

# **PPE 2**

*TRAÇABILITÉ ET SUPERVISION DE LA SANTÉ ET DE LA  
SÉCURITÉ DU SYSTÈME D'INFORMATION*

Ce document est protégé par des droits d'auteurs.

# Sommaire

## 1) Introduction

- a) Contexte
- b) Schéma
- c) Informations utiles

## 2) Zabbix

- a) Serveur Windows
- b) Serveur Linux
- c) Routeur/switch

## 3) Graylog

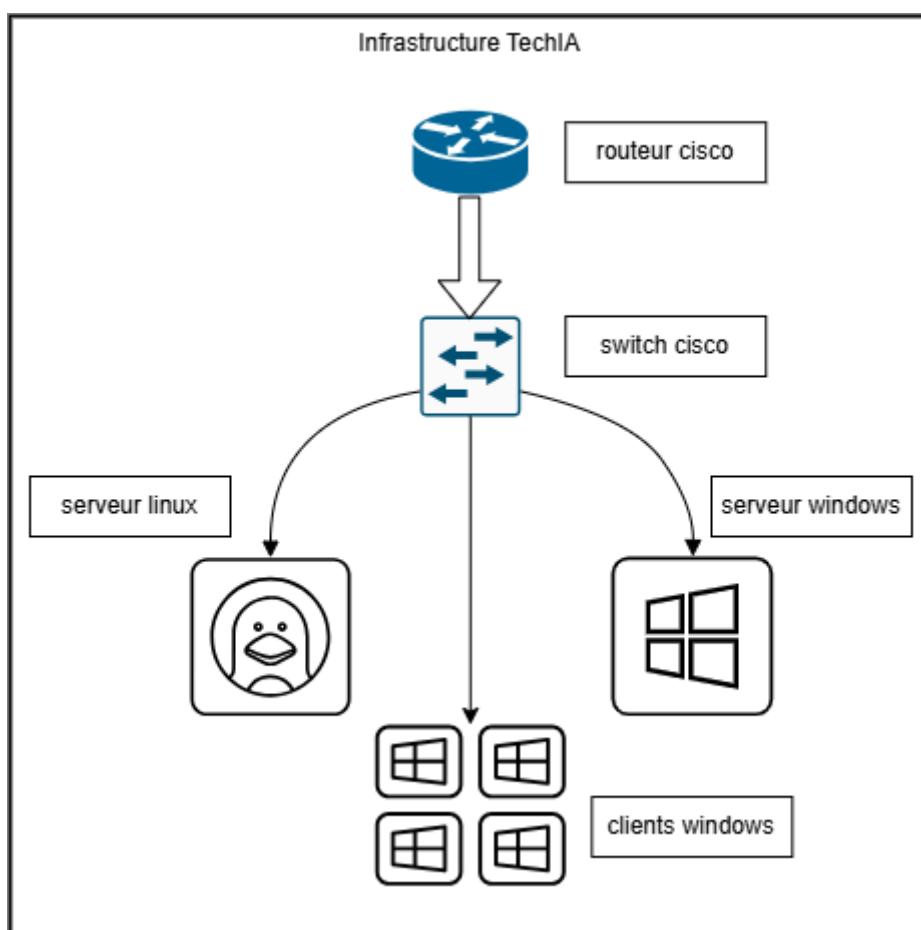
- a) Serveur Windows
- b) Serveur Linux

# 1) Introduction

## a) Contexte

Dans le cadre du projet de supervision et de traçabilité de l'infrastructure de TechIA, l'entreprise ZSecure, prestataire pour TechIA, a choisi d'installer **Zabbix** et **Graylog** sur une même machine Debian. **Zabbix** assurera la supervision en temps réel des équipements, incluant un serveur Windows, un serveur Linux, un switch et un routeur, permettant ainsi une surveillance de l'état de santé des appareils. En parallèle, **Graylog** sera déployé pour centraliser et analyser les logs générés par ces équipements, garantissant une meilleure traçabilité des événements.

## b) Schéma



voici le schéma de l'infrastructure de TechIA.

### c) Informations utiles

équipements	ip
routeur	192.168.1.1
switch	192.168.1.2
serveur linux	192.168.1.3
serveur windows	192.168.1.4
clients windows	192.168.1.10-20
Zabbix/Graylog	192.168.1.100

Les identifiants général : **admin / Azerty123!**

Sera configurer sur les hôtes, sauf windows, SNMP V3 (authPriv) :

- **USERNAME** → **admin**
- **PROTOCOLE\_PASSWORD** → **SHA**
- **PASSWORD** → **Azerty123 !**
- **PROTOCOLE\_PASSPHRASE** → **AES**
- **PASSPHRASE** → **Azerty123 !**

