تقرير مقارنة شامل
مشاريع مراقبة وإدارة الشبكات
عبر بوتات Telegram

تحليل مفصّل لثمانية مشاريع مفتوحة المصدر مع جدول مقارنة نهائي
وتوصيات لشبكات Starlink

يونيو 2026 — إعداد: Z.ai

فهرس المحتويات

1 المشروع الأول: TLGRM Script (Mikrotik-Bot-Telegram)

2 المشروع الثاني: luci-app-telemt

3 المشروع الثالث: Dynamic-IP-Tracker

4 المشروع الرابع: RouterOS Scripts Collection

5 المشروع الخامس: luci-app-trafficctl

6 المشروع السادس: MickroTikBot (Java)

7 المشروع السابع: DroidNet Sentinel

8 المشروع الثامن: NetAlertX

9 جدول المقارنة النهائي

10 الأسئلة الختامية والتوصيات

1. TLGRM Script (Mikrotik-Bot-Telegram)

https://github.com/pantigon/Mikrotik-Bot-Telegram

القسم الأول: وصف عام

TLGRM Script هو سكربت مكتوب بلغة RouterOS الأصلية يعمل مباشرةً داخل راوتر MikroTik بدون الحاجة لأي جهاز خارجي أو خادم إضافي. يقدم السكربت قدرتين أساسيتين: الأولى هي إعادة توجيه سجلات النظام (Logs) المهمة من الراوتر إلى مجموعة Telegram في الوقت الفعلي، والثانية هي تنفيذ أوامر عن بُعد على الراوتر عبر إرسال رسائل Telegram. السكربت مدعوم على RouterOS 6.49+ و7.12+، ويستهدف بشكل رئيسي المستخدمين المنزليين والشركات الصغيرة الذين يمتلكون راوتر MikroTik ويرغبون في مراقبة شبكتهم والتحكم بها من خلال الهاتف المحمول.

القسم الثاني: الإمكانيات والميزات

- **إعادة توجيه السجلات:** يراقب سجلات RouterOS (تحذيرات، أخطاء، حرائق، DHCP، اللاسلكي، الجدار الناري) ويرسلها فوراً إلى Telegram.
- **دعم الأحرف السيريلية:** تحويل كامل من CP1251 إلى UTF-8 للرسائل الروسية.
- **الصمود أمام انقطاع الإنترنت:** يقوم بتخزين الرسائل المعلقة وإرسالها عند عودة الاتصال.
- **دعم عدة راوترات:** يمكن لعدة أجهزة MikroTik مشاركة نفس مجموعة Telegram، وكل جهاز يستجيب فقط للأوامر الموجهة إليه بالاسم أو عبر `forall`.
- **إثراء عناوين MAC:** عند ظهور MAC في السجلات، يربطه تلقائياً باسم الجهاز وعنوان IP من DHCP.
- **إشعارات العملاء اللاسلكيين:** إرسال تنبيهات عند اتصال أو انفصال أجهزة WiFi، خصوصاً الأجهزة غير المعروفة.

أزرار وأوامر Telegram:

لا يوجد أوامر ثابتة مسبقاً. يستخدم نظام أوامر مخصص عبر BotFather بصيغة /<معرف_الجهاز>_<الأمر>. أمثلة:

- `/mikrotik system reboot` — إعادة تشغيل الراوتر
- `/forall log warning [/system resource get uptime]` — طلب وقت التشغيل من كل الراوترات
- `/mikrotik1_wol` — تنفيذ سكربت Wake-on-LAN

ثلاثة أنواع تنفيذ: دوال عامة (launchFnc)، سكربتات (launchScr)، أوامر خام (launchCmd). **لا توجد أزرار تفاعلية (Inline Buttons).**

القسم الثالث: الصلاحيات المطلوبة

- **على الراوتر:** صلاحيات كاملة (Full Admin) عند تفعيل launchCmd، بالإضافة إلى صلاحية /tool fetch للاتصال بـ Telegram API، وقراءة السجلات وجداول DHCP واللاسلكي.
- **على الجهاز المضيف:** لا يحتاج أي جهاز خارجي — السكربت يعمل بالكامل داخل الراوتر.

القسم الرابع: متطلبات التثبيت

- **الأجهزة:** راوتر MikroTik يعمل بـ RouterOS 6.49+ أو 7.12+ فقط.
- **البرامج:** لا يحتاج Python أو Docker أو أي بيئة خارجية — سكربت .rsc واحد فقط.
- **خطوات التثبيت:** إنشاء بوت Telegram عبر BotFather ← نسخ السكربت إلى WinBox ← ضبط المتغيرات (botID, myChatID, broadCast) ← إنشاء جدول زمني (Scheduler) لتشغيله كل دقيقة ← تسجيل الأوامر في BotFather.

القسم الخامس: التحكم والإجراءات

الإجراء	النتيجة	التفاصيل
حظر الأجهزة	نعم	عبر launchCmd بإضافة قواعد جدار ناري مثل `ip firewall filter add chain=forward src-mac=XX action=drop`
تحديد السرعة	نعم	عبر launchCmd بإنشاء Simple Queue مثل `queue simple add name=device target=IP max-limit=1M/1M`
إشعار أجهزة جديدة	محدود	يكشف أجهزة DHCP واللاسلكي الجديدة لكن بدورية المجدول (حوالي دقيقة)، وليس لحظياً
مراقبة الباندويث	لا	لا يوجد مراقبة مدمجة للاستهلاك لكل جهاز (يمكن الاستعلام يدوياً عبر أوامر)
إعادة تشغيل الراوتر	نعم	عبر `/mikrotik system reboot` — مثال مذكور في التوثيق

القسم السادس: التوافق مع Starlink

لا يعمل مباشرة مع راوتر Starlink الأصلي لأن السكربت مخصص لـ MikroTik RouterOS فقط. المطلوب: تفعيل وضع Bypass Mode على Starlink وربط راوتر MikroTik كراوتر رئيسي خلف طبق Starlink، ثم تشغيل TLGRM على MikroTik.

