MONITORING IP & BAN

IDEO-LAB

JANUARY 2026

- 🔍 script de **monitoring auto + ban dynamique**
- 📊 dashboard Django / Grafana
- 🤖 fail2ban custom pour slow TLS
- 🧠 détection ASN / fingerprint réseau

1. **monitoring auto** des connexions TLS “idle / slow” via `ss`
2. **ban dynamique** ultra-rapide via **ipset** (timeout auto)
3. **fail2ban custom** (en complément, basé logs) avec action ipset
4. **enrichissement ASN / country / fingerprint** (cymru whois + GeoIP2 si tu veux)
5. **dashboard** : Django (table + admin) + Grafana via **Prometheus textfile collector**

INSTALLATION

- 0) Pré-requis : ipset + règle DROP
- 1) Script “monitoring auto + ban dynamique”
- 2) Exécuter automatiquement toutes les 30s (systemd timer)
- 3) Fail2ban “custom slow TLS” (complément logs) + action ipset
- 4) Détection ASN / fingerprint (propre)
- 5) Dashboard Grafana (Prometheus textfile collector)
- 6) Dashboard Django (simple + efficace)
- 7) Hardening Nginx (anti slow/idle)

INSTALLATION & EXECUTION

1) Script “monitoring auto + ban dynamique” (connexions 443 idle)

/usr/local/bin/ideolab_connmon.py

python

Copier le code

```python
#!/usr/bin/env python3
import json, os, re, sqlite3, subprocess, time
from datetime import datetime, timezone

# ====== CONFIG ======
LOCAL_HTTPS_PORT = 443

# seuils (ajuste selon ton trafic normal)
TH_CONN_IP = 25          # nb de connexions ESTABLISHED simultanées depu
TH_CONN_SUBNET24 = 220   # nb de connexions sur /24 (attaque distribuée
BAN_SECONDS = 3600       # 1h (ipset timeout)
MIN_IDLE_SCORE = 2       # score minimal pour bannir

# fichiers
LOG_DIR = "/var/log/ideolab"
EVENTS_JSONL = os.path.join(LOG_DIR, "connmon_events.jsonl")
METRICS_PROM = "/var/lib/node_exporter/textfile_collector/ideolab_connmo
DB_PATH = "/var/lib/ideolab/connm        cache.sqlite3"
```

2) Exécuter automatiquement toutes les 30s (systemd timer)

/etc/systemd/system/ideolab-connmon.service

ini Copier le code

```ini
[Unit]
Description=IDEOlab conn monitor + dynamic ban (ipset)
After=network.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/ideolab_connmon.py
```

/etc/systemd/system/ideolab-connmon.timer

ini Copier le code

```ini
[Unit]
Description=Run IDEOlab conn monitor frequently

[Timer]
OnBootSec=30
OnUnitActiveSec=30
AccuracySec=1s

[Install]
WantedBy=timers.target
```

bash

Copier le code

```
systemctl daemon-reload
systemctl enable --now ideolab-connmon.timer
systemctl list-timers | grep connmon
```

3) Fail2ban “custom slow TLS” (complément logs) + action ipset

3.1 Filter fail2ban (Nginx error log : timeouts / client closed / request line)

/etc/fail2ban/filter.d/nginx-ideolab-slowtls.conf

ini Copier le code

```ini
[Definition]
failregex = ^\s*\d{4}/\d{2}/\d{2}\s+\d{2}:\d{2}:\d{2}\s+\[error\]\s+\d+
ignoreregex =
```

3.2 Jail

/etc/fail2ban/jail.d/nginx-ideolab-slowtls.local

ini Copier le code

```ini
[nginx-ideolab-slowtls]
enabled = true
filter  = nginx-ideolab-slowtls
logpath = /var/log/nginx/error.log
findtime = 300
maxretry = 10
bantime  = 3600
action = ipset[name=ideolab_ban, port="443", protocol=tcp]
```

3.3 Action ipset

📄 **/etc/fail2ban/action.d/ipset.conf** (souvent déjà présent, sinon)

ini — Copier le code

```ini
[Definition]
actionstart = ipset create <name> hash:ip timeout 0 -exist
actionstop  =
actioncheck = ipset list <name> >/dev/null 2>&1
actionban   = ipset add <name> <ip> timeout <bantime> -exist
actionunban = ipset del <name> <ip> 2>/dev/null || true
```

Restart :

bash — Copier le code

```bash
systemctl restart fail2ban
fail2ban-client status nginx-ideolab-slowtls
```

> Important : **fail2ban** voit ce qui arrive jusqu'aux logs Nginx. Ton "slow TLS idle" peut ne jamais logguer → c'est pour ça que **connmon (ss)** est le "vrai" détecteur.

