Deep Dive into SAST & Semgrep

Nikhil Sahoo
Ravindra Penumarthi

Agenda

Understanding SAST

SAST for Developers & Pentesters

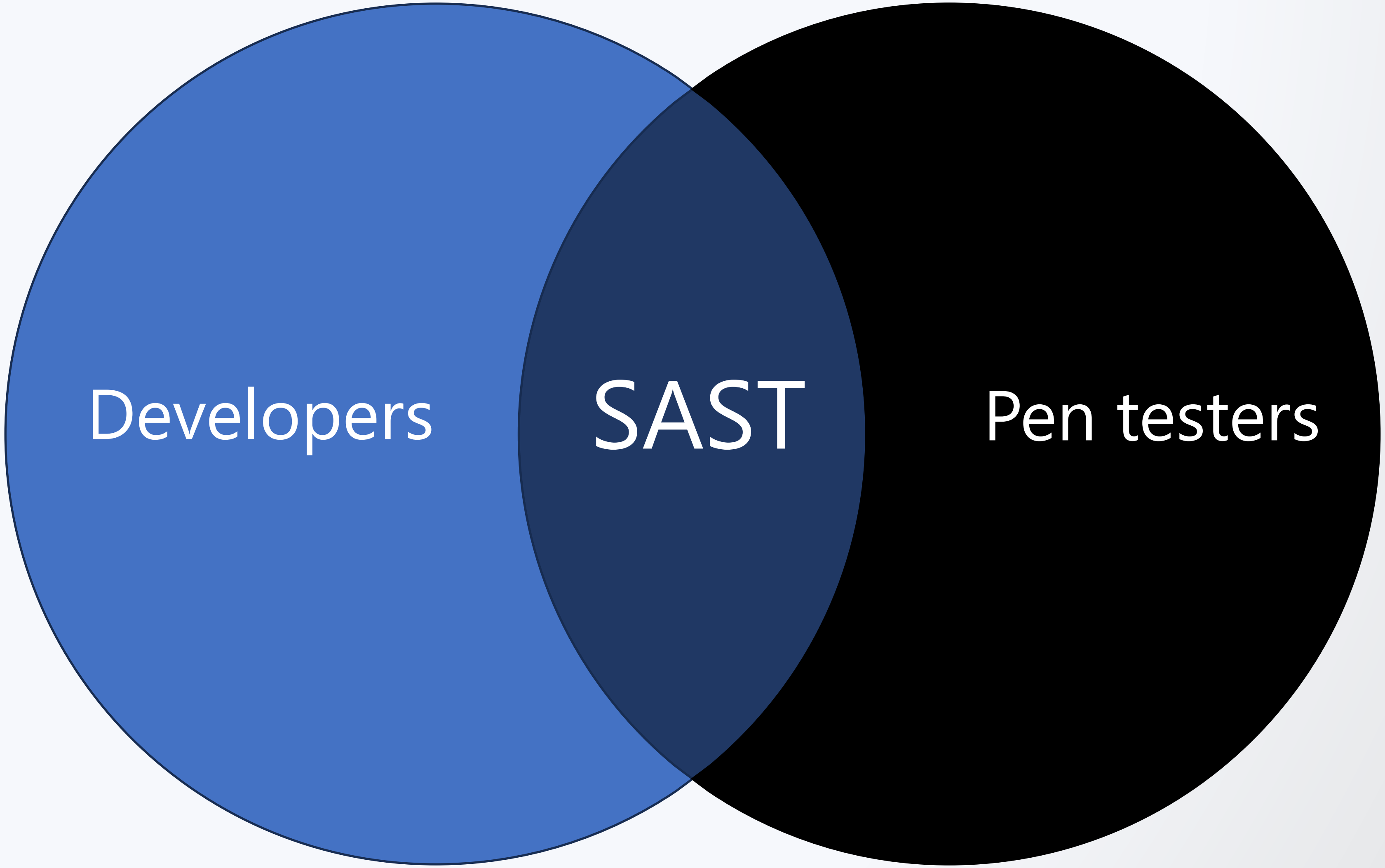
SAST tools & comparison

Semgrep & rules

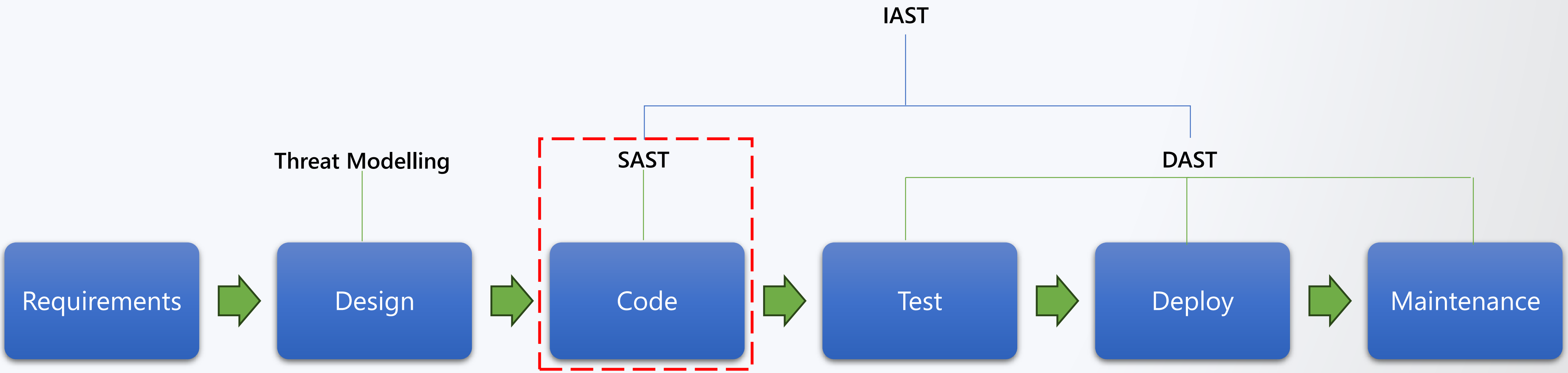
RTF Challenge

```
rules:
  - id: Ravindra Penumarthi
  patterns:
    - pattern:
        work:
        - Security SWE Engineer 2 at Microsoft
        - Azure Edge + Platform Security SERPENT Team
    - pattern:
        education:
        - MTech, Cyber Security from BITS Pilani*
        - MCA from BVRIT
    - pattern:
        affiliation:
        - Chapter Lead, null Hyderabad
        - Distinguished Motorcyclist, Road Thrill Hyderabad
    Polyglot:
        - Telugu, English, Hindi, Deccani, Tamil*
        - C#, JS, PS, PY
  skills: [Can Code, Can Hack, Can Talk, Can Drive, Can Joke]
  metadata
    description: "Ravindra is a Developer turned Ethical Hacker"
    tags: CVE-2025-2102
```

```
rules:
  - id: Nikhil Sahoo
  patterns:
    - pattern:
        work:
        - Security SWE Engineer 2 at Microsoft
        - Azure Edge + Platform Security SERPENT Team
    - pattern:
        certifications:
        - OSCP, eWPTX, CRTP
       achievements: Acknowledged from multiple orgs such as Oracle,
                     Dell, Microsoft, Apple, SAP, Sony etc.
       CVE: CVE-2018-11471, CVE-2018-11472, CVE-2018-11473,
            CVE-2018-11474, CVE-2018-11475
    Polyglot:
       - English, Hindi, Odia
       - C#, JS, PS, PY
  skills: [Can Code, Can Hack, Can Talk]
  metadata
    tags: CVE-2025-2102
    links: https://www.linkedin.com/in/nikhilsahoo,
           https://github.com/nikhil1232
```


Developers
SAST
Pen testers

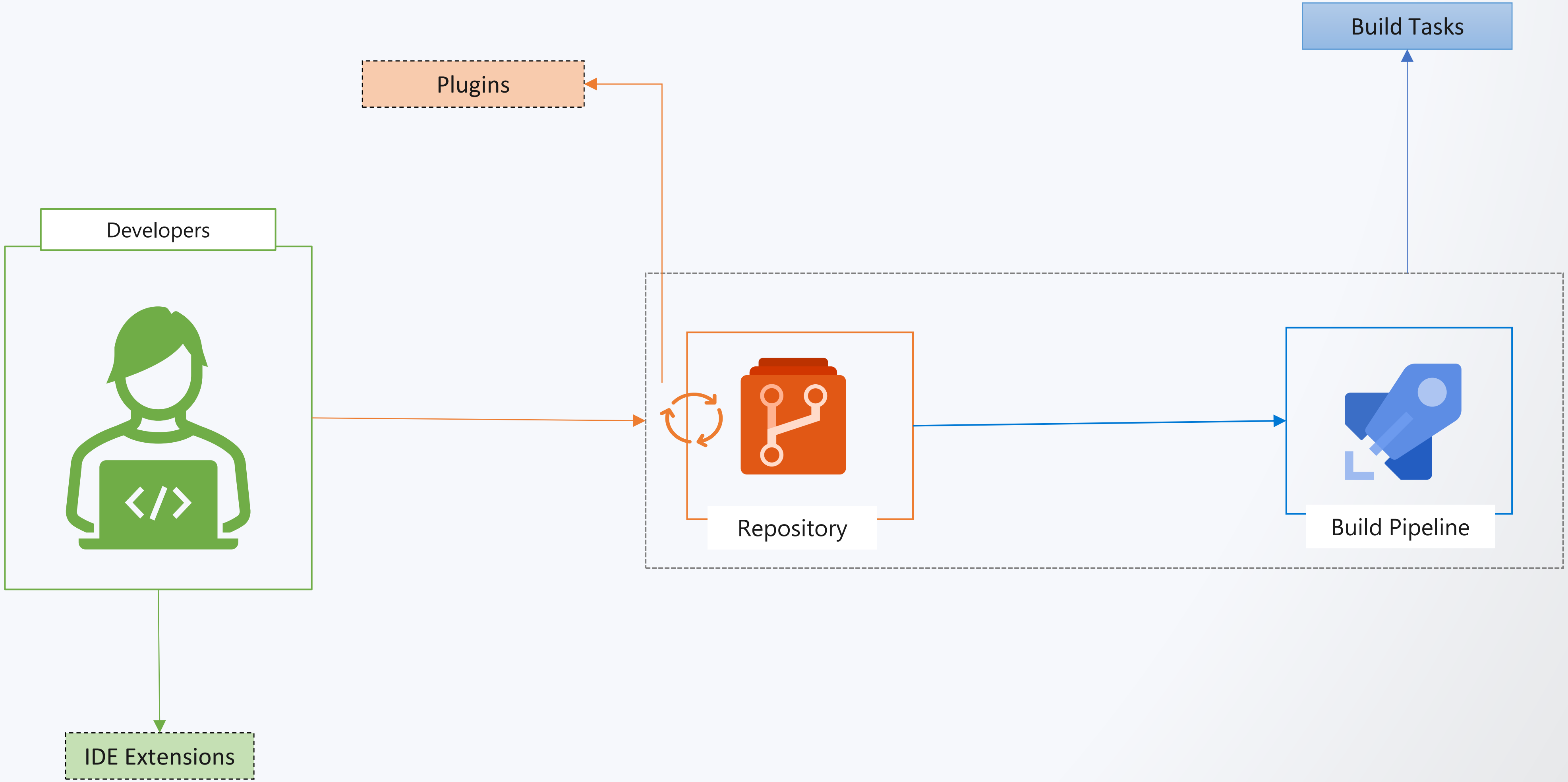
Dev side of the story



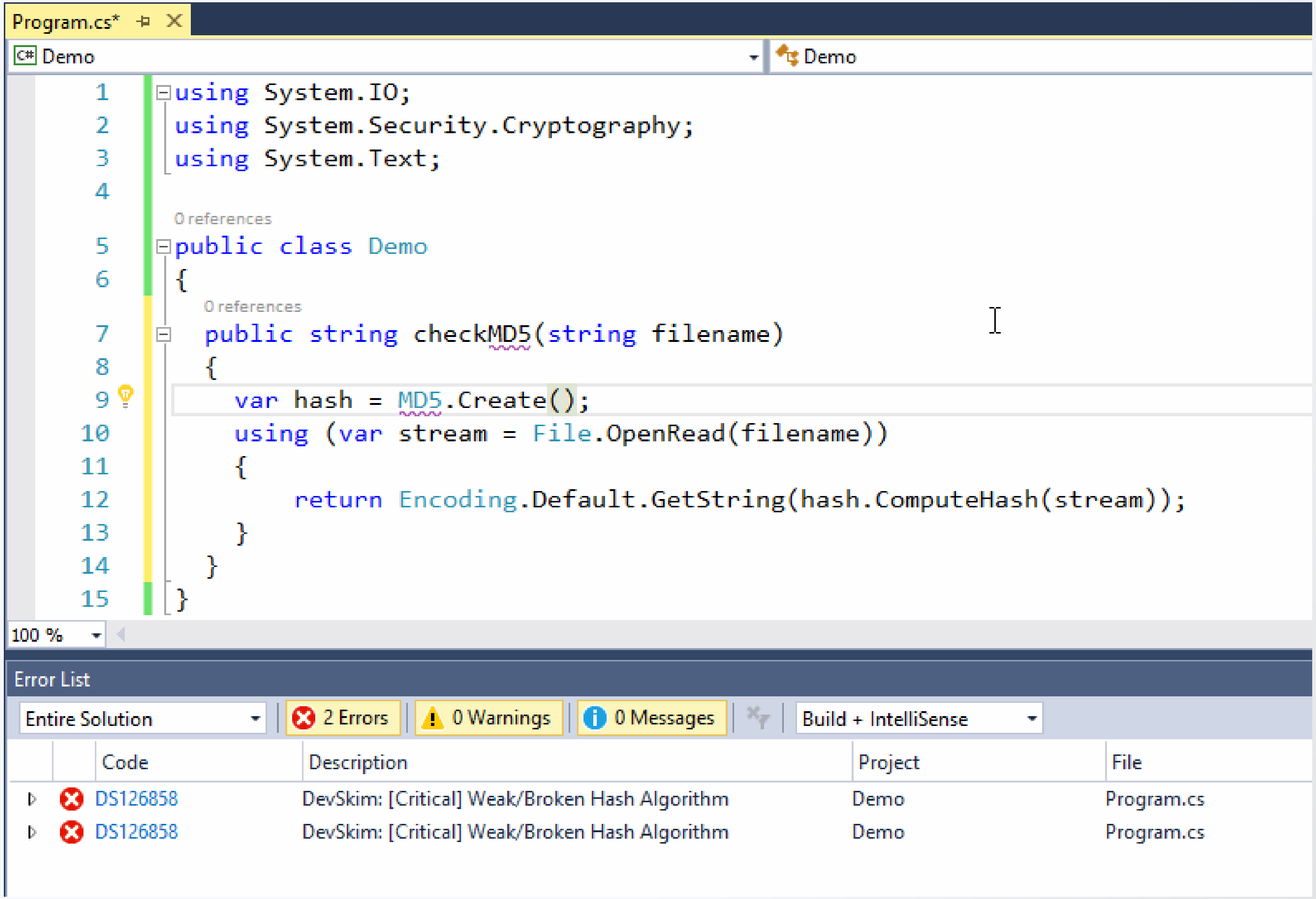
Static Application Security Testing

- Analyzes Source Code without execution
- Most of OWASP vulns are found at code Level
- Originally meant for Developers
- Early identification of Vulnerabilities
- Cost effective
- Various stages of Implementation

When to SAST?


