IoT Vulnerabilities Project

<https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#IoT_Vulnerabilities>

|  |  |  |
| --- | --- | --- |
| **Vulnerability** | **Attack Surface** | **Summary** |
| **Username Enumeration** | * Administrative Interface * Device Web Interface * Cloud Interface * Mobile Application | * Ability to collect a set of valid usernames by interacting with the authentication mechanism |
| **Weak Passwords** | * Administrative Interface * Device Web Interface * Cloud Interface * Mobile Application | * Ability to set account passwords to '1234' or '123456' for example. * Usage of pre-programmed default passwords |
| **Account Lockout** | * Administrative Interface * Device Web Interface * Cloud Interface * Mobile Application | * Ability to continue sending authentication attempts after 3 - 5 failed login attempts |
| **Unencrypted Services** | * Device Network Services | * Network services are not properly encrypted to prevent eavesdropping or tampering by attackers |
| **Two-factor Authentication** | * Administrative Interface * Cloud Web Interface * Mobile Application | * Lack of two-factor authentication mechanisms such as a security token or fingerprint scanner |
| **Poorly Implemented Encryption** | * Device Network Services | * Encryption is implemented however it is improperly configured or is not being properly updated, e.g. using SSL v2 |
| **Update Sent Without Encryption** | * Update Mechanism | * Updates are transmitted over the network without using TLS or encrypting the update file itself |
| **Update Location Writable** | * Update Mechanism | * Storage location for update files is world writable potentially allowing firmware to be modified and distributed to all users |
| **Denial of Service** | * Device Network Services | * Service can be attacked in a way that denies service to that service or the entire device |
| **Removal of Storage Media** | * Device Physical Interfaces | * Ability to physically remove the storage media from the device |
| **No Manual Update Mechanism** | * Update Mechanism | * No ability to manually force an update check for the device |
| **Missing Update Mechanism** | * Update Mechanism | * No ability to update device |
| **Firmware Version Display and/or Last Update Date** | * Device Firmware | * Current firmware version is not displayed and/or the last update date is not displayed |
| **Firmware and storage extraction** | * JTAG / SWD interface * [In-Situ dumping](https://www.flashrom.org/Flashrom) * Intercepting a OTA update * Downloading from the manufacturers web page * [eMMC tapping](https://www.exploitee.rs/index.php/Exploitee.rs_Low_Voltage_e-MMC_Adapter) * Unsoldering the SPI Flash / eMMC chip and reading it in a adapter | * Firmware contains a lot of useful information, like source code and binaries of running services, pre-set passwords, ssh keys etc. |
| **Manipulating the code execution flow of the device** | * JTAG / SWD interface * [Side channel attacks like glitching](https://wiki.newae.com/Main_Page) | * With the help of a JTAG adapter and gdb we can modify the execution of firmware in the device and bypass almost all software based security controls. * Side channel attacks can also modify the execution flow or can be used to leak interesting information from the device |
| **Obtaining console access** | * Serial interfaces (SPI / UART) | * By connecting to a serial interface, we will obtain full console access to a device * Usually security measures include custom bootloaders that prevent the attacker from entering single user mode, but that can also be bypassed. |
| **Insecure 3rd party components** | * Software | * Out of date versions of busybox, openssl, ssh, web servers, etc. |