Projet – R4.B.11 – R4.B.12 – AL4.B.01

Introduction

L'objectif de ce projet est la mise en place d'une infrastructure réseau segmentée et sécurisée pour l'entreprise Reseau4Ever. Pour garantir une isolation optimale des flux et une bonne gestion de la sécurité, nous avons opté pour une architecture articulée autour de trois zones distinctes (VLANs) portées par un routeur central OpenWrt et filtrées par une sonde IDS Debian.

Sommaire

Partie 1 – Réseaux

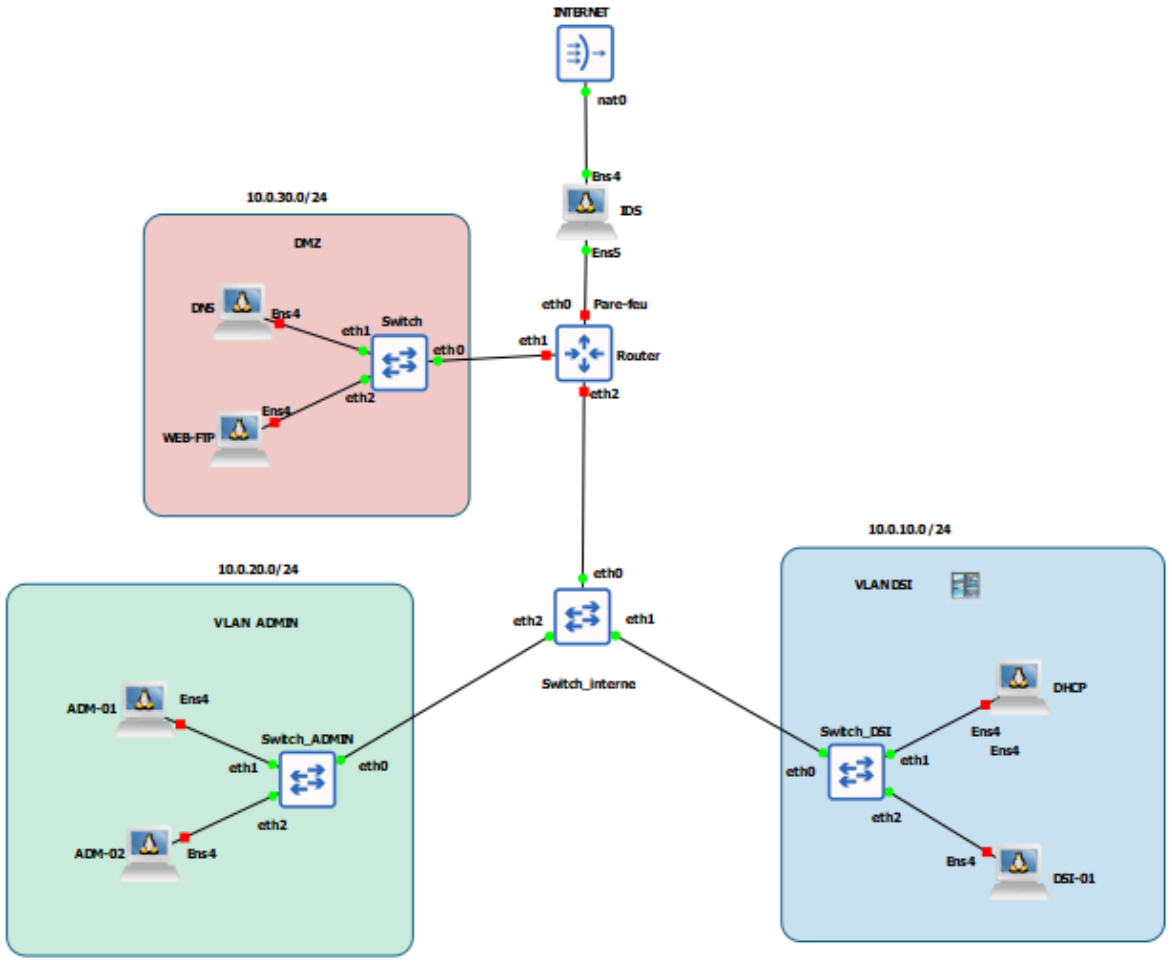
1. Architecture et plan d'adressage

1.1 Segmentation en zones réseau

L'infrastructure est découpée en trois zones logiques distinctes, auxquelles s'ajoute un réseau de liaison dédié entre la machine IDS et le routeur :

- VLAN 10 — DSI : zone technique hébergeant le serveur DHCP et le poste d'administration DSI-01. C'est le réseau de confiance maximale.
- VLAN 20 — Admin : zone réservée aux postes clients administratifs (ADM-01, ADM-02). Accès limité à la DMZ, sans visibilité sur la DSI.
- DMZ (interface dédiée) : zone de services exposés hébergeant le DNS, le serveur Web/HTTPS et le FTP. Isolée des réseaux internes par politique de pare-feu.
- Réseau de liaison 10.10.10.0/24 : segment point-à-point entre la machine Debian IDS et l'interface WAN du routeur. Délibérément séparé des réseaux utilisateurs.

La DMZ est connectée directement au routeur via une interface physique dédiée (eth1), sans passer par le switch interne des VLANs. Cela garantit que le trafic entrant depuis Internet vers la DMZ ne transite jamais par les segments internes, même en cas de mauvaise configuration du switch.



1.2 Plan d'adressage IPv4 et IPv6

L'espace IPv4 général retenu est 10.0.0.0/16, découpé en sous-réseaux /24 pour chaque zone. L'IPv6 utilise le préfixe fc01::/56, subdivisé en /64 par segment. L'adressage IPv4 des hôtes est distribué dynamiquement par DHCP ; l'adressage IPv6 repose sur SLAAC.

Zone	Sous-réseau IPv4	Passerelle IPv4	Préfixe IPv6 (/64)	Serveur / Note clé
DSI (VLAN 10)	10.0.10.0/24	10.0.10.1	fc01:0:0:10::1	DHCP : 10.0.10.254
Admin (VLAN 20)	10.0.20.0/24	10.0.20.1	fc01:0:0:20::1	Adressage dynamique
DMZ	10.0.30.0/24	10.0.30.1	fc01:0:0:30::1	DNS : 10.0.30.10 Web/FTP : 10.0.30.20
Liaison IDS–Routeur	10.10.10.0/24	10.10.10.1 (IDS)	fc01:0:0:100::1	Routeur : 10.10.10.2

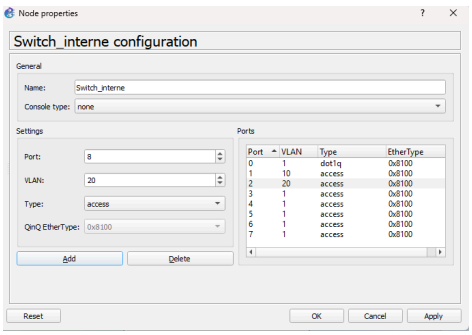
Le choix de /24 par zone offre 254 adresses utilisables par segment, conformément à l'exigence du cahier des charges. La convention d'adressage *.1 pour les passerelles et *.254 pour les serveurs critiques facilite l'identification rapide des équipements lors des phases de diagnostic. Les adresses *.1 à *.9 sont réservées à l'infrastructure et exclues des pools DHCP.

1.3 Topologie physique — switch interne et VLANs

Les réseaux DSI et Admin sont portés par un switch interne GNS3 configuré en 802.1Q. La séparation logique est assurée par l'attribution de chaque port à un VLAN distinct, avec un lien trunk vers le routeur :

- Port 0 (trunk dot1q) → routeur OpenWRT (eth2) : transporte les trames des VLAN 10 et 20 étiquetées.
- Port 1 (access VLAN 10) → machines DSI : les trames entrent et sortent sans étiquette VLAN du point de vue des hôtes.
- Port 2 (access VLAN 20) → machines Admin : même principe pour le réseau administratif.

Le lien trunk unique entre le switch et le routeur est suffisant pour une infrastructure de cette taille. Le routeur crée des sous-interfaces virtuelles (eth2.10, eth2.20) sur ce lien pour traiter chaque VLAN indépendamment. Cette architecture Inter-VLAN routing évite de multiplier les liens physiques entre le switch et le routeur.



2. Machine IDS : passerelle NAT et inspection réseau

2.1 Rôle et positionnement

Une machine Debian est positionnée en amont du routeur OpenWRT, entre Internet et l'ensemble du réseau interne. Elle assume deux fonctions complémentaires : passerelle NAT vers Internet et point d'inspection du trafic (IDS/IPS, détaillé en section 8).

Son interface « côté Internet » est configurée en DHCP afin d'obtenir automatiquement l'adresse publique fournie par le NAT de l'hyperviseur GNS3. Son interface « côté routeur » est configurée en adresse statique (10.10.10.1/24), conformément à la contrainte imposée par le cahier des charges.

Placer la machine Debian en amont du routeur — plutôt que de configurer le NAT directement sur le routeur — respecte le principe de défense en profondeur : l'IDS analyse le trafic brut avant qu'il n'atteigne le pare-feu du routeur. En cas de tentative d'intrusion, la détection intervient dès la première couche, sans impacter les performances du routeur.

2.2 Activation du routage IP

Pour que la machine puisse relayer les paquets entre ses deux interfaces, le forwarding IP doit être activé dans le noyau Linux, pour IPv4 et IPv6. Cette activation est rendue persistante via le fichier /etc/sysctl.conf (paramètres net.ipv4.ip_forward=1 et net.ipv6.conf.all.forwarding=1). Sans cette option, la machine accepterait les paquets destinés à d'autres hôtes mais les rejetterait sans les transmettre.

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
```

2.3 Règle de masquerade (NAT)

Une règle POSTROUTING MASQUERADE est ajoutée dans la table nat d'iptables sur l'interface côté Internet. Elle traduit automatiquement les adresses IPv4 sources privées de l'ensemble du réseau interne en l'adresse publique obtenue par DHCP, permettant à tous les hôtes d'accéder à Internet via une seule adresse IP publique.

Le MASQUERADE a été préféré au SNAT statique parce que l'adresse IP publique de la machine IDS est obtenue dynamiquement par DHCP auprès de l'hyperviseur et peut donc changer. Le MASQUERADE s'adapte automatiquement à tout changement d'adresse, contrairement au SNAT qui nécessiterait une mise à jour manuelle de la règle à chaque renouvellement de bail.

