Difficulty
Hard

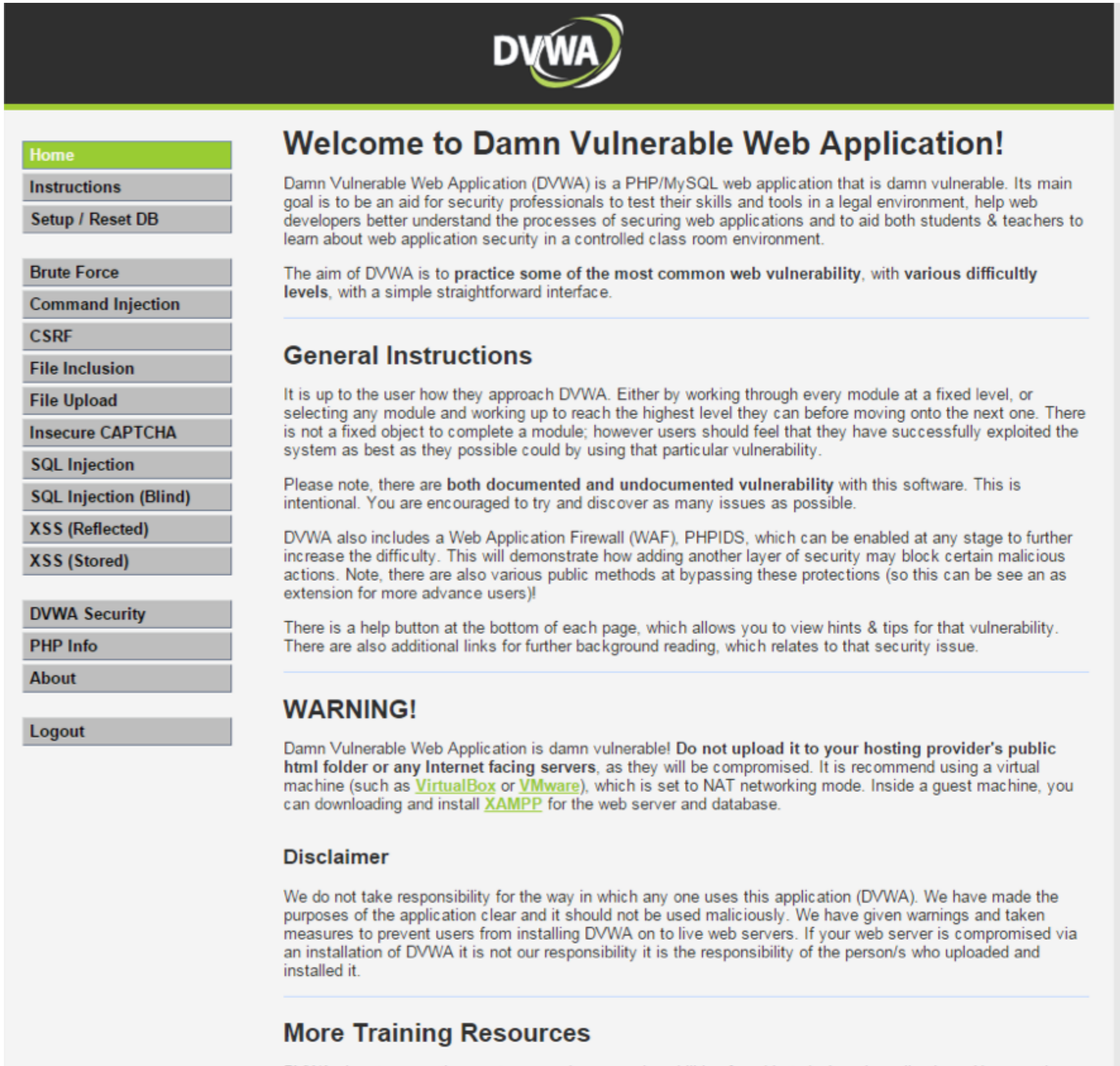
● ● ● ● ○

DAMN VULNERABLE WEB APPLICATION

WEBDEV@RGU

DVWA is, by its very nature, extremely unsecure! Because if this, it is unwise to launch this application on one of your own servers. Instead, we are going to be using it inside a contained virtual machine that does not have any access to the outside world. This should (hopefully) keep it a little bit safer. There are further instructions on moodle about starting this VM instance up.

