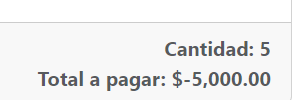
Práctica. CWE-191: Integer Underflow

Instrucciones. Elaborar una aplicación de PHP que permita ejemplificar la vulnerabilidad Desbordamiento inferior de enteros en un servidor Web.

1. Sigue las instrucciones de la práctica en la siguiente página.
2. Crea un reporte de práctica en un archivo de Word con una portada con tu nombre. Guárdalo con el nombre NombreAlumno\_CWE-191.docx.
3. Coloca las siguientes capturas de pantalla, cada una con una descripción textual arriba de la captura.
   1. Coloca la captura de pantalla de tu código fuente en Visual Studio Code del archivo index.php.
   2. Coloca la captura de pantalla de la salida de la página index.php atacada con el payload que muestre un valor negativo en el total de la compra.



* 1. Publica tu código fuente en un proyecto público de GitHub. Coloca la URL del código fuente publicado en GitHub.

1. Sube tu reporte de práctica en un archivo Word a la actividad en Eminus. Prerrequisitos
2. Esta práctica tiene como prerrequisito tener instalado el software Visual Studio Code con la extensión para PHP.

*Puede seguir las instrucciones de cómo instalar Visual Studio Code desde la práctica* [*Instalación de*](https://www.uv.mx/personal/gvera/files/2024/01/PRAC-Instalacion-de-Visual-Studio-Code.pdf)[*Visual Studio Code*](https://www.uv.mx/personal/gvera/files/2024/01/PRAC-Instalacion-de-Visual-Studio-Code.pdf)*.*

1. Esta práctica tiene como prerrequisito tener instalado el intérprete de PHP en su equipo.

*Puede seguir las instrucciones de cómo instalar PHP desde la práctica* [*Instalación de PHP*](https://www.uv.mx/personal/gvera/files/2024/03/PRAC-Instalacion-de-PHP.pdf)*.*

1. Esta práctica tiene como prerrequisito contar con una cuenta activa de GitHub en <https://github.com/>.

*Puede seguir las instrucciones de cómo crear una cuenta de GitHub desde la práctica* [*Crear cuenta*](https://www.uv.mx/personal/gvera/files/2024/01/PRAC-Crear-una-cuenta-de-GitHub.pdf)[*de GitHub*](https://www.uv.mx/personal/gvera/files/2024/01/PRAC-Crear-una-cuenta-de-GitHub.pdf)*.*

1. Puede seguir las instrucciones de cómo publicar un proyecto en Visual Studio Code a GitHub desde la

práctica [Como publicar un proyecto a GitHub desde Visual Studio Code](https://www.uv.mx/personal/gvera/files/2024/01/PRAC-Como-publicar-un-proyecto-a-GitHub-desde-VS-Code.pdf).

Instrucciones

El producto resta un valor de otro, de modo que el resultado es menor que el valor entero mínimo

permitido, lo que produce un valor que no es igual al resultado correcto. En esta práctica vamos a recrear una vulnerabilidad muy común en el manejo de cantidades de elementos, cuando solo se verifica que se envíe un número, es decir, validación sintáctica, pero no se valida que no sea un numero negativo, es

decir, validación semántica.

1. Crear el programa
   1. Asegúrese de contar con una carpeta de trabajo en C:\codigo. Si aún no ha creado la carpeta C:\codigo hágalo antes de continuar.
   2. Seleccione Archivo > Abrir carpeta en el menú principal.
   3. En el cuadro de diálogo Abrir carpeta, cree una carpeta en la ruta *C:\codigo\ps\cwe-191* y selecciónela. Luego haga clic en Seleccionar carpeta.
   4. El nombre de la carpeta se convierte en el nombre del proyecto y el nombre del espacio de nombres de forma predeterminada. Agregará código más adelante en el tutorial que asume que el espacio de nombres del proyecto es *cwe-191*.
   5. Vamos a simular una página de compra de un producto. Cree un nuevo archivo llamado index.php. Escriba el siguiente código de PHP al inicio del archivo.

