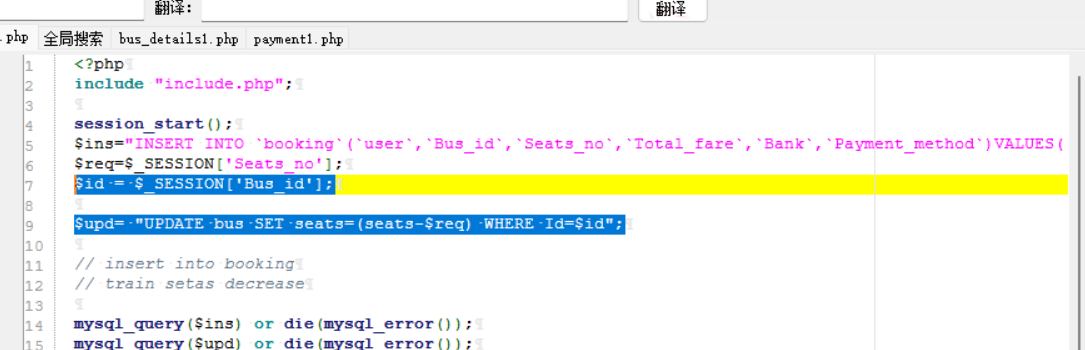
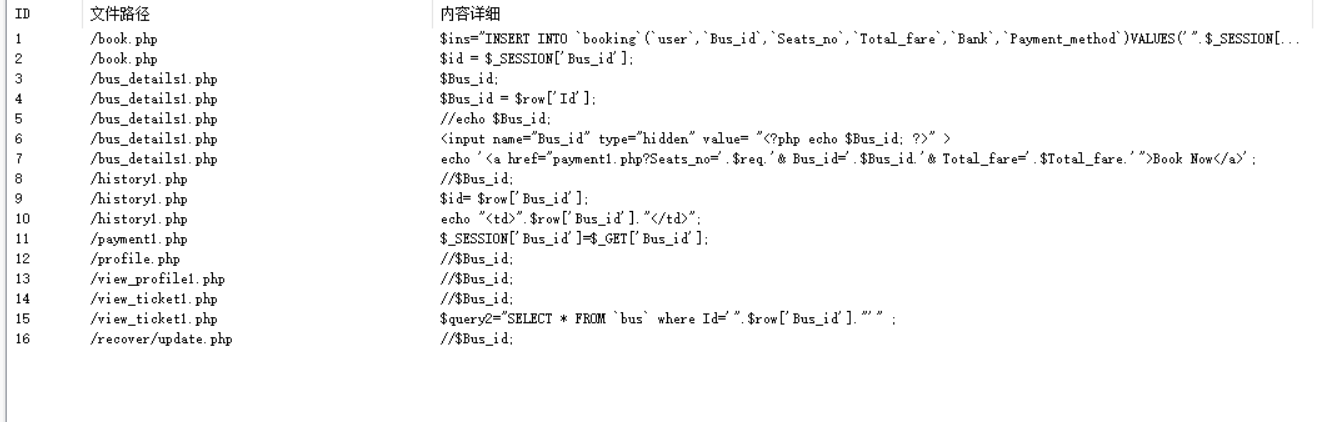
## Online Bus Reservation System Site

The audit of the open source code found that SQL injection existed, and the Bus\_id parameters were not filtered.