Build Tasks
Plugins
Developers
Repository
Build Pipeline
IDE Extensions

SAST at IDE



```
using System.IO;
using System.Security.Cryptography;
using System.Text;

public class Demo
{
    public string checkMD5(string filename)
    {
        var hash = MD5.Create();
        using (var stream = File.OpenRead(filename))
        {
            return Encoding.Default.GetString(hash.ComputeHash(stream));
        }
    }
}
```

Code	Description	Project	File
DS126858	DevSkim: [Critical] Weak/Broken Hash Algorithm	Demo	Program.cs
DS126858	DevSkim: [Critical] Weak/Broken Hash Algorithm	Demo	Program.cs

SAST at Repo level

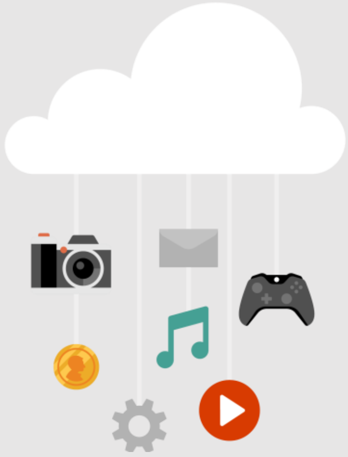
- Scan for Secrets

- Scan for Vulnerable/Outdated packages

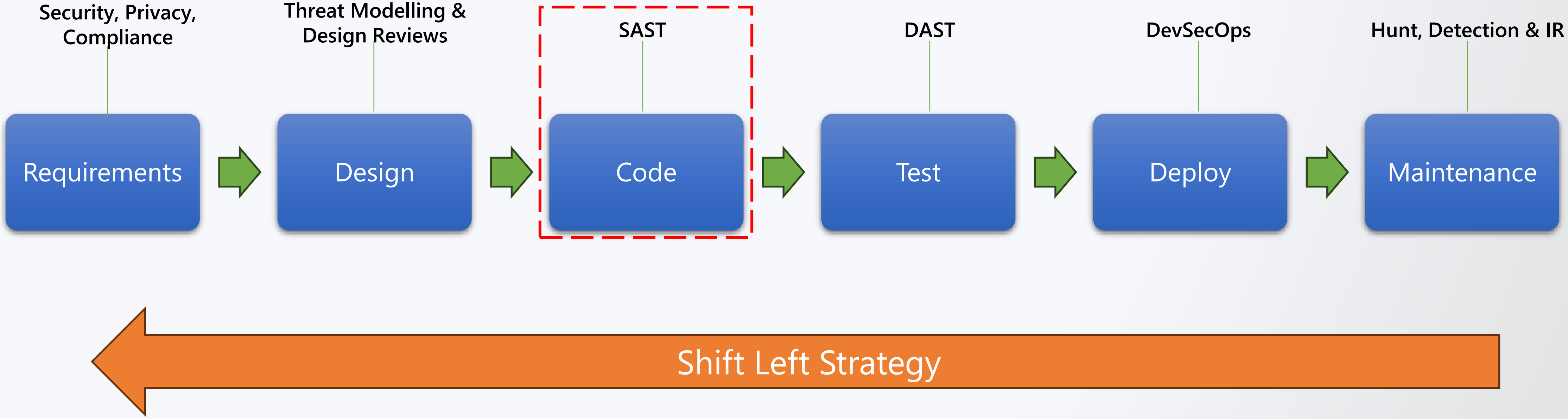
- AI based plugins (ex: PR Reviews)

SAST at Build Pipelines

Demo time!



Pentest side of the story



Pentesters & SAST

- Use SAST results to perform Targeted DAST
- Pentesters perform in-depth SAST
- Perform both Manual & Automated reviews
- Build and enhance SAST tools

A taste of Manual Source Code Review

```
[ApiController]
    [Route("api/[controller]")]
    public class YourController : ControllerBase
    {
        private readonly HttpClient _httpClient;

        [HttpGet]
        public async Task<IActionResult> GetRequest([FromQuery] string url)
        {

            try
            {
                var response = await _httpClient.GetAsync(url);
                var content = await response.Content.ReadAsStringAsync();
                return Ok(content);
            }
            catch (HttpRequestException e)
            {
                return StatusCode(500, $"Internal server error: {e.Message}");
            }
        }
    }
```

https://mydomain/api/your/GetRequest?url=https://attacker.com/

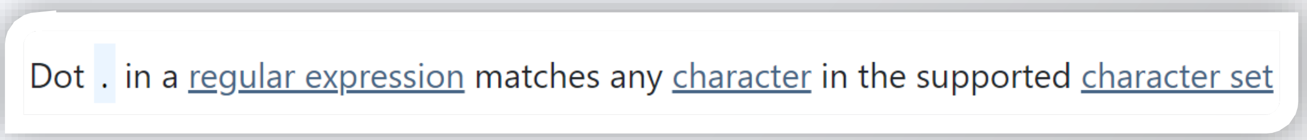
SSRF

Not as easy as you think

String contains dots (.)

```
function validateUri(uri: string): string {
    const keyVaultDnsSuffix = "vault.azure.net";
    const UriRegex = `^https:\/\/[a-zA-Z0-9-]{3,24}\.${keyVaultDnsSuffix}[:0-9]*\/secrets\/[a-
zA-Z0-9-]{1,127}\/*$`;
    const result = uri.match(UriRegex);
    return (result && result?.length) ? "Valid" : "Invalid";
}
```

Dots enter Regex without escaping!



Dot . in a regular expression matches any character in the supported character set

Spot the Bug – Cont..

```
const UriRegex = `^https:\/\/[a-zA-Z0-9-]{3,24}\.${keyVaultDnsSuffix}[:0-9]*\/secrets\/[a-zA-Z0-9-]{1,127}\/*$`;
```

```
const UriRegex = `^https:\/\/[a-zA-Z0-9-]{3,24}\.vault.azure.net[:0-9]*\/secrets\/[a-zA-Z0-9-]{1,127}\/*$`;
```

```
// Test cases
validateUri("https://test.vault.azure.net/secrets/secretname"); // Valid
validateUri("https://malicious-domain.net/secrets/secretname"); // Invalid
validateUri("https://test.vaultAazure.net/secrets/secretname"); // Valid
validateUri("https://test.vaultxazure.net/secrets/secretname"); // Valid
```

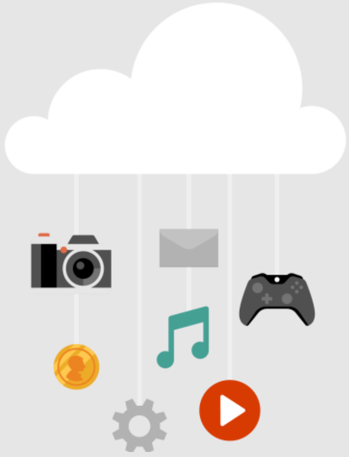
Attacker's domains

Automated Source code Review

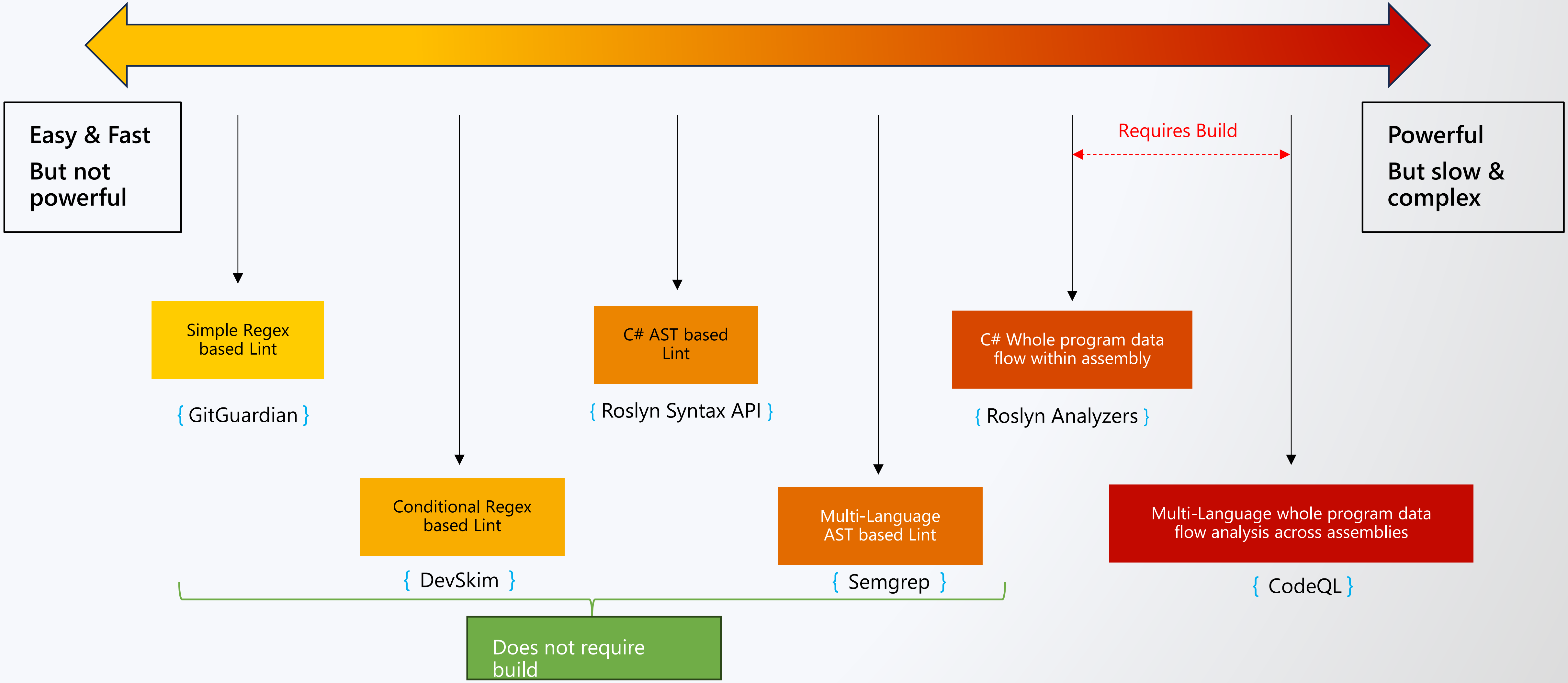
- Regex/String based lint
- AST based lint
- Taint and Data Flow analysis

Demo

DevSkim and its rules



SCA Tools Infographic



Easy & Fast

But not powerful

Powerful

But slow & complex

Requires Build

Simple Regex based Lint

{ GitGuardian }

Conditional Regex based Lint

{ DevSkim }

C# AST based Lint

{ Roslyn Syntax API }

Multi-Language AST based Lint

{ Semgrep }

C# Whole program data flow within assembly

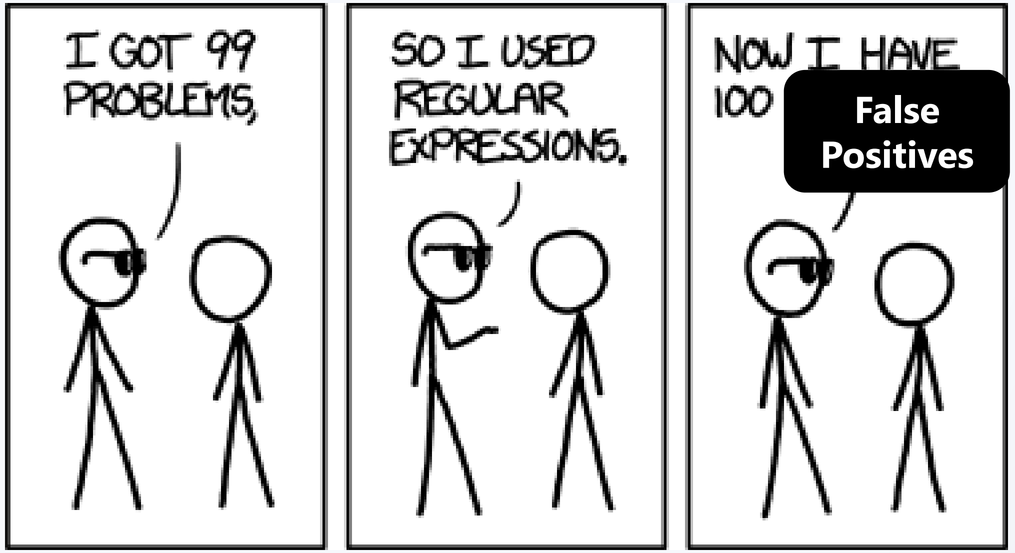
{ Roslyn Analyzers }

Multi-Language whole program data flow analysis across assemblies

{ CodeQL }

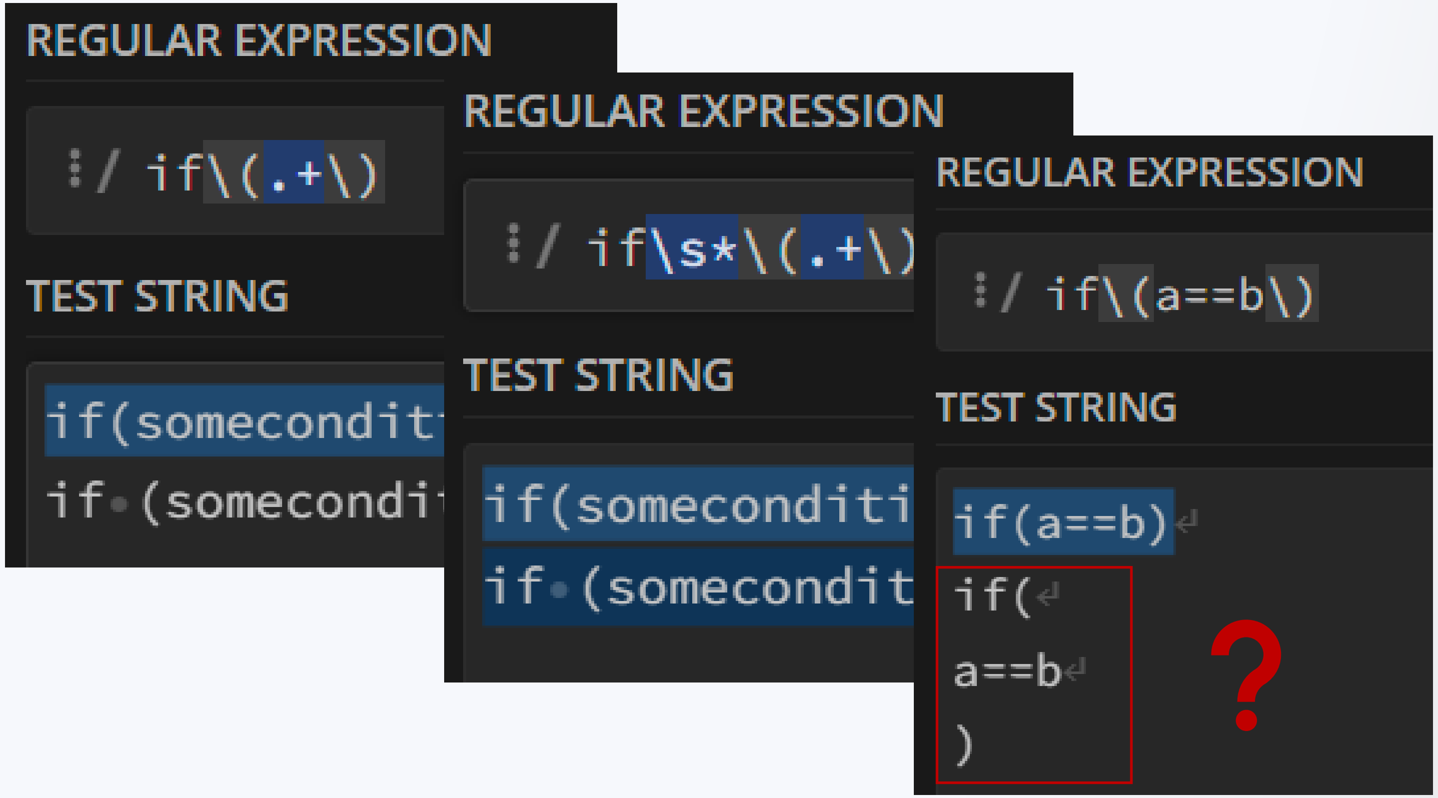
Does not require build

Regex Problems



https://www.explainxkcd.com/wiki/images/1/10/perl_problems.png

Regex Problems



REGULAR EXPRESSION

```
/ if\(.+\)
```

TEST STRING

```
if(somecondit
if (somecondi
```

REGULAR EXPRESSION

```
/ if\s*\(.+\)
```

TEST STRING

```
if(somecondit
if (somecondit
```

REGULAR EXPRESSION

```
/ if\(a==b\)
```

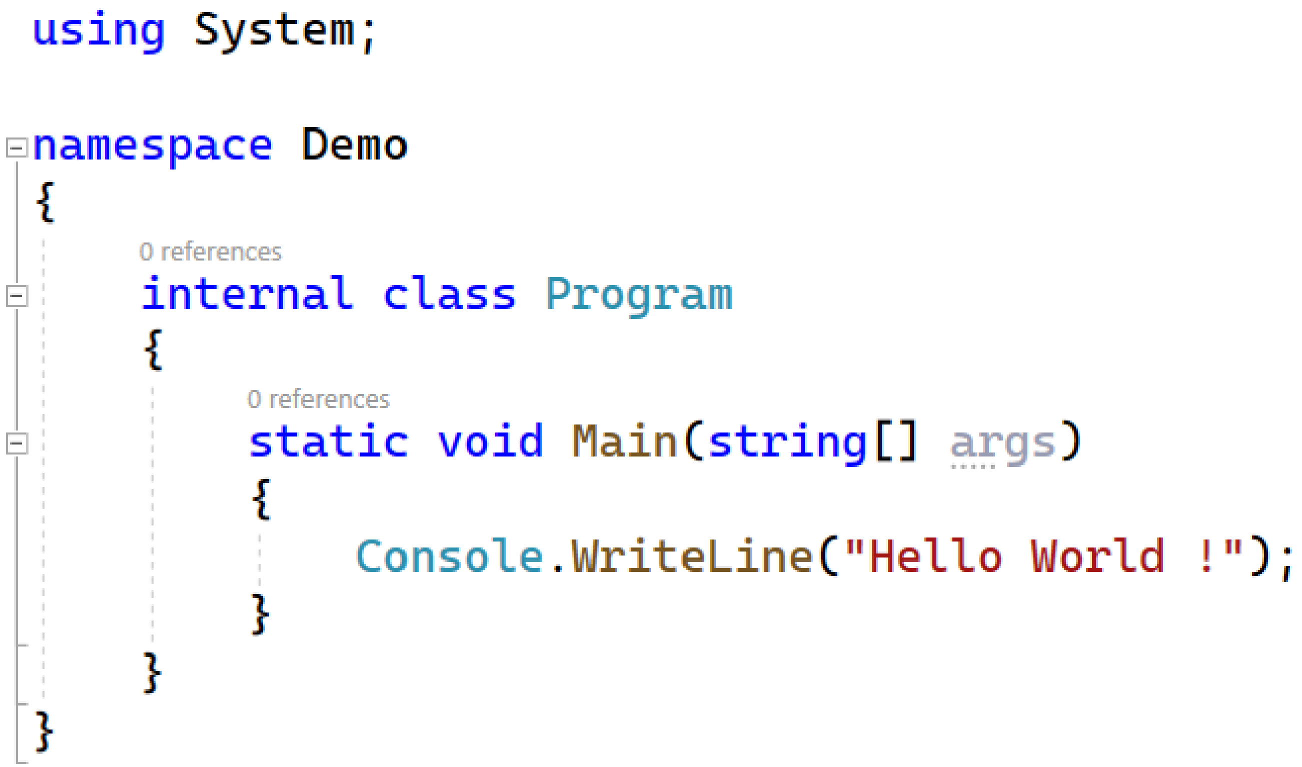
TEST STRING

```
if(a==b)
if(
a==b
)
```

