EXPLOIT
the Fabulous
EXECUTABLE
IMAGE

The Executable Image Exploit
DEFCON XV
Michael Schrenk
What you’ll learn

• What are the origins of this exploit.

• What are the differences between “executable” and “Static” images?

• How to create images with PHP & GD

• How to fool servers into executing images (instead of serving them to browsers)

• How to do cool things with images on Web 2.0 websites
What you’ll learn

• What are the origins of this exploit.
• What are the differences between “executable” and “Static” images?
• How to create images with PHP & GD
• How to fool servers into executing images (instead of serving them to browsers)
• How to do cool things with images on Web 2.0 websites
What you’ll learn

• What are the origins of this exploit.
• What are the differences between “executable” and “Static” images?
• How to create images with PHP & GD
• How to fool servers into executing images (instead of serving them to browsers)
• How to do cool things with images on Web 2.0 websites
What you’ll learn

• What are the origins of this exploit.
• What are the differences between “executable” and “Static” images?
• How to create images with PHP & GD
• How to fool servers into executing images (instead of serving them to browsers)
• How to do cool things with images on Web 2.0 websites
What you’ll learn

• What are the origins of this exploit.
• What are the differences between “executable” and “Static” images?
• How to create images with PHP & GD
• How to fool servers into executing images (instead of serving them to browsers)
• How to do cool things with images on Web 2.0 websites
What you won't learn

• This is not the GDI exploit
• This exploit works on images downloaded from servers
• This is not a client-side exploit.
• That's not entirely true...
What you won't learn

- This is not the GDI exploit
- This exploit works on images downloaded from servers
- This is not a client-side exploit.
- That's not entirely true...
What you won't learn

• This is not the GDI exploit
• This exploit works on images downloaded from servers
• This is not a client-side exploit.
• That's not entirely true...
What you won't learn

- This is not the GDI exploit
- This exploit works on images downloaded from servers
- This is not a client-side exploit.
- That's not entirely true...

`javascript` `image`
Goals

– To learn how program “executable images”
– To learn where they can be applied
– To get you started on your own applications
– This is not a “code-heavy” presentation!
Goals

– To learn how program “executable images”
– To learn where they can be applied
– To get you started on your own applications
– This is not a “code-heavy” presentation!
Goals

– To learn how program “executable images”
– To learn where they can be applied
– To get you started on your own applications
– This is not a “code-heavy” presentation!
Goals

- To learn how program “executable images”
- To learn where they can be applied
- To get you started on your own applications
- This is not a “code-heavy” presentation!
Who’s Schrenk?

• Long-time webbot writer
• 8th DEFCON, 3rd time speaker
• Minneapolis & Madras (Chennai)
Who’s Schrenk?

• Long-time webbot writer
• 8th DEFCON, 3rd time speaker
• Minneapolis & Madras (Chennai)
Who’s Schrenk?

• Long-time webbot writer
• 8th DEFCON, 3rd time speaker
• Minneapolis & Madras (Chennai)
Who’s Schrenk?

• Long-time webbot writer
• 8\textsuperscript{th} DEFCON, 3\textsuperscript{rd} time speaker
• Minneapolis & Madras (Chennai)
Who’s Schrenk?

• Long-time webbot writer
• 8th DEFCON, 3rd time speaker
• Minneapolis & Madras (Chennai)
Who’s Schrenk?

- Long-time webbot writer
- 8th DEFCON, 3rd time speaker
- Minneapolis & Madras (Chennai)
Exploit Origins

- To create a really good MySpace tracker.
- Wanted to add image to “friends” pages that looks like this:
  
  `<IMG src="image.php?im=test">`

- Got frustrated because MySpace doesn’t allow such images.
- Most web 2.0 sites don’t
Exploit Origins

• To create a really good MySpace tracker.

• Wanted to add image to “friends” pages that looks like this:

  <IMG src="image.php?im=test">

• Got frustrated because MySpace doesn’t allow such images.

• Most web 2.0 sites don’t
Exploit Origins

• To create a really good MySpace tracker.

• Wanted to add image to “friends” pages that looks like this:

  `<IMG src="image.php?im=test">`

• Got frustrated because MySpace doesn’t allow such images.

