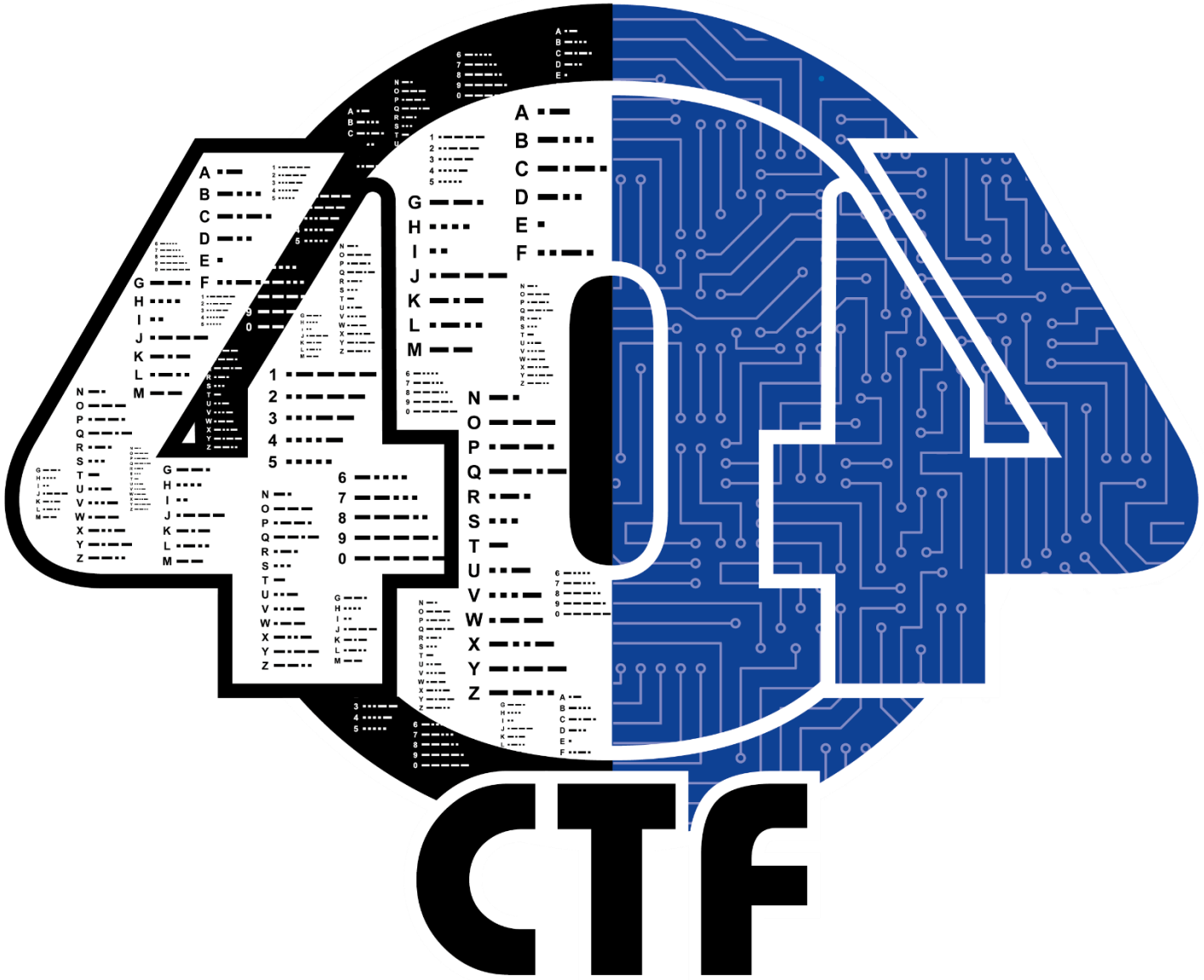
404 CTF : aventures de la sécurisation

Ou comment faire un CTF sans se faire détruire son infra

Le 404CTF

- CTF organisé par HackademINT, Télécom SudParis et la DGSE
- 4 semaines de compétition en mai/juin 2022
- Remise des prix à VivaTech



L’infrastructure

Quels choix ?
Pourquoi ?