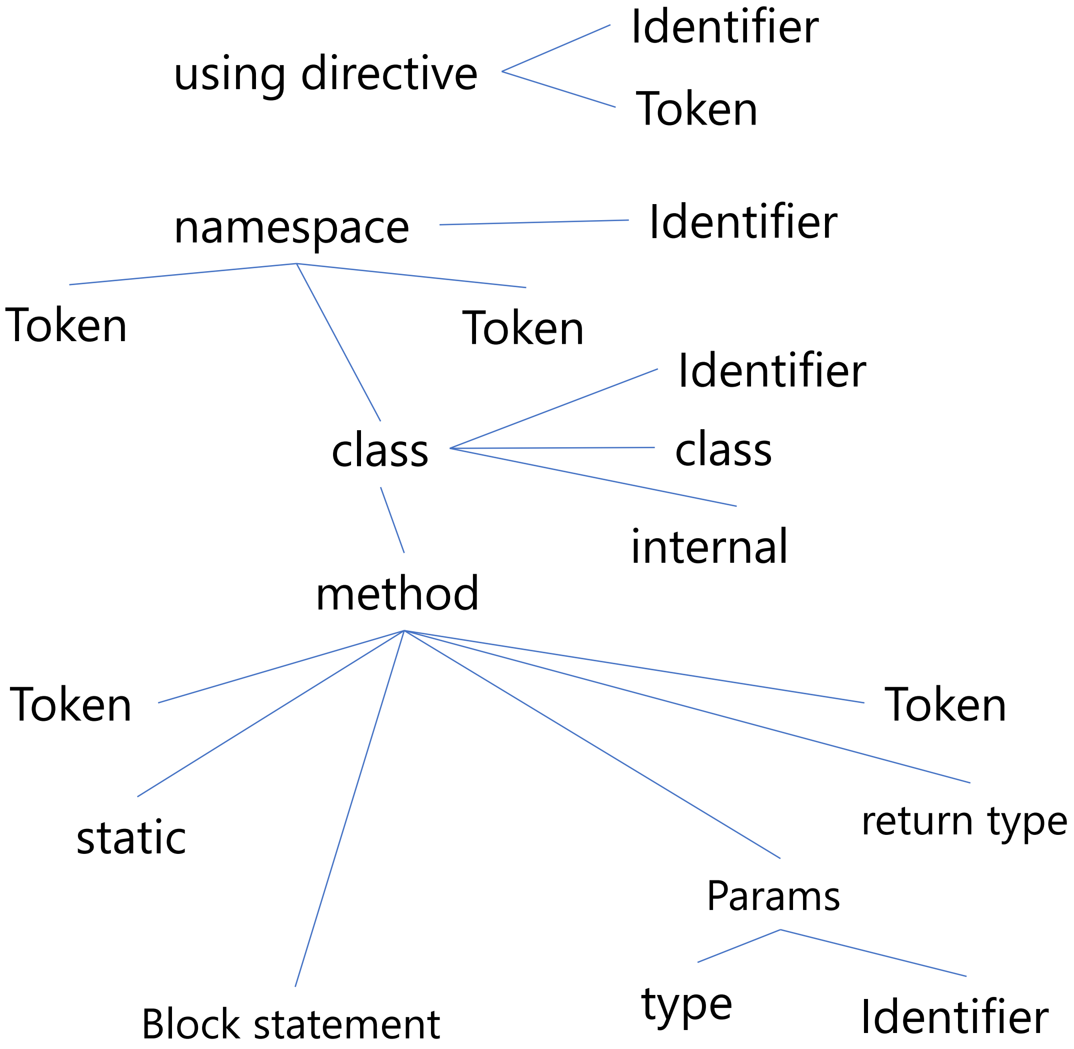
?

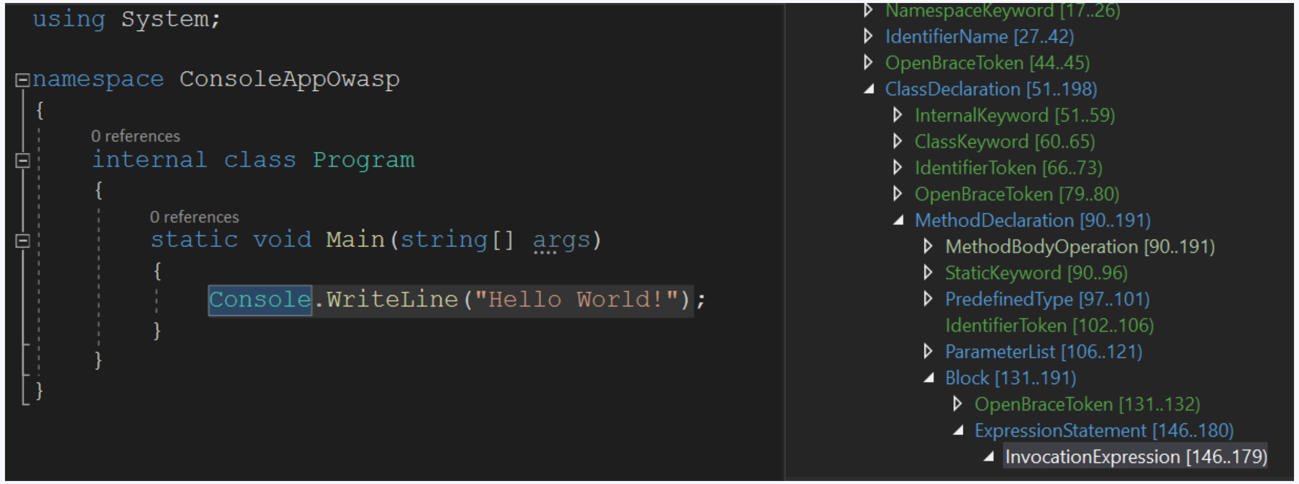
Code is not a string, It’s a Tree

🧶 String != 🌲 Tree



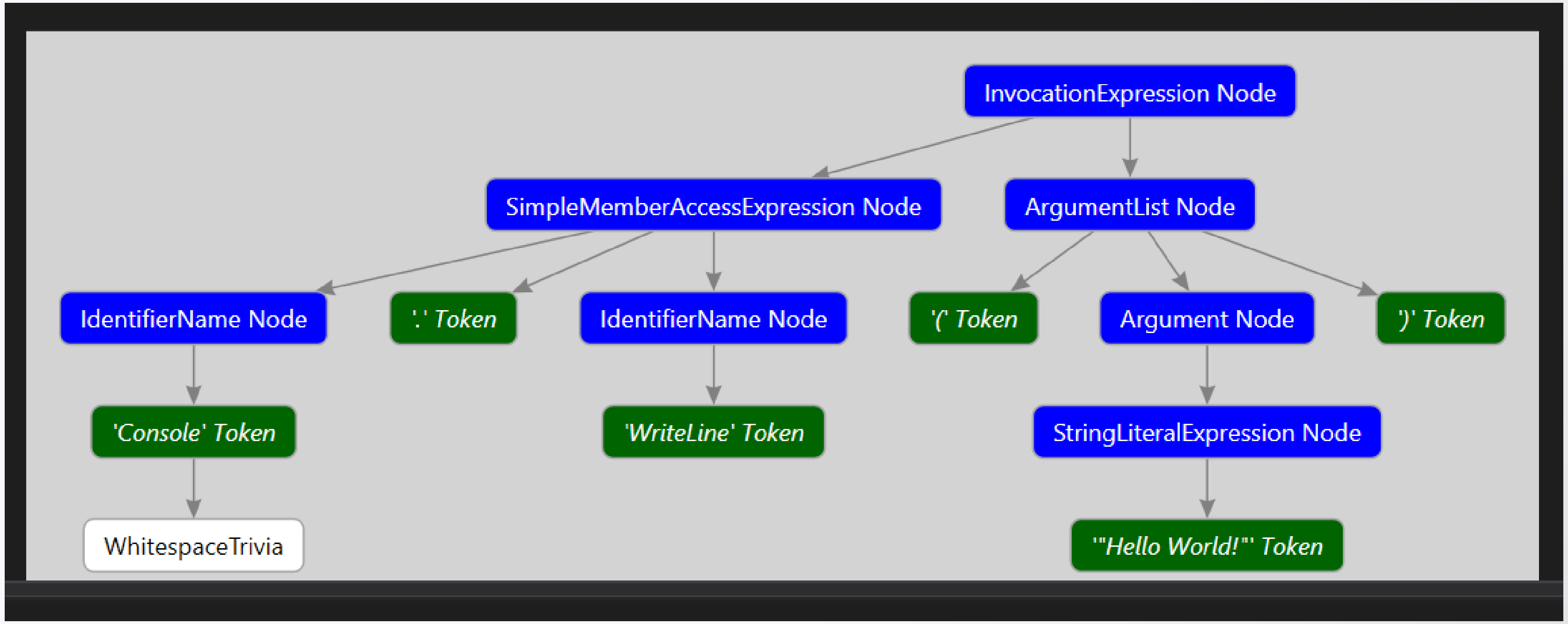
```
using System;

namespace Demo
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World !");
        }
    }
}
```





```
using System;

namespace ConsoleAppOwasp
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

```
▷ NamespaceKeyword [17..26)
▷ IdentifierName [27..42)
▷ OpenBraceToken [44..45)
◢ ClassDeclaration [51..198)
   ▷ InternalKeyword [51..59)
   ▷ ClassKeyword [60..65)
   ▷ IdentifierToken [66..73)
   ▷ OpenBraceToken [79..80)
   ◢ MethodDeclaration [90..191)
      ▷ MethodBodyOperation [90..191)
      ▷ StaticKeyword [90..96)
      ▷ PredefinedType [97..101)
        IdentifierToken [102..106)
      ▷ ParameterList [106..121)
      ◢ Block [131..191)
         ▷ OpenBraceToken [131..132)
         ◢ ExpressionStatement [146..180)
            ◢ InvocationExpression [146..179)
```



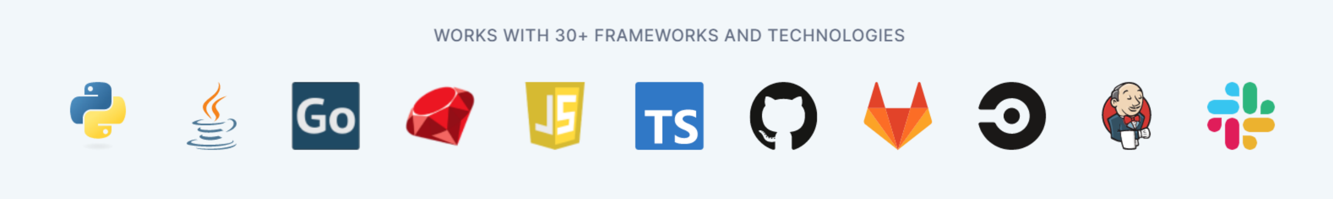
What is Semgrep?

- Fast and lightweight static analysis tool to find bugs and enforce code standards.

Semgrep

Code scanning at ludicrous speed.
Find bugs and reachable dependency vulnerabilities in code.

Reference: https://semg



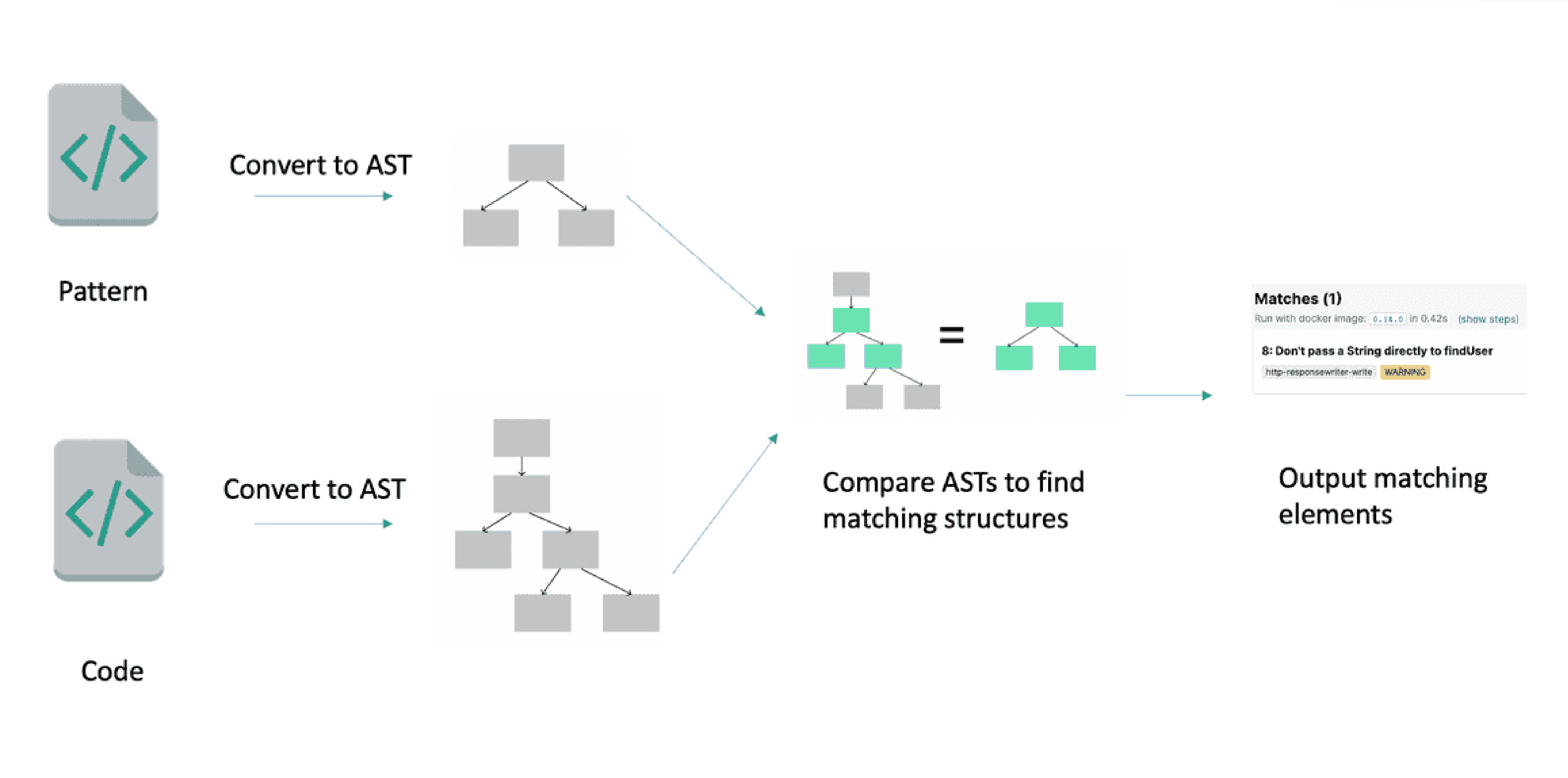
WORKS WITH 30+ FRAMEWORKS AND TECHNOLOGIES

History: Sgrep (Syntactic Grep)

- Initially called Sgrep/Pfff
- Written By Yoann Padioleau at Facebook for analyzing PHP code
- Was used to Enforce Best Practices
- Easy for developers to organize and understand the rules
- Joined R2C and renamed Sgrep to Semgrep
- Goal was to match based on semantics of the code

Reference: https://semgrep.dev/blog/2021/semgrep-a-static-analysis-journey

Internals



Pattern

Convert to AST

Code

Convert to AST

Compare ASTs to find matching structures

Matches (1)

Run with docker image: 0.14.0 in 0.42s [show steps]

8: Don't pass a String directly to findUser

http-responsewriter-write WARNING

Output matching elements

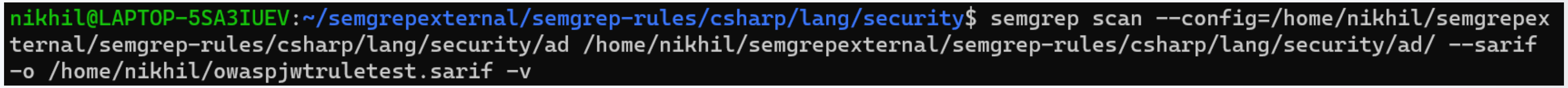
https://r2c.dev/static/00125f77fba64f5350b367c373c4e849/1132d/semgrep-flow.png

Installation and Getting Started

- Must have Python 3.9 or later where the Semgrep CLI is running
- python3 -m pip install semgrep
- Supports installation on multiple OS such as Linux, mac and windows
- Has a docker version
- Supports GUI and CLI mode. CLI scans can also be done without the need of a Github/Gitlab account.

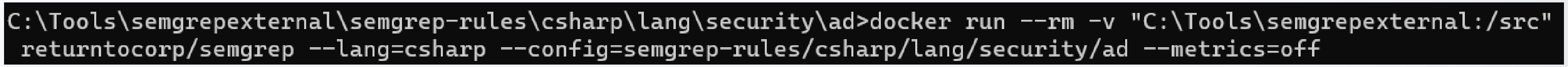
Usage

CLI