When you first load DVWA you will see as screen like this:



Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerability**, with **various difficultly levels**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be see an as extension for more advance users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

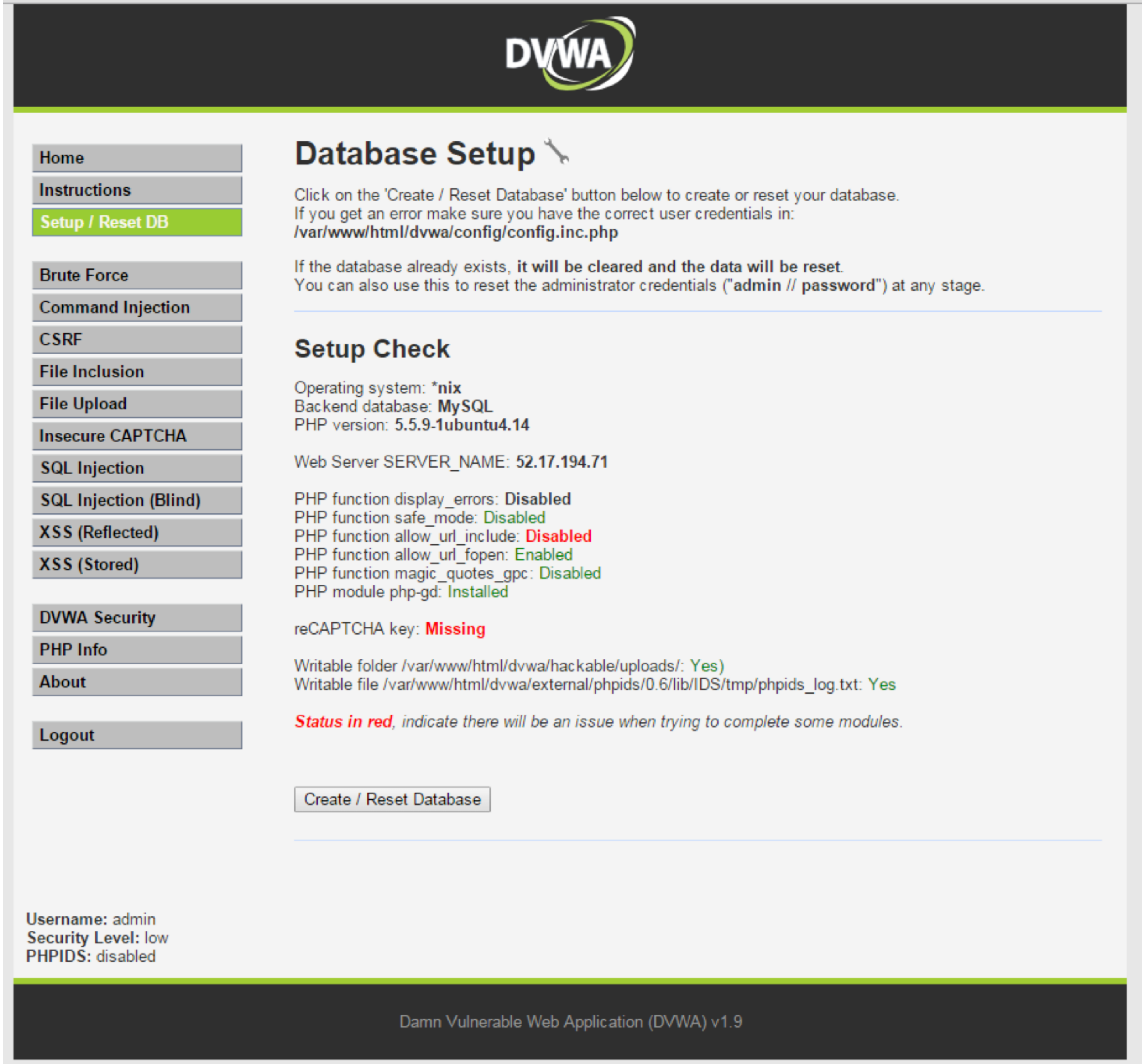
Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can downloading and install XAMPP for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

