Black Hat USA - 2007

August 2nd

Jeff Morin

Sr. Security Consultant

Type Conversion Errors

SpiderLabs

A division of AmbironTrustWave

AmbironTrustWave

Setting the Stage

Certain data types are able to take in multiple inputs of varying construction, which ultimately evaluate to the same value, while others are not:

Integer

- 01 == 1
- 00000000002 == 2

String

- 01 <> 1
- 1.00000 <> 1

Float

- *1.00000 == 1*
- *000000.2 == 0.2*
- *00000.30000 == 0.3*

Vulnerability Explanation

The Problem:

- Data assigned to a variable of type ***int*** or ***float***, when compared to another variable of the same data type, will evaluate correctly in a conditional statement

- In contrast, the same data used in a conditional statement utilizing a ***string*** data type for comparison may evaluate incorrectly or ultimately fail when compared to the original ***int*** or ***float*** variable.

**** If the string conditional statement is being used in a security context, then this could result in a significant risk to the data and application ****

Real World Example

Real World Example - Background

Application was a web based ecommerce site which utilized a shopping cart to browse the site and select items for purchase.

Ordering

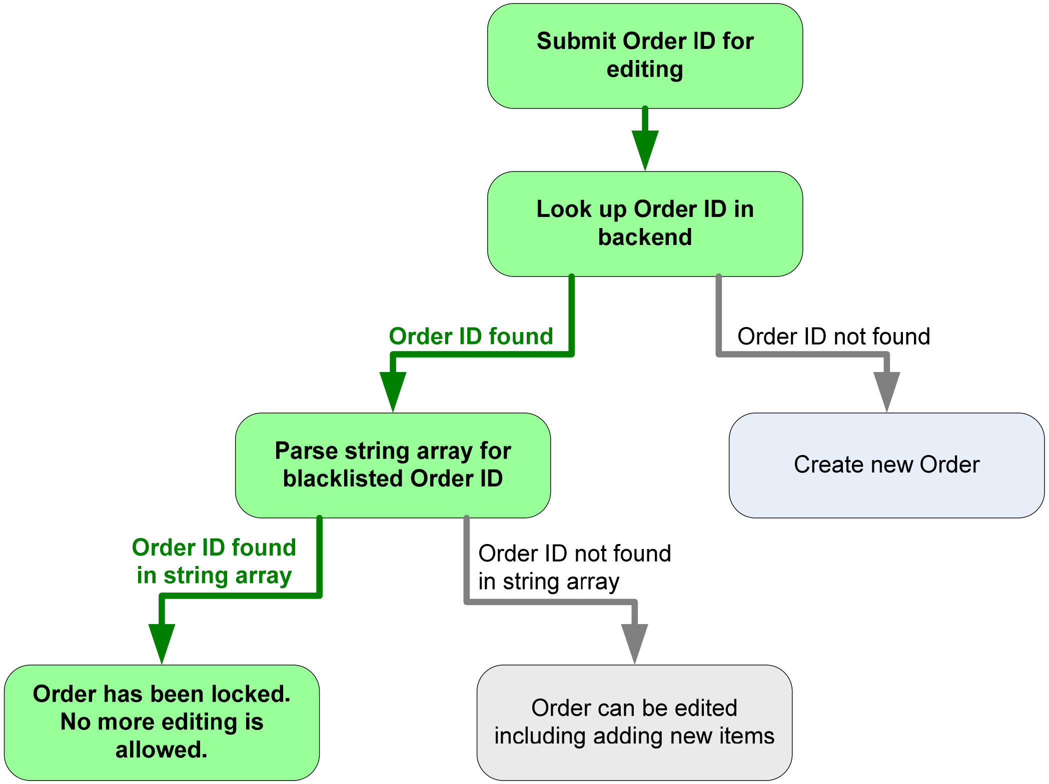
- Shopping cart could be edited at any time prior to actually purchasing the items and was tracked with an order ID number.
- A user could stop shopping and come back later and continue where they left off.

