FISE A5
Informatique
Cybersécurité

Livrable 1 : Infrastructure

DESGRANGES Thomas
FALIGOT Clémence
NGUYEN Frédéric
SUBTS Yann

CESI
ÉCOLE D'INGÉNIEURS

SOMMAIRE

DESGRANGES Thomas
FALIGOT Clémence
NGUYEN Frédéric
SUBTS Yann

INFRASTRUCTURE

Contraintes

Pour rappel, nous avions à notre disposition ce matériel :

- Labo Cisco (1 routeur, 2 switchs, borne Wifi, un téléphone VoIP)
- 1 UTM Stormshield SN210
- 1 Serveurs DELL : Smart Value Flexi | R230 | 4x3.5" | E3-1230v6 | 2x8GB | 2x1TB 7.2K SATA | H330 | iDRAC8 Express
- 2 PC (clients)
- Câbles d'alimentation divers et Ethernet
- 3 écrans
- 1 accès internet : débit & opérateur variable (l'accès en lui-même n'est PAS à intégrer dans l'architecture, il est seulement utilisé pour télécharger les informations et/ou données utiles)

De plus, nous avions également quelques contraintes pour l'architecture et les réseaux :

- Authentification, autorisation et traçabilité centralisée (annuaire Active Directory sous Windows Server 2012 R2 sera obligatoire)
- Respect de la RGPD autant que possible
- Système de messagerie (de votre choix)
- Partage de fichiers cloisonné pour tous les collaborateurs (sous une distribution LINUX de votre choix).
- Filtrage applicatif et d'URL (UTM Stormshield SN210 dispo + système de votre choix si besoin)
- 3 accès Wifi : Interne, Admin & Invités (l'accès invité se fera avec une clé WPA2/PSK)
- Service VoIP disponible (système de votre choix)
- Une architecture hiérarchique sera demandée (au moins un cœur de réseau et une couche accès avec les matériels CISCO dispo)
- Les postes clients seront sous Windows 10 (version 1909 ou ultérieure)
- Une suite logicielle complète pour un poste de travail fonctionnel (lecteur PDF, suite bureautique LibreOffice, logiciel de compression/décompression, lecteur de vidéo, client de messagerie de votre choix, et tout autre software que vous jugerez utiles d'implémenter)

Choix

Système de messagerie

Ayant principalement des postes clients Windows et souhaitant une solution de messagerie professionnelle complète, nous avons choisis **Postfix**. En effet, il est conçu pour s'intégrer avec Active Directory, permettant ainsi une gestion centralisée. De plus, Postfix propose des fonctionnalités de sécurité robustes, comme la détection des menaces, le filtrage anti-phishing, le chiffrement des e-mails, la prévention de la perte de données (DLP), et la conformité aux réglementations de sécurité.

Partage de fichiers

Il nous est demandé de mettre en place un partage de fichiers cloisonné sous une distribution Linux, nous pouvons donc utiliser **Samba**, un logiciel open source permettant de partager des fichiers et des imprimantes entre systèmes Linux et Windows. Les postes clients étant sous Windows, c'est une contrainte à respecter également. De plus, Samba est compatible avec Active Directory et peut être intégré à un domaine Windows pour simplifier la gestion des utilisateurs et des groupes. Samba permet également de chiffrer les communications entre les clients et le serveur, de gérer des droits d'accès et des autorisations, et prend également en charge le VPN pour des accès distants sécurisés.

Filtrage applicatif et d'URL

Nous avons choisi **pfSense**, c'est une puissante solution de pare-feu et de routage qui peut aider à sécuriser notre réseau. Une configuration appropriée et une gestion continue sont essentielles pour tirer le meilleur parti de cet outil et garantir la sécurité de vos données et de votre infrastructure réseau.

VoIP

Pour mettre en place le service VoIP, nous avons choisis **FreePBX** qui peut être intégré à Active Directory, ce qui simplifie la gestion des utilisateurs et des extensions VoIP. Ainsi, il offre une variété de fonctionnalités VoIP, notamment les appels entrants et sortants, la messagerie vocale, la conférence téléphonique, l'IVR (Interactive Voice Response), la surveillance des appels, etc.

