Hardening Our Site Against Attacks

Gill Cleeren
ARCHITECT

@gillcleeren www.snowball.be

Overview

Sanitizing input

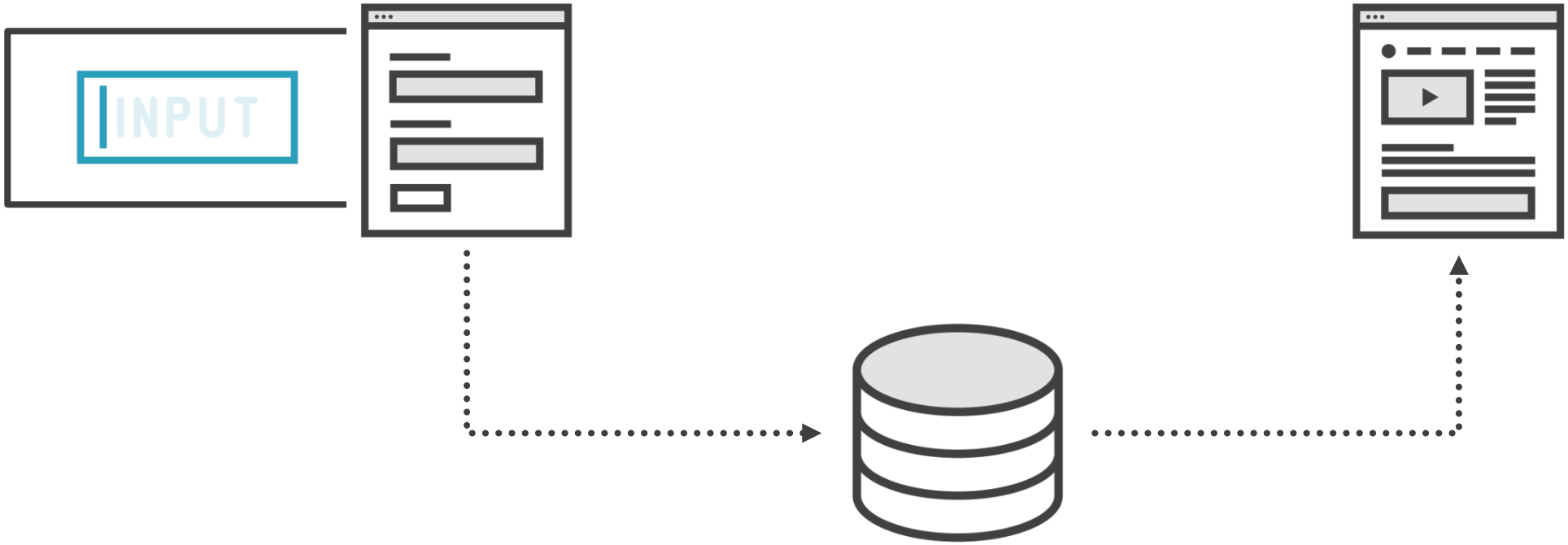
Preventing CSRF

A question from Bethany

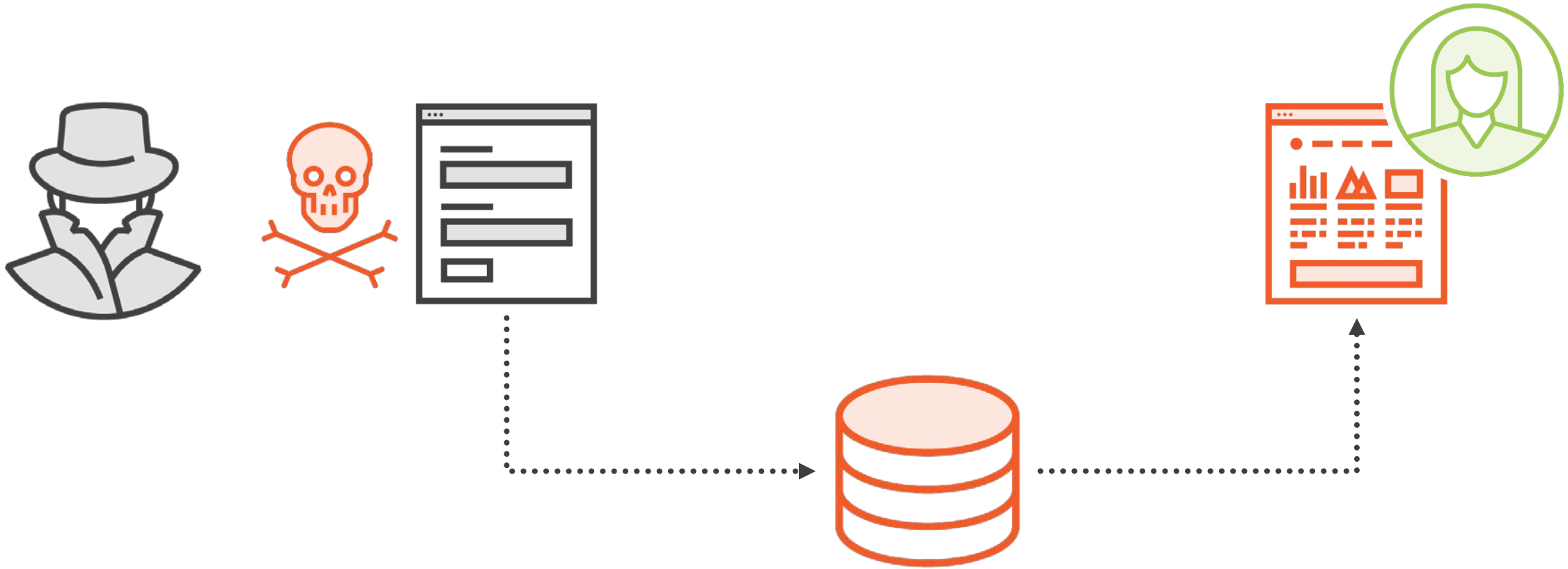
- I want to allow users to leave a review on the site. Is that a good idea?

