

Explorando o Método GET e suas Vulnerabilidades.

NO SERVIDOR:

1. Verificar a Versão do PHP Instalado

Primeiro, verifique qual versão do PHP está instalada no seu sistema:

```
php -v
```

2. Instalar o PHP (se não estiver instalado)

Se o PHP não estiver instalado ou se você precisa de uma versão específica, instale-o usando os seguintes comandos:

- Para PHP 7.4:

```
sudo apt update  
sudo apt install php7.4 libapache2-mod-php7.4
```

3. Reiniciar o Apache

Depois de habilitar o módulo, reinicie o Apache para aplicar as mudanças:

```
sudo systemctl restart apache2
```

4. Montando o laboratório:

form_get.html

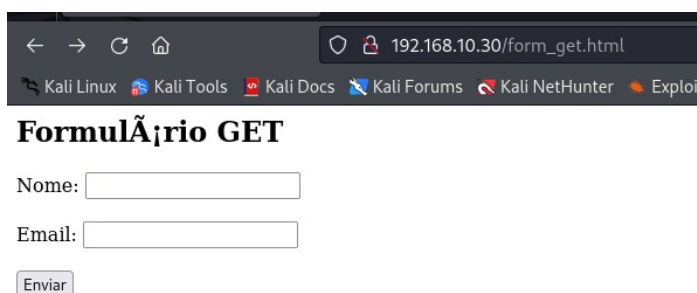
```
<!-- form_get.html -->  
<html>  
<body>  
  <h2>Formulário GET</h2>  
  <form action="process_get.php" method="GET">  
    Nome: <input type="text" name="nome"><br><br>  
    Email: <input type="email" name="email"><br><br>  
    <input type="submit" value="Enviar">  
  </form>  
</body>  
</html>
```

process_get.php

```
<!-- process_get.php -->
<?php
    $nome = $_GET['nome'];
    $email = $_GET['email'];

    echo "<h2>Dados Recebidos via GET</h2>";
    echo "Nome: " . htmlspecialchars($nome) . "<br>";
    echo "Email: " . htmlspecialchars($email) . "<br>";
?>
```

Obs.: enviar estes dois arquivos para a pasta: /var/www/html



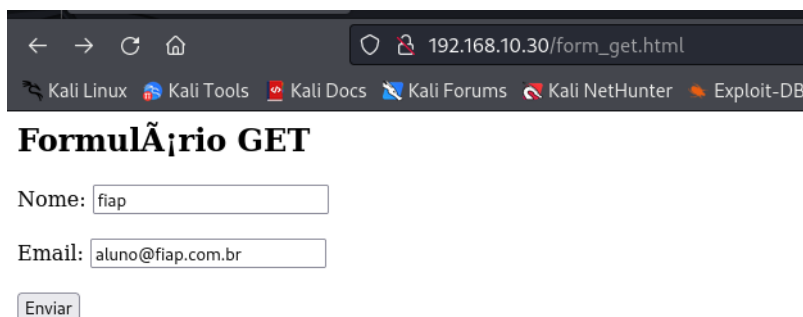
192.168.10.30/form_get.html

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Formulário GET

Nome:

Email:



192.168.10.30/form_get.html

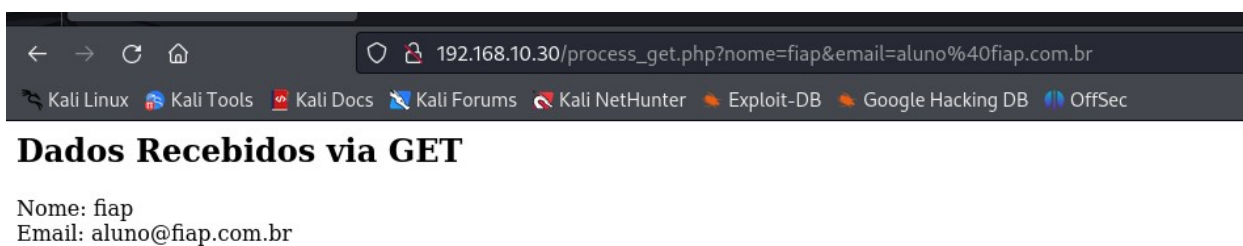
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Formulário GET

Nome:

Email:

Ao enviar:



192.168.10.30/process_get.php?nome=fiap&email=aluno%40fiap.com.br

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Dados Recebidos via GET

Nome: fiap
Email: aluno@fiap.com.br

Análise da Exposição de Dados na URL

- Observar a URL do navegador e notar como os dados inseridos são expostos diretamente na barra de endereço.
- Os riscos associados a essa exposição, como:
 - **Privacidade:** Dados sensíveis podem ser facilmente visualizados por terceiros.
 - **Registro em Logs:** Servidores e proxies podem registrar essas URLs, armazenando dados confidenciais inadvertidamente.
 - **Cache do Navegador:** Navegadores podem armazenar essas URLs, permitindo acesso não autorizado posteriormente.