4) Détection ASN / fingerprint (propre)

Le script ci-dessus fait déjà :

- ASN + CC + AS Name via **whois.cymru.com**
- cache sqlite (7 jours)

Tu peux aussi ajouter un “fingerprint” simple :

- **/24**, **ASN**, **pays**, **UA** (si dispo via Nginx access logs), **JA3** (si tu actives un module/proxy qui l’expose)

👉 Pour du JA3 (TLS fingerprint), si tu veux le niveau “WAF”, on le fait via :

- Cloudflare / AWS WAF (le plus simple)
- ou Nginx + module/sidecar (plus lourd)

Dis-moi lequel tu utilises (Cloudflare ? ALB ? Nginx direct ?).

5) Dashboard Grafana (Prometheus textfile collector)

Le script écrit :

`/var/lib/node_exporter/textfile_collector/ideolab_connmon.prom`

Il te faut **node_exporter** avec textfile collector activé (souvent déjà).

Vérif :

bash Copier le code

```bash
cat /var/lib/node_exporter/textfile_collector/ideolab_connmon.prom
curl -s localhost:9100/metrics | grep ideolab_connmon
```

Dans Grafana :

- datasource Prometheus
- panels sur :
  - `ideolab_connmon_established_443`
  - `ideolab_connmon_suspect_ips`
  - `increase(ideolab_connmon_bans_total[5m])`

6) Dashboard Django (simple + efficace)

6.1 Modèles

python

Copier le code

```python
# app: security/models.py
from django.db import models

class ConnmonEvent(models.Model):
    ts = models.DateTimeField()
    type = models.CharField(max_length=64)
    ip = models.GenericIPAddressField(null=True, blank=True)
    subnet = models.CharField(max_length=32, null=True, blank=True)
    count = models.IntegerField(null=True, blank=True)
    score = models.IntegerField(null=True, blank=True)
    asn = models.CharField(max_length=32, null=True, blank=True)
    asname = models.CharField(max_length=255, null=True, blank=True)
    country = models.CharField(max_length=8, null=True, blank=True)
    reason = models.TextField(null=True, blank=True)
    raw = models.JSONField(null=True, blank=True)

    created_at = models.DateTimeField(auto_now_add=True)
```

6.2 Importer les events JSONL (cron Django)

python

Copier le code

```python
# security/management/commands/import_connmon_events.py
import json
from datetime import datetime
from django.core.management.base import BaseCommand
from django.utils.dateparse import parse_datetime
from security.models import ConnmonEvent

EVENTS_JSONL = "/var/log/ideolab/connmon_events.jsonl"

class Command(BaseCommand):
    help = "Import connmon JSONL events into Django"

    def handle(self, *args, **kwargs):
        imported = 0
        with open(EVENTS_JSONL, "r", encoding="utf-8") as f:
            for line in f:
                line = line.strip()
                if not line:
                    continue
                ev = json.loads(l
                ts = parse_datetime(ev.get("ts")) or datetime.utcnow()
```

7) Hardening Nginx (anti slow/idle) — très conseillé

Dans ton `http {}` :

nginx

Copier le code

```nginx
limit_conn_zone $binary_remote_addr zone=perip:10m;
limit_req_zone  $binary_remote_addr zone=reqip:10m rate=10r/s;

server {
  # limite connexions simultanées
  limit_conn perip 25;

  # limite rafales HTTP
  limit_req zone=reqip burst=40 nodelay;

  # timeouts (anti slowloris)
  client_header_timeout 10s;
  client_body_timeout   10s;
  keepalive_timeout     10s;
  send_timeout          10s;
  reset_timedout_connection on;
}
```

GEOIP

A) Bloquer “tout sauf FR/ES/UK/US/DE/IT” au firewall (GeoIP) ✅

1) Installer GeoIP xtables-addons

bash Copier le code

```
apt update
apt install -y xtables-addons-common geoip-bin
```

2) Construire la DB GeoIP (pays)

bash Copier le code

```
/usr/lib/xtables-addons/xt_geoip_dl
/usr/lib/xtables-addons/xt_geoip_build -D /usr/share/xt_geoip *.csv
```

3) Règles iptables (ALLOW pays, DROP le reste)

⚠️ Mets ça **AVANT** tes règles “accept established” si tu as un firewall custom.

bash Copier le code

```
# Autoriser en priorité les pays ok
iptables -I INPUT 1 -p tcp --dport 443 -m geoip --src-cc FR,ES,GB,US,DE,IT -j ACCEPT

# Bloquer les autres pays sur 443
iptables -I INPUT 2 -p tcp --dport 443 -j DROP
```

> Note : UK = GB en code pays.

