JSON Profiles

https://campus.barracuda.com/doc/99616857/

JavaScript Object Notation (JSON) security performs deep inspection of incoming packets/requests for web applications that use the JSON protocol to exchange data over HTTP. Many applications including mobile applications exchange data with the servers using JSON (RFC 4627), which is a lightweight data-interchange format. JSON-based applications can be attacked in multiple ways, such as sending data in an improper format or embedding attack vectors in the data. It is important for applications using the JSON format to validate the inputs before being processed. The Barracuda WAF-as-a-Service enforces input validations and other security checks to ensure that attacks are not tunneled inside HTTP requests with the JSON content.

Example:

Application Type	Content-Type	Payload
Traditional Application with the HTML Form	application/x-www-form-urlencoded	First-name=John&Last-name=Peter&email=john@fastmail.com
Rest API based Application	application/json	{“First-name”:”John”,”Last-name”:”Peter”,”email”:”john@fastmail.com”}

The JSON key-value pairs in the request body require the same level of input validation as the URL query parameters.

When an application is created, a default JSON profile is automatically created by the system for that application. the default JSON profile is applicable to the entire URL space of the application. You can create multiple JSON profiles for different URL spaces within the application.

To Add a JSON Profile

Perform the following steps to add a JSON profile:

1. On the WAF-as-a-Service web interface, go to the **APPLICATIONS** tab and select the application to which you want to add the JSON profile.
2. Click **APP PROFILES** in the left panel.
3. On the **Application Profiles** page:
    1. Select the URL, scroll down to **JSON profile** under **Add new** in the right panel, and click the **+** icon next to it.
4. In the **Add new JSON profile** section:

1. **Status** - Set to **Enabled** to enforce checks on requests using the JSON profile.
2. **Block Attacks** - Set the mode for the JSON profile.
    1. **Enabled**: If the request violates the configured JSON profile settings, it is blocked and logged under **Web Firewall Logs**.
    2. **Disabled**: Requests are validated against the JSON profile settings and allowed to pass through, but logs the request errors under **Web Firewall Logs**.
3. **Validate Key**: Set to Enabled to enforce validation on the keys in the JSON request.
4. **JSON Policy** - Select the JSON policy to be associated with the profile. See To Add a JSON Policy.
5. **Ignore Keys** - Specify the JSON keys to be exempted from JSON security checks. This is an exact match; a wildcard is not supported, that is, a value with "*" does not work like a wildcard.
6. **Inspect Content Types** - Specify the Content-Type headers to be matched in the request to apply the JSON profile.
7. Click **Add**.

To Add a JSON Key Profile

A JSON key profile is used to validate keys present in JSON requests. Multiple key profiles can be configured with different JSON key validation settings, and associated with the JSON profile.

1. On the WAF-as-a-Service web interface, go to the **APPLICATIONS** tab and select the application to which you want to add the JSON key profile.
2. Click **APP PROFILES** in the left panel.
3. On the **Application Profiles** page:
    1. Select the JSON profile, select the **Parameters** tab in the right panel, and click **Add new**.
    2. In the **Add new** section, scroll down to **JSON key** and click the **+** icon next to it.
    3. In the **Add new JSON key** section:
        1. **Status** - Set to Enabled to match the JSON key profile with the JSON request.
        2. **Name** - Specify a name for the JSON key profile.
        3. **Value Type** - Select the type of value associated with the specified key. The value can be String, Array, Number, Object, or Any.
            1. **Any** - Use this when the value of a key has different data types.
            2. **String** - Use this when the value of a key has plain text characters to form a word.
            3. **Array** - Use this when the value of a key contains an array of values.
            4. **Number** - Use this when the value of a key has integer/numeric characters.
            5. **Object** - Use this when the value of a key contains an array of name/value pairs.
        4. **Max Length** - Define the maximum allowable length for text characters (string) in the key. This setting is available when **Value Type** is set to **Any** or **String**.
        5. **Max Number Value** - Define the maximum allowable number in the key. This setting is available when **Value Type** is set to **Any** or **Number**.

6. **Allow Null** - Set to **Yes** to allow keys with NULL value.
7. **Value Class** - Select a value class to be compared to the key values sent in JSON requests/responses.
8. **Base64 Decode** - Set to **Yes** to apply base64 decoding to the key values. After the decoding is successful, other checks are enforced according to the key profile settings.
9. **Allowed Metacharacters** - Define the meta-characters to be allowed in spite of being marked as denied in **Value Class**.
10. Click **Add**.

Example: **JSON Keys**

```
{
"Id": 10012,
"Title": "Developer",
"Name":"Romin",
"region":"CA",
"emailAddress":"secureyourjourney@barracuda.com",
"address": "#102, 2cross MG road US \n cross",
"Contact": {"Personal": 1234567890, "Office": 9876543210},
"Skills": ["MS Office", "Python", "Java", "C", "C++"]
}
```

The following table displays the value type for the JSON keys.

JSON Key/Value Pair	Value Type
"Id": 10012	Number
"Title": "Developer"	String
"Name":"Romin"	String
"region":"CA"	String
"emailAddress":"secureyourjourney@barracuda.com"	String
"address": "#102, 2cross MG road US \n cross"	Any
"Contact": {"Personal": 1234567890, "Office": 9876543210}	Object
"Projects": ["Container", "AKS", "EKC", "GKE"]	Array

To Add a JSON Policy

You can define a JSON policy to enforce security checks on JSON requests. Associate the defined policy with the JSON profile to validate the requests based on the policy settings.

1. On the WAF-as-a-Service web interface, go to the **APPLICATIONS** tab and select the application to which you want to add the JSON policy.
2. Click **JSON SECURITY** in the left panel.
3. On the **JSON Security** page, click **Add Policy**.
4. On the **Add JSON policy** window:
    1. **Name** - Enter a name for the JSON policy.
    2. **Max Keys** - Enter the maximum allowable keys in the JSON request.
    3. **Max Key Length** - Enter the maximum allowable length for JSON key names.
    4. **Max Value Length** - Enter the maximum allowable length for JSON value. This is applicable only for the datatype "String".
    5. **Max Number Value** - Enter the maximum allowable value for the JSON datatype "Number".
    6. **Max Object Depth** - Enter the maximum allowable depth for nested JSON structure.
    7. **Max Array Elements** - Enter the maximum allowable number of elements in an array.
    8. **Max Siblings** - Enter the maximum allowable number of elements in a single JSON object.
    9. Click **Add**.