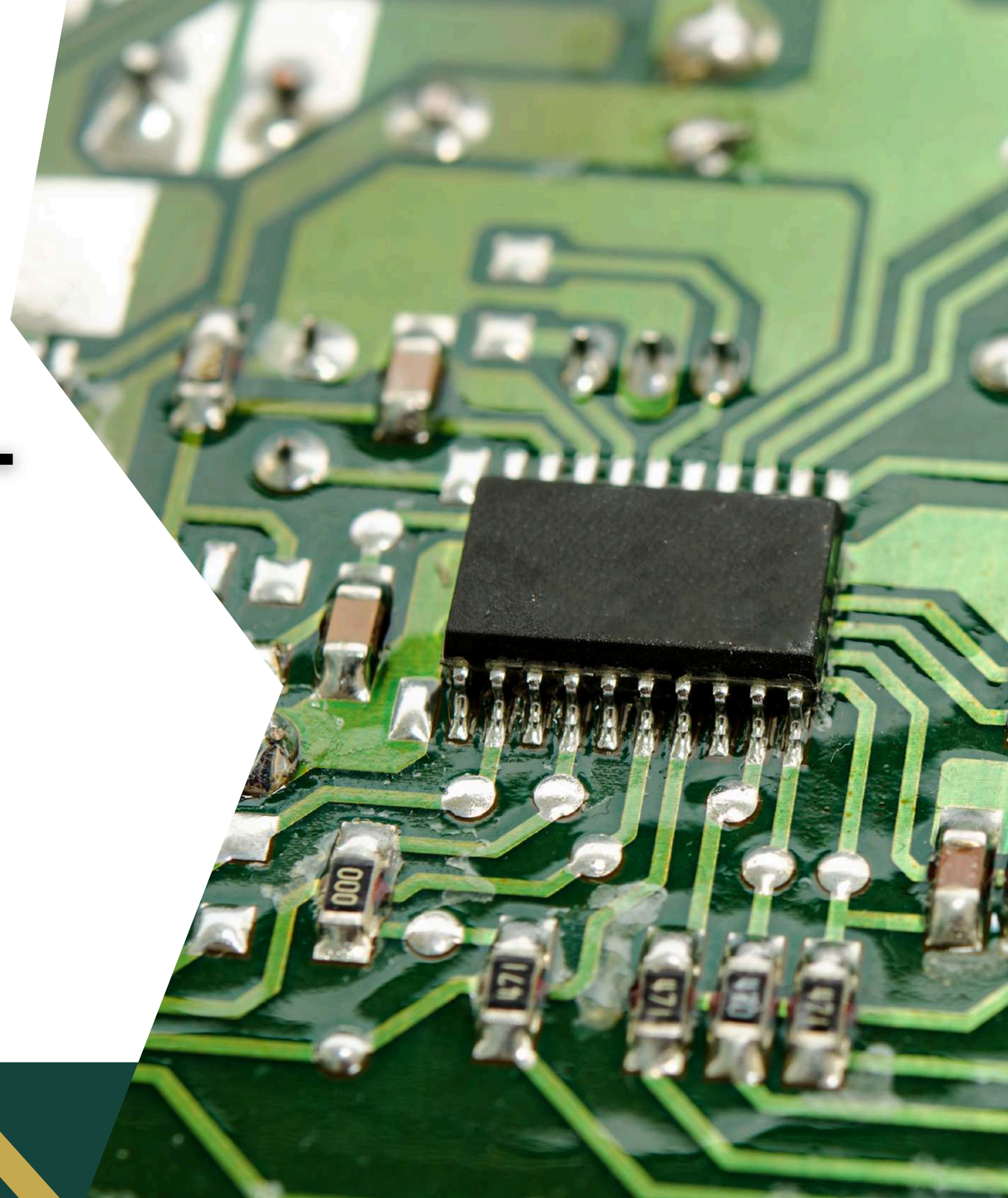
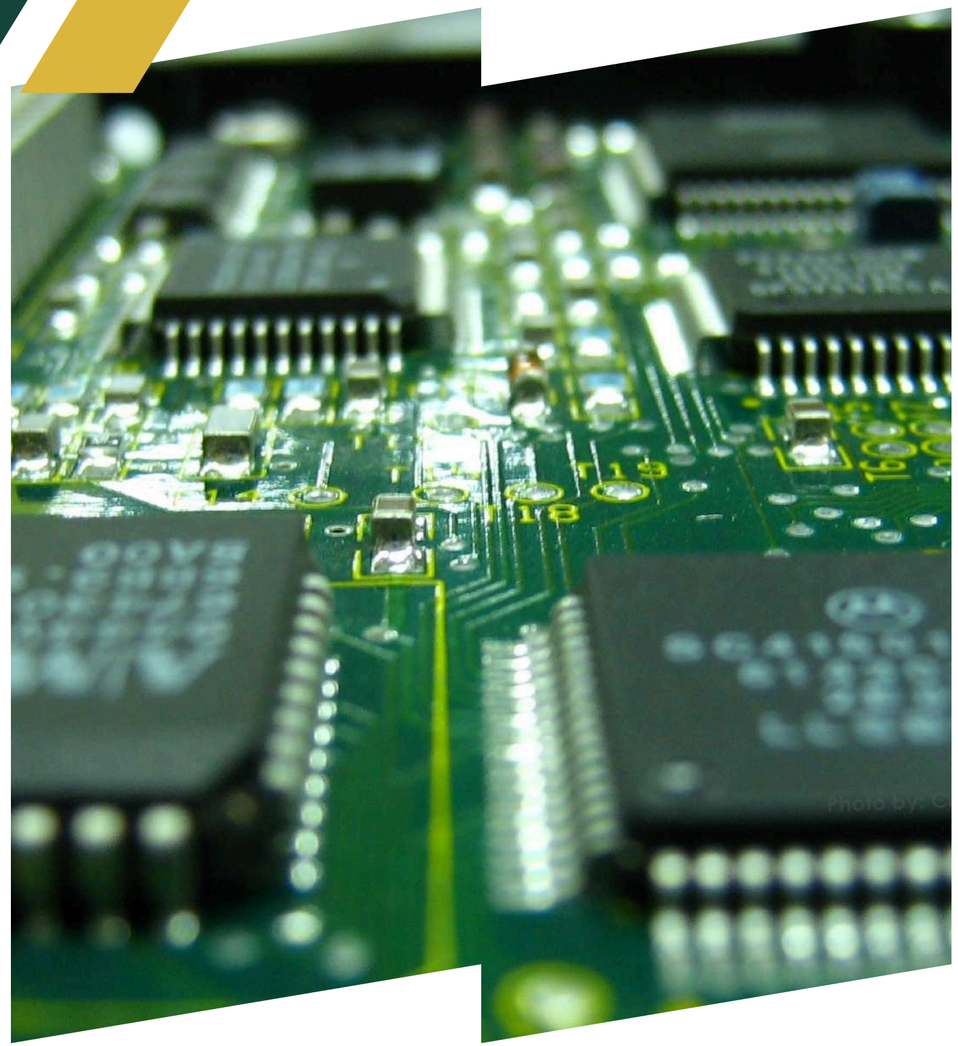
HARDWARE & IOT HACKING - SECURITY OVERVIEW

