Bug Hunting with SAST

Thank You to Our Sponsors and Hosts!

OWASP NEW ZEALAND
owasp.org.nz

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

DATACOM

QUANTUM SECURITY

ATTACK™
DEFEND Group Company

aura INFORMATION SECURITY | POWERED BY KORDIA

INSOMNIA
SECURITY SPECIALISTS :: REST SECURED

myob

Pulse Security
www.pulsesecurity.co.nz

SECURE CODE WARRIOR

VOCUS

ZX SECURITY

flux

internetnz

LATERAL SECURITY

secInit security initiative

SEQA™
Information Security

Without them, this Conference couldn't happen

Whoami

Static
Application
Security
Testing

Dynamic
Application
Security
Testing

XML

eXternal

Entity

```xml
<?xml version="1.0"?>
<!DOCTYPE data [
<!ELEMENT data (#PCDATA)>
<!ENTITY file SYSTEM "file:///etc/passwd">
]>
<data>&file;</data>
```

```
<?xml version="1.0"?>
<!DOCTYPE data [
<!ELEMENT data (#PCDATA)>
<!ENTITY file SYSTEM "file:///etc/passwd">
]>
<data>&file;</data>
```

DTD

```
// Read an XML file
XmlReader reader = XmlReader.Create("EVIL.xml", settings);
```

```
// Read an XML file
XmlReader reader = XmlReader.Create("EVIL.xml", settings);
```

```
System.Xml.XmlException:
For security reasons DTD is prohibited in this XML document.
```

```
// Create some settings for the XML Reader
XmlReaderSettings settings = new XmlReaderSettings();
// Create a URL Resolver
XmlUrlResolver resolver = new XmlUrlResolver();
settings.XmlResolver = resolver;
// Set the max characters
settings.MaxCharactersFromEntities = 0;
settings.DtdProcessing = DtdProcessing.Parse;
// Read an XML file
XmlReader reader = XmlReader.Create("EVIL.xml", settings);
```

```
// Create some settings for the XML Reader
XmlReaderSettings settings = new XmlReaderSettings();
// Create a URL Resolver
XmlUrlResolver resolver = new XmlUrlResolver();  → LFI
settings.XmlResolver = resolver;
// Set the max characters
settings.MaxCharactersFromEntities = 0;  → DOS
settings.DtdProcessing = DtdProcessing.Parse;
// Read an XML file
XmlReader reader = XmlReader.Create("EVIL.xml", settings);
```

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
<!ENTITY lol "lol">
<!ELEMENT lolz (#PCDATA)>
<!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
<!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
<!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
<!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
<!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
<!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
<!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
<!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
<!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

```xml
<?xml version="1.0"?>
<!DOCTYPE lolz [
<!ENTITY lol "lol">
<!ELEMENT lolz (#PCDATA)>
<!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
<!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
<!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
<!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
<!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
<!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
<!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
<!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
<!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

10

100

1000

10^4

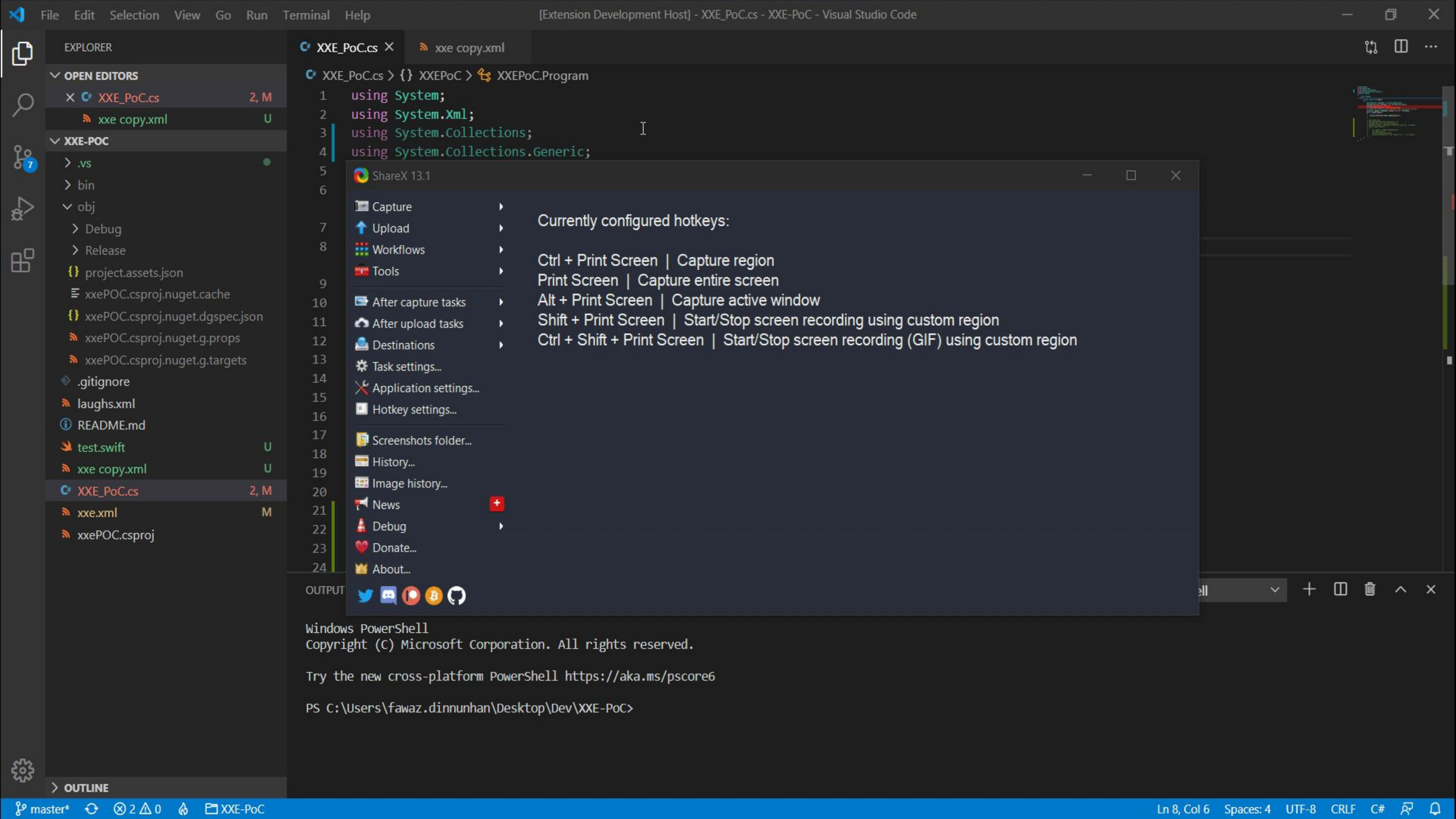
...

10^9

