

Le système Blink de caméras et sonnettes connectées

“ Première identification réseau : **23 août 2026**, depuis le poste Windows `julie` (192.168.1.26, Ethernet). Script de relevé : `Jux-scripts/Reseau-LAN/scan_lan.py`

Ce qui a été cherché

Question de départ : *la sonnette Blink est-elle visible sur le réseau local ?* Réponse : **oui**, un unique appareil au constructeur Amazon répond sur le LAN domestique.

L'appareil Blink

Champ	Valeur
Adresse IP	<code>192.168.1.128</code> (bail DHCP Livebox)
Adresse MAC	<code>e8:4c:4a:62:8d:d9</code>
OUI <code>E8:4C:4A</code>	Amazon Technologies Inc.
Nom DHCP / DNS inverse	<code>device-19.home</code>
Ports TCP ouverts	443 uniquement
Latence	3-6 ms, 0 % de perte

Le port 443 n'accepte aucun client anonyme. Handshake TLS rejeté dans toutes les configurations testées :

Tentative	Résultat
TLS auto (jusqu'à 1.3), <code>ALL:@SECLEVEL=0</code>	<code>SSLV3_ALERT_HANDSHAKE_FAILURE</code>

Tentative	Résultat
TLS 1.2 forcé, ALL:@SECLEVEL=0	SSLV3_ALERT_HANDSHAKE_FAILURE
TLS 1.0 forcé	TLSV1_ALERT_PROTOCOL_VERSION

L'appareil impose donc son propre jeu de suites cryptographiques, et exige vraisemblablement un **certificat client** (authentification mutuelle). Comportement attendu et sain : le point de contact local ne parle qu'à l'application Blink et au cloud Amazon. **Il n'y a pas d'interface web, pas de flux RTSP, pas d'API locale exploitable** — inutile de chercher à brancher Frigate, Scripted ou Home Assistant dessus par cette voie.

Ce que le réseau ne dit pas

- Impossible de distinguer la sonnette du module Sync depuis le LAN.** Les deux portent une OUI Amazon et le même profil de port. Pour trancher : application Blink → *Paramètres de l'appareil* → *Informations*, qui affiche l'adresse MAC de chaque appareil. À reporter ici une fois relevé.
- Un seul appareil Amazon répond.** Si le foyer compte à la fois une sonnette **et** un module Sync, le second est en veille au moment du relevé. La Blink Video Doorbell sur batterie en mode standard **dort entre deux événements** et ne répond alors ni au ping ni à l'ARP ; elle n'est connectée en permanence qu'en câblage secteur (mode « enhanced »). *Point à confirmer par un relevé déclenché juste après avoir sonné à la porte.*

Inventaire complet du LAN — 192.168.1.0/24

Relevé du 2026-08-23 (balayage ping de .1 à .254, puis lecture de la table ARP) :

IP	MAC	Constructeur	Nom DNS	Ports TCP
.1	20:37:f0:a1:fa:dc	Arcadyan Corporation	lan.home	80, 443, 8883
.10	72:75:be:32:27:c4	[MAC aléatoire]	lenovo-tab-k11.home	aucun
.11	4e:16:10:cb:79:51	[MAC aléatoire]	redmi-note-11-pro.home	aucun
.17	00:11:32:9c:8e:c9	Synology Incorporated	—	21, 22, 80, 139, 443, 445
.23	2e:a5:80:20:23:b8	[MAC aléatoire]	s23-de-elio.home	aucun
.128	e8:4c:4a:62:8d:d9	Amazon Technologies Inc.	device-19.home	443

IP	MAC	Constructeur	Nom DNS	Ports TCP
.129	d4:3a:2e:d8:c5:52	SHENZHEN MTC CO LTD	—	80
.200	d4:3a:2e:d8:c5:52	SHENZHEN MTC CO LTD	—	80

- **.1** = la Livebox (Arcadyan en fabrique les modèles récents). Le port **8883** est du MQTT sur TLS — canal de télégestion Orange.
- **.17** = le NAS Synology `NASMAISON`, déjà documenté ailleurs (lecteur `A:` → `\NASMAISON\foxy`).
- **.129 et .200 portent la même MAC**. Un seul appareil Shenzhen MTC (fabricant de cartes mère de téléviseurs et de boîtiers connectés) avec **deux baux DHCP actifs** — il a changé d'adresse sans que l'ancien bail soit purgé. Sans gravité, mais à nettoyer dans l'interface Livebox si la table se remplit.

