Dossier de compétences

Axel Roudaut — Ingénieur DevSecOps & Data Engineer

Montpellier — axel.roudaut@outlook.com — +33 6 29 76 33 91

LinkedIn : www.linkedin.com/in/axel-roudaut-devops/ — CV : axelroudaut.github.io/Resume/

Résumé

DevSecOps et Data Engineer dans le domaine de la cybersécurité avec **7+ ans d'expérience** dans la conception et l'exploitation d'infrastructures distribuées et sécurisées (Thales, Ministère des Armées). Spécialisé en automatisation CI/CD, orchestration de services et déploiement de plateformes de traitement de données. Ouvert à des postes DevOps, Data Engineering et Platform Engineering à **Montpellier** à partir de **septembre 2026**.

Références et contacts pour recommandation communiqués sur demande.

Expériences

1. Thales — février 2022 à aujourd'hui

— **Intitulé du poste** : Ingénieur DevSecOps en Cybersécurité
— **Entreprise / client** : THALES SIX GTS France
— **Lieu** : Rennes
— **Secteur d'activité** : Défense

Contexte

Conception, automatisation et exploitation des plateformes DevOps & Cloud internes supportant les lignes de produits de cybersécurité de l'entité : cloud privé on-premise hébergeant des produits de sécurité, des environnements de simulation d'attaques et des services de traitement de données pour les équipes de cyberdéfense.

Équipe : 16 personnes.

Méthodologie : Agile Scrum, sprints de 2 semaines avec tickets JIRA, workflow GitLab avec revue de code et validation par intégration continue.

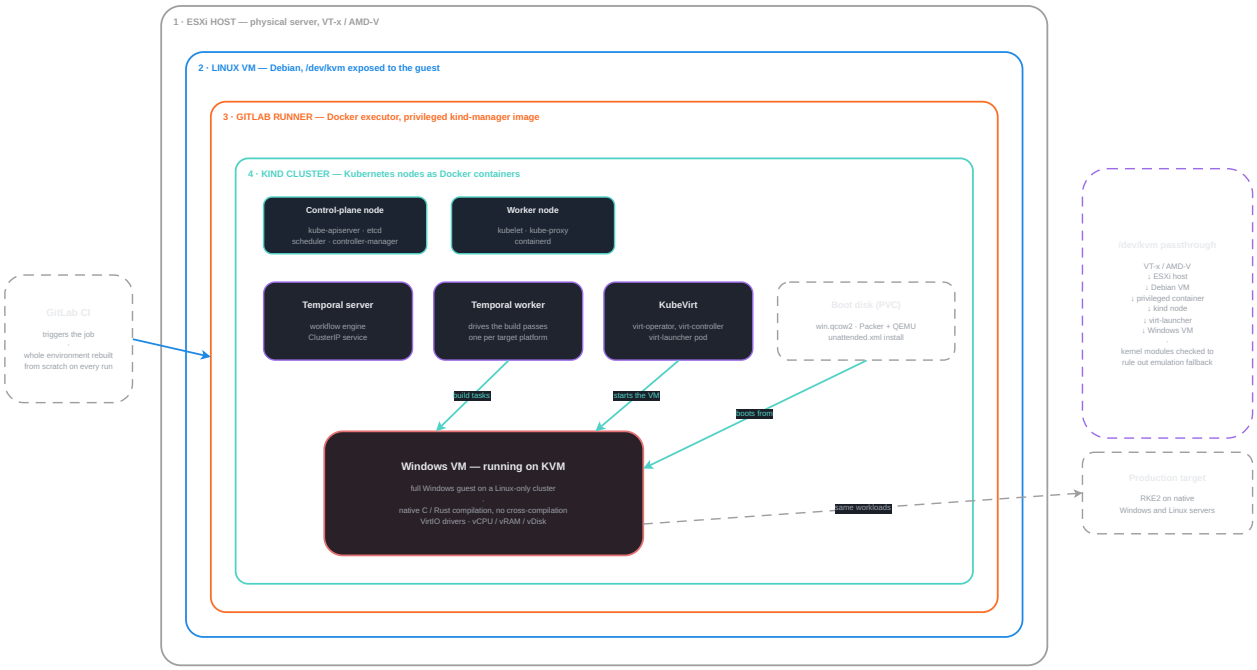
Objectifs : fiabiliser et rendre reproductibles les déploiements, réduire le time-to-market des produits de sécurité développés, standardiser les environnements d'exécution et renforcer la sécurité, l'observabilité et la scalabilité.

