Preventing SQL Injection

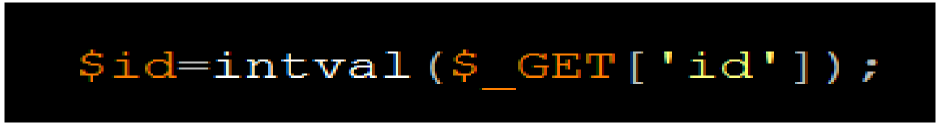
CYBER SECURITY & PRIVACY FOUNDATION

Methods

- Input Validation
- Prepared Statements
- Least Privilege

Input Validation

- Validate and Sanitize all user Input
- Check if the given input has expected data type. For example, is_numeric( )
- If the expected input is numeric, then use 'intval' function to convert it into number format



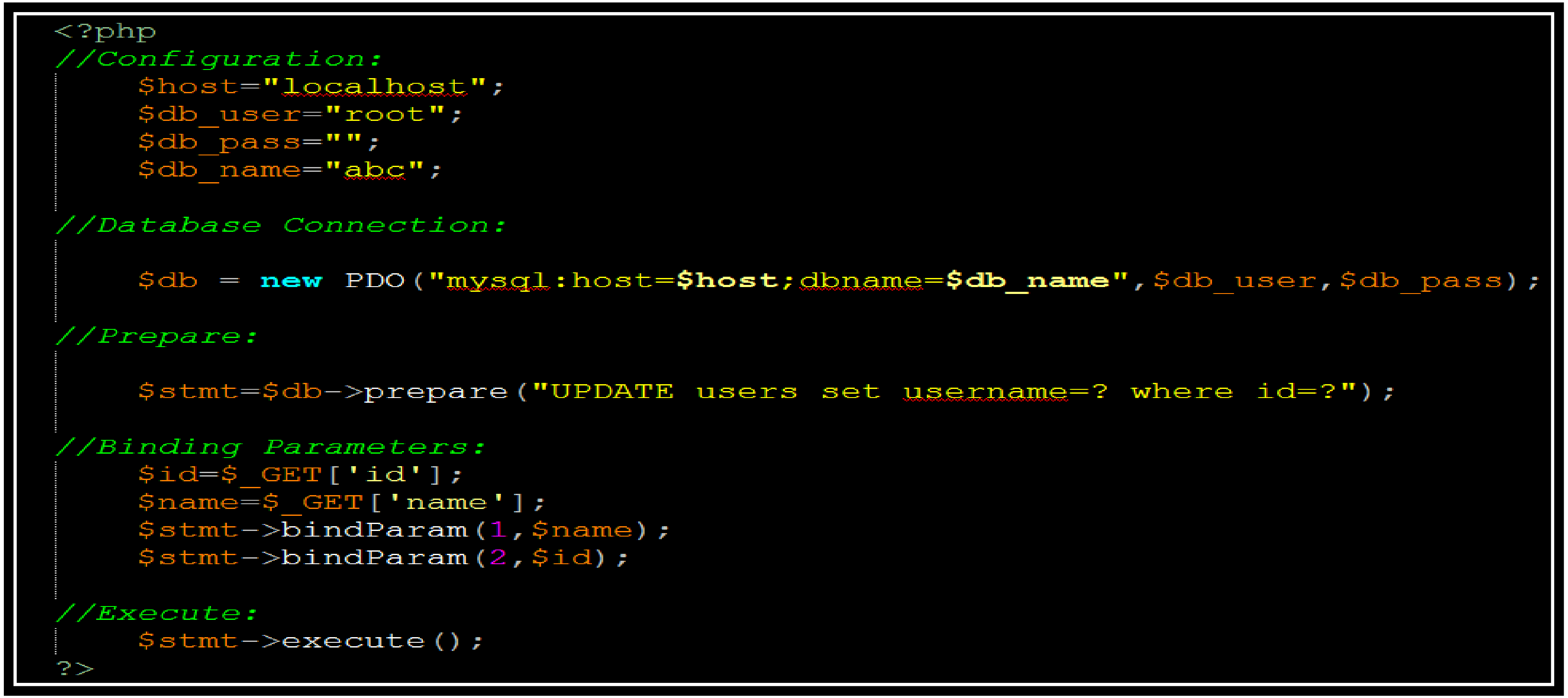
```
$id=intval($_GET['id']);
```

- In the database layer doesn't support parameterized input, then you can use **mysql_real_escape_string( )** function to escape Special characters. However, it is not recommended to use this deprecated function.

