ads-tec Industrial IT GmbH

IRF1000 IRF2000 IRF3000

Application Note – HTTP API Migration Guide

© ads-tec IIT GmbH 2024

Version	Date	Editor	Changes
1.0	02.12.2024	SnPr	Creation

Table of Contents

1 Introduction 2
2 Migration 3
2.1 Classic HTTP API to JSON/RPC API 3
2.2 JSON/RPC HTTP vs HTTPS 3
2.3 HTTPS Server certificates and Device CA 3
2.3.1 Secure Usage of the Self-Signed Device CA 3
2.3.2 Important Security Considerations 3
2.4 Passwords 3
2.5 Overview of operation modes 4
2.5.1 Compatibility mode 4
2.5.2 Secure default mode 4
3 Implicated Security Risks on usage of the Classic HTTP API 4
3.1 Passwords 4
3.1.1 Security Concerns with MD5 Hashes 4
3.2 Key Issue with HTTP Digest Authentication and Browser Behavior 5

1 Introduction

Starting with version 2.0.0 for the IRF1000 and IRF3000, and version 5.0.0 for the IRF2000, the ads-tec Classic HTTP API is hereby officially declared as deprecated. Consequently, this API will be disabled by default in the factory default settings of these devices.

Furthermore, all future ads-tec IIT products will no longer include the Classic HTTP API. Additionally, new software features introduced for these devices may not be compatible with the deprecated API.

However, the Classic HTTP API will remain available and supported within its existing range of functionality for the current IRF1000, IRF2000, and IRF3000 product lines until end of support for the software of these devices.

The first-generation ads-tec firewall router, the IF1000, has officially reached the end of its software support lifecycle and will no longer be considered for updates or further development. It remains on Classic HTTP API and customers are advised to replace the devices.

Both APIs have been in use for a considerable time, allowing you to easily determine which API you are working with:

- **URLs with payloads directed to** https://<IP>/rpc **represent the new JSON/RPC API.**
- **URLs with payloads directed to** http://<IP>/priv/script/php/... **fall under the scope of this migration guide.**

This distinction helps ensure clarity and aids in following the appropriate documentation for each API.

2 Migration

2.1 Classic HTTP API to JSON/RPC API

The successor of the Classic HTTP API is the JSON/RPC device which is already available for all software versions of IRF1000 and IRF3000 and for IRF2000 starting from version 2.5.0. Please see the Application note “IRF3000 IRF2000 IRF1000 Application Note - ads-tec JSON RPC API specification” for more details.

2.2 JSON/RPC HTTP vs HTTPS

Before versions 2.0.0 of IRF1000/IRF3000 and version 5.0.0 the JSON/RPC API was available on HTTP and HTTPS. The support for HTTP is now disabled due to security concerns. Please contact ads-tec IIT Support in case you really need that mode of operation.
The default behavior for HTTP access to the devices is to redirect all requests to HTTPS by using Strict Transport Security (HSTS).

The standard ads-tec web interface and JSON/RCP API will enforce HTTP even if you run the device with explicit enabled Classic HTTP API.
Only the Classic HTTP API endpoints http://<IP>/priv/script/php/get.php and http://<IP>/priv/script/php/get.php will not get HTTPS HSTS redirects if they are enabled.

2.3 HTTPS Server certificates and Device CA

Starting with versions 2.0.0/5.0.0, devices will automatically generate a new self-signed device CA and an HTTPS web server certificate based on the current DNS and IP configuration. If no custom HTTPS web server certificate has been uploaded, the devices will default to using this newly generated certificate.

2.3.1 Secure Usage of the Self-Signed Device CA

The self-signed device CA is stored as a persistent file within the device and is designed to remain intact during normal operation. It will not be deleted or altered, even if the device is reset to factory defaults. Each device generates its own unique CA, which includes the ads-tec IIT serial number (e.g., starting with "AX...").

If desired, you can download and import the public key of the device CA into your browser or add it to your software as a trusted certificate authority for this specific device.

2.3.2 Important Security Considerations

- **Trust the Connection**: Import the device CA only if you fully trust the connection. For example, this is advisable if the device is physically near you, such as on your desk, with a direct Ethernet connection to your PC.
- **Certificate Pinning**: Even if you are not fully confident about the initial connection’s security, you can use this feature for certificate pinning. By trusting the certificate during setup, you can ensure that any future compromise will be detected, as changes to the certificate will raise alerts.
- **Device CA security level**: The device CA is (right now) not generated or stored by any form of secure element or hardware security module. Please switch to custom uploaded certificates if you have requirements for this security level.
- **Browser Behavior:** Most browsers will treat the Device CA as a fully authoritative source, allowing it to issue certificates for any URL unless explicitly restricted. This means the CA can potentially authorize certificates for unintended domains. Addressing this requires solutions tailored to your specific setup, such as leveraging Name Constraints, DNS CAA records, or custom policies, to enforce domain-level restrictions effectively.