Sanitizing Input

Entering Data in a Form

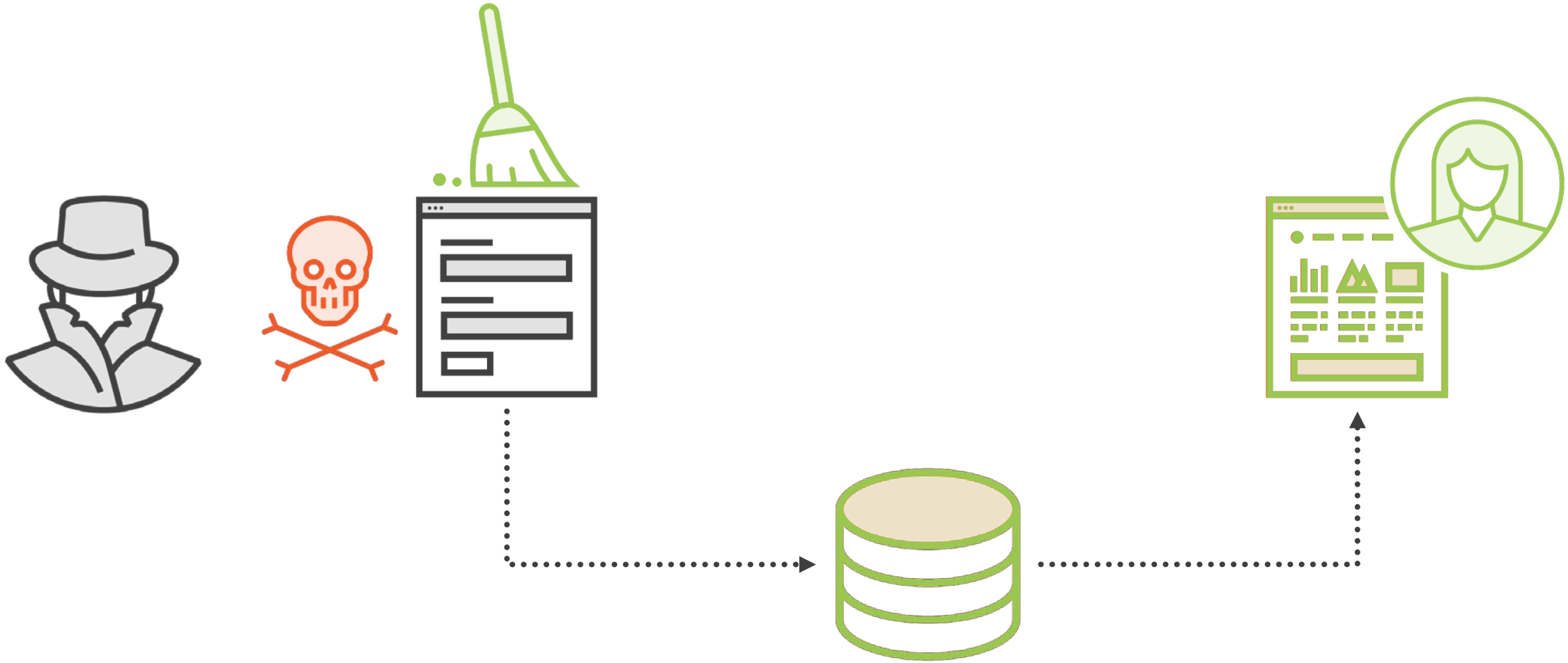


The Danger



XSS enables an attacker to inject script code into a page which will then be viewed by other users

Fighting XSS



Untrusted Data

HTML Input

Headers

Query strings

Attributes

Fighting XSS

Automatic HTML Encoding in Razor

```
@{
    var hackyCode = "<\"Hacked\">";
}
@hackyCode
```

```
&lt;&quot;You are hacked!&quot;&gt;
```

Built-in Encoders

HtmlEncoder

JavaScriptEncoder

