LIFX Smart Bulb — Home Network Pentest Runbook

Discover and control my LIFX bulbs on my own LAN using Kali Linux. Beginner reference — repeatable.

Step 1 — Find my network (know where I am)

On Windows check my LAN; in Kali confirm the attacker box is on the same subnet.

```
ipconfig              # Windows: my IPv4 + gateway
ip a                  # Kali: look at eth0 inet address
```

Note *Windows = 10.0.0.207, Kali eth0 = 10.0.0.209 → same 10.0.0.x subnet, so my bridged Kali VM can see all devices. If Kali were on a different subnet (10.0.2.x, 172.x), switch the VM adapter from NAT to Bridged. My network = 10.0.0.0/24, gateway 10.0.0.1.*

Step 2 — Discover live devices (who's on my network)

```
sudo nmap -sn 10.0.0.0/24
```

Note *Ping sweep: lists every host + MAC vendor. "Lifi Labs Management" = my LIFX bulbs. That vendor tag is how I ID the target.*

Step 3 — Enumerate the bulb (what ports are open)

```
sudo nmap -sV -sC -p- 10.0.0.7
```

Note *Result: ALL TCP ports closed. Lesson: this does NOT mean secure — the bulb only speaks UDP. Always check UDP before calling a device empty.*

```
sudo nmap -sU -sV -p 56700 10.0.0.7
```

Note *Result: 56700/udp open|filtered. "open|filtered" = no reply to generic probes (normal for UDP). The bulb only answers real LIFX-protocol packets.*

Nmap flags I used

- **sudo** runs as root — nmap's raw-packet scans need admin access.
- **-sn** ping/host discovery only, no port scan — just find live devices.
- **-sV** version detection — identify the service AND its version (key for finding known CVEs).
- **-sC** runs nmap's default safe scripts — auto-checks for default creds, exposed info, misconfigs.
- **-p-** scan all 65,535 ports (default is only the top 1,000). Slow but complete.
- **-p 56700** scan one specific port I already know about — fast and targeted.
- **-sU** UDP scan (default is TCP). Essential here — LIFX only listens on UDP.
- **-Pn** skip host discovery, assume host is up — use if a device blocks pings but I know it's on.

Step 4 — Why port 56700? (how I'd find this without help)

1) nmap knows it — listed in nmap-services as "lifx". 2) Vendor docs: search "LIFX LAN protocol" (lan.developer.lifx.com). 3) Read open-source tools (lifxlan, Home Assistant) on GitHub. 4) Wireshark: sniff the official app toggling a light and read the packets.

Step 5 — Speak the protocol: discover, control, recolor (the breach)

```
pip install lifxlan --break-system-packages
```

Note *--break-system-packages lets pip install into system Python on Kali (needed since Kali marks it externally-managed).*

```
python3 -c "from lifxlan import LifxLAN; lan = LifxLAN(); [print(d.get_label(), d.get_ip_addr(), d.get_power()) for d in lan.get_devices()]"
```

Note *Broadcasts a LIFX discovery packet; every bulb replies with name, IP, power state — NO authentication. That reply = proof control is unauthenticated. (Got: Desk Lamp Right 10.0.0.147, Desk Lamp Left 10.0.0.7.)*

```
python3 -c "from lifxlan import LifxLAN; import time; lan=LifxLAN(); d=[x for x in lan.get_devices() if x.get_label()=='Desk Lamp Left'][0]; d.set_power(True); time.sleep(2); d.set_power(False)"
```

Note *set_power(True/False) turns the lamp on for 2s then off — physical control with zero credentials.*

```
python3 -c "from lifxlan import LifxLAN, RED, GREEN, BLUE; import time; d=[x for x in LifxLAN().get_devices() if x.get_label()=='Desk Lamp Left'][0]; [(d.set_color(c), time.sleep(1)) for c in (RED, GREEN, BLUE)]"
```

Note *set_color(c) cycles the bulb red → green → blue, 1s each. RED/GREEN/BLUE are built-in color presets in lifxlan; the same no-auth channel controls color, not just power.*

Finding & fix

Verdict: Old critical flaws (plaintext Wi-Fi creds) needed physical access and are patched. The unauthenticated LAN control protocol is by-design and live: anyone on my Wi-Fi can find and control my bulbs.

Remediation: Put my IoT devices on a separate Guest/IoT network (VLAN) so a compromised device is isolated from my PCs and phones.

Repeat checklist (quick run next time)

1. ip a / ipconfig → confirm same subnet. 2. sudo nmap -sn 10.0.0.0/24 → find "Lifi Labs" IP. 3. sudo nmap -sU -sV -p 56700 <ip> → confirm control port. 4. lifxlan discover → set_power / set_color to control. Swap the IP; the workflow is identical for any device (just research its ports/protocol).