```
debian@debian:~$ sudo iptables -t nat -L -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    1    84 MASQUERADE  all  --  any    enp2s0  anywhere             anywhere
debian@debian:~$
```

2.4 Route retour IPv6 vers les réseaux internes

Pour que la machine IDS puisse répondre aux requêtes provenant des machines internes en IPv6, une route statique vers le préfixe fc01::/56 est ajoutée, pointant vers l'adresse IPv6 du routeur (fc01:0:0:100::2). Cette route est nécessaire car la machine IDS n'est pas le routeur de ces préfixes : elle doit savoir les déléguer vers OpenWRT.

```
#Attribution static sur l'interface communiquant avec le routeur
auto ens1
iface ens1 inet static
        address 10.10.10.1
        netmask 255.255.255.0

iface ens1 inet6 static
    address fc01:0:0:100::1
    netmask 64
    post-up ip -6 route add fc01::/56 via fc01:0:0:100::2
```

```
debian@debian:~$ sudo ip -6 route add fc01::/56 via fc01:0:0:100::2
```

3. Routeur OpenWRT — interfaces, VLANs et dual-stack

3.1 Choix du routeur

L'équipe a retenu OpenWRT comme solution de routage et de pare-feu. Ce choix repose sur plusieurs critères: distribution Linux légère et libre, compatible avec GNS3, offrant nativement la gestion de zones de pare-feu (firewall zones), le relais DHCP, les VLANs 802.1Q et le dual-stack IPv4/IPv6 (SLAAC inclus). Sa documentation est exhaustive et l'équipe dispose d'une bonne maîtrise de la plateforme.

Le cahier des charges laissait le libre choix entre OpenWRT, MikroTik, OPNSense, pfSense ou Fortinet. OPNSense aurait offert une interface graphique plus riche, mais nécessite davantage de ressources en simulation. OpenWRT, plus léger, répond à tous les besoins fonctionnels du projet.

3.2 Configuration des interfaces

```
config interface 'loopback'
        option device 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config globals 'globals'
        option ula_prefix 'fc01::/56'

config device
        option name 'br-lan'
        option type 'bridge'
        list ports 'eth0'

config device
        option name 'eth2.10'
        option type '8021q'
        option ifname 'eth2'
        option vid '10'

config device
        option name 'eth2.20'
        option type '8021q'
        option ifname 'eth2'
        option vid '20'

config interface 'DSI'
        option device 'eth2.10'
        option proto 'static'
        option ipaddr '10.0.10.1'
        option netmask '255.255.255.0'
        list ip6addr 'fc01:0:0:10::1/64'

config interface 'ADMIN'
        option device 'eth2.20'
        option proto 'static'
        option ipaddr '10.0.20.1'
        option netmask '255.255.255.0'
        list ip6addr 'fc01:0:0:20::1/64'

config interface 'DMZ'
        option device 'eth1'
        option proto 'static'
        option ipaddr '10.0.30.1'
        option netmask '255.255.255.0'
        list ip6addr 'fc01:0:0:30::1/64'


config interface 'wan'
        option device 'eth0'
        option proto 'static'
        option ipaddr '10.10.10.2'
        option netmask '255.255.255.0'
        option gateway '10.10.10.1'

config interface 'wan6'
        option device 'eth0'
        option proto 'static'
        list ip6addr 'fc01:0:0:100::2/64'
        option ip6gw 'fc01:0:0:100::1'
```

Le routeur dispose de plusieurs interfaces réseau, chacune correspondant à un segment ou un rôle distinct :

- eth0 (WAN) : interface physique reliée à la machine IDS. Adresse statique 10.10.10.2/24, passerelle 10.10.10.1 (machine IDS). Route par défaut vers l'IDS, qui assure le NAT vers Internet.

- eth0 (WAN6) : même interface physique, adresse IPv6 statique fc01:0:0:100::2/64, passerelle fc01:0:0:100::1 (IDS).

- eth1 (DMZ) : interface physique dédiée à la DMZ. Adresse 10.0.30.1/24 en IPv4, fc01:0:0:30::1/64 en IPv6.

- eth2.10 (DSI) : sous-interface VLAN 10 sur le lien trunk. Adresse 10.0.10.1/24, fc01:0:0:10::1/64.

- eth2.20 (Admin) : sous-interface VLAN 20 sur le lien trunk. Adresse 10.0.20.1/24, fc01:0:0:20::1/64.

Les dispositifs VLAN (eth2.10, eth2.20) sont créés en mode 802.1q sur l'interface physique eth2, permettant de transporter les deux VLANs sur un seul câble physique entre le routeur et le switch interne.

3.3 Adressage IPv6 : SLAAC

Pour l'adressage IPv6, la stratégie retenue est le SLAAC (Stateless Address Autoconfiguration). Le routeur OpenWRT émet des Router Advertisements (RA) sur chaque interface interne (DSI, Admin, DMZ), contenant le préfixe IPv6 du segment. À réception, chaque client Debian combine ce préfixe avec son identifiant d'interface (dérivé de l'adresse MAC) pour s'auto-attribuer une adresse IPv6 unique, sans nécessiter de serveur DHCPv6 dédié.

L'adressage IPv6 a été configuré de manière statique sur chaque interface. Le préfixe global fc01:0:0::/48 a été segmenté manuellement en sous-réseaux /64 (ex: fc01:0:0:30::/64 pour la DMZ). Le routeur annonce ensuite ces préfixes spécifiques via les Router Advertisements (RA) pour permettre l'auto-configuration des hôtes (SLAAC).
Le SLAAC a été préféré à DHCPv6 stateful car il réduit la complexité de l'infrastructure (pas de serveur DHCPv6 supplémentaire) et garantit une connectivité IPv6 immédiate dès la connexion d'un équipement, indépendamment de la disponibilité du serveur DHCP. Cette approche est cohérente avec le cahier des charges qui précise que seul l'IPv4 sera distribué par DHCP.

```
config dhcp 'ADMIN'
    option interface 'ADMIN'
    option ra 'server'              # Active l'envoi des Router Advertisements
    option ra_management '0'        # 0 = SLAAC pur (pas de DHCPv6 pour l'IP)
    option ra_default '1'           # Envoie les options par defaut
    option ra_flags 'none'          # Securite : refuse toute configuration via DHCPv6
    option ignore '1'               # dit a OpenWrt de NE PAS distribuer d'adresses IPv4 automatiquement

config dhcp 'DSI'
    option interface 'DSI'
    option ra 'server'              # Active l'envoi des Router Advertisements
    option ra_management '0'        # 0 = SLAAC pur (pas de DHCPv6 pour l'IP)
    option ra_default '1'           # Envoie les options par defaut
    option ra_flags 'none'          # Securite : refuse toute configuration via DHCPv6
    option ignore '1'               # dit a OpenWrt de NE PAS distribuer d'adresses IPv4 automatiquement

config dhcp 'DMZ'
    option interface 'DMZ'
    option ra 'server'              # Active l'envoi des Router Advertisements
    option ra_management '0'        # 0 = SLAAC pur (pas de DHCPv6 pour l'IP)
    option ra_default '1'           # Envoie les options par defaut
    option ra_flags 'none'          # Securite : refuse toute configuration via DHCPv6
    option ignore '1'               # dit a OpenWrt de NE PAS distribuer d'adresses IPv4 automatiquement
root@OpenWrt:/#
```

4. Service DHCP

4.1 Implantation centralisée sur une machine dédiée

Le service DHCP est hébergé sur une machine Debian dédiée dans le VLAN DSI, à l'adresse fixe 10.0.10.254/24. Le paquet retenu est isc-dhcp-server, solution de référence sous Debian. Ce serveur dessert l'ensemble des VLANs internes (DSI et Admin) grâce au mécanisme de relais DHCP décrit ci-après.
Centraliser le service DHCP sur une machine distincte du routeur présente deux avantages : d'une part, le service reste disponible indépendamment d'une éventuelle reconfiguration ou remplacement du routeur ; d'autre part, la gestion des baux est unifiée dans une seule base de données, ce qui simplifie le diagnostic et la maintenance. La machine hébergeant ce service est impérativement configurée en adresse statique — un serveur DHCP ne peut pas dépendre lui-même du DHCP pour obtenir son adresse.

4.2 Pools d'adresses et options distribuées

Deux plages d'adresses distinctes sont définies dans la configuration du service, une par sous-réseau :

Sous-réseau	Plage dynamique	Passerelle annoncée	DNS annoncé	Domaine
DSI (10.0.10.0/24)	10.0.10.10 - 10.0.10.250	10.0.10.1	10.0.30.10	reseau4ever.iut
Admin (10.0.20.0/24)	10.0.20.10 - 10.0.20.250	10.0.20.1	10.0.30.10	reseau4ever.iut

Les adresses *.1 à *.9 de chaque plage sont réservées à l'infrastructure (routeur, serveurs, équipements réseau). Les options domain-name et domain-name-servers sont distribuées à tous les clients, leur permettant de résoudre les noms du domaine reseau4ever.iut via le serveur BIND9 en DMZ.

```
option domain-name "reseau4ever.iut";
option domain-name-servers 10.0.30.10;

# Serveur DHCP principal
authoritative;

# Sous-réseau DSI
subnet 10.0.10.0 netmask 255.255.255.0 {
    range 10.0.10.10 10.0.10.250;
    option routers 10.0.10.1; # IP de l'OpenWrt
}

# Sous-réseau Admin
subnet 10.0.20.0 netmask 255.255.255.0 {
    range 10.0.20.10 10.0.20.250;
    option routers 10.0.20.1; # IP de l'OpenWrt
}
```

Nous avons également mis en place un subnet pour la DMZ dans le fichier de configuration du DHCP :

Sous-réseau	Plage dynamique	Passerelle annoncée	DNS annoncé	Domaine
DMZ (10.0.30.0/24)	Aucune	10.0.30.1	10.0.30.10	reseau4ever.iut

Les machines du réseau DMZ: DNS, Web et FTP, se voient attribuer une adresse fix en fonction de leur adresse MAC :

Host	Hardware ethernet	fixed-address
Serveur-DNS	0c:a7:3d:ae:00:00	10.0.30.10
Serveur-WEB-FTP	0c:a6:d7:e9:00:00	10.0.30.20

```
# Sous-réseau DMZ (
subnet 10.0.30.0 netmask 255.255.255.0 {
    option routers 10.0.30.1; # IP de l'OpenWrt en DMZ
}

# Adresse static de la machine  DNS
host Serveur-DNS { # Le nom de hote
    hardware ethernet 0c:a7:3d:ae:00:00; # L'adresse MAC
    fixed-address 10.0.30.10; # L'adresse IP à distribuer
}

# Adresse static de la machine Web et FTP
host Serveur-WEB-FTP { # Le nom de hote
    hardware ethernet 0c:a6:d7:e9:00:00; # L'adresse MAC
    fixed-address 10.0.30.20; # L'adresse IP à distribuer
}
```

4.3 Relais DHCP inter-VLAN

Le protocole DHCP repose sur des broadcasts IP qui ne traversent pas naturellement les frontières de VLAN. Or le serveur DHCP étant dans le VLAN DSI, les machines du VLAN Admin ne peuvent pas l'atteindre directement. Un agent de relais DHCP est donc installé sur le routeur OpenWRT via le paquet isc-dhcp-relay-ipv4.

Sur OpenWRT, le relais DHCP ne se configure pas via /etc/rc.local mais via le système UCI, dans le fichier /etc/config/dhcrelay. Ce fichier définit les paramètres du service de relais de manière native et persistante au redémarrage, sans nécessiter de script de démarrage manuel. La configuration précise les interfaces d'écoute (DSI et Admin) ainsi que l'adresse unicast du serveur DHCP cible (10.0.10.254) vers lequel les requêtes broadcast sont retransmises.
L'utilisation du fichier UCI /etc/config/dhcrelay est la méthode recommandée sur OpenWRT pour gérer les services système. Elle garantit l'intégration avec le gestionnaire de services et le démarrage automatique sans intervention manuelle, contrairement à /etc/rc.local qui est une solution de contournement moins robuste.

