03

Cross Site Scripting (XSS)

Transcrição

[00:00] Nós compramos os ingressos aqui da Alura Shows e agora nós queremos deixar um depoimento pra Alura Shows para que todos os usuários saibam que a nossa experiência foi muito boa, que nós conseguimos bons ingressos e que ficamos bastante contentes.

[00:13] Então nessa aba depoimentos nós vamos querer deixar esse depoimento para a Alura Shows e para os demais usuários que forem visualizar, que forem acessar a Alura Shows saberem como é que está sendo a experiência dos usuários com as compras.

[00:27] Vamos colocar o título da nossa mensagem de depoimento como sendo "ingressos ótimos", e falamos: "gostei bastante dos preços dos ingressos, eram acessíveis e com ótimos lugares". E clicamos no botão para enviar esse nosso depoimento. Clicamos aqui "enviar". Agora, vamos tentar entender o que acontece por debaixo dos panos para que nós tenhamos essa nossa postagem aparecendo aqui nessa nossa JSP.

[01:00] Eu tenho aqui essa apresentação que eu acho que vai ficar um pouco mais fácil pra entendermos. Então, eu tenho aqui esse meu computador que é esse computador da gravação, acabamos de rodar esse teste do depoimento. E o que nós estamos fazendo?

[01:13] Quando clicamos no botão de "enviar". Nós estamos fazendo o que? Uma requisição para o nosso Tomcat, para o nosso servidor aqui. Vai ter uma parte do Tomcat, algum método que vai estar cadastrando, persistindo esses dados do depoimento no banco de dados. Nós também temos que ter um método para pegar essas mensagens, esses depoimentos do banco de dados e voltar, aqui no caso, para o usuário.

[01:43] Então, é esse fluxo que tá acontecendo. Se formos aqui na nossa aplicação, vamos clicar com o botão direito do mouse, no botão enviar e inspecionar, e quando nós clicamos no botão "enviar", a nossa mensagem de depoimento está sendo encaminhada aqui pra esse endereço "envia mensagem".

[02:05] Nós temos que ter lá um método no nosso pacote Controller que está respondendo a essa requisição para esse endereço "envia a mensagem". Vamos conferir? Vamos aqui na nossa aplicação no eclipse. E aqui no nosso pacote de Controller, nós temos essa classe depoimento Controller. Vamos clicar nela?

[02:21] Depoimento Controller. E aqui nós temos o nosso endereço "enviar mensagem" e o que tem dentro desse método aqui que está tratando essa requisição para esse endereço "enviar mensagem", nós estamos salvando o depoimento no banco de dados e temos aqui um método chama posts do banco.

[02:42] Nós estamos chamando essas mensagens que foram cadastradas aqui no banco de dados. Então, com isso o que acontece? Se nós formos lá, no caso do nosso usuário Alex, que que nós estamos utilizando-o para fazer essas simulações de ataques na nossa aplicação, o que ele pode pensar? Bom, ele vai voltar para o computador dele, do Kali Linux e ele vai querer visualizar essa parte de depoimentos também para ver o que os usuários da Alura Show estão dizendo.

[03:10] Vou só logar aqui no Kali Linux e aqui nós colocamos o endereço de acesso da nossa aplicação, deixa só eu colocar aqui, Alura Shows, e nós temos aqui a parte de depoimentos, como nós vimos, a mensagem que eu escrevi no meu computador do Windows, que é um outro computador, ela apareceu aqui no Kali Linux do Alex.

[03:35] O que o Alex pensa? Ele pensa “será que os desenvolvedores da Alura Shows chegaram a fazer uma espécie de filtro, alguma checagem dessas informações que podem ser passadas aqui no título e na mensagem?”. Porque se eles não fizeram, o que o Alex vai tentar realizar? Bom, vamos voltar aqui para apresentação para entender o que o Alex vai poder conseguir se os desenvolvedores da Alura Shows não tenham feito uma análise do que o usuário pode passar nesse campo de título e mensagem.

[04:06] Bom, nós sabemos que o nosso browser interpreta código JavaScript. Então, o nosso hacker, o Alex, ao invés de colocar uma mensagem dizendo como é que foi a experiência dele com sites da Alura Shows, ele vai tentar colocar nesse campo um código JavaScript e, como nós vimos, o que vai acontecer?

[04:27] Quando ele clica no botão de enviar, essa informação seria passada para o nosso servidores do Tomcat, para a nossa aplicação e nós temos o método enviar mensagem que está certo, que tá persistindo esse depoimento no banco de dados e nós temos aquele método chama posts do banco, que está pegando todas essas mensagens do banco de dados. Agora, o que acontece se uma dessas mensagens aqui for um código Javascript?

