Enhancing IoT Security: Focus on Weak Passwords and Insecure Default Settings

Rayavarapu Balaji

Andhra University College of Engineering, Department of Information Technology & Computer Applications,
Visakhapatnam, Andhra Pradesh, India
Email: *keats960[at]gmail.com*

Abstract: ***The rapid growth of the Internet of Things (IoT) has improved automation and connectivity but also introduced major security risks. This project focuses on two key vulnerabilities—weak passwords and insecure default settings—which are often overlooked but commonly exploited. These flaws allow attackers easy access through brute-force attacks or open system configurations. Real-world incidents like the Mirai botnet are examined to highlight the severity of these threats. The project also recommends best practices such as strong password enforcement, disabling unnecessary features, and secure configuration during deployment. The goal is to strengthen IoT security by addressing these foundational issues.***

Keywords: Internet of Things (IoT), Weak Passwords, Insecure Default Settings, IoT Security, Mirai Botnet, Secure Configuration

1. Introduction

The Internet of Things (IoT) represents a revolutionary advancement in modern technology by enabling interconnected devices to communicate, collect, and exchange data autonomously. This paradigm has significantly transformed various sectors such as smart homes, healthcare, transportation, and industrial automation, offering enhanced operational efficiency, convenience, and data-driven decision-making [1], [2]. However, this rapid adoption has also introduced a range of security challenges that demand urgent attention. Unlike traditional computing systems, IoT devices often operate with limited processing power, low memory, and minimal user interfaces, making it difficult to implement and maintain strong security protocols [4]. As a result, IoT ecosystems are increasingly vulnerable to a wide array of cyber threats. Among these, two of the most critical and commonly exploited vulnerabilities are weak passwords and insecure default settings [3], [5], [8]. Many IoT devices are shipped with default or hardcoded credentials that users rarely change, making them easy targets for brute-force attacks and credential stuffing [6], [10]. Additionally, insecure default configurations such as open network ports, unencrypted data transmission, and enabled debugging modes leave devices exposed to exploitation [3], [9]. These vulnerabilities are often introduced during the manufacturing or setup phase and persist due to lack of user awareness or negligence [7].

Notably, the Mirai botnet attack in 2016 demonstrated the devastating potential of these weaknesses, where hundreds of thousands of IoT devices with default credentials were hijacked to launch a massive Distributed Denial-of-Service (DDoS) attack [6]. Despite such incidents, many manufacturers continue to prioritize usability and time-to-market over security, while end-users often lack the technical knowledge to harden their devices against threats [1], [2].This project aims to analyze and address the impact of weak passwords and insecure default settings on IoT security. By examining real-world vulnerabilities, technical attack vectors, and case studies, the project will identify how these issues are exploited in practice. Furthermore, it will provide recommendations and best practices for improving device security—emphasizing the role of manufacturers, developers, and users in building a secure IoT environment [9], [10]. Through this research, the project advocates for a "security-by-design" approach, promoting the implementation of hardened security measures from the development phase onward. The ultimate goal is to contribute to a more resilient, secure, and trustworthy IoT ecosystem.

2. IoT Security Threats (OWASP)

The following are the security issues highlighted by the Open Web Application Security Project (OWASP):

- **Weak, Guessable, or Hardcoded Passwords:** Many IoT devices ship with default or hardcoded credentials like “admin/admin.” These are rarely changed by users and are often published online, making them a major target. Attackers exploit such weak authentication to gain unauthorized access and control of the device [1], [6], [10].
- **Insecure Network Services:** Devices frequently expose unnecessary or outdated services (e.g., Telnet, FTP) which can be exploited. Using secure alternatives (e.g., SSH, HTTPS) and disabling unused services greatly reduces exposure [1], [4], [5].
- **Insecure Ecosystem Services:** IoT ecosystems often rely on mobile apps, cloud APIs, or web dashboards. If these lack proper authentication, authorization, or input validation, attackers can bypass controls and manipulate devices remotely [1], [2], [8].
- **Lack of Secure Update Mechanism:** Many IoT products fail to offer secure and authenticated firmware updates. Without verification, malicious firmware can be pushed to compromise the device [1], [4], [9].
- **Use of Insecure or Outdated Components:** IoT systems often run on outdated OS versions or third-party libraries. These vulnerable components are entry points for exploitation if not patched regularly [2], [5], [8].
- **Insufficient Privacy Protection**: IoT devices collecting sensitive personal data (e.g., health, location, audio/video) may transmit it without encryption or consent mechanisms, violating user privacy [1], [2], [8].