```
(.XmlResolver(|\\s)=(|\\s))((?!null)\\w+)
```

```
(.XmlResolver(l\\s)=(l\\s))((?!null)\\w+)
```

Resolver is not null



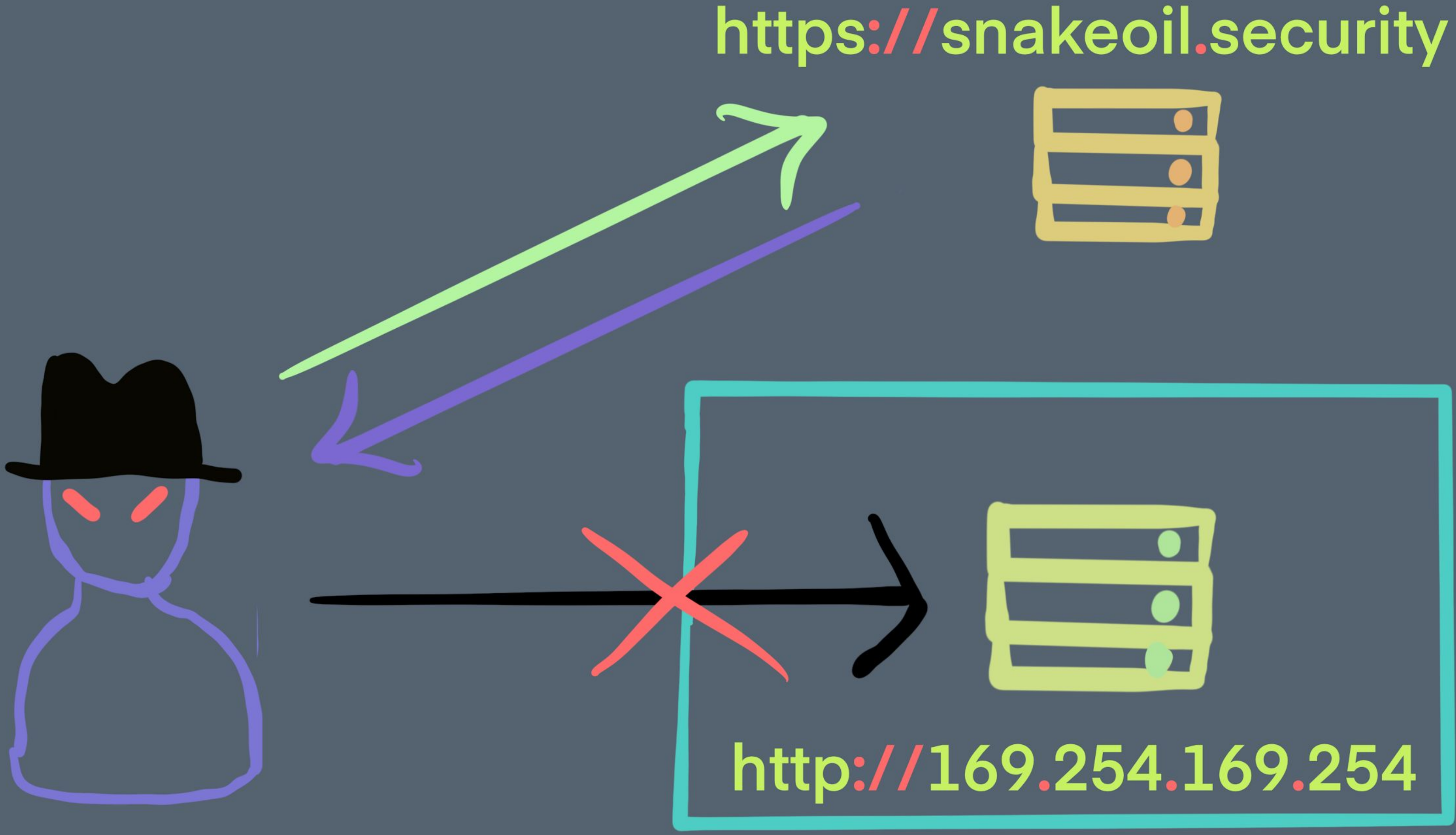
```
using System;
using System.Xml;
using System.Collections;
using System.Collections.Generic;
```

Currently configured hotkeys:

Ctrl + Print Screen | Capture region
