

[skip to content](#)[Research Blog](#) [Advisories](#) [CodeQL Wall of Fame](#) [Resources](#) [Events](#) [Get Involved](#)

- [Resources](#)
- [Open Source Community](#)
- [Enterprise](#)

[Research](#) [Advisories](#) [CodeQL Wall of Fame](#)

Resources

[Open Source Community](#) [Enterprise](#)[Events](#) [Get Involved](#)

July 24, 2026

GHSL-2024-196: Remote Code Execution (RCE) in Fooocus webui - CVE-2025-31114

[Kevin Stubbings](#)

Coordinated Disclosure Timeline

- 2024-08-21: [Issue](#) opened to address security issue. No response.
- 2024-10-02: Email sent privately to the maintainer's personal email address. No response.
- 2026-07-24: In absence of response, public fix proposal via this [PR](#)

Summary

The Fooocus web UI is vulnerable to remote code execution due to the unsafe use of `eval` when processing metadata JSON. An attacker with access to the Fooocus web UI may be able to execute arbitrary code on the instance.

Project

Fooocus

Tested Version

[v2.5.5](#)

Details

Remote Code Execution in webui.py (GHSL-2024-196)

The [load_parameter_button_click](#) function takes in the parameters of the prompt for image generation. It is accessible in multiple places on the single page web interface, including via the `load_parameter_button` and [trigger_metadata_import](#) function.

```
load_parameter_button.click(modules.meta_parser.load_parameter_button_click, inputs=[prompt, state_is_generating, inpaint_mode], outp

def trigger_metadata_import(file, state_is_generating):
    parameters, metadata_scheme = modules.meta_parser.read_info_from_image(file)
    if parameters is None:
        print('Could not find metadata in the image!')
        parsed_parameters = {}
    else:
        metadata_parser = modules.meta_parser.get_metadata_parser(metadata_scheme)
        parsed_parameters = metadata_parser.to_json(parameters)

    return modules.meta_parser.load_parameter_button_click(parsed_parameters, state_is_generating, inpaint_mode)  <----- called

metadata_import_button.click(trigger_metadata_import, inputs=[metadata_input_image, state_is_generating], outputs=load_data_outputs,
    .then(style_sorter.sort_styles, inputs=style_selections, outputs=style_selections, queue=False, show_progress=False))
```

The [load_parameter_button_click](#) function calls [get_list](#), [get_resolution](#), [get_adm_guidance](#) and [get_freeu](#) on various metadata entries.

```
def load_parameter_button_click(raw_metadata: dict | str, is_generating: bool, inpaint_mode: str):
    loaded_parameter_dict = raw_metadata
    if isinstance(raw_metadata, str):
        loaded_parameter_dict = json.loads(raw_metadata)
    assert isinstance(loaded_parameter_dict, dict)

    results = [len(loaded_parameter_dict) > 0]

    get_image_number('image_number', 'Image Number', loaded_parameter_dict, results)
    get_str('prompt', 'Prompt', loaded_parameter_dict, results)
    get_str('negative_prompt', 'Negative Prompt', loaded_parameter_dict, results)
```

```

get_list('styles', 'Styles', loaded_parameter_dict, results) # Sink 1
get_resolution('resolution', 'Resolution', loaded_parameter_dict, results) #Sink 2
get_adm_guidance('adm_guidance', 'ADM Guidance', loaded_parameter_dict, results) #Sink 3
...
get_freeu('freeu', 'FreeU', loaded_parameter_dict, results) #Sink 4

```

The [get_list](#) function calls eval on the value of the styles entry.

```

def get_list(key: str, fallback: str | None, source_dict: dict, results: list, default=None):
    try:
        h = source_dict.get(key, source_dict.get(fallback, default))
        h = eval(h)
        assert isinstance(h, list)
        results.append(h)
    except:
        results.append(gr.update())

```

The [get_resolution](#) function calls eval on the value of the resolution entry.

```

def get_resolution(key: str, fallback: str | None, source_dict: dict, results: list, default=None):
    try:
        h = source_dict.get(key, source_dict.get(fallback, default))
        width, height = eval(h)
        formatted = modules.config.add_ratio(f'{width}*{height}')
        if formatted in modules.config.available_aspect_ratios_labels:
            results.append(formatted)
            results.append(-1)
            results.append(-1)
        else:
            results.append(gr.update())
            results.append(int(width))
            results.append(int(height))
    except:
        results.append(gr.update())
        results.append(gr.update())
        results.append(gr.update())

```

The [get_adm_guidance](#) function calls eval on the value of the adm_guidance entry.

```

def get_adm_guidance(key: str, fallback: str | None, source_dict: dict, results: list, default=None):
    try:
        h = source_dict.get(key, source_dict.get(fallback, default))
        p, n, e = eval(h)
        results.append(float(p))
        results.append(float(n))
        results.append(float(e))
    except:
        results.append(gr.update())
        results.append(gr.update())
        results.append(gr.update())

```

The [get_freeu](#) function calls eval on the value of the freeu entry.

```

def get_freeu(key: str, fallback: str | None, source_dict: dict, results: list, default=None):
    try:
        h = source_dict.get(key, source_dict.get(fallback, default))
        b1, b2, s1, s2 = eval(h)
        results.append(True)
        results.append(float(b1))
        results.append(float(b2))
        results.append(float(s1))
        results.append(float(s2))
    except:
        results.append(False)
        results.append(gr.update())
        results.append(gr.update())
        results.append(gr.update())
        results.append(gr.update())

```

An attacker can gain Remote Code Execution by uploading a modified image file that contains malicious metadata.

Impact

This issue may lead to Remote Code Execution.

CVE

- CVE-2025-31114

Credit

This issue was discovered and reported by GHSL team member [@Kwstubbs \(Kevin Stubbings\)](#) with the help of [CodeQL](#).

Contact

You can contact the GHSL team at securitylab@github.com. Please include a reference to GHSL-2024-196 in any communication regarding this issue.

GitHub

Product

- [Features](#)
- [Security](#)
- [Team](#)
- [Enterprise](#)
- [Customer stories](#)
- [The ReadME Project](#)
- [Pricing](#)
- [Resources](#)
- [Roadmap](#)
- [Compare GitHub](#)

Platform

- [Developer API](#)
- [Partners](#)
- [Atom](#)
- [Electron](#)
- [GitHub Desktop](#)

Support

- [Docs](#)
- [Community Forum](#)
- [Professional Services](#)
- [GitHub Skills](#)
- [Status](#)
- [Contact GitHub](#)

Company

- [About](#)
- [Blog](#)
- [Careers](#)
- [Press](#)
- [Inclusion](#)
- [Social Impact](#)
- [Shop](#)



- GitHub Inc. © 2024
- [Terms](#)
- [Privacy](#)
- [Sitemap](#)
- [What is Git?](#)
- [Manage Cookies](#)
- [Do not share my personal information](#)