```
config dhcrelay ipv4
        option 'enabled' '1'

        # IP address of the server
        option 'dhcpserver' '10.0.10.254'

        # network interfaces to listen on (e.g. lan or wan)
        list interfaces 'ADMIN'
        list interfaces 'DSI'
        list interfaces 'DMZ'

        list upstream_interfaces 'DSI'
        list downstream_interfaces 'ADMIN'
        list downstream_interfaces 'DMZ'

        # What to do about packets that already have a relay option:
        # 'append': Forward and append our own relay option
        # 'replace': Forward, but replace theirs with ours (default)
        # 'forward': Forward without changes
        # 'discard': Don't forward
        option relay_mode 'replace'

        # enable RFC3527 link selection sub-option and use the IP address of
        # the specified network interface as "uplink" IP address (e.g. wan)
        option 'link_selection' ''
```

5. Services hébergés en DMZ

La DMZ (10.0.30.0/24) héberge les trois services exposés de l'entreprise : DNS, Web (HTTPS) et FTP. Ces services sont déployés avec leurs variantes sécurisées, conformément aux exigences du cahier des charges.

5.1 Serveur DNS : BIND9 (10.0.30.10)

Choix et rôle

Le serveur DNS est hébergé sur une machine dédiée en DMZ (srv-dns.reseau4ever.iut, 10.0.30.10). Le logiciel retenu est BIND9, référence en matière de serveur DNS autoritaire sous Debian. Il est responsable de la gestion du domaine reseau4ever.iut pour l'ensemble des machines de l'entreprise, en dual-stack (enregistrements A et AAAA).

Configuration et zones

BIND9 est configuré en serveur autoritaire pour la zone reseau4ever.iut, avec les paramètres suivants :

- ACL de confiance : le réseau 10.0.0.0/16 est autorisé à effectuer des requêtes récursives. Les requêtes extérieures à ce périmètre sont refusées, limitant l'exposition du service.
- Forwarders : les résolutions externes (hors domaine local) sont transmises aux résolveurs publics 8.8.8.8 et 9.9.9.9, garantissant la résolution des noms Internet depuis les postes internes.
- DNSSEC : la validation DNSSEC est activée (dnssec-validation auto), renforçant l'intégrité des réponses DNS reçues de l'extérieur.
- Dual-stack : chaque enregistrement de la zone dispose d'un enregistrement A (IPv4) et AAAA (IPv6) pour garantir la résolution dans les deux protocoles.

Les enregistrements définis dans la zone reseau4ever.iut sont les suivants :

Nom	Type A (IPv4)	Type AAAA (IPv6)
srv-dns.reseau4ever.iut	10.0.30.10	fc01:30:ea7:3dff:feae:0
srv-web.reseau4ever.iut	10.0.30.20	fc01:30:ea6:d7ff:fee9:0
srv-ftp.reseau4ever.iut	10.0.30.20	fc01:30:ea6:d7ff:fee9:0

```
debian@debian:~$ cat /etc/bind/reseau4ever.iut
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     srv-dns.reseau4ever.iut. admin.reseau4ever.iut. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      srv-dns.reseau4ever.iut.
@   IN   A   10.0.30.10
@   IN   AAAA    fc01::30:ea7:3dff:feae:0
;  IPs statiques
srv-dns     IN      A       10.0.30.10
srv-dns     IN      AAAA    fc01::30:ea7:3dff:feae:0
srv-web     IN      A       10.0.30.20
srv-web     IN      AAAA    fc01::30:ea6:d7ff:fee9:0
srv-ftp     IN      A       10.0.30.20
srv-ftp     IN      AAAA    fc01::30:ea6:d7ff:fee9:0


;clé dnssec
$INCLUDE /etc/bind/keys/Kreseau4ever.iut.+008+52398.key
$INCLUDE /etc/bind/keys/Kreseau4ever.iut.+008+16861.key
debian@debian:~$
```

```
debian@debian:~$ cat /etc/bind/named.conf.options

acl "zone_confiance" { // Nom de la zone
 10.0.0.0/16; // Réseau distant de confiance
 localhost; // Confiance a soi meme
 localnets;
};

options {
 directory "/var/cache/bind"; // Répertoire de travail
 forwarders { // Resolvers externes
   8.8.8.8; // Google
   9.9.9.9; // Quad9
 };
 // activer la recursivité pour noms externes
 recursion yes;
 listen-on { any; }; // Ecouter sur toutes les interfaces v4
 listen-on-v6 { any; }; // v6
 allow-query { zone_confiance; };
 dnssec-validation auto;
};
debian@debian:~$
```

```
debian@debian:~$ cat  /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "reseau4ever.iut" {
    type master;
    file "/etc/bind/reseau4ever.iut.signed";
};
debian@debian:~$
```

Nous pouvons voir que le serveur DNS et la résolution des noms de domaines fonctionnent bien.

```
debian@debian:~$ nslookup srv-dns.reseau4ever.iut 10.0.30.10
Server:         10.0.30.10
Address:        10.0.30.10#53

Name:   srv-dns.reseau4ever.iut
Address: 10.0.30.10

debian@debian:~$
```

```
debian@debian:~$ nslookup google.com 10.0.30.10
Server:         10.0.30.10
Address:        10.0.30.10#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.142.78
Name:   google.com
Address: 2a00:1450:4006:815::200e

debian@debian:~$
```

Intégration au service DHCP

Pour que les clients reçoivent automatiquement les informations de résolution DNS, la configuration du serveur DHCP (isc-dhcp-server) est mise à jour avec deux options globales : option domain-name "reseau4ever.iut" et option domain-name-servers 10.0.30.10. Ces paramètres sont distribués à tous les clients lors de l'attribution de leur bail DHCP, garantissant que l'ensemble des machines internes résout les noms de domaine via le serveur BIND9.

```
  GNU nano 7.2                    /etc/dhcp/dhcpd.conf *
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks..
option domain-name "reseau4ever.iut";
option domain-name-servers 10.0.30.10;

default-lease-time 600;
```

5.2 Serveur Web — Apache2 / HTTPS (10.0.30.20)

Choix et adressage

Le serveur Web est hébergé sur la même machine que le service FTP (srv-web.reseau4ever.iut / srv-ftp.reseau4ever.iut, 10.0.30.20), conformément à la topologie du cahier des charges. Apache2 a été retenu comme serveur HTTP/HTTPS, solution mature et bien intégrée sous Debian.

Configuration HTTP et redirection HTTPS

Un VirtualHost HTTP (port 80) est créé pour le nom de domaine reseau4ever.iut. Ce VirtualHost ne sert aucun contenu directement : il se contente de rediriger l'intégralité du trafic HTTP vers HTTPS via une directive Redirect permanente. Cette approche garantit que tout accès non sécurisé est systématiquement redirigé vers le canal chiffré, sans possibilité de navigation en clair.

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName reseau4ever.iut.ext
        ServerAlias www.reseau4ever.iut.ext
        DocumentRoot /var/www/reseau4ever.iut.ext
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        Redirect permanent / https://www.reseau4ever.iut.ext
</VirtualHost>
```

Configuration HTTPS et durcissement TLS

Le VirtualHost HTTPS (port 443) utilise les certificats fournis par le cahier des charges (/etc/ssl/certs/cert.pem et /etc/ssl/private/key.pem). La configuration TLS retenue vise un équilibre entre compatibilité et sécurité :

- Protocoles autorisés : TLSv1, TLSv1.1, TLSv1.2 (SSLv2, SSLv3 et TLSv1.3 exclus selon la directive SSLProtocol).
- Suite de chiffrement : ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:HIGH:!MD5:!aNULL privilégie les algorithmes à forward secrecy, exclut MD5 et les suites anonymes.
- SSLHonorCipherOrder On : le serveur impose l'ordre de priorité des suites de chiffrement, évitant qu'un client impose et une suite plus faible.
- SSLCompression Off : désactivation de la compression SSL, qui expose à la vulnérabilité CRIME.