Détail des réalisations

— **Industrialisation du déploiement d'un cluster Kubernetes chez le client** : un méta-fichier JSON unique (nom de domaine, nombre de nœuds, certificats, URL applicatives) et une seule commande enchaînent le provisionnement des VM nœuds par Terraform sur plateforme on-premise VMware ou OpenStack, la vérification des prérequis (SSH, DNS, noms d'hôtes, certificats), le déploiement du cluster via Kubespray et l'installation des charts Helm de nos produits, générés à partir de ce même fichier. Orchestration en Ansible et exécution depuis GitLab CI avec tests de non-régression, garantissant une installation reproductible d'un site client à l'autre. (Kubernetes, Terraform, VMware, OpenStack, Kubespray, Helm, Ansible, GitLab CI)

Chaîne d'installation d'un cluster Kubernetes : un fichier de configuration, une commande.

— **Preuve de concept d'orchestration Kubernetes de compilations natives C/Rust** sur Windows et Linux (sans cross-compilation), pilotée par Temporal, pour une cible de production RKE2. Socle local choisi après benchmark de minikube, kind, k3s et k3d sur les performances, la reproductibilité en air-gap et la maturité : kind, pour son intégration en CI. Pods Windows exécutés sur des nœuds exclusivement Linux via KubeVirt, préféré à dockur/windows pour ses performances malgré une mise en œuvre plus lourde : accès direct à /dev/kvm et contrôle des modules noyau pour écarter l'émulation, image « golden » qcow2 construite sous Packer et QEMU avec pilotes VirtIO via unattended.xml. (Kubernetes, kind, KubeVirt, RKE2, Temporal, KVM, QEMU, Packer, GitLab CI, Rust, C)



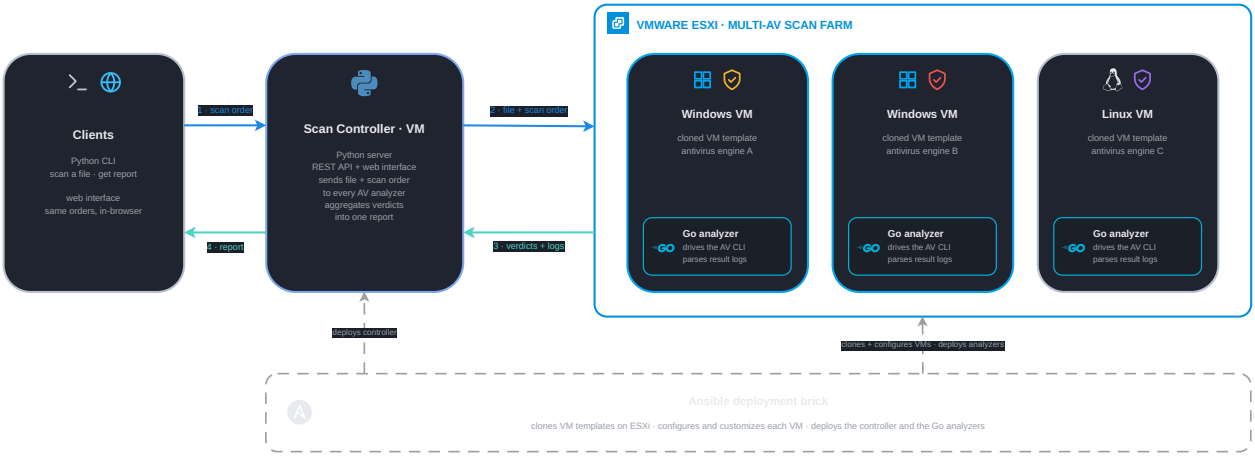
Compilations natives Windows et Linux orchestrées par Kubernetes, pods Windows portés par KubeVirt.

