Tutorial

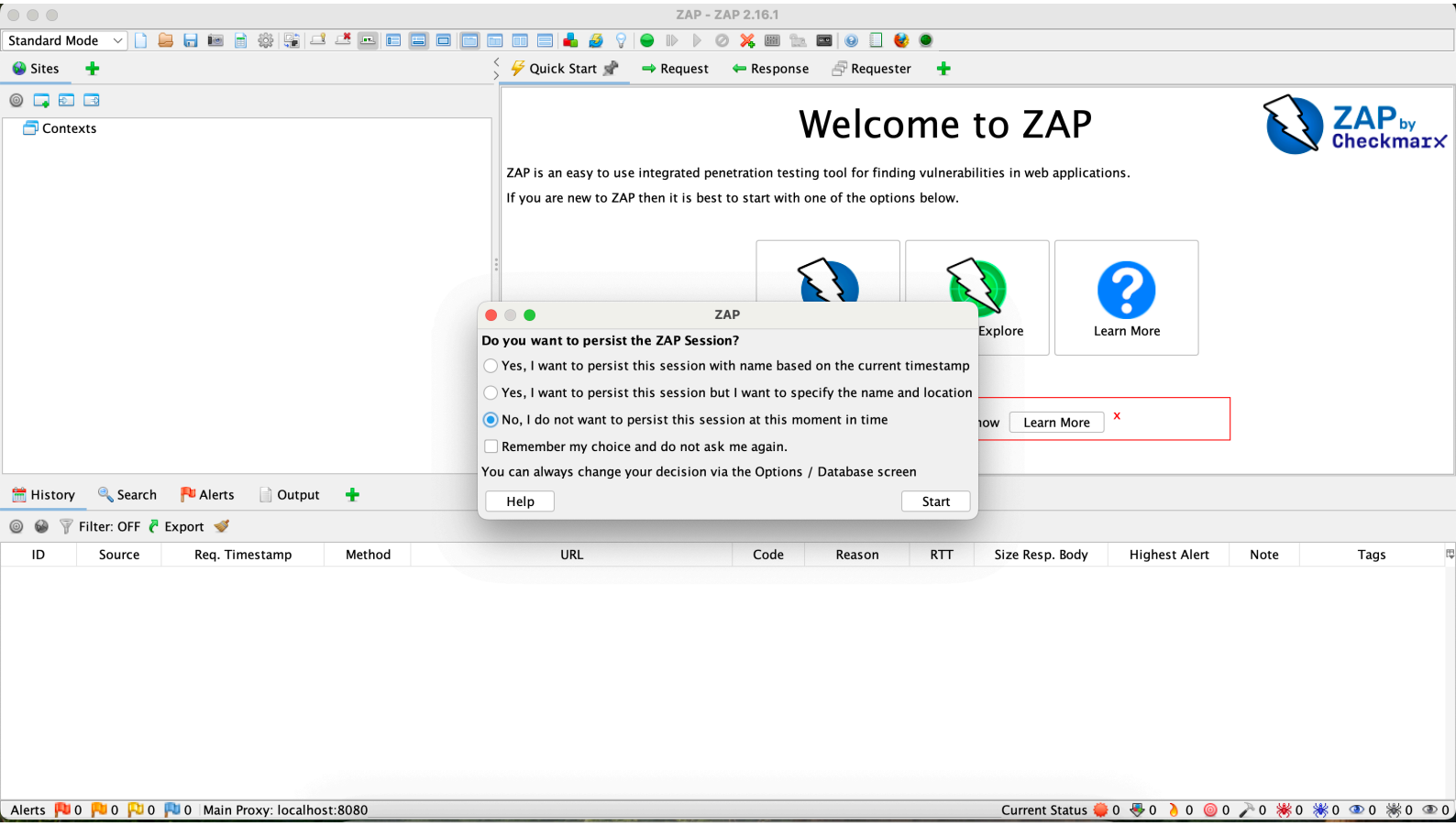
Zed Attack Proxy (OWASP ZAP)