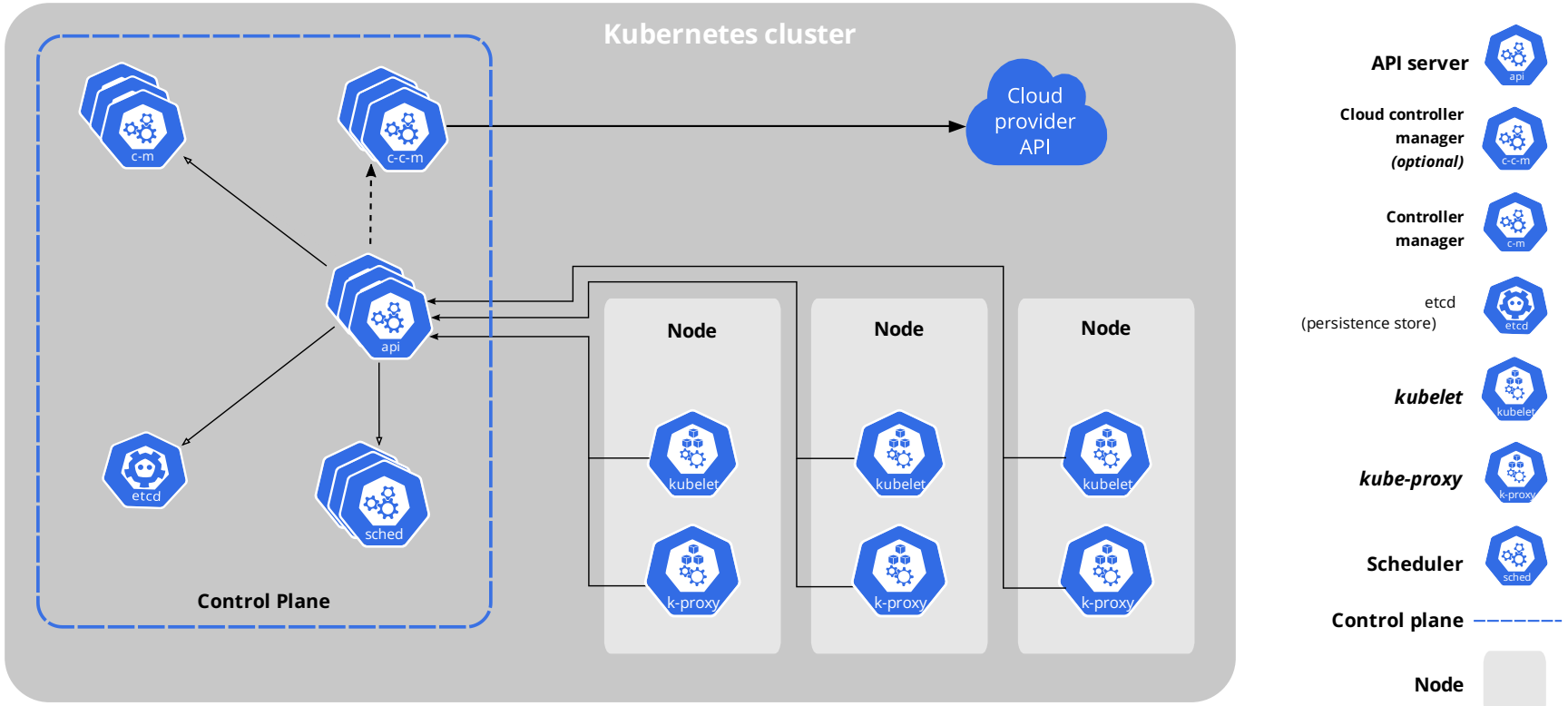
Martin Spiering

Kubernetes

- Pourquoi ?
  - Conteneur
  - Architecture multi-serveurs
  - Orchestration
  - Répartition de charge
  - Autoscaling
  - Déjà de l’expérience avec notre propre infrastructure

Kubernetes

- “Kubernetes, also known as k8s, is an open-source system for automating deployment, scaling, and management of containerized applications.”