— **Déploiement et maintien d'environnements Linux/Windows de simulation d'attaques** sur plateforme on-premise VMware : industrialisation de la fabrication des images avec Packer,

provisionnement des machines virtuelles et configuration système par rôles Ansible et scripts PowerShell. La préparation manuelle des images a laissé place à une chaîne versionnée, rendant les environnements jetables et reproductibles à l'identique. (API VMware, Packer, Ansible, Terraform, PowerShell)

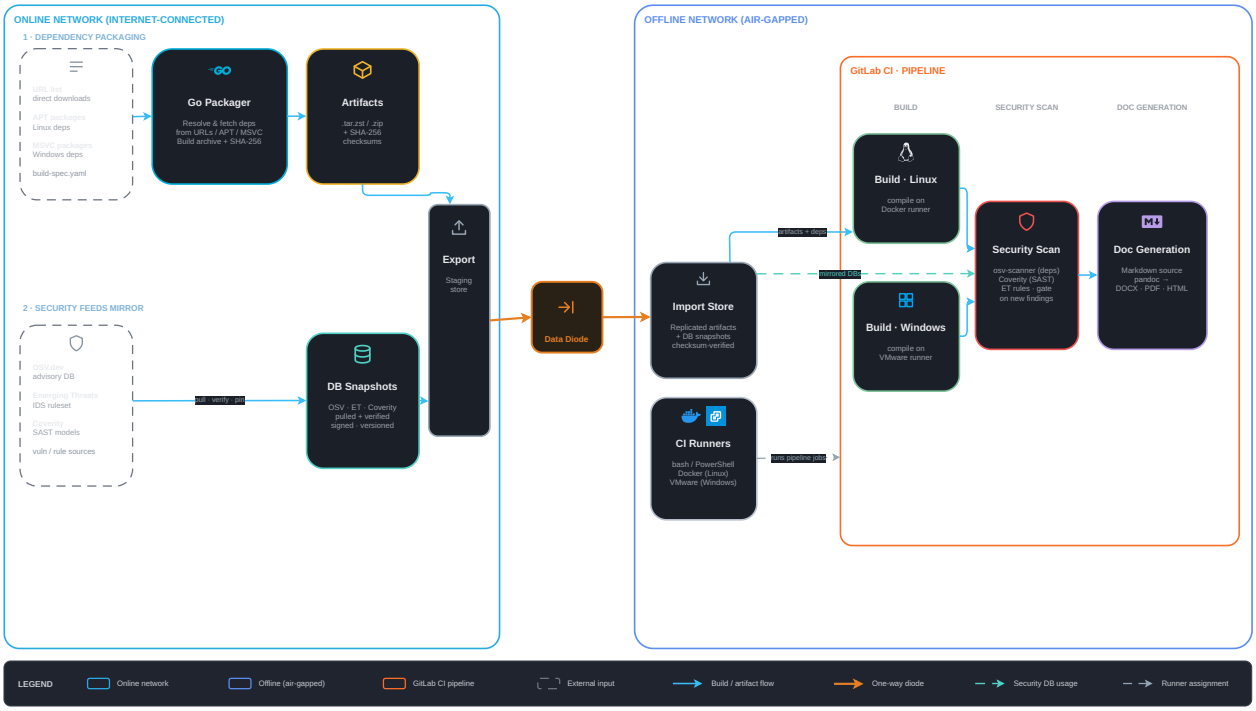
Fabrique d'images de machines virtuelles : de la source versionnée au gabarit déployable.

