

PHP Lecture 3

- isset()
- Rudimentary server management && protection against black hats
- Connecting to a database
- Submitting and receiving information from a database
- Website management via server-side processing

isset(): the essential function

Very often a developer will be uncertain if a particular variable, vital to a process, actually exists.

For instance, imagine if someone uses a URL that connects to the backend of a form (the page that processes the forms results) without using the form.

The form backend will expect to obtain information from `$_POST` that doesn't exist.

`isset()` returns true if the argument it is given exists.
even if it contains nothing!

`empty()` is similar but returns true if the variable exists but contains NULL

<h2>Thank you

<?php

echo (\$_POST["name"]);

?>

for applying for the position</h2>

Thank you

**Notice: Undefined index: name in fubar.php on line 14
for applying for the position**

<?php if (isset(\$_POST["name"]))

echo ('<h2>Thank you! \$_POST["name"].for applying
for the position</h2>'); ?>

Architecture of a server

Being able to upload a website, and in particular, the use of server side processing and databases, necessitates being able to understand some of the inner mechanics of both the web server, and the OS the web server is on.

Everything, from choice of OS, to specific scripting language is important; but more than anything else, a good server admin is crucial!

Growing popularity of VPS (Virtual Private Server) as a platform

Many VPSs on server, many potential users on VPS

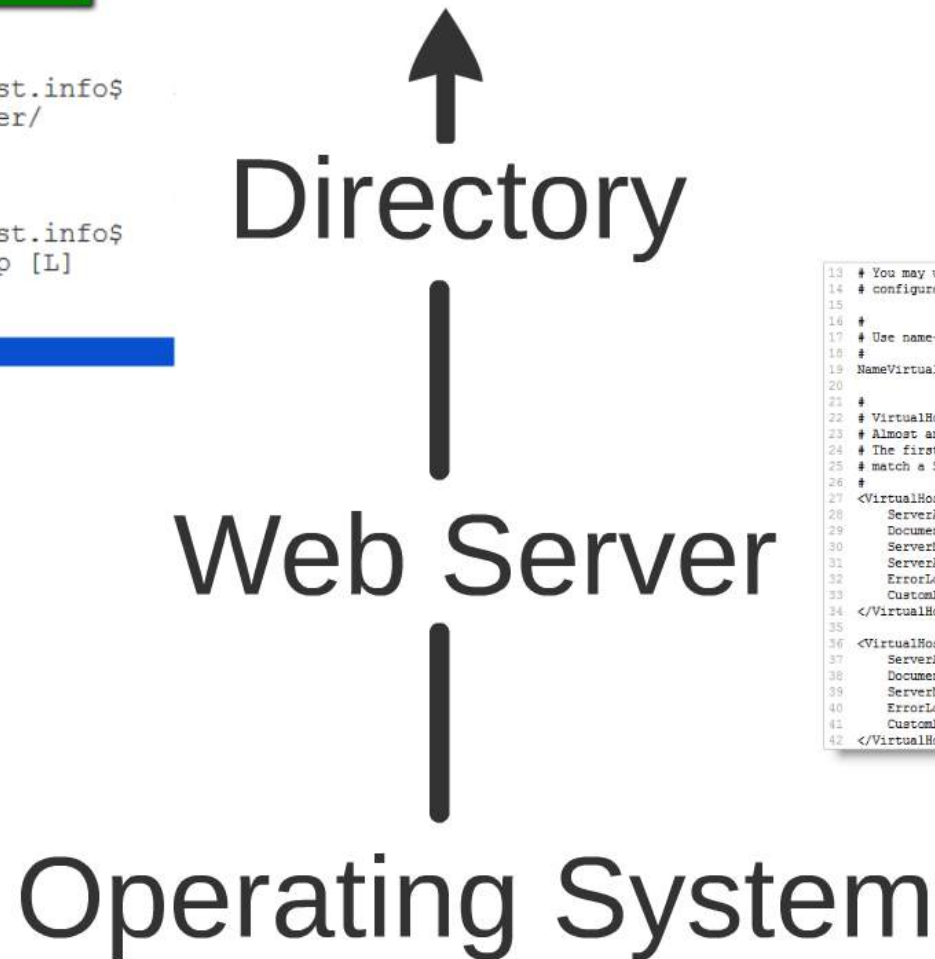
Permissions

Many web and application servers rely on access control lists provided by the file system of the underlying platform. Even if almost all data is stored on backend servers, there are always files stored on the web and application server that should not be publicly accessible, particularly configuration files, default files, and scripts that are installed on most web and application servers. Only files specifically intended to be presented to web users should be marked as readable using the OS's permissions mechanism, most directories should not be readable, and very few files, if any, should be marked executable.

```
#RewriteEngine On
#RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
#RewriteRule ^(.*)$ http://%1/$1 [R=301,L]
```