Points de méthode — à ne pas redécouvrir

1. `arp -a` **seul ne montre presque rien**. Le cache ARP ne contient que les hôtes contactés récemment — 3 entrées au premier relevé, contre 8 après balayage. **Toujours pinger le /24 avant de lire la table ARP**.
2. **La sortie de `arp -a` est traduite sous Windows** (`dynamique` et non `dynamic`). Ne jamais filtrer sur ce mot-clé : parser par expression régulière sur le couple IP + MAC. C'est ce que fait `scan_lan.py`.
3. **Une MAC dont le 2e bit du premier octet est posé est aléatoire**, pas un vrai constructeur : `72:`, `4e:`, `2e:`... Test : `int(octet1, 16) & 0x02`. C'est la randomisation MAC des OS modernes (Android, iOS, Windows) — inutile d'interroger une base OUI, elle ne renverra rien. Trois des huit appareils du LAN sont dans ce cas.
4. **Le DNS inverse de la Livebox est la source la plus riche**. `socket.gethostbyaddr()` renvoie les noms d'hôte du domaine `.home` (`redmi-note-11-pro.home`, `s23-de-elio.home`) — bien plus parlant que l'OUI pour les téléphones à MAC aléatoire. Ne pas s'arrêter à l'OUI.
5. `api.macvendors.com` **limite à ~1 requête/seconde** en accès gratuit. Le script temporise 1,5 s et met en cache dans `oui_cache.json` ; les relevés suivants peuvent tourner en `--hors-ligne`.
6. `curl.exe` **est absent du poste julie** — tout passe par Python (`urllib`) ou `Invoke-WebRequest`.
7. **Un TTL de 64 dans la réponse ping** indique une pile réseau Linux/BSD embarquée (Windows répondrait 128). Indice utile pour typer un objet connecté muet.

Commandes de vérification

Inventaire complet, avec sondage des ports :

```
py -3.12 D:\Syncthing\Jux_univers\Jux-scripts\Reseau-LAN\scan_lan.py --ports
```

Relevé rapide sans rien pinger ni interroger en ligne (lit le cache ARP et le cache OUI) :

```
py -3.12 scan_lan.py --sans-balayage --hors-ligne
```

Vérifier que l'appareil Blink est éveillé :

```
ping -n 4 192.168.1.128
```

Le script est en **stdlib Python pure** — aucune dépendance à installer, il tourne tel quel sur Windows, Ubuntu et le VPS. Il détecte le préfixe /24 tout seul et accepte `--reseau` pour le forcer.

Reste à faire

- ☐ Relever dans l'application Blink les adresses MAC de la sonnette **et** du module Sync, pour savoir lequel est `e8:4c:4a:62:8d:d9`
- ☐ Refaire un balayage **immédiatement après avoir sonné**, pour voir apparaître le second appareil Amazon s'il existe
- ☐ Identifier l'appareil Shenzhen MTC (.129/.200) — probablement un téléviseur ou un boîtier ; son port 80 est peut-être une interface de configuration
- ☐ Purger le bail DHCP en double dans l'interface Livebox
- ☐ Réserver une IP fixe pour la Blink dans le DHCP, si l'on veut la surveiller dans la durée

Le problème d'usage — caméras muettes après une période d'inactivité

Analysé le 30 août 2026, à la suite de l'ajout de la **caméra rampe**.

Le symptôme

Quand quelqu'un sonne au portail, **tout fonctionne**. En revanche, après un laps de temps sans utilisation, **ni la caméra du portail ni la caméra rampe ne répondent** lorsqu'on les interroge depuis l'application Blink.

Cette asymétrie est l'indice le plus précieux du dossier, et elle oriente tout le diagnostic. Deux chemins bien distincts sont en jeu :

- **Le chemin sortant** — un appui sur le bouton, ou une détection de mouvement, est un événement *local* : l'appareil se réveille de lui-même et pousse vers le cloud. **Ce chemin marche.**
- **Le chemin entrant** — la vue en direct demande de réveiller un appareil endormi *depuis l'extérieur*. **C'est celui qui échoue.**

Le défaut est donc dans la chaîne de réveil entrante, pas dans les caméras elles-mêmes ni dans leur liaison au cloud.