Représentation

Ainsi, notre infrastructure suivra le schéma suivant :

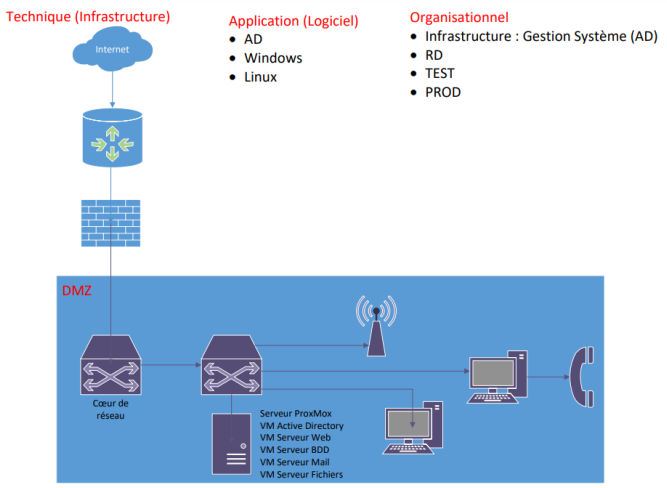

Figure 1 - Schéma de l'infrastructure

DESGRANGES Thomas
FALIGOT Clémence
NGUYEN Frédéric
SUBTS Yann

Configuration

Plan d'adressage

Nom	Réseau	Début de plage	Fin de plage	Broadcast	Hosts	Netmask
LAN	10.0.0.0/22	10.0.0.2	10.0.3.254	10.0.3.255	1022	255.255.252.0
Serveur Proxmox	10.0.0.0/27	10.0.0.2	10.0.0.30	10.0.0.31	30	255.255.255.224
Borne Wifi	10.0.0.0/24	10.0.2.2	10.0.2.254	10.0.2.255	254	255.255.255.0
SSID ADMIN (VLAN 1)	10.0.2.0/26	10.0.2.2	10.0.2.62	10.0.2.63	62	255.255.255.192
SSID INTERN (VLAN 2)	10.0.2.0/26	10.0.2.64	10.0.2.124	10.0.2.125	62	255.255.255.192
SSID GUEST (VLAN 3)	10.0.2.0/26	10.0.2.126	10.0.2.186	10.0.2.187	62	255.255.255.192
PC	10.0.3.0/27	10.0.3.2	10.0.3.30	10.0.3.31	30	255.255.255.224
Téléphone IP	10.0.3.0/27	10.0.3.32	10.0.3.50	10.0.3.51	30	255.255.255.224

VM serveur	IP
VM - AD	10.0.0.2
VM - Web	10.0.0.3
VM - Mail	10.0.0.4
VM - VoIP	10.0.0.5
VM - BDD	10.0.0.6
VM - File	10.0.0.7

VLAN	Nom
10	ADMIN
20	INTERN
30	GUEST

Routeur

La configuration minimale du routeur est la suivante :

```
configure terminal
hostname R1-G1
service password-encryption
service unsupported-transceiver
interface GigabitEthernet0/0/0
ip address 10.0.0.2 255.255.252.0
no shutdown
exit
ip default-gateway 10.0.0.1
line con 0
exec-timeout 15 0
logging synchronous
line vty 0 4
exec-timeout 15 0
```

```
login local
transport input ssh
line vty 5 15
exec-timeout 15 0
login local
transport input ssh
end
write memory
copy running-config startup-config
```

Ensuite pour ce qui est du routage, la configuration est la suivante :

```
conf t
ip default-gateway 10.0.0.1
interface GigabitEthernet0/0/0
no shutdown
no ip routing
interface GigabitEthernet0/0/1
ip address 10.0.0.2 255.255.252.0
no shutdown
ip routing
router ospf 1
network 10.0.0.0 0.0.3.255 area 100
interface GigabitEthernet0/0/1
ip ospf 1 area 100
```

Firewall

Une fois l'installation de Pfsense réalisée, nous avons commencé par configurer les interfaces réseau en fonction de notre topologie réseau. En spécifiant quels ports Ethernet sont connectés à Internet, au LAN interne, et éventuellement à d'autres réseaux. Nous avons défini des règles de pare-feu pour contrôler le flux de trafic, et bloqué tout le trafic non nécessaire et autorisé uniquement les ports et protocoles essentiels.

Switch (cœur de réseau)

La configuration du switch nous servant de cœur de réseau est la suivante :

```
configure terminal
hostname HEART-G1
vlan 1
name VLAN-ADMIN
exit
vlan 2
name VLAN-ADMIN
exit
vlan 3
name VLAN-ADMIN
exit
service password-encryption
no errdisable detect cause gbic-invalid
```

```
service unsupported-transceiver
interface Vlan 1
ip address 10.0.0.4 255.255.252.0
exit
ip default-gateway 10.0.0.1
line con 0
exec-timeout 15 0
logging synchronous
line vty 0 4
exec-timeout 15 0
login local
transport input ssh
line vty 5 15
exec-timeout 15 0
login local
transport input ssh
end
write memory
copy running-config startup-config


reload
```

