

NOUVEAU

TUTORIEL PRATIQUE



INSTALLER ET CONFIGURER ZEEK 8.1.1 SUR DEBIAN 13

De l'installation de base aux premières manipulations

POINTS CLES DU TUTORIEL



ZEEK : QU'EST-CE QUE C'EST ?



INSTALLATION PAS A PAS DE ZEEK (version 8.1.1)



TESTS D'ALERTE ZEEK



POUR ALLER PLUS LOIN... CREATION D'UN TABLEAU DE BORD EN PHP

Notes :

Ce tutoriel décrit l'installation de ZEEK 8 sur une machine Debian 13.3. L'installation est présentée sous la forme d'une machine virtuelle créée à partir de l'hyperviseur © Proxmox.

Dans ce tutoriel, nous supposons que la machine virtuelle est connectée à l'interface « LAN » de votre pare-feu et qu'elle a obtenu une adresse IP dynamique et à accès à Internet.

1 – ZEEK : QU'EST-CE QUE C'EST ?

Zeek (anciennement Bro), tout comme Snort, est un **outil de surveillance réseau** mais fondamentalement différent dans son approche par rapport à Snort. Le tableau ci-dessous permet de comparer les 2 outils :

Critère	Snort	Zeek
Approche	Détection par règles de signatures	Analyse comportementale et journalisation
Paradigme	IDS/IPS (blocage possible)	NSM — Network Security Monitor (analyse)
Détection	Correspondance de patterns	Scripts Zeek (langage propre)
Logs générés	Alertes uniquement	Logs structurés par protocole (dns, http...)
Faux positifs	Plus fréquents	Moins fréquents (analyse de contexte)
Ressources	Légères	Moyennes (plus de traitements)
Usage typique	Détection d'intrusion temps réel	Audit réseau, détection avancée
Format des logs	alert_fast, alert_json	TSV ou JSON par type de protocole

Zeek ne bloque pas le trafic : c'est un outil d'analyse et de surveillance. Il est souvent déployé en complément de Snort ou d'un pare-feu.

Zeek génère des logs structurés et organisés par protocole, tous stockés dans « **/opt/zeek/logs/current/** » :

Fichier de log	Contenu
conn.log	Toutes les connexions réseau (TCP, UDP, ICMP) avec durée, volume, état
dns.log	Toutes les requêtes et réponses DNS
http.log	Requêtes HTTP : méthode, URL, user-agent, réponse
ssl.log	Connexions TLS/SSL : certificat, version, cipher
ssh.log	Sessions SSH détectées
ftp.log	Commandes FTP
smtp.log	Sessions SMTP (emails)
files.log	Fichiers transférés sur le réseau

Fichier de log	Contenu
weird.log	Comportements anormaux détectés par Zeek
notice.log	Alertes générées par les scripts Zeek (équivalent alertes Snort)
intel.log	Correspondances avec des indicateurs de compromission (IOC)

2 – INSTALLATION DE ZEEK 8 (depuis le dépôt officiel)

La documentation officielle est disponible ici : [Zeek Documentation — Book of Zeek \(8.0.6\)](#)

Etape 1 : Installation des prérequis pour gérer les dépôts

```
sudo apt update && sudo apt full-upgrade -y
sudo apt install -y curl gnupg apt-transport-https ca-certificates
```

Téléchargement et ajout de la clé GPG (depuis le dépôt Zeek) :

```
curl -fsSL https://download.opensuse.org/repositories/security:/zeek/Debian_Testing/Release.key \
| sudo gpg --dearmor -o /usr/share/keyrings/zeek.gpg
```

Ajout du dépôt Zeek pour Debian Testing (compatible Debian 13) :

```
echo "deb [signed-by=/usr/share/keyrings/zeek.gpg] \
https://download.opensuse.org/repositories/security:/zeek/Debian_Testing/ /" \
| sudo tee /etc/apt/sources.list.d/zeek.list
```

Etape 2 : Installation de Zeek (patientez pendant l'installation des nombreux paquets)

```
sudo apt update
sudo apt install -y zeek
```

Etape 3 : ajout de Zeek au « path » de la machine Debian

```
echo 'export PATH=$PATH:/opt/zeek/bin' | sudo tee /etc/profile.d/zeek.env
source /etc/profile.d/zeek.env
```

Vérification

```
which zeek
which zeekctl
```

Vérification de la version installée

```
zeek -version
```

Retour attendu :

```
prof@DEBIAN13-ZEEK-PROF:~$ zeek -version
zeek version 8.1.1
```

Zeek s'installe dans « /opt/zeek ». Le binaire principal est « /opt/zeek/bin/zeek ». ZeekControl (zeekctl) est l'outil de gestion.

Comme pour Snort, l'offloading doit être désactivé pour que Zeek reçoive des paquets complets et non réassemblés par la carte réseau et le mode « promiscuité » doit être activé sur l'interface réseau de la machine Zeek.

A propos de « l'offloading »

Les cartes réseau modernes intègrent des fonctionnalités matérielles (« offloading ») pour alléger le travail du CPU. Parmi elles, **Generic Receive Offload (GRO)** et **Large Receive Offload (LRO)** sont problématiques pour Zeek .

Ces fonctionnalités réassemblent les fragments de paquets en paquets plus gros avant de les transmettre au système d'exploitation. Pour un NSM comme Zeek, cela pose deux problèmes majeurs :

- **Perte de visibilité** : Zeek ne voit plus les petits fragments individuels, ce qui peut lui faire rater des attaques conçues pour se cacher dans la fragmentation.
- **Problèmes d'analyse** : Les gros paquets reconstitués peuvent dépasser la taille maximale analysable par Zeek (snaplen) et perturber le moteur de reassemblage de flux (Stream), entraînant des erreurs d'analyse et des faux négatifs.