```
nikhil@LAPTOP-5SA3IUEV:~/semgrepexternal/semgrep-rules/csharp/lang/security$ semgrep scan --config=/home/nikhil/semgrepex
ternal/semgrep-rules/csharp/lang/security/ad /home/nikhil/semgrepexternal/semgrep-rules/csharp/lang/security/ad/ --sarif
-o /home/nikhil/owaspjwtruletest.sarif -v
```

Docker

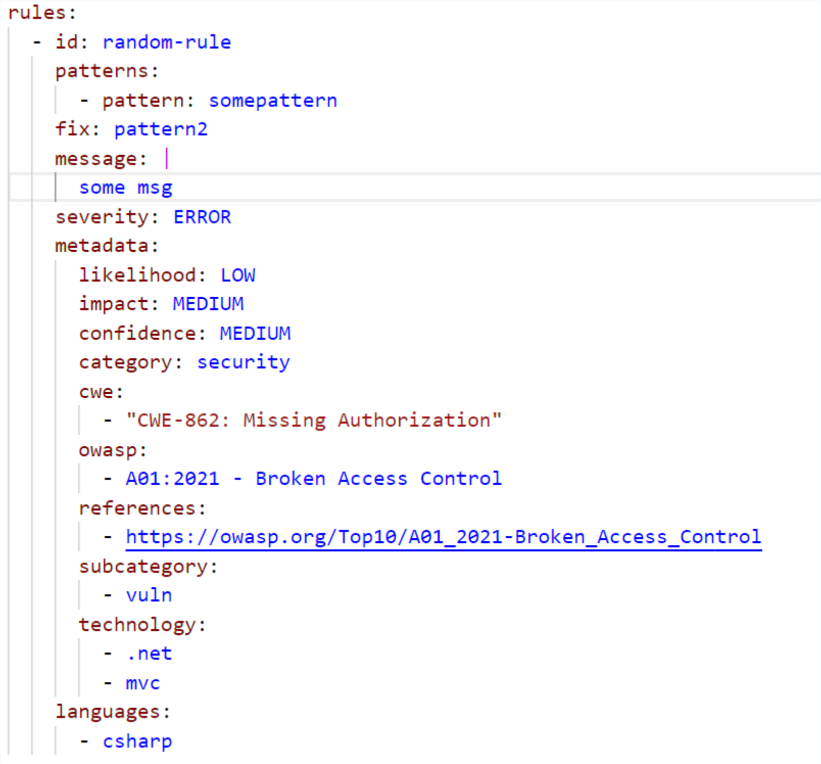


```
C:\Tools\semgrepexternal\semgrep-rules\csharp\lang\security\ad>docker run --rm -v "C:\Tools\semgrepexternal:/src"
 returntocorp/semgrep --lang=csharp --config=semgrep-rules/csharp/lang/security/ad --metrics=off
```

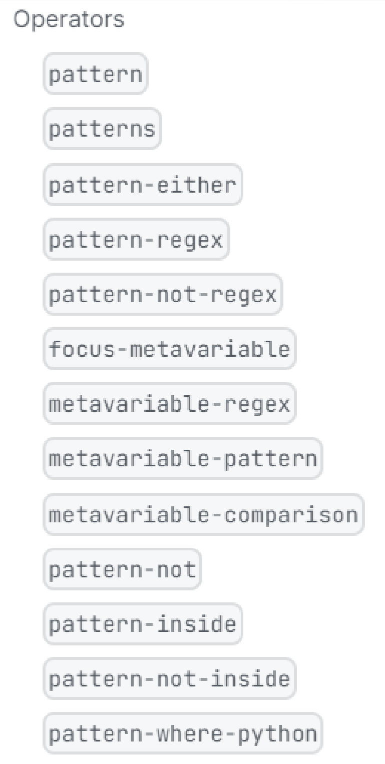
Semgrep in Editors

- IntelliJ extension: semgrep-intellij
- Microsoft Visual Studio Code: semgrep-vscode

Rule Syntax



```
rules:
  - id: random-rule
    patterns:
      - pattern: somepattern
    fix: pattern2
    message: |
      some msg
    severity: ERROR
    metadata:
      likelihood: LOW
      impact: MEDIUM
      confidence: MEDIUM
      category: security
      cwe:
        - "CWE-862: Missing Authorization"
      owasp:
        - A01:2021 - Broken Access Control
      references:
        - https://owasp.org/Top10/A01_2021-Broken_Access_Control
      subcategory:
        - vuln
      technology:
        - .net
        - mvc
    languages:
      - csharp
```

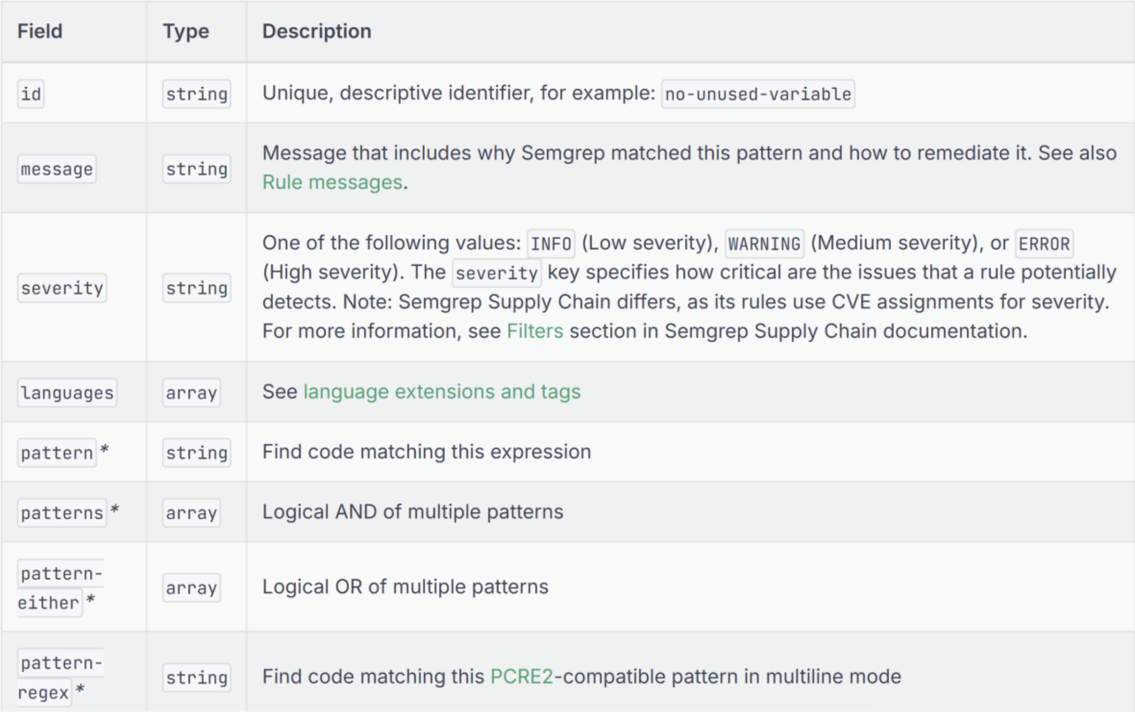


Operators

- `pattern`
- `patterns`
- `pattern-either`
- `pattern-regex`
- `pattern-not-regex`
- `focus-metavariable`
- `metavariable-regex`
- `metavariable-pattern`
- `metavariable-comparison`
- `pattern-not`
- `pattern-inside`
- `pattern-not-inside`
- `pattern-where-python`

https://semgrep.dev/docs/writing-rules/rule-syntax/

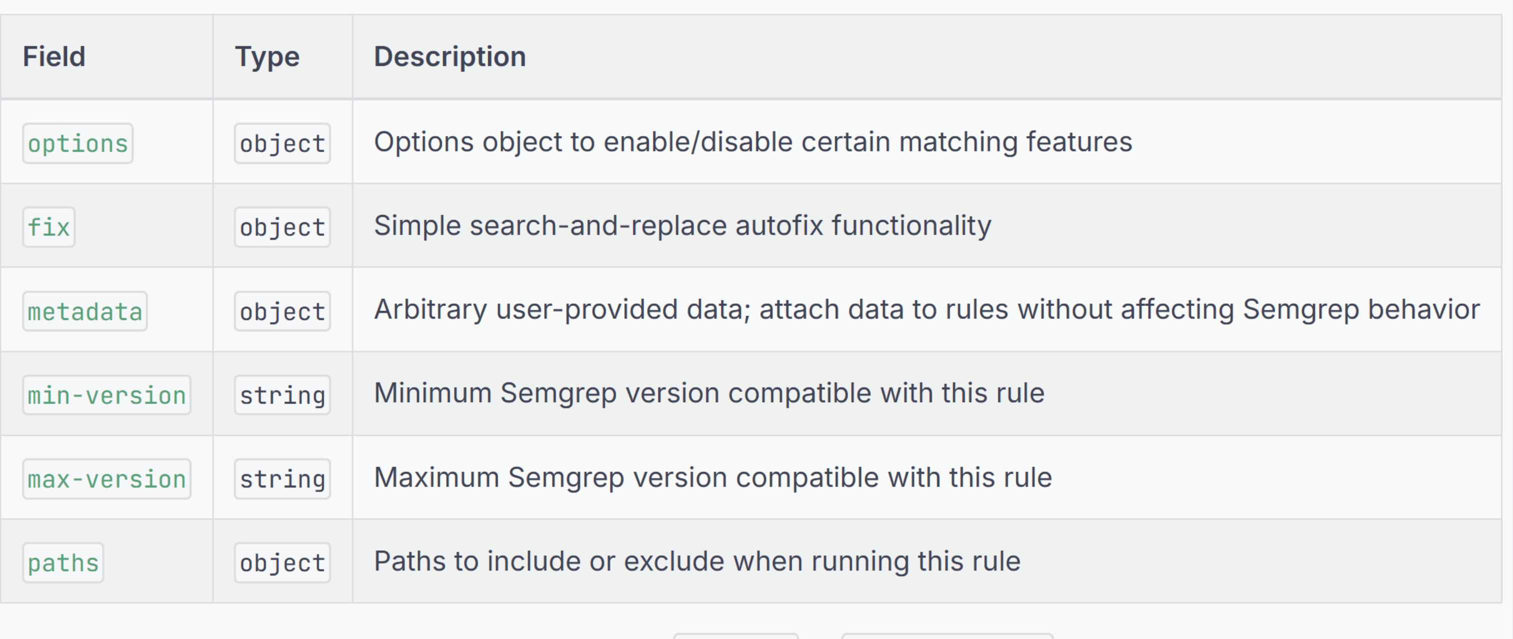
Rule Syntax – Required Fields



Field	Type	Description
`id`	`string`	Unique, descriptive identifier, for example: `no-unused-variable`
`message`	`string`	Message that includes why Semgrep matched this pattern and how to remediate it. See also Rule messages.
`severity`	`string`	One of the following values: `INFO` (Low severity), `WARNING` (Medium severity), or `ERROR` (High severity). The `severity` key specifies how critical are the issues that a rule potentially detects. Note: Semgrep Supply Chain differs, as its rules use CVE assignments for severity. For more information, see Filters section in Semgrep Supply Chain documentation.
`languages`	`array`	See language extensions and tags
`pattern` *	`string`	Find code matching this expression
`patterns` *	`array`	Logical AND of multiple patterns
`pattern-either` *	`array`	Logical OR of multiple patterns
`pattern-regex` *	`string`	Find code matching this PCRE2-compatible pattern in multiline mode

https://semgrep.dev/docs/writing-rules/rule-syntax

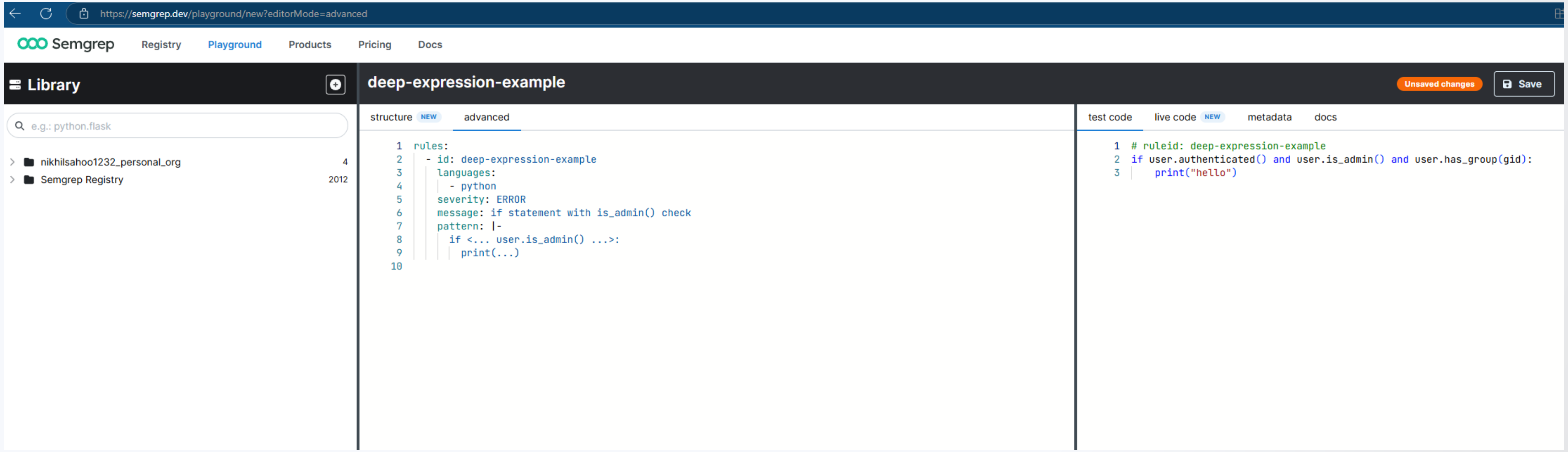
Rule Syntax – Optional Fields



Field	Type	Description
`options`	`object`	Options object to enable/disable certain matching features
`fix`	`object`	Simple search-and-replace autofix functionality
`metadata`	`object`	Arbitrary user-provided data; attach data to rules without affecting Semgrep behavior
`min-version`	`string`	Minimum Semgrep version compatible with this rule
`max-version`	`string`	Maximum Semgrep version compatible with this rule
`paths`	`object`	Paths to include or exclude when running this rule

https://semgrep.dev/docs/writing-rules/rule-syntax

Playground



```yaml
rules:
  - id: deep-expression-example
    languages:
      - python
    severity: ERROR
    message: if statement with is_admin() check
    pattern: |-
      if <... user.is_admin() ...>:
        print(...)

```