<?php session\_start();

$precio = 5000;

if (!isset($\_SESSION['total'])) {

$total = 0;

$cantidad = 0;

} else {

$total = $\_SESSION['total'];

$cantidad = $\_SESSION['cantidad'];

}

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if (is\_numeric($\_POST['cantidad'])) { // Valida que sea numero if (!isset($\_POST['btnQuitar'])) {

$total = $total + ($precio \* $\_POST['cantidad']);

$cantidad = $cantidad + abs($\_POST['cantidad']);

} else {

$total = 0;

$cantidad = 0;

}

}

$\_SESSION['total'] = $total;

$\_SESSION['cantidad'] = $cantidad;

}

?>

* 1. Observe el inicio del código. Utilizar sesiones en el servidor para almacenar valores que deseamos conservar entre peticiones. Esto se le llama manejo de estado de la sesión.
  2. Debajo de este código, agregue el código de HTML.

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="icon" href="data:;base64,iVBORw0KGgo=" />

<title>Mercado Libre - Programación segura</title>

<link href=["htt](https://cdn.jsdelivr.net/npm/bootstrap%405.3.2/dist/css/bootstrap.min.css)p[s://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"](https://cdn.jsdelivr.net/npm/bootstrap%405.3.2/dist/css/bootstrap.min.css) rel="stylesheet">

<link rel="stylesheet" href="estilos.css">

</head>

<body>

<nav class="navbar bg-mercadolibre">

<div class="container-fluid">

<a class="navbar-brand" href="#">

<img src="images/logoml.png" alt="Logo" class="img-fluid logo">

</a>

</div>

</nav>

<div class="container mt-3">

<h2>Tienda en línea <small class="text-muted fs-5">Programación segura</small></h2>

<div class="mt-4">

<div class="row">

<div class="col">

<img src="images/cel.jpg" alt="Celular" class="img-fluid mt-4">

</div>

<div class="col">

<div class="card">

<div class="card-header"> Oferta

</div>

<div class="card-body">

<h5 class="card-title">Samsung Galaxy S24 Plus 5G Dual SIM 512 GB negro 12 GB RAM</h5>

<p class="fs-4">$<?php echo number\_format($precio, 2); ?></p>

<ul class="list-unstyled">

<li>Dispositivo desbloqueado para que elijas la compañía telefónica que prefieras.</li>

<li>Compatible con redes 5G.</li>

<li>Pantalla Dynamic AMOLED 2X de 6.7".</li>

<li>Tiene 3 cámaras traseras de 50Mpx/12Mpx/10Mpx.</li>

<li>This may still come in handy in some situations.</li>

</ul>

<form method="post" action="index.php">

<div class="mb-3">

<label for="cantidad" class="form-label">Cantidad:</label>

<select id="cantidad" name="cantidad" class="form-select form-select-sm" aria-label="Small select example">

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

</select>

</div>

<button type="submit" class="btn btn-primary">Agregar al total</button>

<button name="btnQuitar" type="submit" class="btn btn-secondary">Quitar todo</button>

</form>

</div>

<div class="card-footer text-body-secondary fw-bold text-end">

<div>Cantidad: <span><?php echo number\_format($cantidad); ?></div>

<div>Total a pagar: <span>$<?php echo number\_format($total, 2); ?></span></div>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

* 1. Ahora vamos a agregar la hoja de estilos que usa este archivo. Cree un nuevo archivo llamado

estilos.css. Escriba el siguiente código.