B) Ban dynamique ultra rapide via ipset (script + seuil 50) ✅

1) ipset + règle DROP (persistante)

bash Copier le code

```
apt install -y ipset
ipset create ideolab_ban hash:ip timeout 0 -exist

# drop des IP bannies
iptables -I INPUT 1 -m set --match-set ideolab_ban src -j DROP
```

2) Mettre à jour le script (seuil = 50)

Dans ton `ideolab_connmon.py`, mets :

- `TH_CONN_IP = 50`
- `TH_CONN_SUBNET24 = 350` (je monte un peu car tu autorises 6 pays, on veut éviter faux positifs)

👉 Je te redonne un *diff* clair :

diff Copier le code

```
-TH_CONN_IP = 25
-TH_CONN_SUBNET24 = 220
+TH_CONN_IP = 50
+TH_CONN_SUBNET24 = 350
```

3) Activer le timer systemd (30s)

(tu l'as déjà dans mon message précédent)

bash

Copier le code

```
systemctl enable --now ideolab-connmon.timer
systemctl list-timers | grep connmon
```

C) Nginx hardening (anti slow TLS / keepalive abusif) ✅

Dans ton `server {}` HTTPS :

nginx

Copier le code

```nginx
# anti slowloris / slow headers
client_header_timeout 10s;
client_body_timeout   10s;
send_timeout          10s;

# keepalive plus court (tes captures montrent des connexions idle)
keepalive_timeout 10s;

# coupe les connexions qui traînent
reset_timedout_connection on;

# limites
limit_conn_zone $binary_remote_addr zone=perip:10m;
limit_req_zone  $binary_remote_addr zone=reqip:10m rate=10r/s;

limit_conn perip 50;                # ton seuil
limit_req zone=reqip burst=40 nodelay;
```

Reload :

bash

Copier le code

```bash
nginx -t && systemctl reload nginx
```

D) Fail2ban custom (log-based) + action ipset ✅

Tu gardes le jail que je t'ai donné, mais on ajuste un peu pour éviter de ban trop agressif :

`/etc/fail2ban/jail.d/nginx-ideolab-slowtls.local`

ini — Copier le code

```ini
[nginx-ideolab-slowtls]
enabled = true
filter  = nginx-ideolab-slowtls
logpath = /var/log/nginx/error.log
findtime = 300
maxretry = 15
bantime  = 3600
action = ipset[name=ideolab_ban, port="443", protocol=tcp]
```

Restart :

bash — Copier le code

```bash
systemctl restart fail2ban
fail2ban-client status nginx-ideolab-slowtls
```

E) ASN / Fingerprint (pratique et exploitable) ✅

Ce qu'on va sortir automatiquement (par IP bannie)

- IP
- /24
- ASN + AS Name + Country (via whois.cymru)
- nombre de connexions établies
- heuristic score

C'est déjà dans le script.

Tu peux consulter en live :

bash Copier le code

```
tail -n 50 /var/log/ideolab/connmon_events.jsonl | jq .
```

Et liste des bannis :

bash Copier le code

```
ipset list ideolab_ban | head -n 50
```

F) Dashboard Grafana (prêt) ✅

Le script écrit un fichier metrics :

`/var/lib/node_exporter/textfile_collector/ideolab_connmon.prom`

Vérif :

bash

Copier le code

```bash
cat /var/lib/node_exporter/textfile_collector/ideolab_connmon.prom
curl -s localhost:9100/metrics | grep ideolab_connmon
```

Dans Grafana, tu crées 3 panels :

- `ideolab_connmon_established_443`
- `ideolab_connmon_suspect_ips`
- `increase(ideolab_connmon_bans_total[5m])`

G) Dashboard Django (vite fait, efficace) ✅

Vu ton style IDEO-Lab, le plus utile :

- table “Derniers bans”
- top ASN (pie / bar)
- top /24
- bouton “ban ASN” / “ban /24” (si tu veux)

Je peux te générer :

- le `models.py`
- `admin.py` (avec filtres country/asn + search IP)
- une page template IDEO-Lab cards/modals (kcard + ig-modal)
- un management command d’import JSONL incrémental (avec offset)

⚡ Je te donne la recommandation “prod” (sans douleur)

1) GeoIP allowlist sur 443 (FR ES GB US DE IT)

2) ipset dynamic ban (connmon ss)

3) Nginx timeouts + limit_conn

4) fail2ban en complément