Purchasing

- Once the items had been purchased, the order was locked in and the user could not modify the contents, as that would have resulted in the user getting items for free.
- This was accomplished by taking the Order ID and placing it into a string array of blacklisted IDs.
- If a user went back to check on the order items, the Order ID was first referenced in the database for the order items, and then the array was parsed to determine if the ID was blacklisted, and thus not allowed to be edited.

Real World Example

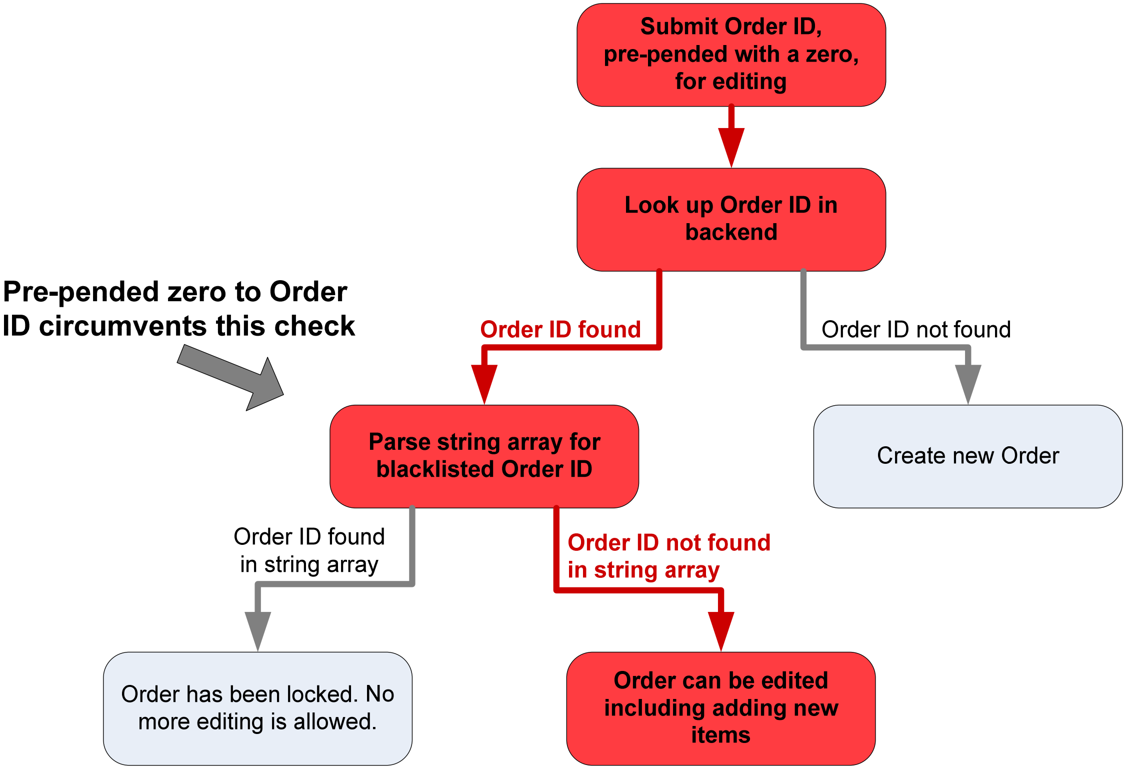
Process flow for attempting to edit a locked order:



Submit Order ID for editing
Look up Order ID in backend
Order ID found
Order ID not found
Parse string array for blacklisted Order ID
Create new Order
Order ID found in string array
Order ID not found in string array
Order has been locked. No more editing is allowed.
Order can be edited including adding new items

Real World Example

Process flow for attempting to edit a locked order with a ***modified*** Order ID:


Submit Order ID, pre-pended with a zero, for editing
Look up Order ID in backend
Order ID found
Order ID not found
Pre-pended zero to Order ID circumvents this check
Parse string array for blacklisted Order ID
Create new Order
Order ID found in string array
Order ID not found in string array
Order has been locked. No more editing is allowed.
Order can be edited including adding new items

Conclusions

- Always be consistent in the data types that you use throughout the application, especially in security related conditional statements

- When testing applications for security vulnerabilities, ensure that you test for type conversion errors by pre-pending and appending zeros to input variables

- Be alert for resultant modified application behavior in other areas of the site

Questions

Questions?