القسم السابع: التقييم

المعيار	التقييم
سهولة الإعداد	متوسط
التقييم العام	**6/10**

تحذير أمني: تفعيل launchCmd يسمح بتنفيذ أي أمر من Telegram بدون مصادقة إضافية. أي عضو في المجموعة يمكنه إعادة تشغيل الراوتر أو تغيير قواعد الجدار الناري.

2. luci-app-telemt (مع Telegram Bot)

https://github.com/Medvedolog/luci-app-telemt

القسم الأول: وصف عام

luci-app-telemt هو تطبيق LuCI لـ OpenWrt مخصص لإدارة بروكسي Telemt MTProto — وهو خادم بروكسي عالي الأداء مكتوب بلغة Rust يتيح لمستخدمي Telegram تجاوز الرقابة والقيود الشبكية. يوفر النظام لوحة تحكم ويب كاملة، وإدارة دورة حياة الخدمة، وبوت Telegram مدمج للإدارة عن بُعد من الهاتف. **هذا المشروع ليس أداة إدارة شبكة عامة**، بل هو مدير بروكسي MTProto متخصص يتحكم في من يمكنه استخدام البروكسي وكمية البيانات المسموح بها.

القسم الثاني: الإمكانيات والميزات

- **استضافة بروكسي MTProto:** تشغيل خادم Telemt Rust كبروكسي كامل على الراوتر.
- **إعادة تحميل بدون توقف:** إضافة/إزالة مستخدمين وتغيير حصص البيانات بدون قطع الاتصالات النشطة.
- **إدارة المستخدمين:** إنشاء، تجميد، حذف المستخدمين مع حصص بيانات فردية وتواريخ انتهاء صلاحية.
- **تخفي DPI:** ميزة Self-Stealth لإخفاء حركة البروكسي كحركة TLS عادية.
- **توليد FakeTLS ورموز QR:** إنشاء روابط بروكسي مقنّعة ورموز QR لمشاركتها.
- **تتالي SOCKS5/Shadowsocks:** توجيه حركة البروكسي عبر خوادم وسيطة.
- **بوت Telegram:** يعمل كخدمة procd مستقلة، مكتوب بـ POSIX sh فقط (3-5 ميجابايت RAM)، مع فشل ثلاثي الطبقات للاتصال.

أزرار وأوامر Telegram:

- `/start` و`/menu` — القائمة الرئيسية مع لوحة مفاتيح تفاعلية
- `/status` — حالة النظام والاتصال والنسخة
- `/users` — قائمة المستخدمين مع أزرار إدارة
- `/upstreams` — إدارة البروكسيات الوسيطة
- **أزرار تفاعلية (Inline Buttons):** تجميد/إلغاء تجميد مستخدم، تعديل حصة البيانات، حذف، تعطيل جميع Upstreams، تبديل Middle Proxy، تحديث، رجوع.

القسم الثالث: الصلاحيات المطلوبة

- **على الراوتر:** صلاحيات Root/Full Access لتثبيت الحزم، كتابة إعدادات UCI، إدارة خدمات procd، وحقن قواعد الجدار الناري.
- **على الجهاز المضيف:** تطبيق Telegram فقط — لا صلاحيات خاصة مطلوبة.

القسم الرابع: متطلبات التثبيت

- **الأجهزة:** أي راوتر OpenWrt (MIPS, ARM, x86, aarch64) بحد أدنى 64 ميجابايت RAM وIP عام.
- **البرامج:** حزم luCI-base, ca-bundle, qrencode, curl, jsonfilter, والثنائي telemt_wrt منفصلاً.
- **خطوات التثبيت:** تثبيت محرك telemt_wrt الأساسي ← تثبيت حزمة luci-app-telemt (opkg/apk) ← تثبيت حزمة telemt-bot اختيارياً ← ضبط bot_token وbot_chat_id في UCI ← تشغيل الخدمة.

القسم الخامس: التحكم والإجراءات

الإجراء	النتيجة	التفاصيل
حظر الأجهزة	لا	يحظر مستخدمي البروكسي فقط وليس أجهزة الشبكة العامة
تحديد السرعة	محدود	حصص بيانات لكل مستخدم بروكسي فقط، وليس تحديد سرعة لكل جهاز شبكة
إشعار أجهزة جديدة	لا	لا يكتشف أجهزة الشبكة — يراقب مستخدمي البروكسي فقط
مراقبة الباندويث	محدود	مراقبة حركة البروكسي لكل مستخدم فقط
إعادة تشغيل الراوتر	لا	يمكنه إعادة تشغيل خدمة telemt فقط وليس الراوتر

القسم السادس: التوافق مع Starlink

غير متوافق بشكل مباشر لأنه يحتاج IP عام لاستقبال اتصالات Telegram الواردة. Starlink يستخدم CGNAT في معظم المناطق، مما يمنع الاتصالات الواردة. يمكن التغلب عبر SOCKS5/Shadowsocks cascade أو PROXY protocol عبر VPS خارجي.