Print Screen | Capture entire screen
Alt + Print Screen | Capture active window
Shift + Print Screen | Start/Stop screen recording using custom region
Ctrl + Shift + Print Screen | Start/Stop screen recording (GIF) using custom region

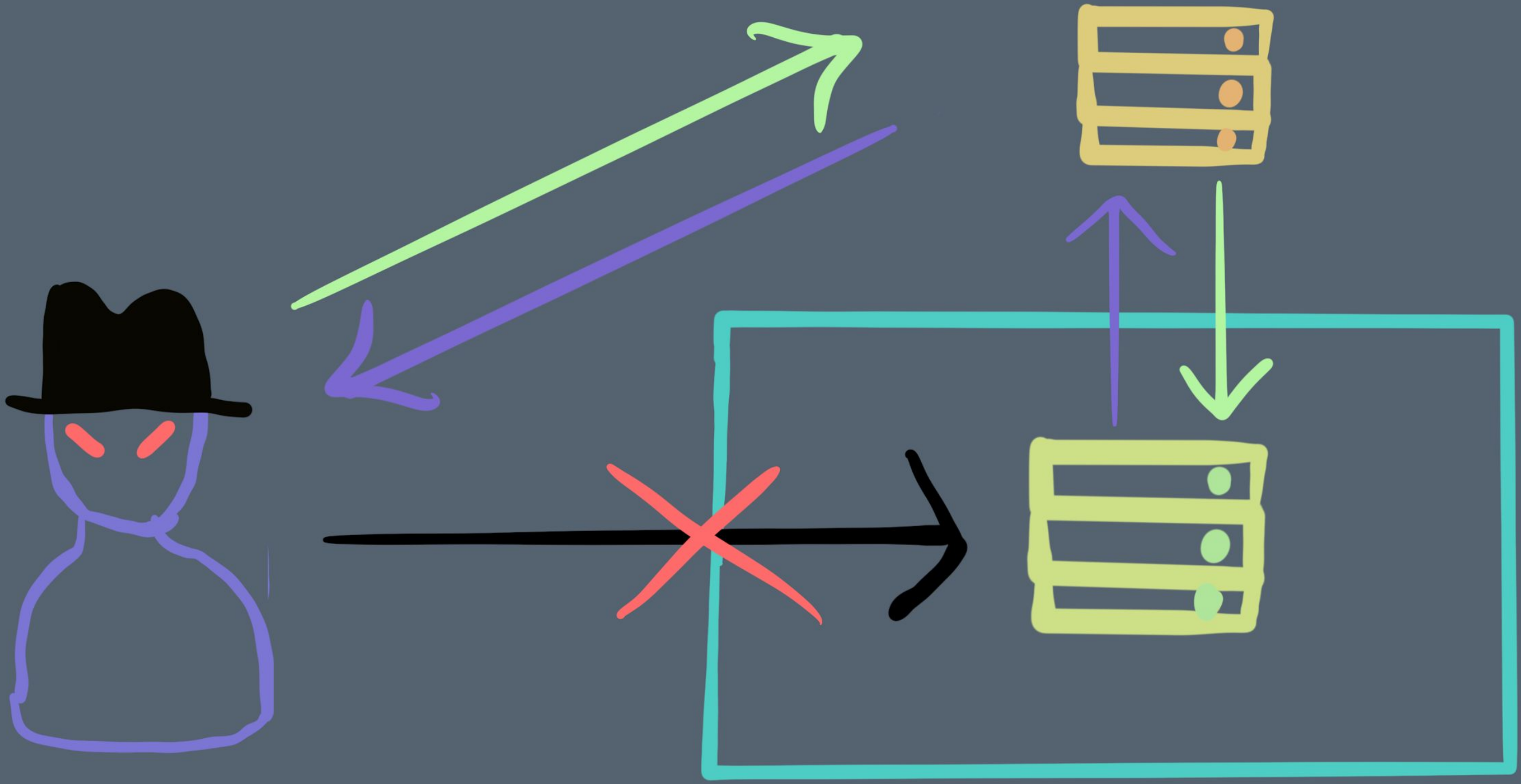
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\fawaz.dinnunhan\Desktop\Dev\XXE-PoC>
```