.bg-mercadolibre {

background-color: #ffe600;

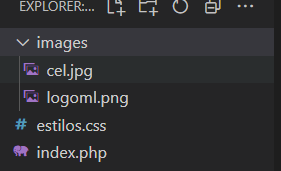
}

.logo {

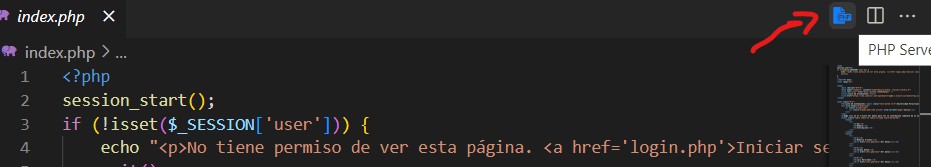
height: 30px;

}

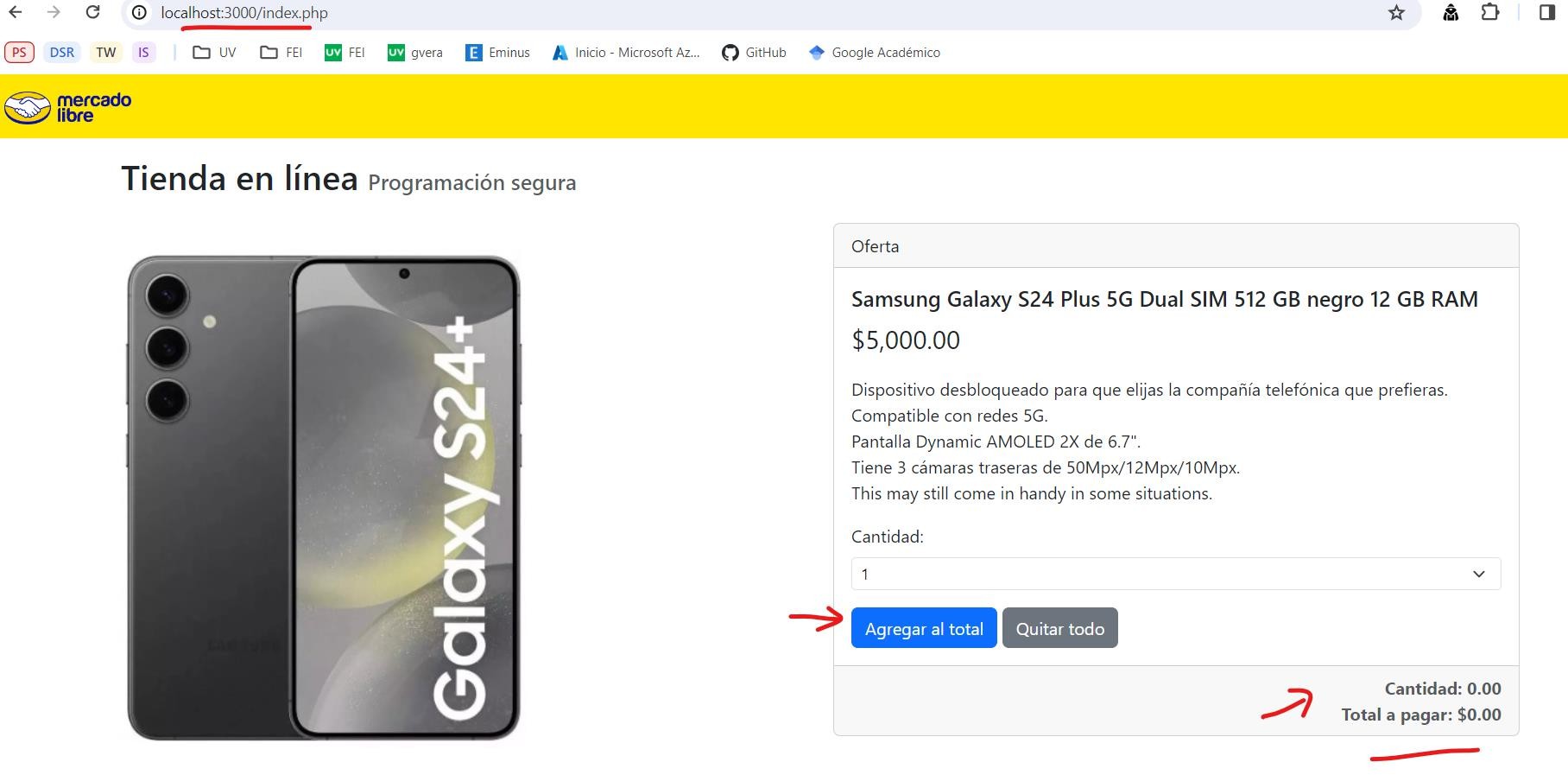
* 1. Cree una nueva carpeta llamada images en su proyecto.
  2. Descargue las siguientes dos imágenes de la Internet y colóquelas en esta nueva carpeta.
     + <https://github.com/gverafei/images/blob/main/logoml.png>
     + <https://github.com/gverafei/images/blob/main/cel.jpg>
  3. Debería contar con los siguientes archivos en su proyecto.