DOI: https://dx.doi.org/10.21275/SR25729122027

- **Insecure Data Transfer and Storage:** Data is often transmitted over unencrypted channels (e.g., HTTP instead of HTTPS) or stored in plain text. This puts it at risk of interception, leakage, or tampering [1], [3], [8].
- **Lack of Device Management:** Without centralized control, it's difficult to monitor, update, or secure IoT fleets. Lack of visibility increases the window for exploitation [2], [4], [7].
- **Insecure Default Settings:** Devices are often deployed with insecure settings such as open ports, active debug modes, or overly permissive configurations. These defaults can easily be exploited if not hardened before use [1], [3], [7], [9].
- **Lack of Physical Hardening:** Physical access to IoT devices can allow attackers to bypass software controls, retrieve stored credentials, or modify firmware directly [1], [2], [4].

3. Dedicated to Vulnerability Analysis

3.1 Weak, Guessable, or Hardcoded Passwords

One of the most widespread and dangerous vulnerabilities in IoT systems is the use of weak or hardcoded passwords,[1] such as “admin” or “1234”, which are often left unchanged by users. [2],[10] These default credentials are well-known and publicly listed in hacker databases,[5] making it easy for attackers to exploit devices using methods like brute-force or credential stuffing attacks.[8] Because IoT devices typically run autonomously with little user oversight, a compromised device can be manipulated to steal data, disable operations, or serve as an entry point into larger networks.[4],[7] For instance, breaching a smart camera can lead to unauthorized access to other connected home devices.[3][6] A notable example is the Mirai botnet attack in 2016, where thousands of IoT devices with default credentials were hijacked and used to launch massive DDoS attacks, demonstrating the severe impact of weak password practices.[6] To address this threat, manufacturers should enforce password changes on first use, apply password complexity rules, and avoid embedding admin credentials in firmware. [1] Additional protections like two-factor authentication (2FA) and rate-limiting login attempts can significantly enhance device security.[2],[9]

Weak passwords in IoT systems present one of the most persistent and dangerous threats to device security. These credentials are often simple, predictable, or even hardcoded into the firmware—such as “admin/admin” or “root/1234”—and are commonly left unchanged by users. The ease with which attackers can exploit these weak credentials allows them to gain unauthorized access to IoT devices, especially when the devices are exposed to the internet or lack basic protections like account lockout or rate limiting [1], [6]. This vulnerability is particularly concerning in systems using controllers like ESP32 or Raspberry Pi, where web interfaces, Telnet, or SSH services may be running with default access credentials [3], [8]. Technically, attackers can launch automated brute-force attacks using tools like Hydra or custom Python scripts. They begin by scanning the network to detect open ports on devices like ESP32. Once a vulnerable login portal is found, these tools try hundreds or thousands of common username-password combinations in rapid succession. If a match is found, the attacker gains access to the controller's operating environment, allowing them to execute arbitrary commands, upload malicious scripts, or hijack the connected IoT functions, such as sensors, relays, or cameras [5], [10]. Attackers frequently exploit these openings not just to take control of a single device, but to pivot laterally across the network, compromising other connected systems.

