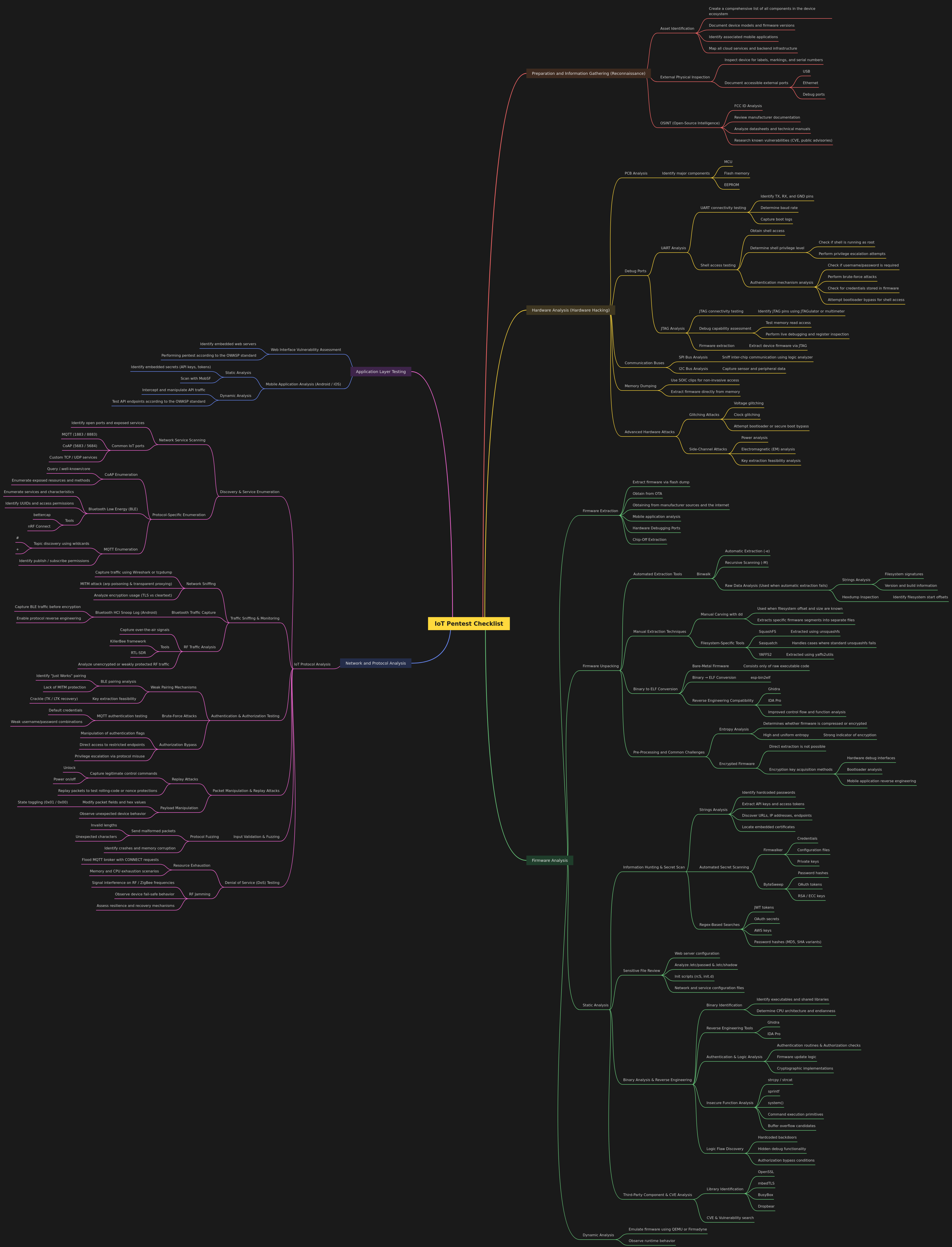

IoT Pentest Checklist
Preparation and Information Gathering (Reconnaissance)
Asset Identification
Create a comprehensive list of all components in the device ecosystem
Document device models and firmware versions
Identify associated mobile applications
Map all cloud services and backend infrastructure
External Physical Inspection
Inspect device for labels, markings, and serial numbers
Document accessible external ports
USB
Ethernet
Debug ports
OSINT (Open-Source Intelligence)
FCC ID Analysis
Review manufacturer documentation
Analyze datasheets and technical manuals
Research known vulnerabilities (CVE, public advisories)
Hardware Analysis (Hardware Hacking)
PCB Analysis
Identify major components
MCU
Flash memory
EEPROM
Debug Ports
UART Analysis
UART connectivity testing
Identify TX, RX, and GND pins
Determine baud rate
Capture boot logs
Shell access testing
Obtain shell access
Determine shell privilege level
Check if shell is running as root
Perform privilege escalation attempts
Authentication mechanism analysis
Check if username/password is required
Perform brute-force attacks
Check for credentials stored in firmware
Attempt bootloader bypass for shell access
JTAG Analysis
JTAG connectivity testing
Identify JTAG pins using JTAGulator or multimeter
Debug capability assessment
Test memory read access
Perform live debugging and register inspection
Firmware extraction
Extract device firmware via JTAG
Communication Buses
SPI Bus Analysis
Sniff inter-chip communication using logic analyzer
I2C Bus Analysis
Capture sensor and peripheral data
Memory Dumping
Use SOIC clips for non-invasive access
Extract firmware directly from memory
Advanced Hardware Attacks
Glitching Attacks
Voltage glitching
Clock glitching
Attempt bootloader or secure boot bypass
Side-Channel Attacks
Power analysis
Electromagnetic (EM) analysis
Key extraction feasibility analysis
Firmware Analysis
Firmware Extraction
Extract firmware via flash dump
Obtain from OTA
Obtaining from manufacturer sources and the internet
Mobile application analysis
Hardware Debugging Ports
Chip-Off Extraction
Firmware Unpacking
Automated Extraction Tools
Binwalk
Automatic Extraction (-e)
Recursive Scanning (-M)
Raw Data Analysis (Used when automatic extraction fails)
Strings Analysis
Filesystem signatures
Version and build information
Hexdump Inspection
Identify filesystem start offsets
Manual Extraction Techniques
Manual Carving with dd
Used when filesystem offset and size are known
Extracts specific firmware segments into separate files
Filesystem-Specific Tools
SquashFS
Extracted using unsquashfs
Sasquatch
Handles cases where standard unsquashfs fails
YAFFS2
Extracted using yaffs2utils
Binary to ELF Conversion
Bare-Metal Firmware
Consists only of raw executable code
Binary → ELF Conversion
esp-bin2elf
Reverse Engineering Compatibility
Ghidra
IDA Pro
Improved control flow and function analysis
Pre-Processing and Common Challenges
Entropy Analysis
Determines whether firmware is compressed or encrypted
High and uniform entropy
Strong indicator of encryption
Encrypted Firmware
Direct extraction is not possible
Encryption key acquisition methods
Hardware debug interfaces
Bootloader analysis
Mobile application reverse engineering
Static Analysis
Information Hunting & Secret Scan
Strings Analysis
Identify hardcoded passwords
Extract API keys and access tokens
Discover URLs, IP addresses, endpoints
Locate embedded certificates
Automated Secret Scanning
Firmwalker
Credentials
Configuration files
Private keys