La recommandation officielle de l'équipe Zeek est donc de désactiver GRO et LRO sur l'interface dédiée à l'inspection.

A propos de la configuration de l'interface réseau en mode « promiscuité »

Pour que Zeek puisse analyser l'ensemble du trafic réseau sur une interface, **il est souvent nécessaire de placer l'interface réseau en mode promiscuous (ou mode promisc)**. Cette configuration est essentielle pour un système de détection d'intrusion, car **elle permet de capturer tous les paquets qui passent par l'interface**, même ceux qui ne lui sont pas directement destinés. Voyons en détail ce qu'est le mode promisc et comment l'activer pour une interface réseau.

Qu'est-ce que le mode « PROMISC » ?

En fonctionnement normal, une carte réseau filtre le trafic pour ne traiter que les paquets adressés spécifiquement à son adresse MAC ou diffusés en **broadcast**. Autrement dit, elle ignore les paquets destinés aux autres adresses du réseau, ce qui est généralement suffisant pour des usages standards.

En revanche, en « **mode promiscuous** » (ou mode « promisc »), l'interface réseau capture **tous les paquets** qui transitent, peu importe leur destination. Cette configuration est idéale pour des outils de surveillance réseau comme Zeek, car elle permet d'obtenir une vue complète du trafic circulant sur le réseau. Cela aide Zeek à détecter des menaces, même si celles-ci visent d'autres hôtes que celui sur lequel Zeek est installé.

Comme pour Snort, l'offloading doit être désactivé pour que Zeek reçoive des paquets complets et non réassemblés par la carte réseau.

Etape 1 : Installation de l'outil « ethtool » et désactivation de l'offloading

sudo apt install -y ethtool

Vérification de l'état actuel de l'interface réseau « ens18 »

sudo ethtool -k ens18 | egrep 'large-receive-offload|generic-receive-offload|tcp-segmentation-offload'

```
tcp-segmentation-offload: on
generic-receive-offload: on
large-receive-offload: off [fixed]
```

Désactivation GRO et TSO

sudo ethtool -K ens18 gro off tso off

Rendre la désactivation permanente

sudo nano /etc/systemd/system/ethtool-zeek.service

Copier le contenu suivant :

[Unit]

Description=Disable NIC offloading for Zeek

Before=network.target zeek.service

Wants=network.target

[Service]

Type=oneshot

ExecStart=/usr/sbin/ethtool -K ens18 gro off tso off

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

sudo systemctl daemon-reload

sudo systemctl enable ethtool-zeek.service

Quittez et sauvegardez le fichier.

Activation immédiate du mode promiscuité

Assurez-vous que l'interface réseau de votre machine Zeek porte bien le nom « **ens18** » avec la commande « **ip a** » :

```
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:8a:3e:c1 brd ff:ff:ff:ff:ff:ff
    altname enp6s18
    altname enxbc24118a3ec1
    inet 10.10.10.103/24 brd 10.10.10.255 scope global dynamic noprefixroute ens18
        valid_lft 2121sec preferred_lft 2671sec
```

Saisissez la commande suivante :

sudo ip link set ens18 promisc on

Rendre le mode promiscuité permanent en modifiant le fichier « /etc/network/interfaces » :

sudo nano /etc/network/interfaces

Ajoutez ce bloc au fichier :

Activation du mode promiscuité

up ip link set dev ens18 promisc on

down ip link set dev ens18 promisc off

```
# Activation du mode promiscuité
up ip link set dev ens18 promisc on
down ip link set dev ens18 promisc off
```

Quittez et sauvegardez le fichier. Relancez le service réseau avec la commande suivante :

sudo systemctl restart networking

Vérifiez que le mode promiscuité est bien actif sur l'interface de votre machine avec la commande suivante :

ip link show ens18 | grep PROMISC

Retour attendu :

```
proF@DEBIAN13-ZEEK-PROF:~$ ip link show ens18 | grep PROMISC
2: ens18: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
```

4 – CONFIGURATION DE ZEEK

Etape 1 : modification du fichier « **node.cfg** »

Le fichier « **node.cfg** » définit l'architecture de déploiement de Zeek. En mode standalone (une seule machine), la configuration est simple.

Editez le fichier « **node.cfg** » qui se trouve dans « **/opt/zeek/etc** » :

sudo nano /opt/zeek/etc/node.cfg

Modifiez l'interface réseau en indiquant le nom de votre interface réseau ; ici nous notons « **ens18** » :

```
[zeek]
type=standalone
host=localhost
interface=ens18
```

Etape 2 : modification du fichier « **networks.cfg** »

Ce fichier indique à Zeek quelles plages IP constituent le réseau local. Les connexions internes seront traitées différemment des connexions externes. Pour cela, saisissez la commande suivante :

sudo nano /opt/zeek/etc/networks.cfg

Contenu à adapter (**attention, adaptez l'adresse réseau à votre architecture !**) :

```
# Réseau local surveillé (à modifier en fonction de votre architecture réseau)
10.10.10.0/24   Réseau LAN BTS SIO
```

```
# Réseau local à surveiller
10.10.10.0/24   Réseau LAN BTS SIO
```

Quittez et sauvegardez le fichier.