Switch

La configuration du switch est la suivante :

```
configure terminal
hostname SW1-G1
vlan 1
name VLAN-ADMIN
exit
vlan 2
name VLAN-ADMIN
exit
vlan 3
name VLAN-ADMIN
exit
service password-encryption
no errdisable detect cause gbic-invalid
service unsupported-transceiver
interface Vlan 1
ip address 10.0.0.5 255.255.252.0
exit
ip default-gateway 10.0.0.1
line con 0
exec-timeout 15 0
logging synchronous
line vty 0 4
exec-timeout 15 0
login local
transport input ssh
line vty 5 15
```

```
exec-timeout 15 0
login local
transport input ssh
end
write memory
copy running-config startup-config


reload
```

Avec une répartition des VLAN sur les différents ports, comme suit :

	Ports	VLAN
FastEthernet	1	1 - ADMIN
FastEthernet	2	1 - ADMIN
FastEthernet	3	1 - ADMIN
FastEthernet	4	1 - ADMIN
FastEthernet	5	2 - INTERN
FastEthernet	6	2 - INTERN
FastEthernet	7	2 - INTERN
FastEthernet	8	2 - INTERN
FastEthernet	9	2 - INTERN
FastEthernet	10	2 - INTERN
FastEthernet	11	2 - INTERN
FastEthernet	12	2 - INTERN
FastEthernet	13	2 - INTERN
FastEthernet	14	2 - INTERN
FastEthernet	15	2 - INTERN
FastEthernet	16	2 - INTERN
FastEthernet	17	2 - INTERN
FastEthernet	18	2 - INTERN
FastEthernet	19	2 - INTERN
FastEthernet	20	3 - GUEST
FastEthernet	21	3 - GUEST
FastEthernet	22	3 - GUEST
FastEthernet	23	3 - GUEST
FastEthernet	24	Trunk (wifi)
GigabitEthernet	1	Trunk
GigabitEthernet	2	Trunk

Sécurité

Pour la sécurité des différents équipements (Routeur, switch), nous avons ajouté des mots de passe dans un premier temps :

```
conf t
service password-encryption
line con 0
```

```
exec-timeout 0 0
password <password>
logging synchronous
login
exit
enable algorithm-type scrypt secret <secret>
```

Nous avons choisi « scrypt » pour chiffrer le mot de passe nous permettant de passer en mode « enable » pour plus de sécurité. En effet, par défaut avec la commande « service password-encryption », Cisco chiffre les mots de passe avec un simple algorithme de substitution et de masquage, facile à contourner. Toutefois, SCrypt permet de mieux résister aux attaques par force brute.

Serveur

Pour le serveur, nous avons choisis **Proxmox** comme OS de départ. Proxmox est une plateforme de virtualisation open-source qui combine la virtualisation basée sur conteneurs (via LXC) et la virtualisation matérielle (via KVM) dans une seule interface conviviale. Il dispose d'une interface web de gestion. Une fois connecté à l'interface webnous pouvons créer des machines virtuelles (VMs) et des conteneurs. Nous pouvons également choisir entre la virtualisation basée sur conteneurs, idéale pour des applications légères et la virtualisation matérielle, qui offre une isolation complète pour des systèmes d'exploitation complets. Proxmox offre également des fonctionnalités avancées telles que la migration en direct des VMs, le stockage distribué via Ceph, la gestion des sauvegardes, la planification de tâches, etc.

PLATEFORME

Sécurité

Wireshark

Tout d'abord, nous avons mis en place **Wireshark**. En effet, c'est un outil de capture et d'analyse de paquets réseau. Ainsi, lorsque des problèmes surviennent sur un réseau, Wireshark permet aux administrateurs réseau de capturer le trafic en temps réel et d'analyser les paquets pour identifier les problèmes. Cela peut inclure des problèmes de latence, de perte de paquets, de congestion, etc. Nous pouvons également utiliser Wireshark pour examiner le trafic réseau afin de détecter des activités suspectes, des intrusions ou des tentatives d'exploitation de vulnérabilités. Cela aide à renforcer la sécurité des réseaux et des systèmes. Il peut être utilisé aussi pour surveiller le trafic réseau en temps réel ou pour effectuer des analyses historiques. Cela permet de suivre l'utilisation du réseau, de détecter les anomalies et de planifier la capacité du réseau.

Architecture Logicielle

Serveur Proxmox

Le serveur Proxmox est l'hyperviseur principal qui gère la virtualisation. Il assure la gestion des ressources matérielles telles que le processeur, la mémoire RAM, le stockage, et les interfaces réseau pour les VMs.

VM Ubuntu

Cette VM héberge notre application web et notre base de données. Elle est également configurée avec un serveur de messagerie Postfix et Dovecot pour gérer les e-mails de l'organisation.

VM Windows Server

Cette VM exécute Windows Server et sert de contrôleur de domaine Active Directory (AD) et de serveur LDAP. Elle gère l'authentification et l'autorisation des utilisateurs dans l'ensemble du réseau.

VM pfSense

La VM pfSense est dédiée à la sécurité du réseau. Elle agit comme un pare-feu et un routeur, fournissant une protection contre les menaces et une gestion du trafic réseau.

