Felipe M Saboya
.NET Developer

Impedindo ataques CSRF em ASP.NET Core

http://www.felipemsaboya.net/Blog/AspNetCore-CSRF-AntiForgeryToken

2018/04/01

Felipe M Saboya - .NET Developer

Microsoft MCSA, MCPS, MCP • EXIN ITIL Foundation

www.felipemsaboya.net

Introdução

Quando desenvolvemos Aplicativos Web devemos nos preocupar com a "insegurança da informação", sim "insegurança", pois a abordagem da segurança deve partir do princípio que ninguém está 100% seguro se está conectado à internet, ou seja, nós, profissionais que se propõem a desenvolver e entregar produtos complexos e conectados, devemos ter em mente todos os recursos de segurança que nos são oferecidos pelas tecnologias a que nos propomos a trabalhar, como neste caso, a **ASP.NET** da **Microsoft**.

Dito isto, trago este meu primeiro artigo sobre a importância do uso do recurso chamado **AntiForgeryToken** que nos auxilia a impedir os ataques de **Falsificação de Solicitação entre Sites** ou **CSRF - Cross-Site Request Forgery** (sigla em inglês pronunciada como C-surf) também conhecida como XSRF.

O PROBLEMA

O **CSRF** é um ataque que força um usuário autenticado em um Aplicativo Web a executar ações indesejadas em nome do usuário, ou seja, o ataque força o navegador da vítima a enviar uma requisição.

Por exemplo, você pode estar conectado no site do banco e decide navegar em outro site e, ao clicar em alguma propaganda ela executa uma função de "Post" e transfere seus fundos, sem que você perceba, para outra conta.

A SOLUÇÃO

A solução é bem simples pois o **ASP.NET MVC** possui desde suas versões iniciais o atributo de classe **ValidateAntiForgeryToken**, que deve ser aplicado em duas etapas, e isto fará com que o Aplicativo Web reconheça apenas a ação do usuário atualmente autenticado e rejeitará de qualquer outro que tente executar uma ação por ele, no caso, o **ataque CSFR será impedido** e o Aplicativo Web **retornará o erro de Bad Request**.

ETAPA I

Na decoração da classe "Post" adicionando também o atributo ValidateAntiForgeryToken. Conforme o exemplo ao lado.

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references | 0 changes | 0 authors, 0 changes
public async Task<IActionResult> Edit(TestViewModel viewModel)
{
    // Your code here!

    return await Task.Run(() => View());
}
```

ETAPA 2

Adicionando o Helper no formulário da View. Isto irá gerar o "Token" a ser reconhecido pelo Aplicativo Web.

```
<form asp-controller="Test" asp-action="Edit" method="post">
    @Html.AntiForgeryToken()
</form>
```

* Será gerado em tempo de execução algo similar ao input:

```
<input name="__RequestVerificationToken" type="hidden"
value="CfDJ8PIdqkEW0YRGiAj8BeRzSBbihqUysw0OL9LaEmtunQYO_YPhN1jS7YTAhPIsUY0HgD3yc1_ItbFMdcusdHHjQzuQGCrnew-
orLK5GjvV_Ar3N-gomVX7UoHFGOjSKqW84rh4xFcDCqVjbOzzd12qOnENuOLbUmpx_jb7mWyJa_5W5dEGL-b6Siqw" />
```

Solução adicional para ASP.NET Core

Adicionalmente, a nova tecnologia ASP.NET Core tem a facilidade de aplicarmos na classe **Startup** a validação para toda a solução e que se aplicará apenas às classes de mudanças de estado (POST, PUT e DELETE).

A **OWASP** recomenda que os verbos citados acima (POST, PUT e DELETE) estejam protegidos deste tipo de ataque (ver referências no final do artigo), então, em resumo, a implementação nos auxilia a atender a esta recomendação e, **caso seja necessário ignorar a validação** do CSRF podemos apenas decorar a classe com o atributo **IgnoreAntiforgeryToken**.

CLASSE STARTUP

Complemente o "AddMvc" com a linha apresentada ao lado na sua classe Startup.

```
// This method gets called by the runtime. Use this method to add services to the container.
0 references | Felipe Monteiro, 22 hours ago | 1 author, 1 change
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc(options => { options.Filters.Add(new AutoValidateAntiforgeryTokenAttribute()); });
}
```

Conclusão

- Sugiro que sempre seja utilizado o recurso apresentado neste artigo;
- Sugiro a leitura do documento com um TOP 10 que encontrei em https://www.owasp.org/images/4/42/OWASP_TOP_10_2007_PT-BR.pdf que cita, além desta, outra ameaças a serem analisadas e tratadas;
- Pesquisar de modo geral o site da OWASP, pois existem informações que simplesmente devemos saber.

Felipe M Saboya
.NET Developer

Obrigado!

Felipe M Saboya - .NET Developer

Microsoft MCSA, MCPS, MCP • EXIN ITIL Foundation

www.felipemsaboya.net