Le choix du fournisseur de cloud

- Nos besoins :
  - Européen a minima, idéalement français
  - Kubernetes managé
- Les hébergeurs répondant à nos besoins :
  - OVHCloud ou Scaleway
- Choix final :
  - OVHCloud

Stockage

- Longhorn
  - Allocation sur demande du stockage
  - Triple réplication à travers le cluster

- Sauvegardes sur le stockage S3 d’OVHCloud sur un site différent

Monitoring

- Grafana + Prometheus + Metrics Server :
  - Cluster
  - Nœuds
  - Conteneurs

Sécurisation des conteneurs

- Surtout pas de conteneurs privilégiés
- Ne pas monter le token de l’API k8s
- Séparation des conteneurs en 2 catégories :
  - Présence ou non d’une RCE

Détection du bruteforce

- Problème :
  - SQLmap
  - Outils de fuzzing, de bruteforce, etc.
- Solutions :
  - Filtrage sur les user-agent
  - Rate-limiting

Mise en place de CTFd

Des aventures... Et des réflexions

Quentin Michaud

CTFd, kézako ?

Quand vous faites un CTF, il vous faut :

- Des challenges
- De quoi gérer la validation des challenges : c'est ce que fait CTFd (https://ctfd.io)
- Un projet open-source et libre (licence Apache 2.0)
- Technologies : Python (Flask + Jinja), on maîtrise !

Bref, que demander de mieux ?

Spoiler :

Si tout s’était passé comme prévu, je ne serais pas ici à vous raconter ma vie.

Divers problèmes rencontrés

Traduction de la plateforme en français

- Les textes mélangés entre le HTML pur, les templates Jinja, et le backend Python : un petit enfer à chercher et traduire

  + le code ultra moche en cas de maj !!

Customisation du CSS

- Un système de customisation limité qui ne convenait pas à nos besoins, obligés de replonger dans le code

Le s3 ne fonctionnait pas avec notre provider cloud (OVHcloud) !

- Assurément le problème le plus important, que je vais vous détailler

Pourquoi utiliser s3 ?

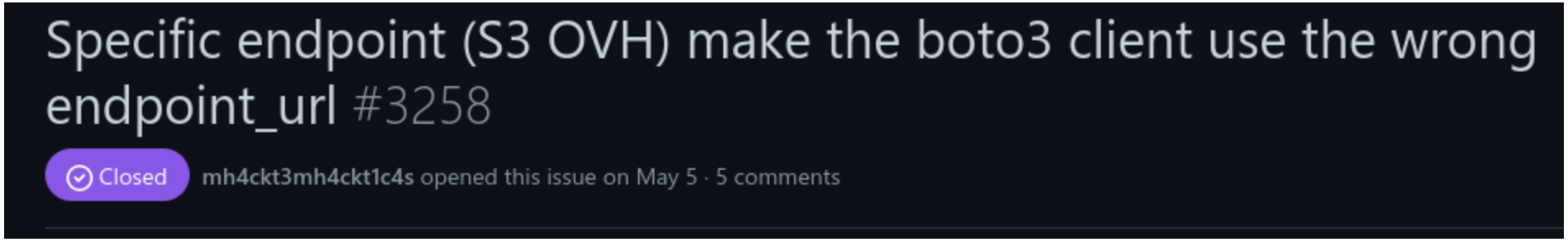
- S3 : Simple Storage Service, créé par Amazon, protocole rendu accessible à tous

- L'idée :
  - Le CTFd, hébergé sur notre serveur, gère les connexions et la validation des challenges
  - Les fichiers des challenges sont hébergés sur le service S3 du CDN d'OVHcloud, afin de ne pas surcharger nos serveurs à causes de téléchargements massifs
  - Lors du téléchargement du challenge via le CTFd, le CTFd s'authentifie auprès du S3 et donne un lien de téléchargement menant vers le CDN

- Problème : cela ne fonctionne pas avec l'API s3 d'OVHcloud !

