

Using Burp to Test for Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or a URL or form parameter. An attacker can manipulate direct object references to access other objects without authorization, unless an **access control** check is in place.

In our example the application's **authentication** / authorization functionality does not prevent one user from gaining access to another user's data or records by modifying the key value identifying the data.

In this example we will demonstrate how to use Burp Intruder and Repeater to check for insecure direct object reference vulnerabilities. This tutorial uses an exercise from the "Cyclone" training tool. The version of "Cyclone" 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**.

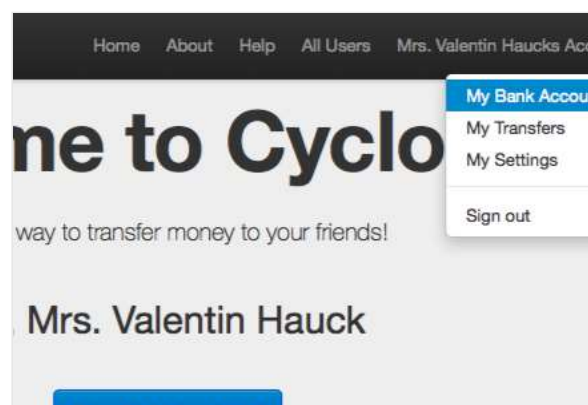
With intercept turned off in the **Proxy** "Intercept" tab, visit the web application you are testing in your browser.



Visit the page of the web application you are going to attack.

In this example log in to "Cyclone" using the login details provided on the homepage.

Then click the "My Bank Accounts" link from the "Account" drop down menu.

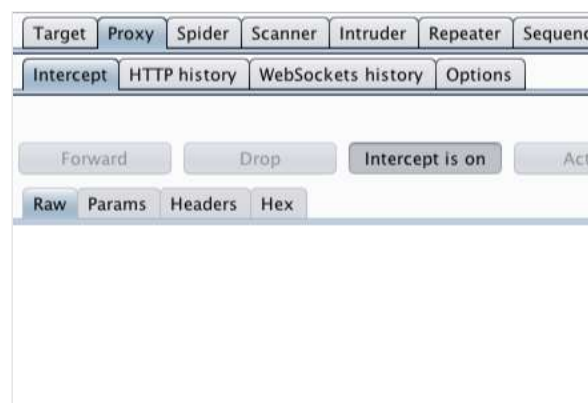


Return to Burp.

In the **Proxy** "Intercept" tab, ensure "Intercept is on".

In your browser, reload the page.

The request will be captured by Burp.

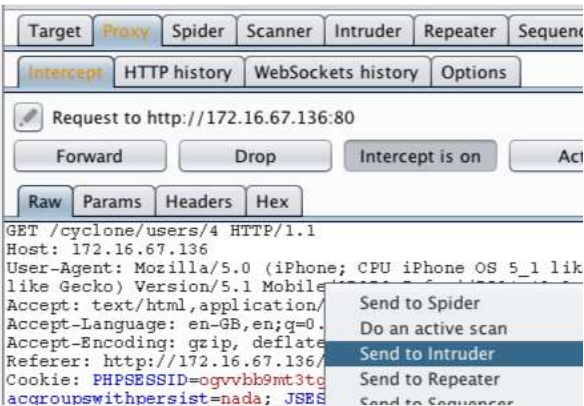


View the request in the **Proxy** "Intercept" tab.

Right click on the raw request to bring up the context menu.

Click "Send to **Intruder**."

Note: You can also send requests to Intruder via the context menu in any location where HTTP requests are shown, such as the site map or Proxy history.

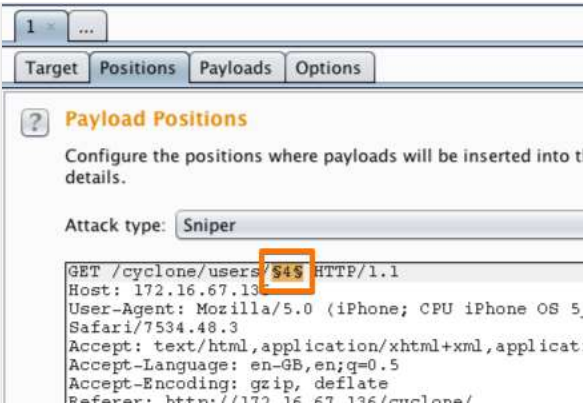


Go to the **Intruder** tab, then the **Positions** tab.

Use the "Clear" function to remove the preset payload positions..

Highlight the section of the URL that refers to an object. In this case the user number in the URL.

Use the "Add" button on the right of the request editor to add the selected payload position.



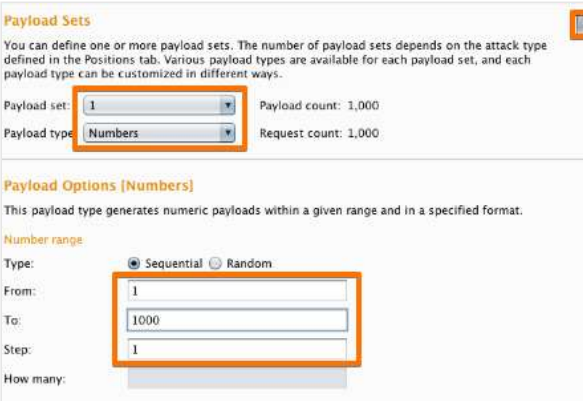
Next, go to the **Payloads** tab.

