verifone

# PayPal Fraudnet

## Overview

This tutorial guides you to use FraudNet, a JavaScript library from PayPal to be embedded into your web page. Its purpose is to collect browser-based data to help reduce fraud. Upon checkout, these data elements are sent directly to PayPal Risk Services for fraud and risk assessment.

To integrate FraudNet into your web application, follow these steps:

1. Get the Tracking ID to be used in further steps. Refer PayPal Risk Analysis for this.
2. Embed a FraudNet JavaScript or NoScript snippet into your web page.
3. Pass a Tracking ID (also known as the `correlationId`) to the FraudNet Session Identifier `f` variable used by the `JavaScript` and `noscript` tags. This enables the FraudNet JavaScript to post data asynchronously by using the Session Identifier `f`.
4. Pass the above Tracking ID to Verifone (via the `paypalFraudId` header) in the backend. This enables PayPal Risk to pull data that the FraudNet JavaScript stores.

The bulk of the integration code is based on the non-blocking script loader pattern described below. There are three parts to the integration:

- `script/` element used as a parameter block for passing input parameters to FraudNet
- `script/` element with code for asynchronously loading the FraudNet JavaScript
- `noscript/` element if JavaScript is not enabled for the application

## Content Security Policy integration

#### CSP tags

If you are using Content Security Policy (CSP), you must allowlist the following URLs in CSP:

| Tag | Attribute (Live) |
|---|---|
| `img-src` | `https://c.paypal.com`, `https://b.stats.paypal.com` |
| `frame-src` | `https://c.paypal.com` |
| `script-src` | `https://c.paypal.com` |

#### CSP scripts

If your Content Security Policy (CSP) does *not* allow inline scripts, you may use one of the following options:

- Add `unsafe-inline` as a directive in your `script-src policy`, such as `Content-Security-Policy: script-src 'unsafe-inline'`. This allows access to all inline resources throughout your app.
- Implement a nonce value to allowlist the script.

#### Allowlist inline scripts

You can allowlist specific inline scripts without using the `unsafe-inline` directive, by using either a cryptographic nonce (a number used once) or an SHA hash.

To use a nonce, add a nonce attribute in the script tag. You must generate a nonce at random *with each page load* and insert it into the CSP and the FraudNet script. PayPal recommends encoding a nonce value in Base64 using a cryptographically secure random number generator with at least 128 bits of data.

Nonce example:

```
<script nonce=abcRANDOM_NONCE_VALUExyz>
alert('Hello, world.');
</script>

Content-Security-Policy: script-src 'nonce-abcRANDOM_NONCE_VALUExyz'
```

Alternately, you can create an SHA hash of the script itself (without its tags), and place that value in the CSP `script-src`.

```
<script>
alert('Hello, world.');
</script>

Content-Security-Policy: script-src 'sha256-abc_hash-of-MixEd1-CaSE2&numS_xyz='
```

## Add a JavaScript code block

The block below should work on any modern browser that has JavaScript enabled.

This JavaScript passes parameters to FraudNet. All FraudNet parameters except parameter `s` and parameter `f` are optional.

The `fncls` attribute is mandatory, and its value must be `fnparams-dede7cc5-15fd-4c75-a9f4-36c430ee3a99`. To find and process parameters, FraudNet JavaScript searches for a script of type application/json with an attribute `fncls`, and its value match that string.

**Copy and update the following code snippet into the page where you are integrating FraudNet.**

```
<script type="application/json" fncls="fnparams-dede7cc5-15fd-4c75-a9f4-36c430ee3a99">
{
"f":"change_this_to_32char_guid", // Tracking Id
"s":"flowid_provided_to_you" // unique for each web page, will be provided by PayPal directly
}
</script>
```

**There are two options for passing the data:**

```
// Option 2: Or, run FraudNet after your logic by appending it
// pass your configuration as options: { fnUrl: "https://c.paypal.com/da/r/fb.js" }
function _loadBeaconJS(options) {
```

```
var script = document.createElement('script');
script.src = options.fnUrl;
document.body.appendChild(script);
}
```

**Add a noscript code block:**

```
<noscript>
<img src="https://c.paypal.com/v1/r/d/b/ns?f=change_this_to_32char_guid&s=flowid_provided_to_you&js=0&r=1" />
</noscript>
```

## Data collection, usage, and privacy

Data collected by FraudNet is used for risk analysis and authentication. PayPal does not share FraudNet data with third parties for their own independent benefit.

Please note that FraudNet is for desktop browsers only. For risk analysis data gathered on mobile devices, please refer to Magnes documentation.