

TP dirigé: Améliorer son portfolio avec des projets Incident et Gestion du Parc informatique

Formateur: Boris Rose

26 Novembre 2025

Consignes générales

- Chaque projet doit être fait un dépôt Github ou Gitlab
- Chaque projet doit contenir un fichier README.md devenant contenir:
 - Une explication du projet
 - Une analyse sans faute d'orthographe au vue de notre matière : la gestion du parc informatique et des incidents et demandeurs utilisateurs.
 - Une section en anglais expliquant le projet

Dépôt GoogleDrive du cours:

<https://drive.google.com/drive/folders/1F2XaSgW-MXi0l1sAx17JGF3LpZyBN4oN>

Projet n°1 : Packet Tracer

Faire une topologie simple:

- 4 hôtes
 - 1 dans un VLAN_IT
 - 2 dans un VLAN_MARKETING
 - 1 dans un VLAN_DIRECTION
- Un serveur WEB (appartient au VLAN_SERVICES)
- Un serveur DHCP / DNS (appartient au VLAN_CORE_SERVICES)
- Un serveur de Messagerie (appartient au VLAN_SERVICES)
- un switch L3
 - Il ne fait PAS le routage inter-vlan
 - Il définit des VLANs
 - Il a la responsabilité de l'établissement du port-security (facultatif)
- un routeur pour l'organisation en architecture router-on-a-stick
 - Il a la responsabilité du routage inter-vlan
 - Il a la responsabilité de l'implémentation des politiques ACLs (pare-feu) (facultatif)
- un routeur faisant office d'Internet connecté à l'autre routeur
- un serveur web derrière le routeur faisant office d'Internet
- Dans un dossier en local (ex: packet-tracer-gpi) que vous transformez en dépôt git

`git init`

- Vous créez un dossier topologie-1. Dans ce dossier:
 - * un dossier diagrams
 - Représenter la topologie que vous allez mettre en place dans PT grâce à l’extension VSCode Draw.io et enregistrer le fichier en png
 - Rendre visible dans le README.md cette image de la topologie mise en place dans PT
 - * un dossier switches
 - * Dans le dossiers switches
 - Un dossier multilayer-switch
 - un fichier vlans.config
 - un fichier interfaces.config
 - * un dossier router
 - un fichier subinterfaces.config
 - un fichier global.config
 - * un dossier hosts

Je pense que vous avez compris où je veux en venir:

- Chaque type de configuration doit avoir son fichier où vous écrivez les commandes CLI que vous avez tapées dans le CLI de PT

Ensuite, pour l’analyse dans le fichier README.md:

- Vous devez analyser votre topologie à la lumière de ISO 27001, 2000 et ITIL
 - Quels sont les incidents et les demandeurs utilisateurs qui peuvent résulter de cette topologie -
 - Quels sont les risques et les vulnérabilités qui peuvent résulter des cette topologie
 - Que peut-on mettre en oeuvre comme changement dans les configurations pour résoudre les problèmes sous-jacents ?

Projet n°2: Cartographie/ Scénario Figma

En vous aidant des différents exemples que vous trouverez dans la Figma ci-joint, créer un scénario avec obligation:

- Des actifs matériels
- Des actifs logiciels (obligatoirement: GLPI, Zabbix)
- Des actifs organisationnels
- Des actifs informationnels
- Des actifs humaines (vous utiliserez le plugin Figma User Profile)
- Des incidents
- Des demandeurs utilisateurs
- Des références aux normes ISO et ITIL
- Des blocs de code pour des configurations réseau
- Des flèches de couleur exprimant:
 - l’appartenance (bleu)
 - la liaison physique (violet)
 - la liaison logique (orange)
 - la décoration (gris)

Tout élément placé sur le diagramme doit être représenté sous forme de **tableau**. En fonction de votre niveau en réseau et en programmation vous ajouterez:

- des configurations réels d'équipements réseaux
- des représentations des différents actifs sous forme de code

Exemples:

```
conf t
    interface GigabitEthernet0/0/1
    ip address 192.168.200.1 255.255.255.0
    no shutdown
exit

// C# exemple

class User
{
    string passwordHash;
    string email;
    string firstname;
    string lastname;

    public string Password
    {
        get{ return passwordHash; }
        set{ passwordHash = value; }
    }

    public User (string firstname, string lastname, string email)
    {
        this.firstname = firstname;
        this.lastname = lastname;
        this.email = email;
    }
}
```

Exemples

Scénario C1: Deux établissements: Tunnel GRE, EtherChannel 3 et Radius

<https://www.figma.com/board/7tc1gkdVbMGXBP06ucIDCu/Sc%C3%A9narios?node-id=1919-5759&t=Jh4QR9cLaSpo8idP-1>

Projet n°3: Conteneuriser un service GLPI

Dans un dépôt git en local appelé docker-gpi:

- créer un fichier README.me expliquant:
 - comment déployer le service GLPI grâce au fichier docker-compose.yaml que vous aurait écrit
 - la marche à suivre pour installer les agents GLPI sur les machines à inventorier (pour linux, macos et windows os) (vous pouvez joindre des captures d'écran en plus des commandes bash ou powershell utilisées)
- Expliquez également l'intérêt de faire un inventaire des éléments du parc informatique
- Ecrire le fichier de configuration docker-compose.yaml
- Tester le fichier de configuration
- Ecrire des SLAs avec des SLOs et les SLIs associés à la gestion des demandes utilisateurs et incidents dans l'organisation
- Appliquer les bonnes pratiques (.env)

Projet n°4 Conteneuriser un service Zabbix

Dans un dépôt git en local appelé docker-zabbix:

- créer un fichier README.me expliquant:
 - comment déployer le service Zabbix grâce au fichier docker-compose.yaml que vous aurait écrit et comment l'associer au service GLPI
 - la marche à suivre pour installer les agents Zabbix sur les machines à inventorier (pour linux, macos et windows os) (vous pouvez joindre des captures d'écran en plus des commandes bash ou powershell utilisées)
- Expliquez également l'intérêt d'utiliser conjointement Zabbix et GLPI
- Ecrire le fichier de configuration docker-compose.yaml
- Tester le fichier de configuration
- Appliquer les bonnes pratiques (.env)