composants venaient chacun de différents groupes).

J'ai principalement travaillé sur :

- L'implémentation du jeu Pac-Man
- L'exécutable principal
- L'implémentation graphique de ncurses, et certaines parties de l'implémentation graphique SFML et SDL, ainsi que la gestion du son dans toutes les implémentations graphiques
- L'implémentation d'une petite bibliothèque wrapper C++ pour la gestion dynamique des bibliothèques (dlopen, dlclose, etc.)

Vidéo du projet ici : <https://www.youtube.com/watch?v=yMk1y3bvGwQ>

Epitech – Hydra and Chisel

2022-03 – 2022-04, 2022-11 – 2022-12

Deux projets CTF dans lesquels nous avons pour tâche de trouver des flags sur divers systèmes en remote, avec un total de 15 défis hébergés sur la plateforme TryHackMe (i.e. ils sont similaires à certains des défis qui y sont trouvés), spécialement créés pour ce projet.

Les défis se concentraient sur les services hébergés sur le Web dans différents langages (c'est-à-dire la plupart du temps accessible par navigateur web mais avec des backends par exemple Python/Node/PHP/Wordpress/etc, impliquant parfois d'autres services accessibles sur le Web comme FTP ou NFS) dans lesquels il fallait trouver une vulnérabilité pour obtenir un accès au niveau utilisateur à une machine, avant de devoir ensuite trouver une autre vulnérabilité dans le système pour passer des privilèges au niveau utilisateur aux privilèges au niveau root/administrateur (par exemple en exploitant des configurations mal sécurisées de certains services système, reverse-engineering de certains processus administratifs, devoir parfois même échapper à un conteneur quelconque ou devoir pénétrer encore un autre service Web).

Un classement mondial était présent où tous les étudiants participants d'Epitech étaient classés en fonction du nombre de défis qu'ils avaient relevés et de la vitesse à laquelle ils l'avaient fait - j'ai terminé premier sur le campus de Strasbourg dans les deux projets.

Epitech – Binary Security/Binsecu

2021-05

Un projet CTF dans lequel nous étions chargés de trouver des flags sur divers systèmes remote spécialement créés à cet effet, en trouvant des vulnérabilités/exploits pour les amener à révéler les flags secrets nécessaires à la réalisation du projet, avec les différents défis présents sur chacun devenant de plus en plus élaborés.

Dans ce CTF, la plupart des défis impliquaient d'avoir accès aux utilisateurs à un serveur et de devoir trouver une vulnérabilité dans un programme SUID créé spécialement pour l'occasion (cela impliquerait pratiquement systématiquement un reverse-engineering du binaire du programme pour inspecter son comportement et trouver où pourrait se trouver une vulnérabilité) pour obtenir un accès root à la machine.

Un classement mondial était présent où tous les étudiants participants d'Epitech étaient classés en fonction du nombre de défis qu'ils avaient relevés et de la vitesse à laquelle ils l'avaient fait - j'ai terminé premier de tous les campus d'Epitech.

Epitech – Hunter

2020-11

Il s'agissait d'un projet scolaire dans lequel nous devions recréer un jeu ressemblant à Duck Hunt, dans lequel le joueur tirait sur des cibles en mouvement sur l'écran. J'ai décidé de réaliser une reproduction de Duck Hunt la plus fidèle possible, ce que j'ai fait avec succès. Vidéo du projet ici : <https://www.youtube.com/watch?v=TFZgotSo8wM>

CSE2

2018-06 – 2020-10

Un projet visant à créer un équivalent open source exact de l'exécutable utilisé dans le jeu indie Cave Story, afin de faciliter le modding du jeu. De nombreuses compétences en reverse-engineering, C et C++ ont été nécessaires pour travailler là-dessus. J'ai été l'un des plus grands contributeurs du projet, qui était pour l'essentiel entièrement achevé fin 2020. Vidéo du projet ici : <https://www.youtube.com/watch?v=VYRTTI5pk90>

EDUCATION

Computer Software Engineering2020-09 – 2025-09

EPITECH – European Institute of Technology

J'ai étudié la programmation pendant cinq ans à Epitech. L'approche pédagogique d'Epitech est basée sur des projets : il y avait beaucoup plus de projets à réaliser que de cours. La majorité des projets étaient axés sur l'étude des langages de programmation C et C++, principalement sous Linux, mais, aussi souvent, sur l'écriture de code multiplateforme (Linux/Windows), pour des applications variées (utilities de command-line, shells, jeux vidéo, réseau, multi-threading, compilers, etc.).

Les autres matières enseignées comprenaient la cybersécurité (plusieurs projets étaient axés sur les défis CTF ou OSINT), Haskell, le développement web, DevOps, etc. J'ai également acquis diverses compétences en gestion de projet au cours de ma formation à Epitech.

Computer Software Engineering2023-09 - 2024-06

DCU – Dublin City University

J'ai passé un an ici en tant qu'étudiant d'échange venant d'Epitech, étudiant divers sujets tels que la construction de compilateurs, la OOP/POO (avec Java), les applications web full-stack (avec Python+Django ainsi qu'avec Java+Spring), le machine learning (avec scikit learn) et DevOps (e.g. du traitement des données à l'aide des utilitaires POSIX et d'autres outils comme jq ou R)

Primaire/Collège/Lycée2011-10 – 2020-06

Lycée Français Victor Hugo – Frankfurt, Germany

Ecole internationale française en Allemagne. J'ai obtenu un baccalauréat scientifique spécialisation mathématiques avec la mention très bien.

INTÉRÊTS

La programmation en général. Je passe beaucoup de temps à contribuer aux projets que j'aime. J'étudie et utilise régulièrement le C et le C++, ainsi que l'assembleur (sur plusieurs processeurs), depuis l'âge de 13 ans, sur divers projets hobbyistes, scolaires ou professionnels. Je fais également partie d'un certain nombre de communautés en ligne dédiées à la programmation.

Je m'intéresse à la cybersécurité, ayant pu acquérir une certaine expérience lors de quelques concours CTF à Epitech et durant un de mes stages.

COMPÉTENCES LINGUISTIQUES

Je parle un anglais presque parfait et j'ai obtenu un 965 au test TOEIC. Ma langue maternelle est le français.