Prepared Statements

- The most recommended Defense mechanism against SQL Injection attacks.
- Also known as parameterized statement, makes your queries run faster and securely.
- Parsed once and can be executed multiple times.
- **Separates SQL logic and the Data:**
  - **Prepare**: At the Prepare stage, the DB Server parses the given query and allocates space for the parameters (labeled “ ? “).
  - **Bind Parameters:** Once it is prepared, you can pass the data to the parameters. Whatever is being passed in the parameters will be considered as Data only. This prevents SQL Injection.

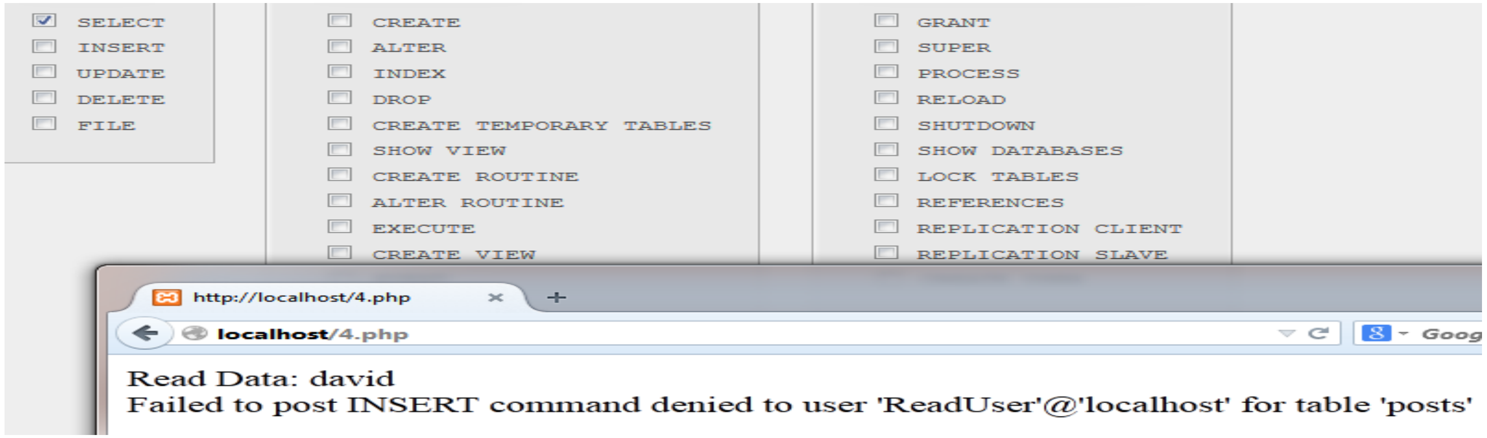
Example usage of Prepared Statement with PDO



```php
<?php
//Configuration:
    $host="localhost";
    $db_user="root";
    $db_pass="";
    $db_name="abc";

//Database Connection:

    $db = new PDO("mysql:host=$host;dbname=$db_name",$db_user,$db_pass);

//Prepare:

    $stmt=$db->prepare("UPDATE users set username=? where id=?");

//Binding Parameters:
    $id=$_GET['id'];
    $name=$_GET['name'];
    $stmt->bindParam(1,$name);
    $stmt->bindParam(2,$id);

//Execute:
    $stmt->execute();
?>
```

Least Privilege

- Minimize the privilege assigned to every Database accounts
- If an account is used only for reading the data, then just assign only read permission
- Never give root access to database accounts



✓ SELECT	☐ CREATE	☐ GRANT
☐ INSERT	☐ ALTER	☐ SUPER
☐ UPDATE	☐ INDEX	☐ PROCESS
☐ DELETE	☐ DROP	☐ RELOAD
☐ FILE	☐ CREATE TEMPORARY TABLES	☐ SHUTDOWN
	☐ SHOW VIEW	☐ SHOW DATABASES
	☐ CREATE ROUTINE	☐ LOCK TABLES
	☐ ALTER ROUTINE	☐ REFERENCES
	☐ EXECUTE	☐ REPLICATION CLIENT
	☐ CREATE VIEW	☐ REPLICATION SLAVE

```
Read Data: david
Failed to post INSERT command denied to user 'ReadUser'@'localhost' for table 'posts'
```