## 2) Zabbix

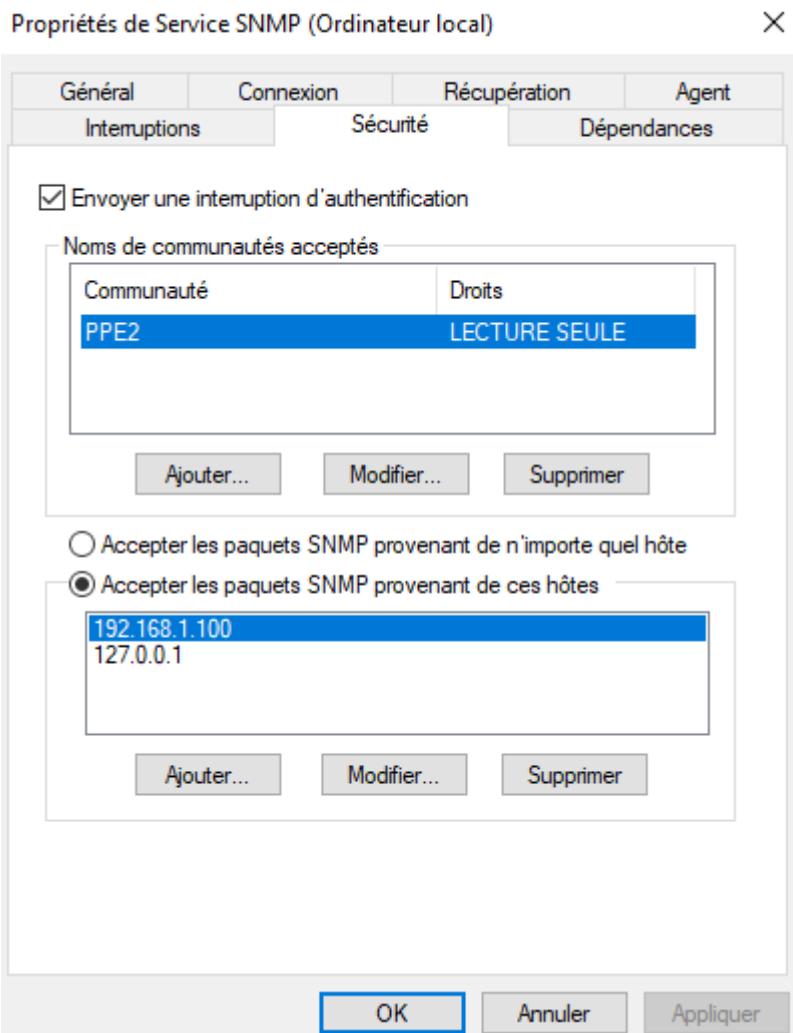
Routeur	192.168.1.1:161	SNMP	class: network target: cisco target: cisco-ios	Activé	Dernières données 96
SRV-LINUX	192.168.1.3:10050	ZBX	class: os class: software target: apache ...	Activé	Dernières données 78
SRV-WINDOWS	192.168.1.4:161	SNMP	class: os target: windows	Activé	Dernières données 23
switch	192.168.1.2:161	SNMP	class: network target: cisco target: cisco-ios	Activé	Dernières données 263

### a) Serveur Windows

Le serveur Windows sera supervisé pour les performances. La template Windows by SNMP sera utilisée.

Conf Windows :

Add-WindowsFeature SNMP-Service



## b) Serveur Linux

Le serveur linux sera supervisé pour les performances et le service web apache. La template Apache by Zabbix agent et Linux by Zabbix agent seront utilisées.

Conf linux :

```
sudo nano /etc/snmp/snmpd.conf
createUser admin SHA Azerty123! AES Azerty123!
rouser admin authPriv

sudo nano /etc/zabbix/zabbix_agentd.conf
Server=192.168.1.100
ServerActive=192.168.1.100
Hostname=SRV-LINUX
```

## c) Routeur / switch

Les appareils cisco seront supervisés pour le trafic réseaux et la connectique de câbles. La template Cisco IOS SNMP sera utilisée.

Configuration cisco :

```
snmp-server group SNMPv3Group v3 priv
snmp-server user admin SNMPv3Group v3 auth sha Azerty123!
priv aes 128 Azerty123!
```

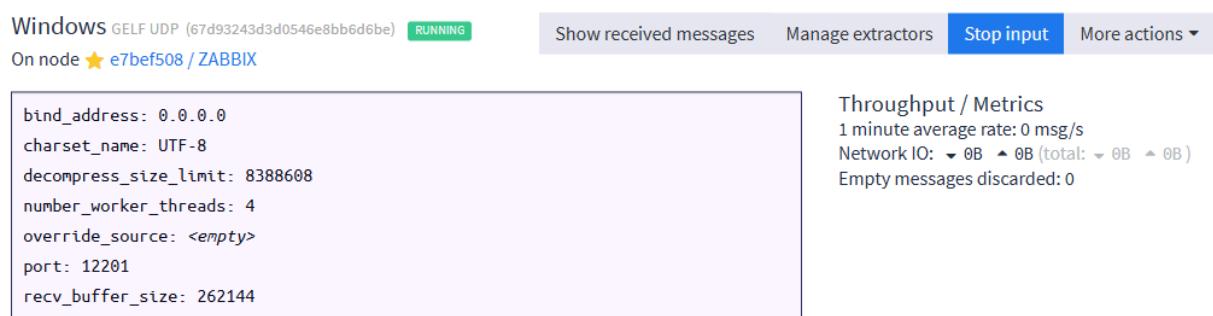
On peut donc maintenant surveiller nos hôtes :