What attackers look for, simple techniques, and how to defend devices



INTRODUCTION

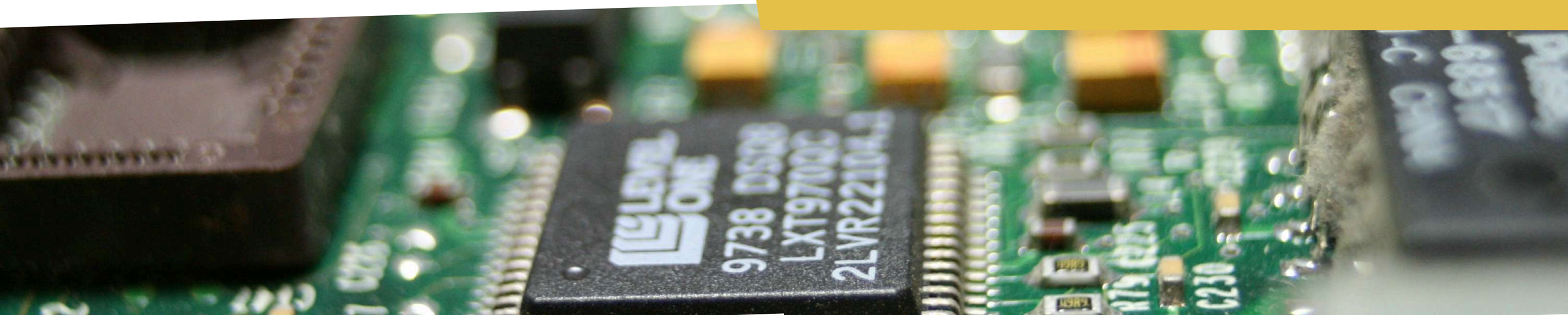


“IoT devices are everywhere — hacking them can expose privacy or let attackers control things. We’re looking at how that happens and how to stop it.”

- One-line definition: IoT = everyday devices connected to the Internet (cameras, sensors, smart plugs).
- What hardware hacking is: Exploring device internals to find weaknesses or fix problems.
- Why it matters: Vulnerabilities can leak data, allow device takeover, or spread inside networks.

WHAT IS IOT & WHY HARDWARE HACKING?

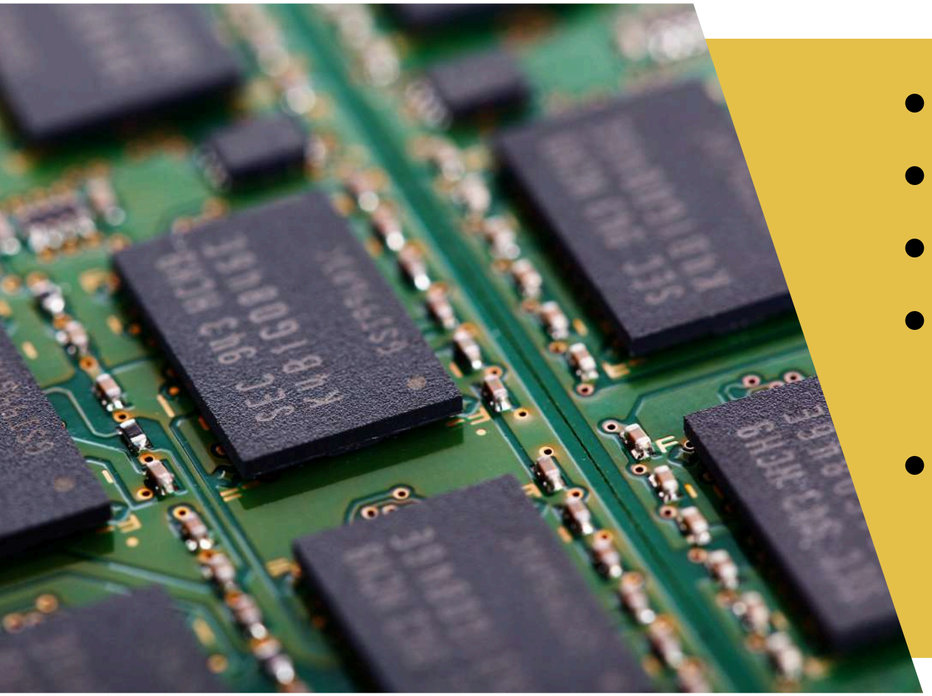
- Devices have small computers (MCUs), flash memory, radios, sensors.
- Why attackers care: These parts can contain secrets, credentials, or debug ports.
- Goal of hacking: Find bugs, recover bricked devices, or test security.
- Think of each device as a tiny computer — if an attacker finds the wrong part exposed, they can get into it.”



TYPICAL IOT COMPONENTS

- MCU / SoC (the device brain)
- Flash memory (stores firmware), RAM, power components
- Wireless radios (BLE, Wi-Fi, Zigbee) and sensors
- “Inside every product are chips and radios — those are the components attackers target.”