Etape 3 : configuration du fichier « **zeekctl.cfg** » en ajoutant quelques options supplémentaires

sudo nano /opt/zeek/etc/zeekctl.cfg

LogRotationInterval

- 3600 = rotation toutes les heures
- Pour un lab : OK
- En production : souvent 1h ou 24h

LogExpireInterval

- 0 = ne supprime jamais
- En production : mettre par exemple 7day

```
#####
# Logging Options

# Rotation interval in seconds for log files on manager (or standalone) node.
# A value of 0 disables log rotation.
LogRotationInterval = 3600

# Compression des logs archivés
CompressLogs = 1
CompressCmd = gzip

# Expiration interval for archived log files in LogDir. Files older than this
# will be deleted by "zeekctl cron". The interval is an integer followed by
# one of these time units: day, hr, min. A value of 0 means that logs
# never expire.
LogExpireInterval = 1day
```

Etape 4 : Activation des scripts de détection

Zeek utilise des « **scripts.zeek** » pour l'analyse. Le répertoire « **local.zeek** » est le point d'entrée pour les personnalisations.

sudo nano /opt/zeek/share/zeek/site/local.zeek

Ce fichier est déjà prérempli. Vous pouvez en ajouter d'autres comme celles ci-dessous :

```
# =====
# local.zeek — Scripts de détection personnalisés
# VM Zeek — BTS SIO SISR
# =====

# Chargement des scripts standard Zeek
@load base/frameworks/notice
@load base/protocols/conn
@load base/protocols/dns
@load base/protocols/ftp
@load base/protocols/http
@load base/protocols/smtp
@load base/protocols/ssh
@load base/protocols/ssl

# Détection de scans de ports
@load policy/misc/scan

# Détection de mots de passe en clair sur FTP/HTTP
@load policy/protocols/ftp/detect
@load policy/protocols/http/detect-sqli

# Extraction des métadonnées de fichiers
@load base/frameworks/files

# Détection des connexions SSH par brute force
@load policy/protocols/ssh/detect-bruteforcing

# Intelligence sur les menaces (IOC)
@load frameworks/intel/seen
@load frameworks/intel/do_notice
```

```
# Journalisation JSON (recommandé pour intégration avec dashboards)
@load tuning/json-logs
```

```
# =====
# Seuils personnalisés
# =====
```

```
# Brute force SSH : alerte après 10 tentatives en 5 minutes
redef SSH::password_guesses_limit = 10;
```

```
# Scan de ports : seuil de détection
redef Scan::addr_scan_threshold = 20.0;
redef Scan::port_scan_threshold = 20.0;
```

Etape 5 : création d'un utilisateur « zeek »

```
# Créer un utilisateur système « zeek »
sudo useradd -r -g zeek -d /opt/zeek -s /usr/sbin/nologin zeek
```

```
# Donner la propriété des répertoires
sudo chown -R zeek:zeek /opt/zeek/logs
sudo chown -R zeek:zeek /opt/zeek/spool
```

```
# Capacités réseau sur le binaire (écoute sans root)
sudo setcap cap_net_raw,cap_net_admin=eip /opt/zeek/bin/zeek
```

```
# Vérification
sudo getcap /opt/zeek/bin/zeek
```

Retour attendu :

```
/opt/zeek/bin/zeek cap_net_admin,cap_net_raw=eip
```

5 – DEMARRAGE ET UTILISATION DE ZEEK

ZeekControl (zeekctl) est l'outil de gestion de Zeek. Il gère le démarrage, l'arrêt, la rotation des logs et le monitoring.

```
# Installez le module « python3-websockets »
sudo apt install python3-websockets
```

```
# Initialisation de Zeek (à refaire si vous modifier le fichier « local.zeek »)
sudo -u zeek /opt/zeek/bin/zeekctl deploy
```

Retour attendu :

```
prof@DEBIAN13-ZEEK-PROF:~$ sudo -u zeek /opt/zeek/bin/zeekctl deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
starting ...
starting zeek ...
```

La commande deploy effectue :

1. install → copie les fichiers de configuration
2. start → démarre Zeek

La sortie doit afficher "starting zeek ... done"

Afin de pouvoir saisir les commandes ci-dessous et si vous avez un utilisateur autre que « zeek », nous devons modifier le « **secure_path** » actuel. Pour cela, effectuez les manipulations ci-dessous :

Vérifiez le **secure_path** actuel

sudo visudo

Recherchez la ligne « **Defaults secure_path** ». Sur Debian elle ressemble généralement à :

« **Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"** »

```
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
```

Ajoutez « **/opt/zeek/bin** » à la fin de la ligne :

Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:**/opt/zeek/bin**"

```
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/opt/zeek/bin"
```

Quittez et enregistrez le fichier.

Déconnectez-vous de la session SSH et reconnectez-vous en SSH à votre machine Zeek.

Vous devriez pouvoir saisir les commandes ci-dessous, depuis un utilisateur autre que « zeek », sans erreur.

Vérifiez la configuration avant démarrage

sudo zeekctl check

```
prof@DEBIAN13-ZEEK-PROF:~$ sudo zeekctl check
zeek scripts are ok.
```

Quelques commandes à connaître :

Démarrer Zeek

sudo zeekctl start

```
prof@DEBIAN13-ZEEK-PROF:~$ sudo zeekctl start
starting zeek ...
```

Arrêter Zeek

sudo zeekctl stop

```
prof@DEBIAN13-ZEEK-PROF:~$ sudo zeekctl stop
stopping zeek ...
```

Redémarrer (après modification de config)

sudo zeekctl restart

```
prof@DEBIAN13-ZEEK-PROF:~$ sudo zeekctl restart
stopping ...
stopping zeek ...
starting ...
starting zeek ...
```

Statut des processus

sudo zeekctl status

```
prof@DEBIAN13-ZEEK-PROF:~$ sudo zeekctl status
Name          Type          Host          Status      Pid    Started
zeek          standalone   localhost     running     45559  04 Mar 16:06:37
```

Déployer après modification du fichier « **local.zeek** »

sudo zeekctl deploy

6 – COMMANDES ESSENTIELLES

Avant d'exécuter les commandes ci-dessous, il est recommandé de mettre votre utilisateur actuel dans le groupe « zeek ». Pour cela saisissez les commandes suivantes (**ici notre utilisateur s'appelle « prof » ; à adapter !**) :

sudo usermod -aG zeek prof

sudo chmod -R 750 /opt/zeek/logs/

Déconnectez-vous et reconnectez-vous en SSH à votre machine.