Here you can select a payload type to suit the attack you are implementing. In this case select "Payload type:" "Numbers" from the "Payload Sets" options.

Beneath "Payload Options" "you can choose the number range and increments.

In this example we are using the numbers 1-1000 in increments of 1.

Once you have tailored your attack, click the "Start Attack" Button.



In the **Intruder** attack window you can sort the results of the attack by a variety of means.

In this example we can use "Status" and/or "Length".

The results are split quite clearly and provide us with means for further investigation.

Request	Payload	Status	Error	Timeout	Length
0		304	<input type="checkbox"/>	<input type="checkbox"/>	667
4	4	304	<input type="checkbox"/>	<input type="checkbox"/>	667
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	756
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	756
17	17	200	<input type="checkbox"/>	<input type="checkbox"/>	756
23	23	200	<input type="checkbox"/>	<input type="checkbox"/>	756
43	43	200	<input type="checkbox"/>	<input type="checkbox"/>	756
68	68	200	<input type="checkbox"/>	<input type="checkbox"/>	756
75	75	200	<input type="checkbox"/>	<input type="checkbox"/>	756
15	15	200	<input type="checkbox"/>	<input type="checkbox"/>	756
20	20	200	<input type="checkbox"/>	<input type="checkbox"/>	756

To perform further investigation of interesting results, you can:

Send the item to the Repeater tool, via the context menu.

Copy the URL, via the context menu, and paste it into your browser.

Explore the request and response in the attack window.

In this example we are able to examine the request and response in the "Intruder attack" window.

Requests 1-100 (apart from the original user ID of 4) enumerate the user names of other accounts in the web application.



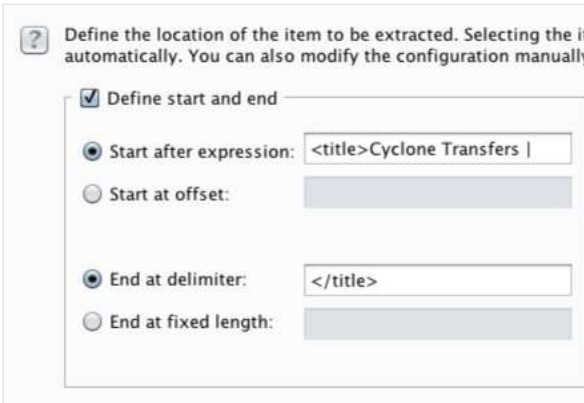
Additionally, you can use the "Grep - Extract" function to add the user names to the results table.

Go to the "Options" tab in the attack window.

Then locate the "Grep - Extract" options and click the "Add" button.



Here you can define the location of the item to be extracted from the HTTP response.



With the grep extraction configured, the results table will be populated with the defined items, in this example the usernames of other account holders.

Status	Error	Timeout	Length	<title>Cyclone Tr
304	<input type="checkbox"/>	<input type="checkbox"/>	667	
200	<input type="checkbox"/>	<input type="checkbox"/>	7565	John Smith
200	<input type="checkbox"/>	<input type="checkbox"/>	7585	Herminio Langwor
200	<input type="checkbox"/>	<input type="checkbox"/>	7579	Luciano Connelly
200	<input type="checkbox"/>	<input type="checkbox"/>	7567	Rocky Jast
200	<input type="checkbox"/>	<input type="checkbox"/>	7575	Ally Greenholt
200	<input type="checkbox"/>	<input type="checkbox"/>	7581	Keshaun Wilderma
200	<input type="checkbox"/>	<input type="checkbox"/>	7577	Gussie Halvorson
200	<input type="checkbox"/>	<input type="checkbox"/>	7577	Matt Harvey III
200	<input type="checkbox"/>	<input type="checkbox"/>	7573	Adella Zemlak
200	<input type="checkbox"/>	<input type="checkbox"/>	7575	Manuel Gislason
200	<input type="checkbox"/>	<input type="checkbox"/>	7591	Miss Gudrun McCa
200	<input type="checkbox"/>	<input type="checkbox"/>	7579	Maximilian Purdy

You can use Burp **Scanner** alongside your manual testing methodology to quickly identify many types of common vulnerabilities.

File path traversal is one example of the Scanner's ability to locate issues of this nature.

AdvisoryRequestResponse

!

File path traversal

Issue:

File path traversal

Severity:

High

Confidence:

Firm

Host:

http://172.16.67.136

Path:

/mutillidae/index.php

Issue detail

The **page** parameter is vulnerable to path traversal attacks, enabling read access to arl

The payload `../../../../../../../../../../../../etc/passwd` was submitted in the pag response.

Issue background

Related articles:

Using Burp Intruder

Getting started with Burp Proxy

Using Burp Repeater

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

PortSwigger

Follow us

© 2020 PortSwigger Ltd

⬆