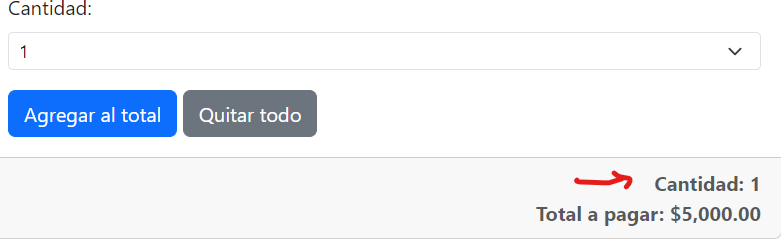
* 1. Para probar el funcionamiento de su aplicación, debe tener instalado la extensión PHP Server de Visual Studio Code. Abra su archivo index.php. En la parte superior derecha de su ventana de código, presione el icono de PHP Server. También puede ejecutar el comando php -S localhost:3000.



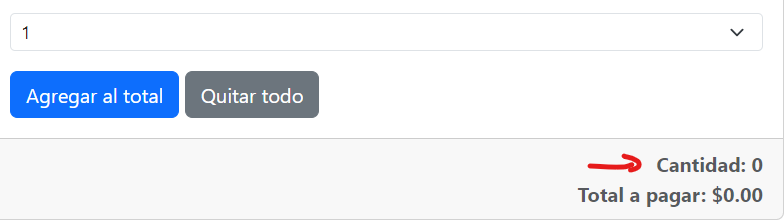
* 1. Se abrirá su navegador predeterminado en la página index.php.



* 1. Seleccione una cantidad de producto y presione el botón Agregar al total. Observe que la cantidad del producto aumenta al igual que el precio.

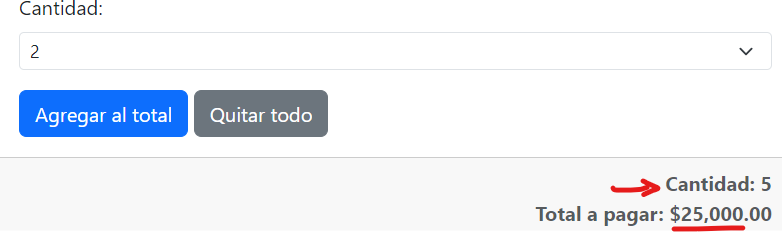


* 1. Presione el botón Quitar todo y observe que los valores vuelven a cero.



* 1. Seleccione varias cantidades y agréguelos al total. Observe como los valores solo pueden

incrementar. Al usar un campo de selección, no es posible para el usuario agregar otro tipo de números de cantidades.



* 1. Esta es la estructura más común de una aplicación web simple. Incluso hay varios sistemas actualmente en producción tanto dentro como fuera de la institución construidos de esta manera.
  2. Felicidades, ha creado su aplicación de manera correcta.

1. Explotar la vulnerabilidad

Lo primero que hace un atacante es revisar si hay puntos vulnerables en la validación de entradas. Pueden ser tanto campos de texto, peticiones url en el navegador, como llamadas a procedimientos remotos.