Le FQDN du serveur est déclaré dans la configuration globale Apache (ServerName srv-web.reseau4ever.iut) afin d'éviter les avertissements au démarrage et d'assurer la correspondance avec les enregistrements DNS.

```
 GNU nano 7.2  /etc/apache2/sites-available/reseau4ever.iut.ext.conf
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName reseau4ever.iut.ext
        ServerAlias www.reseau4ever.iut.ext
        DocumentRoot /var/www/reseau4ever.iut.ext
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        Redirect permanent / https://www.reseau4ever.iut.ext
</VirtualHost>

<VirtualHost *:443>
        ServerAdmin webmaster@localhost
        ServerName reseau4ever.iut.ext
        ServerAlias www.reseau4ever.iut.ext
        DocumentRoot /var/www/reseau4ever.iut.ext

        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/cert.pem
        SSLCertificateKeyFile /etc/ssl/private/key.pem
        SSLProtocol -ALL +TLSv1 +TLSv1.1 +TLSv1.2
        SSLHonorCipherOrder On
        SSLCipherSuite ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:HIGH:!MD5:!aNU
        SSLCompression off

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Voici le résultat sur une machine présente sur des VLAN différents :

Voici ce qui s'affiche lorsqu'on essaie d'accéder au site depuis http :

```
debian@debian:~$ curl http://srv-web.reseau4ever.iut
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/st
rict.dtd">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://www.reseau4ever.iut.ext">here</a>.</p
>
```

Sur une machine du VLAN ADMIN et DSI :

ADM-01 - PuTTY

```
debian@debian:~$ curl -k https://srv-web.reseau4ever.iut
<html>
 <head>
  <title>Bienvenu à reseau4ever.iut.ext !</title>
 </head>
 <body>
  <h1>Bravo, votre domaine fonctionne !</h1>
 </body>
</html>
debian@debian:~$
```

DSI-01 - PuTTY

```
debian@debian:~$ curl -k https://srv-web.reseau4ever.iut
<html>
 <head>
  <title>Bienvenu à reseau4ever.iut.ext !</title>
 </head>
 <body>
  <h1>Bravo, votre domaine fonctionne !</h1>
 </body>
</html>
```

Depuis l'hote :

5.3 Serveur FTP (10.0.30.20)

Le service FTP est colocalisé sur la machine du serveur Web (srv-ftp.reseau4ever.iut, 10.0.30.20). Le démon retenu est vsftpd, qui supporte nativement le FTPS (FTP over TLS), soit le FTP classique chiffré via AUTH TLS.

Le choix du FTPS plutôt que du SFTP s'explique par la nature des protocoles : le SFTP (SSH File Transfer Protocol) est un sous-système du protocole SSH, totalement indépendant du FTP. Déployer du SFTP aurait nécessité d'exposer SSH sur le serveur Web depuis les réseaux clients, ce qui est contraire à la politique de sécurité du projet SSH étant réservé à la DSI. Le FTPS, en revanche, s'appuie sur le protocole FTP standard avec une couche TLS, ce qui permet de sécuriser les échanges sans toucher à la configuration SSH.

L'accès anonyme est désactivé, seul l'utilisateur dédié user-web peut se connecter. Les protocoles SSL obsolètes sont désactivés, et les certificats fournis par le cahier des charges sont réutilisés. Le mode passif (ports 40000–40100) est activé pour assurer la compatibilité avec le pare-feu OpenWRT. Les utilisateurs sont confinés dans leur répertoire home via chroot, limitant l'exposition du système de fichiers.

```
# --- Mode écoute : IPv4 + IPv6 dual-stack ---
listen=NO
listen_ipv6=YES

# --- Authentification : utilisateurs locaux uniquement ---
anonymous_enable=NO
local_enable=YES
write_enable=YES

# --- Liste blanche des utilisateurs autorisés à se connecter en FTPS ---
userlist_enable=YES
userlist_deny=NO
userlist_file=/etc/vsftpd.userlist

# --- Confinement des utilisateurs dans leur home ---
chroot_local_user=YES
allow_writeable_chroot=YES

# --- Logs et affichage ---
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES

# --- Transfert de données actif depuis port 20 ---
connect_from_port_20=YES

# --- Répertoire vide pour le chroot sécurisé interne ---
secure_chroot_dir=/var/run/vsftpd/empty

# --- Authentification PAM ---
pam_service_name=vsftpd

# --- TLS / FTPS explicite ---
# Le client doit envoyer AUTH TLS avant de s'authentifier
ssl_enable=YES
force_local_logins_ssl=YES
force_local_data_ssl=YES

# Désactivation des protocoles obsolètes et vulnérables
ssl_tlsv1=NO
ssl_sslv2=NO
ssl_sslv3=NO

# Compatibilité clients FTPS modernes (pas de réutilisation de session SSL)
require_ssl_reuse=NO

# Certificat et clé privée fournis par l'entreprise
rsa_cert_file=/etc/ssl/certs/cert.pem
rsa_private_key_file=/etc/ssl/private/key.pem

# --- Mode passif (nécessaire derrière le pare-feu OpenWrt) ---
pasv_enable=YES
pasv_min_port=40000
pasv_max_port=40100
debian@debian:~$
```

Le client lftp a été utilisé pour la validation, car il supporte nativement le FTPS explicite et permet de forcer le chiffrement sur les deux canaux (contrôle et données) via set ftp:ssl-force true et set ftp:ssl-protect-data true ce qu'un client FTP basique ne ferait pas systématiquement.

Un utilisateur système dédié a été créé avec pour répertoire personnel la racine du site Web (/var/www/reseau4ever.iut.ext). La propriété du répertoire lui est attribuée via chown, et les permissions sont fixées à 755. Ce compte est le seul autorisé à se connecter en FTP : il dispose des droits nécessaires pour déposer et modifier les fichiers du site, sans pour autant avoir accès au reste du système grâce au confinement chroot activé dans vsftpd. Cette séparation garantit qu'un administrateur peut mettre à jour le site Web à distance sans disposer de privilèges système étendus.

```
debian@debian:~$ sudo usermod -d /var/www/reseau4ever.iut.ext user-web
debian@debian:~$ sudo chown -R user-web:user-web /var/www/reseau4ever.iut.ext
debian@debian:~$ sudo chmod -R 755 /var/www/reseau4ever.iut.ext
debian@debian:~$
```

commande	**rôle**
set ftp:ssl-force true	oblige FTPS
set ftp:ssl-protect-data true	chiffre les fichiers
set ssl:verify-certificate no	ignore certificat
login user-web	connexion
ls	liste fichiers

Connexion avec user-web (utilisateur autorisé)

```
lftp debian@10.0.30.20:/> login user-web
Password:
lftp user-web@10.0.30.20:~> ls
-rwxr-xr-x    1 1002     1002          145 Mar 24 16:38 index.html
lftp user-web@10.0.30.20:/> echo "test" > /tmp/test.txt
lftp user-web@10.0.30.20:/> put /tmp/test.txt
5 bytes transferred in 5 seconds (1 B/s)
lftp user-web@10.0.30.20:/> ls
-rwxr-xr-x    1 1002     1002          145 Mar 24 16:38 index.html
-rw-------    1 1002     1002            5 Mar 26 11:40 test.txt
lftp user-web@10.0.30.20:/>
```

Connexion avec debian (utilisateur bloqué)

```
debian@debian:~$ lftp 10.0.30.20
lftp 10.0.30.20:~> set ftp:ssl-force true
lftp 10.0.30.20:~> set ftp:ssl-protect-data true
lftp 10.0.30.20:~> set ssl:verify-certificate no
lftp 10.0.30.20:~> login debian
Password:
lftp debian@10.0.30.20:~> ls
ls: Login failed: 530 Permission denied.
```

5.4 Server SSH

Actuellement la connexion ssh fonctionne depuis toutes les machines il vas falloir restreindre cela depuis le pare feu du routeur pour que uniquement la **dsi-01** puisse se connecter modifier le fichier : **/etc/config/firewall**
les changements réalisés sont :

- 3 zones séparées (dsi, admin, dmz) pour isoler chaque réseau au lieu de tout mettre dans une zone lan commune
- 3 forwardings pour définir qui peut accéder à qui :
  - dsi vers admin : le réseau DSI peut joindre les machines du réseau Admin
  - dsi vers dmz : le réseau DSI peut joindre les serveurs de la DMZ
  - admin vers dmz : le réseau Admin peut joindre les serveurs de la DMZ
- Sans forwarding dans l'autre sens, le retour est bloqué, par exemple Admin ne peut pas se connecter vers DSI.
- Les règles SSH :
  - Allow-SSH-DSI : les machines du réseau DSI peuvent se connecter en SSH à n'importe quelle machine
  - Block-SSH-Admin-DMZ : les machines Admin n'ont pas le droit de se connecter en SSH aux serveurs de la DMZ
  - Block-SSH-WAN-DMZ : personne depuis Internet ne peut se connecter en SSH aux serveurs de la DMZ

```
config zone
        option name             wan
        list   network          'wan'
        list   network          'wan6'
        option input            REJECT
        option output           ACCEPT
        option forward          REJECT
        option masq             1
        option mtu_fix          1

config zone
    option name 'dsi'
    list network 'DSI'
    option input ACCEPT
    option output ACCEPT
    option forward REJECT

config zone
    option name 'admin'
    list network 'ADMIN'
    option input ACCEPT
    option output ACCEPT
    option forward REJECT

config zone
    option name 'dmz'
    list network 'DMZ'
    option input REJECT
    option output ACCEPT
    option forward REJECT

config forwarding
    option src 'dsi'
    option dest 'admin'

config forwarding
    option src 'dsi'
    option dest 'dmz'

config forwarding
    option src 'admin'
    option dest 'dmz'
```

```
#regles ssh
config rule
    option name 'Allow-SSH-DSI'
    option src 'dsi'
    option proto 'tcp'
    option dest_port '22'
    option target 'ACCEPT'

config rule
    option name 'Block-SSH-Admin-DMZ'
    option src 'admin'
    option dest 'dmz'
    option proto 'tcp'
    option dest_port '22'
    option target 'DROP'

config rule
    option name 'Block-SSH-WAN-DMZ'
    option src 'wan'
    option dest 'dmz'
    option proto 'tcp'
    option dest_port '22'
    option target 'DROP'
```

Depuis la DSI vers un pc admin cela fonctionne :

```
debian@debian:~$ ssh debian@10.0.20.210
The authenticity of host '10.0.20.210 (10.0.20.210)' can't be established.
ED25519 key fingerprint is SHA256:Gojcktfl97qAXF9KaBhKTK93gIYeN/574t8Ph9Z12D0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.20.210' (ED25519) to the list of known hosts.
debian@10.0.20.210's password:
Linux debian 6.1.0-22-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.94-1 (2024-06-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar 25 17:02:13 2026 from 10.0.10.195
debian@debian:~$
```

Depuis le pc admin vers la DSI cela ne fonctionne plus la connexion est bien refuser :