القسم السابع: التقييم

المعيار	التقييم
سهولة الإعداد	متوسط

التقييم العام	**7/10** (متميز كمدير بروكسي MTProto لكنه ليس أداة إدارة شبكة عامة)

3. Dynamic-IP-Tracker

https://github.com/v019f1n9er/Dynamic-IP-Tracker

القسم الأول: وصف عام

Dynamic-IP-Tracker هو سكربت خفيف جداً لـ MikroTik RouterOS يقوم بمهمتين محددتين فقط: مراقبة تغيّرات عنوان IP العام للراوتر وإرسال إشعار عبر Telegram عند التغيير، بالإضافة إلى تحديث قواعد NAT (dst-nat) تلقائياً لتعكس عنوان IP الجديد. مصمم خصيصاً للمستخدمين الذين يحصلون على IP ديناميكي من مزود الخدمة ويحتاجون إلى توجيه المنافذ (Port Forwarding) ليعمل بشكل موثوق رغم تغير الـ IP.

القسم الثاني: الإمكانيات والميزات

- **مراقبة IP العام:** فحص دوري لعنوان IP على واجهة WAN.
- **إشعار Telegram:** إرسال رسالة "New public IP: x.x.x.x" عند التغيير.
- **تحديث NAT تلقائي:** تعديل حقل dst-address في قاعدة NAT المحددة بتعليق (comment).
- **تنفيذ مجدول:** يعمل عبر Scheduler كل 5 دقائق (قابل للتعديل).

أزرار وأوامر Telegram:

لا توجد أي أوامر أو أزرار تفاعلية. النظام أحادي الاتجاه (إشعارات فقط) بدون استقبال أوامر.

القسم الثالث: الصلاحيات المطلوبة

- **على الراوتر:** صلاحيات read, write, test, sniff, policy في Scheduler.
- **على الجهاز المضيف:** لا يحتاج أي جهاز خارجي.

القسم الرابع: متطلبات التثبيت

- **الأجهزة:** أي راوتر MikroTik يعمل بـ RouterOS 6.x أو x.7.
- **البرامج:** RouterOS فقط — سكربت ~25 سطر.
- **خطوات التثبيت:** إنشاء بوت Telegram → تعديل المتغيرات (TOKEN, CHATID, الواجهة، تعليق NAT) → نسخ السكربت إلى الراوتر → إنشاء Scheduler.

القسم الخامس: التحكم والإجراءات

الإجراء	النتيجة
حظر الأجهزة	✗
تحديد السرعة	✗
إشعار أجهزة جديدة	✗
مراقبة الباندويث	✗
إعادة تشغيل الراوتر	✗

القسم السادس: التوافق مع Starlink

متوافق جزئياً: يعمل فقط إذا كان Starlink يوفر IP عام حقيقي. معظم مستخدمي Starlink يحصلون على IP خاص (CGNAT) مما يجعل تحديثات NAT عديمة الفائدة. يحتاج تعديل السكربت ليقرأ IP العام من خدمة خارجية مثل ifconfig.me.

القسم السابع: التقييم

المعيار	التقييم
سهولة الإعداد	سهل
التقييم العام	**3/10** (محدود جداً في النطاق)

4. RouterOS Scripts Collection

https://github.com/henriquesebastiao/routeros-scripts

القسم الأول: وصف عام

مجموعة من سكربتات الأتمتة المكتوبة بلغة RouterOS لتنفيذ مهام إدارية شائعة على أجهزة MikroTik. تشمل تحديث النظام التلقائي، مراقبة الواجهات، التحكم في عرض نطاق DHCP، إرسال تنبيهات عبر Telegram والبريد الإلكتروني، وغيرها. جميع السكربتات تعمل مباشرة على الراوتر بدون أي متطلبات خارجية. المشروع مرخص تحت MIT ومستهدف لمسؤولي الشبكات في البيئات الصغيرة والمتوسطة.

القسم الثاني: الإمكانيات والميزات

- **تحديث تلقائي لـ RouterOS:** فحص التحديثات، إشعار بالبريد، تثبيت وإعادة تشغيل تلقائي، ترقية Firmware.
- **رسائل Telegram:** إرسال حالة الجهاز (حرارة، جهد، تحميل CPU، وقت التشغيل)، إشعارات DHCP lease، تنبيهات محطة لاسلكية.
- **مراقبة سرعة الربط:** تحذير عبر Telegram عند انخفاض سرعة منفذ Ethernet عن الحد المحدد.
- **تحكم بعرض نطاق DHCP:** إنشاء/إزالة Simple Queue تلقائياً لكل عميل DHCP مع حدود قابلة للتخصيص.
- **نسخ احتياطي بالبريد:** إنشاء ملف backup. وإرساله كمرفق بريد إلكتروني.
- **شهادة SSL ذاتية:** إنشاء شهادة CA وشهادة خادم لـ WebFig على HTTPS.

أزرار وأوامر Telegram:

لا توجد أوامر أو أزرار تفاعلية. جميع إشعارات Telegram أحادية الاتجاه (push notifications فقط).

القسم الثالث: الصلاحيات المطلوبة

- **على الراوتر:** صلاحيات read, write, policy, test, reboot حسب السكربت.
- **على الجهاز المضيف:** لا يحتاج أي جهاز خارجي — كل شيء يعمل على الراوتر.