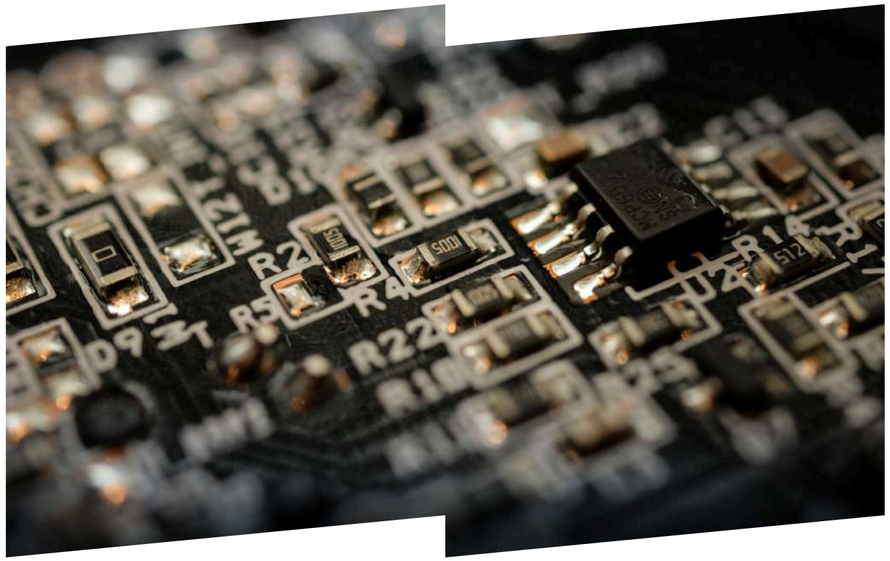
COMMON ATTACK SURFACES

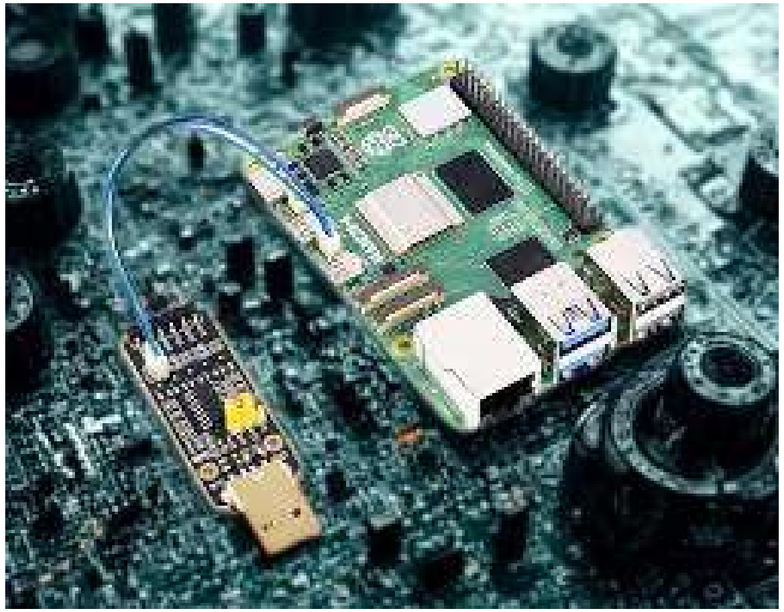


- Debug ports: UART, JTAG, SWD
- External storage: SPI/NOR flash, SD cards
- Wireless interfaces and OTA update paths
- Developer leftovers: hardcoded credentials or enabled debug
- “Attackers look for physical ports, stored firmware, wireless links, and leftover developer settings.”

UART / SERIAL DEBUG + SHORT DEMO NOTE

- UART: small 4-pin pads (VCC, GND, TX, RX) — shows boot logs
- Tool note: USB-TTL cable (FTDI) to read logs
- “UART is the simplest debug port — it can reveal boot messages and sometimes give a shell.”





“JTAG & SWD”

JTAG / SWD — low-level debug ports on many boards.