Vérifier que les logs sont générés

sudo ls -la /opt/zeek/logs/current/

```
prof@DEBIAN13-ZEEK-PROF:~$ sudo ls -la /opt/zeek/logs/current/
total 192
drwxr-sr-x 2 root zeek 4096 4 mars 16:08 .
drwxrws--- 7 zeek zeek 4096 4 mars 16:07 ..
-rw-r--r-- 1 root zeek 250 4 mars 16:08 capture_loss.log
-rw-r--r-- 1 root zeek 129 4 mars 16:07 .cmdline
-rw-r--r-- 1 root zeek 13447 4 mars 16:11 conn.log
-rw-r--r-- 1 root zeek 366 4 mars 16:07 .env_vars
-rw-r--r-- 1 root zeek 34933 4 mars 16:07 loaded_scripts.log
-rw-r--r-- 1 root zeek 753 4 mars 16:08 notice.log
-rw-r--r-- 1 root zeek 251 4 mars 16:07 packet_filter.log
-rw-r--r-- 1 root zeek 6 4 mars 16:07 .pid
-rw-r--r-- 1 root zeek 666 4 mars 16:07 reporter.log
-rw-r--r-- 1 root zeek 60 4 mars 16:07 .startup
-rw-r--r-- 1 root zeek 710 4 mars 16:07 stats.log
-rwxr-xr-x 1 root zeek 19 4 mars 16:07 .status
-rw-r--r-- 1 root zeek 20 4 mars 16:07 stderr.log
-rw-r--r-- 1 root zeek 204 4 mars 16:07 stdout.log
-rw-r--r-- 1 root zeek 79290 4 mars 16:11 telemetry.log
-rw-r--r-- 1 root zeek 1449 4 mars 16:11 weird.log
```

```
sudo tail -f /opt/zeek/logs/current/conn.log
```

Suivre les alertes (notices)

```
sudo tail -f /opt/zeek/logs/current/notice.log
```

Suivre les requêtes DNS

```
sudo tail -f /opt/zeek/logs/current/dns.log
```

Suivre les connexions SSH

```
sudo tail -f /opt/zeek/logs/current/ssh.log
```

Suivre les comportements anormaux

```
sudo tail -f /opt/zeek/logs/current/weird.log
```

```
# Extraire uniquement les colonnes utiles de conn.log
```

```
sudo zeek-cut id.orig_h id.resp_h id.resp_p proto duration \  
< /opt/zeek/logs/current/conn.log | head -20
```

Top 10 des IP sources les plus actives

```
sudo zeek-cut id.orig_h < /opt/zeek/logs/current/conn.log \
| sort | uniq -c | sort -rn | head -10
```

Toutes les requêtes DNS du réseau

```
sudo zeek-cut ts id.orig_h query answers \  
< /opt/zeek/logs/current/dns.log | head -20
```

Connexions SSH détectées

```
sudo zeek-cut ts id.orig_h id.resp_h auth_success \  
< /opt/zeek/logs/current/ssh.log
```

```
# Alertes de sécurité
sudo zeek-cut ts note msg src \
< /opt/zeek/logs/current/notice.log
```

7 – CREATION D'UN SERVICE ZEEK AU DEMARRAGE DE LA MACHINE

Création du fichier « **zeek.service** »

```
sudo nano /etc/systemd/system/zeek.service
```

Copiez le contenu suivant :

```
[Unit]
Description=Zeek Network Security Monitor
After=network.target ethtool-zeek.service
Wants=network.target
```

```
[Service]
Type=forking
User=root
ExecStart=/opt/zeek/bin/zeekctl start
ExecStop=/opt/zeek/bin/zeekctl stop
ExecReload=/opt/zeek/bin/zeekctl restart
Restart=on-failure
RestartSec=10
```

```
[Install]
WantedBy=multi-user.target
```

Activation et démarrage du service zeek

```
sudo systemctl daemon-reload
sudo systemctl enable zeek.service
sudo systemctl start zeek.service
```

Vérification du statut de Zeek

```
sudo systemctl status zeek.service
```

```
prof@DEBIAN13-ZEEK-PROF:/opt/zeek/logs/current$ sudo systemctl status zeek.service
○ zeek.service - Zeek Network Security Monitor
   Loaded: loaded (/etc/systemd/system/zeek.service; enabled; preset: enabled)
   Active: inactive (dead) since Wed 2026-03-04 16:34:32 CET; 45s ago
 Invocation: a55e4bb0eae349ab9f946c88a7c658de
    Process: 47763 ExecStart=/opt/zeek/bin/zeekctl start (code=exited, status=0/SUCCESS)
    Process: 47782 ExecStop=/opt/zeek/bin/zeekctl stop (code=exited, status=0/SUCCESS)
   Mem peak: 24M
      CPU: 405ms
```

8 – CREATION D'UN TABLEAU DE BORD

Etape 1 : installation Apache et PHP

```
sudo apt install -y apache2 php libapache2-mod-php php-json
sudo systemctl enable apache2
sudo systemctl start apache2
```

Etape 2 : attribution des permissions sur les logs Zeek

Ajout de l'utilisateur « www-data » au groupe zeek

```
sudo usermod -aG zeek www-data
```

Réglage des permissions sur le répertoire de logs

```
sudo chmod 750 /opt/zeek/logs
```

```
sudo chmod 750 /opt/zeek/logs/current
```

```
sudo chmod 640 /opt/zeek/logs/current/*.log 2>/dev/null || true
```