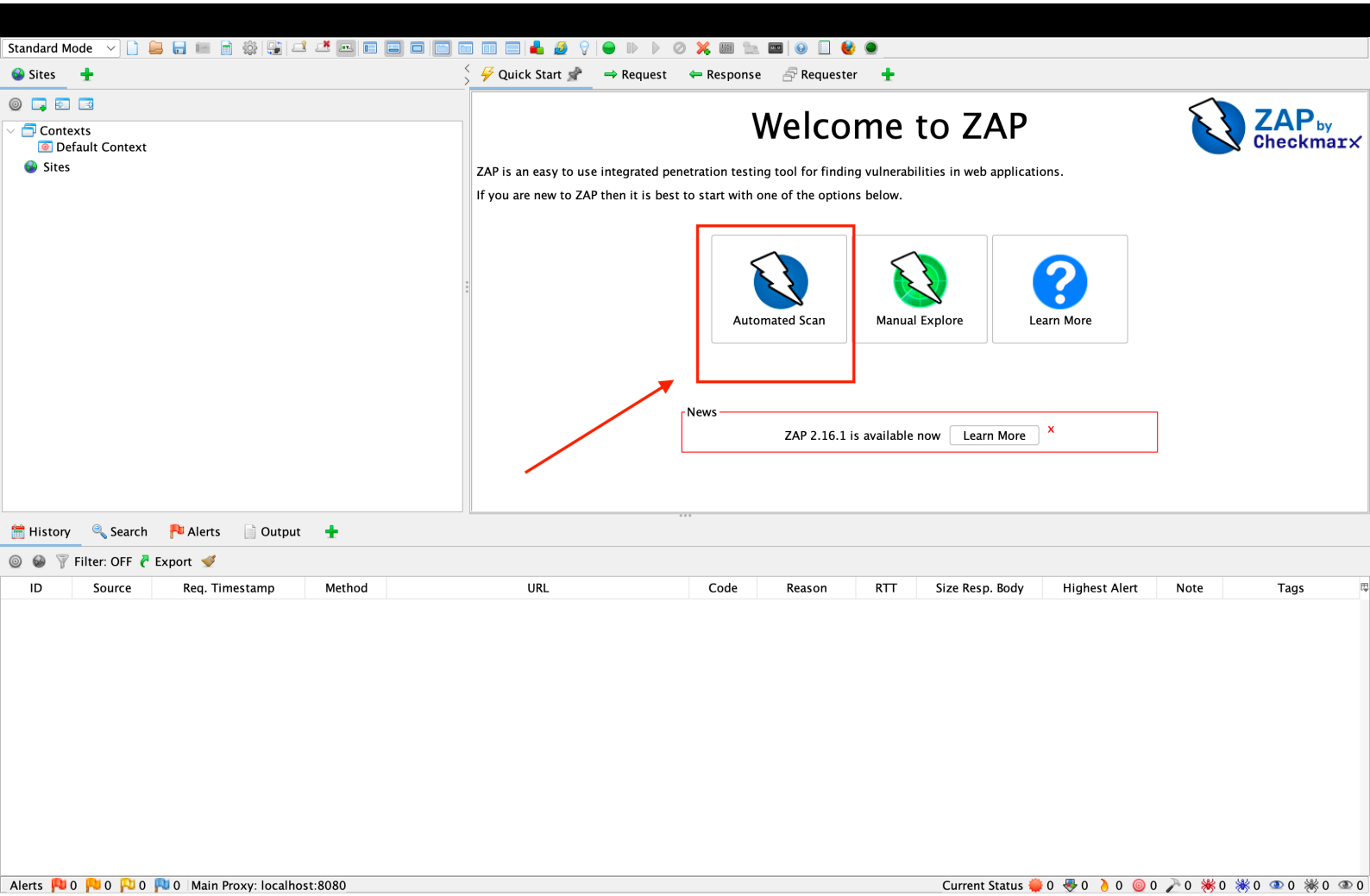
1. Baixe a ferramenta (https://www.zaproxy.org/)
2. Abra o programa e escolha a opção "*No, I do not want to persist this session at this moment in time*"
3. Na janela central, clique em "Automatic Scan"
4. Rode o site localmente e digite o endereço do servidor local
5. Clique em "*Attack*"
6. Espere a execução terminar. Os alertas aparecerão na parte de baixo
7. Para cada alerta faça o seguinte:
    1. Clique no alerta
    2. Verifique a CWE da vulnerabilidade
    3. Verifique se a CWE faz parte de algum dos 3 tipos de vulnerabilidades que estamos buscando mitigar (ao final de cada página a seguir há uma seção chamada "*List of Mapped CWEs*")
        1. https://owasp.org/Top10/A01_2021-Broken_Access_Control/
        2. https://owasp.org/Top10/A02_2021-Cryptographic_Failures/
        3. https://owasp.org/Top10/A03_2021-Injection/
8. Caso faça parte, leia sobre o alerta e como mitigá-lo (tanto na ferramenta como nos endereços acima)
9. Crie uma *issue* no repositório para a correção do problema e como corrigi-o
10. Gere o relatório no programa e envie no repositório o arquivo HTML gerado (Menu Superior > Report > Generate Report) dentro da pasta "entregas"