La chaîne de réveil, maillon par maillon

Une demande de vue en direct traverse tout ceci :

1. Application → cloud Amazon (Internet, réseau mobile ou Wi-Fi du téléphone)
2. Cloud → **module Sync**, via la connexion Internet de la maison — le module est le seul élément en permanence alimenté et associé au Wi-Fi
3. Module Sync → caméra, par une **liaison radio propriétaire basse fréquence** (de l'ordre de 900 MHz). C'est le seul canal encore actif quand la caméra dort, et il est **hors de portée du réseau IP**
4. La caméra allume sa radio Wi-Fi, **se ré-associe au point d'accès** et obtient un bail DHCP
5. Elle ouvre sa connexion vers le cloud, puis le flux vidéo remonte

Les maillons **3, 4 et 5** sont les points fragiles. Ils ne sont sollicités que lors d'un réveil entrant — ce qui explique exactement pourquoi la sonnerie fonctionne et pas la vue en direct.

Pourquoi un « ping léger et permanent » ne peut pas fonctionner

L'idée de maintenir les appareils éveillés par un ping périodique est séduisante, mais elle se heurte à un fait établi par nos propres relevés.

Sur trois balayages complets du /24, la table ARP n'a jamais contenu qu'une seule adresse Amazon. Les caméras ne sont pas « silencieuses » : elles ne sont **pas associées au Wi-Fi du tout**. Leur radio est éteinte, elles n'ont aucune adresse IP.

Deux conséquences, l'une technique et l'autre pratique :

- **On ne peut pas pinger ce qui n'a pas d'adresse.** Un paquet envoyé vers une IP inexistante n'atteint personne — aucun récepteur n'est allumé pour l'entendre. Le réveil ne passe pas par IP mais par la liaison radio du module Sync (maillon 3), à laquelle aucun outil réseau n'a accès.
- **Même si c'était possible, ce serait nuisible.** L'autonomie de deux ans annoncée repose entièrement sur le fait que la radio Wi-Fi reste éteinte : c'est le poste de consommation dominant. La maintenir active viderait les piles en quelques jours.

Il n'existe donc pas de solution réseau à ce problème. Ce qui suit agit sur les causes réelles.

Causes plausibles, par rendement décroissant

Le fait que les deux caméras échouent ensemble est significatif : deux appareils indépendants tombant en panne simultanément est peu probable. Cela désigne un **élément commun** — module Sync, Wi-Fi de la maison, ou Livebox — plutôt qu'un défaut propre à chaque caméra. À traiter dans cet ordre :

1. **Piles faibles.** Cause n°1 des réveils manqués, et la plus trompeuse. L'allumage de la radio Wi-Fi demande une *pointe* de courant ; des piles fatiguées s'effondrent sous cette charge alors que l'application les affiche encore « correctes », car elle mesure une tension à vide. Symptôme caractéristique : la caméra répond quand elle vient d'être sollicitée, mais plus après un long repos. **Exiger des piles lithium AA** (type Energizer Ultimate Lithium) — les alcalines tiennent mal les pointes de courant par temps froid, et les rechargeables NiMH à 1,2 V n'atteignent pas la tension attendue.
2. **Signal Wi-Fi insuffisant à l'emplacement.** Au réveil, la caméra doit se ré-associer *de zéro* : c'est bien plus exigeant que maintenir une liaison déjà établie. Un signal limite passe quand tout va bien et échoue au portail, en bout de portée. L'application Blink affiche **deux indicateurs distincts par caméra** — un pour le Wi-Fi, un pour la liaison au module Sync. **Les relever tous les deux, sur les deux caméras** : c'est la mesure la plus utile disponible, et elle est hors de ma portée depuis le réseau.
3. **Placement du module Sync.** Il doit être à la fois dans la portée radio des deux caméras et bien couvert en Wi-Fi. Le rapprocher des caméras améliore le maillon 3 ; le rapprocher de la Livebox améliore le maillon 2. C'est un arbitrage, souvent mal réglé par défaut.