القسم الرابع: متطلبات التثبيت

- **الأجهزة:** أي جهاز MikroTik RouterBOARD.

- **البرامج:** RouterOS فقط، وWinBox/WebFig للإدارة، وإعداد خادم SMTP للبريد.
- **خطوات التثبيت:** تحميل السكربتات ← لصقها في System ← Scripts ← تعديل العناصر النائبة (bot token, chat ID, emails) ← ربط سكربت DHCP بـ Lease Script ← إنشاء Scheduler.

القسم الخامس: التحكم والإجراءات

الإجراء	النتيجة	التفاصيل
حظر الأجهزة	لا	لا يوجد سكربت لحظر الأجهزة
تحديد السرعة	نعم	سكربت dhcp-server-queue-script ينشئ Simple Queue لكل عميل DHCP تلقائياً (افتراضي 2M/5M)
إشعار أجهزة جديدة	نعم	سكربت dhcp-server-monitor يرسل إشعار Telegram عند كل منح/تحرير lease جديد
مراقبة الباندويث	محدود	الـ Queues توفر حدوداً لكن بدون تقارير استهلاك مدمجة
إعادة تشغيل الراوتر	غير مباشر	إعادة التشغيل ضمن عملية التحديث التلقائي فقط

القسم السادس: التوافق مع Starlink

يعمل بشكل طبيعي مع أي اتصال WAN بما فيها Starlink. السكربتات تعمل على جانب LAN وهي مستقلة عن مزود الخدمة. لا يوجد تكامل خاص بـ Starlink.

القسم السابع: التقييم

المعيار	التقييم
سهولة الإعداد	متوسط
التقييم العام	**5.5/10**

5. luci-app-trafficctl

https://forum.openwrt.org/t/250531

https://github.com/YusDyr/luci-app-trafficctl

القسم الأول: وصف عام

luci-app-trafficctl هو تطبيق LuCI متكامل لمراقبة والتحكم في حركة مرور كل جهاز على حدة على راوترات OpenWrt. وُلد المشروع من رغبة المؤلف في وجود طريقة موثوقة بضغطة واحدة لقطع الإنترنت عن أطفاله، وإحباطه من صعوبة القيام بذلك بشكل صحيح على OpenWrt. يوفر مراقبة فورية لكل جهاز مع رسوم بيانية حية، وحظر إنترنت/WiFi فوري، وتحديد سرعة (Rate Limit/Traffic Shaping)، وبوت Telegram تفاعلي كامل مع أزرار مضمّنة للتحكم عن بُعد. يدعم OpenWrt 21.02 حتى 25.12 والنسخ التطويرية.

القسم الثاني: الإمكانيات والميزات

- **مراقبة فورية لكل جهاز:** اتصالات نشطة، حالة TCP، عناوين IP الوجهة، سرعة الباندويث الحية.
- **رسوم بيانية حية (Sparklines):** رسوم متحركة تتحدث كل ثانيتين لكل جهاز مع نافذة منبثقة تفاعلية.
- **حظر الإنترنت:** قواعد Layer 3 مع إنهاء conntrack فوري — الجهاز يُقطع فوراً.
- **حظر WiFi:** عبر hostapd_cli — فصل الجهاز المستهدف فقط بدون إعادة تحميل WiFi.
- **محدد السرعة (Policer):** nftables/iptables limit rate — إسقاط الحزم فوق الحد فوراً.
- **مشكّل الحركة (Shaper):** tc/HTB + fq_codel لتنعيم الحركة بدلاً من إسقاطها.
- **اكتشاف أجهزة جديدة:** عبر ARP + DHCP + WiFi stations مع إشعار فوري عبر DHCP hotplug.
- **بوت Telegram تفاعلي:** أوامر وأزرار مضمّنة كاملة للتحكم عن بُعد.
- **استمرارية عبر إعادة التشغيل:** القواعد تنجو من إعادة التشغيل عبر hotplug restore.
- **قوالب إشعار مخصصة:** 17 متغيراً لقوالب إشعار الأجهزة الجديدة.

أزرار وأوامر Telegram:

- `start/` و`help/` — المساعدة والبداية
- `devices/` — قائمة الأجهزة النشطة مع أزرار لكل جهاز
- `status/` — ملخص الأجهزة المحظورة/المحدودة

أزرار تفاعلية (Inline Buttons) لكل جهاز:

- حظر/إلغاء حظر الإنترنت — حظر/إلغاء حظر WiFi
- تحديد سرعة: 1M, 2M, 5M, 10M, 25M, 50M, 100M kbit/s
- تشكيل حركة (Shaper): 1M, 2M, 5M, 10M, 25M, 50M, 100M kbit/s
- إزالة التحديد/التشكيل — رجوع

القسم الثالث: الصلاحيات المطلوبة

- **على الراوتر:** صلاحيات Root كاملة (nftables/iptables, tc/HTB, hostapd_cli, conntrack, rpcd, ACL).
- **على الجهاز المضيف:** متصفح ويب أو تطبيق Telegram فقط.

القسم الرابع: متطلبات التثبيت

- **الأجهزة:** أي راوتر OpenWrt (MIPS/ARM/x86) بحد أدنى 64 ميجابايت RAM.
- **البرامج:** حزمة IPK/APK واحدة (luci-app-trafficctl)، مع تبعيات اختيارية: tc-full, iw-full, curl لكل الميزات.
- **خطوات التثبيت:** `opkg install luci-app-trafficctl.ipk` (أو apk) → ضبط bot_token وchat_id في إعدادات LuCI → اختبار الاتصال → حفظ.

القسم الخامس: التحكم والإجراءات

الإجراء	النتيجة	التفاصيل
حظر الأجهزة	نعم	حظر Layer 3 فوري مع flush conntrack + حظر WiFi عبر hostapd_cli
تحديد السرعة	نعم	Rate Limiter (policer) + Traffic Shaper (tc/HTB) مع خيارات متعددة
إشعار أجهزة جديدة	نعم	3 مصادر: ARP + DHCP + WiFi stations مع hotplug فوري و17 متغير قالب
مراقبة الباندويث	نعم	مراقبة فورية عبر conntrack مع sparklines حية ورسوم تفاعلية
إعادة تشغيل الراوتر	لا	لا يوجد أمر إعادة تشغيل في البوت أو LuCI

القسم السادس: التوافق مع Starlink

يعمل بشكل طبيعي مع Starlink كاتصال WAN لأنه يعمل على جانب LAN فقط. ملاحظة: متغير wan_ip في القوالب قد يظهر IP خاص (CGNAT) بدلاً من IP العام.

القسم السابع: التقييم

المعيار	التقييم
سهولة الإعداد	سهل
التقييم العام	**8.5/10**

الأقوى في فئته: luci-app-trafficctl هو الأكثر شمولاً من حيث مراقبة الأجهزة والتحكم فيها عبر Telegram مع أزرار تفاعلية حقيقية. مثالي للتحكم الأبوي وإدارة الشبكات المنزلية.

6. MickroTikBot (Java)

https://github.com/skiftare/mickrotikbot

القسم الأول: وصف عام

MickroTikBot هو بوت Telegram مكتوب بلغة Java لإدارة راوتر MikroTik يبيع وصول VPN من نوع L2TP للمستخدمين. **ليس أداة إدارة شبكة عامة**، بل هو منصة VPN-as-a-Service مبنية فوق MikroTik CHR (Cloud Hosted Router). يقوم المستخدمون بالتسجيل عبر Telegram، إيداع عملة مشفرة (Stellar/XLM)، والحصول على بيانات اعتماد L2TP VPN. المشروع طُوّر كمشروع تعليمي لتعلم مبادئ البرمجة كائنية التوجه (OOP) في الجامعة.

القسم الثاني: الإمكانيات والميزات

- **تسجيل المستخدمين:** إنشاء حسابات VPN عبر Telegram.
- **دفع عملة مشفرة:** دمج مع شبكة Stellar (XLM) للدفع.
- **إنشاء اتصالات L2TP:** إنشاء بيانات اعتماد VPN تلقائياً على MikroTik CHR.
- **ملف شخصي للمستخدم:** رصيد، تاريخ الدفع، بيانات الاتصال.
- **تكامل gRPC:** اتصال بخادم RADIUS للمصادقة.
- **قاعدة بيانات مشفرة:** تخزين بيانات المستخدمين مع تشفير AES.

أزرار وأوامر Telegram:

أوامر تسجيل، دفع، عرض الرصيد، الحصول على بيانات VPN. لا توجد أزرار تفاعلية معروفة لإدارة الشبكة.

القسم الثالث: الصلاحيات المطلوبة

- **على الراوتر:** وصول MikroTik API كامل (username/password لـ CHR).
- **على الجهاز المضيف:** خادم Java (JDK)، قاعدة بيانات، خادم RADIUS.

القسم الرابع: متطلبات التثبيت

- **الأجهزة:** خادم (VPS/سحابي) لتشغيل MikroTik CHR (Cloud Hosted Router) + Java + خادم RADIUS.
- **البرامج:** IntelliJ IDEA، Java JDK، قاعدة بيانات، محفظة gRPC، Stellar.

- **خطوات التثبيت:** استنساخ المشروع ← إنشاء ملف .env بالمتغيرات (11 متغير) ← تشغيل خادم Java.

القسم الخامس: التحكم والإجراءات

الإجراء	النتيجة
حظر الأجهزة	✗
تحديد السرعة	✗
إشعار أجهزة جديدة	✗
مراقبة الباندويث	✗
إعادة تشغيل الراوتر	✗

القسم السادس: التوافق مع Starlink

لا علاقة له بـ Starlink. يعمل مع MikroTik CHR على خادم سحابي ويتطلب IP عام.

القسم السابع: التقييم

المعيار	التقييم
سهولة الإعداد	صعب
التقييم العام	**3/10** (مشروع تعليمي متخصص في بيع VPN وليس أداة مراقبة شبكة)

7. DroidNet Sentinel

https://github.com/kahz12/DroidNet-Sentinel

القسم الأول: وصف عام

DroidNet Sentinel هو مجموعة أدوات تدقيق لشبكات Wi-Fi تعمل بشكل مستقل على Android (عبر Termux) وLinux (Debian/Ubuntu/Kali). يكتشف الأجهزة النشطة على الشبكة المحلية، يبصم الخدمات المفتوحة عبر Nmap، يصنف مستوى المخاطر لكل جهاز، يبحث عن الثغرات المعروفة (CVEs)، يتتبع التغييرات بين عمليات المسح، ويمكنه اتخاذ إجراءات مضادة نشطة ضد الأجهزة غير الموثوقة. يوفر واجهة سطر أوامر (CLI) ولوحة تحكم ويب Flask.