Manipulação de Parâmetros na URL

- Vamos alterar manualmente os parâmetros na URL e recarregarem a página para ver como os dados exibidos mudam.
- Para dados sensíveis **nunca devem** ser enviados via método GET.

Explorando o Método POST e suas Vulnerabilidades

Criação de um Formulário Simples com Método POST

- Desenvolva um formulário HTML que coleta informações de cadastro, como nome de usuário e senha.
- Configure o formulário para enviar os dados usando o método POST para uma página de processamento.

form_post.html

```
<!-- form_post.html -->
<html>
<body>
  <h2>Formulário POST</h2>
  <form action="process_post.php" method="POST">
    Usuário: <input type="text" name="usuario"><br><br>
    Senha: <input type="password" name="senha"><br><br>
    <input type="submit" value="Registrar">
  </form>
</body>
</html>
```

Obs.: o arquivo deverá ser salvo em: /var/www/html

Processamento e Exibição dos Dados Enviados

- Na página de processamento (process_post.php), capture e exiba os dados recebidos.

process_post.php

```
<!-- process_post.php -->
<?php
  $usuario = $_POST['usuario'];
  $senha = $_POST['senha'];



  echo "<h2>Dados Recebidos via POST</h2>";
  echo "Usuário: " . htmlspecialchars($usuario) . "<br>";
  echo "Senha: " . htmlspecialchars($senha) . "<br>";
?>
```

Obs.: o arquivo deverá ser salvo em: /var/www/html

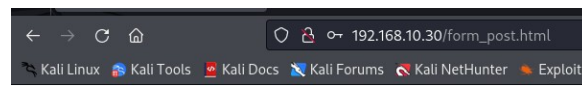
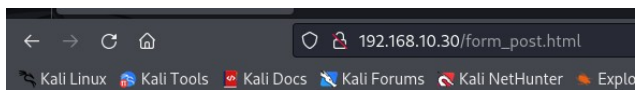
NO CLIENTE:



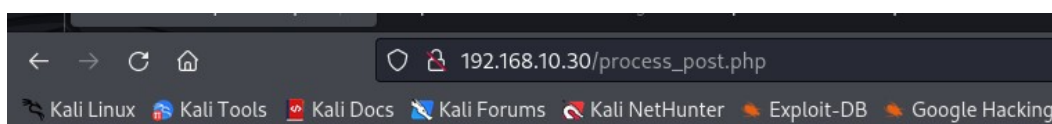
Index of /

Name	Last modified	Size	Description
 form_post.html	2024-09-01 16:36	316	
 process_post.php	2024-09-01 16:36	263	

Apache/2.4.56 (Debian) Server at 192.168.10.30 Port 80



Ao enviar:

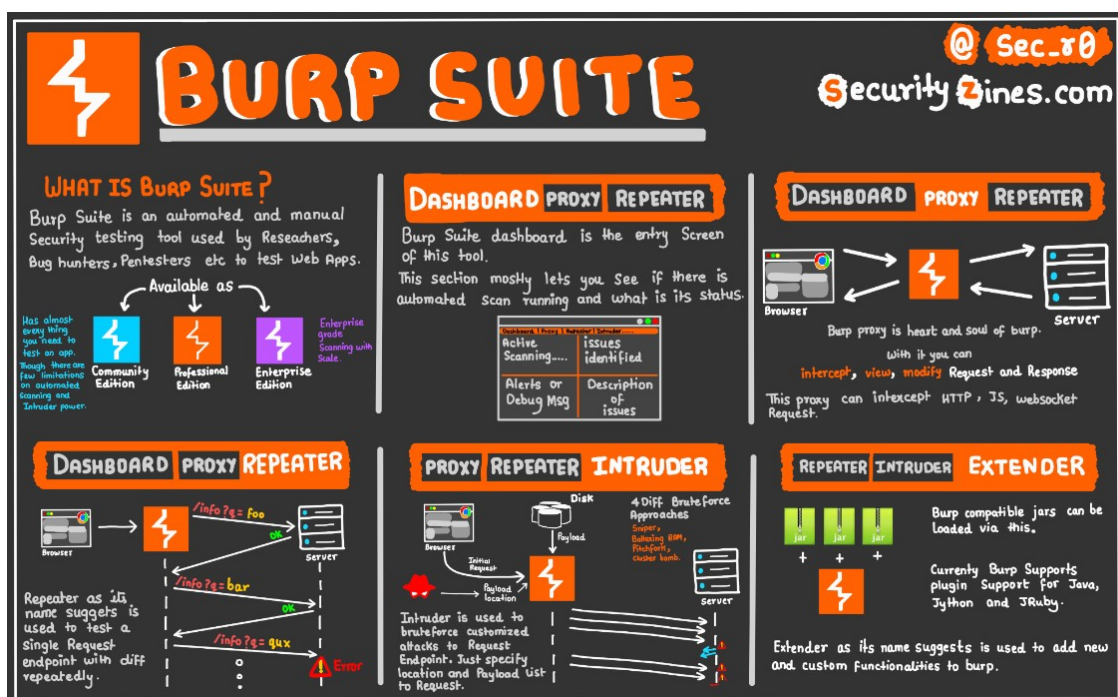


Dados Recebidos via POST

Usuário: aluno
Senha: aluno123

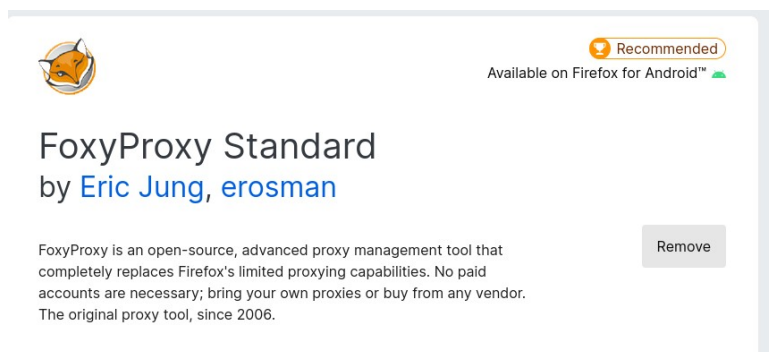
PAUSA EM MÉTODO GET X POST e vamos entender um pouco sobre o BURP SUITE:

Burp Suite é um conjunto líder de ferramentas usadas para testes de segurança de aplicativos da web. Foi desenvolvido pela PortSwigger, uma empresa sediada no Reino Unido, e é amplamente usado por profissionais de segurança e hackers éticos para identificar vulnerabilidades de segurança em aplicativos da web.

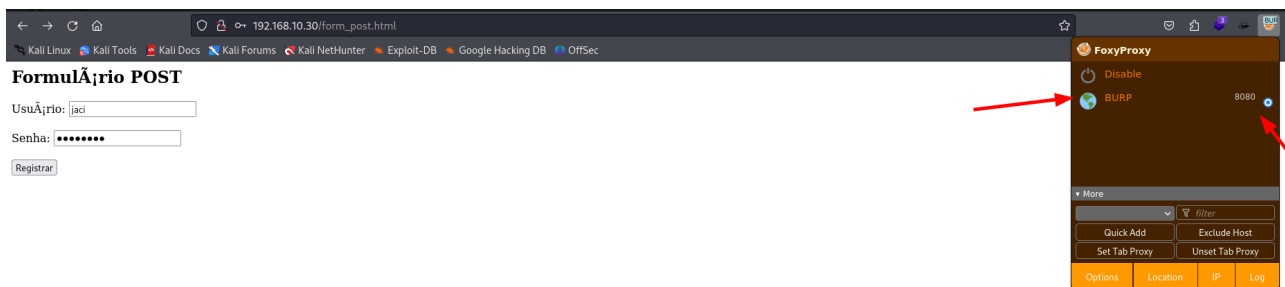


Manipulando dados com Método POST via PROXY

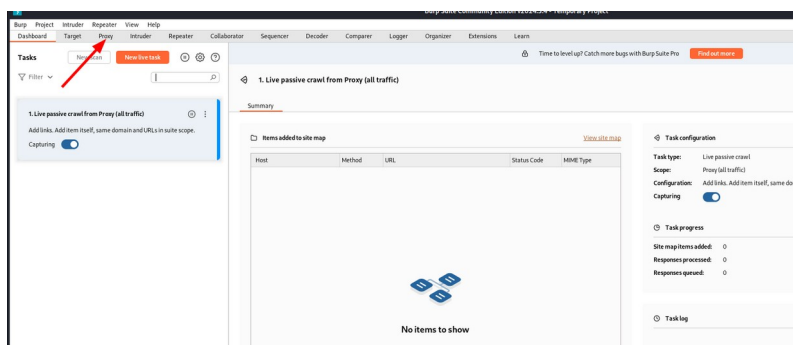
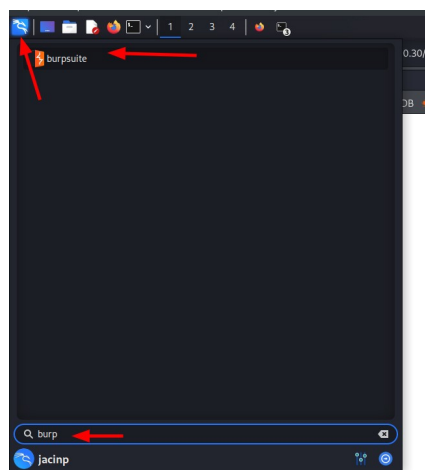
1. Instalação da extensão do Foxproxy no mozilla.