```
debian@debian:~$ ssh debian@10.0.10.195
ssh: connect to host 10.0.10.195 port 22: Connection refused
debian@debian:~$
```

Partie 2 – Sécurité

1. Pare-feu

1.1 Modèle de zones OpenWRT

Le pare-feu OpenWRT contient quatre zones :

Zone	Interfaces	Politique input	Politique output	Politique forward	Description
DSI	eth2.10	DROP	ACCEPT	ACCEPT	Réseau technique
ADMIN	eth2.20	DROP	ACCEPT	DROP	Réseau administratif
DMZ	eth1	DROP	ACCEPT	DROP	Services exposés
wan	eth0	DROP	ACCEPT	DROP	Internet

Politique input DROP sur toutes les zones : Par défaut, aucun paquet entrant n'est accepté sur le routeur lui-même, quelle que soit la zone d'origine.Cela protège le routeur contre tout accès non sollicité, y compris depuis les réseaux internes.
Politique output ACCEPT sur toutes les zones : Le routeur peut émettre du trafic vers toutes les zones. Cela est nécessaire pour qu'il puisse répondre aux requêtes comme DHCP, DNS, etc, et relayer les paquets vers leur destination.
Politique forward ACCEPT sur la DSI uniquement : Le trafic en provenance de la DSI peut être forwardé vers n'importe quelle destination par défaut. Cette politique est la car le statut de la DSI est privilégié, il doit pouvoir administrer l'ensemble des réseaux.
Politique forward DROP sur Admin, DMZ et WAN : Tout forwarding est bloqué par défaut depuis ces zones. Seules les communications explicitement autorisées sont permises. Ce principe garantit qu'aucune communication non prévue ne peut traverser le routeur.

1.2 Hiérarchie des accès inter-zones

Une hiérarchie de confiance est appliquée entre les zones internes. Cette hiérarchie se traduit par les règles de filtrage suivantes :

- La DSI a un accès complet à Admin et DMZ : La DSI doit pouvoir administrer l'ensemble du réseau.
- Le réseau Admin a un accès autorisé à la DMZ. Tout accès vers le réseau DSI est bloqué
- La DMZ n'initie aucune connexion vers les réseaux internes : toute connexion émise depuis la DMZ pour les réseaux DSI ou Admin est bloquée.

1.3 Règles spécifiques à la DMZ

Des restrictions supplémentaires s'appliquent au trafic à destination de la DMZ, selon son origine :

Origine	SSH (port 22)	ICMP Echo (ping)	HTTP/HTTPS	DNS (53)	FTP
Internet	BLOQUÉ	BLOQUÉ	BLOQUÉ / AUTORISÉ	BLOQUÉ	BLOQUÉ
Réseau Admin	BLOQUÉ	AUTORISÉ	AUTORISÉ	AUTORISÉ	BLOQUÉ
Réseau DSI	AUTORISÉ	AUTORISÉ	AUTORISÉ	AUTORISÉ	AUTORISÉ

Voici pourquoi nous avons fait c'est choix :

SSH est bloqué depuis Internet : Un attaquant ne peut pas tenter de se connecter directement aux serveurs de la DMZ. Ça limite les surfaces d'attaques depuis l'extérieur.

ICMP est bloqué depuis Internet: Cela empêche un attaquant de scanner quels serveurs sont en ligne dans la DMZ via des pings.

ICMP est autorisé depuis Admin : Les admin ont besoin de pouvoir diagnostiquer la connectivité réseau, sans pour autant avoir accès à l'administration SSH des serveurs.

DNS est bloqué depuis Internet : Le domaine reseau4ever.iut est interne, personne depuis Internet peux y accéder.

FTP uniquement depuis DSI : Le FTP sert uniquement à administrer le serveur web. Seule la DSI a ce rôle, donc ni Internet ni Admin n'ont besoin d'y accéder.

Le réseau DSI dispose d'un accès complet à tous les ports et services de la DMZ, cela est dû à son rôle d'administration et d'audit de l'ensemble de l'infrastructure.

2. Détection et prévention d'intrusions (IDS/IPS)

2.1 Positionnement et outil retenu

Le système IDS/IPS est déployé sur la machine Debian positionnée en amont du routeur. Cette position permet que tout le trafic entrant et sortant passe obligatoirement par cette machine avant d'atteindre le routeur. L'IDS/IPS peut donc inspecter la totalité du trafic entrant.
Le logiciel utilisé est Suricata, un moteur IDS/IPS open source est capable d'analyser le trafic en temps réel, de détecter les comportements suspects sur la base de règles (signatures) et de bloquer automatiquement certaines attaques en mode IPS.
Suricata a été préféré à Snort pour sa capacité de traitement (meilleures performances sur des flux importants). Il s'intègre avec les règles iptables mises en place sur la machine.

```
debian@debian:~$ sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 DROP       6    --  enp2s0 *       0.0.0.0/0            0.0.0.0/0
          tcp dpt:22
    0     0 DROP       1    --  enp2s0 *       0.0.0.0/0            0.0.0.0/0
          icmptype 8
    4  1312 NFQUEUE    0    --  *      *       0.0.0.0/0            0.0.0.0/0
          NFQUEUE num 0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

   55  4215 NFQUEUE    0    --  *      *       0.0.0.0/0            0.0.0.0/0
          NFQUEUE num 0
    0     0 NFQUEUE    0    --  *      *       0.0.0.0/0            0.0.0.0/0
          NFQUEUE num 0 bypass
    0     0 NFQUEUE    0    --  *      *       0.0.0.0/0            0.0.0.0/0
          NFQUEUE num 0 bypass

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    1   328 NFQUEUE    0    --  *      *       0.0.0.0/0            0.0.0.0/0
          NFQUEUE num 0
debian@debian:~$
```

```
debian@debian:~$ sudo iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 22 packets, 4106 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DNAT       6    --  enp2s0 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443 to:10.0.30.20:443
    0     0 DNAT       6    --  enp2s0 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80 to:10.0.30.20:80
    0     0 DNAT       6    --  enp2s0 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:21 to:10.0.30.20:21
    0     0 DNAT       6    --  enp2s0 *       0.0.0.0/0            0.0.0.0/0           tcp dpts:40000:50000 to:10.0.30.20:40000-50000
    0     0 DNAT       17   --  enp2s0 *       0.0.0.0/0            0.0.0.0/0           udp dpt:53 to:10.0.30.10:53
    0     0 DNAT       6    --  enp2s0 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:53 to:10.0.30.10:53

Chain INPUT (policy ACCEPT 1 packets, 328 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 1 packets, 328 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   19  1720 MASQUERADE  0    --  *      enp2s0  0.0.0.0/0            0.0.0.0/0
debian@debian:~$
```

2.2 Détection et blocage des scans réseau

L'IDS/IPS doit être capable de détecter et bloquer automatiquement les scans réseau. Ces scans constituent une phase de reconnaissance préalable à toute intrusion, permettant à un attaquant d'identifier les services exposés avant de tenter d'exploiter des vulnérabilités.

La détection des scans repose principalement sur une approche signature-based, les scans de ports en séquence rapide et les tentatives de fingerprinting OS. Cette méthode à quand même une limite : elle est inefficace contre des techniques de scan inconnues ou zero-day.

```
debian@debian:~$ sudo suricata -T -c /etc/suricata/suricata.yaml
26/3/2026 -- 14:59:50 - <Info> - Running suricata under test mode
26/3/2026 -- 14:59:50 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
26/3/2026 -- 15:00:51 - <Notice> - Configuration provided was successfully loaded. Exiting.
debian@debian:~$ cat /var/log/suricata/fast.log
03/25/2026-22:02:32.348718  [**] [1:1000001:1] TEST ICMP DETECTED [**] [Classification: (null)] [Priority: 3] (ICMP) 10.10.10.2:8 -> 10.10.10.1:0
03/25/2026-22:02:32.349411  [**] [1:1000001:1] TEST ICMP DETECTED [**] [Classification: (null)] [Priority: 3] (ICMP) 10.10.10.1:0 -> 10.10.10.2:0
03/25/2026-22:02:49.426396  [**] [1:1000001:1] TEST ICMP DETECTED [**] [Classification: (null)] [Priority: 3] (IPv6-ICMP) fe80:0000:0000:0000:0ed9:e7ff:fea2:0000:135 -> fc01:0000:0000:0100:0000:0000:0000:0001:0
03/25/2026-22:02:49.426668  [**] [1:1000001:1] TEST ICMP DETECTED [**] [Classification: (null)] [Priority: 3] (IPv6-ICMP) fc01:0000:0000:0100:0000:0000:0000:0001:136 -> fe80:0000:0000:0000:0ed9:e7ff:fea2:0000:0
03/25/2026-22:02:58.726802  [**] [1:1000001:1] TEST ICMP DETECTED [**] [Classification: (null)] [Priority: 3] (IPv6-ICMP) fe80:0000:0000:0000:0e4c:adff:fe58:0001:135 -> fe80:0000:0000:0000:0ed9:e7ff:fea2:0000:0
03/25/2026-22:02:58.731632  [**] [1:1000001:1] TEST ICMP DETECTED [**] [Classification: (null)] [Priority: 3] (IPv6-ICMP) fe80:0000:0000:0000:0ed9:e7ff:fea2:0000:136 -> fe80:0000:0000:0000:0e4c:adff:fe58:0001:0
03/26/2026-07:43:47.685245  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:55184
03/26/2026-07:43:47.847350  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:55184
03/26/2026-07:43:48.853423  [**] [1:1000001:1] TEST ICMP DETECTED [**] [Classification: (null)] [Priority: 3] (ICMP) 192.168.1.168:8 -> 192.168.1.113:0
03/26/2026-07:43:48.853521  [**] [1:1000001:1] TEST ICMP DETECTED [**] [Classification: (null)] [Priority: 3] (ICMP) 192.168.1.113:0 -> 192.168.1.168:0
03/26/2026-07:43:48.986566  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:54885
03/26/2026-07:43:48.986836  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:54885
03/26/2026-07:43:49.049860  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:54885
03/26/2026-07:43:51.017213  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:55184
03/26/2026-07:43:51.020841  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:55184
03/26/2026-07:43:51.911646  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:54885
03/26/2026-07:43:51.916507  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:54885
03/26/2026-07:43:52.687746  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:55184
03/26/2026-07:43:52.788163  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:55184
03/26/2026-07:43:53.076720  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:5043 -> 192.168.1.168:56018
03/26/2026-07:43:53.988135  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:54885
03/26/2026-07:43:53.988368  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:54885
03/26/2026-07:43:54.053182  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:54885
03/26/2026-07:43:57.689384  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:55184
03/26/2026-07:43:57.781641  [**] [1:2200074:2] SURICATA TCPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (TCP) 192.168.1.30:3080 -> 192.168.1.168:55184
```

2.3 Configuration de Suricata

Nous avons configuré des règles personnalisées (*Custom Rules*) afin de détecter les comportements suspects et de bloquer activement les tentatives d'intrusion sur les segments critiques.

Détection des activités de reconnaissance (Scans)

Les premières règles visent à identifier les phases de découverte d'un attaquant :

- **Test ICMP (sid: 1000001) :** Une règle d'alerte simple permettant de vérifier le bon fonctionnement de la sonde lors d'un ping.
- **Détection de Scan SYN (sid: 1000002) :** Utilise un seuil de déclenchement (*threshold*). Si plus de 20 paquets SYN sont reçus en 3 secondes depuis une même source vers notre réseau interne (10.0.0.0/16), une alerte de type "attempted-recon" est générée.
- **Blocage de scans furtifs (sid: 1000003 à 1000005) :** Ces règles utilisent l'action drop pour rejeter les paquets malformés utilisés par Nmap (scans FIN, NULL et Xmas). Ces paquets n'ont aucune utilité légitime et servent uniquement à contourner des pare-feux basiques.

Sécurisation des accès d'administration (SSH)

Le protocole SSH est une cible prioritaire. Nous avons mis en place une politique de filtrage granulaire :

- **Protection périmétrique (sid: 1000006) :** Rejet de tout flux SSH provenant de l'extérieur (réseaux autres que 10.0.0.0/16) vers la DMZ.
- **Traçabilité DSI (sid: 1000009 & 1000010) :** Contrairement aux autres zones, les flux SSH provenant de la zone de confiance (DSI) sont autorisés mais font l'objet d'une alerte de journalisation (alert) pour garantir un historique des accès administrateur.

Protection des services de la DMZ

La DMZ héberge des services exposés qui nécessitent une surveillance accrue :

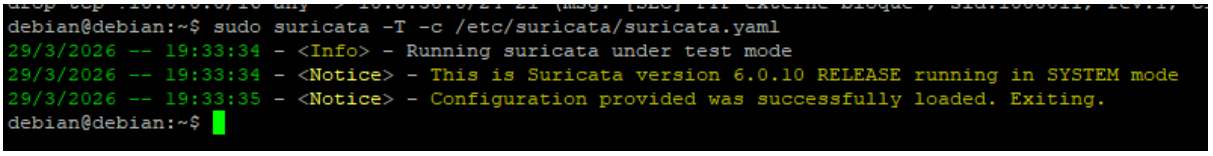
- **Filtrage ICMP Externe (sid: 1000007) :** Blocage des requêtes Echo Request venant d'Internet. Cela complète le filtrage du routeur pour rendre la DMZ invisible aux scans de découverte.
- **Surveillance HTTP (sid: 1000008) :** Alerte lorsqu'un flux circule en clair sur le port 80. Cela permet d'identifier des défauts de configuration où le trafic ne serait pas correctement redirigé vers le HTTPS (443).
- **Blocage FTP Externe (sid: 1000011) :** Le service FTP est jugé trop vulnérable pour être exposé sur Internet. Toute tentative de connexion depuis l'extérieur est immédiatement abandonnée par la sonde.