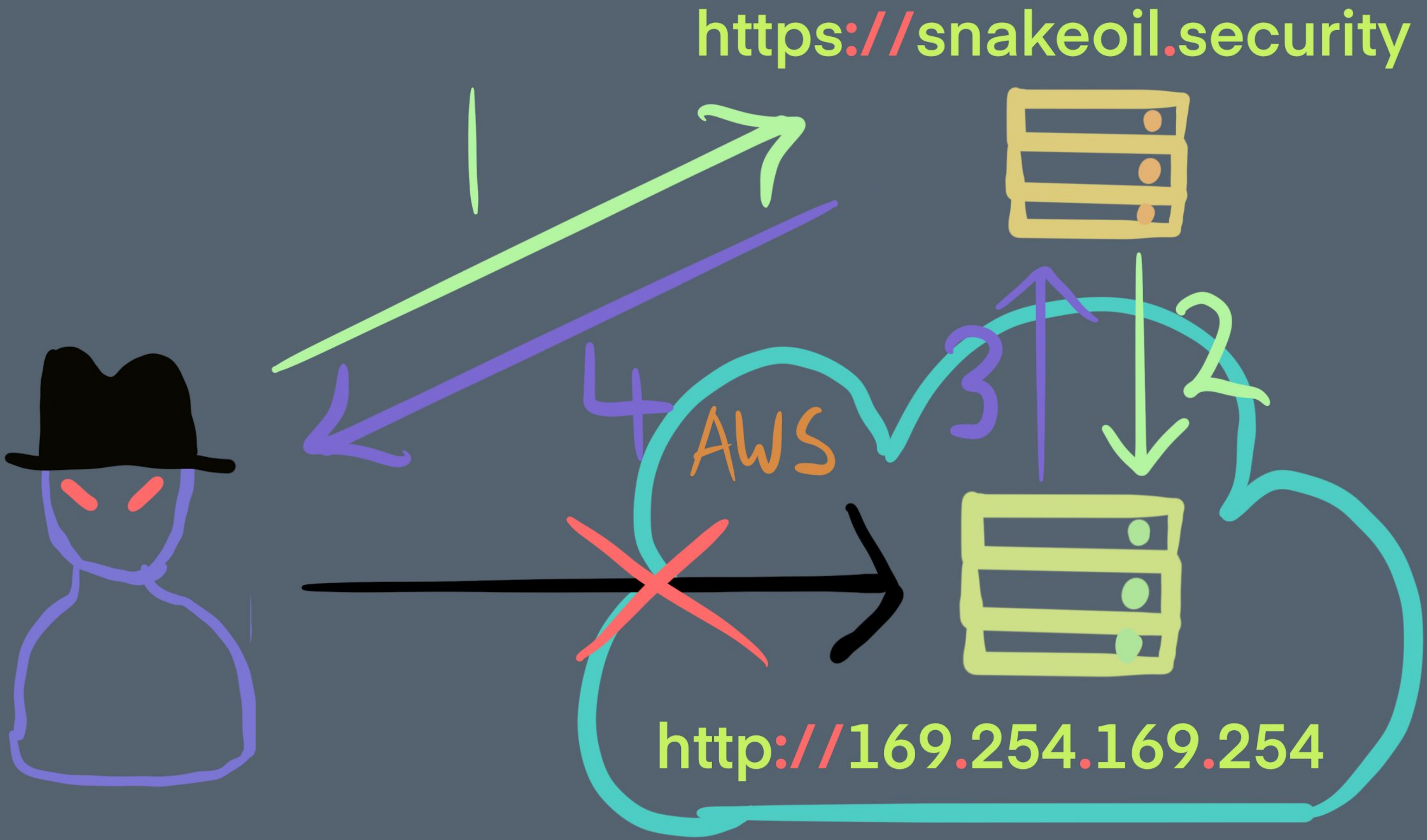
Server
Side
Request
Forgery


https://snakeoil.security
http://169.254.169.254

https://snakeoil.security/scan
?url=http://google.com