2. Ativação do proxy com o site e os dados inseridos e não enviar/registrar:

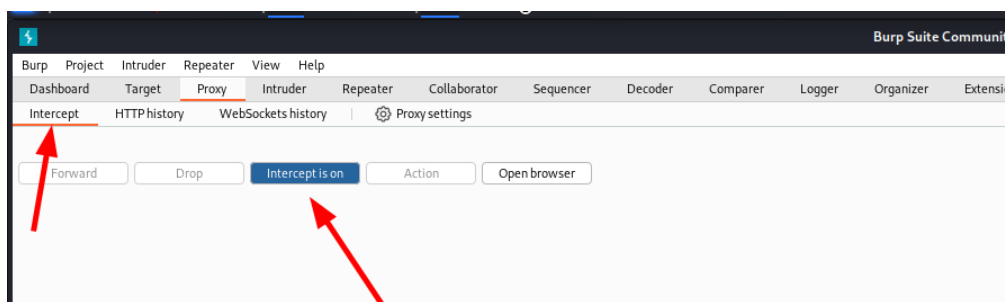


3. Ligar o burp suite

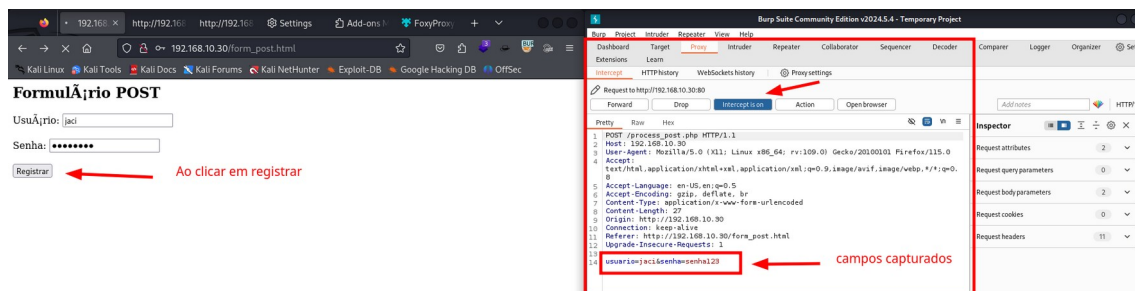


clicar em proxy

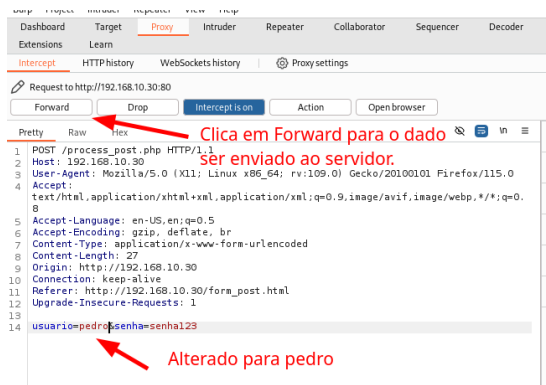
Liga o proxy



Enviar a requisição para o proxy

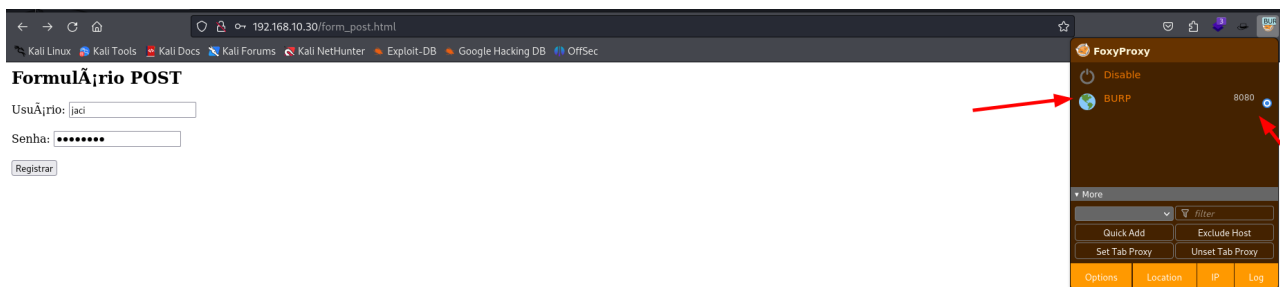


Agora é só alterar o dado que quiser:

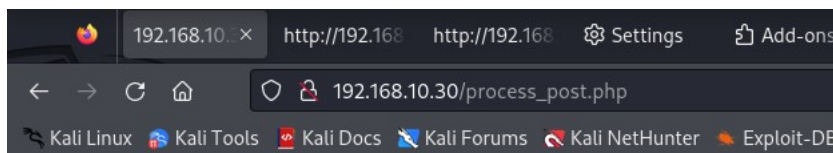


Dados alterados:

1. Antes do proxy



2. Depois do proxy:



Dados Recebidos via POST

Usuário: **pedro** ← Dados alterado
Senha: senha123

ATAQUE DE FORÇA BRUTA EM WEBSITES

Um ataque de força bruta em websites é um método utilizado por atacantes para **adivinhar credenciais (como nomes de usuários e senhas) ou chaves de criptografia**, enviando repetidamente várias combinações até que uma combinação correta seja encontrada. No contexto de websites, esse ataque geralmente é direcionado a **formulários de login**, áreas de administração ou outras partes do site que requerem autenticação.

Para montar um ambiente com login e senha para realizar ataques de força bruta, vamos seguir os seguintes passos:

1. Escolha uma plataforma:

- **PHP e MySQL:** Use um servidor Apache com PHP para criar uma interface simples de login e um banco de dados MySQL para armazenar as credenciais.

2. Página de Login:

- Crie um formulário básico de login com campos de **nome de usuário e senha**.
- O formulário deve ser vulnerável, sem bloqueios de tentativa excessiva de login (não implemente limites de login, CAPTCHA ou 2FA para simular vulnerabilidades).

login.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Login</title>
</head>
<body>
  <h2>Login Form</h2>
  <form action="login.php" method="post">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username"><br><br>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password"><br><br>
    <input type="submit" value="Login">
  </form>
</body>
</html>
```


process_login.php

```
<?php
// Dados de conexão com o MariaDB
$servername = "localhost";
$username = "root";
$password = ""; // Use a senha que você definiu para o root, se aplicável
$dbname = "loginDB";

// Crie a conexão
$conn = new mysqli($servername, $username, $password, $dbname);

// Verifique a conexão
if ($conn->connect_error) {
    die("Conexão falhou: " . $conn->connect_error);
}

// Coletar os dados do formulário de login
$user = $_POST['username'];
$pass = $_POST['password'];

// Consultar o banco de dados
$sql = "SELECT * FROM users WHERE username='$user' AND password='$pass'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "sucesso";
} else {
    echo "incorreto";
}

// Fechar a conexão
$conn->close();
?>
```

3. Configuração do Banco de Dados:

- No MySQL, crie uma tabela com campos de usuário e senha.

O **MariaDB** é uma alternativa compatível ao MySQL e está disponível em muitas distribuições.

Veja como instalar:

1. Atualize a lista de pacotes:

```
sudo apt update
```

2. Instale o MariaDB (substituto do MySQL):

```
sudo apt install mariadb-server mariadb-client
```

3. Verifique a instalação:

Inicie o serviço MariaDB:

```
sudo systemctl start mariadb
```

Verifique se o serviço está em execução:

```
sudo systemctl status mariadb
```

4. Iniciar o MariaDB

Primeiro, certifique-se de que o MariaDB está em execução:

```
sudo systemctl start mariadb  
sudo systemctl enable mariadb
```

```
root@debian:~# systemctl start mariadb  
root@debian:~# systemctl enable mariadb  
Synchronizing state of mariadb.service with SysV service script with /lib/systemd/systemd-sysv-instal  
11.  
Executing: /lib/systemd/systemd-sysv-install enable mariadb  
root@debian:~#
```

5. Acessar o MariaDB

Entre no MariaDB com o usuário root:

```
sudo mysql -u root
```

```
root@debian:~# mysql -u root  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 810  
Server version: 10.5.23-MariaDB-0+deb11u1 Debian 11  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]>
```

6. Criar o Banco de Dados

Agora, vamos criar o banco de dados para armazenar os usuários de login.

```
CREATE DATABASE loginDB;
```

7. Criar a Tabela de Usuários

Dentro do banco de dados, crie uma tabela chamada users com colunas para armazenar o nome de usuário e a senha.

```
USE loginDB;
```

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) NOT NULL,  
  password VARCHAR(255) NOT NULL  
);
```

8. Inserir Dados na Tabela de Usuários

Agora, adicione alguns dados de exemplo (usuário e senha) na tabela para testar o login posteriormente:

```
INSERT INTO users (username, password) VALUES ('admin', '12345');  
INSERT INTO users (username, password) VALUES ('user', 'password123');
```

Você pode usar senhas simples ou, em um ambiente real, criptografar as senhas usando funções como MD5, SHA1, ou preferencialmente bcrypt para maior segurança.