Aventure du s3 : partie 1

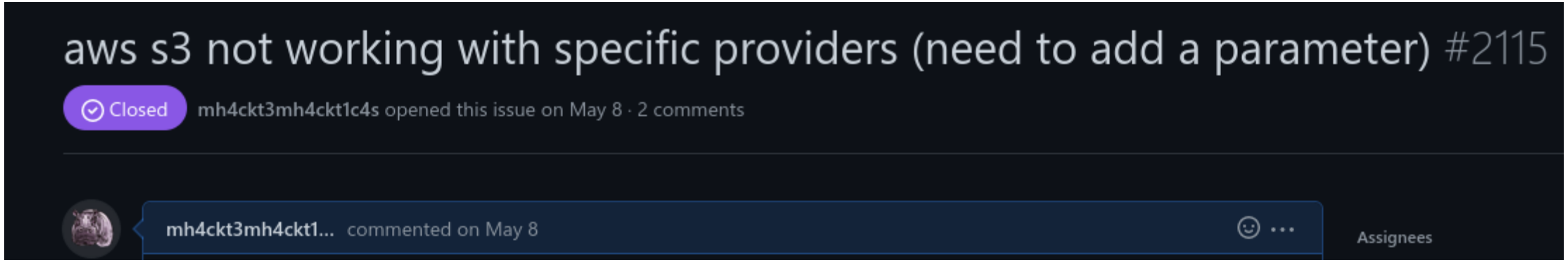
- Trouver le code permettant la connexion au s3
- Identifier la librairie utilisée (boto3)
- Ouvrir une issue sur GitHub
- Petit doute : boto3 est le AWS SDK de Python



- Conclusion : après plusieurs échanges, la librairie n’est pas en cause

Aventure du s3 : partie 2

- Grâce à la conversation avec le développeur AWS, la cause du problème a été identifiée : un paramètre manquant dans la configuration du s3 de CTFd
- Ouvrir une autre issue sur GitHub
- Aucune réponse du développeur…



- On a finalement modifié le code “de manière propre” (en rajoutant la possibilité de choisir le paramètre au lieu de hardcoder la bonne valeur)

Aventure du s3 : partie 3

- Quelques mois plus tard, nous avons fait l'effort de faire une Pull Request pour proposer notre fix

- Qui a été merge rapidement ! (Merci à Hector (SmylerMC) pour son aide)

Des leçons à en tirer sur les FLOSS

- Un exemple des limites de l'open source
- Un projet intéressant, utile et utilisé, MAIS :
  - X Maintenu par une seule personne qui n'arrive pas à absorber la quantité de travail (plus de 200 issues ouvertes)
  - ➢ Contre-productif et contraire aux idées de collaboration portées par les logiciels libres
  - X Une collaboration peu encouragée
  - ➢ Cela n'attire pas les contributions, alors que les utilisateurs en sont largement capables !
  - X Des bonnes pratiques qui ne se retrouvent pas appliquées
  - ➢ Exemple de la gestion des droits admins
  - X "given enough eyes, all bugs are shallow" (loi de Linus)
  - ➢ Hélas un contre-exemple…

Sécurisation via nsjail

Ou comment ne pas se faire pwn son chall de pwn

Matthieu Touloucanon

Vous avez dit “chall de pwn” ?

- Conteneur Docker
  - Serveur écoutant en TCP
  - Binaire vulnérable
  - RCE non contrôlée

- Fuite du conteneur, compromission du cluster ?

Déni de service

- Fork bombs
- Trop nombreuses connexions
- Suppression de fichiers
- Lecture forcée dans les pty
- Téléchargement de fichiers
- Utilisation malveillante du serveur
- ... et autres commandes possibles
  - Potentielle sortie du conteneur

Nsjail

- Namespaces
- Cgroups
- Mounts
- Interfaces réseaux isolées
- Peut agir comme socat !
- Nécessite donc privileged et seccomp unconfined

Attention

- Privileged
  - Configuration parfaite -> documentation
- Unconfined
  - Whitelist avec Kafel

- Vérifier tous les Dockerfiles, fournir un modèle fiable

“Le chall est down”

- Plusieurs répliques k8s
  - Fonctionnement aléatoire
  - Investigations rapides -> rien / incompréhensible

- Investigation avec strace
  - “*ce qui bloque nsjail apparemment : clone(child_stack=0x55c7aeac7220, flags=CLONE_NEWNS|CLONE_NEWCGROUP|CLONE_NEWUTS|CLONE_NEWIPC|CLONE_NEWUSER|CLONE_NEWPID|CLONE_NEWNET|SIGCHLD) = ? ERESTARTNOINTR (To be restarted)*” – Moi à 2h du matin

Un problème complexe

- Un bug kernel
  - Complexité calculatoire création namespace CLONE_NEWNET
- Màj kernel impossible
- Mitiger le bug
  - Trouver le bug
  - Corriger le bug
  - Modifier tous les challenges concernés

“Le chall est up” – Tout le monde