https://snakeoil.security/scan
?url=http://169.254.169.254

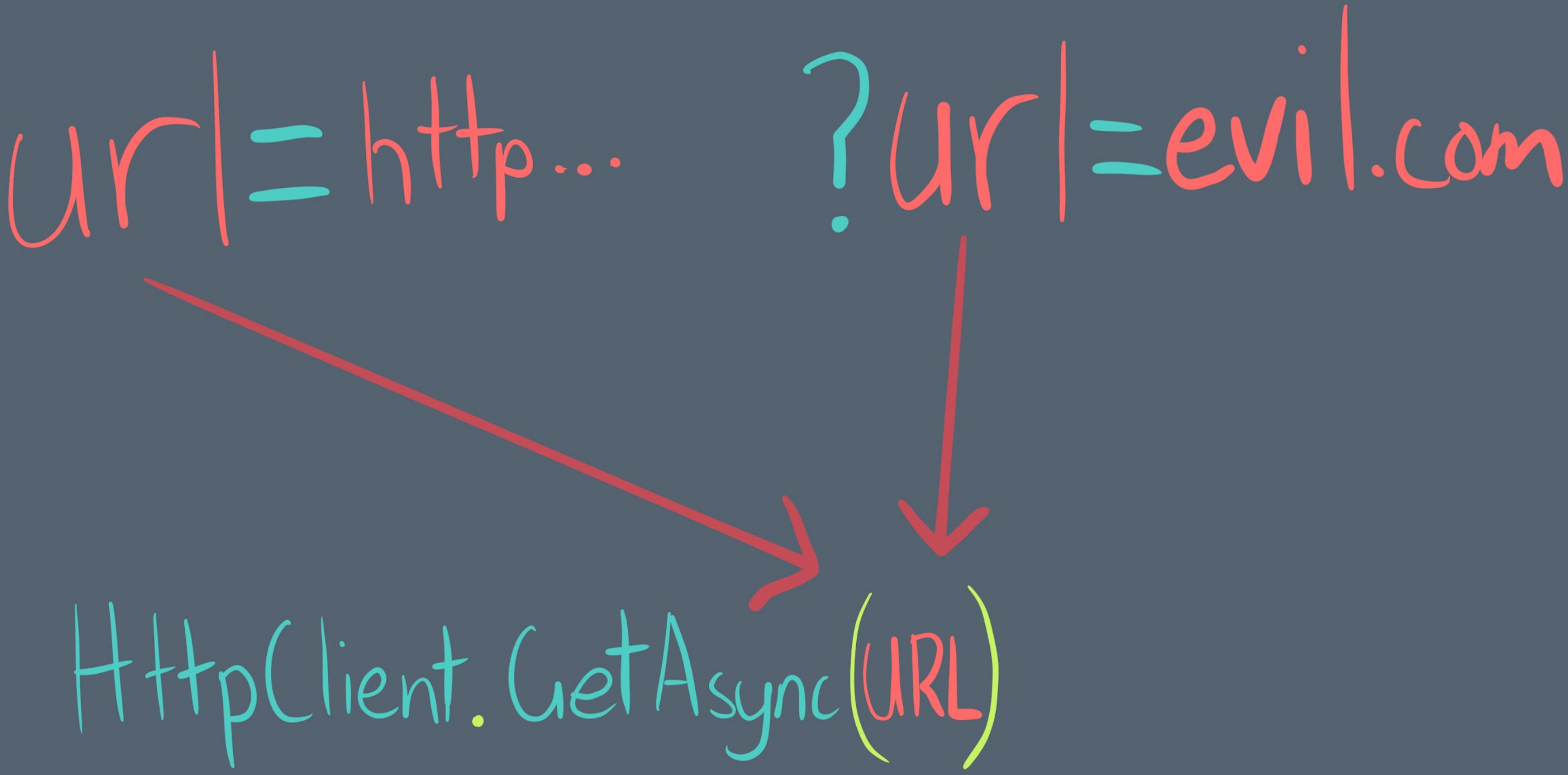


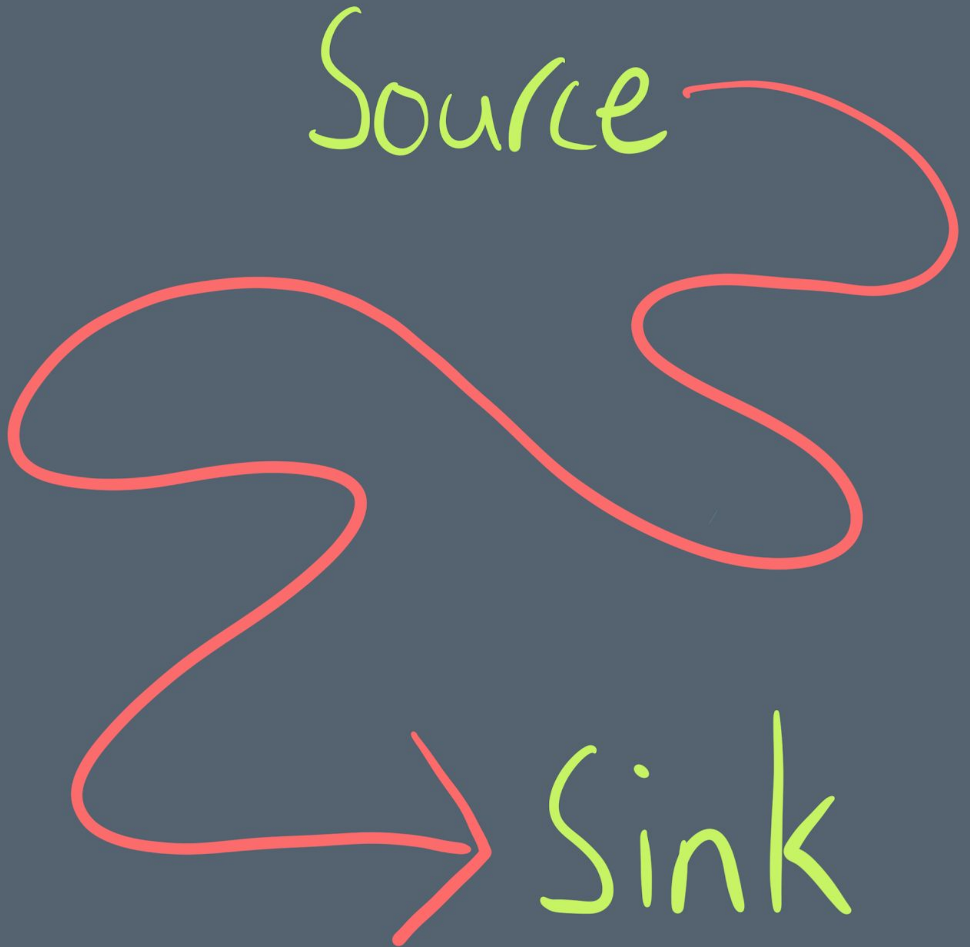


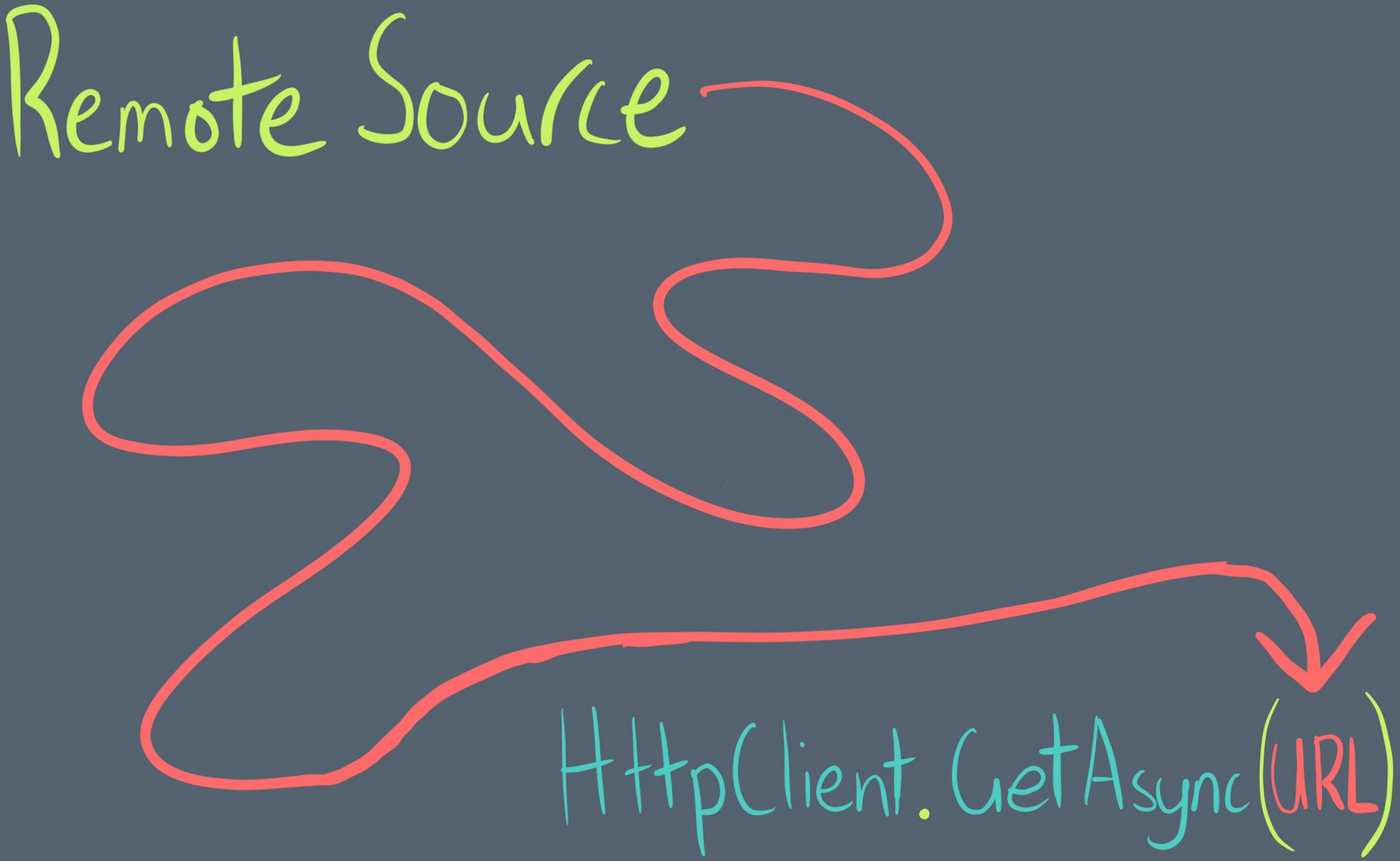