```python
# ruleid: deep-expression-example
if user.authenticated() and user.is_admin() and user.has_group(gid):
    print("hello")
```

https://semgrep.dev/playground/new

Demo

Writing SEMGREP RULES

https://semgrep.dev/playground/new

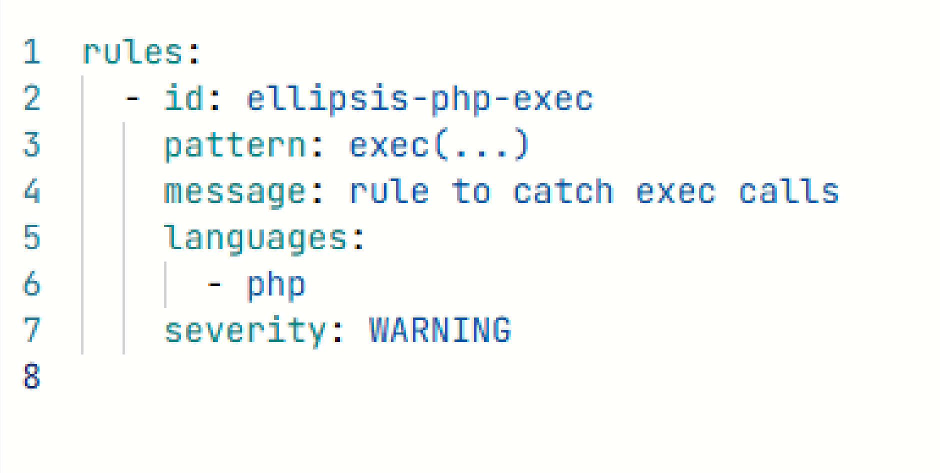
https://aka.ms/seasidessemgrep

Ellipsis

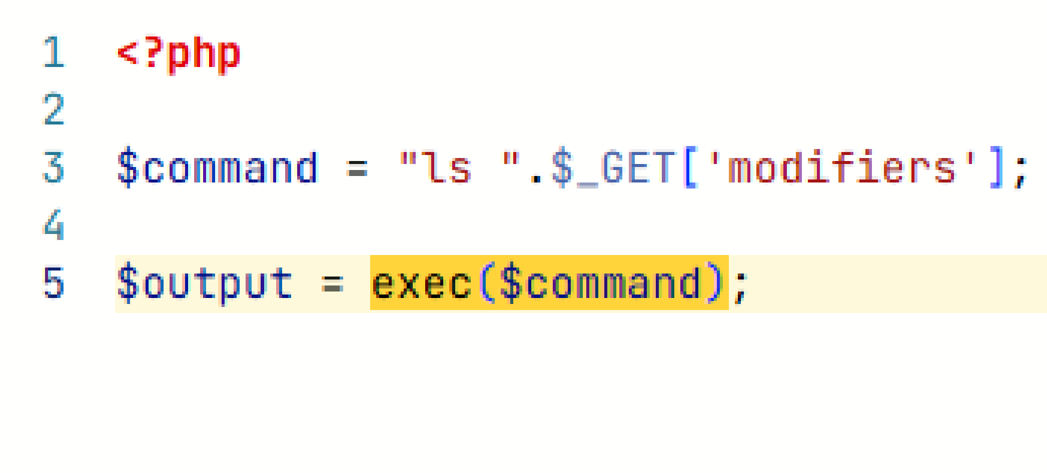
- Ellipsis Operator: "…"
- Matches zero or more items such as arguments, statements, parameters, fields, characters.
- Exercise 1: Find PHP Command Injection Functions
- https://github.com/nikhil1232/sastwithsemgrepseasides25/blob/main/Exercises/Exercise1.php
- Exercise 2: Find call to the unsafe custom function callsystem()
- https://github.com/nikhil1232/sastwithsemgrepseasides25/blob/main/Exercises/Exercise2.php

Ellipsis

- Exercise 1: Find PHP Command Injection Functions
- Solution: https://semgrep.dev/playground/r/7KUg3vd/nikhilsahoo1232_personal_org.ellipsis-php-exec



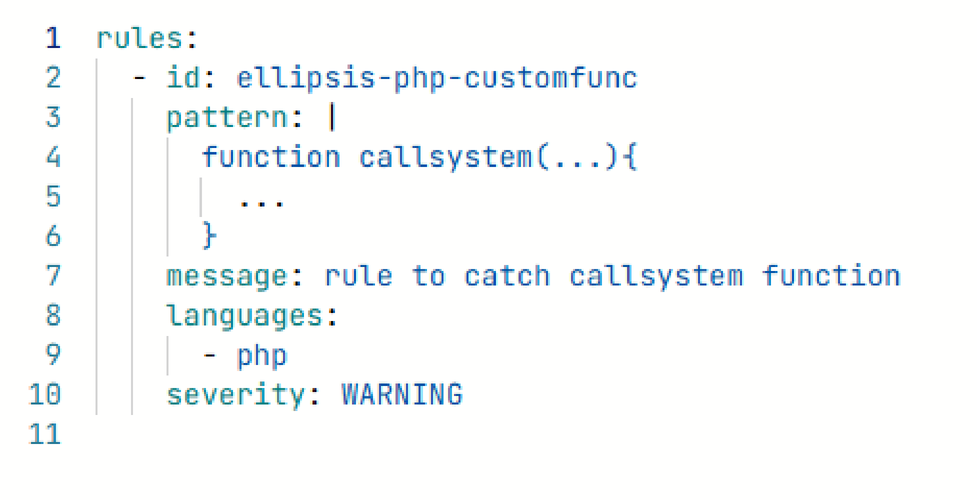
```yaml
rules:
  - id: ellipsis-php-exec
    pattern: exec(...)
    message: rule to catch exec calls
    languages:
      - php
    severity: WARNING

```



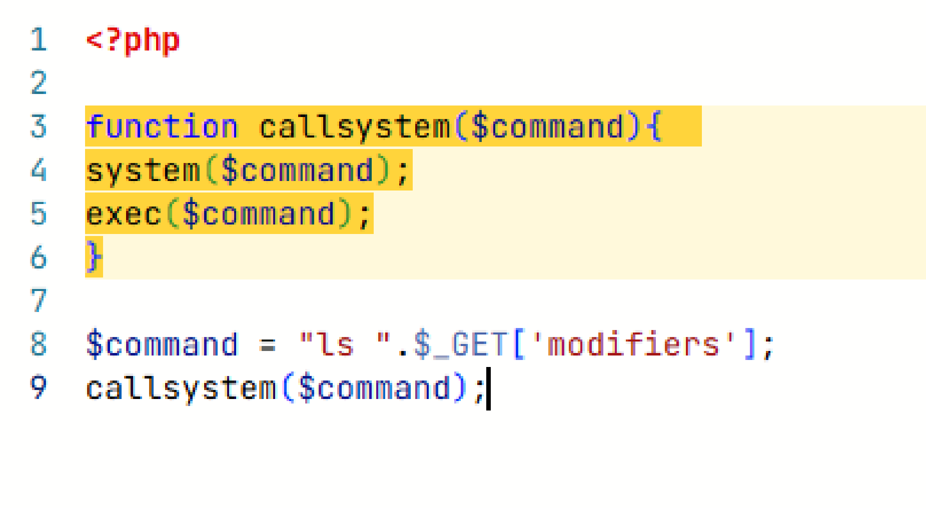
```php
<?php

$command = "ls ".$_GET['modifiers'];

$output = exec($command);
```

Ellipsis

- Exercise 2: Find call to the unsafe custom function callsystem()
- Solution: https://semgrep.dev/playground/r/L1Uqg5Z/nikhilsahoo1232_personal_org.ellipsis-php-customfunc



```yaml
rules:
  - id: ellipsis-php-customfunc
    pattern: |
      function callsystem(...){
        ...
      }
    message: rule to catch callsystem function
    languages:
      - php
    severity: WARNING

```



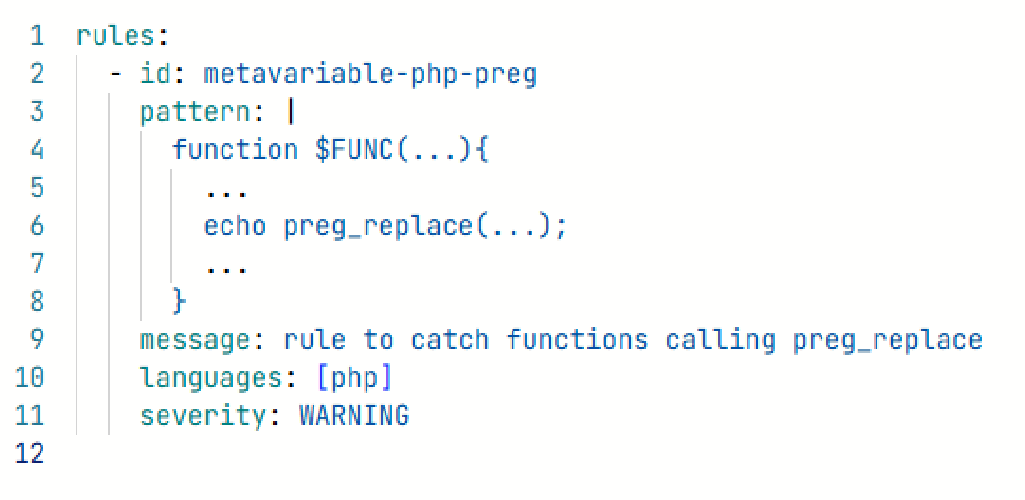
```php
<?php

function callsystem($command){
system($command);
exec($command);
}

$command = "ls ".$_GET['modifiers'];
callsystem($command);
```

Metavariable

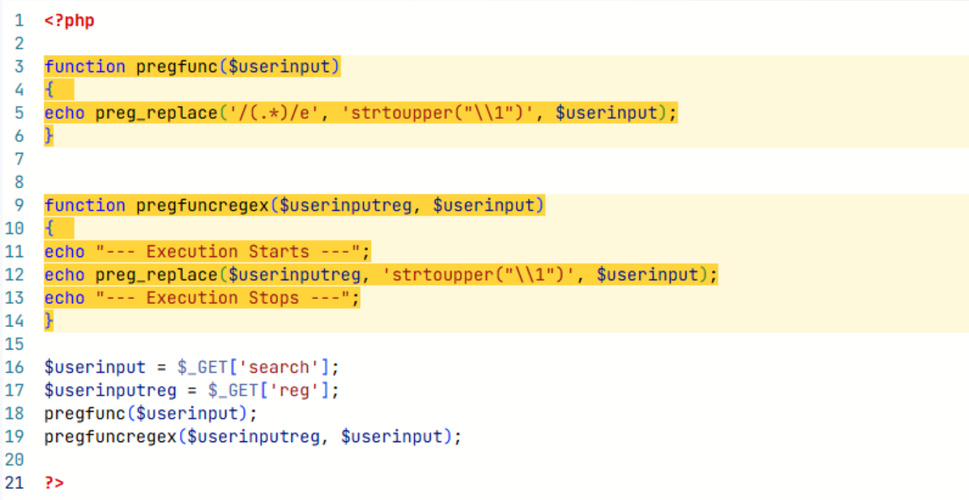
- Metavariable Operator: “$X”
- Way to match code when you don’t know the value or contents ahead of time.
- Should begin with a $ and can only contain uppercase characters, _, or digits.
- Exercise 3: Find all functions calling preg_replace()

Metavariable

- Exercise 3: Find all functions calling preg_replace()
- Solution: https://semgrep.dev/playground/r/8GUQrzD/nikhilsahoo1232_personal_org.metavariable-php-preg



```yaml
rules:
  - id: metavariable-php-preg
    pattern: |
      function $FUNC(...){
        ...
        echo preg_replace(...);
        ...
      }
    message: rule to catch functions calling preg_replace
    languages: [php]
    severity: WARNING

```



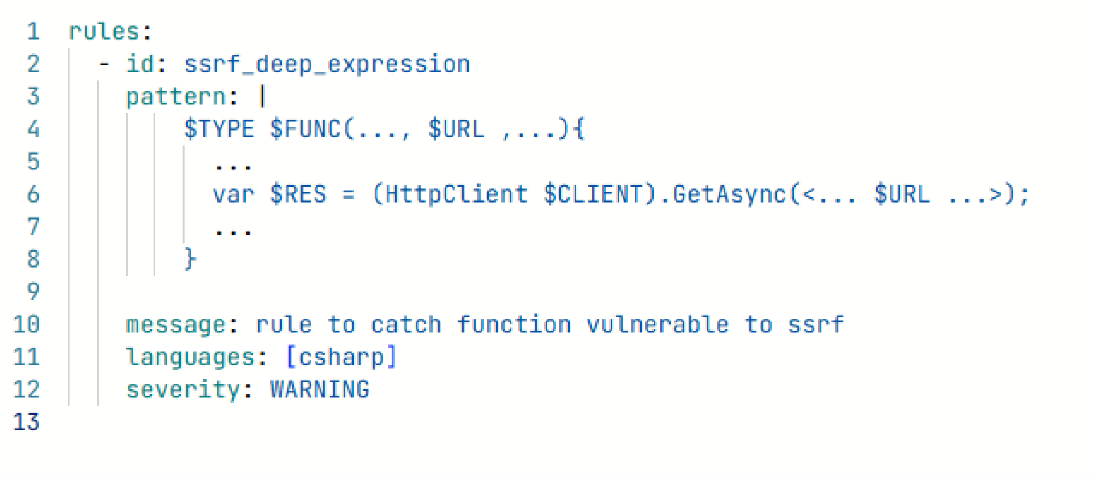
```php
<?php

function pregfunc($userinput)
{
echo preg_replace('/(.*)/e', 'strtoupper("\\1")', $userinput);
}


function pregfuncregex($userinputreg, $userinput)
{
echo "--- Execution Starts ---";
echo preg_replace($userinputreg, 'strtoupper("\\1")', $userinput);
echo "--- Execution Stops ---";
}

$userinput = $_GET['search'];
$userinputreg = $_GET['reg'];
pregfunc($userinput);
pregfuncregex($userinputreg, $userinput);

?>
```

Deep Expression

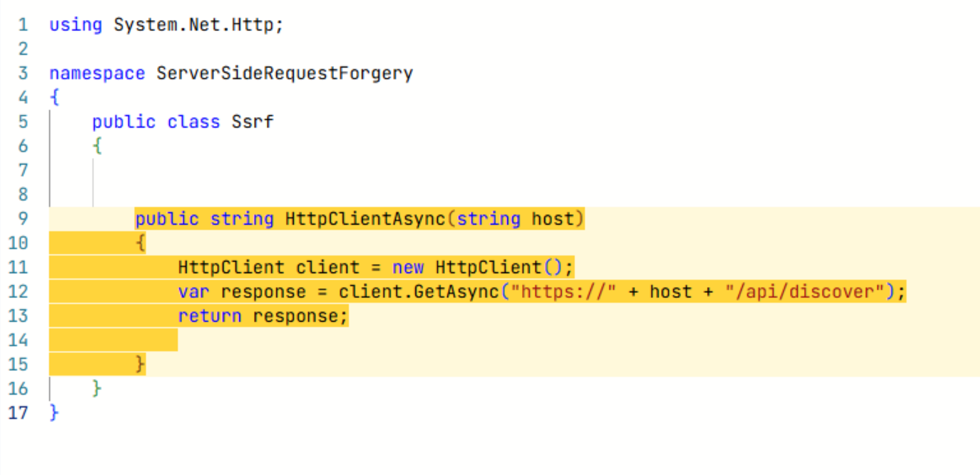
- Deep Expression Operator: “<… [your pattern]…>”
- Used to match an expression that could be deeply nested within another expression
- Exercise 3: Catch the function vulnerable to SSRF

Deep Expression

- Exercise 4: Catch the function vulnerable to SSRF
- Solution: https://semgrep.dev/playground/r/3qUkXjA/nikhilsahoo1232_personal_org.ssrf_deep_expression



```yaml
rules:
  - id: ssrf_deep_expression
    pattern: |
      $TYPE $FUNC(..., $URL ,...){
        ...
        var $RES = (HttpClient $CLIENT).GetAsync(<... $URL ...>);
        ...
      }

    message: rule to catch function vulnerable to ssrf
    languages: [csharp]
    severity: WARNING

```



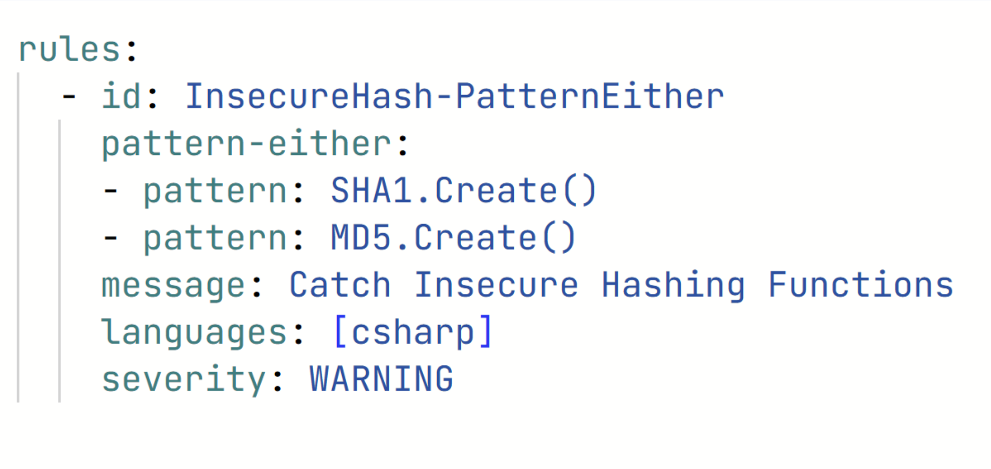
```csharp
using System.Net.Http;

namespace ServerSideRequestForgery
{
    public class Ssrf
    {


        public string HttpClientAsync(string host)
        {
            HttpClient client = new HttpClient();
            var response = client.GetAsync("https://" + host + "/api/discover");
            return response;