القسم الثاني: الإمكانيات والميزات

- **اكتشاف الأجهزة:** مسح ARP ping sweep مع fallback لأساليب اكتشاف متقدمة.
- **مسح عميق:** فحص المنافذ والتعرف على الإصدارات لكل جهاز (Nmap -sV).
- **محرك تقييم المخاطر:** تصنيف MINIMAL/LOW/MEDIUM/CRITICAL بناءً على المنافذ المكشوفة.
- **تتبع التغييرات:** رصد الأجهزة الجديدة والمختفية وتغيرات المنافذ بين عمليات المسح.
- **البحث عن ثغرات:** استعلام Exploit-DB محلي + NIST NVD API لآخر 120 يوماً.
- **استجابة نشطة (ARP Cut):** قطع الإنترنت عن الأجهزة غير الموثوقة عبر ARP spoofing.
- **Deauth 802.11:** إرسال إطارات deauth ضد عميل معين أو بث عام.
- **لوحة تحكم ويب:** واجهة Flask مصادق عليها مع سجل المسح وعرض الفروقات.
- **وضع Daemon:** مسح مستمر كل 5 دقائق في الخلفية.

أزرار وأوامر Telegram:

لا توجد أوامر أو أزرار تفاعلية. إشعارات أحادية الاتجاه فقط (ملخص المسح + تنبيهات CVE) عبر متغيرات بيئة TELEGRAM_TOKEN وTELEGRAM_CHAT_ID.

القسم الثالث: الصلاحيات المطلوبة

- **على الراوتر:** لا يحتاج أي وصول للراوتر.
- **على الجهاز المضيف:** Root/Sudo لـ ARP spoofing و,Deauth. Nmap, termux-api (Android) dsniff, scapy, aircrack-ng للهجمات.

القسم الرابع: متطلبات التثبيت

- **الأجهزة:** هاتف Android مع Termux أو جهاز Linux. محول WiFi يدعم monitor mode للـ Deauth.
- **البرامج:** Python 3.10+, Nmap, Flask, Rich, Scapy اختيارياً.
- **خطوات التثبيت (Android):** `pkg install python nmap termux-api` → `git clone` → `pip install -r requirements.txt` → `python main.py`

القسم الخامس: التحكم والإجراءات

الإجراء	النتيجة	التفاصيل
حظر الأجهزة	نعم	ARP poisoning (Spoofer module) — عبر Linux فقط مع root
تحديد السرعة	لا	قطع كامل فقط بدون تحديد سرعة
إشعار أجهزة جديدة	نعم	رصد الأجهزة الجديدة عبر diff engine مع إشعارات Telegram
مراقبة الباندويث	لا	لا يراقب استهلاك الباندويث
إعادة تشغيل الراوتر	لا	لا يتكامل مع إدارة الراوتر

القسم السادس: التوافق مع Starlink

يعمل كأداة تدقيق عامة على أي شبكة محلية بما فيها Starlink. لا يوجد تكامل خاص بـ Starlink (لا يمكنه قراءة إحصائيات الطبق أو التفاعل مع Starlink API).

القسم السابع: التقييم

المعيار	التقييم
سهولة الإعداد	متوسط
التقييم العام	**7.5/10** (ممتاز كأداة تدقيق أمني لكنه ليس لوحة تحكم شبكة)

8. NetAlertX

https://github.com/jokob-sk/NetAlertX

القسم الأول: وصف عام

NetAlertX (سابقاً Pi.Alert) هو إطار عمل مفتوح المصدر لرؤية الشبكة وذكاء الأصول يعمل كمرجع حقيقة للشبكة (Network Source of Truth). يكتشف ويصنف ويراقب جميع الأجهزة على الشبكات المحلية والبعيدة، ويكشف الأجهزة الجديدة/غير المصرح بها، ويتتبع تغيرات حالة الاتصال، ويرسل إشعارات عبر أكثر من 80 خدمة. يعمل كحاوية Docker على أي نظام Linux، مع أكثر من 6,500 نجمة على GitHub ودعم نشط. مشروع ناضج ومناسب بشدة لبيئات Homelab وفرق IT ومزودي الخدمات المدارة (MSPs).

القسم الثاني: الإمكانيات والميزات

- **اكتشاف شامل:** ARP-scan, Nmap, ICMP, DHCP leases, Pi-hole, UniFi, MikroTik, AsusWRT, Fritz!Box, OpenWRT, SNMP, OMADA.
- **80+ خدمة إشعار:** Telegram, NTFY, Pushover, Pushsafer, Email, MQTT, Webhooks, Apprise (Discord, Slack, Signal...).
- **تكاملات واسعة:** Home Assistant MQTT, REST API, GraphQL, Prometheus, MCP Server لـ AI Bridge.
- **وحدات Workflow:** أتمتة حوكمة IT مع محفزات وشروط AND/OR وإجراءات.
- **مخطط طوبولوجيا الشبكة:** خريطة شبكة بصرية مولّدة تلقائياً.
- **لوحة NOC:** شاشة جدار حائط لمراكز العمليات.
- **مزامنة متعددة المواقع:** Sync Nodes (PUSH/PULL) للمراقبة المركزية.
- **إضافات نظام:** WOL, اختبار سرعة الإنترنت، مراقبة IP العام، مراقبة المواقع، نسخ CSV.
- **تصميم أمني:** نظام ملفات للقراءة فقط، مستخدم غير root، قدرات Linux محدودة.