— **Exploitation d'une plateforme d'analyse multi-antivirus auto-hébergée**, type VirusTotal : orchestration de moteurs d'analyse isolés en machines virtuelles sur VMware ESXi, soumission des fichiers et restitution des verdicts via une API REST, services d'analyse développés en Python et Go puis déployés par Ansible. Permet aux analystes de qualifier un fichier sensible sans le transmettre à un service externe. (Python, Go, Ansible, VMware ESXi, API REST)



Plateforme d'analyse multi-antivirus : soumission, moteurs isolés en machines virtuelles, restitution des verdicts.

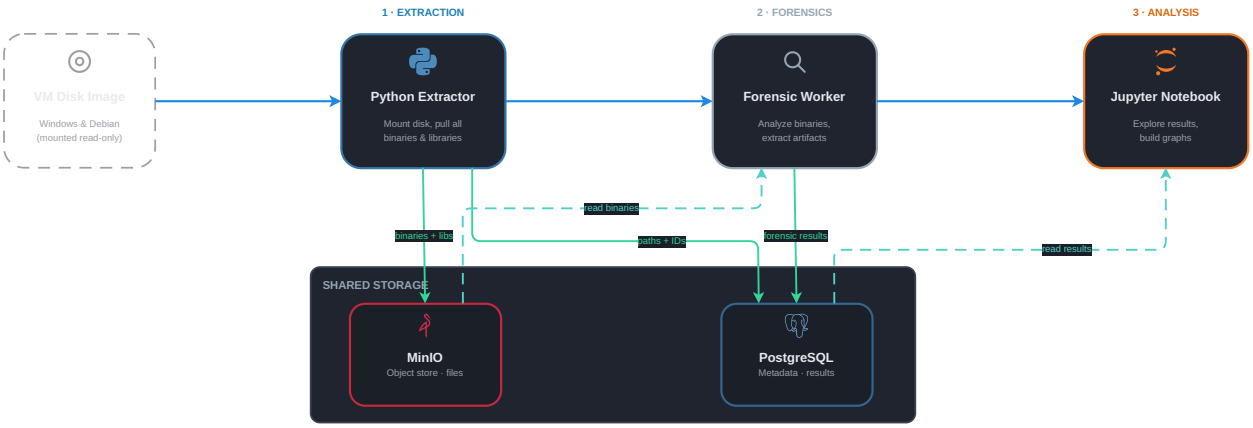
— **Automatisation des pipelines CI/CD pour optimiser les chaînes de production** : packaging des dépendances, construction des images conteneurisées, analyse de sécurité et promotion des artefacts vers les environnements cibles, avec des étapes outillées en Python, Go, Bash et PowerShell pour couvrir aussi bien les composants Linux que Windows. Suppression des étapes de livraison manuelles pour optimiser le temps de livraison. (GitLab CI, Docker, Python, Go, Bash, PowerShell)



Chaîne CI/CD : packaging des dépendances, construction des images, analyse de sécurité et promotion des artefacts.

— **Mise en place de la supervision et de l'observabilité des infrastructures** : collecte des métriques système et applicatives par Prometheus, tableaux de bord et alerting par service sous Grafana, stack conteneurisée déployée par Ansible. Indicateurs suivis : disponibilité des services, consommation CPU/mémoire des nœuds, durée et taux d'échec des pipelines — permettant d'anticiper les saturations de capacité avant qu'elles n'affectent les utilisateurs. (SNMP, Prometheus, Grafana, Docker, Ansible)

— **Développement d'un outil de statistiques forensic sur des binaires** : montage du système de fichiers d'un disque Windows ou Linux, extraction de tous les binaires présents vers un serveur MinIO puis calcul d'indicateurs sur les exécutables stockés sur un serveur PostgreSQL, exploration et restitution des résultats sous Jupyter Notebook. Donne aux analystes une caractérisation reproductible d'un binaire sur un volume d'échantillons croissant. (Python, Jupyter, MinIO, PostgreSQL)



Extraction des binaires d'un disque, stockage objet et calcul d'indicateurs exploitables sous Jupyter.

