# Vulnerabilidades – Práticas de Burp Suite

## 1️⃣ GET – Parâmetro simples

- URL: http://httpbin.org/get?param=teste

- Alteração: param=burp

- Status Code: 200 OK

- Body recebido:

{

"args": {

"param": "burp"

}

}

- Observação: Payload enviado corretamente, servidor ecoou o valor

---

## 2️⃣ POST – Formulário simples

- URL: http://httpbin.org/post

- Alteração: username=admin&password=1234

- Status Code: 200 OK

- Body recebido:

{

"form": {

"username": "admin",

"password": "1234"

}

}

- Observação: Dados recebidos corretamente no Body

---

## 3️⃣ Headers customizados

- Headers modificados:

User-Agent: BurpTest

X-Test-Header: Teste123

- Status Code: 200 OK

- Body recebido: Headers enviados aparecem no Response

- Observação: Teste de alteração de headers concluído

---

## 4️⃣ Cookies

- Cookie inicial: role=user

- Alteração: role=admin

- Status Code: 200 OK

- Body recebido: Cookie modificado aparece no Response

- Observação: Simulação de alteração de privilégios

---

## 5️⃣ XSS Simulado GET

- URL: http://httpbin.org/get?input=%3Cscript%3Ealert(%22burp%22)%3C/script%3E

- Status Code: 200 OK

- Body recebido:

{

"args": {

"input": "<script>alert(\"burp\")</script>"

}

}

- Observação: Payload codificado enviado corretamente, sem risco, apenas simulação

---

## 6️⃣ XSS Simulado POST

- URL: http://httpbin.org/post

- Alteração: comment=%3Cscript%3Ealert(%22burp%22)%3C/script%3E

- Status Code: 200 OK

- Body recebido: Payload recebido no Body do POST

- Observação: Simulação segura de XSS em formulário POST

---

Resumo: Todos os testes foram realizados em ambiente seguro (httpbin.org), permitindo treinar manipulação de GET/POST, headers, cookies e payloads sem risco. Status Code e Body foram registrados para análise futura.