En este ejemplo vamos a encontrar el punto de entrada de la vulnerabilidad sobre el valor de cantidad de unidades a comprar. Es un número entero, por lo que un atacante intentará modificarlo por un número

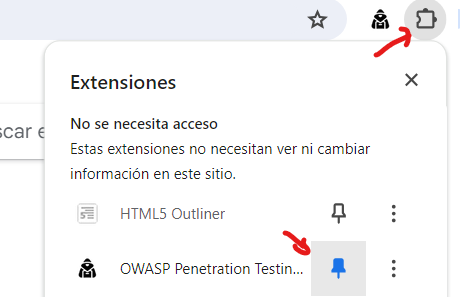
negativo, para identificar si puede comprar un producto a menor precio del inicial.

1. Abra su navegador Chrome e instale la extensión desde

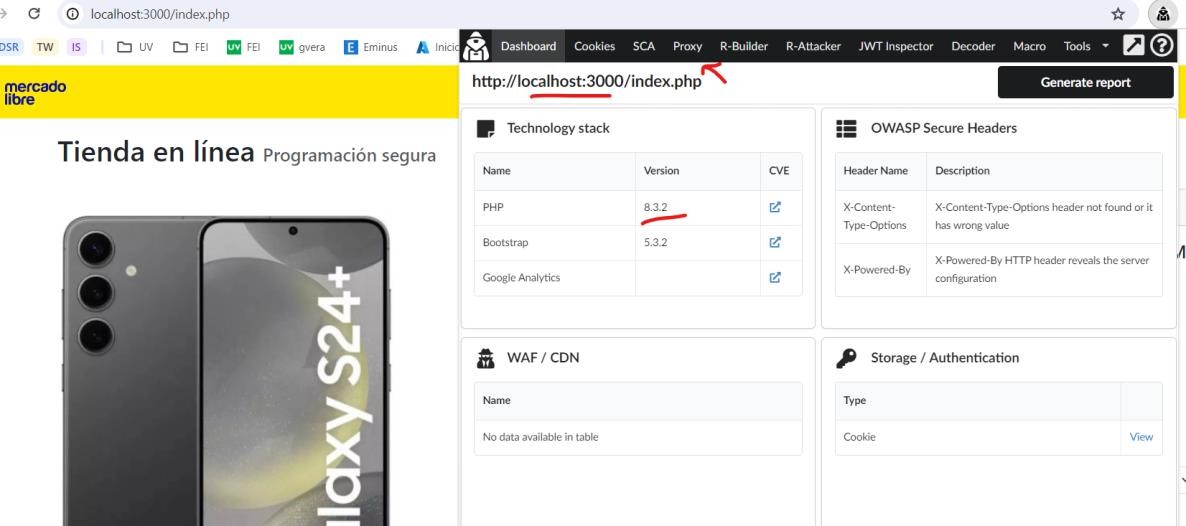
[https://chromewebstore.google.com/detail/ojkchikaholjmcnefhjlbohackpeeknd.](https://chromewebstore.google.com/detail/ojkchikaholjmcnefhjlbohackpeeknd)