أزرار وأوامر Telegram:

لا توجد أوامر أو أزرار تفاعلية. إشعارات Telegram أحادية الاتجاه فقط (أجهزة جديدة، أجهزة غير متصلة، تغيرات IP). يمكن استخدام Webhook + n8n لبناء بوت تفاعلي مخصص.

القسم الثالث: الصلاحيات المطلوبة

- **على الراوتر:** قراءة فقط (API/SSH/SNMP) لاستيراد بيانات الأجهزة من الراوتر.

- **على الجهاز المضيف:** Docker مع `--network=host` و CAP_NET_RAW لمسح ARP. يعمل كمستخدم غير root (UID 20211).

القسم الرابع: متطلبات التثبيت

- **الأجهزة:** أي جهاز يدعم Docker (Raspberry Pi, NAS, mini PC, VM) بحد أدنى 256 ميجابايت RAM.
- **البرامج:** Docker + Docker Compose (أساسي) أو Home Assistant (add-on).
- **خطوات التثبيت:** `docker run -d --network=host --restart=unless-stopped -v /local_data_dir:/data ghcr.io/netalertx/netalertx:latest` → الوصول عبر المتصفح على المنفذ 20211 → ضبط SCAN_SUBNETS → تفعيل الإضافات.

القسم الخامس: التحكم والإجراءات

الإجراء	النتيجة	التفاصيل
حظر الأجهزة	لا	أداة مراقبة فقط — لا يمكنها حظر أجهزة
تحديد السرعة	لا	لا يوجد إدارة QoS أو عرض نطاق
إشعار أجهزة جديدة	نعم	الميزة الأساسية — كشف فوري عبر 80+ خدمة إشعار
مراقبة الباندويث	محدود	اختبار سرعة الإنترنت فقط — لا مراقبة لكل جهاز
إعادة تشغيل الراوتر	لا	لا يتحكم بالراوتر — مراقبة فقط

القسم السادس: التوافق مع Starlink

يعمل كمراقب سلبي على أي شبكة متصلة بـ Starlink. يكتشف طبق Starlink كجهاز عادي على الشبكة. لا يوجد تكامل مع Starlink gRPC API. يمكن بناء إضافة مخصصة لقراءة إحصائيات الطبق.

القسم السابع: التقييم

المعيار	التقييم
سهولة الإعداد	سهل
التقييم العام	**8.5/10** (الأقوى في الاكتشاف والمراقبة لكنه لا يتحكم في الأجهزة)

جدول المقارنة النهائي

المشروع	النظام الأساسي	إشعار أجهزة جديدة	حظر الأجهزة	تحديد السرعة	مراقبة الباندويث	إعادة تشغيل الراوتر	أزرار Telegram	التوافق مع Starlink	سهولة الإعداد
TLGRM Script	MikroTik RouterOS	محدود	نعم	نعم	لا	نعم	أوامر نصية فقط	عبر MikroTik + Bypass	متوسط
luci-app-telemt	OpenWrt	لا	لا	محدود	محدود	لا	Inline Buttons كاملة	مشاكل CGNAT	متوسط
Dynamic-IP-Tracker	MikroTik RouterOS	لا	لا	لا	لا	لا	لا يوجد	جزئي (IP عام فقط)	سهل
RouterOS Scripts	MikroTik RouterOS	نعم	لا	نعم	محدود	غير مباشر	لا يوجد (إشعارات فقط)	يعمل مع أي WAN	متوسط
luci-app-trafficctl	OpenWrt	نعم	نعم	نعم	نعم	لا	Inline Buttons كاملة	يعمل على جانب LAN	سهل
MickroTikBot	Java + MikroTik CHR	لا	لا	لا	لا	لا	أوامر تسجيل/ دفع فقط	لا ينطبق	صعب
DroidNet Sentinel	Android/Linux	نعم	نعم	لا	لا	لا	إشعارات فقط	يعمل على أي LAN	متوسط
NetAlertX	Docker (Linux)	نعم	لا	لا	محدود	لا	إشعارات فقط	مراقب سلبي	سهل

الأسئلة الختامية والتوصيات

السؤال الأول: لشبكة Starlink التي تضم أكثر من 100 مستخدم، ما هو المشروع الذي تنصح به؟

التوصية: مزيج من ثلاثة مشاريع للحصول على أفضل تغطية:

1. **luci-app-trafficctl (التحكم الأساسي):** هذا هو المشروع الأمثل لشبكة Starlink كبيرة. يعمل على راوتر OpenWrt خلف Starlink بوضع Bypass، ويوفر كل ما تحتاجه: حظر أجهزة (Layer 3 + WiFi)، تحديد سرعة لكل جهاز (Policer + Shaper)، مراقبة باندويث حية لكل جهاز، إشعارات فورية عند اتصال أجهزة جديدة عبر Telegram مع أزرار تفاعلية. يدعم OpenWrt على أجهزة قوية مثل Raspberry Pi 4/5 أو mini PCs التي تتحمل أكثر من 100 مستخدم بسهولة.
2. **NetAlertX (الاكتشاف والمراقبة المركزية):** كمكمل على خادم منفصل (Raspberry Pi أو NAS)، يوفر اكتشافاً شاملاً للأصول، ومخطط طوبولوجيا للشبكة، ومزامنة متعددة المواقع، وإشعارات عبر 80+ خدمة، وواجهة NOC لشاشة العمليات. هذا مثالي لمراقبة الشبكة ككل وتوثيق كل جهاز متصل.
3. **DroidNet Sentinel (التدقيق الأمني الدوري):** كأداة تدقيق أمني على هاتف Android أو جهاز Linux للفحص الدوري وتحديد الثغرات والأجهزة المشبوهة.