This approach balances convenience with security, enabling robust usage while maintaining a high level of trustworthiness for your device's operations.

2.4 Passwords

Prior to versions 2.0.0/5.0.0, device passwords were stored as MD5 digest hashes, as the Classic HTTP API relied on HTTP Digest Authentication as its authentication mechanism.

Starting with the new versions, factory default settings now store passwords using the Argon2 hashing algorithm, offering improved security.

This change impacts the various modes of operation. Specifically, if the device is switched between compatibility mode with the Classic HTTP API and the new JSON/RPC API (or vice versa), all passwords must at least be reentered (or better rotated!) to ensure proper functionality. For assistance with integrating this switch into your software, please contact ads-tec IIT support. We provide ready-to-use code samples to facilitate the migration process. You can use the Password migration Wizard of the ads-tec web interface for conversion of single devices.

2.5 Overview of operation modes

2.5.1 Compatibility mode

Devices running on > 2.0.0/5.0.0 with settings configured with previous versions.
The device will run in compatibility mode:

- Passwords are still stored as htdigest.
- JSON/RPC is on HTTPS only
- Classic HTTP API is enabled on HTTP and HTTPS.

2.5.2 Secure default mode

Devices running on > 2.0.0/5.0.0 with factory defaults:

- Passwords are still stored as Argon2 hashes
- JSON/RPC is on HTTPS only
- Classic HTTP API is disabled.

3 Implicated Security Risks on usage of the Classic HTTP API

There are several security risks to consider if you choose to continue using the Classic HTTP API. Whether these risks are purely theoretical or have practical implications depends on your specific setup and threat model.

In many scenarios, devices are administered exclusively through secure channels, such as Big-LinX VPN or similar VPN solutions. Additionally, local API access often occurs over trusted Ethernet connections within a closed, secured cabinet. These factors can significantly mitigate the criticality of the risks, making them less relevant to your particular use case.

However, it is essential to evaluate your unique environment carefully to ensure the continued secure operation of the device.

3.1 Passwords

Passwords are stored as MD5 hashes, which present several potential vulnerabilities to attackers:

1. **Storage on the Device**:
   MD5 hashes are stored unencrypted on the device's flash chip. An attacker with professional equipment could physically remove the chip and extract its contents.
2. **Configuration Files**:
   The hashes are also included in the ".cf2" configuration files. While these files are protected by an additional encryption layer, the encryption key is shared across all ads-tec IIT firewall devices to enable configuration file interchange, making this protection less robust.
3. **Network Exposure**:
   Passwords are transmitted as HTTP MD5 digest hashes during usage, whether over HTTP or HTTPS. If HTTPS is used without proper certificate validation, the hashes may still be exposed to attackers.

3.1.1 Security Concerns with MD5 Hashes

MD5 hashes can be cracked easily with modern hardware, especially if passwords are short or weak. To mitigate these risks, it is strongly recommended to:

- **Use Long Passwords**: Always use the maximum allowable password length to increase resistance against brute-force attacks.
- **Rotate Passwords Regularly**: Periodically update passwords to reduce the risk of exposure.
- **Avoid Reusing Passwords**: Ensure each device has a unique password to prevent cascading vulnerabilities across multiple devices.

These practices will substantially improve the security of your devices and help mitigate the risks associated with using MD5 hashing in the short term. However, it is essential to develop and implement a migration plan to transition to the new API and adopt Argon2 hashes, which provide a more secure and modern approach to cryptographic hashing.

3.2 Key Issue with HTTP Digest Authentication and Browser Behavior

Browsers typically store the HTTP Digest authentication session internally, which results in automatic authentication for all URLs linked to the device while the session is open. This behavior introduces a significant security risk:

- Automatic Execution of Embedded Commands: If a malicious email or webpage contains URLs with embedded commands targeting the device's API, the browser will automatically authenticate and execute these commands without user intervention – if the session is open in another browser tab or windows.
- Vulnerability to Cross-Site Attacks: Attackers can exploit this weakness by embedding commands in emails or web pages, tricking users into unintentionally executing unauthorized actions on their devices.

© ads-tec Industrial IT GmbH 2024