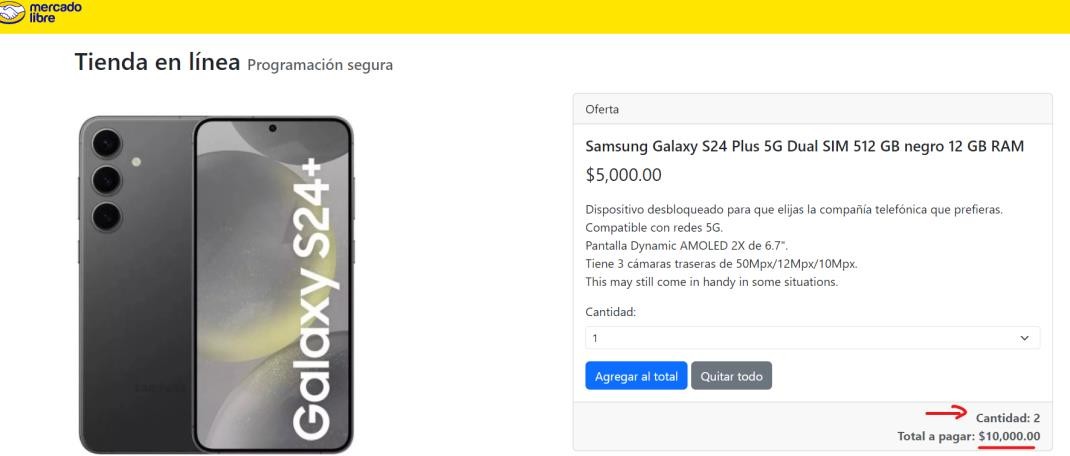
1. Asegúrese de tenerlo activado haciendo clic en el pin de la barra de Chrome.



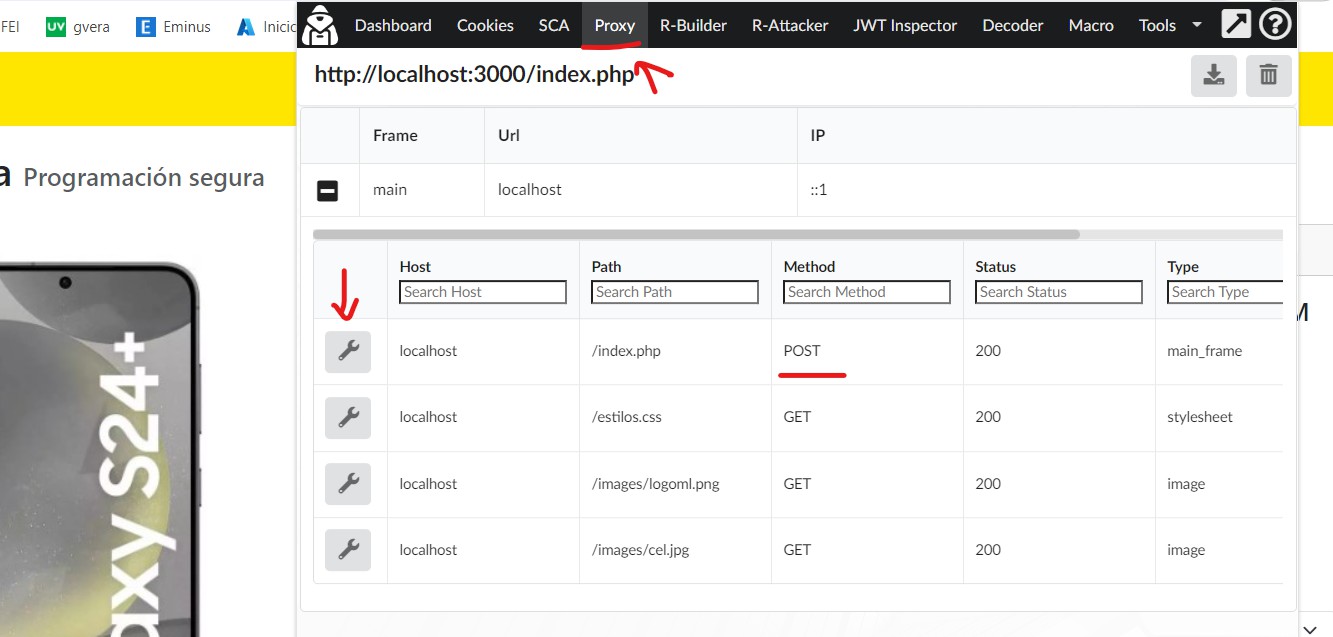
1. Vuelva a abrir la página index.php en su navegador web.
2. Haga clic en la nueva extensión OWASP y verifique que está recopilando datos. Si es necesario recargue la página para que se intercepten los datos.



