API - Webhook

- [Webhook Implementation Documentation](#)

Webhook Implementation Documentation

Introduction

This document provides technical details for integrating your system with our webhook service. The webhook operates over SSL and requires a bearer token for authentication. Once set up, it will send all scan results to your specified endpoint in real-time.

Table of Contents

Step-by-Step Implementation within ReConfirm Dashboard

1. Configuring the Webhook URL

1. Go to the system settings in your ReConfirm platform.
2. Navigate to the **Webhooks** section.
3. Click on **Add New Webhook**.
4. Enter the URL where you want to receive the scan results. This should be the endpoint on your system that will process the incoming JSON data.
5. Create a secure bearer token for authentication.
6. Click **Test Connection** to verify the setup.
   - The system will send the latest scan results to your endpoint. Ensure that the data is received and processed correctly.
7. Save the token securely; it will be required for authenticating webhook transmissions.

2. Adding the Webhook to a Scan Profile

1. Go to your **Scan Profiles** section.
2. Edit the profile for which you wish to activate the webhook.
3. In the profile settings, select the desired webhook under **Notification Options**.
4. Save your changes or start the scan.

Multiple Webhook Configurations

You can configure multiple webhooks for different customers or scan types. To do so, set up a separate webhook in the system settings for each customer or scan type. Repeat the steps in the **Configuring the Webhook URL** section above for each webhook configuration.

Webhook Data Format

The webhook sends all findings from each scan in a structured JSON format. An example of this format is provided in the **Demodata JSON Output** file.

Integration:

Overview

Our webhook service enables seamless integration by sending scan results directly to your application's endpoint. Communication is secured using SSL/TLS encryption, and access is controlled via a bearer token that you provide.

Prerequisites

- HTTPS Endpoint: An SSL-enabled endpoint capable of receiving HTTP POST requests.
- Bearer Token: A secure token for authenticating requests (you must generate and provide this token).
- JSON Parsing: Ability to parse JSON-formatted data in your application.

Authentication

All webhook requests include a bearer token in the Authorization header for security.

- Header Format:

  ```
  Authorization: Bearer YOUR_BEARER_TOKEN
  ```

- Replace "YOUR_BEARER_TOKEN" with the token you have generated.

Setting Up the Webhook

1. Provide Endpoint URL: Supply us with your HTTPS endpoint URL where you wish to receive the scan results.
2. Generate Bearer Token: Create a secure, random token for authentication purposes.
3. Configure Firewall: Ensure that your server accepts incoming connections from our IP addresses (contact support for the IP range if necessary).
4. Test the Connection: Verify that your endpoint correctly handles incoming POST requests.

Data Format

Scan results are sent as JSON in the body of a POST request.

- Example Request:

```
POST /your-endpoint HTTP/1.1
Host: yourdomain.com
Content-Type: application/json
Authorization: Bearer YOUR_BEARER_TOKEN

{
  "scanId": "abc123",
  "status": "completed",
  "results": {
    "threatsFound": 0,
    "details": []
  },
  "timestamp": "2023-10-23T14:55:00Z"
}
```

- JSON Fields:
  - scanId: Unique identifier for the scan.
  - status: Current status of the scan ("completed", "in_progress", "failed").
  - results: Object containing the scan results.
    - threatsFound: Number of threats detected.
    - details: Array with detailed information on each threat.
  - timestamp: ISO 8601 timestamp of when the scan was performed.

Security Considerations

- SSL/TLS Encryption: All data is transmitted over HTTPS to ensure encryption in transit.
- Bearer Token Confidentiality: Keep your bearer token secure; do not expose it in client-side code or logs.
- Input Validation: Validate and sanitize all incoming data on your server.
- Rate Limiting: Implement rate limiting if necessary to protect against denial-of-service attacks.

Error Handling

- Successful Processing:
  - Return HTTP status code `200 OK`.
  - Optionally, include a JSON response acknowledging receipt.

    ```
    {
      "message": "Scan results received successfully."
    }
    ```

- Authentication Failure:
  - Return HTTP status code `401 Unauthorized`.
  - Include an error message indicating invalid authentication.

- Other Errors:
  - Return appropriate HTTP status codes (`400 Bad Request`, `500 Internal Server Error`, etc.).
  - Include a JSON object with an error field describing the issue.

Webhook Implementation Guide for ReConfirm

This document provides technical guidance for integrating ReConfirm's Webhook API to receive scan results at a specified URL. This setup allows you to automatically integrate scanned data into your own systems, such as SIEM platforms or other analytical tools.