— **Gestion centralisée des secrets des playbooks de déploiement** : chiffrement par ansible-vault et injection à l'exécution via les variables protégées GitLab CI, afin qu'aucun identifiant ne soit stocké en clair dans les dépôts. (Ansible Vault, GitLab CI)

Environnement technique / outils / logiciels

— **Conteneurs & orchestration** : Kubernetes, Kubespray, RKE2, kind, KubeVirt, Helm, Docker, Temporal
— **Virtualisation** : KVM, QEMU (qcow2), VMware ESXi, API VMware, OpenStack, pilotes VirtIO
— **Infrastructure as Code** : Terraform, Packer, Ansible, Ansible Vault
— **CI/CD** : GitLab CI, registre d'images conteneurisées
— **Observabilité** : Prometheus, Grafana
— **Langages & scripting** : Python, Golang, Rust, C, Bash, PowerShell
— **Données & stockage** : PostgreSQL, MinIO, Jupyter Notebook
— **Systèmes** : Linux (RHEL/Debian), Windows Server
— **Méthodes** : Agile, GitOps, revue de code

2. Ministère des Armées — février 2019 à février 2022

— **Intitulé du poste** : Ingénieur Data & DevSecOps en Cyberdéfense
— **Entreprise / client** : Ministère des Armées
— **Lieu** : Paris
— **Secteur d'activité** : Défense

Contexte

Conception et mise en œuvre d'une chaîne de supervision de sécurité (SIEM) et des pipelines de traitement de données associés, pour les besoins de cyberdéfense du ministère : détection d'incidents sur des systèmes d'information sensibles à partir de la collecte et de la corrélation de journaux hétérogènes (réseau, système, Windows).

Équipe : 14 personnes.

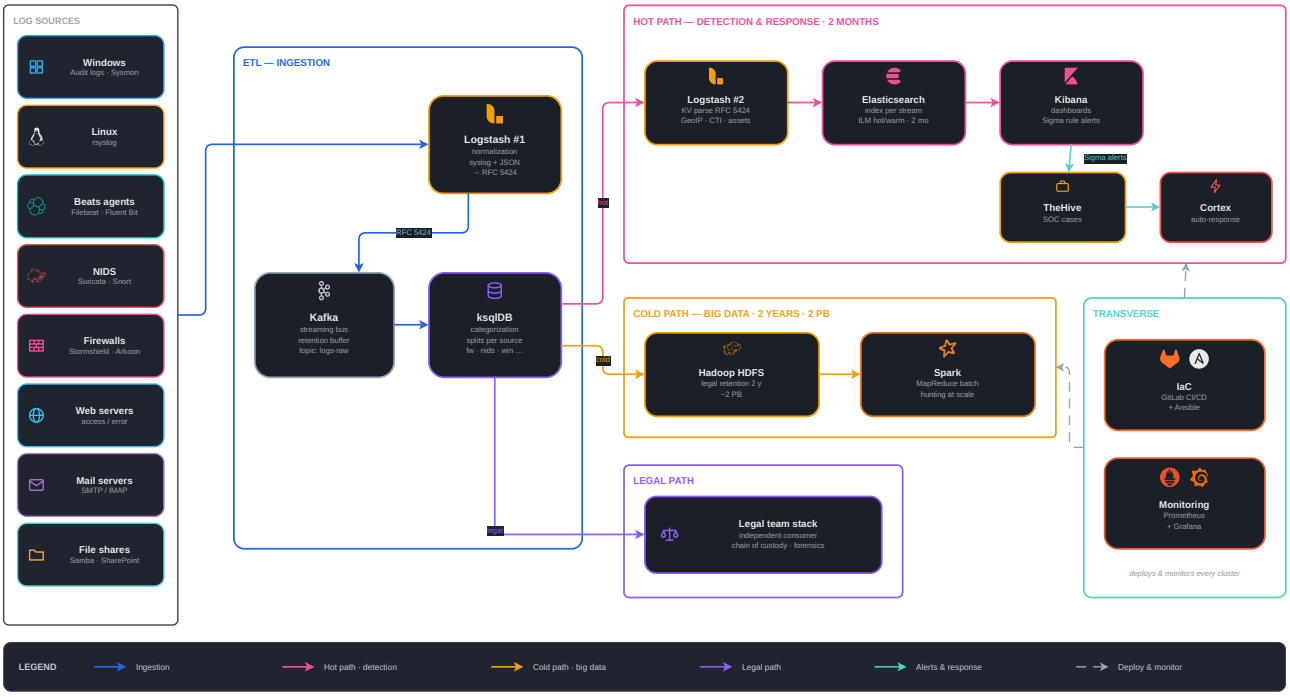
Méthodologie : Agile, sprints de 2 semaines sous Jira.