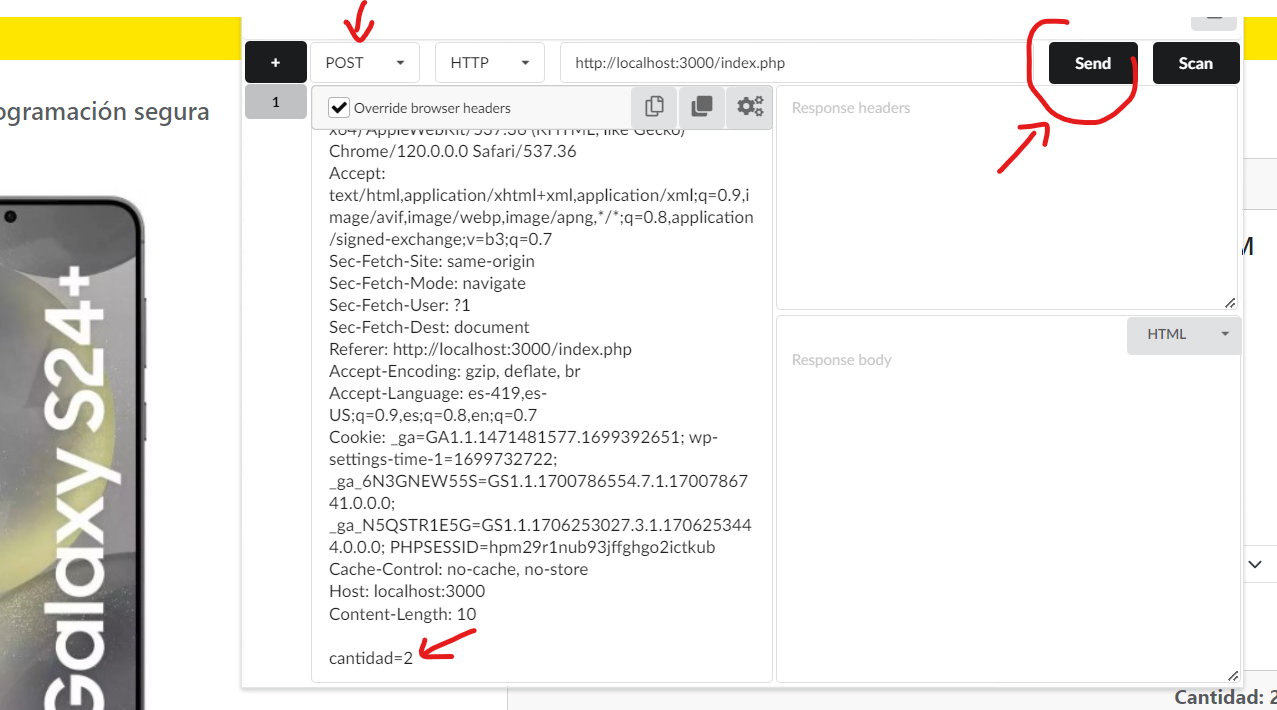
1. Ingrese 2 unidades de productos, de manera que la cantidad sea 2 y el total $10,000 pesos.



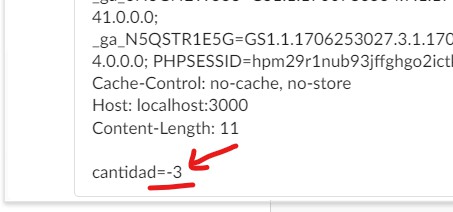
1. En la extensión OWASP, vaya a la pestaña de Proxy. Busque la petición con el método POST.