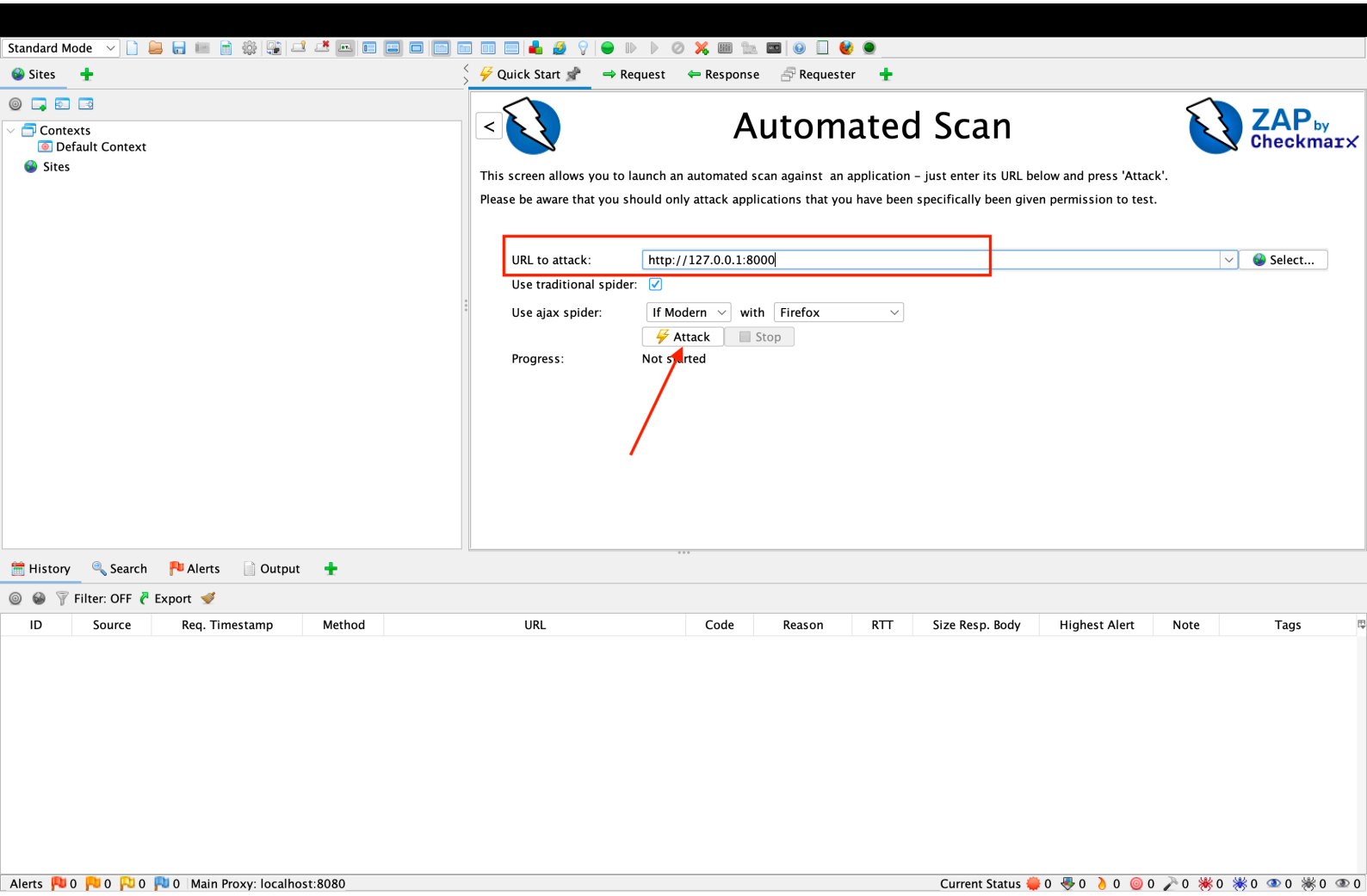
Observação:

