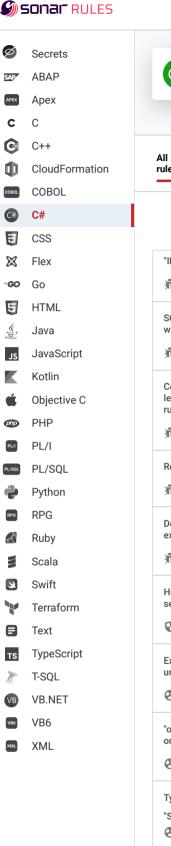
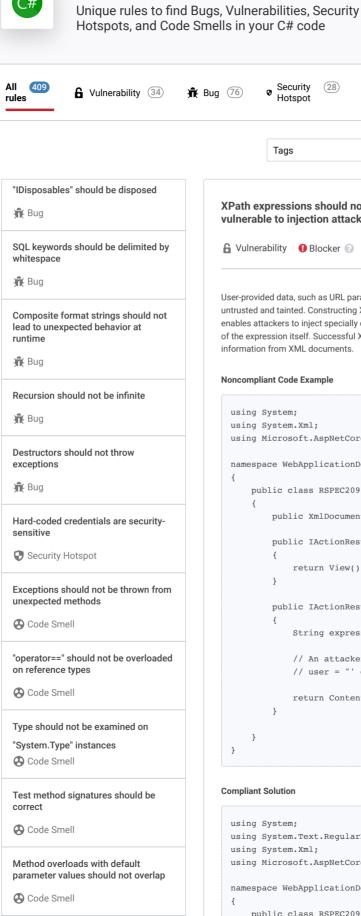
Analyze your code

injection cwe owasp



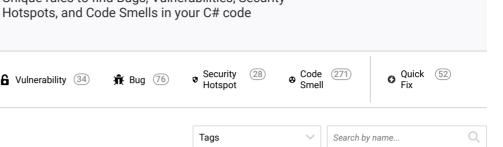




"value" parameters should be used

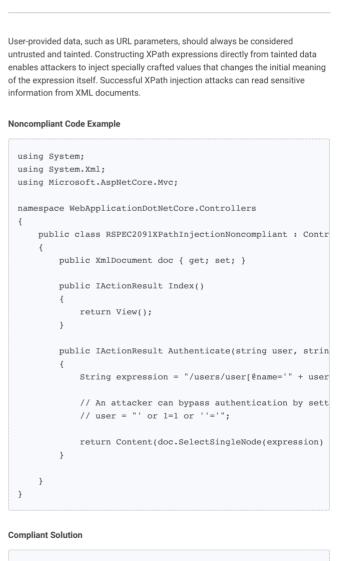
Code Smell

C# static code analysis



XPath expressions should not be

vulnerable to injection attacks



```
using System;
using System.Text.RegularExpressions;
using System.Xml:
using Microsoft.AspNetCore.Mvc;
namespace WebApplicationDotNetCore.Controllers
    public class RSPEC2091XPathInjectionCompliant : Controll
        public XmlDocument doc { get; set; }
```

"is" should not be used with "this"

Code Smell

Methods named "Dispose" should implement "IDisposable.Dispose"

Code Smell

Tests should include assertions

Code Smell

Silly bit operations should not be performed

Code Smell

```
public IActionResult Index()
{
    return View();
}

public IActionResult Authenticate(string user, strin
{
    // Restrict the username and password to letters
    if (!Regex.IsMatch(user, "^[a-zA-Z]+$") || !Rege
    {
        return BadRequest();
    }

    String expression = "/users/user[@name='" + user
    return Content(doc.SelectSingleNode(expression))
}

}
```

See

- OWASP Top 10 2021 Category A3 Injection
- OWASP Top 10 2017 Category A1 Injection
- MITRE, CWE-20 Improper Input Validation
- MITRE, CWE-643 Improper Neutralization of Data within XPath Expressions

Available In:

sonarcloud 🚳 | sonarqube | Developer Edition

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy