Using Burp to Test for Security Misconfiguration Issues

Application misconfiguration attacks exploit configuration weaknesses found in web applications.

Security misconfiguration can happen at any level of an application stack, including the platform, web server, application server, database, and framewor

Many applications come with unnecessary and unsafe features, such as debug and QA features, enabled by default. These features may provide a mea hacker to bypass authentication methods and gain access to sensitive information, perhaps with elevated privileges.

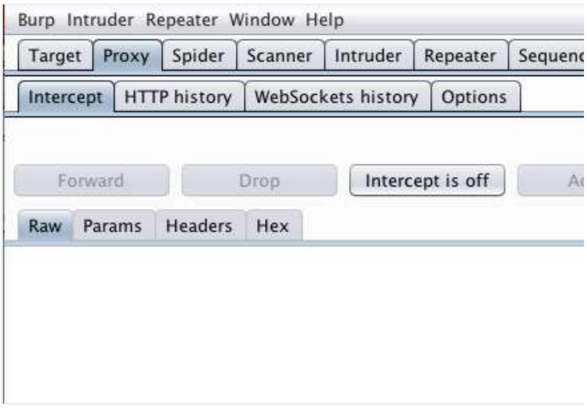
Likewise, default installations may include well-known usernames and passwords, hard-coded backdoor accounts, special access mechanisms, and inco permissions set for files accessible through web servers.

In this example we will demonstrate how to use Burp Spider and/or Site map to check for directory listings. This tutorial uses an exercise from the “Mutilli training tool.

The version of “Mutillidae” we are using is taken from OWASP’s Broken Web Application Project. Find out how to download, install and use this project.

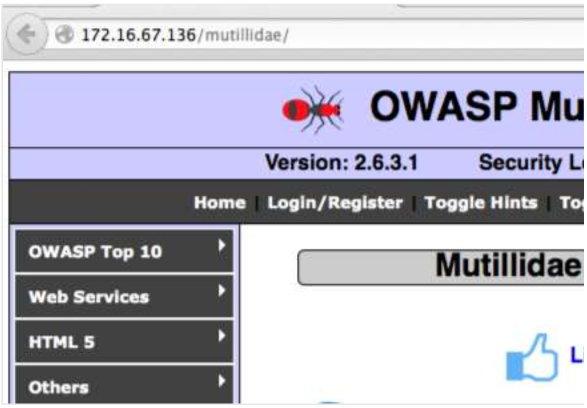
First, ensure that Burp is correctly configured with your browser.

Ensure Burp Proxy "Intercept is off".



In your browser, visit the page of the web application you are testing.

In this example start by browsing to the Mutillidae home page.

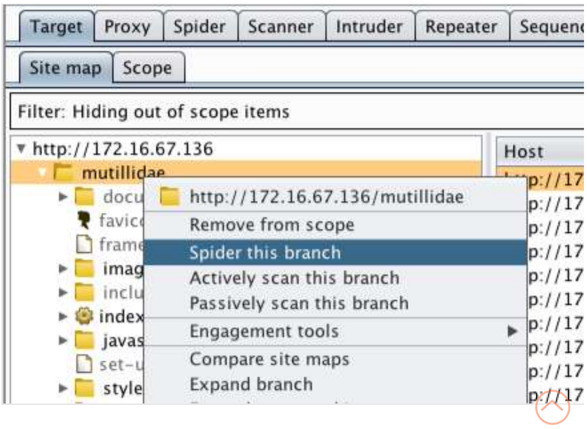


Return to Burp.

Select the "Target" tab and then the "Site map" tab.

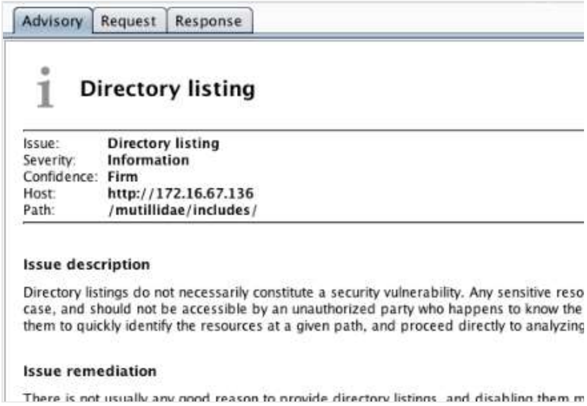
Locate and right click on the "Mutillidae" folder to bring up the context menu..

Click "Spider from here".



Although not necessarily a security vulnerability, directory listings are reported by Burp Scanner.

For example, if you have passive scanning enabled when you spider this application, "Directory listing" will be included in the Scanner "Results" tab.