Redémarrer Apache

```
sudo systemctl restart apache2
```

Etape 3 : création du tableau de bord en PHP

```
sudo mkdir -p /var/www/zeek-dashboard && sudo nano /var/www/zeek-dashboard/index.php
```

Contenu à copier :

```
<?php
// =====
// Zeek Dashboard — Interface BTS SIO SISR
// Lecture des logs JSON Zeek 8.x
// =====

$logDir = '/opt/zeek/logs/current/';
$maxRows = 150;

function readLog($file, $max=100) {
    if (!file_exists($file)) return [];
    $lines = array_reverse(array_filter(explode("\n", file_get_contents($file))));
    $out = [];
    foreach (array_slice($lines, 0, $max) as $l) {
        $d = json_decode(trim($l), true);
        if ($d) $out[] = $d;
    }
    return $out;
}

$conns = readLog($logDir.'conn.log', $maxRows);
$notices = readLog($logDir.'notice.log', $maxRows);
$dns = readLog($logDir.'dns.log', 50);
$ssh = readLog($logDir.'ssh.log', 50);
$http = readLog($logDir.'http.log', 50);
$weird = readLog($logDir.'weird.log', 50);

// Stats connexions
$totalConn = count($conns);
$totalNotice= count($notices);
$protos = array_count_values(array_column($conns, 'proto'));
$topSrc = array_count_values(array_column($conns, 'id.orig_h'));
arsort($topSrc);
$topSrc5 = array_slice($topSrc, 0, 5, true);
$topDst = array_count_values(array_column($conns, 'id.resp_h'));
arsort($topDst);
$topDst5 = array_slice($topDst, 0, 5, true);
```

```

?>
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="UTF-8">
<meta http-equiv="refresh" content="30">
<title>Zeek — Dashboard</title>
<style>
*{box-sizing:border-box;margin:0;padding:0}
body{font-family:'Segoe UI',Arial,sans-serif;background:#0a1a0f;color:#e0e0e0}
header{background:linear-gradient(135deg,#0d2b14,#1a5c2a);padding:20px 30px;
display:flex;align-items:center;justify-content:space-between;
border-bottom:2px solid #2ecc71}
header h1{font-size:1.6em;color:#fff;letter-spacing:1px}
header h1 span{color:#2ecc71}
.badge{background:#e74c3c;color:#fff;padding:6px 16px;border-radius:20px;font-size:.85em;font-weight:bold}
.container{max-width:1400px;margin:0 auto;padding:24px}
.tabs{display:flex;gap:4px;margin-bottom:20px;flex-wrap:wrap}
.tab{padding:8px 18px;background:#1a2e1c;border:1px solid #2d4a2f;border-radius:6px 6px 0 0;
cursor:pointer;font-size:.85em;color:#aaa;transition:all .2s}
.tab.active,.tab:hover{background:#1e6b3a;color:#fff;border-color:#2ecc71}
.tab-content{display:none}.tab-content.active{display:block}
.cards{display:grid;grid-template-columns:repeat(auto-fit,minmax(160px,1fr));gap:14px;margin-bottom:20px}
.card{background:#0f2412;border-radius:10px;padding:18px;text-align:center;border-left:4px solid #2ecc71}
.card.red{border-color:#e74c3c}.card.blue{border-color:#3498db}.card.orange{border-color:#f39c12}
.card .num{font-size:2em;font-weight:bold;color:#2ecc71}
.card.red .num{color:#e74c3c}.card.blue .num{color:#3498db}.card.orange .num{color:#f39c12}
.card .label{font-size:.8em;color:#666;margin-top:5px;text-transform:uppercase;letter-spacing:1px}
.grid2{display:grid;grid-template-columns:1fr 1fr;gap:16px;margin-bottom:20px}
@media(max-width:900px){.grid2{grid-template-columns:1fr}}
.panel{background:#0f2412;border-radius:10px;padding:18px}
.panel h2{font-size:.9em;color:#2ecc71;margin-bottom:14px;text-transform:uppercase;
letter-spacing:1px;border-bottom:1px solid #1a3a1c;padding-bottom:8px}
.bar-row{display:flex;align-items:center;margin-bottom:8px;font-size:.85em}
.bar-label{width:180px;white-space:nowrap;overflow:hidden;text-overflow:ellipsis;color:#ccc}
.bar-wrap{flex:1;background:#1a2e1c;border-radius:4px;height:16px;margin:0 8px}
.bar-fill{height:100%;border-radius:4px;background:linear-gradient(90deg,#1a5c2a,#2ecc71);min-width:4px}
.bar-fill.red{background:linear-gradient(90deg,#7b1818,#e74c3c)}
.bar-count{color:#f39c12;font-weight:bold;min-width:30px;text-align:right}
table{width:100%;border-collapse:collapse;font-size:.82em}
thead th{background:#071209;color:#2ecc71;padding:9px 10px;text-align:left;
font-size:.78em;text-transform:uppercase;letter-spacing:1px;position:sticky;top:0}
tbody tr{border-bottom:1px solid #122115;transition:background .15s}
tbody tr:hover{background:#142918}
td{padding:7px 10px;color:#ccc}
table-wrap{overflow-x:auto;max-height:400px;overflow-y:auto}
.proto{display:inline-block;padding:2px 7px;border-radius:4px;font-size:.78em;font-weight:bold}
.proto-tcp{background:#0d2b3e;color:#3498db}
.proto-udp{background:#0d2b14;color:#2ecc71}
.proto-icmp{background:#2b1a0d;color:#f39c12}
.warn-row td{color:#f39c12}
.notice-row td{color:#e74c3c;font-weight:500}
.refresh{color:#556;font-size:.78em;margin-top:3px}
footer{text-align:center;padding:16px;color:#333;font-size:.78em;
border-top:1px solid #0f2412;margin-top:10px}
</style>