        }
    }
}
```

Pattern-Either

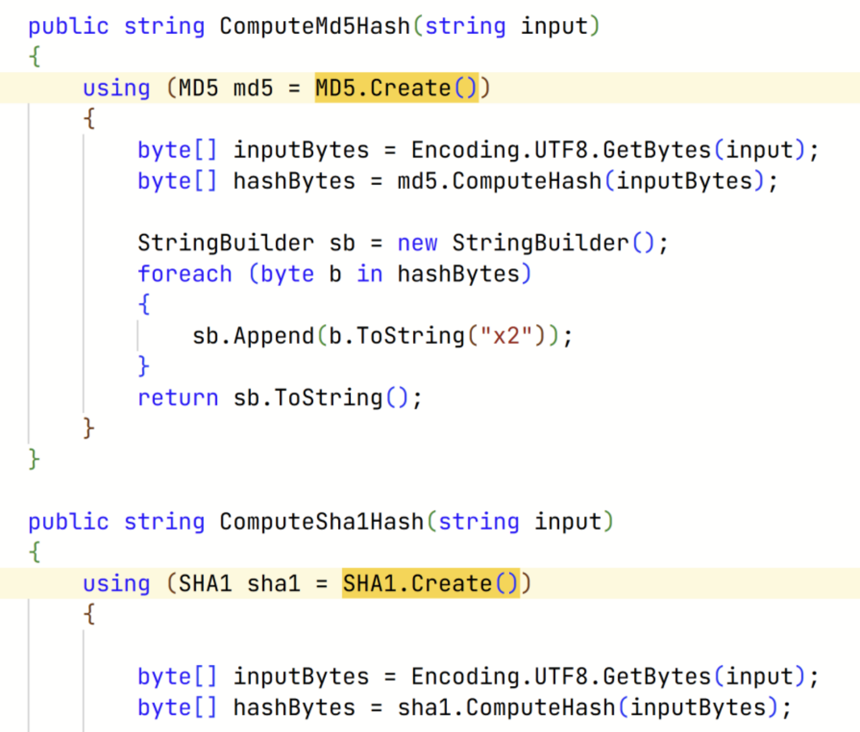
- Performs a logical OR operation on one or more child patterns.
- Helps to combine multiple patterns together where any pattern maybe be true.
- Exercise 4: Catch all MD5 and SHA1 instances
- Solution: https://semgrep.dev/playground/r/JDUNxp8/nikhilsahoo1232_personal_org.insecurehash-patterneither

Pattern-Either

- Exercise 5: Catch all MD5 and SHA1 instances



```yaml
rules:
  - id: InsecureHash-PatternEither
    pattern-either:
    - pattern: SHA1.Create()
    - pattern: MD5.Create()
    message: Catch Insecure Hashing Functions
    languages: [csharp]
    severity: WARNING
```



```csharp
public string ComputeMd5Hash(string input)
{
    using (MD5 md5 = MD5.Create())
    {
        byte[] inputBytes = Encoding.UTF8.GetBytes(input);
        byte[] hashBytes = md5.ComputeHash(inputBytes);

        StringBuilder sb = new StringBuilder();
        foreach (byte b in hashBytes)
        {
            sb.Append(b.ToString("x2"));
        }
        return sb.ToString();
    }
}

public string ComputeSha1Hash(string input)
{
    using (SHA1 sha1 = SHA1.Create())
    {

        byte[] inputBytes = Encoding.UTF8.GetBytes(input);
        byte[] hashBytes = sha1.ComputeHash(inputBytes);
```

Patterns

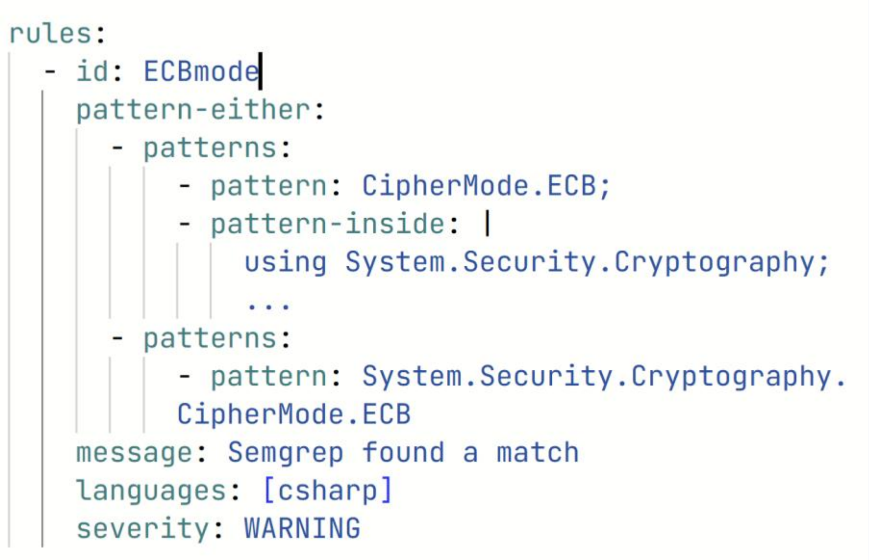
- Performs a logical AND operation on one or more child patterns.
- Helps to combine multiple patterns together where all must be true.

Pattern-Inside

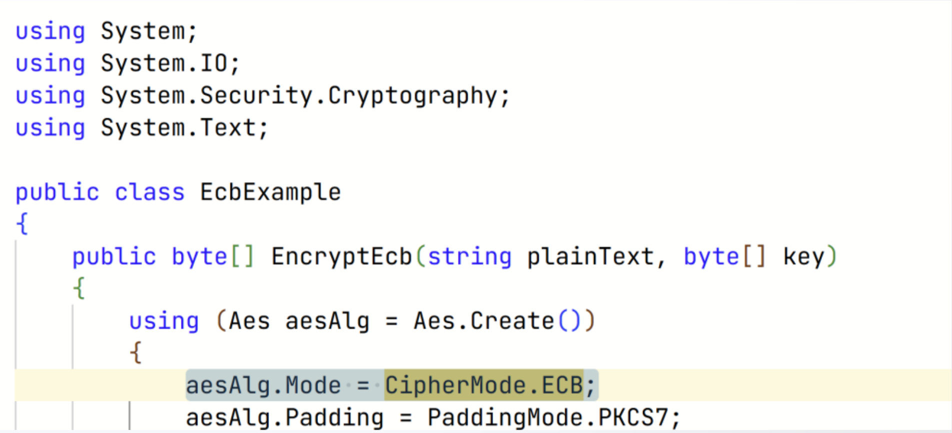
- Keeps matched findings that reside within its expression.
- Helps to find code inside other pieces of code like functions or if blocks.
- Exercise 6: Catch ECB instances
- Solution: https://semgrep.dev/playground/r/5rUd1Ag/nikhilsahoo1232_personal_org.ecbmode

Pattern-Inside

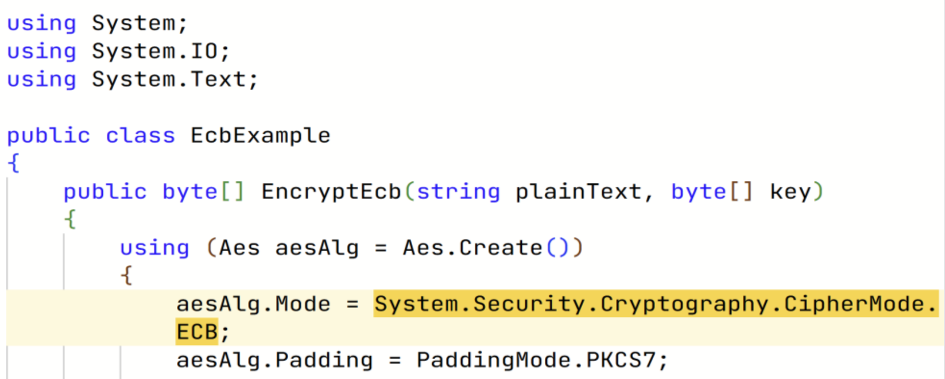
- Exercise 6: Catch ECB instances



```
rules:
  - id: ECBmode
    pattern-either:
      - patterns:
          - pattern: CipherMode.ECB;
          - pattern-inside: |
              using System.Security.Cryptography;
              ...
      - patterns:
          - pattern: System.Security.Cryptography.
        CipherMode.ECB
    message: Semgrep found a match
    languages: [csharp]
    severity: WARNING
```



```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

public class EcbExample
{
    public byte[] EncryptEcb(string plainText, byte[] key)
    {
        using (Aes aesAlg = Aes.Create())
        {
            aesAlg.Mode = CipherMode.ECB;
            aesAlg.Padding = PaddingMode.PKCS7;
```



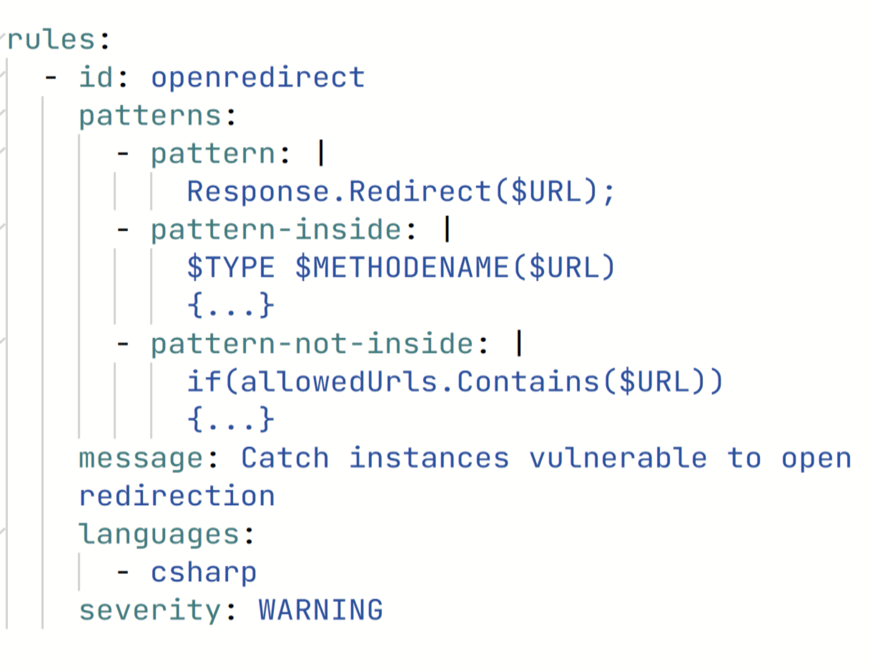
```
using System;
using System.IO;
using System.Text;

public class EcbExample
{
    public byte[] EncryptEcb(string plainText, byte[] key)
    {
        using (Aes aesAlg = Aes.Create())
        {
            aesAlg.Mode = System.Security.Cryptography.CipherMode.
            ECB;
            aesAlg.Padding = PaddingMode.PKCS7;
```

Pattern-Not-Inside

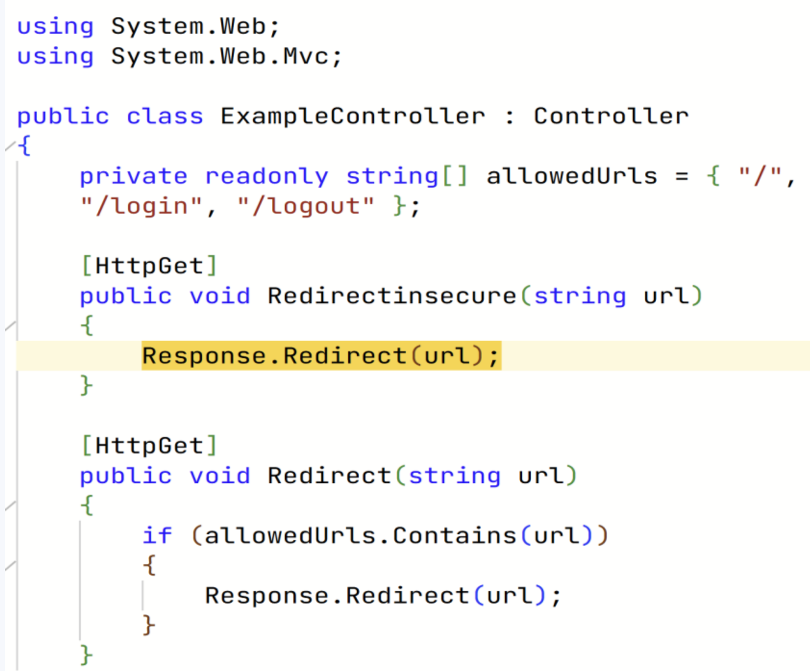
- Keeps matched findings that reside within its expression.
- Helps to find code inside other pieces of code like functions or if blocks.
- Exercise 7: Catch functions vulnerable to open redirection
- Solution: https://semgrep.dev/playground/r/GdUve6E/nikhilsahoo1232_personal_org.openredirect

Pattern-Not-Inside

- Exercise 7: Catch functions vulnerable to open redirection



```yaml
rules:
  - id: openredirect
    patterns:
      - pattern: |
          Response.Redirect($URL);
      - pattern-inside: |
          $TYPE $METHODENAME($URL)
          {...}
      - pattern-not-inside: |
          if(allowedUrls.Contains($URL))
          {...}
    message: Catch instances vulnerable to open
    redirection
    languages:
      - csharp
    severity: WARNING
```



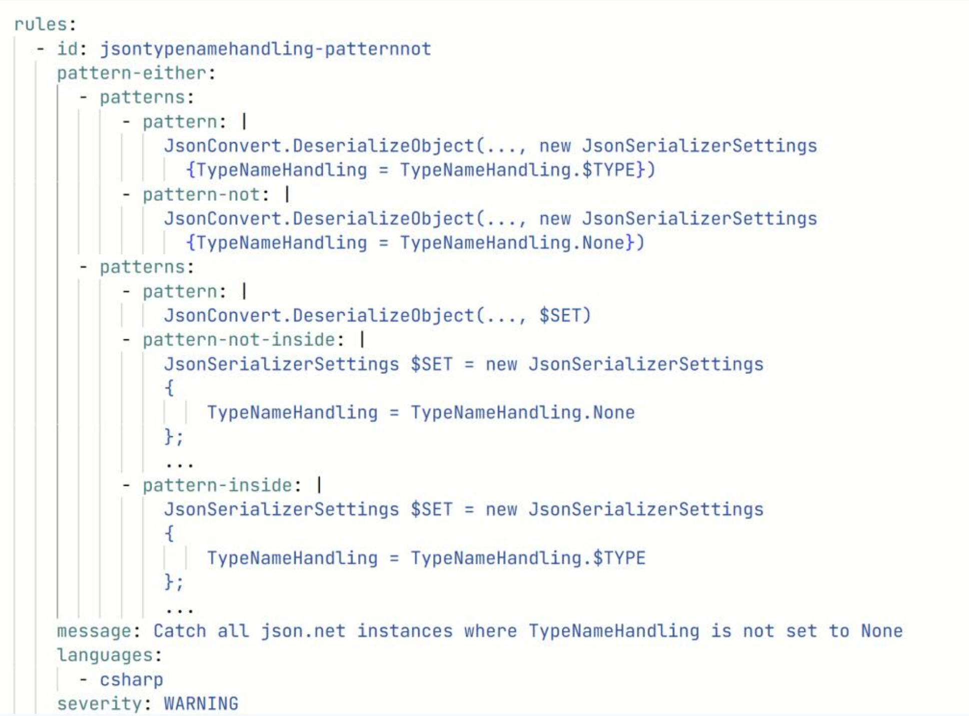
```csharp
using System.Web;
using System.Web.Mvc;

public class ExampleController : Controller
{
    private readonly string[] allowedUrls = { "/",
    "/login", "/logout" };

    [HttpGet]
    public void Redirectinsecure(string url)
    {
        Response.Redirect(url);
    }