Directory listing

Issue: Directory listing
Severity: Information
Confidence: Firm
Host: http://172.16.67.136
Path: /mutillidae/includes/

Issue description

Directory listings do not necessarily constitute a security vulnerability. Any sensitive reso
case, and should not be accessible by an unauthorized party who happens to know the
them to quickly identify the resources at a given path, and proceed directly to analyzing

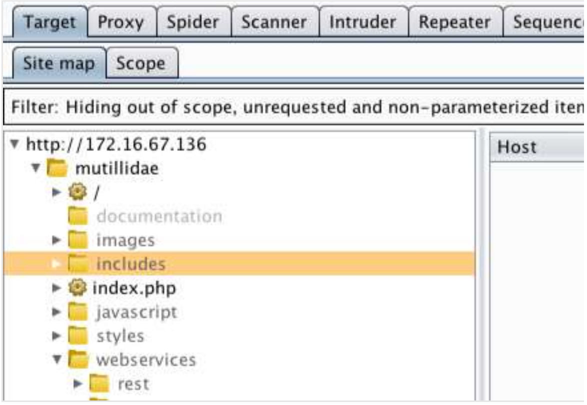
Issue remediation

There is not usually any good reason to provide directory listings, and disabling them m

Go to the "Target" tab and then the "Site map" tab.

Here you can view the site map for the web application which has been populated by Burp Spider.

Select an interesting branch from the Site map. In this case we will explore the "Includes" directory.

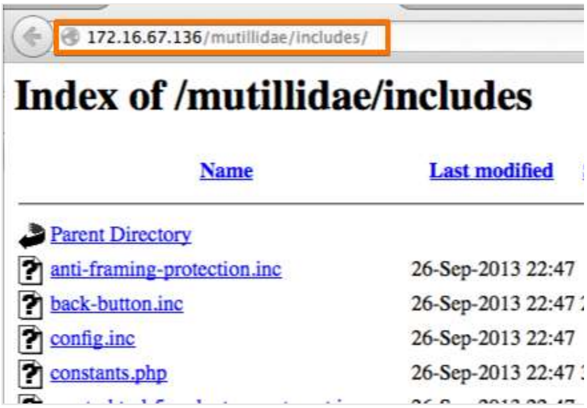


Filter: Hiding out of scope, unrequested and non-parameterized iten

```
▼ http://172.16.67.136
  ▼ mutillidae
    ▶ /
      documentation
    ▶ images
    ▶ includes
    ▶ index.php
    ▶ javascript
    ▶ styles
    ▼ webservices
      ▶ rest
```

Return to your browser and access the directories you have chosen to investigate by adding the directory name to the URL.

In this example: /mutillidae/includes/.

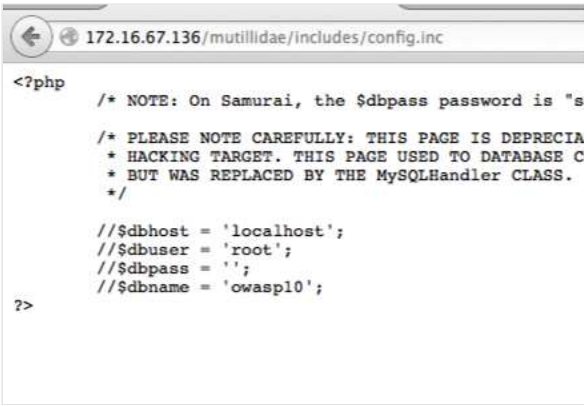


172.16.67.136/mutillidae/includes/

Index of /mutillidae/includes

Name	Last modified
Parent Directory	
anti-framing-protection.inc	26-Sep-2013 22:47
back-button.inc	26-Sep-2013 22:47
config.inc	26-Sep-2013 22:47
constants.php	26-Sep-2013 22:47

Explore the links in each file and directory you are able to find.



172.16.67.136/mutillidae/includes/config.inc

```
<?php
        /* NOTE: On Samurai, the $dbpass password is "s

        /* PLEASE NOTE CAREFULLY: THIS PAGE IS DEPRECIA
         * HACKING TARGET. THIS PAGE USED TO DATABASE C
         * BUT WAS REPLACED BY THE MySQLHandler CLASS.
         */

        //$dbhost = 'localhost';
        //$dbuser = 'root';
        //$dbpass = '';
        //$dbname = 'owasp10';
?>
```

Related articles:

Getting started with Burp Proxy

Getting started with Burp Scanner

Burp's target site map

Getting Started with Burp Spider

Burp Suite

Web vulnerability scanner
Burp Suite Editions
Release Notes

Vulnerabilities

Cross-site scripting (XSS)
SQL injection
Cross-site request forgery
XML external entity injection
Directory traversal
Server-side request forgery

Customers

Organizations
Testers
Developers

Company

About
PortSwigger News
Careers
Contact
Legal
Privacy Notice

Insights

Web Security Academy
Blog
Research
The Daily Swig

PortSwig

Follow us