https://snakeoil.security

http://169.254.169.254

```
HttpClient.GetAsync(URL)
```

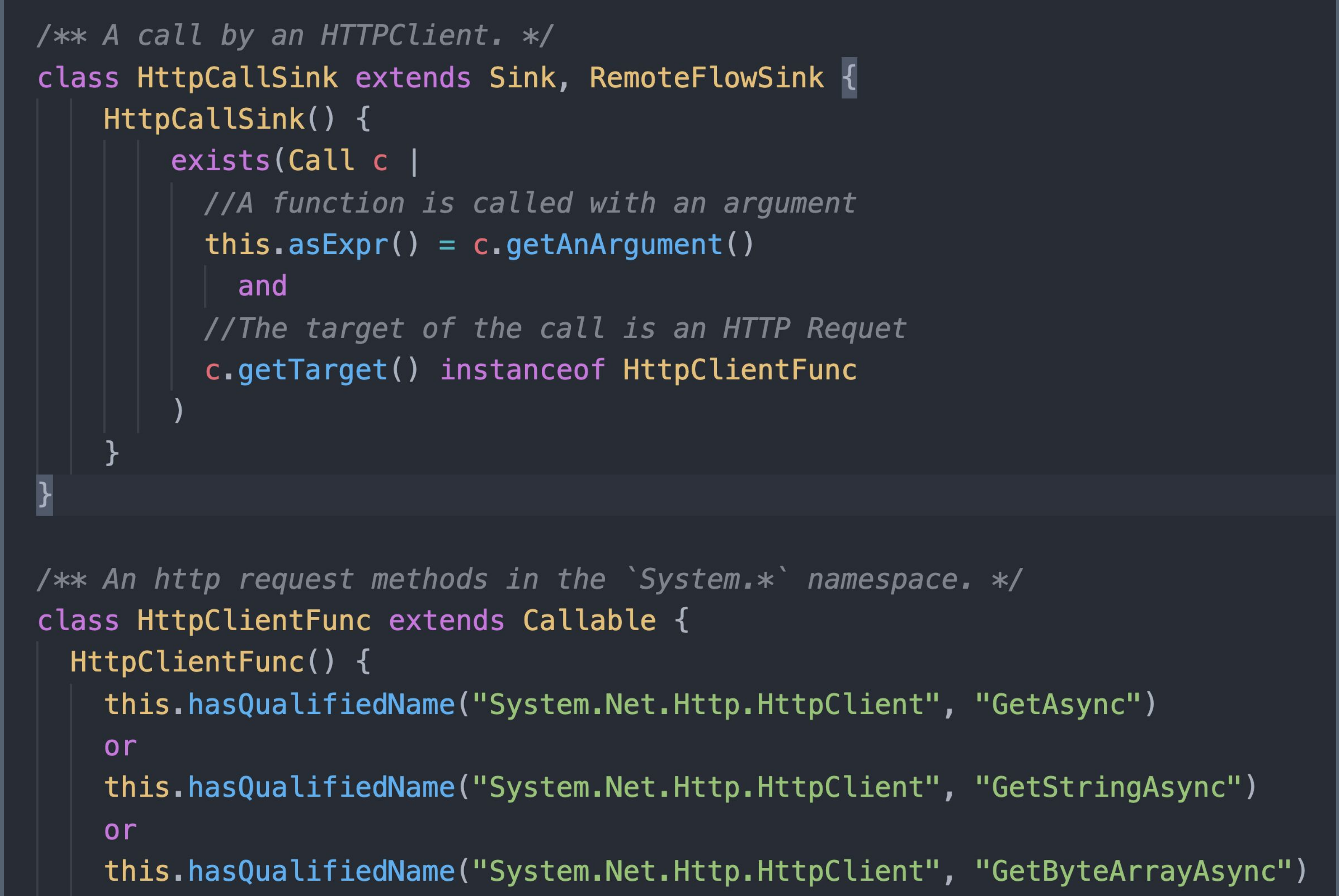

url=http...
?url=evil.com
HttpClient.GetAsync(URL)