A common real-world attack sequence begins with reconnaissance. Tools such as Nmap are used to identify exposed services on the device—like HTTP dashboards or Telnet. Once identified, brute-force attacks are launched using commonly available dictionaries containing millions of known credentials. If successful, attackers may escalate privileges, modify firmware, or integrate the device into larger botnets, as seen in historical incidents like the Mirai botnet attack [6]. These attacks can cause critical disruptions, particularly in environments where IoT devices are connected to physical infrastructure such as doors, lights, or surveillance equipment. To prevent such compromises, it is essential to embed security directly into the device firmware. Developers should enforce strong password creation, disable any unnecessary interfaces, and use encrypted protocols like HTTPS or secure MQTT. Additionally, mechanisms like rate limiting, IP blocking after repeated failed login attempts, and optional two-factor authentication significantly reduce the success rate of brute-force attempts [2], [7], [9].

Addressing weak password vulnerabilities is a shared responsibility. Device manufacturers must stop shipping products with default logins and enforce secure configuration practices. Developers need to integrate authentication safeguards during development, and end-users must be educated to change default credentials immediately upon installation. Only through such collaborative efforts can the IoT ecosystem become resilient to one of its simplest, yet most damaging attack vectors.

1) Weak passwords remain prevalent in IoT systems due to default or reused login credentials left unchanged by users.
2) Attackers use brute-force tools (e.g., Hydra, Python scripts) to target devices like ESP32 through exposed services.
3) Real-world execution includes scanning, brute-force login, credential reuse, and device compromise through firmware or sensor control.
4) Effective mitigations include strong password enforcement, encryption, disabling unnecessary ports, rate limiting, and 2FA.
5) Security must be addressed at all levels—by manufacturers, developers, and users—to reduce this widely exploited vulnerability.

3.2 Insecure Default Settings

Insecure default settings refer to weak or unsafe configurations that come pre-enabled when IoT devices are first deployed. [1],[2],[3],[9] These often include open network ports, active debugging modes, unsecured communication protocols (like HTTP instead of HTTPS),[3] unrestricted user access, and unused services that remain running.[5],[8] While these settings are meant to simplify setup,[4] they leave devices highly vulnerable to attacks if not reconfigured securely after installation.[2],[9]

Most users—especially non-technical consumers—either do not recognize these vulnerabilities or lack the skills to change them.[2] As a result, attackers can exploit these weak configurations to access devices, intercept data, or gain control remotely.[5] A common example is an IP camera accessible via an open, unencrypted port, allowing attackers to hijack video streams or modify settings.[6],[8],[3] These issues often stem from poor product design, where security is not prioritized by manufacturers.[1] Some devices may not even allow disabling risky services or modifying default configurations,[7] making them permanently insecure.[9] To mitigate this threat, manufacturers should follow a "secure by default" strategy—disabling unnecessary features, enforcing encryption, limiting permissions, and requiring users to review and set up key security options during initial configuration. [1],[2],[9] Additional measures like regular firmware updates, clear documentation, and user education are essential to maintain long-term device security.[4],[8]