4. **Bail DHCP expiré pendant le sommeil.** Après une longue absence, le bail a disparu de la Livebox : la caméra doit refaire une négociation complète, ce qui allonge le réveil et peut le faire échouer. **Correctif : réserver une IP fixe** pour le module Sync et pour chaque appareil Blink dans le DHCP de la Livebox.
5. **SSID unique 2,4 + 5 GHz avec orientation de bande.** Les appareils Blink sont **2,4 GHz uniquement**. Une Livebox qui diffuse un SSID unique et arbitre les bandes peut mal orienter une ré-association. Si le modèle le permet, **exposer un SSID 2,4 GHz distinct** et y fixer les appareils Blink.
6. **Programmation horaire du Wi-Fi.** Les Livebox savent couper le Wi-Fi la nuit. Vérifier qu'aucune plage d'extinction n'est active.
7. **Firmware.** Vérifier les mises à jour du module Sync et des caméras dans l'application.

Mesurer avant de conclure

Le module Sync est le seul élément observable depuis le réseau, et c'est justement le suspect n°1 en tant qu'élément commun. **S'il décroche du Wi-Fi pendant les périodes creuses, tout le reste s'effondre** — et cela se mesure.

Script : `Jux-scripts/Reseau-LAN/surveiller_blink.py`, stdlib seule.

```
py -3.12 D:\Syncthing\Jux_univers\Jux-scripts\Reseau-LAN\surveiller_blink.py --intervalle 30
py -3.12 surveiller_blink.py --resume
```

Il sonde toutes les 30 s, journalise en CSV dans `D:\Logs\Blink\blink_AAAAMMJJ.csv` et signale à la seconde près chaque apparition, perte, retour et disparition ARP. **Ce n'est pas un maintien en éveil** — la sonde ne réveille rien, elle constate.

⚠ **Les journaux sont volontairement écrits hors de l'arbre Syncthing**, dans `D:\Logs\Blink\` (surchargeable par la variable d'environnement `BLINK_LOGS`). Le script sonde toutes les 30 s : rangé dans `Jux-scripts/`, chaque sonde aurait déclenché une propagation vers les **7 appareils du maillage** — bruit de synchronisation permanent pour quelques octets de mesure. *Le script est synchronisé, ses journaux ne le sont pas.* Même raisonnement que pour les dossiers de la procédure Photos-Caesium.

Deux lectures possibles au bout de quelques jours :

- **Disponibilité du module Sync inférieure à 100 %** → le problème est le Wi-Fi de la maison ou le module lui-même. Agir sur les points 3 à 6.
- **Disponibilité à 100 % alors que la vue en direct échoue** → le module tient, la rupture est en aval : liaison radio ou réveil des caméras. Agir sur les points 1 et 2.

Ce test coupe le problème en deux et évite d'agir au hasard. Référence relevée le 30 août 2026 : **30 sondes, 0 % de perte, latence 4 à 26 ms** — le module Sync est irréprochable en journée. Reste à savoir ce qu'il fait la nuit et pendant les longues absences.

Un point à confirmer par l'application : **lequel des appareils est** `192.168.1.128` . Sa disponibilité parfaite et permanente désigne le module Sync, mais tant que la MAC `e8:4c:4a:62:8d:d9` n'a pas été relevée dans *Paramètres de l'appareil* → *Informations*, ce n'est qu'une déduction.

Relevé du 30 août 2026

Nouveau balayage après ajout de la caméra rampe. **Toujours un seul appareil Amazon**, en IPv4 comme en IPv6 (le voisinage IPv6 ne connaît que la Livebox et le NAS). Évolutions constatées, sans rapport avec Blink :

Évolution	Appareil
Apparu	<code>.12</code> — <code>redmi-13c.home</code> (MAC aléatoire)
Apparu	<code>.28</code> — <code>pc-elio.home</code> , Intel <code>08:b4:d2:6b:3b:63</code>
Disparu	<code>.23</code> — <code>s23-de-elio.home</code>
Inchangé	<code>.128</code> — <code>e8:4c:4a:62:8d:d9</code> , seul appareil Amazon

Le NAS résout désormais en `NASMAISON` . Le doublon de bail `.129` / `.200` persiste.

