561 Final Project

Description:

The PHP Type Juggling Vulnerability CTF challenge is designed to test the understanding of a common vulnerability in PHP applications. Type juggling is a technique used by attackers to manipulate PHP variables by converting them into different types or values. This can lead to unexpected behaviour, which can be exploited to gain unauthorized access to sensitive information or execute malicious code.

In this challenge, a PHP code snippet that contains a vulnerability that allows an attacker to bypass authentication and gain access to restricted parts of the application will be displayed initially. The code may use weak comparison operators or rely on implicit type conversions to compare user input with hardcoded values.

To solve the challenge, the vulnerability needs to be identified and demonstrated how it can be exploited to bypass authentication. This may involve manipulating input data to coerce the comparison to evaluate to true or using a specific value to trigger a specific behaviour.

Solution

In this challenge, the index.php code requires a match between the hash and target values. Initially, the code checks that the hash parameter is not equal to aaK1STfY. Then, it checks that the sha1 hash of the hash parameter is equal to that of aaK1STfY. Since sha1 hashes are unique, brute-forcing is not feasible. However, the code compares the $hash and $target variables using == instead of ===, making it vulnerable to type juggling attacks. The hash of aaK1STfY starts with 0e..., so any other hash that starts with 0e will match this with == since == does not check types and treats these as numbers. Therefore, the number of values that need to be brute-forced is considerably reduced. Several values match this condition, such as aaroZmOk and aaO8zKZF. Passing any of these values as the hash parameter with a GET request will retrieve the flag.

For example: /?hash=aaO8zKZF.