Home IoT Network Monitoring & Defense: From Quick Wins to Local AI

Compiled on November 08, 2025 02:03

User Question / Scenario

In a simple example, a home router does a fair job of being a firewall mostly just by implementing NAT protocols that keep out inbound traffic.
But when a person adds an IOT device that allows remote operation, effectively firewall has a hole in it.
In a worst case the IOT device at some point requests a host somewhere on the internet for any software updates. And the software update contains exploits a destructive kind. Say ransomware for our discussion, or an enemy state actor that just wants to disable home networks.

The problem seems to be that a typical home user does not have much of a way to know what internet traffic the devices on a network is happening.

I have a new ASUS router and I was interested to see that it has a basic traffic tracking function. But it seems more focused on finding excessive data use. Maybe someone using all the bandwidth running a web game. Or a bad phone app that is forwarding everything the camera can see.

I also have some familiarity with Wireshark and have had a collage class where we decoded some traffic.

But the manual use of Wireshark is not well adapted to finding a stray packet here or there from something like a wall thermostat to an address in China.

It would seem that an expert system that is trained with knowledge of world wide IP addressing could look at the stream and pick out things that a manual operator could not.

And now there is talk of running compressed AI models on PC level hardware. Such a model would probably need special training.

There is also a certain element of learning or dynamic adaption that is needed. Maybe a distributed network between different nodes (home networks) to update other members when bad things are found on one node.

Assistant Response

Got it—great expansion. You’re describing a home-grade IDS/“expert system” that can (1) see everything on-net, (2) flag risky destinations (geo/IP/reputation), (3) learn each device’s normal behavior, and (4) optionally share new bad indicators with other homes. Here’s a practical, buildable path in layers—from “you can do this tonight” to “small-lab research with local AI.”

0) Quick wins (weekend setup)
• Put IoT on its own VLAN/SSID
– On the ASUS: create an “IoT” guest SSID; disable “Intranet access”/“Access to local network.”
– Bonus: a “Quarantine” SSID for brand-new devices until vetted.
• Log all DNS, then enforce it
– Run Pi-hole or AdGuard Home on a tiny box (Pi/mini-PC). Point ASUS DHCP to that DNS.
– Turn on query logging + blocklists; add DoH/DoT upstreams; enable “never forward non-FQDN/private” to catch odd beacons.
– Add a small GeoIP/ASN note in the logs (plug-in or sidecar script) so queries resolve to

“Country / ASN / Provider”.

- Baseline reports, not packets
  – Instead of diving into Wireshark, export NetFlow/IPFIX or conntrack summaries from the router (or mirror IoT VLAN to a tap) and visualize with ntopng.

1) Expert-system style detection (rule + intel + geo)

- Suricata (IDS/IPS, rule-based) in alert mode with curated categories.
- Zeek (behavioral): scripts for new-country, new-service, beaconing.
- Reputation & Geo policy: light feeds + local allowlist; soft geo-blocking by ASN/country.

2) Lightweight local ML (no cloud, runs on a PC)

- Learn per-device flow features (destinations, ports, timing, size).
- Isolation Forest / One-Class SVM; daily refit; simple thresholds.
- Explain alerts with top features (“14 new domains; new ASN; 3:12am”).

3) Policy enforcement loop

- Observe-only first; then auto-quarantine MAC to a “Quarantine VLAN” on High alerts.
- Captive portal for Allow once / Allow always / Keep blocked.

4) Cooperative defense (privacy-preserving)

- Share only indicators (domain hash/ASN/port/time bucket/device class) with a small trusted group.
- Require ≥2 peers to confirm before auto-quarantine; decay reputation in 7–14 days.

5) Where ASUS fits

- Use built-in visibility but mirror IoT traffic to a sensor box if possible.
- If needed, place an OpenWrt/pfSense bridge inline to run Suricata/Zeek/ntopng and enforce egress.

6) Hardware sketches

- Fanless mini-PC (Intel N100/8–16GB, dual NICs) as sensor/gateway.
- Keep 7–14 days of logs; rotate.

7) First build checklist

[ ] IoT + Quarantine SSIDs on ASUS; block LAN access.
[ ] Pi-hole/AdGuard with logging; DHCP points to it.
[ ] ntopng for daily “New Destinations by Device” report.
[ ] Suricata + Zeek with basic scripts and ET Open rules.
[ ] 50-line Python job for Isolation Forest over Zeek rollups.
[ ] Quarantine script to move MAC → VLAN; test on a spare bulb.
[ ] Optional review portal to approve blocks and generate allow/deny rules.

Why this helps the “thermostat to China” case

- DNS/flow logs surface new country/ASN and never-seen domain.
- Rules catch odd protocols/TLS patterns; ML elevates 3am beacons.
- Enforcement moves device to Quarantine and asks you what to do.