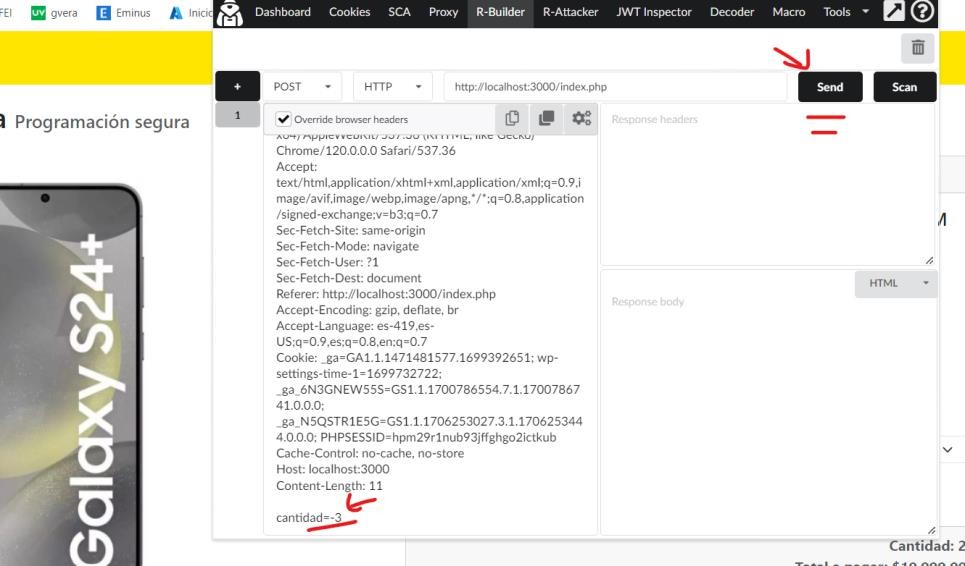
1. Haga clic en el icono de la llave de herramienta. Observe las secciones que aparecen.



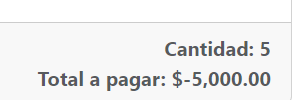
1. Del lado izquierdo aparece la petición realizada por el método POST. Del lado izquierdo aparecen la respuesta, dividida en encabezados y en cuerpo.
2. Observe el valor cantidad=2 de la última petición POST realizada. Ese valor puede variar del 1 al 3, dependiendo como haya efectuado su ejemplo.
3. Modifique el valor de cantidad y coloque el número negativo -3. Lo que estamos buscando es sumar 3 unidades del producto, pero que el precio sea $-5,000 pesos.



1. Ya que está modificado el nuevo valor presione el botó de Send en la parte superior derecha.



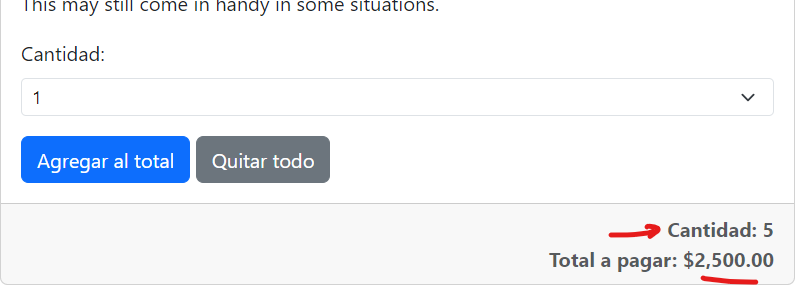
1. Observe como la petición es enviada dentro de OWASP. Ahora, recargue la página de Chrome.



1. Observe que hemos tenido éxito!
2. Puede continuar probando con distintos valores dentro de OWASP. Incluso puede probar con cadenas de caracteres en lugar de números. El programa válida que sean números válidos. La vulnerabilidad

se produce porque no válida que sean números positivos.

1. Lo que un atacante haría es modificar los valores de manera que los encargados no se den cuenta de este problema. Es probable que agreguen valores de cantidad negativos o con puntos decimales para crear un monto a pagar muy bajo, pero pidiendo varios artículos.



1. Por ejemplo, puedo ordenar 5 productos y pagar solo $2,500 pesos.
2. Felicidades, ha creado su aplicación de manera correcta.