[04:55] Se é um código JavaScript, essa mensagem seria devolvida aqui pro nosso usuário, porque as mensagens estão sendo carregadas na página. E uma dessas mensagens é um código JavaScript. E o que o Alex vai conseguir fazer com isso? Se qualquer usuário que for acessar agora essa parte do depoimento, outro usuário, por exemplo, do meu computador do Windows, caso acesse a página de depoimento e caso uma dessas mensagens que foram cadastradas no banco de dados foi de um código JavaScript, esse código JavaScript vai parar no browser desse outro usuário também.

[05:29] E com isso nós temos um ataque conhecido aqui como cross-site-scripting, que é justamente onde nós inserimos códigos JavaScript podendo comprometer a segurança de outros usuários. Então, vamos tentar fazer essa simulação aqui do Alex?

[05:44] O Alex é um ativista do grupo do Anonymous e quer testar, verificar se a aplicação da Alura Shows apresenta essa vulnerabilidade. Então ele vai querer colocar uma imagem do grupo do Anonymous, para que todo mundo saiba que o grupo do Anonymous que descobriu essa vulnerabilidade na Alura Shows.

[06:03] Vamos voltar no computador do Alex, aqui no Kali Linux, e o Alex vai tentar fazer esse teste com a aplicação da Alura Shows para ver se por um acaso os desenvolvedores esqueceram de validar o que os usuários pode estar passando nesses campos aqui. Então, vamos colocar o título "gostei bastante", até aqui tudo bem, e agora na mensagem, o Alex vai colocar esse código JavaScript.

[06:27] Como é que você insere um código JavaScript? Nós temos que abrir a tag, e colocar "script" e fechamos a tag. Aqui dentro dessa tag script, nós vamos configurar essa imagem do grupo Anonymous. Então, o primeiro passo, o que eu quero fazer? Deixar todo esse documento da parte de depoimentos com um fundo branco para que eu coloque a imagem em cima.

[06:52] Eu vou colocar aqui “document.body.innerHTML=””.Uma vez que nós já colocamos esse código, a nossa tela do depoimentos ficaria em branco, e o que nós vamos fazer? Nós queremos colocar essa imagem do Anonymous aqui no lugar da parte dos depoimentos.

[07:14] Vamos criar esse objeto imagem. Colocamos “var imagem=new.image():” E agora nós queremos vincular o endereço dessa imagem com uma imagem do Anonymous. Então, eu estava pesquisando umas imagens do Anonymous e encontrei essa imagem aqui para usarmos como base. Eu vou só copiar aqui.

[07:37] Vou copiar o local da imagem e nós vamos colocar imagem.src =”, a origem da imagem está onde? Está nesse endereço aqui que do Pixabay, é a imagem do Pixabay saber que eu estou usando. Então está aqui. E agora o que nós queremos fazer? Nós queremos juntar esse objeto imagem no corpo do nosso depoimento JSP, no corpo dessa nossa página.

[08:03] Então, para isso nós temos que vir aqui e colocar “document.body.”, e nós queremos fazer o quê? Nós queremos fazer essa junção desse objeto imagem no body. Nós chamamos aqui “appendChild(imagem).”.

[08:21] Com isso se nós já terminamos a nossa configuração nós pode fechar a tag scrip. Isso quer dizer aqui, que caso os desenvolvedores da Alura Shows não tenham feito nenhuma espécie de filtro do que o usuário pode estar passando nesses campos, o que vai acontecer? Esse código JavaScript vai para o nosso banco de dados e temos o nosso método que chama todas essas mensagens do banco de dados.

[08:47] Agora, se uma dessas mensagens for um código JavaScript, esse código JavaScript é interpretado pelo browser e todos os usuários vão estar visualizando essa imagem do grupo do Anonymous. Então, vou verificar se o Alex vai ter sucesso nessa tentativa de ataque dele. E se por um acaso os desenvolvedores da Alura Show esqueceram de fazer essa verificação, vamos lá. Vou clicar em e enviar.

[09:12] Olha aqui, pra mim já apareceu. Para mim aqui no computador do Kali Linux já apareceu a imagem do Anonymous. Então, isso nos leva a crer que de fato os desenvolvedores da Alura Shows se esqueceram de validar o que o usuário pode estar passando como parâmetro nesses campos. Agora vamos fazer o seguinte só para confirmar, vamos sair aqui do Kali Linux e vamos voltar para o nosso computador do Windows, que é um outro usuário, outro computador e vamos clicar na aba depoimentos e vamos ver se vamos a imagem do grupo do Anonymous.

[09:46] Vamos lá. Depoimentos e está aqui a imagem do grupo do Anonymous. O Alex conseguiu, de fato, ter sucesso nesse ataque de cross-site-scripting que ele acabou de realizar. Agora nós temos que justamente encontrar uma forma de evitar que esse ataque aqui aconteça. Vamos ver na próxima etapa.