```
# 1 - Test ICMP
alert icmp any any -> any any (msg:"[TEST] Ping detecte"; sid:1000001; rev:1; classtype:icmp-event;)

# 2 - Detection scan SYN
alert tcp any any -> 10.0.0.0/16 any (msg:"[SEC] Scan SYN detecte"; flags:S; threshold:type threshold,track by_src,count 20,seconds 3; sid:1000002; rev:1; classtype:attempted-recon;)

# 3 - Scans FIN NULL Xmas bloquants
drop tcp any any -> 10.0.0.0/16 any (flags:F; msg:"[SEC] Scan FIN bloque"; sid:1000003; rev:1; classtype:attempted-recon;)
drop tcp any any -> 10.0.0.0/16 any (flags:0; msg:"[SEC] Scan NULL bloque"; sid:1000004; rev:1; classtype:attempted-recon;)
drop tcp any any -> 10.0.0.0/16 any (flags:FPU; msg:"[SEC] Scan Xmas bloque"; sid:1000005; rev:1; classtype:attempted-recon;)

# 4 - SSH Extérieur -> DMZ bloqué
```


debian@debian:~$ sudo suricata -T -c /etc/suricata/suricata.yaml
29/3/2026 -- 19:33:34 - <Info> - Running suricata under test mode
29/3/2026 -- 19:33:34 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
29/3/2026 -- 19:33:35 - <Notice> - Configuration provided was successfully loaded. Exiting.
debian@debian:~$

Partie 3 — Audit réseau

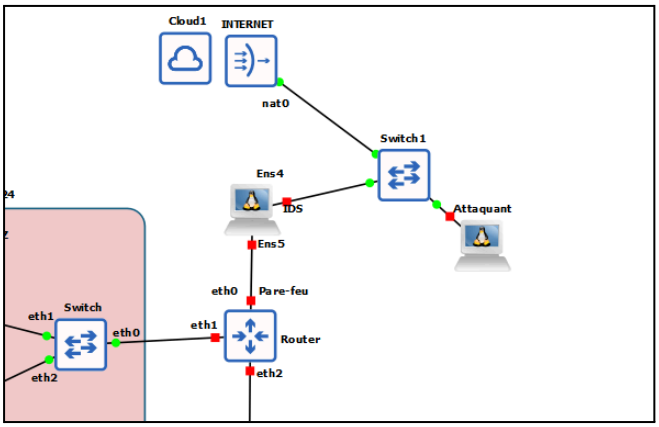
1 Contexte et démarche

L'audit a été conduit depuis une machine extérieure positionnée sur le même segment réseau que l'interface WAN de l'IDS/IPS, simulant un attaquant Internet. Tout le trafic de la machine attaquante transite par l'IDS Suricata avant d'atteindre le routeur OpenWrt et les réseaux internes.

Les tests ont été conduits en quatre phases progressives :

- Découverte réseau et cartographie des hôtes
- Scan des ports ouverts
- Identification des versions de services
- Recherche de vulnérabilités et test de l'IDS/IPS

Voici les modifications apportées à l'architecture du réseau, avec l'ajout d'un switch et de la machine attaquant :



2. Découverte réseau

Commande :

Découverte d'hôtes sur le segment DMZ : nmap -sn 10.0.30.0/24

```
debian@debian:~$ nmap -sn 10.0.30.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-29 16:00 UTC
Nmap done: 256 IP addresses (0 hosts up) scanned in 103.55 seconds
```

Résultat : 0 hôtes détectés sur les 256 adresses scannées.

Analyse : Le pare-feu OpenWrt bloque correctement les requêtes ICMP depuis l'extérieur vers la DMZ. Aucune machine n'est visible par une simple découverte réseau — un attaquant ne peut pas cartographier le réseau par ping seul.

3. Scan des ports

Commande :

Scan SYN - Discret et rapide : sudo nmap -sS -Pn 10.0.30.20

```
debian@debian:~$ sudo nmap -sS -Pn 10.0.30.20
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-29 18:12 UTC
Nmap scan report for 10.0.30.20
Host is up (0.060s latency).
Not shown: 995 closed tcp ports (reset)
PORT    STATE    SERVICE
21/tcp  filtered ftp
22/tcp  filtered ssh
53/tcp  open     domain
80/tcp  open     http
443/tcp open     https

Nmap done: 1 IP address (1 host up) scanned in 2.77 seconds
debian@debian:~$
```

Logs de Suricata :

```
03/29/2026-18:12:02.328906  [**] [1:1000002:1] [SEC] Scan SYN detecte [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:50004 -> 10.0.30.20:2160
03/29/2026-18:12:02.334598  [**] [1:1000002:1] [SEC] Scan SYN detecte [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:50004 -> 10.0.30.20:27000
03/29/2026-18:12:02.342273  [**] [1:1000002:1] [SEC] Scan SYN detecte [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:50004 -> 10.0.30.20:4004
03/29/2026-18:12:02.367610  [**] [1:1000002:1] [SEC] Scan SYN detecte [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:50004 -> 10.0.30.20:5050
03/29/2026-18:12:02.414018  [**] [1:1000002:1] [SEC] Scan SYN detecte [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:50004 -> 10.0.30.20:10617
```

Résultats :

Port	État	Service
21/tcp	filtered	ftp
22/tcp	filtered	ssh
53/tcp	open	domain
80/tcp	open	http
443/tcp	open	https

Analyse : FTP et SSH sont correctement filtrés par Suricata. DNS, HTTP et HTTPS sont accessibles depuis l'extérieur, conformément aux services publics attendus.

Un test de scan FIN a également été effectué :

Scan FIN (Furtif) : sudo nmap -sF -Pn 10.0.30.20

```
debian@debian:~$ sudo nmap -sF -Pn 10.0.30.20
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-29 18:13 UTC
Nmap scan report for 10.0.30.20
Host is up.
All 1000 scanned ports on 10.0.30.20 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 202.10 seconds
debian@debian:~$
```

Logs de Suricata :

```
03/29/2026-18:16:25.392612  [Drop] [**] [1:1000003:1] [SEC] Scan FIN bloque [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:49726 -> 10.0.30.20:50006
03/29/2026-18:16:25.392627  [Drop] [**] [1:1000003:1] [SEC] Scan FIN bloque [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:49726 -> 10.0.30.20:1247
03/29/2026-18:16:25.403559  [Drop] [**] [1:1000003:1] [SEC] Scan FIN bloque [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:49726 -> 10.0.30.20:84
03/29/2026-18:16:26.374589  [Drop] [**] [1:1000003:1] [SEC] Scan FIN bloque [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.122.61:49728 -> 10.0.30.20:2038
```

Aucun résultat retourné — Suricata a bloqué l'intégralité des paquets FIN, confirmant l'efficacité des règles IPS contre les scans de reconnaissance furtifs.

4. Identification des services et versions

Commandes :

Détection de version des services : sudo nmap -sV -Pn 10.0.30.20

```
debian@debian:~$ sudo nmap -sV -Pn 10.0.30.20
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-29 17:54 UTC
Nmap scan report for 10.0.30.20
Host is up (0.035s latency).
Not shown: 995 closed tcp ports (reset)
PORT    STATE    SERVICE  VERSION
21/tcp  filtered ftp
22/tcp  filtered ssh
53/tcp  open     domain   ISC BIND 9.18.44-1~deb12u1 (Debian Linux)
80/tcp  open     http     Apache httpd 2.4.66
443/tcp open     ssl/http Apache httpd 2.4.66 ((Debian))
Service Info: Host: reseau4ever.iut.ext; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.28 seconds
debian@debian:~$
```

Scan Agressif (OS Fingerprinting + Scripts par défaut) : sudo nmap -A -Pn 10.0.30.20

