Xss - Refletido.

Introdução

Opa, queria dizer antes tudo, que esse vai ser o meu primeiro artigo escrito! E portanto, não imagino que tenha uma qualidade tão boa, mas ainda sim espero que dê para entender mais ou menos o conceito da "vul". Então prazer! Me chamo Frolich e falarei um pouco sobre XSS.

Sobre o que é XSS?

Xss em sua tradução bruta significa "script em sites", e isso se refere em executar scripts dentro de um site. Um site normalmente é composto por 2 tipos de linguagem, sendo uma de Hipertexto e outra de Back-End! Sendo as mais comuns utilizadas:

* HTML
* CSS
* JavaScript
* PHP

E XSS pode entrar no contexto de todas! Já que XSS se trata de uma execução de código não proposta pelo código fonte do site.

O que é o XSS refletido?

Existem 3 tipos de XSS, que são os: "Refletidos", "Permanentes", " Dom". Nesse artigo falaremos sobre o *permanente* mas você pode ver meus outros artigos falando sobre os outros dois tipos no meu **Github**!

Nós encontramos XSS em pontos de entrada dentro de um site! Como uma barra de procura, um parametro ou em postagens. Então em um exemplo onde usamos o JS para realizar tal vulnerabilidade e passarmos o link da página em que testamos para o nosso alvo ele receberá o mesmo resultado que a gente! Isso é um xss refletido! Uma execução de código de uma url contaminada e refletida no navegador do nosso alvo!

Exemplo:

Então abrindo o metasploitable e indo para a aba de "xss", podemos ver esse campo de inserção perguntando o nosso nome. Se a gente colocar o nome "Carlos" e dar o **"Submit"** ele **exibirá** "Carlos", **dentro da página**. Então sabendo disso a gente coloca um script simples em JS e vemos o resultado.



Executado esse processo. Temos o seguinte resultado:



Aqui podemos ver que na url existe o parametro "*name*" que executa o nome/código que passamos dentro dela! Sendo assim fizemos um XSS funcionar, e se copiarmos a url e esconder dentro de um outro link para o nosso alvo, vai ter o mesmo resultado que o nosso! E em uma situação real poderia ser um script malicioso e assim roubando cookies, sessões, ou até colocando um keylogger dentro do navegador do alvo. Por isso é uma vulnerabilidade tão séria.

Então se você pretende ir testando XSS em um site manualmente, vizualise sempre o código fonte da página que carregou o payload e verifique a filtragem que ocorreu.



Finalização

Se algum dia alguém ler isso, muito obrigado por ler! Espero que você tenha entendido e espero que tenha ficado claro. Futuramente postarei mais artigos de diferentes falhas, explicações do que são diferentes coisas e assim por diante, sempre focando na parte hacking!

Obs: É sempre importante dizer que o conteúdo colocado nesse artigo é feito meramente para fins educativos e didáticos e não me responsabilizo por qualquer ato ilegal feito a partir do conhecimento desse artigo por mais simples que ele seja!

Frolich - (24/08/02)