```

```

</head>
<body>
<header>
  <div>
    <h1>&#128200; Zeek <span>7</span> — Dashboard</h1>
    <div class="refresh">Actualisation automatique toutes les 30s</div>
  </div>
  <div style="text-align:right">
    <div class="badge"><?=$totalConn?> connexions</div>
    <?php if($totalNotice>0): ?>
      <div class="badge" style="background:#e74c3c;margin-top:4px"><?=$totalNotice?> alertes</div>
    <?php endif; ?>
    <div style="color:#446;font-size:.75em;margin-top:4px"><?=date('d/m/Y H:i:s')?></div>
  </div>
</header>

<div class="container">

<div class="tabs">
  <div class="tab active" onclick="showTab('vue')">&#128202; Vue générale</div>
  <div class="tab" onclick="showTab('conn')">&#128279; Connexions</div>
  <div class="tab" onclick="showTab('dns')">&#127760; DNS</div>
  <div class="tab" onclick="showTab('ssh')">&#128272; SSH</div>
  <div class="tab" onclick="showTab('http')">&#127760; HTTP</div>
  <div class="tab" onclick="showTab('notice')">&#128680; Alertes</div>
  <div class="tab" onclick="showTab('weird')">&#9888;&#65039; Weird</div>
</div>

<!-- VUE GÉNÉRALE -->
<div id="tab-vue" class="tab-content active">
  <div class="cards">
    <div class="card"><div class="num"><?=$totalConn?></div><div class="label">Connexions</div></div>
    <div class="card red"><div class="num"><?=$totalNotice?></div><div class="label">Alertes</div></div>
    <div class="card blue"><div class="num"><?=$protos['tcp']??0?></div><div class="label">TCP</div></div>
    <div class="card blue"><div class="num"><?=$protos['udp']??0?></div><div class="label">UDP</div></div>
    <div class="card orange"><div class="num"><?=$protos['icmp']??0?></div><div
class="label">ICMP</div></div>
    <div class="card"><div class="num"><?=count($dns)?></div><div class="label">Requêtes DNS</div></div>
    <div class="card"><div class="num"><?=count($ssh)?></div><div class="label">Sessions SSH</div></div>
    <div class="card orange"><div class="num"><?=count($weird)?></div><div class="label">Weird</div></div>
  </div>
  <div class="grid2">
    <div class="panel">
      <h2>&#127760; Top 5 — IP sources</h2>
      <?php $maxV=max(array_values($topSrc5)?:[1]);
      foreach($topSrc5 as $ip=>$cnt): ?>
        <div class="bar-row">
          <div class="bar-label"><?=htmlspecialchars($ip??'N/A')?></div>
          <div class="bar-wrap"><div class="bar-fill red" style="width:<?=round(($cnt/$maxV)*100)?%></div></div>
          <div class="bar-count"><?=$cnt?></div>
        </div>
      <?php endforeach; ?>
    </div>
    <div class="panel">
      <h2>&#127989; Top 5 — IP destinations</h2>
      <?php $maxV=max(array_values($topDst5)?:[1]);

```

```

foreach($stopDst5 as $ip=>$cnt): ?>
<div class="bar-row">
  <div class="bar-label"><?=htmlspecialchars($ip??'N/A')?></div>
  <div class="bar-wrap"><div class="bar-fill" style="width:<?=round(($cnt/$maxV)*100)?>%"></div></div>
  <div class="bar-count"><?=$cnt?></div>
</div>
<?php endforeach; ?>
</div>
<?php if(!empty($notices)): ?>
<div class="panel">
  <h2>&#128680; Dernières alertes (notice.log)</h2>
  <div class="table-wrap"><table>
    <thead><tr><th>Timestamp</th><th>Type</th><th>Message</th><th>Source</th></tr></thead>
    <tbody>
      <?php foreach($notices as $n): ?>
      <tr class="notice-row">
        <td><?=htmlspecialchars($n['ts']??'-')?></td>
        <td><?=htmlspecialchars($n['note']??'-')?></td>
        <td><?=htmlspecialchars($n['msg']??'-')?></td>
        <td><?=htmlspecialchars($n['src']??'-')?></td>
      </tr>
      <?php endforeach; ?>
    </tbody>
  </table></div>
</div>
<?php endif; ?>
</div>

<!-- CONNEXIONS -->
<div id="tab-conn" class="tab-content">
  <div class="panel">
    <h2>&#128279; Connexions réseau (conn.log) — <?=count($conns)?> entrées</h2>
    <div class="table-wrap"><table>
      <thead><tr><th>Timestamp</th><th>Src IP</th><th>Src Port</th><th>Dst IP</th><th>Dst
Port</th><th>Proto</th><th>Durée</th><th>Octets</th><th>État</th></tr></thead>
      <tbody>
        <?php foreach($conns as $c):
          $p=strtolower($c['proto']??''); ?>
        <tr>
          <td><?=htmlspecialchars($c['ts']??'-')?></td>
          <td><?=htmlspecialchars($c['id.orig_h']??'-')?></td>
          <td><?=htmlspecialchars($c['id.orig_p']??'-')?></td>
          <td><?=htmlspecialchars($c['id.resp_h']??'-')?></td>
          <td><?=htmlspecialchars($c['id.resp_p']??'-')?></td>
          <td><span class="proto"><?=$p?><?=strtoupper($p)?></span></td>
          <td><?=htmlspecialchars(round($c['duration']??0,3))?>s</td>
          <td><?=htmlspecialchars($c['orig_bytes']??0)?></td>
          <td><?=htmlspecialchars($c['conn_state']??'-')?></td>
        </tr>
        <?php endforeach; ?>
      </tbody>
    </table></div>
  </div>
</div>