ByteSweep
Password hashes
OAuth tokens
RSA / ECC keys
Regex-Based Searches
JWT tokens
OAuth secrets
AWS keys
Password hashes (MD5, SHA variants)
Sensitive File Review
Web server configuration
Analyze /etc/passwd & /etc/shadow
Init scripts (rcS, init.d)
Network and service configuration files
Binary Analysis & Reverse Engineering
Binary Identification
Identify executables and shared libraries
Determine CPU architecture and endianness
Reverse Engineering Tools
Ghidra
IDA Pro
Authentication & Logic Analysis
Authentication routines & Authorization checks
Firmware update logic
Cryptographic implementations
Insecure Function Analysis
strcpy / strcat
sprintf
system()
Command execution primitives
Buffer overflow candidates
Logic Flaw Discovery
Hardcoded backdoors
Hidden debug functionality
Authorization bypass conditions
Third-Party Component & CVE Analysis
Library Identification
OpenSSL
mbedTLS
BusyBox
Dropbear
CVE & Vulnerability search
Dynamic Analysis
Emulate firmware using QEMU or Firmadyne
Observe runtime behavior
Application Layer Testing
Web Interface Vulnerability Assessment
Identify embedded web servers
Performing pentest according to the OWASP standard
Mobile Application Analysis (Android / iOS)
Static Analysis
Identify embedded secrets (API keys, tokens)
Scan with MobSF
Dynamic Analysis
Intercept and manipulate API traffic
Test API endpoints according to the OWASP standard
Network and Protocol Analysis
IoT Protocol Analysis
Discovery & Service Enumeration
Network Service Scanning
Identify open ports and exposed services
Common IoT ports
MQTT (1883 / 8883)
CoAP (5683 / 5684)
Custom TCP / UDP services
Protocol-Specific Enumeration
CoAP Enumeration
Query /.well-known/core
Enumerate exposed resources and methods
Bluetooth Low Energy (BLE)
Enumerate services and characteristics
Identify UUIDs and access permissions
Tools
bettercap
nRF Connect
MQTT Enumeration
Topic discovery using wildcards
#
+
Identify publish / subscribe permissions
Traffic Sniffing & Monitoring
Network Sniffing
Capture traffic using Wireshark or tcpdump
MITM attack (arp poisoning & transparent proxying)
Analyze encryption usage (TLS vs cleartext)
Bluetooth Traffic Capture
Bluetooth HCI Snoop Log (Android)
Capture BLE traffic before encryption
Enable protocol reverse engineering
RF Traffic Analysis
Capture over-the-air signals
Tools
KillerBee framework
RTL-SDR
Analyze unencrypted or weakly protected RF traffic
Authentication & Authorization Testing
Weak Pairing Mechanisms
BLE pairing analysis
Identify "Just Works" pairing
Lack of MITM protection
Key extraction feasibility
Crackle (TK / LTK recovery)
Brute-Force Attacks
MQTT authentication testing
Default credentials
Weak username/password combinations
Authorization Bypass
Manipulation of authentication flags
Direct access to restricted endpoints
Privilege escalation via protocol misuse
Packet Manipulation & Replay Attacks
Replay Attacks
Capture legitimate control commands
Unlock
Power on/off
Replay packets to test rolling-code or nonce protections
Payload Manipulation
Modify packet fields and hex values
State toggling (0x01 / 0x00)
Observe unexpected device behavior
Input Validation & Fuzzing
Protocol Fuzzing
Send malformed packets
Invalid lengths
Unexpected characters
Identify crashes and memory corruption
Denial of Service (DoS) Testing
Resource Exhaustion
Flood MQTT broker with CONNECT requests
Memory and CPU exhaustion scenarios
RF Jamming
Signal interference on RF / ZigBee frequencies
Observe device fail-safe behavior
Assess resilience and recovery mechanisms