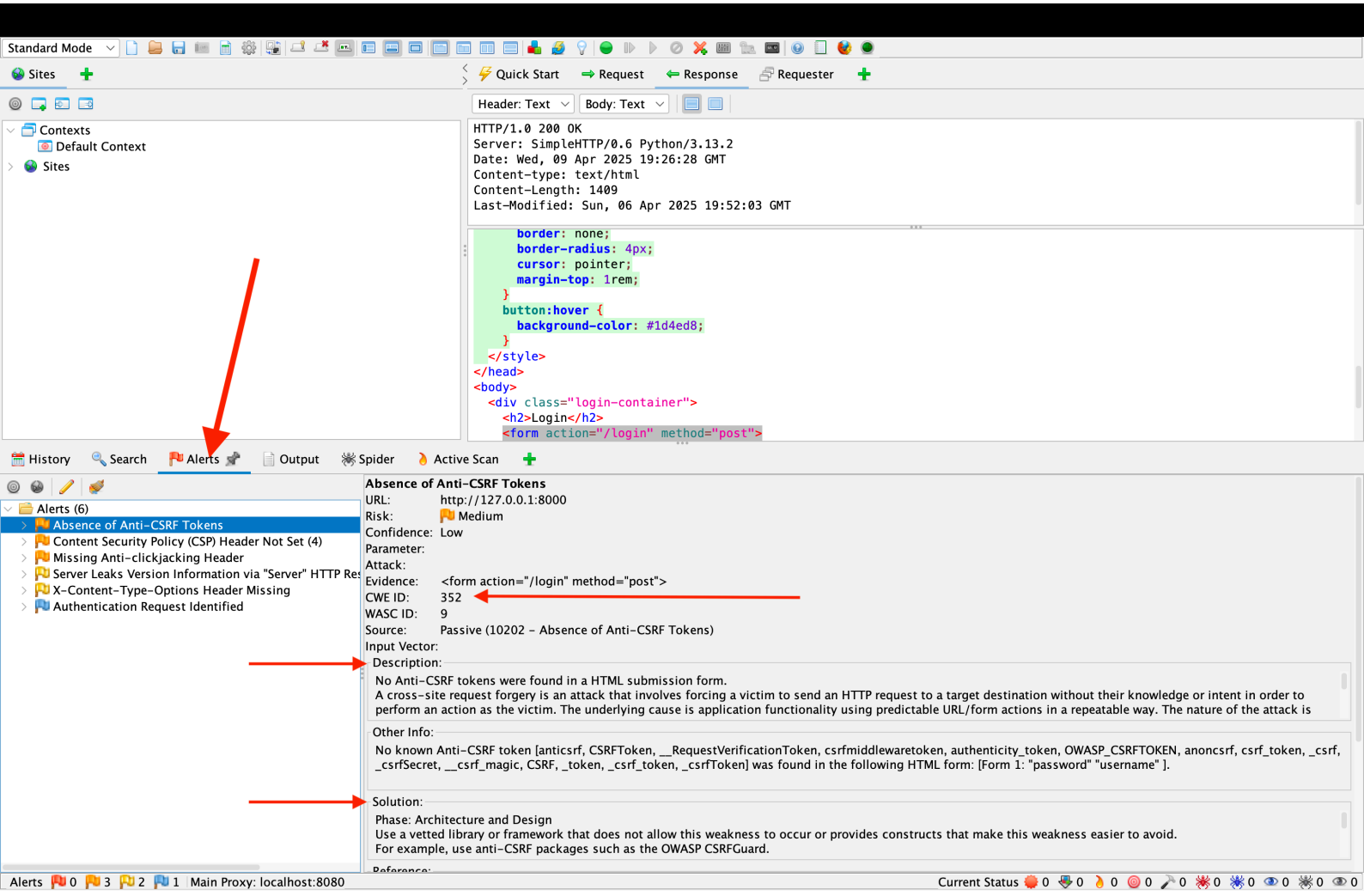
Caso suas telas não estejam todas referenciadas e navegáveis a partir do arquivo-raiz, você terá que fazer a etapa anterior para cada página.
Quando o projeto estiver todo interconectado e navegável, isso não será mais necessário.