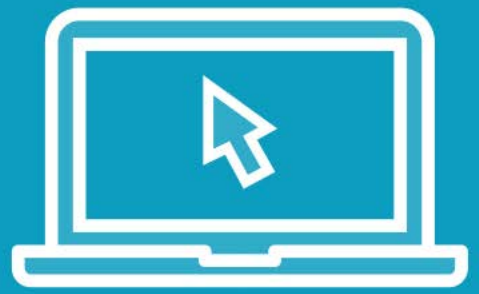
UrlEncoder

```
public class HomeController : Controller
{
    HtmlEncoder _htmlEncoder;

    public HomeController(HtmlEncoder htmlEncoder)
    {
        _htmlEncoder = htmlEncoder;
    }
}
```

Using the Encoders

Demo



Adding the possibility to add reviews

Sanitizing the input to prevent XSS

Preventing CSRF

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.

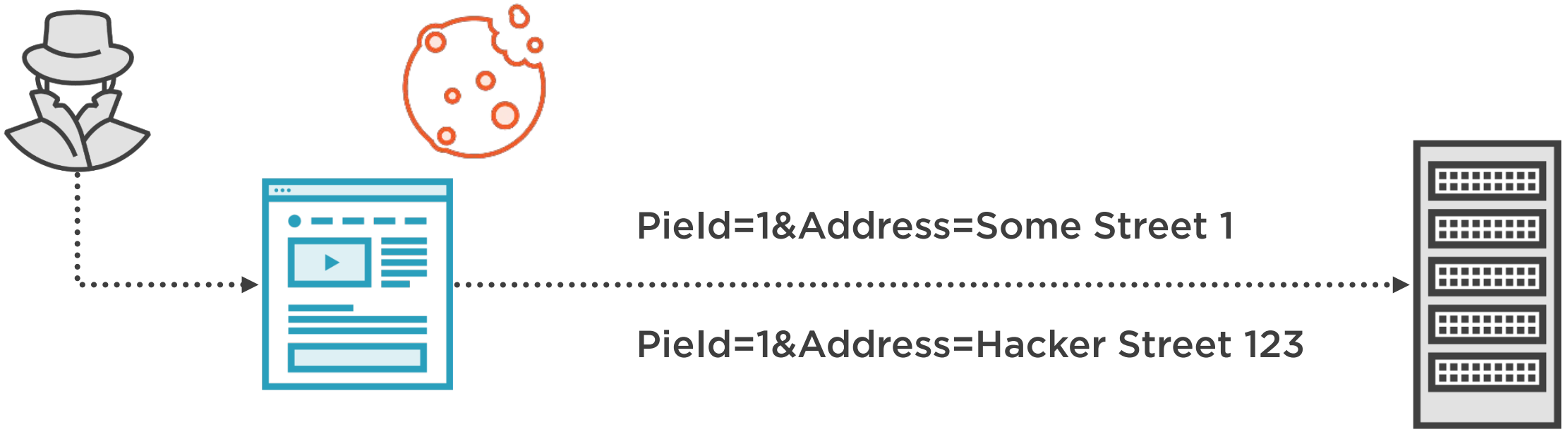
CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request.