9. Testar a Inserção de Dados

Para garantir que os dados foram inseridos corretamente, você pode usar o comando:

```
SELECT * FROM users;
```

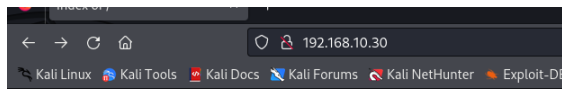
Isso mostrará todos os usuários presentes na tabela.

```
Database changed  
MariaDB [loginDB]> select * from users;  
+-----+-----+-----+  
| id | username | password |  
+-----+-----+-----+  
| 1 | admin | 12345 |  
| 2 | USER | PASSWORD123 |  
| 3 | aluno | fiap |  
+-----+-----+-----+  
3 rows in set (0,000 sec)  
  
MariaDB [loginDB]>
```

4. Ferramenta de Ataque:

- Ferramentas como **Hydra** ou **Burp Suite** podem ser usadas para realizar o ataque de força bruta no ambiente configurado.
- Assegure-se de que o site responde corretamente a tentativas de login inválidas para que os testes possam ser executados de forma eficaz.

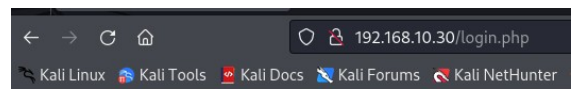
TESTE NO WEBSITE DO LABORATÓRIO



Index of /

Name	Last modified	Size	Description
login.php	2024-09-07 17:55	456	Clica
process_login.php	2024-09-07 21:15	743	

Apache/2.4.56 (Debian) Server at 192.168.10.30 Port 80



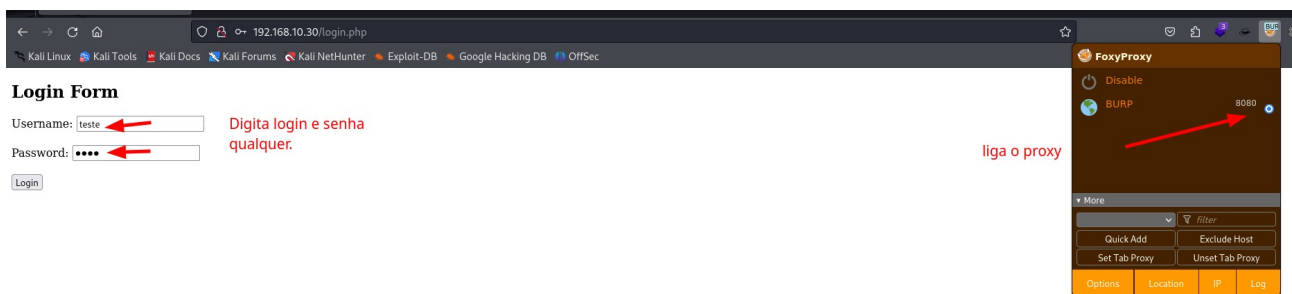
Login Form

Username:

Password:

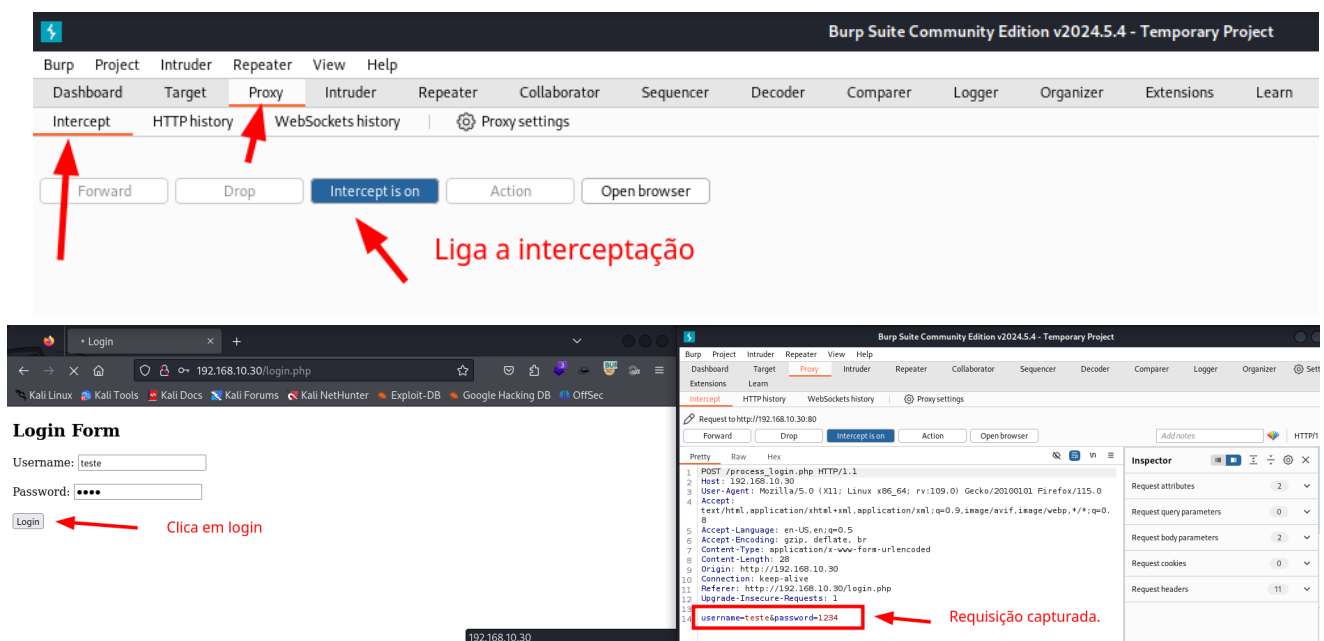
Login

ligar o proxy:

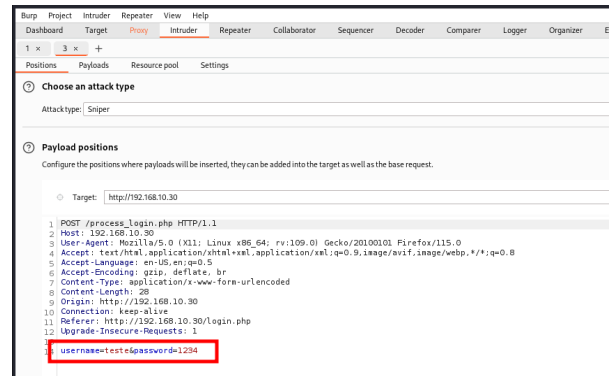
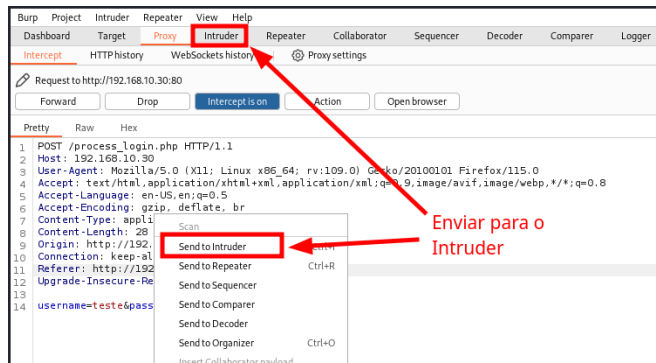


OBS: colocar os dados no formulário e não enviar.

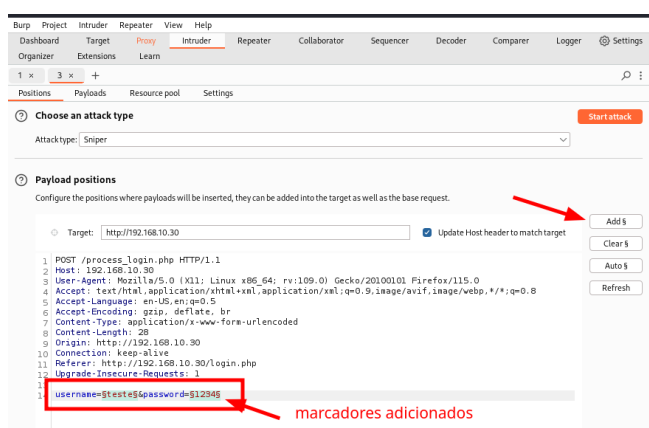
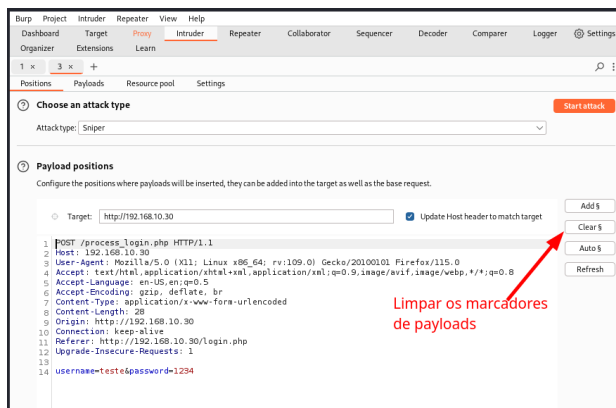
Abrir o burp suite e liga o proxy para interceptar o site:



Enviar para o Intruder:

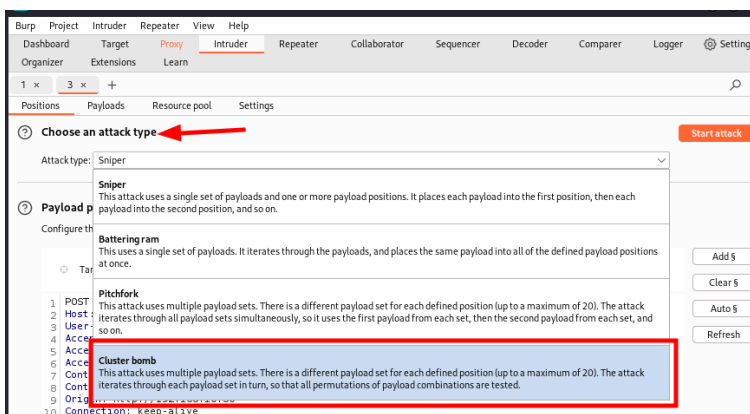


Limpar os marcadores de payloads e adicioná-los novamente.



Iniciar o ataque de força bruta:

Agora vamos escolher o tipo de ataque, neste caso o cluster bomb e configurar o payload:

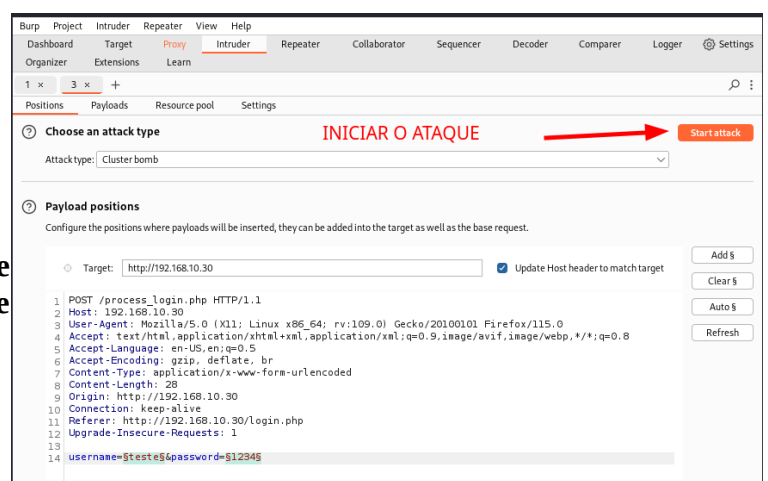


Usuário:



AO CONFIGURAR TODO O ATAQUE, NOS RESTA A INICIAR O ATAQUE!

Ataque iniciado e com flag de erro. O que não estiver a flag de erro, será o login e senha.



Attack Save

2. Intruder attack of http://192.168.10.30

Attack Save

Results Positions Payloads Resource pool Settings

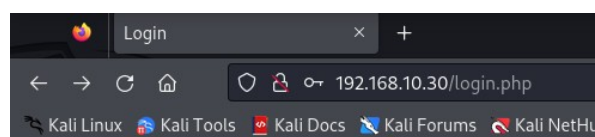
Intruder attack results filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	incorrecto	Comment
0			200	1			212	1	
1	admin	admin	200	1			211	1	
2	manager	admin	200	4			212	1	
3	root	admin	200	2			211	1	
4	cisco	admin	200	2			212	1	
5	apc	admin	200	2			211	1	
6	pass	admin	200	2			212	1	
7	security	admin	200	3			211	1	
8	user	admin	200	2			212	1	
9	system	admin	200	2			211	1	
10	sys	admin	200	3			212	1	
11	wampp	admin	200	2			211	1	
12	newuser	admin	200	3			212	1	
13	xampp-dav-unsecure	admin	200	2			211	1	
14	vagrant	admin	200	3			212	1	
15	admin	12345	200	2			209		
16	manager	12345	200	3			212	1	
17	root	12345	200	2			211	1	
18	cisco	12345	200	3			212	1	

Sem flag!

Login: admin
Senha: 12345

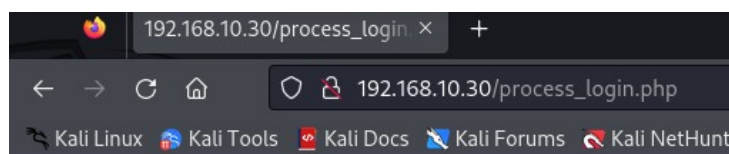
Confirmando no ambiente o login e senha:



Login Form

Username:

Password:



sucesso