```
debian@debian:~$ sudo nmap -A -Pn 10.0.30.20
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-29 17:55 UTC
Nmap scan report for 10.0.30.20
Host is up (0.012s latency).
Not shown: 995 closed tcp ports (reset)
PORT    STATE    SERVICE  VERSION
21/tcp  filtered ftp
22/tcp  filtered ssh
53/tcp  open     domain   ISC BIND 9.18.44-1~deb12u1 (Debian Linux)
| dns-nsid:
|_  bind.version: 9.18.44-1~deb12u1-Debian
80/tcp  open     http     Apache httpd 2.4.66
|_http-title: Did not follow redirect to https://www.reseau4ever.iut.ext
|_http-server-header: Apache/2.4.66 (Debian)
443/tcp open     ssl/http Apache httpd 2.4.66 ((Debian))
|_http-server-header: Apache/2.4.66 (Debian)
|_http-title: Bienvenu \xC3\xA0 reseau4ever.iut.ext !
| ssl-cert: Subject: commonName=www.reseau4ever.iut/organizationName=Universite Bretagne Sud/stateOrProvinceName=Morbihan/countryName=FR
| Not valid before: 2026-03-17T22:00:40
|_Not valid after:  2026-03-24T22:00:40
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=3/29%OT=53%CT=1%CU=33382%PV=Y%DS=3%DC=T%G=Y%TM=69C967B
OS:C%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=10E%TI=Z%TS=A)OPS(O1=M4C4ST
OS:11NW6%O2=M4C4ST11NW6%O3=M4C4NNT11NW6%O4=M4C4ST11NW6%O5=M4C4ST11NW6%O6=M4
OS:C4ST11)WIN(W1=FED0%W2=FED0%W3=FED0%W4=FED0%W5=FED0%W6=FED0)ECN(R=Y%DF=Y%
OS:T=40%W=FC94%O=M4C4NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)
OS:T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=
OS:N)T7(R=N)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G
OS:)IE(R=N)

Network Distance: 3 hops
Service Info: Host: reseau4ever.iut.ext; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 3389/tcp)
HOP RTT      ADDRESS
1   ...
2   32.34 ms 10.10.10.2
3   35.34 ms 10.0.30.20

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.17 seconds
debian@debian:~$
```

Résultats :

Port	Service	Version identifiée
53/tcp	DNS	ISC BIND 9.18.44-1~deb12u1 (Debian Linux)
80/tcp	HTTP	Apache httpd 2.4.66 (Debian)
443/tcp	HTTPS	Apache httpd 2.4.66 (Debian)

Informations complémentaires obtenues :

- Système d'exploitation : Linux (Debian)
- Nom d'hôte : reseau4ever.iut.ext
- HTTP redirige automatiquement vers HTTPS
- Traceroute : 3 sauts — machine attaquante → IDS (10.10.10.1) → routeur OpenWrt (10.10.10.2) → serveur DMZ (10.0.30.20)

Analyse : Les versions d'Apache 2.4.66 et BIND 9.18.44 sont récentes et ne présentent pas de CVE critiques connues à ce jour. Cependant, le scan révèle une information sensible : la version exacte d'Apache est exposée dans les en-têtes HTTP (Server: Apache/2.4.66), ce qui facilite le ciblage par un attaquant.

5. Vulnérabilité identifiée : certificat SSL expiré

Le scan intensif a mis en évidence une vulnérabilité significative au niveau du certificat SSL:

ssl-cert: Subject: commonName=www.reseau4ever.iut
organizationName=Universite Bretagne Sud
stateOrProvinceName=Morbihan / countryName=FR
Not valid before: 2026-03-17T22:00:40
Not valid after: 2026-03-24T22:00:40 ← EXPIRÉ DEPUIS 5 JOURS

Nature de la vulnérabilité : Le certificat SSL/TLS du serveur web est expiré depuis le 24 mars 2026, soit 5 jours avant la date de l'audit. Cette situation présente plusieurs risques :

- Les navigateurs et clients modernes affichent une alerte de sécurité ou refusent la connexion
- Un attaquant peut exploiter ce contexte pour réaliser une attaque man-in-the-middle, les utilisateurs étant habitués à ignorer les avertissements de certificat
- La confiance dans le service HTTPS est compromise, ce qui va à l'encontre de l'objectif de sécurisation des communications

6. Recherche de vulnérabilités applicatives

Commandes :

Scan de vulnérabilités automatiques : sudo nmap --script vuln -Pn 10.0.30.20

```
debian@debian:~$ sudo nmap --script vuln -Pn 10.0.30.20
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-29 17:58 UTC
Nmap scan report for 10.0.30.20
Host is up (0.043s latency).
Not shown: 995 closed tcp ports (reset)
PORT    STATE    SERVICE
21/tcp  filtered ftp
22/tcp  filtered ssh
53/tcp  open     domain
80/tcp  open     http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-passwd: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-vuln-cve2013-7091: ERROR: Script execution failed (use -d to debug)
443/tcp open     https
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.

Nmap done: 1 IP address (1 host up) scanned in 51.50 seconds
debian@debian:~$
```

Test spécifique de la faille Shellshock : sudo nmap --script http-shellshock -Pn -p 80,443 10.0.30.20

```
debian@debian:~$ sudo nmap --script http-shellshock -Pn -p 80,443 10.0.30.20
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-29 18:20 UTC
Nmap scan report for 10.0.30.20
Host is up (0.015s latency).

PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.04 seconds
debian@debian:~$
```

Tentative de transfert de zone DNS, teste si le serveur DNS est mal configuré et autorise la copie de toute sa base. : sudo nmap --script dns-zone-transfer --script-args dns-zone-transfer.domain=reseau4ever.iut -Pn -p 53 10.0.30.20

```
debian@debian:~$ sudo nmap --script dns-zone-transfer --script-args dns-zone-transfer.domain=reseau4ever.iut -Pn -p 53 10.0.30.20
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-29 18:21 UTC
Nmap scan report for 10.0.30.20
Host is up (0.014s latency).