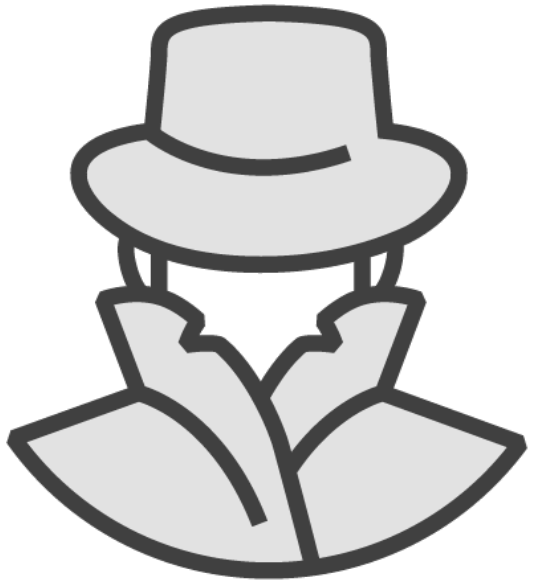
OWASP.org

A question from Bethany

- I want to add a “Send a pie” feature to the site, allowing the user to send a pie to a friend.

A CSRF Attack



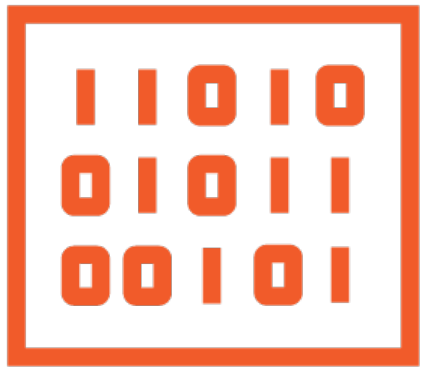


Works because we're logged in

- Confused deputy

Social engineering is required

OWASP Cheat Sheet Solution



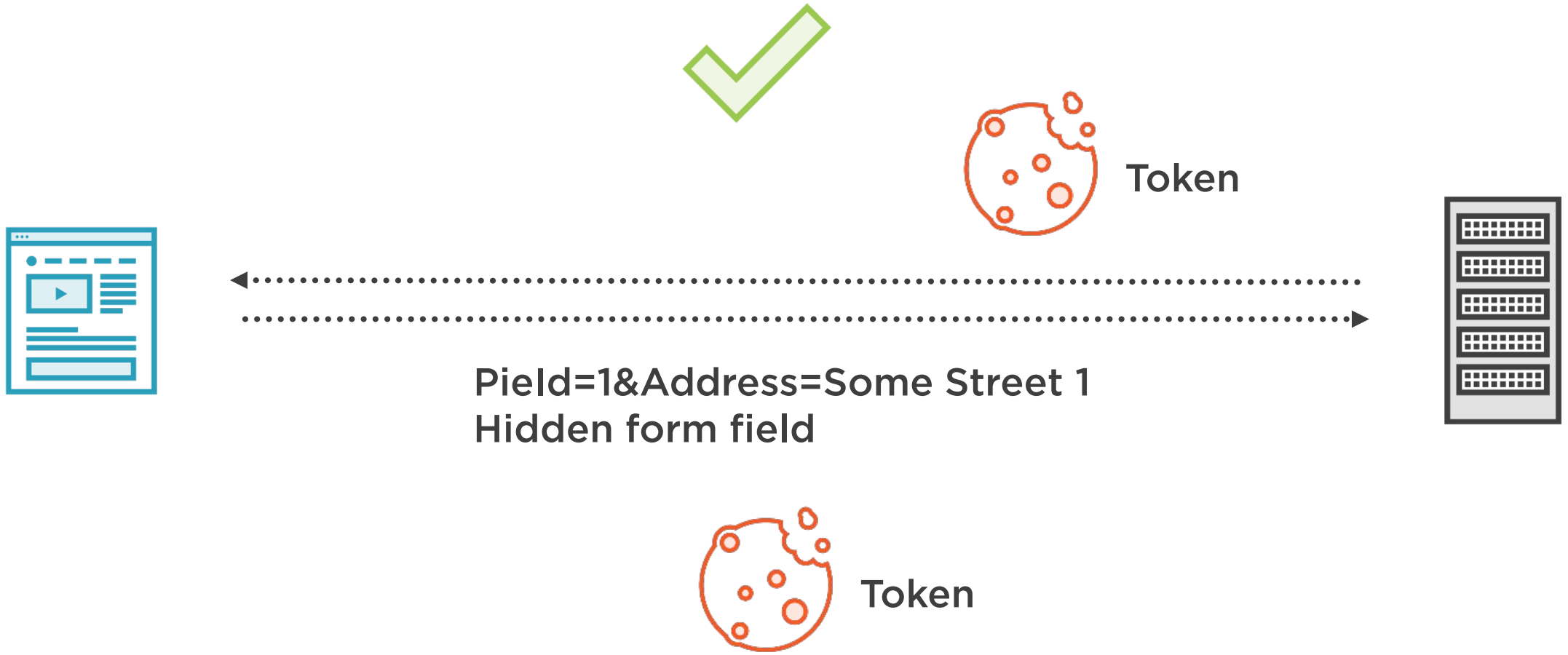
Header validation

Synchronizer token

Double submit token

https://www.owasp.org/index.php/
Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet

Applied in ASP.NET Core: Antiforgery



Applied in ASP.NET Core: Antiforgery

ASP.NET Core Antiforgery

Microsoft.AspNetCore.Antiforgery package

- Already included via Microsoft.AspNetCore.MVC
- Services are registered in DI

Extra configuration in possible in Startup

ASP.NET Core Antiforgery Filters

[ValidateAntiForgeryToken]