The setup/reset page allows you to check the current stability of the web app and reset it is someone has really pwned it. Hopefully as this is on your own separate instance…nothing should really break.



Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in:
/var/www/html/dvwa/config/config.inc.php

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

Setup Check

Operating system: ***nix**
Backend database: **MySQL**
PHP version: **5.5.9-1ubuntu4.14**

Web Server SERVER_NAME: **52.17.194.71**

PHP function display_errors: **Disabled**
PHP function safe_mode: Disabled
PHP function allow_url_include: Disabled
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module php-gd: Installed

reCAPTCHA key: **Missing**

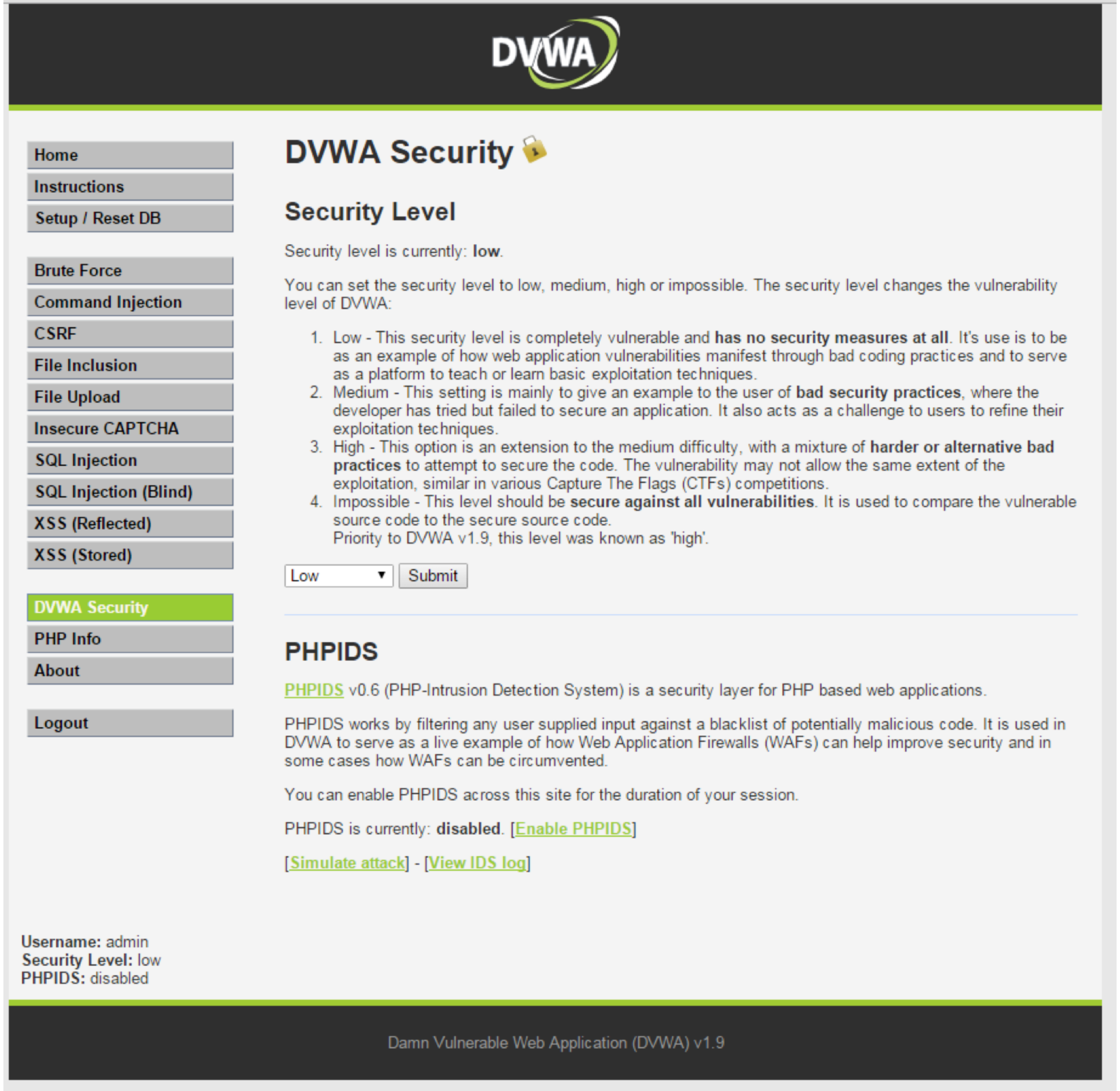
Writable folder /var/www/html/dvwa/hackable/uploads/: Yes)
Writable file /var/www/html/dvwa/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

