EUROPEAN COMMISSION

DIRECTORATE-GENERAL INFORMATICS

WiFi4EU Policy Enforcement Component

Implementation Guide

Contents

1. Introduction

Any network participating in WiFi4EU should comply with the conditions stated in the applicable Grant Agreement and annexes. This implies compliance with certain requirements regarding EU visual identity, usage and network quality. The European Commission is verifying the compliance of the participating networks by providing a "Policy Enforcement Component", also known as "the snippet".

The snippet is a piece of JavaScript code that needs to be integrated in the captive portal page of the participating network. It verifies the compliance of the captive portal in terms of visual identity, counts the number of users and measures connection speed and latency. These monitoring data is then collected by the WiFi4EU monitoring platform where it is used to monitor the operational state and usage, allowing financial control prior to voucher payment or potential recovery in case of non-compliance.

In order to validate if a captive portal is compliant, it is possible to use the snippet in a "self-test" mode which will provide detailed feedback regarding validation of the portal and snippet installation. The data of the self-test is not collected by the WiFi4EU monitoring platform.

2. Browser support

The snippet is written in plain vanilla JavaScript and works on all modern and widely used web browsers.

3. Data collected

The WiFi4EU monitoring platform gathers following data:

- Network identifier
- Page load time metrics
- Validation of installation of the snippet
- Portal validation results
- Network metrics test (once every n page loads)

The WiFi4EU monitoring platform does not collect any personal user data.

4. Installation of the snippet

4.1 Including the snippet in the portal page

The following code should be put in top of the `<head>` tag of the portal page:

```
<script type="text/javascript">

        var wifi4euTimerStart = Date.now();

        var wifi4euNetworkIdentifier = 'NETWORKIDENTIFIER';

        var wifi4euLanguage = 'PORTALLANGUAGE';

        </script>
<script type="text/javascript" src="http://52.174.44.162:8080/euwifiops/snippet/dist/wifi4eu.min.js"></script>
```

To obtain correct metrics it's important that the code is put at the top of the `<head>` tag.

4.2 Providing the network identifier

A network identifier must be provided as a variable in the `<head>` of the portal page, e.g. 'network-200':

```
<script type="text/javascript">

        var wifi4euTimerStart = Date.now();

        var wifi4euNetworkIdentifier = 'network-200';

        var wifi4euLanguage = 'EN';

</script>
<script type="text/javascript" src="http://52.174.44.162:8080/euwifiops/snippet/dist/wifi4eu.min.js"></script>
```

The provided `wifi4euNetworkIdentifier` should be of type string and should correspond to the 'url of the captive portal' as declared in the Installation Report.

4.3 Specifying the portal language

Also the `'PORTALLANGUAGE'` placeholder should be replaced with the correct language code in which the content of the portal page is served. The language code should be one of the 24 predefined language codes:

Language	Code	Language	Code	Language	Code	Language	Code
Bulgarian	BG	Estonian	ET	Irish	GA	Portugese	PT
Croation	HR	Finnish	FI	Italian	IT	Romanian	RO
Czech	CS	French	FR	Latvian	LV	Slovak	SK
Danish	DA	German	DE	Lithuanian	LT	Slovenian	SL
Dutch	NL	Greek	EL	Maltese	MT	Spanish	ES
English	EN	Hungarian	HU	Polish	PL	Swedish	SV

(1) Language code according to the official Interinstitutional Style Guide

For example, if the portal page content is served in English then the language code is `'EN'`.

4.4 Installation of the WiFi4EU logo

In order to be compliant with WIFI4EU requirements the portal page should contain a specific placeholder that will be used by the snippet to insert a logo image. The snippet loads the correct logo according to the language setting as defined by the `wifi4euLanguage` parameter.

The logo placeholder should:
- Have `wifi4eulogo` as HTML id for the `img` element
- Have original width and height dimensions
- Not be overlapped by another element
- Be visible
- Be in viewport upon loading
- Have no opacity applied

To install the logo insert following code in your portal page HTML:
```
<img id="wifi4eulogo">
```

4.5 Sizing and positioning of the WiFi4EU logo

In order to be compliant with WIFI4EU requirements the logo should always be visible upon pageload. Concerning the size of the logo the original aspect ratio should be respected.
On different viewport sizes there are different requirements for the logo:
- On viewports smaller than 600px wide: The logo should take up at least 90% of the viewport width
- On viewports from 600px and wider: The logo should take up at least 50% of the viewport width

Sizing and positioning should be applied on the logo placeholder using css.

4.6 Installation of the banner text placeholder

In order to be compliant with WIFI4EU requirements the portal page should contain a specific placeholder that will be used by the snippet to insert a banner text. The snippet sets the banner text according to the language setting as defined by the `wifi4euLanguage` parameter.

To install the banner text placeholder include following code in your portal page HTML:
`<p id="wifi4eutext"></p>`
The banner text should be visible in the browser upon page load without requiring scrolling on the page.

5. Self-test modus

A WiFi4EU supplier can test if its captive portal is compliant by enabling the self-test modus. When running in self-test modus:

- No data is sent to the Wifi4EU monitoring platform
- No metrics are calculated
- Validation results are shown in the developer console of your browser.
- Visual warning is shown on the portal page indicating that the snippet is running in self-test modus

In addition to the network identifier and language variables, the `var selftestModus = true;` should be added in the captive portal page.