Source
Sink

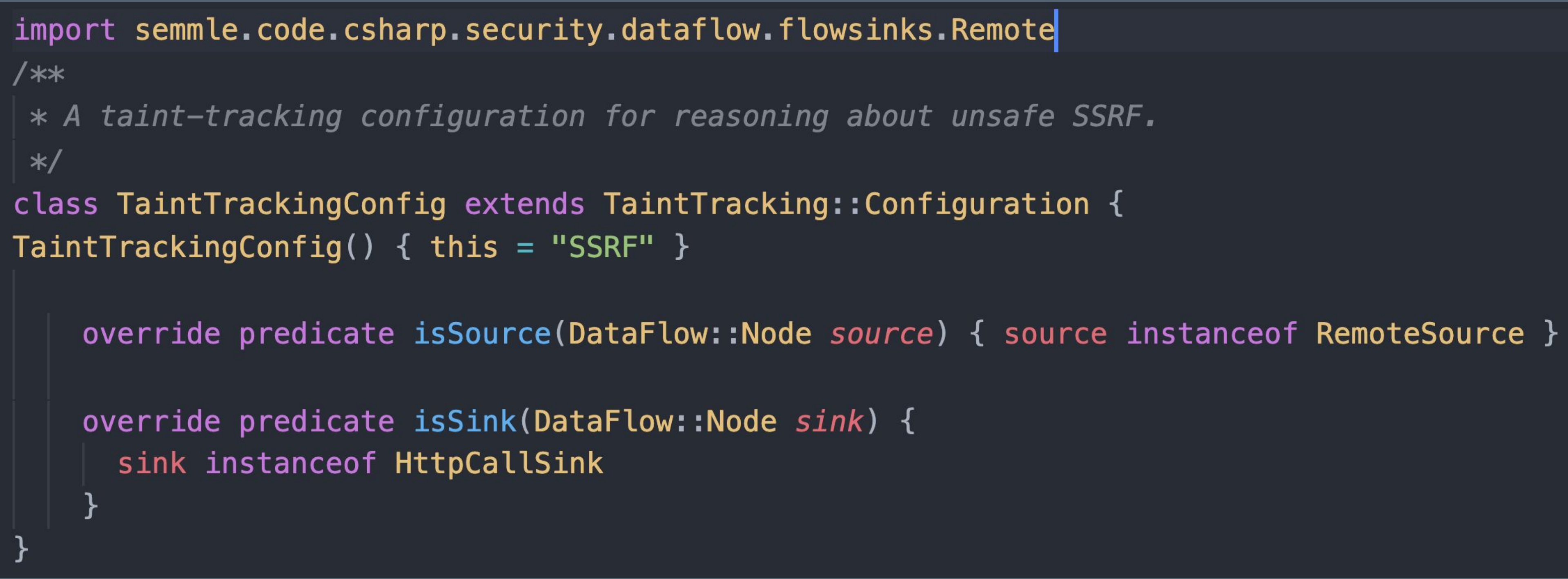


Remote Source

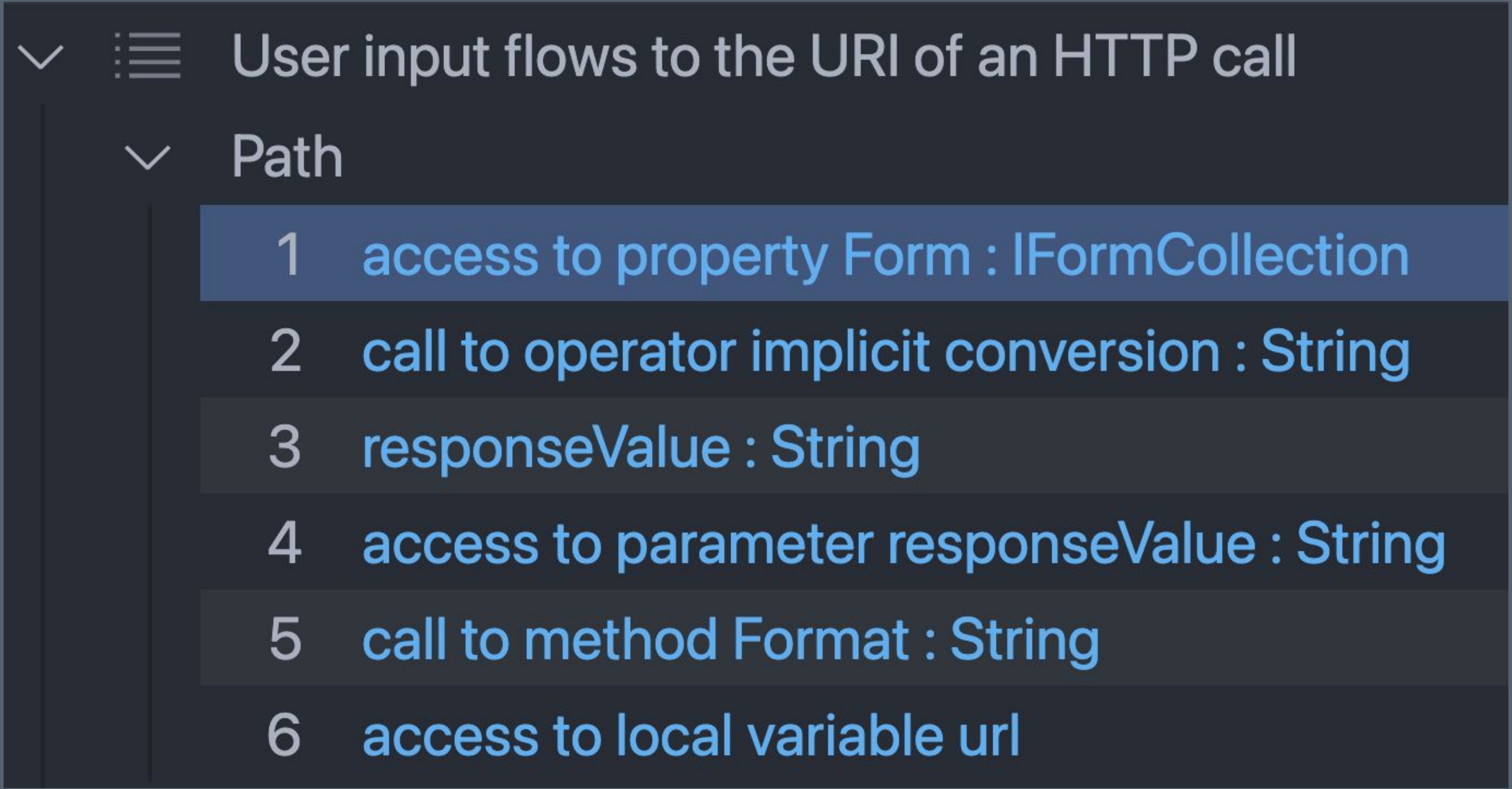
HttpClient.GetAsync(URL)



```
/** A call by an HTTPClient. */
class HttpCallSink extends Sink, RemoteFlowSink {
    HttpCallSink() {
        exists(Call c |
          //A function is called with an argument
          this.asExpr() = c.getAnArgument()
            and
          //The target of the call is an HTTP Requet
          c.getTarget() instanceof HttpClientFunc
        )
    }
}

/** An http request methods in the `System.*` namespace. */
class HttpClientFunc extends Callable {
  HttpClientFunc() {
    this.hasQualifiedName("System.Net.Http.HttpClient", "GetAsync")
    or
    this.hasQualifiedName("System.Net.Http.HttpClient", "GetStringAsync")
    or
    this.hasQualifiedName("System.Net.Http.HttpClient", "GetByteArrayAsync")
```



```
import semmle.code.csharp.security.dataflow.flowsinks.Remote
/**
 * A taint-tracking configuration for reasoning about unsafe SSRF.
 */
class TaintTrackingConfig extends TaintTracking::Configuration {
TaintTrackingConfig() { this = "SSRF" }

    override predicate isSource(DataFlow::Node source) { source instanceof RemoteSource }

    override predicate isSink(DataFlow::Node sink) {
      sink instanceof HttpCallSink
    }
}
```