PORT   STATE SERVICE
53/tcp open  domain

Nmap done: 1 IP address (1 host up) scanned in 7.54 seconds
debian@debian:~$
```

Résultats :

- XSS stocké : aucune vulnérabilité trouvée
- XSS DOM : aucune vulnérabilité trouvée
- CSRF : aucune vulnérabilité trouvée
- Shellshock : non vulnérable
- Transfert de zone DNS (AXFR) : refusé — le serveur DNS ne permet pas le transfert de zone depuis l'extérieur

Analyse : Aucune vulnérabilité applicative critique n'a été détectée sur les services web et DNS. La configuration Apache et BIND est correcte de ce point de vue.

7. Test de l'IDS/IPS Suricata

Les scans de reconnaissance ont été effectués depuis la machine attaquante pendant que les logs Suricata étaient surveillés en temps réel.

Type de scan	Commande	Résultat nmap	Action Suricata
SYN	nmap -sS	Ports open/filtered visibles	Alerte [SEC] Scan SYN detecte
FIN	nmap -sF	Aucun résultat	Drop [SEC] Scan FIN bloque
NULL	nmap -sN	Aucun résultat	Drop [SEC] Scan NULL bloque
Xmas	nmap -sX	Aucun résultat	Drop [SEC] Scan Xmas bloque
Ping	ping	100% perte	Drop [SEC] ICMP externe->DMZ bloque
FTP	port 21	filtered	Drop [SEC] FTP externe bloque

Suricata a correctement détecté et bloqué l'ensemble des scans furtifs (FIN, NULL, Xmas). Le scan SYN est détecté et logué mais les ports légitimement ouverts (53, 80, 443) restent accessibles, ce qui est le comportement attendu.

8. Synthèse des résultats

Élément testé	Résultat	Conformité
Ping depuis WAN vers DMZ	Bloqué	Conforme
SSH depuis WAN vers DMZ	Filtered	Conforme
FTP depuis WAN vers DMZ	Filtered	Conforme
HTTP accessible	Open + redirection HTTPS	Conforme
HTTPS accessible	Open	Conforme
DNS accessible	Open	Conforme
Transfert de zone DNS	Refusé	Conforme
Scan FIN/NULL/Xmas	Bloqué par Suricata	Conforme
Certificat SSL	Expiré depuis le 24/03/2026	⚠️ Non conforme
Version Apache exposée	Visible dans les en-têtes	⚠️ Amélioration possible

9. Recommandations

Critique — à corriger immédiatement :

- Renouveler le certificat SSL/TLS du serveur web. En production, mettre en place un renouvellement automatique (Let's Encrypt + certbot) pour éviter toute expiration future.

Amélioration recommandée :

Masquer la version d'Apache dans les en-têtes HTTP en ajoutant dans la configuration Apache :
ServerTokens ProdServerSignature Off

- Cela empêche un attaquant de cibler précisément la version du serveur.

Maintenance :

- Maintenir les signatures Suricata à jour pour faire face aux menaces émergentes
- Effectuer des audits réguliers depuis le réseau DSI pour valider la cohérence des règles internes
- Vérifier périodiquement la validité des certificats SSL sur l'ensemble des services exposés

10. Conclusion

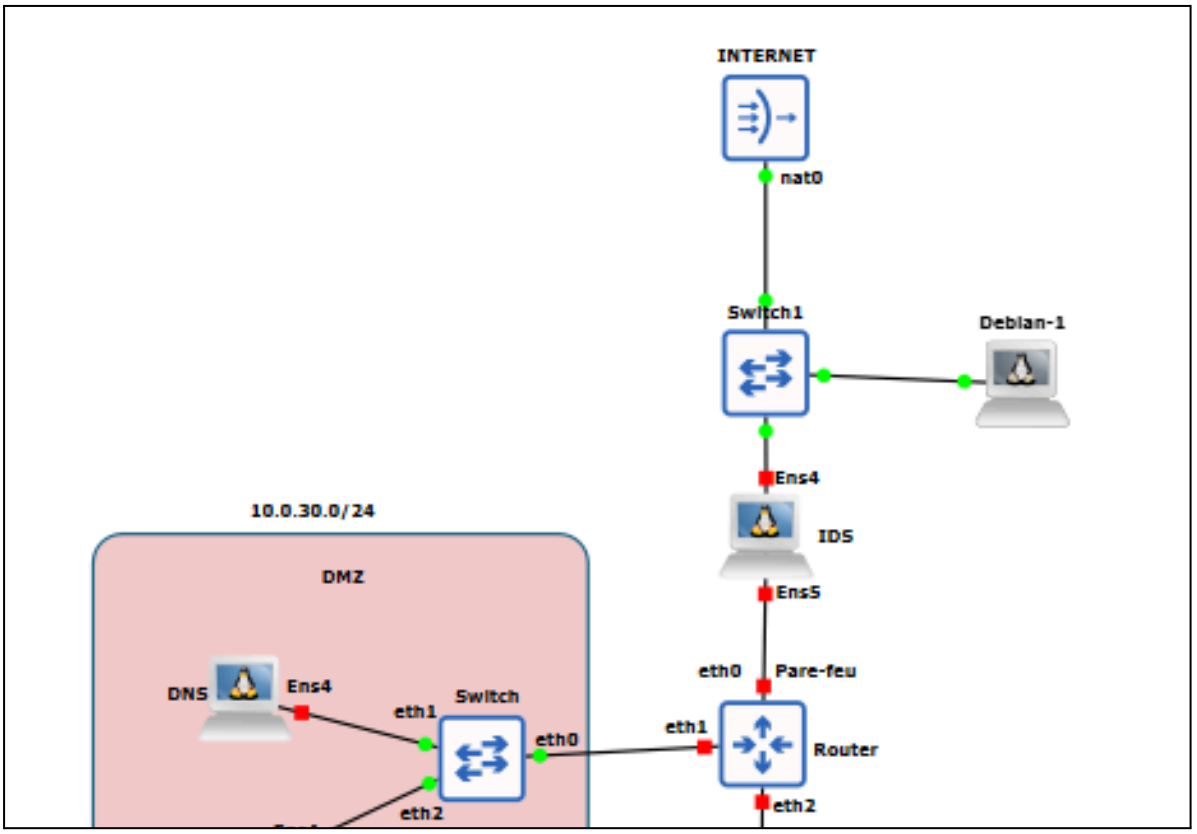
L'audit a démontré que le périmètre réseau est globalement bien sécurisé. Les règles de filtrage du pare-feu OpenWrt et de l'IDS/IPS Suricata sont efficaces : les scans de reconnaissance furtifs sont bloqués, SSH et FTP sont inaccessibles depuis l'extérieur, et les tentatives de ping sont rejetées silencieusement.

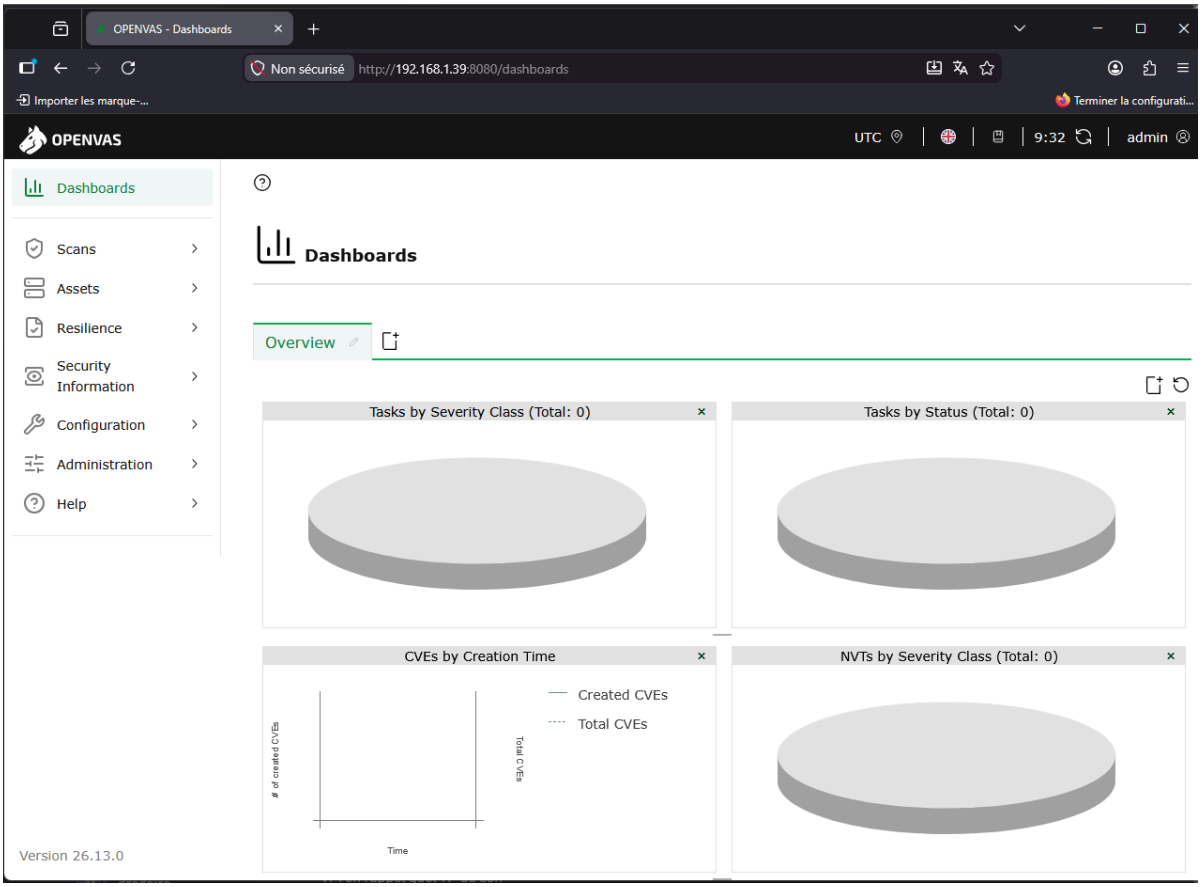
La seule vulnérabilité identifiée concerne le certificat SSL expiré, qui compromet la confiance dans le service HTTPS et pourrait être exploitée dans le cadre d'une attaque man-in-the-middle. Cette faille, bien que non critique sur le plan technique immédiat, représente un risque opérationnel réel et devrait être corrigée en priorité.

Les versions de services déployées (Apache 2.4.66, BIND 9.18.44) sont récentes et ne présentent pas de CVE exploitables connues à la date de l'audit, ce qui témoigne d'une bonne politique de mise à jour logicielle.

11. Tentative d’audit Audit avec Openvas

L’installation repose sur un switch qui relie le NAT, une machine “attaquante” et l’entrée du réseau à auditer. La machine attaquante est une Debian sur laquelle est déployé OpenVAS via Docker.
L’interface web d’OpenVAS n’est pas directement accessible depuis le LAN. Elle est exposée en passant par le serveur GNS3, qui joue le rôle d’intermédiaire grâce à des règles de port forwarding configurées avec iptables. L’attaquant dispose d’une adresse IP dans le réseau NAT, et l’accès externe se fait via l’IP du serveur GNS3.
La configuration réseau a nécessité des ajustements manuels, notamment l’ajout de serveurs DNS pour permettre la résolution de noms et l’accès aux dépôts. Plusieurs problèmes techniques ont été rencontrés : absence de DNS, espace disque insuffisant (empêchant le bon fonctionnement des services comme PostgreSQL), ainsi qu’un manque de mémoire RAM impactant les performances du conteneur. La machine attaquante finale nécessite 8 CPU, 14 Go de RAM et 80 Go de stockage. Cette configuration n’est pas idéale car elle est très coûteuse en ressources et toujours très lente au vu de l’empilement des virtualisations.
Initialement, l’idée était d’accéder à l’interface via une machine Firefox intégré dans GNS3, puis via une machine Webterm. Ces approches n’ont pas fonctionné correctement, notamment à cause de limitations réseau et d’affichage (page blanche).
Enfin, un incident est survenu lors de la suppression de la machine Firefox plus utilisée pour la version finale. Elle a entraîné de manière inattendue la suppression de la machine Debian attaquante. Cela a nécessité une réinstallation complète. Il n’y avait aucune trace du disque virtuel suite à ce bug.
Architecture finale a l’entrée du réseaux :





Nous allons maintenant faire passer plusieurs scans de vulnérabilités sur le réseau. depuis différent point d'entrée dans l'objectif de trouver des CVE exploitable via metasploit ensuite. Pour cela on connecte l'attaquant aux différents switch du réseau pour évaluer la sécurité y compris en interne.

Pour une raison inconnue, la lenteur du réseau a empêché le pull de l'image docker de openvas pour la réinstaller. Suite au bug nous n'avons pas pu retourner sur le dashboard pour faire les scans de vulnérabilités.

Bien que l'audit via les scripts NSE de Nmap ait permis de valider la robustesse des services exposés (Apache, BIND), une analyse complémentaire via un scanner de vulnérabilités dédié tel qu'OpenVAS aurait permis d'approfondir l'évaluation de la surface d'attaque. Contrairement à Nmap qui se base principalement sur les numéros de version (analyse passive), OpenVAS réalise des tests actifs et exhaustifs qui auraient pu révéler :

- **Failles de configuration applicative :** Détection de suites de chiffrement TLS faibles ou obsolètes (ex: support de TLS 1.0/1.1), ou l'absence de flags de sécurité sur les cookies HTTP (*HttpOnly*, *Secure*).
- **Vulnérabilités de la chaîne logistique (Supply Chain) :** Identification de bibliothèques JavaScript obsolètes utilisées par le serveur Web (ex: une version vulnérable de jQuery) qui n'apparaissent pas dans les en-têtes du serveur Apache.
- **Audit des services DNS/FTP :** Analyse plus fine des options de configuration, comme la détection de comptes FTP anonymes avec droits d'écriture ou des récurrences DNS permettant des attaques par amplification.

- **Corrélation avec les bases CVE :** Une mise en relation automatisée et en temps réel de chaque service détecté avec les bases de données de vulnérabilités les plus récentes, au-delà des scripts standards de Nmap.

L'absence de ce scan ne remet pas en cause les conclusions de cet audit, mais son intégration dans un cycle de maintenance continue permettrait une gestion des risques, notamment pour anticiper les failles de type "Zero-Day" sur les services critiques de la DMZ.

12. Exploitation de vulnérabilité avec metasploit

L'installation de Metasploit n'a finalement pas pu être menée à terme en raison des difficultés rencontrées avec Docker dans notre environnement GNS3 qui ont rendu les machines instables. Le téléchargement étaient impossible (timeouts TLS, interruptions réseau) et provoquaient régulièrement des blocages de la machine virtuelle, qui a cessé de répondre à plusieurs reprises. Ces instabilités nous ont contraints à réinitialiser une partie de l'environnement à plusieurs reprises. Compte tenu du temps restant avant le rendu, il n'était plus réaliste de relancer une installation complète et fiable de Metasploit.

Pour rappel, Metasploit est un framework d'exploitation largement utilisé en pentest, permettant d'identifier, tester et exploiter des vulnérabilités sur des systèmes cibles. Dans le cadre de cet audit, il aurait pu être utilisé en complément d'OpenVAS afin de valider concrètement certaines vulnérabilités détectées, par exemple via des modules d'exploitation ou des payloads spécifiques. Il aurait également permis de simuler des scénarios d'attaque plus avancés, comme l'obtention d'un accès distant ou l'élévation de privilèges, afin d'évaluer plus précisément le niveau de risque du système audité. Cependant, en raison des contraintes techniques et temporelles, cette étape n'a pas pu être intégrée à l'environnement final.

Machine debian ne démarre plus, print en boucle :