Status in red, *indicate there will be an issue when trying to complete some modules.*

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.9

Initially make sure the DVWA Security setting is set to low. This is the lowest (most vulnerable) setting. You can always increase this later to look at the source to see how to protect your site more fully.



DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Priority to DVWA v1.9, this level was known as 'high'.

Low ▾ Submit

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.9

Protecting Against Attacks

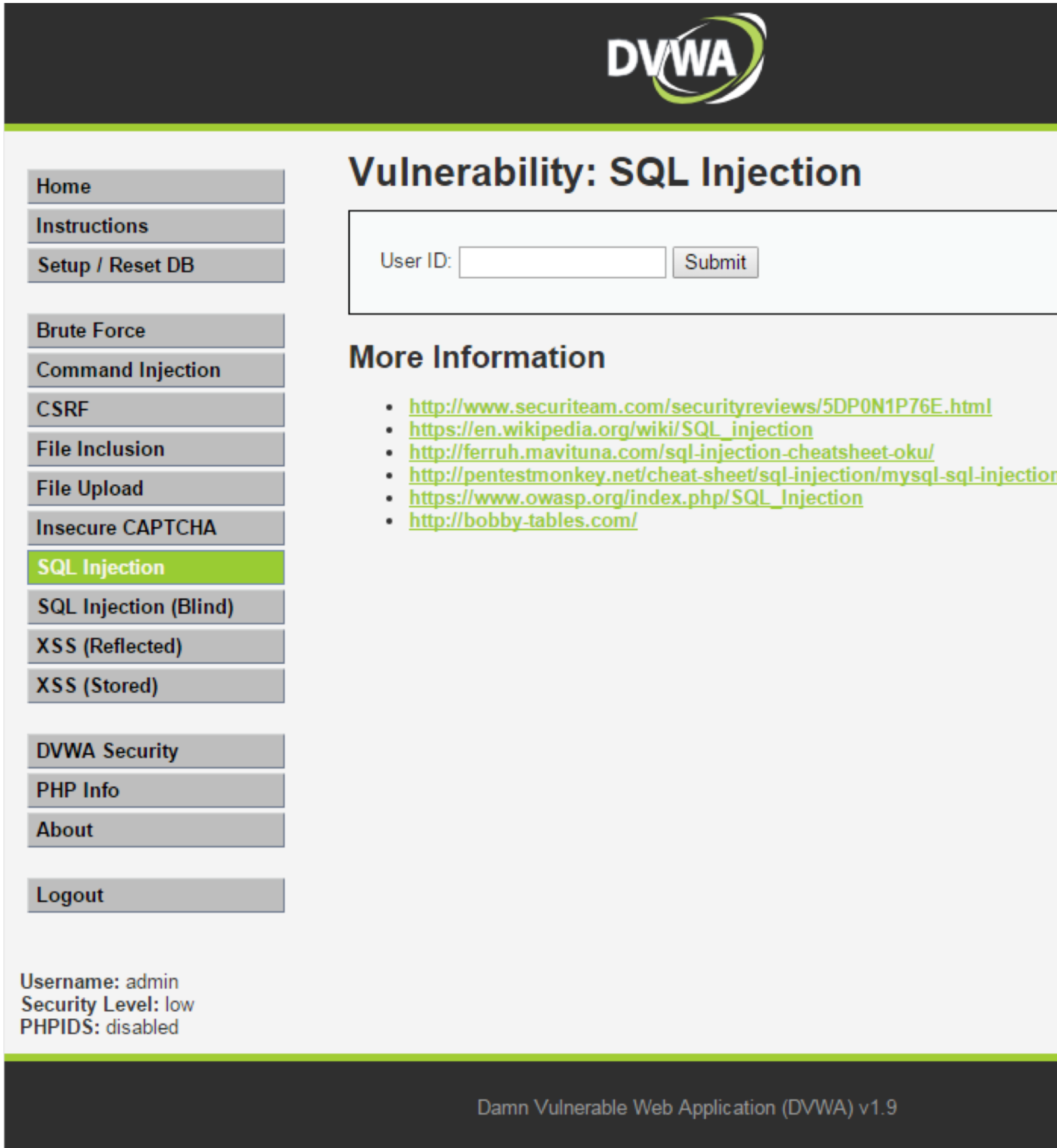
The great thing about DVWA is that it also shows you how to protect against the attacks.
For example if you go to the SQL injection button, you will see the page below.