Insecure default settings are one of the most overlooked yet dangerous security flaws in Internet of Things (IoT) systems. These vulnerabilities stem from manufacturer-provided configurations that prioritize convenience and ease of deployment over security. Devices are often shipped with open ports, unencrypted communication protocols, disabled authentication, and unnecessary services enabled by default, leaving them vulnerable from the moment they are powered on [1], [3]. Such settings are commonly found in microcontroller-based platforms like the ESP32, where development interfaces or web-based dashboards are left active without requiring any login, making them easily exploitable by attackers [4], [8]. A practical scenario can be seen in an ESP32-based temperature sensor module. If this device is accessible via an unsecured HTTP interface without authentication, any user connected to the same local network—or even remotely, if port forwarding is enabled—can directly access the web panel. From there, the attacker can read or manipulate sensor data, trigger device actions, or inject malicious JavaScript or shell commands. Since the device offers no protection by default, no hacking skills are required to cause significant damage, and the attacker may remain unnoticed [2], [5]. In typical attacks, cybercriminals first use tools like Nmap or Shodan to scan networks for devices with open ports such as HTTP (port 80) or Telnet (port 23). Once a device is discovered, they fingerprint its firmware or server headers to determine the specific model or microcontroller. If developer options or debugging interfaces like UART, OTA endpoints, or configuration panels are active, the attacker can connect to these and control or reprogram the device. Advanced attacks may involve placing scripts that persist across reboots, changing wireless credentials, or redirecting the device to malicious servers for data exfiltration [6], [7]. These security oversights are usually a result of manufacturers trying to make setup as seamless as possible. To reduce technical barriers for non-technical users, vendors often leave default interfaces wide open. At the same time, many end-users remain unaware that these insecure settings even exist. In some cases, developers may forget to disable test features or debugging tools before shipping, leaving these entry points available to anyone with network access [3], [9]. To prevent such vulnerabilities, secure configuration must become a design standard rather than an afterthought. Devices should ship with all unnecessary ports and services disabled. Users must be required to set secure passwords and confirm critical settings during the initial setup. All communication should take place over encrypted channels like HTTPS or secure MQTT, and APIs should require authentication tokens. Moreover, the firmware should be hardened for security and support over-the-air (OTA) updates, allowing vendors to patch vulnerabilities post-deployment [1], [2], [10]. Through these practices, the IoT ecosystem can avoid the recurring exploitation of insecure defaults and significantly reduce attack surfaces across smart devices.

1) Insecure default settings leave IoT devices vulnerable from initial deployment, due to open ports, no authentication, or unencrypted communication.
2) Attackers exploit these weaknesses using tools like Nmap and Shodan, identifying and accessing unsecured services for device takeover.
3) Microcontroller-based devices such as ESP32 are especially at risk, where developer/debugging interfaces are often left exposed.
4) These vulnerabilities result from manufacturer shortcuts, user ignorance, and development oversights, prioritizing ease of use over security.
5) Mitigation involves disabling unnecessary services, enforcing secure setup steps, using encryption, and providing hardened firmware with OTA update support.

4. Conclusion

As IoT continues to integrate into all aspects of modern life, securing connected devices is crucial. This project focused on two major vulnerabilities—weak passwords and insecure default settings—which are commonly exploited entry points for attackers. These flaws can lead to data breaches, device hijacking, and large-scale attacks like botnets. The project stresses the need for secure-by-default practices, urging both manufacturers and users to take responsibility. Key measures include strong authentication, enforced password policies, disabling unnecessary services, and secure firmware updates. In conclusion, addressing these basic but critical security flaws is essential to building a safer and more trustworthy IoT ecosystem.

References

[1] OWASP, "OWASP IoT Top 10," OWASP Foundation, 2024. [Online].
[2] ENISA, "Baseline Security Recommendations for IoT," European Union Agency for Cybersecurity, 2022.
[3] McAfee, "The Hidden Dangers of Default Settings in IoT," McAfee Research, 2023.
[4] Palo Alto Networks, "The IoT Security Landscape: Risks and Remedies," 2023.
[5] Trend Micro, "Insecurity in the Internet of Things," White Paper, 2023.
[6] B. Krebs, "KrebsOnSecurity Hit With Record DDoS," 2016. [Online]. Available: https://krebsonsecurity.com
[7] CERT/CC, "Vulnerabilities in Consumer-Grade IoT Devices," 2024. [Online]. Available: https://www.kb.cert.org/

[8] Symantec, “Internet of Things: Security vulnerabilities and threats,” Symantec Labs, 2023.

[9] IEEE, “Secure Deployment of IoT Systems Using Hardened Defaults,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2122-2131, 2022.

[10] S. Lee and K. Chen, “Brute-Force and Default Credentials in Smart Devices,” *ACM Computing Surveys*, vol. 55, no. 3, 2022