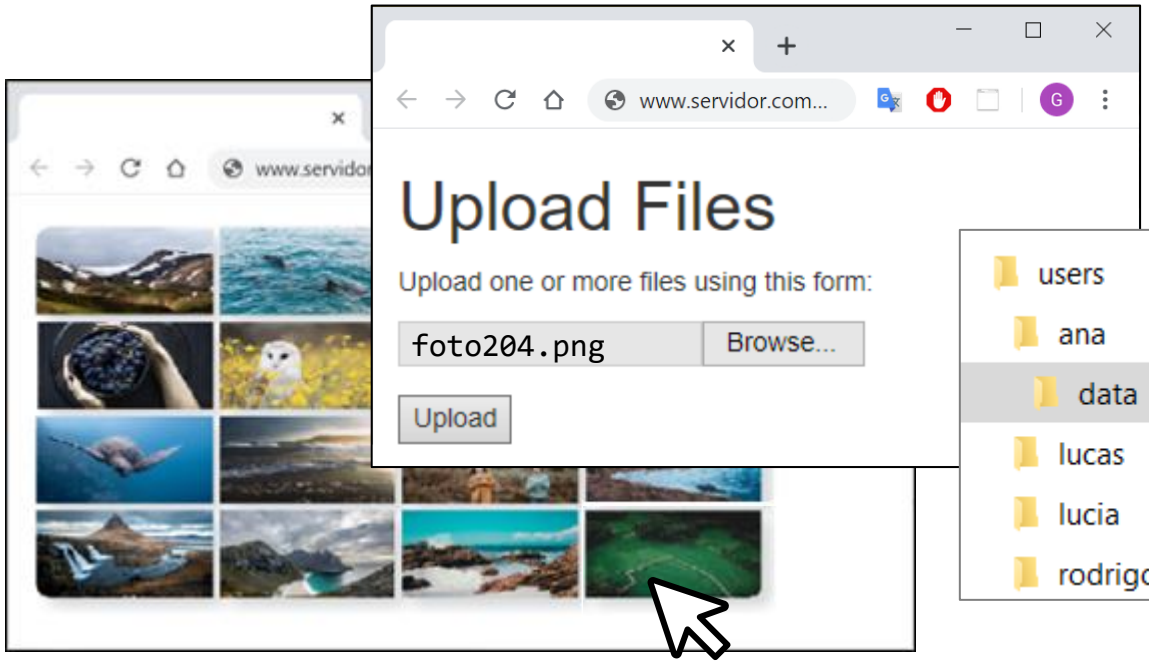
Upload de arquivos maliciosos

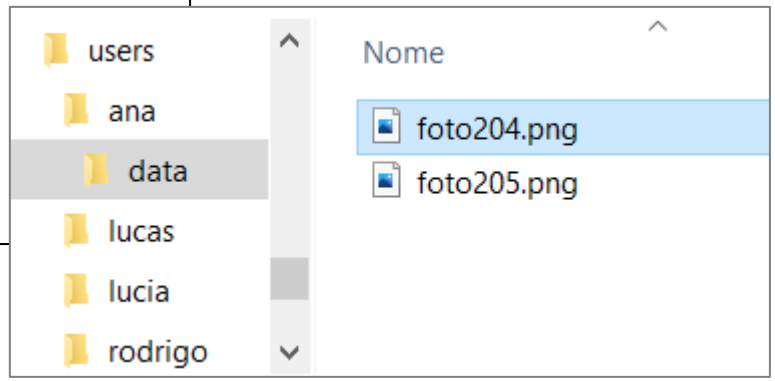
CWE-434: Unrestricted Upload of File with Dangerous Type

Motivação

Considerar um website que permite que o usuário envie arquivos (fotos, vídeos, etc)



No servidor:




```
http://www.servidor.com/dados/foto204.png
```

Upload de arquivos

- O *upload* de arquivos podem tornar a aplicação vulnerável, caso não manusear o arquivo de forma apropriada.

- Dependendo de como o arquivo é processado ou onde o arquivo é armazenado, o web site pode estar vulnerável a esta ameaça

Exemplo 1

O código abaixo permite que o usuário faça o *upload* de uma imagem no servidor web. O código HTML utiliza um campo `input` com o tipo `file`:

```
<form action="upload_picture.php" method="post" enctype="multipart/form-data">

  Escolha um arquivo para enviar:
  <input type="file" name="filename"/>
  <br/>
  <input type="submit" name="submit" value="Submit"/>

</form>
```

Exemplo 1

Uma vez submetido, o formulário web envia o arquivo para `upload_picture.php` no servidor:

```php
// Define o local de destino onde a imagem
// enviada será gravada
$target = "dados/" . basename($_FILES['uploadedfile']['name']);

// Move o arquivo enviado para o novo local.
if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target))
{
   echo "A imagem foi enviada com sucesso.";
}
else
{
   echo "Houve uma falha ao enviar a imagem. Tente novamente.";
}
```

Exemplo 1

- O problema com o código anterior é que não há qualquer validação quanto ao tipo de arquivo que está sendo enviado.

- Considerando que os arquivos que estão em "`dados/`" estão disponíveis para acesso, um agente poderia enviar um arquivo com este nome:

```
malicioso.php
```

- Como o arquivo possui a extensão ".php", o servidor de páginas executa o programa, ao invés de enviá-lo ao navegador do usuário.

Exemplo 1

- Uma vez que o arquivo foi “instalado”, o agente invocar o programa através da URL abaixo:

```
http://www.servidor.com/dados/malicioso.php
```

- O programa abaixo expande o problemas que podem ser causados:

```php
<?php
system($_GET['cmd']);

?>
```

```
http://www.servidor.com/dados/malicioso.php?cmd=ls%20-l
```

Riscos

- Há a possibilidade de ser feito *upload* de *malwares*
- O servidor pode ser comprometido com o *upload* e a execução de um *web-shell* que pode executar comandos, navegar em arquivos, atacar outros servidores e explorar outras vulnerabilidades
- O atacante pode substituir arquivos do servidor
- O servidor pode ser usado para hospedar conteúdo ilegal
- O *atacante* pode colocar uma página falsa no servidor ou manipular uma página do servidor
- Pode ser explorada para causar falta de recursos (espaço de armazenamento)

Formas de mitigar

- Limitar os nomes de arquivos que podem ser submetidos. Somente aceitar arquivos com determinadas extensões.
  - Alguns servidores de página podem processar extensões internas. Exemplo, algumas versões do Apache consideram “filename.php.gif” como sendo um arquivo PHP
  - Se for possível, somente aceitar uma única ocorrência do caractere ponto como nome de arquivo.

- Ao armazenar o arquivo, salvá-lo com um nome único, gerado pelo programa, ao invés de utilizar o nome fornecido pelo usuário. Assim, nenhum nome externo é utilizado.