Current problems						
Temps	▼	Info	Hôte	Problème • Sévérité	Durée	Actualiser
14:39:15	•		SRV-LINUX	Service ping Apache	7s	<a href="#">Actualiser</a>
14:24:33	•		switch	Cisco IOS: Interface Fa0/5(): Link down	14m 49s	<a href="#">Actualiser</a>

# 3) Graylog

## a) Serveur Windows

Sur graylogs : création d'un inputs GELF UDP



The screenshot shows the Graylog web interface with a running GELF UDP input. The configuration details are as follows:

```
bind_address: 0.0.0.0
charset_name: UTF-8
decompress_size_limit: 8388608
number_worker_threads: 4
override_source: <empty>
port: 12201
recv_buffer_size: 262144
```

Metrics on the right side show:

- Throughput / Metrics
- 1 minute average rate: 0 msg/s
- Network IO: ▼ 0B ▲ 0B (total: ▼ 0B ▲ 0B)
- Empty messages discarded: 0

Windows :

On va télécharger NXlog

Modifié dans C:\Program Files\Nxlog\conf\Nxlog.conf et ajouter à la fin:

```
<Input in>
    Module      im_msvistalog
</Input>

<Extension gelf>
    Module      xm_gelf
</Extension>

<Output graylog_udp>
    Module      om_udp
    Host        192.168.1.100 #IP graylog
    Port        12201 # PORT
    OutputType  GELF_UDP # TYPE de input
</Output>

<Route 1>
    Path        in => graylog_udp
</Route>
```

La stratégie de groupe est aussi modifiée pour auditer le rdp.

## b) Serveur Linux

Sur graylog : Création d'un inputs Syslog UDP

Local inputs 1 configured

Linux Syslog UDP (67d92e43d3d0546e8bb6c808) RUNNING

On node ★ e7bef508 / ZABBIX

Show received messages Manage extractors Stop input More actions ▾

```
allow_override_date: true
bind_address: 0.0.0.0
charset_name: UTF-8
expand_structured_data: false
force_rdns: false
number_worker_threads: 4
override_source: <empty>
port: 514
recv_buffer_size: 262144
store_full_message: true
timezone: NotSet
```

Throughput / Metrics

1 minute average rate: 0 msg/s

Network IO: ▼ 0B ▲ 0B (total: ▼ 0B ▲ 0B)

Empty messages discarded: 0

Création d'un index linux :

Linux index Stats are disabled by default

Index LINUX

Edit More Actions ▾

<b>Index prefix:</b>	linux_index
<b>Shards:</b>	1
<b>Replicas:</b>	0
<b>Field type refresh interval:</b>	5 seconds
<b>Field type profile:</b>	Not set
<b>Rotation strategy:</b>	Data Tiering
<b>Max. in storage:</b>	9 days
<b>Min. in storage:</b>	7 days

Création d'un stream et règles stream :

## Rules of Stream "Linux stream"

This screen is dedicated to an easy and comfortable creation and manipulation of stream rules. The rules have an effect on message matching here.

### 1. Load a message to test rules

Recent Message

Message ID

Select an Input from the list below and click "Load Message" to load the most recent message in the hour.

Select an Input

Load Message

### 2. Manage stream rules

- A message must match all of the following rules
- A message must match at least one of the following rules

**! Please load a message in Step 1 above to check if it would match against these rules.**

  **gl\_source\_input must** match input *Linux (Syslog UDP: 67d92e43d3d0546e8bb6c808)*

I'm done!

Sur linux :

```
sudo apt-get install rsyslog
sudo nano /etc/rsyslog.d/10-graylog.conf
.*.* @192.168.1.100:514;RSYSLOG_SyslogProtocol23Format
sudo systemctl restart rsyslog.service
```

On peut donc voir les logs de notre linux et du windows :

2025-03-18 12:48:39.368	SRV-LINUX
pan_unix(login:session): session closed for user root	
2025-03-18 12:48:39.368	SRV-LINUX
pan_unix(login:session): session closed for user root	
2025-03-18 12:48:34.000	SRV-WIN-PPE2
2025-03-18 13:48:34 SRV-WIN-PPE2 INFO 2050 AUTORITE NT\SERVICE L	
2025-03-18 12:48:34.000	SRV-WIN-PPE2
2025-03-18 13:48:34 SRV-WIN-PPE2 INFO 2050 AUTORITE NT\SERVICE L	