VMs Windows 10

Ces VMs Windows 10 servent de poste de travail pour les utilisateurs finaux.

Elles sont intégrées dans le domaine Active Directory configuré dans la VM Windows Server pour une gestion centralisée.

Ainsi, cette architecture permet une isolation des services et une gestion centralisée. Chaque VM peut être configurée en fonction de ses besoins spécifiques, tandis que le serveur Proxmox offre la flexibilité de répartir les ressources selon les exigences de charge de travail.

APPLICATION

Backend

Le Backend est composé de 3 micro-services et d'une API. Il est codé en Node.js

API

C'est l'API qui va être chargée de faire le lien entre l'application (frontend), et les micro-services.

De ce fait, il est important de sécuriser son accès.

Pour cela, nous avons utilisé plusieurs moyens. Tout d'abord, nous avons utilisé la bibliothèque Cors (Cross-origin resource sharing) qui permet d'ajouter des entêtes dans les requêtes afin de permettre à un utilisateur d'accéder à des ressources situées sur une origine différentes.

Nous avons aussi mis-en-place un système d'authentification, afin de nous assurer que seuls les utilisateurs autorisés puissent accéder aux ressources. Pour cela, nous avons utilisé LDAP, qui permet de se connecter à un Active Directory et de communiquer avec. De cette manière, nous pouvons gérer les utilisateurs avec l'AD et leur permettre d'utiliser l'application en se connectant à celle-ci via leur compte utilisateur.

Autre point de sécurité, l'API n'envois que des données précises afin d'éviter les risques d'injection SQL.

Enfin, un système de chiffrement a été mis en place. Les données envoyées à l'API sont chiffrées avant d'être envoyées à la base de données. Elles sont ensuite déchiffrées lorsqu'elles sont appelées par l'API. De cette manière, si une personne extérieure venait à accéder à la base de données, aucune donnée ne sera en clair.

Micro-services

La mission des micro-services est du communiquer avec la base de données. Ces micro-services utilisent des requêtes SQL pour communiquer avec la base et poster et récupérer les données.

Les micro-services possédant des informations permettant d'accéder directement à la base de données, ils sont isolés du reste. Ils ne peuvent recevoir des requêtes que de l'API avec la bibliothèque Cors qui bloque les requêtes d'origine autre.

Chiffrement

Le cahier des charges nous impose de construire notre système de chiffrement des communications avec au moins un chiffrement par transposition et au moins deux substitutions

différentes. Les échanges entre un client et serveur, sur la partie applicative qui nous est demandé de développer, sont fait de manière confidentielle (donc communications chiffrées).

Afin de respecter les contraintes, nous avons choisis de faire un chiffrement par transposition (par inversion de l'ordre) et deux chiffrements par substitutions qui possèdent deux dictionnaires de substitutions différents.

Les données envoyées à l'API sont chiffrées par deux chiffrements par substitutions puis par un chiffrement de transposition avant d'être envoyées à la base de données. Elles sont ensuite déchiffrées lorsqu'elles sont appelées par l'API. De cette manière, si une personne extérieure venait à s'introduire dans la base de données, aucune donnée ne serait en clair.

Lors de l'élaboration de cette sécurisation, nous avons eu la possibilité d'implémenter un système qui utilise un chiffrement par clé secrète qui est plus robuste en termes de sécurité mais nous serions hors contexte par rapport au cahier des charges.

Frontend

Le frontend utilise le framework Vue.js. Il s'agit d'un framework qui travaille avec un système de vues, de composants et de routes pour naviguer.

L'avantage d'utiliser ce framework comparé à du HTML/CSS et Javascript purs est que nous pouvons plus facilement gérer les autorisations. Dans notre situation, nous regardons si l'utilisateur est connecté et quel est son rôle. S'il n'est pas connecté, il est renvoyé vers la page de connexion et les autres pages ne sont accessibles. Le rôle permet de déterminer quels pages sont accessibles à l'utilisateur.

Nous avons 3 pages principales (en plus de la page de connexion) :

- Freezbe : Qui permet aux utilisateurs de gérer les freezbes.
- Ingrédients : Qui permet aux utilisateurs de gérer les ingrédients.
- Procédés de Fabrication : Qui permet aux utilisateurs de gérer les procédés.

Avec cela nous avons 3 rôles avec des accès différents :

- SCH_RD : Qui peut accéder à toutes les pages.
- SCH_TEST : Qui peut accéder à la page des procédés.
- SCH_PROD : Qui peut accéder à la page Freezbe.

Vue.js utilise un système de routeur, auquel nous pouvons ajouter des conditions. Dans notre cas, nous avons mis des rôles à chaque page nécessitant un rôle, de façon à ce qu'un utilisateur ne puisse pas accéder aux ressources qui ne sont pas pour lui.