```

```

<!-- DNS -->
<div id="tab-dns" class="tab-content">
  <div class="panel">
    <h2>#127760; Requêtes DNS (dns.log)</h2>
    <div class="table-wrap"><table>

<thead><tr><th>Timestamp</th><th>Client</th><th>Requête</th><th>Type</th><th>Réponse</th></tr></thead>
<tbody>
  <?php foreach($dns as $d): ?>
    <tr>
      <td><?=htmlspecialchars($d['ts']??'-')?></td>
      <td><?=htmlspecialchars($d['id.orig_h']??'-')?></td>
      <td><?=htmlspecialchars($d['query']??'-')?></td>
      <td><?=htmlspecialchars($d['qtype_name']??'-')?></td>
      <td><?=htmlspecialchars(is_array($d['answers']??null)?implode(' ', $d['answers']):($d['answers']??'-'))?></td>
    </tr>
  <?php endforeach; ?>
</tbody>
</table></div>
</div>
</div>

<!-- SSH -->
<div id="tab-ssh" class="tab-content">
  <div class="panel">
    <h2>#128272; Sessions SSH (ssh.log)</h2>
    <div class="table-wrap"><table>
      <thead><tr><th>Timestamp</th><th>Client</th><th>Serveur</th><th>Port</th><th>Auth OK</th><th>Client
SW</th><th>Serveur SW</th></tr></thead>
      <tbody>
        <?php foreach($ssh as $s):
          $ok=$s['auth_success']??false; ?>
          <tr style="<?=$ok?'':color:#e74c3c'?>">
            <td><?=htmlspecialchars($s['ts']??'-')?></td>
            <td><?=htmlspecialchars($s['id.orig_h']??'-')?></td>
            <td><?=htmlspecialchars($s['id.resp_h']??'-')?></td>
            <td><?=htmlspecialchars($s['id.resp_p']??'-')?></td>
            <td><?=$ok?'<span style="color:#2ecc71">✓ OUI</span>':<span style="color:#e74c3c">X
NON</span>'?></td>
            <td><?=htmlspecialchars($s['client']??'-')?></td>
            <td><?=htmlspecialchars($s['server']??'-')?></td>
          </tr>
        <?php endforeach; ?>
      </tbody>
    </table></div>
  </div>
</div>

<!-- HTTP -->
<div id="tab-http" class="tab-content">
  <div class="panel">
    <h2>#127760; Requêtes HTTP (http.log)</h2>
    <div class="table-wrap"><table>

```

```

<thead><tr><th>Timestamp</th><th>Client</th><th>Méthode</th><th>Hôte</th><th>URI</th><th>Code</th><
th>User-Agent</th></tr></thead>
  <tbody>
    <?php foreach($http as $h):
      $code=$h['status_code']??0;
      $color=$code>=400?'color:#e74c3c':($code>=300?'color:#f39c12': ''); ?>
    <tr style="<?=$color?>">
      <td><?=htmlspecialchars($h['ts']??'-')?></td>
      <td><?=htmlspecialchars($h['id.orig_h']??'-')?></td>
      <td><?=htmlspecialchars($h['method']??'-')?></td>
      <td><?=htmlspecialchars($h['host']??'-')?></td>
      <td style="max-width:200px;overflow:hidden;text-overflow:ellipsis"><?=htmlspecialchars($h['uri']??'-
')?></td>
      <td><?=htmlspecialchars($code?:"-")?></td>
      <td style="max-width:150px;overflow:hidden;text-overflow:ellipsis;font-
size:.8em"><?=htmlspecialchars($h['user_agent']??'-')?></td>
    </tr>
    <?php endforeach; ?>
  </tbody>
</table></div>
</div>
</div>

<!-- ALERTES -->
<div id="tab-notice" class="tab-content">
  <div class="panel">
    <h2>&#128680; Alertes de sécurité (notice.log)</h2>
    <?php if(empty($notices)): ?>
    <p style="color:#446;padding:20px;text-align:center">Aucune alerte pour le moment.</p>
    <?php else: ?>
    <div class="table-wrap"><table>
      <thead><tr><th>Timestamp</th><th>Type
d'alerte</th><th>Message</th><th>Source</th><th>Destination</th></tr></thead>
      <tbody>
        <?php foreach($notices as $n): ?>
        <tr class="notice-row">
          <td><?=htmlspecialchars($n['ts']??'-')?></td>
          <td><?=htmlspecialchars($n['note']??'-')?></td>
          <td><?=htmlspecialchars($n['msg']??'-')?></td>
          <td><?=htmlspecialchars($n['src']??'-')?></td>
          <td><?=htmlspecialchars($n['dst']??'-')?></td>
        </tr>
        <?php endforeach; ?>
      </tbody>
    </table></div>
    <?php endif; ?>
  </div>
</div>

<!-- WEIRD -->
<div id="tab-weird" class="tab-content">
  <div class="panel">
    <h2>&#9888;&#65039; Comportements anormaux (weird.log)</h2>
    <?php if(empty($weird)): ?>
    <p style="color:#446;padding:20px;text-align:center">Aucun comportement anormal détecté.</p>