Objectifs : industrialiser la collecte et le traitement temps réel des événements de sécurité, améliorer la couverture de détection et automatiser le déploiement et la supervision des infrastructures.

Détail des réalisations

Architecture système

— Conception et déploiement d'une architecture de supervision de sécurité (SIEM) sur environnement distribué : sondes réseau et agents système alimentent un bus de messages, les événements sont normalisés puis indexés pour restitution aux analystes.
— Dimensionnement et déploiement des composants de collecte, de traitement et de stockage (4 To de logs par jour).



Architecture de la chaîne de supervision de sécurité : collecte, bus de messages, chemins chaud, froid et légal.

Traitement de données et performance

— Développement et mise en production de pipelines temps réel pour des charges de type streaming : ingestion via Kafka, transformations et enrichissements en continu (ksqlDB, Apache Spark), indexation Elasticsearch et archivage HDFS.

— Optimisation du débit et de la latence de traitement afin de réduire le délai entre l'occurrence d'un événement et sa mise à disposition des analystes. Capacité de mise en cache en cas de problème réseau ou matériel.

Sécurité et conformité

— Mise en œuvre de la détection : règles au format SIGMA, collecte Windows (Sysmon, Windows Security), système (Auditd, OSSEC) et réseau (Suricata).

— Amélioration continue de la couverture de détection en lien avec les analystes SOC.

Infrastructure as Code et supervision

— Automatisation du provisionnement et de la configuration des infrastructures avec Ansible et Docker, versionnées sous GitLab.

— Supervision de l'état des chaînes de traitement (Prometheus, Grafana) pour détecter les pertes de collecte et les ruptures de pipeline.

Collaboration

— Recueil des besoins de détection auprès des analystes SOC et traduction en règles et tableaux de bord.

— Rédaction de la documentation d'architecture et des procédures d'exploitation.

Environnement technique / outils / logiciels

— **Supervision de sécurité** : Suricata, Auditd, OSSEC, Sysmon, Windows Security, SIGMA

— **Données & streaming** : Elasticsearch, Kafka, ksqlDB, Apache Spark, HDFS

— **Automatisation & CI/CD** : Ansible, Docker, GitLab

— **Observabilité** : Prometheus, Grafana

— **Langages** : Python, Bash

— **Systèmes** : Linux, Windows

3. Trusted Labs — mars à septembre 2018 (stage)

— **Intitulé du poste** : Stage — Audit de cybersécurité matérielle
— **Entreprise / client** : Trusted Labs
— **Lieu** : Meudon
— **Secteur d'activité** : Sécurité des composants embarqués, évaluation et certification

Contexte

Laboratoire d'évaluation de la sécurité de composants embarqués et de cartes à puce. Travaux de recherche appliquée sur les attaques matérielles, visant à évaluer la résistance de systèmes embarqués aux attaques physiques.

Équipe : 10 personnes. **Méthodologie** : cycle en V.