[AutoValidateAntiforgeryToken]

[IgnoreAntiforgeryToken]

```
<form asp-controller="PieGift" asp-action="Index"
    method="post" role="form">

    <input type="submit" value="Send gift now" />

</form>
```

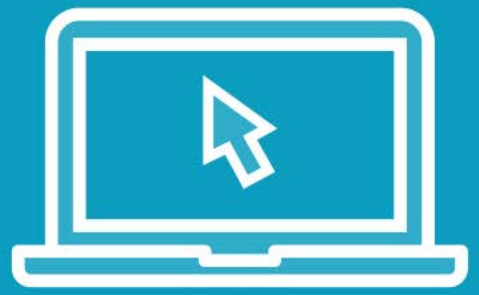
Antiforgery in Action

__RequestVerificationToken (Generated HTML)

```
<input type="submit" class="btn btn-primary" value="Send
gift now" />

<input name="__RequestVerificationToken" type="hidden"
value="CfDJ8O5qWbkFcRhIoKAnIUDVacPkO1BSi-_oW5-ELI4wn1s-
eBgkdEDhNrgeH8JCC-
vE7bZcOYg61NsAH2AkBKprq7z1cPLTMaB7WCT7Lw4-
vX37bnyjWmfdFbH1_uHsE-nMMTQk-
IIIByEh9SI1ifCCyZwM_ukpdc1a147FZPHm_N1yvWaqJ45bbqpHHRqB0Whj
gA" />
```

Demo

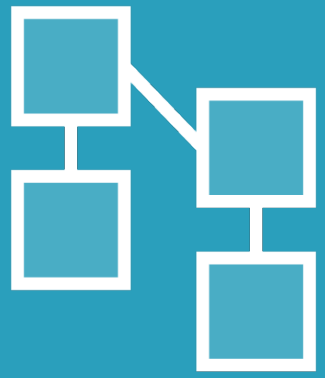


Protecting against CSRF

Summary

XSS countered by sanitizing input

CSRF countered by AntiForgeryToken



Up next:
Validating complex models