```

```

<?php else: ?>
<div class="table-wrap"><table>
  <thead><tr><th>Timestamp</th><th>Src IP</th><th>Dst IP</th><th>Anomalie</th></tr></thead>
  <tbody>
    <?php foreach($weird as $w): ?>
    <tr class="warn-row">
      <td><?=htmlspecialchars($w['ts']??'-')?></td>
      <td><?=htmlspecialchars($w['id.orig_h']??'-')?></td>
      <td><?=htmlspecialchars($w['id.resp_h']??'-')?></td>
      <td><?=htmlspecialchars($w['name']??'-')?></td>
    </tr>
    <?php endforeach; ?>
  </tbody>
</table></div>
<?php endif; ?>
</div>
</div>

</div><!-- /container -->

<footer>Zeek 7.x — BTS SIO SISR — <?=date('d/m/Y à H:i:s')?></footer>

<script>
function showTab(name){
  document.querySelectorAll('.tab-content').forEach(t=>t.classList.remove('active'));
  document.querySelectorAll('.tab').forEach(t=>t.classList.remove('active'));
  document.getElementById('tab-'+name).classList.add('active');
  event.target.classList.add('active');
}
</script>
</body>
</html>

```

Etape 4 : création du virtualhost Apache

sudo nano /etc/apache2/sites-available/zeek.conf

Contenu à copier (**attention, modifier le sous-domaine en fonction de votre équipe !**) :

```

<VirtualHost *:80>
  ServerName zeek.equipeX.siosecure.fr
  DocumentRoot /var/www/zeek-dashboard

  <Directory /var/www/zeek-dashboard>
    Options -Indexes -FollowSymLinks
    AllowOverride None
    Require all granted
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/zeek_error.log
  CustomLog ${APACHE_LOG_DIR}/zeek_access.log combined
</VirtualHost>

```

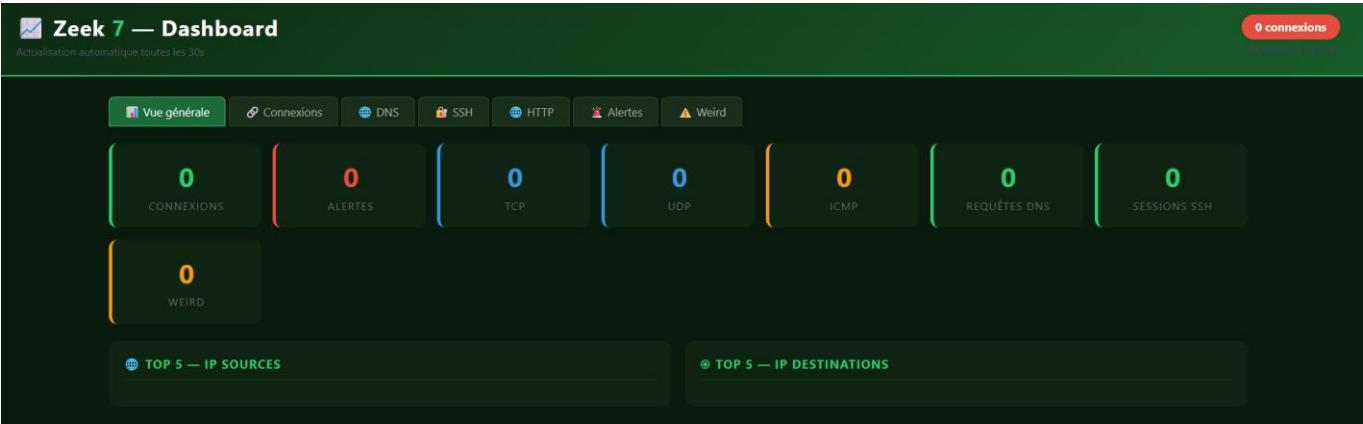
Activer le site et relancer Apache

```
sudo a2ensite zeek.conf
sudo systemctl reload apache2
```

Important :

Créez une règle dans votre pare-feu qui autorise le port 80 vers votre machine Zeek.

Lancez un navigateur et saisissez l’URL <https://zeek.equipeX.siosecur.fr> vous devriez obtenir ceci :



Testez Zeek depuis une autre machine de votre réseau local, en actualisant le tableau de bord.

RECAPITULATIF DES COMMANDES ZEEK DE REFERENCE

Commande	Description
zeekctl deploy	Déploie la config et démarre — à utiliser après toute modification
zeekctl start	Démarre Zeek
zeekctl stop	Arrête Zeek
zeekctl restart	Redémarre Zeek
zeekctl status	Affiche l'état des processus
zeekctl check	Valide la configuration sans démarrer
zeekctl cleanup	Supprime les fichiers temporaires (spool)
zeekctl crontab	Affiche les tâches cron configurées par Zeek

RECAPITULATIF DES REPERTOIRES ZEEK DE REFERENCE

Répertoire	Rôle
/opt/zeek/bin/	Binaires (zeek, zeekctl, zeek-cut)
/opt/zeek/etc/	Fichiers de configuration (node.cfg, networks.cfg, zeekctl.cfg)
/opt/zeek/share/zeek/	Scripts Zeek (base/, policy/, site/)
/opt/zeek/logs/current/	Logs en cours de génération
/opt/zeek/logs/	Archives des logs (classées par date)
/opt/zeek/spool/	Fichiers temporaires de travail

DEPANNAGE COURANT

Problème	Solution
zeekctl deploy échoue	Lancez zeekctl check pour voir l'erreur précise
Aucun log généré	Vérifiez que l'interface ens18 est en mode promisc et que l'offloading est désactivé
Logs vides après démarrage	Générez du trafic (ping, DNS) et attendez 10-15 secondes
Permission denied sur les logs	Vérifiez sudo usermod -aG zeek www-data et redémarrez Apache
Dashboard PHP n'affiche rien	Vérifiez que @load tuning/json-logs est dans local.zeek puis zeekctl deploy
Zeek consomme trop de CPU	Réduisez LogRotationInterval ou désactivez les scripts non nécessaires dans local.zeek