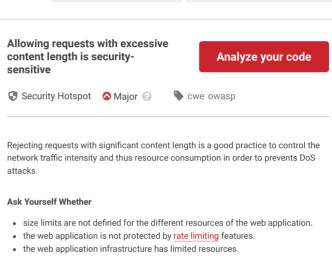


should implement "IDisposable"

📆 Bug



## Recommended Secure Coding Practices • For most of the features of an application, it is recommended to limit the size of

requests to:

o lower or equal to 8mb for file uploads.

There is a risk if you answered yes to any of those questions.

o lower or equal to 2mb for other requests.

It is recommended to customize the rule with the limit values that correspond to the web application.

## Sensitive Code Example

```
using Microsoft.AspNetCore.Mvc;

public class MyController : Controller
{
    [HttpPost]
    [DisableRequestSizeLimit] // Sensitive: No size limit
    [RequestSizeLimit(10000000)] // Sensitive: 10MB is more
    public IActionResult PostRequest(Model model)
    {
        // ...
    }

[HttpPost]
    [RequestFormLimits(MultipartBodyLengthLimit = 8000000)]
    public IActionResult MultipartFormRequest(Model model)
    {
        // ...
    }
}
```

In Web.config:

Calls to "async" methods should not be blocking

Code Smell

Child class fields should not shadow parent class fields

Code Smell

Track lack of copyright and license headers

Code Smell

Exit methods should not be called

Code Smell

## **Compliant Solution**

```
using Microsoft.AspNetCore.Mvc;

public class MyController : Controller
{
    [HttpPost]
    [RequestSizeLimit(8000000)] // Compliant: 8MB
    public IActionResult PostRequest(Model model)
    {
        // ...
    }

    [HttpPost]
    [RequestFormLimits(MultipartBodyLengthLimit = 8000000)]
    public IActionResult MultipartFormRequest(Model model)
    {
        // ...
    }
}
```

In Web.config:

## See

- OWASP Top 10 2021 Category A5 Security Misconfiguration
- Owasp Cheat Sheet Owasp Denial of Service Cheat Sheet
- OWASP Top 10 2017 Category A6 Security Misconfiguration
- MITRE, CWE-770 Allocation of Resources Without Limits or Throttling
- MITRE, CWE-400 Uncontrolled Resource Consumption
- Web.config XML-formatted config file for IIS applications

Available In:

sonarcloud 👌 | sonarqube