PKTMON - Cheat Sheet Administrateur Windows

> PKTMON (Packet Monitor) est un outil natif Windows permettant de capturer et d'analyser le trafic réseau sans installer de logiciel tiers.

Vérifier la configuration actuelle

Afficher les filtres actifs :

```
pktmon filter list
```

Afficher les composants monitorés :

```
pktmon comp list
```

Afficher l'aide :

```
pktmon help
```

Nettoyer les filtres

Supprimer tous les filtres :

```
pktmon filter remove
```

Capturer un port spécifique

UDP

Capturer le port UDP 20500 :

```
pktmon filter add -p 20500
```

TCP

Capturer le port TCP 443 :

```
pktmon filter add -p 443
```

Lister les filtres :

```
pktmon filter list
```

Capture temps réel

Démarrer une capture en temps réel :

```
pktmon start --etw -m real-time
```

Arrêter la capture :

```
pktmon stop
```

Capture vers un fichier

Démarrer l'enregistrement :

```
pktmon start --capture --pkt-size 0
```

Arrêter :

```
pktmon stop
```

Fichier généré :

```
PktMon.etl
```

Export de capture

Conversion en texte

```
pktmon format PktMon.etl -o Capture.txt
```

Conversion Wireshark

```
pktmon etl2pcap PktMon.etl -o Capture.pcapng
```

Exemples de lecture

Trafic entrant

```
Direction Rx
192.168.1.10.50000 > 10.0.0.10.20500
```

Signification :

```
Client  : 192.168.1.10
Port    : 50000

Serveur : 10.0.0.10
Port    : 20500
```

Le serveur reçoit un paquet.

Trafic sortant

```
Direction Tx
10.0.0.10.20500 > 192.168.1.10.50000
```

Le serveur répond au client.

Validation d'un flux fonctionnel

```
Rx
192.168.1.10.50000 > 10.0.0.10.20500

Tx
10.0.0.10.20500 > 192.168.1.10.50000
```

Interprétation :

```
✓ Trafic reçu
✓ Application joignable
✓ Réponse envoyée
```

Filtrer par adresse IP

Capturer uniquement une IP :

```
pktmon filter add -i 192.168.1.10
```

Capturer plusieurs IP :

```
pktmon filter add -i 192.168.1.10
pktmon filter add -i 192.168.1.20
```

Capture pour analyse Wireshark

Méthode recommandée :

```
pktmon filter remove

pktmon filter add -p 20500

pktmon start --capture --pkt-size 0
```

Faire le test.

```
pktmon stop
```

Convertir :

```
pktmon etl2pcap PktMon.etl -o Capture.pcapng
```

Ouvrir directement dans Wireshark.

Rechercher une application liée à un port

UDP

```
Get-NetUDPEndpoint |
Where-Object LocalPort -eq 20500
```

TCP

```
Get-NetTCPConnection |
Where-Object LocalPort -eq 443
```

Associer un PID à un processus

```
Get-Process -Id 1234
```

Ou :

```
(Get-Process -Id 1234).Path
```

Résultat :

```
C:\Program Files\Application\App.exe
```

Vérifier rapidement le pare-feu

Règles contenant un port :

```
Get-NetFirewallRule |
Where-Object DisplayName -match "20500"
```

Ports ouverts :

```
Get-NetFirewallPortFilter |
Where-Object LocalPort -eq 20500
```

Workflow recommandé

1. Supprimer les filtres

```
pktmon filter remove
```

2. Ajouter le port

```
pktmon filter add -p 20500
```

3. Démarrer la capture

```
pktmon start --etw -m real-time
```

4. Réaliser le test applicatif

5. Vérifier

```
Rx uniquement
```

→ Requête reçue.

```
Rx + Tx
```

→ Communication bidirectionnelle.

```
Aucune trace
```

→ Le trafic n'arrive pas sur le serveur.

6. Arrêter

```
pktmon stop
```

Commandes essentielles à retenir

```
pktmon filter remove

pktmon filter add -p 20500

pktmon start --etw -m real-time

pktmon start --capture --pkt-size 0

pktmon stop

pktmon format PktMon.etl -o Capture.txt

pktmon etl2pcap PktMon.etl -o Capture.pcapng
```

Revision #3
Created 2026-10-23 07:11:04 UTC by Zayad
Updated 2026-10-23 07:17:12 UTC by Zayad