Links are provided to give you more information about the vulnerability.

The USERID box here replicates the type of user input box you would see on a form with a SQL injection vulnerability.

Try typing our test injection code from the lecture. You can also click the View Help button and try some of the suggestions there. You should see when you get the right hack, that you get a dump of the entire database.

If you hit the view source button, you can see what code was implemented for this feature.



Vulnerability: SQL Injection

User ID:

More Information

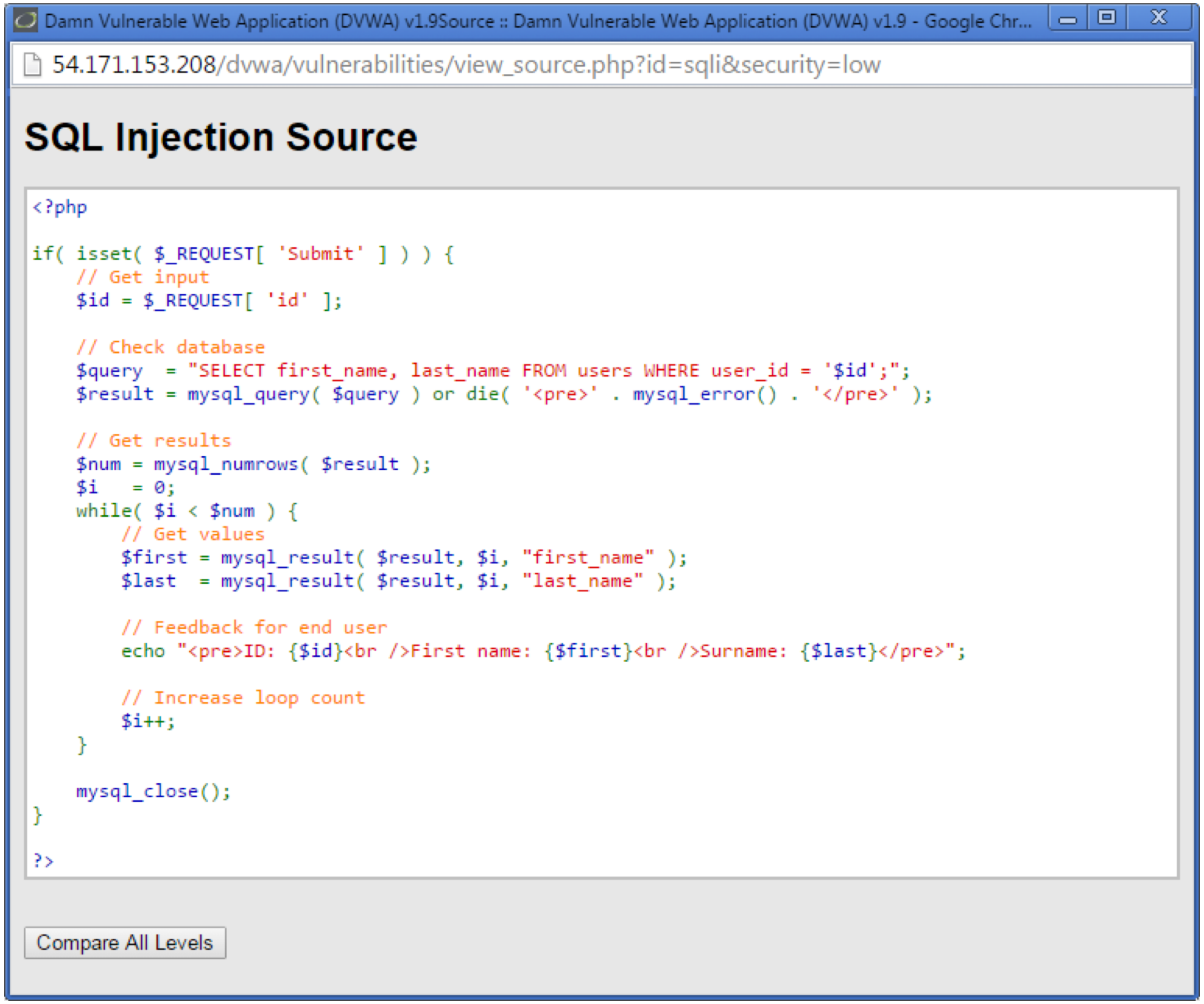
- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injectio
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.9

If you want to see the more secure code, you can just click the “compare all Levels” button and see the code for the different levels of security.

This is the same process for any of the vulnerabilities covered in the lecture.


Damn Vulnerable Web Application (DVWA) v1.9Source :: Damn Vulnerable Web Application (DVWA) v1.9 - Google Chr...
54.171.153.208/dvwa/vulnerabilities/view_source.php?id=sqli&security=low

SQL Injection Source

```
<?php

if( isset( $_REQUEST[ 'Submit' ] ) ) {
    // Get input
    $id = $_REQUEST[ 'id' ];

    // Check database
    $query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
    $result = mysql_query( $query ) or die( '<pre>' . mysql_error() . '</pre>' );

    // Get results
    $num = mysql_numrows( $result );
    $i   = 0;
    while( $i < $num ) {
        // Get values
        $first = mysql_result( $result, $i, "first_name" );
        $last  = mysql_result( $result, $i, "last_name" );

        // Feedback for end user
        echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";

        // Increase loop count
        $i++;
    }

    mysql_close();
}

?>
```

File Upload Vulnerability

For the file upload vulnerability you can get a good php shell below

http://b374k-shell.googlecode.com/files/b374k-2.8.php

renaming this to something easier to find, like break.php might be a good idea.

This is a very very powerful script. Don’t abuse it

For The Rest Of Today

You will want to:

- Try out a variety of attacks on DVWA (you can use the lecture slides or anything else that you think may help)
- Have a look at the different security levels and the underlying code for these. You may want to use this code within your own work to make it more secure (remember to put in //