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^(www.)?nctest.info$
RewriteCond %{REQUEST_URI} !~/subfolder/
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ /subfolder/$1
RewriteCond %{HTTP_HOST} ^(www.)?nctest.info$
RewriteRule ^(/)?$_subfolder/index.php [L]
```

	Filesize	File
		wp-content
		wp-includes
		.htaccess
		favicon.gif
		favicon.ico
		index.php
		license.txt
.00.13...	12,292	File
.46.12...	1,042,681	png
.53.17...	422,228	png
	740,635	png



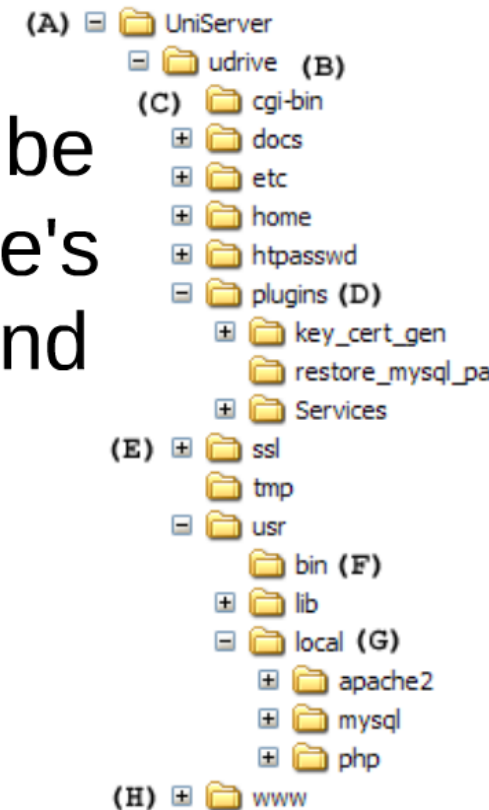
```
13 # You may use the command line option '-S' to verify your virtual host
14 # configuration.
15
16 #
17 # Use name-based virtual hosting.
18 #
19 NameVirtualHost *:80
20
21 #
22 # VirtualHost example:
23 # Almost any Apache directive may go into a VirtualHost container.
24 # The first VirtualHost section is used for all requests that do not
25 # match a ServerName or ServerAlias in any <VirtualHost> block.
26 #
27 <VirtualHost *:80>
28     ServerAdmin webmaster@dummy-host.home
29     DocumentRoot /www/docs/dummy-host.home
30     ServerName dummy-host.home
31     ServerAlias www.dummy-host.home
32     ErrorLog logs/dummy-host.home-error_log
33     CustomLog logs/dummy-host.home-access_log common
34 </VirtualHost>
35
36 <VirtualHost *:80>
37     ServerAdmin webmaster@dummy-host2.home
38     DocumentRoot /www/docs/dummy-host2.home
39     ServerName dummy-host2.home
40     ErrorLog logs/dummy-host2.home-error_log
41     CustomLog logs/dummy-host2.home-access_log common
42 </VirtualHost>
```

Architecture of a server

Depending on your setup, many of these security issues will be handled by your sysadmin.

Nevertheless, your website itself can still be compromised, and compromising a website's security can potentially compromise backend systems (like databases).

The main way that websites and servers are compromised is by their providing information to malicious agents which may provide a means of exploitation.



Exploitations

Information is often obtained by using server-side scripting (e.g. PHP) to give diagnostic information about the system and software.

file system, default messages, etc. are defined in config files

phpinfo, error messages, etc. can be run using PHP

Considerations re: SSP

At the front-end, users should be treated as nice as possible; at the backend users should be treated as hostile entities. Even if they are just normal users!

Even non-malicious users can be a pain, as they are guaranteed to enter junk into databases if allowed to.

Always consider bots. Help friendly bots, hinder unfriendly ones, and prevent friendly/neutral bots from inadvertently doing damage.

One of the reasons that only HTTP POST should be used on actions that may change backend information.

Connecting to the DB

In order to gain information from, or send information to a database, a connection will have to be established between the application and the db.

RDMSs e.g. MySQL, have users with associated privileges. These "users" are specific to the db, and have no innate overlap with any other type of user.

The two ways to connect to databases with PHP are either using `mysqli` or **PDO**. We'll be using **mysqli**.

Don't ever use the **mysql()** function. It is unsafe; was deprecated in PHP5 and removed in PHP7.

Edit Privileges: User 'duncan2'@'hope1.ucd.ie'

Global privileges

Check All

Note: MySQL privilege names are expressed in English.

Data

☒ SELECT

☒ INSERT

☒ UPDATE

☒ DELETE

☒ FILE

Structure

☐ CREATE

☐ ALTER

☐ INDEX

☐ DROP

☐ CREATE TEMPORARY TABLES

☐ SHOW VIEW

☐ CREATE ROUTINE

☐ ALTER ROUTINE

☐ EXECUTE

☐ CREATE VIEW

☐ EVENT

☐ TRIGGER

Administration

☐ GRANT

☐ SUPER

☐ PROCESS

☐ RELOAD

☐ SHUTDOWN

☐ SHOW DATABASES

☐ LOCK TABLES

☐ REFERENCES

☐ REPLICATION CLIENT

☐ REPLICATION SLAVE

☐ CREATE USER

Resource limits

Note: Setting these options to 0 (zero) removes the limit.

MAX QUERIES PER HOUR

0

MAX UPDATES PER HOUR

0

MAX CONNECTIONS PER HOUR

0

MAX USER_CONNECTIONS

0

Require SSL

☐ SPECIFIED

REQUIRE CIPHER

REQUIRE ISSUER

REQUIRE SUBJECT

☐ REQUIRE X509

☒ REQUIRE SSL