User input flows to the URI of an HTTP call

Path

1 access to property Form : IFormCollection

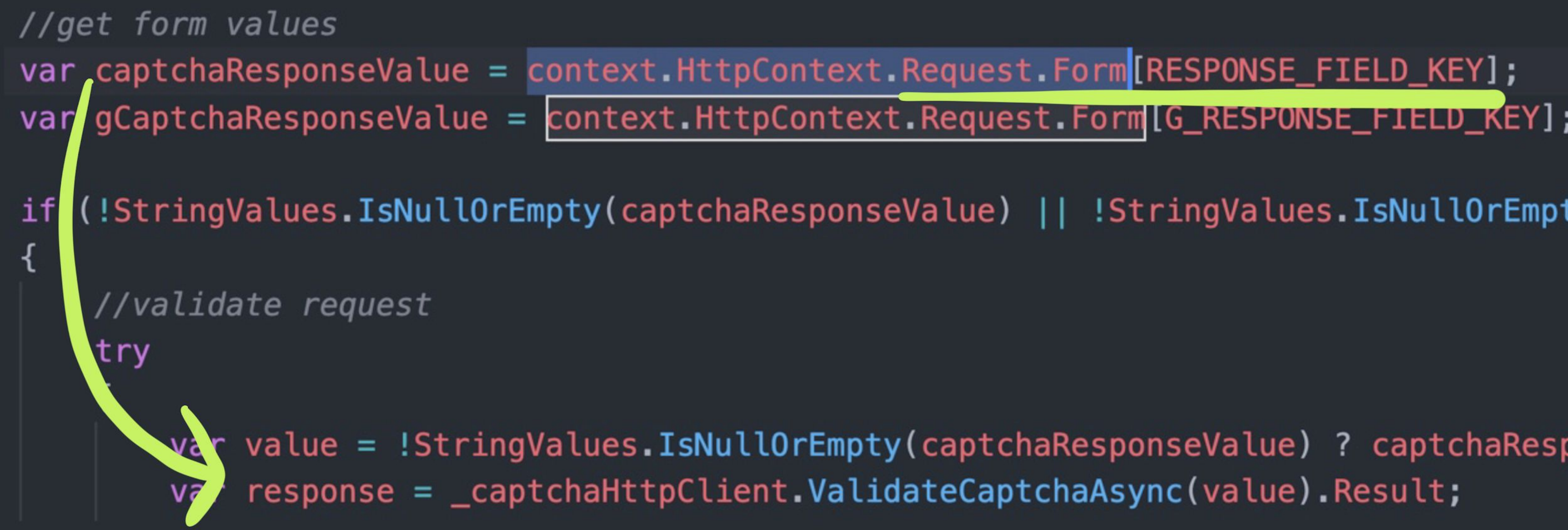
2 call to operator implicit conversion : String

3 responseValue : String

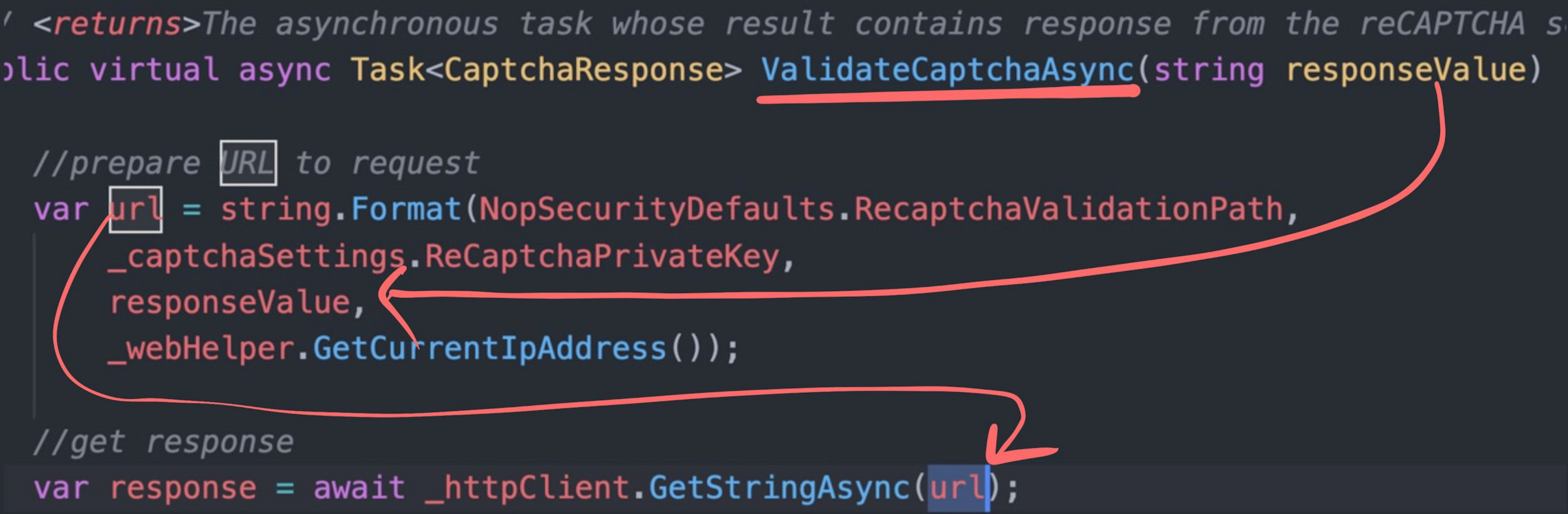
4 access to parameter responseValue : String

5 call to method Format : String

6 access to local variable url



```
//get form values
var captchaResponseValue = context.HttpContext.Request.Form[RESPONSE_FIELD_KEY];
var gCaptchaResponseValue = context.HttpContext.Request.Form[G_RESPONSE_FIELD_KEY];

if (!StringValues.IsNullOrEmpty(captchaResponseValue) || !StringValues.IsNullOrEmpt
{
    //validate request
    try
    {
        var value = !StringValues.IsNullOrEmpty(captchaResponseValue) ? captchaResp
        var response = _captchaHttpClient.ValidateCaptchaAsync(value).Result;
```



```
/ <returns>The asynchronous task whose result contains response from the reCAPTCHA s
lic virtual async Task<CaptchaResponse> ValidateCaptchaAsync(string responseValue)

  //prepare URL to request
  var url = string.Format(NopSecurityDefaults.RecaptchaValidationPath,
      _captchaSettings.ReCaptchaPrivateKey,
      responseValue,
      _webHelper.GetCurrentIpAddress());

  //get response
  var response = await _httpClient.GetStringAsync(url);
```

Recap