Détail des réalisations

— Réalisation d'attaques par canaux auxiliaires électroniques (side-channel) sur systèmes embarqués dans le cadre de tests d'intrusion : acquisition des traces à l'oscilloscope, traitement et analyse statistique des mesures en Python.
— Mise en œuvre d'attaques par injection de faute sur le noyau Android par manipulation du signal d'horloge (attaque de type CLKSCREW), démontrant l'exploitabilité de la faute au niveau logiciel.
— Restitution des résultats sous forme de rapport d'évaluation technique.

Environnement technique / outils / logiciels

Python, Bash, Android (noyau), oscilloscope numérique, banc d'injection de faute, instrumentation électronique, analyse par canaux auxiliaires.

4. Altair — juin à août 2017 (stage)

— **Intitulé du poste** : Stage — Ingénieur d'intégration Cloud
— **Entreprise / client** : Altair
— **Lieu** : Antony
— **Secteur d'activité** : Édition logicielle, calcul haute performance

Contexte

Plateforme de déploiement d'applications de calcul haute performance (HPC) sur clouds publics, permettant l'exécution de charges de calcul intensif sur les principales offres du marché.

Équipe : 10 personnes. **Méthodologie** : Agile.

Détail des réalisations

— Développement d'un connecteur Python assurant l'intégration cloud d'applications HPC : provisionnement des ressources de calcul, soumission des travaux et récupération des résultats sur AWS, Azure et GCP.
— Intégration du connecteur dans la chaîne de build et de tests automatisés (Jenkins), et suivi des développements sous Jira.

Environnement technique / outils / logiciels

Python, Jenkins, CI/CD, AWS, Azure, GCP, Jira, calcul haute performance (HPC).

Formations

1. Master M2 — Sécurité des Systèmes d'Information et des Réseaux (2018)

— **Établissement** : TLS-SEC
— **Lieu** : Toulouse
— **Niveau** : Bac +5, spécialisation post-diplôme

Contenu

Spécialisation en sécurité des systèmes d'information co-accréditée par trois écoles d'ingénieurs toulousaines (INP-ENSEEIHT, INSA Toulouse, ENAC), adossée aux laboratoires LAAS-CNRS, IRIT et ENAC et à des partenaires industriels (Airbus, Thales, CNES, Quarkslab, Cert-IST, ANSSI). Environ 400 heures d'enseignement en neuf unités, à forte dominante travaux pratiques, closes par un stage de six mois et une soutenance. (Cryptographie, Assembleur, Rétro-ingénierie, Malware, Noyau Linux, Débordements mémoire, Sécurité réseau)

2. Diplôme d'ingénieur — Télécommunications et Réseaux (2015 – 2018)

— **Établissement** : ENSEEIHT (Toulouse INP)
— **Lieu** : Toulouse
— **Niveau** : Bac +5, diplôme d'ingénieur

Contenu

Cycle ingénieur menant au diplôme Sciences du Numérique. Tronc commun en programmation impérative et objet, architecture des ordinateurs et systèmes d'exploitation, modélisation formelle, mathématiques appliquées, traitement du signal, communications numériques et réseaux IP ; puis parcours architecture systèmes et réseaux. Troisième année remplacée par la spécialisation TLS-SEC. (Java, C, Assembleur, Protocoles réseau, Cisco, Algorithmes de routage, Transmission du signal)

3. Échange Erasmus — Big Data et cybersécurité (2017)

— **Établissement** : Dublin City University
— **Lieu** : Dublin, Irlande
— **Niveau** : semestre d'échange en deuxième année de cycle ingénieur

Contenu

Semestre d'échange effectué en anglais. (Cryptographie, Forensique, Sécurité réseau, Rétro-ingénierie, Big Data, Hadoop)

4. Classe préparatoire aux grandes écoles (2013 – 2015)

— **Établissement** : Lycée Michelet
— **Lieu** : Vanves
— **Filière** : Physique et Sciences de l'Ingénieur

Langues

— **Français** : langue maternelle.
— **Anglais** : niveau professionnel — semestre universitaire suivi en anglais à Dublin, documentation et échanges techniques quotidiens.