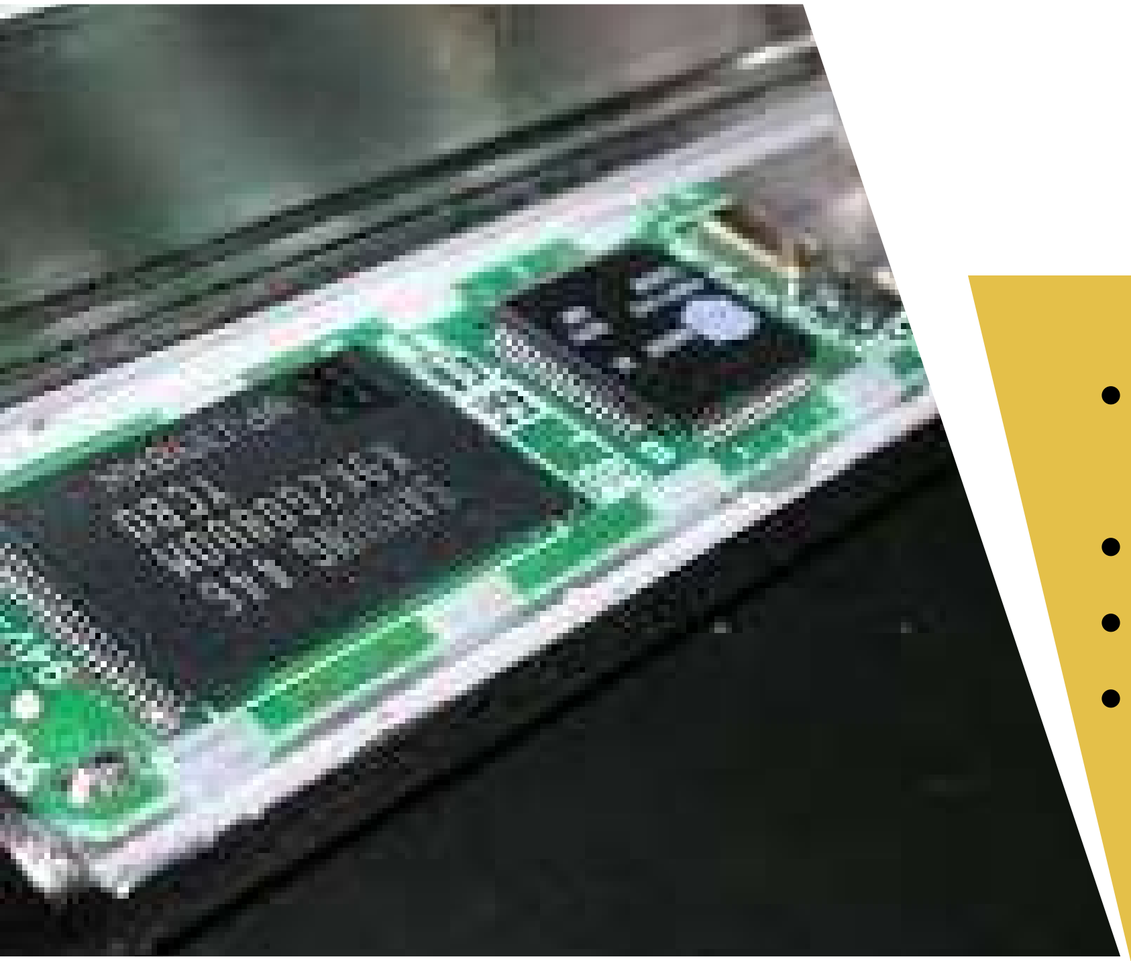
When enabled they give direct access to the CPU: halt execution, read/write memory, and run code.

If not protected, attackers can dump secrets, extract keys, or bypass boot checks.

Common tools: JTAGulator (find pins), OpenOCD / pyOCD (talk to target), and hardware debuggers (STM-link, Segger).

Protections: disable or fuse debug in production, lock JTAG with password/boundary scan protections, and enable secure boot + memory encryption.

FLASH MEMORY & FIRMWARE EXTRACTION



- Flash chips store firmware → can be read with SPI programmer or clip (CH341A)
- Extracted image → inspect with binwalk, Ghidra
- If OTA locked, direct flash dump is a fallback
- “When update channels are locked, attackers can clip the flash chip and read the firmware directly.”

WIRELESS & ADVANCED ATTACKS

- Wireless risks: BLE pairing sniffing, Wi-Fi deauth/rogue APs, Zigbee replay
- Advanced: power/EM side-channel, voltage glitching (needs special gear)
- Note: these are harder but very effective against weak devices
- “Wireless attacks are common; side-channel and fault attacks are harder but can break strong protections.”

DEFENSES & CHECKLIST

- Disable debug ports in production or physically fuse them
- Use secure boot & sign firmware; encrypt sensitive flash regions
- Avoid hardcoded credentials — use unique per-device keys
- Lab safety: test only with permission; isolate test networks
- “Defend devices by removing debug access, using secure boot, and giving each device its own keys. And always test legally.”

THANK YOU

“ If you have any questions just feel free to ask.”