Ce que la sonde voit — et ses angles morts

Point clarifié le 30 août 2026, parce qu'il conditionne les décisions de placement.

Il n'y a pas un lien mais trois, et celui qui relie le module Sync aux caméras n'est pas du Wi-Fi.

Lien	Nature	Actif quand	Observable par la sonde ?
Livebox ↔ module Sync	Wi-Fi 2,4 GHz	en permanence	Oui , c'est la mesure principale
Module Sync ↔ caméras	radio propriétaire ~900 MHz	en permanence (réveil, commandes)	Non , hors du monde IP
Livebox ↔ caméras	Wi-Fi 2,4 GHz	uniquement caméra réveillée	Oui, par intermittence

Conséquence pratique, contre-intuitive : le Wi-Fi des caméras va vers la **Livebox**, pas vers le module Sync. Rapprocher le module Sync des caméras améliore la liaison radio de réveil, mais **ne change rien** à la portée Wi-Fi des caméras. Si le portail est loin de la Livebox, déplacer le module Sync ne corrigera pas ce problème-là. C'est précisément pour cela que l'application affiche **deux indicateurs séparés** par caméra.

On n'est donc pas aveugle aux caméras. La liaison radio de réveil est inobservable, mais *son résultat* l'est : une caméra qui se réveille s'associe à la Livebox et prend une adresse IP, donc elle apparaît. Si elle n'apparaît **jamais**, c'est que le réveil ou l'association Wi-Fi échoue — ce qui est exactement la question posée.

Angles morts assumés

- **La liaison module Sync ↔ cloud Amazon n'est pas testée.** La sonde mesure le trajet PC → Livebox → module Sync, entièrement local. Le module peut être parfaitement joignable sur le réseau alors que sa connexion vers Internet est rompue — cas qui produirait exactement le symptôme observé sans que le journal montre quoi que ce soit.
- **La liaison radio ~900 MHz est définitivement hors de portée** de tout outil réseau. Seule l'application Blink en donne une mesure.
- **Les fenêtres de réveil sont brèves.** Une caméra peut se réveiller et se rendormir entre deux sondes. Réduire `--intervalle` pendant un test ciblé augmente les chances de la saisir.

Trois correctifs du 30 août 2026

1. **Balayage périodique du /24 — sans lui, les caméras seraient restées invisibles.**
La première version lisait la table ARP sans jamais l'alimenter : or `arp -a` ne montre que les hôtes déjà contactés. Une caméra qui se réveille n'y serait entrée que par hasard, et la détection d'apparition n'aurait probablement *jamais* fonctionné. Le script balaie désormais le /24 toutes les 5 min (option `--balayage`). *Défaut de conception, pas de réglage : la fonctionnalité annoncée était inopérante.*
2. **Le script ne meurt plus sur une erreur d'écriture.** Une exception sur le journal faisait tomber toute la surveillance — une sonde perdue est sans conséquence, un script mort laisse un trou de plusieurs jours et fait croire à une panne réseau.
3. **Journal de repli `blink_AAAAMMJJ_b.csv`. Ouvrir le CSV dans LibreOffice Calc pose un verrou exclusif** qui bloque toute écriture : consulter ses propres mesures suffisait à les interrompre. Constaté en conditions réelles le 30 août — le coupable a été identifié par le *Restart Manager* de Windows, seul moyen sans outil tiers :

```
Add-Type -TypeDefinition $sig -Language CSharp      # binding rstrtmgr.dll
[RM]::Who('D:\Logs\Blink\blink_20260830.csv')      # -> 10336 : LibreOffice
```

`--resume` agrège tous les fichiers `blink_*`, les deux journaux sont donc relus ensemble. À retenir : **ni `arp -a`, ni la liste des processus Python ne désignent le verrou** — aucun processus Python ne subsistait, et le fichier restait bloqué.

Mise en service de la surveillance — méthode retenue le 30 août 2026

Voie retenue : le dossier Démarrage de Windows, sans élévation. Le planificateur de tâches a été écarté après deux échecs successifs, documentés ci-dessous.