Therefore, the Bus\_id was tracked and discovered

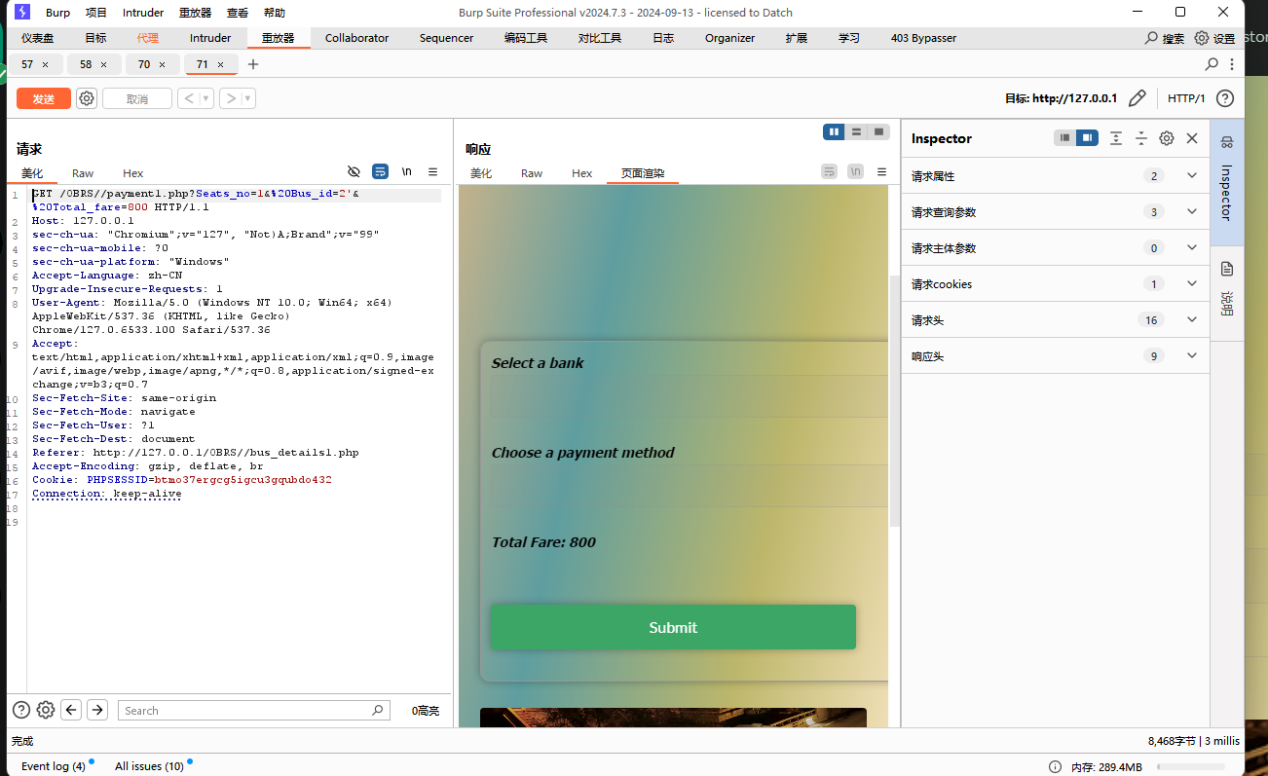


Therefore, the request bus\_details1.php is captured, and the following packets are found to have Bus\_id variables

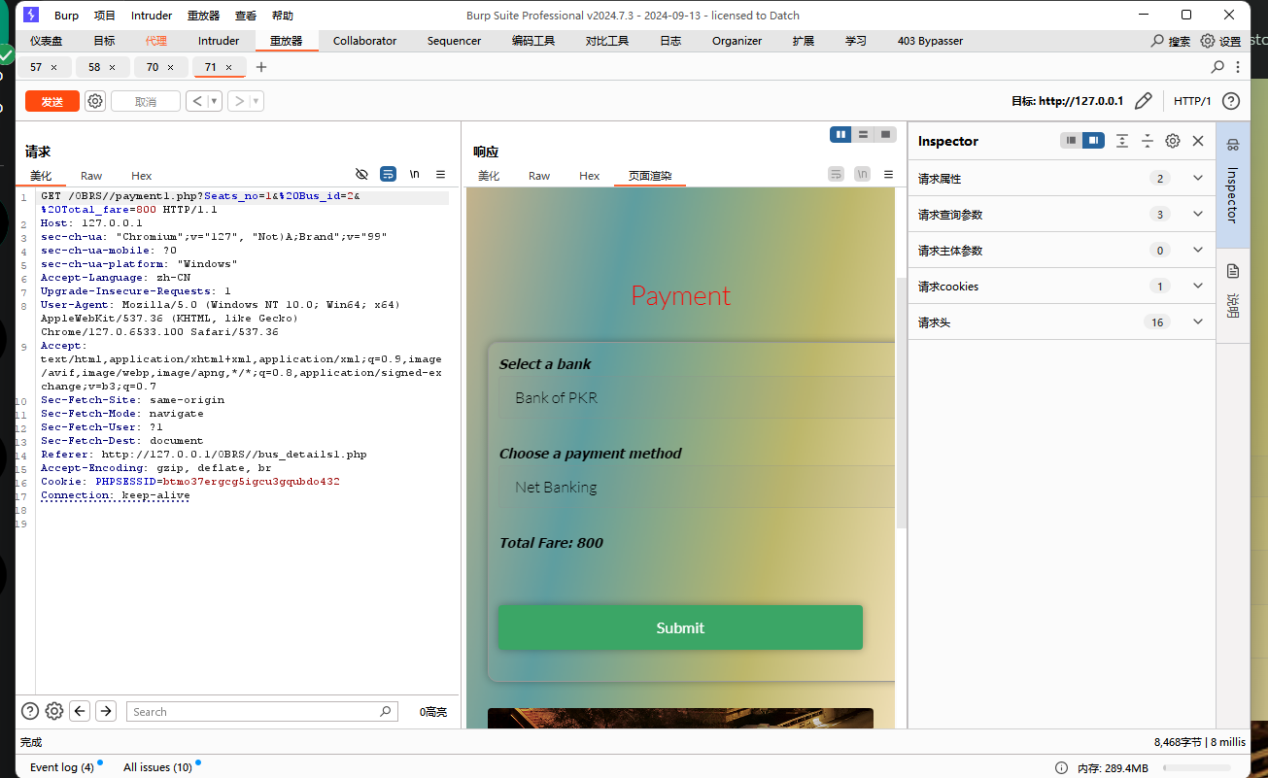


So it can be injected here.

So modify the request header /OBRS//payment1.php? Seats\_no=1&%20Bus\_id=2'&%20Total\_fare=800 found that the web page is inaccessible, so there is an injection at this point.



Normal web page access is as follows:



Modify the code:

<?php

include "include.php";

session\_start();

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

$bus\_id = $\_SESSION['Bus\_id'];

$seats\_no = $\_SESSION['Seats\_no'];

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

$bank = $\_POST['Bank'];

$payment\_method = $\_POST['Payment\_method'];

$conn = new mysqli($host, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$ins = $conn->prepare("INSERT INTO booking (user, Bus\_id, Seats\_no, Total\_fare, Bank, Payment\_method) VALUES (?, ?, ?, ?, ?, ?)");

$ins->bind\_param("ssssss", $user, $bus\_id, $seats\_no, $total\_fare, $bank, $payment\_method);

if (!$ins->execute()) {

die("Error executing insert query: " . $ins->error);

}

$upd = $conn->prepare("UPDATE bus SET seats = seats - ? WHERE Id = ?");

$upd->bind\_param("ii", $seats\_no, $bus\_id);

if (!$upd->execute()) {

die("Error executing update query: " . $upd->error);

}

$ins->close();

$upd->close();

$conn->close();

$\_SESSION['pay'] = 'Payment successful...';

header('location: generic.php');

exit();

?>