Example of installing the snippet in self-test modus:

```
<script type="text/javascript">
        var wifi4euTimerStart = Date.now();

        var wifi4euNetworkIdentifier = 'network-200';

        var wifi4euLanguage = 'EN';

        var selftestModus = true;
</script>

<script type="text/javascript" src="http://52.174.44.162:8080/euwifiops/snippet/dist/wifi4eu.min.js"></script>
```

This is an example of the developer console when the snippet is running in self-test modus:

```
--- Policy Enforcement Component selftest modus activated ---
--------------------------------------------------------------------------------
--- Validating snippet installation ---
--- Snippet timer correctly installed. ---
--- Snippet language correctly installed. ---
--- Snippet networkIdentifier is correctly installed ---
--- Validating snippet installation end ---
--------------------------------------------------------------------------------
--- Validating portal compliance ---
--- Validating WiFI4EU logo ---
--- Viewport larger than 600px detected, make sure to repeat this test in a smaller viewport ---
srcCheck: true
existsCheck: true
typeCheck: true
aspectRatioCheck: true
widthToViewportCheck: false
opacityCheck: true
visibilityCheck: true
overlapCheck: true
completelyInViewPortAfterLoadingCheck: true
--- => Logo requirements check FAILED => non-compliant portal. ---
--- Validating WiFI4EU Text ---
contentCheck: true
existsCheck: true
typeCheck: true
visibilityCheck: true
overlapCheck: true
--- Validating portal compliance end ---
--------------------------------------------------------------------------------
--- Summary ---
--- The Policy Enforcement Component is correctly installed. ---
--- The portal is NON-compliant with WIFI4EU regulations. ---
--------------------------------------------------------------------------------
```
.

The self-test modus should only be used for self-validation of the portal. After finishing the self-test, remove the `var selftestModus = true;` setting.

5.1 Snippet installation validation

Following validations are performed to assess the correct snippet installation:
- Is the variable `var wifi4euTimerStart = Date.now();` initialised before loading of the snippet? This is crucial for gathering metrics about the page-loading process.
- Is the variable `var wifi4euLanguage = xx;` initialised before loading of the snippet and does the specified language code match one of the 24 predefined language codes?
- Is the `wifi4euNetworkIdentifier` correctly specified? This will be used to identify the incoming data in the WiFi4EU monitoring platform.

5.2 Logo validation

The results of the validations are displayed as `true` or `false`. True means that the validation has successfully passed.
Example:

```
existsCheck: true

typeCheck: true

srcCheck: true

aspectRatioCheck: true

viewportWidthCheck: true

opacityCheck: true

visibilityCheck: true

overlapCheck: true

completelyInViewPortAfterLoadingCheck: true
```

Self-test modus returns following parameters regarding the logo validation:

Parameter	Use
srcCheck	Set to true if the injection of the logo src attribute in the placeholder was successful.
existsCheck	Set to true if an HTML element with the attribute wifi4eulogo is present on the page.
typeCheck	Set to true if the element identified by wifi4eulogo is of type img.
aspectRatioCheck	Set to true if the element identified by wifi4eulogo has the correct width to height ratio.
viewportWidthCheck	Set to true if the element identified by wifi4eulogo has the correct width relative to the viewport.
opacityCheck	Set to true if the element identified by wifi4eulogo has opacity set to 1.
visibilityCheck	Set to true if the element identified by wifi4eulogo is visibile in the viewport upon page load.
overlapCheck	Set to true if the element identified by wifi4eulogo is not overlapped by another element.

5.3 Banner Text validation

The results of the checks are displayed as true or false. True means that the check has been passed successfully.
Example:

```
existsCheck: true

typeCheck: false

visibilityCheck: true

overlapCheck: true

contentCheck: false
```

Self-test modus returns following parameters regarding banner text validation:

Parameter	Use
contentCheck	Set to true if the injection of the banner text in the placeholder was successful.
existsCheck	Set to true if an HTML element with the attribute wifi4eutext is present on the page.
typeCheck	Set to true if the element identified by wifi4eutext is of type p.

visibilityCheck	Set to true if the element identified by `wifi4eutext` is visibile in the viewport upon page load.
overlapCheck	Set to true if the element identified by `wifi4eutext` is not overlapped by another element.

6. Walled garden

A walled garden consists of web resources accessible before a user has authenticated using the portal. In order for the snippet to function properly following addresses should be whitelisted in the walled garden:
http://52.174.44.162:8080

This must be configured on the NAS providing the hotspot functionality.

7. Troubleshooting

Potential Problem:
The snippet is properly installed in the `<head>` of the portal but is not getting loaded. This can be detected by going to network tab in the developer tools of the browser.

Solution:
Please check if the domain on which the snippet is hosted is available when the user is not logged in yet. This can be achieved by adding the domain to the whitelist or walled garden. The location where the snippet is hosted should be accessible before being logged in and receiving full internet access.

Potential Problem:
The snippet is properly installed and loaded but displaying self-test validation results or pushing validation results takes a lot of time.

Solution:
Remove all slow-loading assets from the website. The snippet is triggered by the `window.onload` event fired by the browser when all assets (images, files, ..) of the portal `HTML` document are loaded.