    [HttpGet]
    public void Redirect(string url)
    {
        if (allowedUrls.Contains(url))
        {
            Response.Redirect(url);
        }
    }
```

Pattern-Not

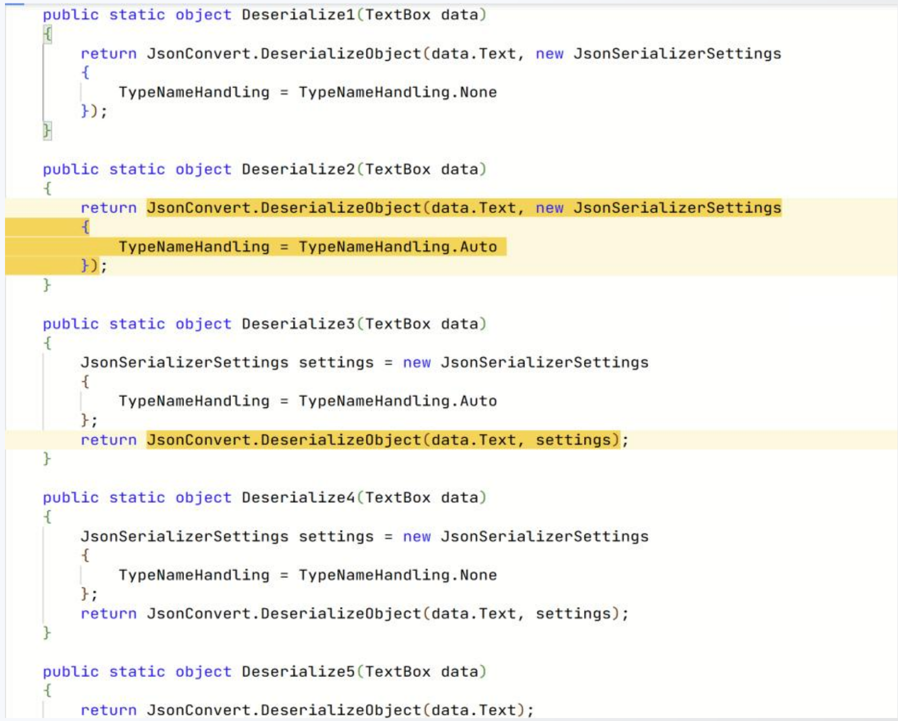
- Opposite of the pattern operator.
- Helps to find code that does not match its expression.
- Useful for eliminating common false positives.
- Exercise 8: Catch all instances vulnerable to Json.net deserialization
- Solution: https://semgrep.dev/playground/r/X5UQy4l/nikhilsahoo1232_personal_org.jsontypenamehandling-patternnot

Pattern-Not

- Exercise 8: Catch all instances vulnerable to Json.net deserialization



```yaml
rules:
  - id: jsontypenamehandling-patternnot
    pattern-either:
      - patterns:
        - pattern: |
            JsonConvert.DeserializeObject(..., new JsonSerializerSettings
              {TypeNameHandling = TypeNameHandling.$TYPE})
        - pattern-not: |
            JsonConvert.DeserializeObject(..., new JsonSerializerSettings
              {TypeNameHandling = TypeNameHandling.None})
      - patterns:
        - pattern: |
            JsonConvert.DeserializeObject(..., $SET)
        - pattern-not-inside: |
            JsonSerializerSettings $SET = new JsonSerializerSettings
            {
                TypeNameHandling = TypeNameHandling.None
            };
            ...
        - pattern-inside: |
            JsonSerializerSettings $SET = new JsonSerializerSettings
            {
                TypeNameHandling = TypeNameHandling.$TYPE
            };
            ...
    message: Catch all json.net instances where TypeNameHandling is not set to None
    languages:
      - csharp
    severity: WARNING
```



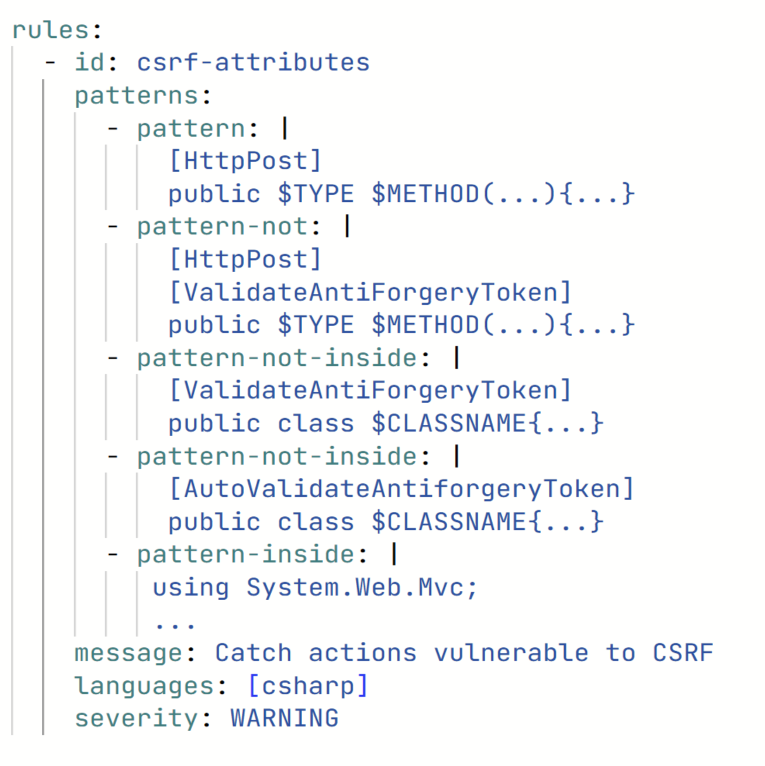
```csharp
public static object Deserialize1(TextBox data)
{
    return JsonConvert.DeserializeObject(data.Text, new JsonSerializerSettings
    {
        TypeNameHandling = TypeNameHandling.None
    });
}

public static object Deserialize2(TextBox data)
{
    return JsonConvert.DeserializeObject(data.Text, new JsonSerializerSettings
    {
        TypeNameHandling = TypeNameHandling.Auto
    });
}

public static object Deserialize3(TextBox data)
{
    JsonSerializerSettings settings = new JsonSerializerSettings
    {
        TypeNameHandling = TypeNameHandling.Auto
    };
    return JsonConvert.DeserializeObject(data.Text, settings);
}

public static object Deserialize4(TextBox data)
{
    JsonSerializerSettings settings = new JsonSerializerSettings
    {
        TypeNameHandling = TypeNameHandling.None
    };
    return JsonConvert.DeserializeObject(data.Text, settings);
}

public static object Deserialize5(TextBox data)
{
    return JsonConvert.DeserializeObject(data.Text);
```

Catching Attributes

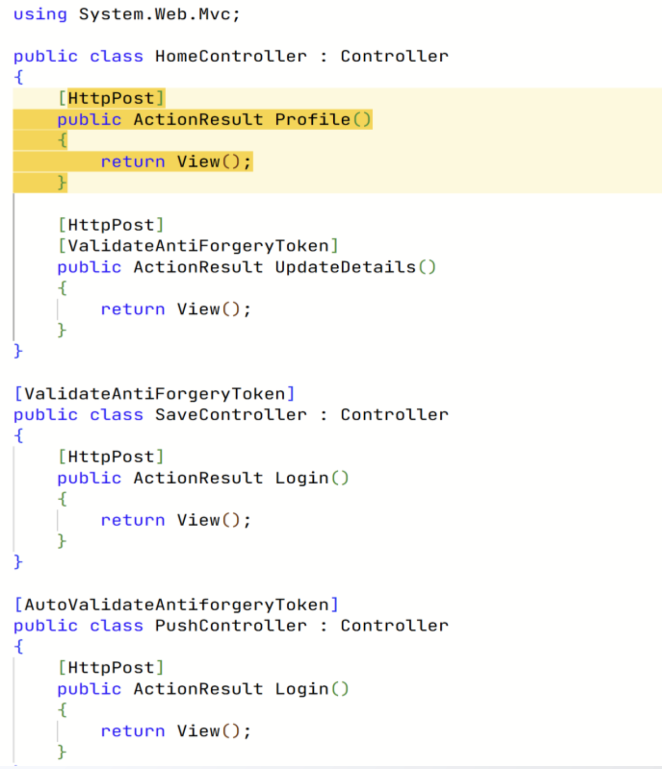
- Exercise 9: Find all functions vulnerable to CSRF
- Solution: https://semgrep.dev/playground/r/r6UyK1g/nikhilsahoo1232_personal_org.csrf-attributes
- Exercise 10: Broken Access Control
- Solution: https://semgrep.dev/playground/r/bwUb3ZD/nikhilsahoo1232_personal_org.possible-anonymous-action

Catching Attributes

- Exercise 9 : Find all functions vulnerable to CSRF



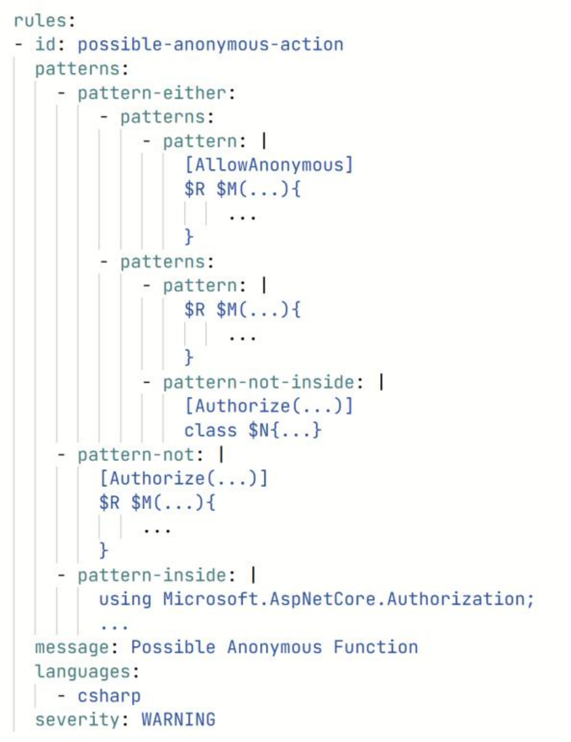
```yaml
rules:
  - id: csrf-attributes
    patterns:
      - pattern: |
          [HttpPost]
          public $TYPE $METHOD(...){...}
      - pattern-not: |
          [HttpPost]
          [ValidateAntiForgeryToken]
          public $TYPE $METHOD(...){...}
      - pattern-not-inside: |
          [ValidateAntiForgeryToken]
          public class $CLASSNAME{...}
      - pattern-not-inside: |
          [AutoValidateAntiforgeryToken]
          public class $CLASSNAME{...}
      - pattern-inside: |
         using System.Web.Mvc;
         ...
    message: Catch actions vulnerable to CSRF
    languages: [csharp]
    severity: WARNING
```



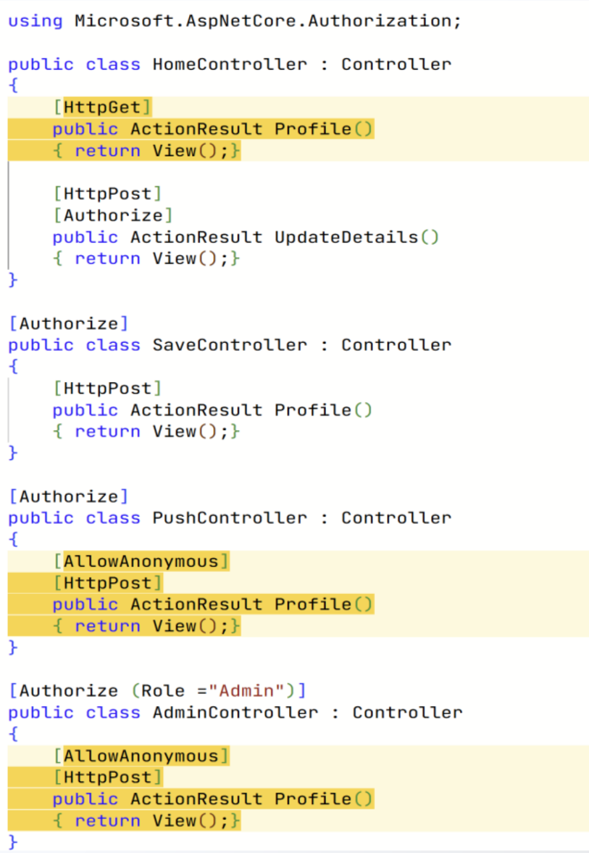
```csharp
using System.Web.Mvc;

public class HomeController : Controller
{
    [HttpPost]
    public ActionResult Profile()
    {
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult UpdateDetails()
    {
        return View();
    }
}

[ValidateAntiForgeryToken]
public class SaveController : Controller
{
    [HttpPost]
    public ActionResult Login()
    {
        return View();
    }
}

[AutoValidateAntiforgeryToken]
public class PushController : Controller
{
    [HttpPost]
    public ActionResult Login()
    {
        return View();
    }
```

Catching Attributes

- Exercise 10 : Broken Access Control



```
rules:
- id: possible-anonymous-action
  patterns:
    - pattern-either:
        - patterns:
            - pattern: |
                [AllowAnonymous]
                $R $M(...){
                    ...
                }
        - patterns:
            - pattern: |
                $R $M(...){
                    ...
                }
            - pattern-not-inside: |
                [Authorize(...)]
                class $N{...}
    - pattern-not: |
        [Authorize(...)]
        $R $M(...){
            ...
        }
    - pattern-inside: |
        using Microsoft.AspNetCore.Authorization;
        ...
  message: Possible Anonymous Function
  languages:
    - csharp
  severity: WARNING
```



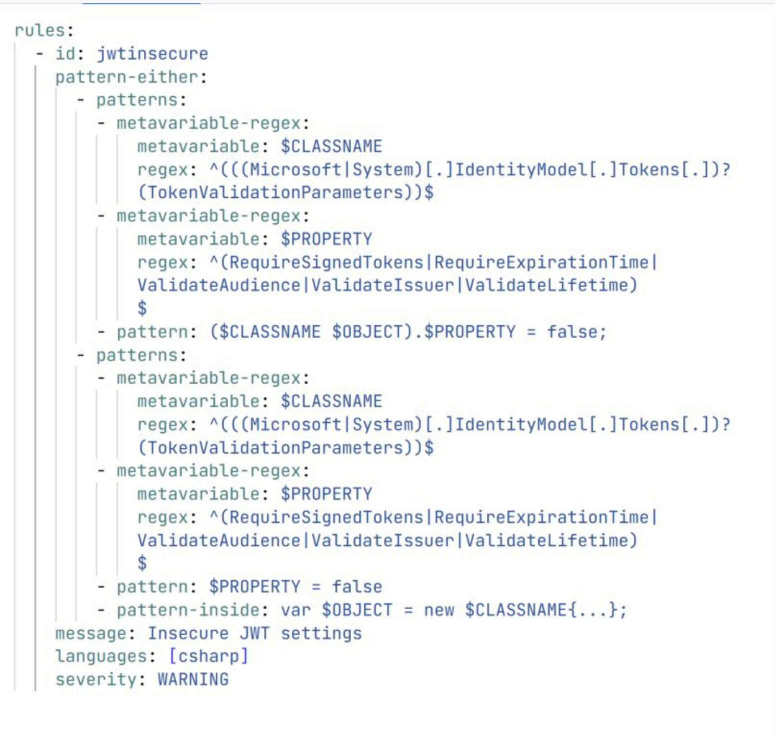
```
using Microsoft.AspNetCore.Authorization;

public class HomeController : Controller
{
    [HttpGet]
    public ActionResult Profile()
    { return View();}

    [HttpPost]
    [Authorize]
    public ActionResult UpdateDetails()
    { return View();}
}

[Authorize]
public class SaveController : Controller
{
    [HttpPost]
    public ActionResult Profile()
    { return View();}
}

[Authorize]
public class PushController : Controller
{
    [AllowAnonymous]
    [HttpPost]
    public ActionResult Profile()
    { return View();}
}

[Authorize (Role ="Admin")]
public class AdminController : Controller
{
    [AllowAnonymous]
    [HttpPost]
    public ActionResult Profile()
    { return View();}
}
```

Metavariable regex

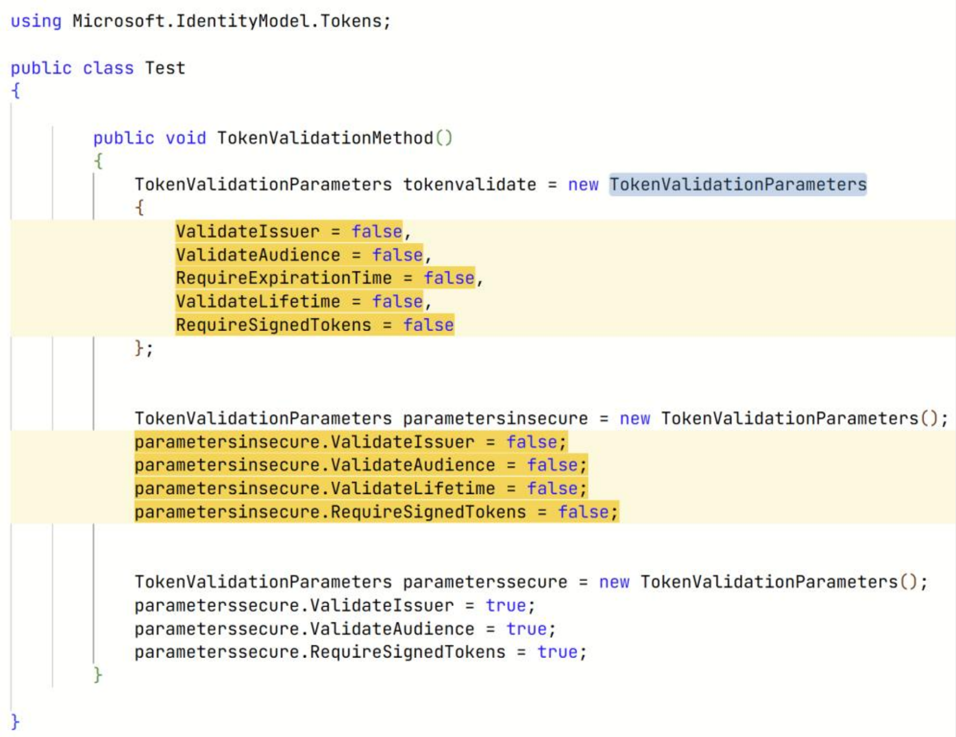
- Searches metavariables for a PCRE2 regular expression
- Exercise 11 : Catch the vulnerable settings responsible for insecure token validation
- Solution: https://semgrep.dev/playground/r/6JUv9R2/nikhilsahoo1232_personal_org.jwtinsecure

Metavariable regex

- Exercise 11 : Catch the vulnerable settings responsible for insecure token validation