Lanceur : `Jux-scripts/Reseau-LAN/Surveillance Blink.cmd` (fins de ligne CRLF, appelle `pythonw.exe` — donc aucune fenêtre visible).

```
& "D:\Syncthing\Jux_univers\Jux-scripts\Reseau-LAN\Surveillance Blink.cmd"
Copy-Item "D:\Syncthing\Jux_univers\Jux-scripts\Reseau-LAN\Surveillance Blink.cmd"
"$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup\"
```

État au 30 août 2026, 08:54 — sonde active (PID 3724), journal alimenté toutes les 30 s, module Sync joignable à 4-6 ms.

Pièges de mise en service — à ne pas refaire

1. **La syntaxe** `schtasks /tr "\"chemin\" \"script\""` **est du `cmd.exe`, pas du PowerShell.** En PowerShell, `\` n'est pas une séquence d'échappement : la commande échoue sur « *Argument ou option non valide* » avec les antislashes restés collés au chemin. Soit utiliser le jeton d'arrêt d'analyse `--%`, soit — mieux — se passer des guillemets internes, aucun des deux chemins en jeu ne comportant d'espace.
2. **Register-ScheduledTask exige l'élévation** pour écrire dans le dossier racine du planificateur : `Accès refusé, HRESULT 0x80070005`. Ce n'est pas contournable par la syntaxe. **Le dossier Démarrage, lui, ne demande aucun droit administrateur** — c'est ce qui a tranché.
3. **Si l'on passe malgré tout par le planificateur en console élevée**, préciser `-User "julie" -RunLevel Limited` : sans cela la tâche s'exécuterait avec les privilèges administrateur, ce qu'une simple sonde réseau n'a aucune raison d'obtenir. Ajouter `-ExecutionTimeLimit ([TimeSpan]::Zero)`, faute de quoi la tâche est **tuée au bout de 72 h** (limite par défaut) — fatal pour une surveillance censée durer.
4. **La copie vers le dossier Démarrage doit être tapée par Julien, jamais lancée depuis Claude Code.** `%APPDATA%` est redirigé par le conteneur MSIX : la copie atterrirait dans `...\Packages\Claude_pzs8srxjxfjfc\LocalCache\Roaming\...`, invisible de Windows, **sans le moindre message d'erreur**. Piège général documenté page 282.
5. **Méthode de vérification, elle, exécutable depuis Claude Code** — la liste des processus et les chemins réels ne sont pas virtualisés en lecture :

```
Get-CimInstance Win32_Process -Filter "Name='pythonw.exe'" | Select-Object ProcessId,
CommandLine
```

Pour lever tout doute sur une écriture dans `%APPDATA%`, comparer le chemin réel et son équivalent `...\Packages\Claude_pzs8srxjxfjfc\LocalCache\Roaming\...` : **le fichier présent uniquement dans le chemin réel prouve une écriture authentique** ; présent dans les deux avec les mêmes taille et horodatage, c'est le même fichier redirigé.

6. **Coller un message d'erreur PowerShell dans PowerShell produit une cascade d'erreurs trompeuses.** Les lignes de rappel commencent par `+`, que l'interpréteur lit comme un opérateur unaire : « *Expression manquante après l'opérateur unaire +* ».

Symptôme sans rapport avec le problème d'origine — ne pas se lancer dans son diagnostic.

Contrepartie de la voie retenue : une fenêtre de console apparaît une fraction de seconde à l'ouverture de session, et le script n'est pas relancé automatiquement s'il s'interrompt. Le planificateur offrirait les deux, au prix d'une élévation.

À faire, dans l'ordre

- Relever les **deux indicateurs de signal** (Wi-Fi et module Sync) des deux caméras dans l'application — mesure la plus rentable, à faire en premier
- Remplacer les piles par des **lithium AA** si elles ne le sont pas déjà, même si l'application les dit correctes
- Laisser tourner `surveiller_blink.py` plusieurs jours, puis lire `--resume`
- Réserver des **IP fixes** pour les appareils Blink dans le DHCP de la Livebox
- Vérifier qu'aucune **programmation horaire du Wi-Fi** n'est active sur la Livebox
- Relever la MAC de chaque appareil Blink dans l'application, pour lever l'ambiguïté sur `.128`

Revision #7

Created 2026-08-23 08:57:06 UTC by Julien

Updated 2026-08-30 07:09:50 UTC by Julien