Prints do passo-a-passo abaixo


ZAP - ZAP 2.16.1
Standard Mode
Sites
Contexts
Quick Start
Request
Response
Requester
Welcome to ZAP
ZAP by Checkmarx
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
If you are new to ZAP then it is best to start with one of the options below.
Explore
Learn More
Learn More
ZAP
Do you want to persist the ZAP Session?
Yes, I want to persist this session with name based on the current timestamp
Yes, I want to persist this session but I want to specify the name and location
No, I do not want to persist this session at this moment in time
Remember my choice and do not ask me again.
You can always change your decision via the Options / Database screen
Help
Start
History
Search
Alerts
Output
Filter: OFF
Export
ID
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0
Main Proxy: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0 0


Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
Welcome to ZAP
ZAP by Checkmarx
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
If you are new to ZAP then it is best to start with one of the options below.
Automated Scan
Manual Explore
Learn More
News
ZAP 2.16.1 is available now
Learn More
History
Search
Alerts
Output
Filter: OFF
Export
ID
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0
Main Proxy: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0 0


Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
Automated Scan
ZAP by Checkmarx
This screen allows you to launch an automated scan against an application – just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack:
http://127.0.0.1:8000
Select...
Use traditional spider:
Use ajax spider:
If Modern
with
Firefox
Attack
Stop
Progress:
Not started
History
Search
Alerts
Output
Filter: OFF
Export
ID
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0
Main Proxy: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0 0


Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.13.2
Date: Wed, 09 Apr 2025 19:26:28 GMT
Content-type: text/html
Content-Length: 1409
Last-Modified: Sun, 06 Apr 2025 19:52:03 GMT
border: none;
border-radius: 4px;
cursor: pointer;
margin-top: 1rem;
}
button:hover {
background-color: #1d4ed8;
}
</style>
</head>
<body>