```
$conn = mysqli_connect("servername", "username", "password", "db");  
// Check connection  
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}
```

Domain name
IP address

database user

password

name of db

Databases can be either local or remote.

Databases often only accept local connections.

The database connection (`$conn`) is reused in queries. It should be nullified when no longer in use.

Information from the DB

```
$query = "SELECT `id`, `first_name`, `last_name` FROM `employee`";
```

```
$result = mysqli_query($conn, $query);
```

```
while ($row = mysqli_fetch_array($result) )
```

```
{
```

```
    echo ($row[0] . ':' . $row[1] . ' ' . $row[2]);
```

```
    echo ('<br />');
```

```
}
```

SQL

\$result can sometimes be very messy.
var_dump(\$result) to check contents

SQL injection

PHP doesn't have any understanding of what SQL is, in much the same way as it doesn't understand HTML

Like HTML, SQL statements can be built up in simple strings in PHP (as seen in the previous slide)

However, this simple approach can be quite naive when dealing with user supplied field, for security reasons.

Traditional method is for attacker to submit data that looks legitimate to PHP, but would appear as SQL to a database


Frontend

`= `0` OR 1=1` email

Backend

```
$email = $_POST["email"];  
$pw = $_POST["pw"];
```

```
$sql = "SELECT email FROM Users  
WHERE email = '" . $email . "'";  
$query = $conn->query($sql);
```



Database

```
"SELECT email FROM Users WHERE email = `0` OR 1=1;"
```

Building a library

While markup, client-side scripting, etc. can be populated throughout websites using `include()` (or, to be more precise, printed in output from a web server), PHP functions can also be "included" in a similar manner

This allows for the reuse of functions (or classes) throughout a web app.

```

1 <?php
2 // ERROR REPORTING FUNCTION
3 function failure($error)
4 {
5     $tempCount = 0;
6     global $dbh;
7
8     echo "<h1>Sorry! Something went wrong!</h1>";
9     //check if multidimensional
10    if (isset($error[0][1])) {
11        echo ('<ul>');
12
13        //print out everything in multidimensional array
14        foreach ($error as $result) {
15            $result = str_replace('_', ' ', $result); //remove underscores
16            echo ('<n <li>' . $result . '</li>');
17            $tempCount++;
18        }
19        if ($tempCount > 1) {
20            echo ('</ul><span class="offset-by-two"> are missing </span><br /><br />');
21        } else {
22            echo ('</ul> is missing <br /><br />');
23        }
24    } else {
25        echo ('<h3>' . $error . '</h3>');
26    }
27
28    echo ('<p class="sixteen columns">Please go back and fix these errors.</p>
29    <p class="sixteen columns">If you have fixed any possible errors, and you believe that there is a problem w
30    please contact me at duncan.wallace[at]ucdconnect.ie detailing the nature of the fault encountered <br /> <s
31    the email address)</small></p>
32    </body></html>');
33    if (!empty($dbh))
34        $dbh = null; //CLOSING CONNECTION (IF IT EXISTS)
35    die();
36 }

```


Developing a Framework

In terms of website maintenance one can distinguish between fixed and variable content

fixed

- **boilerplate**
- **imports**
- **meta information**
- **navigation**
- **page structure**
- **sessions/cookies/globals**
- **database connection, etc.**

index.php



```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="UTF-8" />
    <title>A title</title>
    <style>
      body {font-family:Arial;}
    </style>
  </head>
  <body>
    <h1 class="big">
      Homepage</h1>
  </body>
</html>
```

variable

- **imports**
- **text**
- **page level CSS**
- **specific page elements & content**

Content Management System

Content Management Systems (CMS) provide a way for a user/administrator to make changes to a website through their browser.

While traditionally this is considered in terms of developers providing clients who don't have any development skills the means to alter their websites, CMS are also habitually used by developers due to ease-of-use

Wordpress, Moodle, and Drupal are all PHP based CMS platforms.