```yaml
rules:
  - id: jwtinsecure
    pattern-either:
      - patterns:
        - metavariable-regex:
            metavariable: $CLASSNAME
            regex: ^(((Microsoft|System)[.]IdentityModel[.]Tokens[.])?
            (TokenValidationParameters))$
        - metavariable-regex:
            metavariable: $PROPERTY
            regex: ^(RequireSignedTokens|RequireExpirationTime|
            ValidateAudience|ValidateIssuer|ValidateLifetime)
            $
        - pattern: ($CLASSNAME $OBJECT).$PROPERTY = false;
      - patterns:
        - metavariable-regex:
            metavariable: $CLASSNAME
            regex: ^(((Microsoft|System)[.]IdentityModel[.]Tokens[.])?
            (TokenValidationParameters))$
        - metavariable-regex:
            metavariable: $PROPERTY
            regex: ^(RequireSignedTokens|RequireExpirationTime|
            ValidateAudience|ValidateIssuer|ValidateLifetime)
            $
        - pattern: $PROPERTY = false
        - pattern-inside: var $OBJECT = new $CLASSNAME{...};
    message: Insecure JWT settings
    languages: [csharp]
    severity: WARNING
```



```csharp
using Microsoft.IdentityModel.Tokens;

public class Test
{

        public void TokenValidationMethod()
        {
            TokenValidationParameters tokenvalidate = new TokenValidationParameters
            {
                ValidateIssuer = false,
                ValidateAudience = false,
                RequireExpirationTime = false,
                ValidateLifetime = false,
                RequireSignedTokens = false
            };


            TokenValidationParameters parametersinsecure = new TokenValidationParameters();
            parametersinsecure.ValidateIssuer = false;
            parametersinsecure.ValidateAudience = false;
            parametersinsecure.ValidateLifetime = false;
            parametersinsecure.RequireSignedTokens = false;


            TokenValidationParameters parameterssecure = new TokenValidationParameters();
            parameterssecure.ValidateIssuer = true;
            parameterssecure.ValidateAudience = true;
            parameterssecure.RequireSignedTokens = true;
        }

}
```

Metavariable comparison

- Compares metavariables against a basic comparison expression.
- Useful for filtering results based on a metavariable's numeric value.
- Exercise 12: Find all instances of persistent cookies (set to more than 5 mins)
- Solution: https://semgrep.dev/playground/r/gxU37vX/nikhilsahoo1232_personal_org.persistent-cookie

Taint Analysis

- Data-flow analysis that tracks the flow of untrusted, or tainted data throughout the body of a function or method
- Exercise 13: Find all instances of XML unsafe parsing.
- Solution: https://semgrep.dev/playground/r/DbU6boq/nikhilsahoo1232_personal_org.xmlreadersettings-unsafe-parser-override

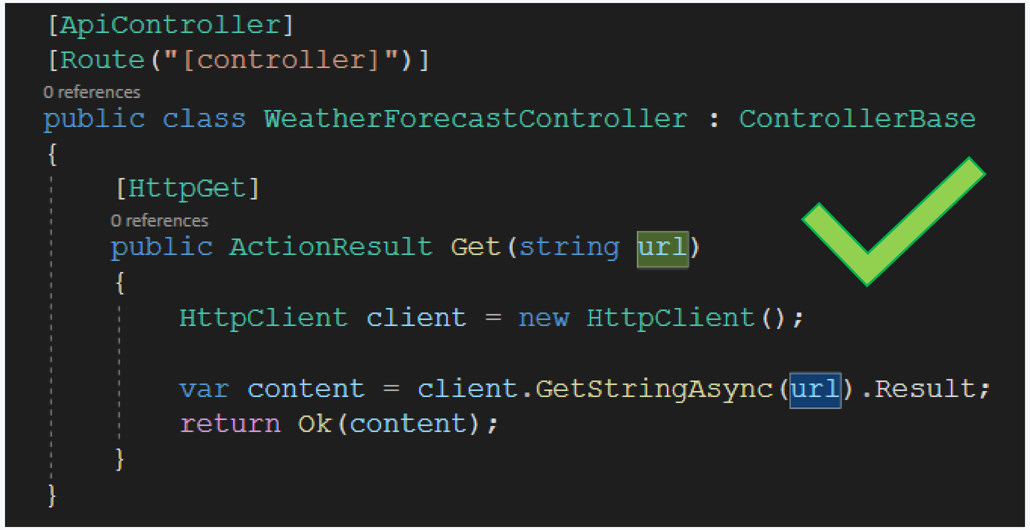
Pattern-sanitizers

- Exercise 14: Path.Combine() Path Traversal
- Solution: https://semgrep.dev/playground/r/WAUWK5l/nikhilsahoo1232_personal_org.modified-unsafe-path-combine

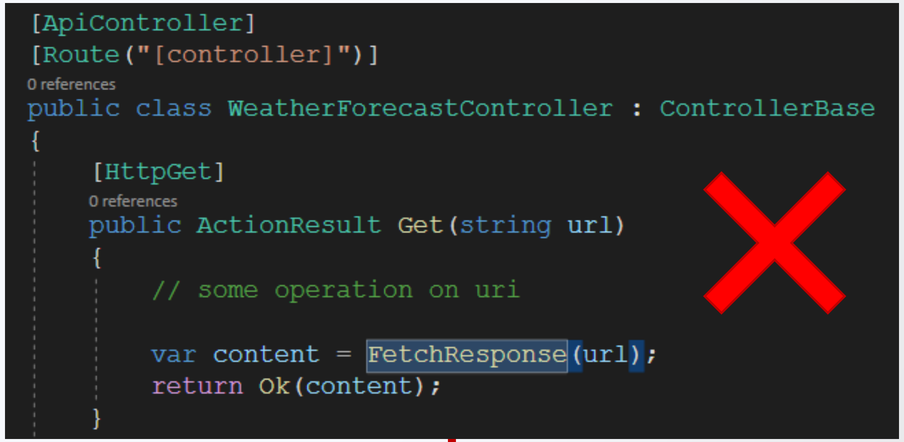
Generic Pattern matching

- Exercise 15: Debugging Enabled(Web.Config)
- Solution: https://semgrep.dev/playground/r/0oULzk5/nikhilsahoo1232_personal_org.generic-debuggingenabled

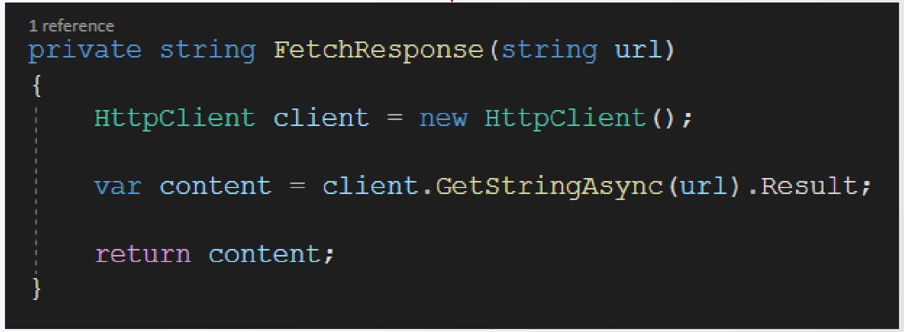
Limitation



```
[ApiController]
[Route("[controller]")]
0 references
public class WeatherForecastController : ControllerBase
{
    [HttpGet]
    0 references
    public ActionResult Get(string url)
    {
        HttpClient client = new HttpClient();

        var content = client.GetStringAsync(url).Result;
        return Ok(content);
    }
}
```



```
[ApiController]
[Route("[controller]")]
0 references
public class WeatherForecastController : ControllerBase
{
    [HttpGet]
    0 references
    public ActionResult Get(string url)
    {
        // some operation on uri

        var content = FetchResponse(url);
        return Ok(content);
    }
```

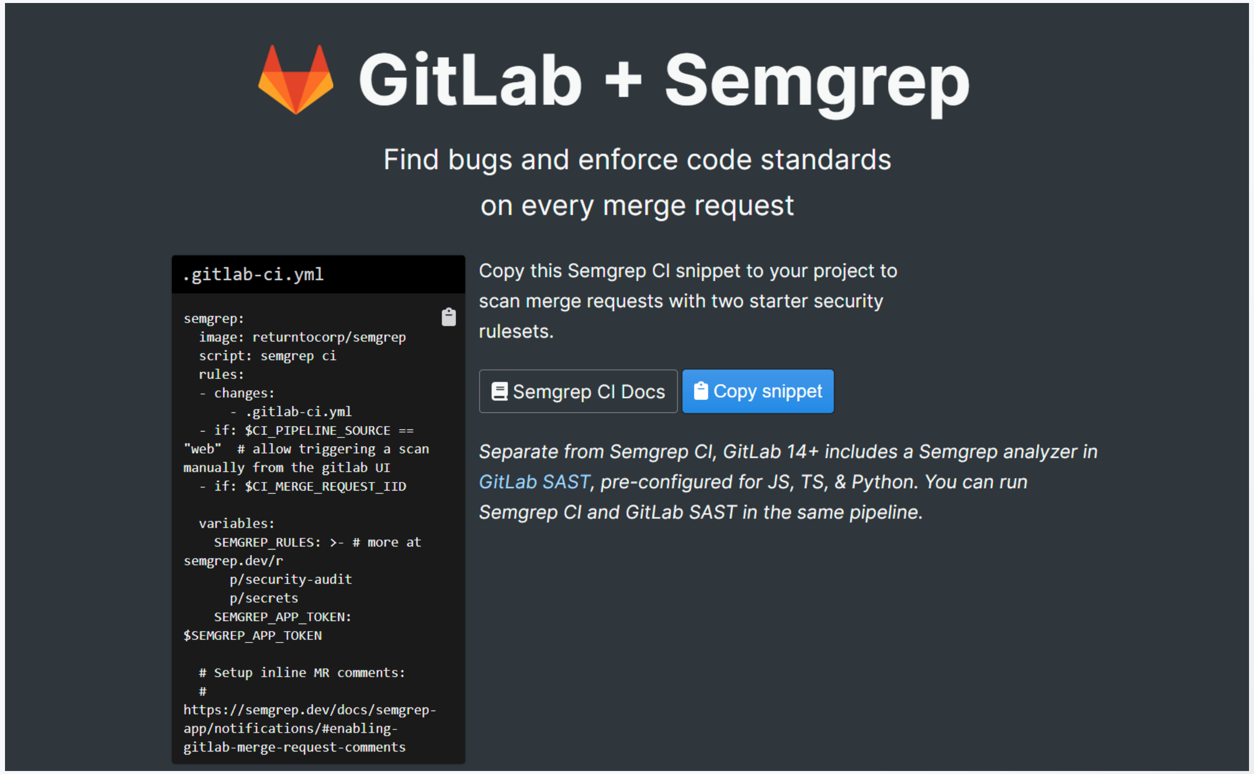


```
1 reference
private string FetchResponse(string url)
{
    HttpClient client = new HttpClient();

    var content = client.GetStringAsync(url).Result;

    return content;
}
```

Multi File Analysis:

Semgrep Pro Engine

Semgrep on Push



GitLab + Semgrep

Find bugs and enforce code standards
on every merge request

.gitlab-ci.yml

```
semgrep:
  image: returntocorp/semgrep
  script: semgrep ci
  rules:
  - changes:
      - .gitlab-ci.yml
  - if: $CI_PIPELINE_SOURCE ==
"web"  # allow triggering a scan
manually from the gitlab UI
  - if: $CI_MERGE_REQUEST_IID

  variables:
    SEMGREP_RULES: >- # more at
semgrep.dev/r
      p/security-audit
      p/secrets
    SEMGREP_APP_TOKEN:
$SEMGREP_APP_TOKEN

  # Setup inline MR comments:
  #
https://semgrep.dev/docs/semgrep-
app/notifications/#enabling-
gitlab-merge-request-comments
```

Copy this Semgrep CI snippet to your project to scan merge requests with two starter security rulesets.

Separate from Semgrep CI, GitLab 14+ includes a Semgrep analyzer in GitLab SAST, pre-configured for JS, TS, & Python. You can run Semgrep CI and GitLab SAST in the same pipeline.

https://semgrep.dev/docs/semgrep-ci/overview/

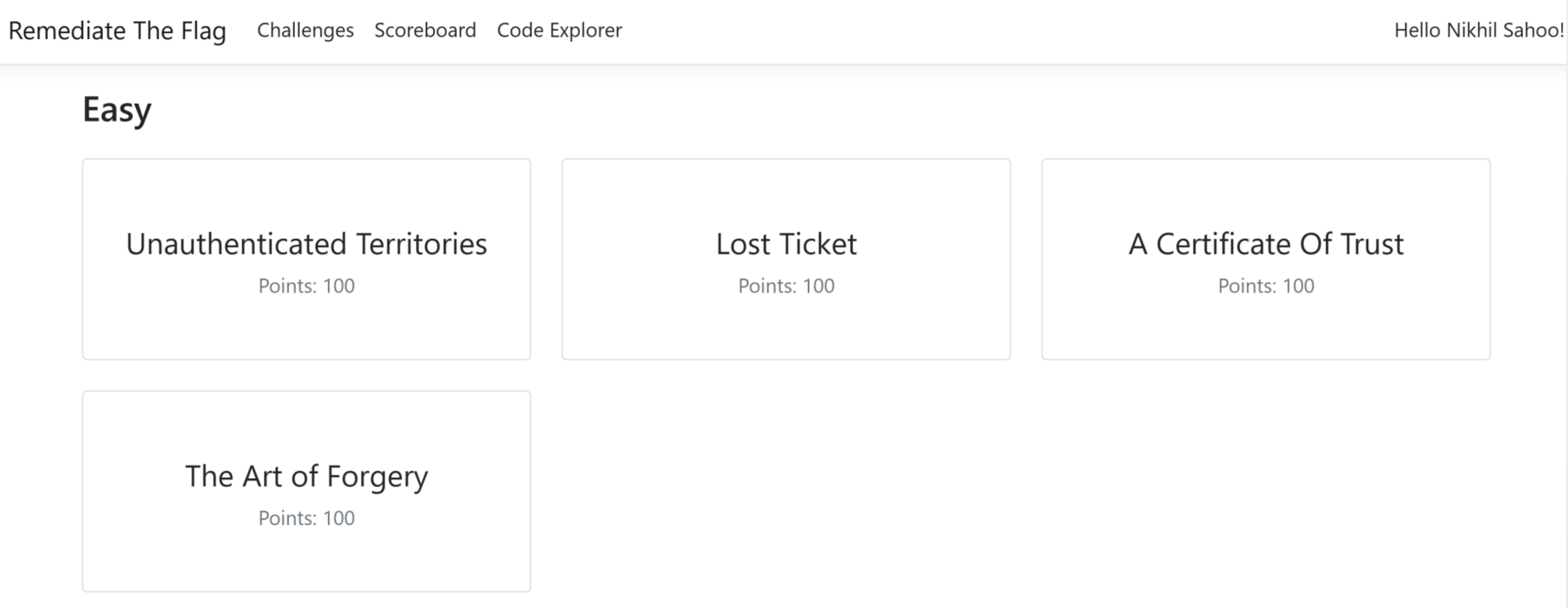
Tools

- Semgrep: https://github.com/returntocorp/semgrep
- Semgrep External Rules: https://github.com/returntocorp/semgrep-rules
- Semgrep Playground: https://semgrep.dev/playground
- CodeQL: https://codeql.github.com/
- Devskim: https://github.com/microsoft/DevSkim

References

- https://semgrep.dev/docs/
- https://www.youtube.com/watch?v=kb8oo7Wyk84
- https://youtube.com/watch?v=O5mh8j7-An8
- https://semgrep.dev/playground/
- https://rules.sonarsource.com/
- https://semgrep.dev/blog/2021/semgrep-a-static-analysis-journey
- https://github.com/returntocorp/semgrep-rules

RTF



Remediate The Flag

Hello Nikhil Sahoo!

Easy

Unauthenticated Territories
Points: 100

Lost Ticket
Points: 100

A Certificate Of Trust
Points: 100

The Art of Forgery
Points: 100

https://aka.ms/rtfseasides

Thank You