<h2>Login</h2>
<form action="/login" method="post">
History
Search
Alerts
Output
Spider
Active Scan
Alerts (6)
Absence of Anti-CSRF Tokens
Content Security Policy (CSP) Header Not Set (4)
Missing Anti-clickjacking Header
Server Leaks Version Information via "Server" HTTP Res
X-Content-Type-Options Header Missing
Authentication Request Identified
Absence of Anti-CSRF Tokens
URL: http://127.0.0.1:8000
Risk: Medium
Confidence: Low
Parameter:
Attack:
Evidence: <form action="/login" method="post">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)
Input Vector:
Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is
Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "password" "username"].
Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.
Alerts 0 3 2 1
Main Proxy: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0 0

owasp.org

A01 Broken Access Control

OWASP/Top10 4.6k 872

OWASP Top 10:2021

- Home
- Notice
- Introduction
- How to use the OWASP Top 10 as a standard
- How to start an AppSec program with the OWASP Top 10
- About OWASP

Top 10:2021 List

- A01 Broken Access Control
- A02 Cryptographic Failures
- A03 Injection
- A04 Insecure Design
- A05 Security Misconfiguration
- A06 Vulnerable and Outdated Components
- A07 Identification and Authentication Failures
- A08 Software and Data Integrity Failures
- A09 Security Logging and Monitoring Failures
- A10 Server Side Request Forgery (SSRF)

Next Steps

List of Mapped CWEs

CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

CWE-23 Relative Path Traversal

CWE-35 Path Traversal: '.../...//'

CWE-59 Improper Link Resolution Before File Access ('Link Following')

CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

CWE-201 Exposure of Sensitive Information Through Sent Data

CWE-219 Storage of File with Sensitive Data Under Web Root

CWE-264 Permissions, Privileges, and Access Controls (should no longer be used)

CWE-275 Permission Issues

CWE-276 Incorrect Default Permissions

CWE-284 Improper Access Control

CWE-285 Improper Authorization

CWE-352 Cross-Site Request Forgery (CSRF)

CWE-359 Exposure of Private Personal Information to an Unauthorized Actor

CWE-377 Insecure Temporary File

CWE-402 Transmission of Private Resources into a New Sphere ('Resource Leak')

CWE-425 Direct Request ('Forced Browsing')

CWE-441 Unintended Proxy or Intermediary ('Confused Deputy')

CWE-497 Exposure of Sensitive System Information to an Unauthorized Control Sphere

Table of contents

- Factors
- Overview
- Description
- How to Prevent
- Example Attack Scenarios
- References
- List of Mapped CWEs

ZAP File Edit View Analyse Report Tools Import Export Online Help

Wed 9 Apr 16:37

Compare with Another Session...
Generate Report ...

Untitled Session - ZAP 2.16.1

Sites | Quick Start | Request | Response | Requester

Header: Text | Body: Text

Contexts
Default Context
Sites

```
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.13.2
Date: Wed, 09 Apr 2025 19:26:28 GMT
Content-type: text/html
Content-Length: 1409
Last-Modified: Sun, 06 Apr 2025 19:52:03 GMT
```

```
      border: none;
      border-radius: 4px;
      cursor: pointer;
      margin-top: 1rem;
    }
    button:hover {
      background-color: #1d4ed8;
    }
  </style>
</head>
<body>
  <div class="login-container">
    <h2>Login</h2>
    <form action="/login" method="post">
```

History | Search | Alerts | Output | Spider | Active Scan

Alerts (6)
- Absence of Anti-CSRF Tokens
- Content Security Policy (CSP) Header Not Set (4)
- Missing Anti-clickjacking Header
- Server Leaks Version Information via "Server" HTTP Res
- X-Content-Type-Options Header Missing
- Authentication Request Identified

Absence of Anti-CSRF Tokens
URL: http://127.0.0.1:8000
Risk: Medium
Confidence: Low
Parameter:
Attack:
Evidence: <form action="/login" method="post">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)
Input Vector:

Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is

Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "password" "username"].

Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Reference:

Alerts 0 3 2 1 | Main Proxy: localhost:8080 | Current Status 0 0 0 0 0 0 0 0 0 0