إذا كان يجب اختيار مشروع واحد فقط، فـ **luci-app-trafficctl** هو الخيار الأفضل لأنه يوفر التحكم الكامل في الأجهزة والسرعة مع واجهة Telegram تفاعلية حقيقية، وهو الأسهل إعداداً والأكثر شمولاً في فئة التحكم النشط.

السؤال الثاني: ما هو راوتر Starlink Bypass Mode؟ وكيف يفعّل؟

وضع Bypass Mode هو ميزة في راوتر Starlink تسمح بتعطيل وظائف الراوتر المدمجة (NAT، DHCP، WiFi) وتحويل طبق Starlink إلى مودم جسر (Bridge Modem) فقط. عند تفعيله، يتوقف راوتر Starlink عن العمل كراوتر ويصبح مجرد جسر يمرر عنوان IP العام مباشرة إلى الراوتر المتصل به عبر كابل Ethernet. هذا ضروري لمن يريد استخدام راوتر خاص به (مثل OpenWrt أو MikroTik) لإدارة الشبكة بشكل كامل.

كيفية التفعيل:

- الخطوة 1: افتح تطبيق Starlink على هاتفك.

- الخطوة 2: اذهب إلى الإعدادات (Settings) ← Advanced.
- الخطوة 3: فعّل خيار **Bypass Mode**.
- الخطوة 4: ستحصل على رسالة تحذيرية تؤكد أنك تفهم العواقب — اقبلها.
- الخطوة 5: بعد التفعيل، لن يمكنك الوصول إلى راوتر Starlink إلا عبر العنوان `192.168.100.1` (أو `192.168.1.1` حسب النموذج).
- الخطوة 6: صل راوترك الخاص (OpenWrt/MikroTik) بمنفذ WAN من طبق Starlink عبر كابل Ethernet.
- الخطوة 7: اضبط منفذ WAN على راوترك للحصول على IP عبر DHCP من Starlink.

تحذير مهم: عند تفعيل Bypass Mode، لن يعمل WiFi المدمج في Starlink بعد الآن. ستحتاج بالتأكيد إلى راوتر إضافي لتوفير WiFi وإدارة الشبكة. أيضاً، لن يكون تطبيق Starlink قادراً على عرض إحصائيات الشبكة بشكل طبيعي — ستحتاج للوصول عبر عنوان IP المحلي.

السؤال الثالث: ما نوع الراوتر الإضافي الذي تحتاجه لتشغيل هذه المشاريع على Starlink؟

اختيار الراوتر الإضافي يعتمد على المشاريع التي ستستخدمها:

بالنسبة لـ luci-app-trafficctl (التوصية الرئيسية):

تحتاج راوتر OpenWrt بقوة معالجة كافية لأكثر من 100 مستخدم. الخيارات الموصى بها:

- **Raspberry Pi 4/5 مع OpenWrt:** خيار اقتصادي وقوي. RPi 5 بـ 4-8 جيجابايت RAM يكفي لـ 100+ مستخدم بسهولة، مع دعم كامل لـ luci-app-trafficctl والتحكم في الحركة.
- **FriendlyElec NanoPi R4S/R5S:** مصمم خصيصاً كراوتر OpenWrt بمعالج ARM قوي وواجهتي شبكة Gigabit.
- **mini PC (x86) مع OpenWrt:** خيارات مثل Intel N100 mini PC توفر أداءً ممتازاً مع دعم كامل لـ SQM/QoS لتحسين أداء Starlink.
- **راوترات OpenWrt التجارية:** مثل Banana Pi BPI-R3 أو Dynalink DL-WRX36 مع ذاكرة كافية.

بالنسبة لمشاريع MikroTik (TLGRM, RouterOS Scripts):

تحتاج راوتر MikroTik يدعم RouterOS v7. الخيارات:

- **hAP ax3 (C53UiG+5HPaxD2HPaxD):** راوتر WiFi 6 حديث بسعر معقول.
- **RB5009UG+S+IN:** راوتر متقدم بـ 9 منافذ Gigabit + SFP+ 10G، مثالي للشبكات الكبيرة.
- **CCR2004-1G-12S+2XS:** راوتر من فئة Cloud Core للشبكات الكبيرة جداً.

الحد الأدنى من المواصفات لشبكة 100+ مستخدم:

- **المعالج:** ARM Cortex-A72 أو أفضل (أو Intel x86 مكافئ).
- **RAM:** 1 جيجابايت كحد أدنى، 2+ جيجابايت مفضل.
- **التخزين:** 16+ ميجابايت فلاش أو SD card.
- **الشبكة:** Gigabit Ethernet (منفذان على الأقل: WAN + LAN).
- **WiFi:** WiFi 6 مفضل (AX1800 أو أفضل) مع دعم نطاقين (2.4GHz + 5GHz).

مخطط التوصيل:

طبق Starlink → (كابل Ethernet) → منفذ WAN على الراوتر الإضافي (OpenWrt/MikroTik) → أجهزة الشبكة عبر WiFi و/أو منافذ LAN. الراوتر الإضافي يدير DHCP، الجدار الناري، QoS، ويعمل عليه بوت Telegram أو سكربت المراقبة.