• Most web 2.0 sites don’t
Exploit Origins

- To create a really good MySpace tracker.
- Wanted to add image to “friends” pages that looks like this:
  
  `<IMG src="image.php?im=test">`

- Got frustrated because MySpace doesn’t allow such images.
- Most web 2.0 sites don’t
Exploit Origins

• MySpace won’t allow you to reference images like `<IMG src="image.php?im=test">` because:
  – This is a program not actually an image
  – It is a executable image
    • May still send an image to the browser
    • May also:
      – Write cookies
      – Track environment variables
      – Access databases, Send instant messages, etc
Exploit Origins

- MySpace won’t allow you to reference images like `<IMG src="image.php?im=test">` because:
  - This is a program not actually an image
  - It is a executable image
    - May still send an image to the browser
    - May also:
      - Write cookies
      - Track environment variables
      - Access databases, Send instant messages, etc
What is a executable image?

• Executable images are programs
• Often used when images are stored in databases
• Can dynamically deliver “altered” images
  – Watermarks (with time stamp or IP addresses)
  – CAPTCHAs
What is a executable image?

- Executable images are programs
- Often used when images are stored in databases
- Can dynamically deliver “altered” images
  - Watermarks (with time stamp or IP addresses)
  - CAPTCHAs
What is a executable image?

- Executable images are programs
- Often used when images are stored in databases
- Can dynamically deliver “altered” images
  - Watermarks (with time stamp or IP addresses)
  - CAPTCHAs
Executable image example #1

You can “pull” an image from a database with code like this

```php
<?php
    # show_img.php
    // Get image (blob) from database
    include("mysql_library.php");
    $id = $_GET['id'];
    $sql = "select IMAGE from db_table where ID = '$id'";
    $img = execute_sql($sql);

    //Send image to the browser
    header("Content-type: image/jpeg");
    echo base64_decode($img);
    exit;
?>
```
Executable image example #1

You can “pull” an image from a database with code like this

```php
<?php
# show_img.php
// Get image (blob) from database
include("mysql_library.php");
$id = $_GET['id'];
$sql = "select IMAGE from db_table where ID = '$id';
$img = execute_sql($sql);

//Send image to the browser
header("Content-type: image/jpeg");
echo base64_decode($img);
exit;
?>
```
Executable image example #1

- Doesn’t require any special graphics libraries
- Image must be previously stored in database as a blob
- Images may be referenced by index or by name.
- Useful when web servers lack file permissions to read/write files
Executable image example #1

- Doesn’t require any special graphics libraries
- Image must be previously stored in database as a blob
- Images may be referenced by index or by name.
- Useful when web servers lack file permissions to read/write files
Executable image example #1

• Doesn’t require any special graphics libraries
• Image must be previously stored in database as a blob
• Images may be referenced by index or by name.
• Useful when web servers lack file permissions to read/write files
Executable image example #1

- Doesn’t require any special graphics libraries
- Image must be previously stored in database as a blob
- Images may be referenced by index or by name.
- Useful when web servers lack file permissions to read/write files
Executable image example #2

You can identify the image to display in a query string.

<img src="some_image.php?id=riviera.jpg"/>
<?php

// Create mime type for a jpg image
header("Content-type: image/jpeg");

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg($_GET['id']);

// Create an image and send to browser
imagejpeg($im);

// Destroy the old image (no longer needed)
imagedestroy($im);

// Ensure file execution is over
exit;
?>
<?php

// Create mime type for a jpg image
header("Content-type: image/jpeg");

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg($_GET['id']);

// Create an image and send to browser
imagejpeg($im);

// Destroy the old image (no longer needed)
imagedestroy($im);

// Ensure file execution is over
exit;
Executable image example #2

```php
<?php

// Create mime type for a jpg image
header("Content-type: image/jpeg");

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg($_GET['id']);

// Create an image and send to browser
imagejpeg($im);

// Destroy the old image (no longer needed)
imagedestroy($im);

// Ensure file execution is over
exit;
?>
```
<?php

// Create mime type for a jpg image
header("Content-type: image/jpeg");

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg($_GET['id']);

// Create an image and send to browser
imagejpeg($im);

// Destroy the old image (no longer needed)
imagedestroy($im);

// Ensure file execution is over
exit;
?>
<?php

// Create mime type for a jpg image
header("Content-type: image/jpeg");

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg($_GET['id']);

// Create an image and send to browser
imagejpeg($im);

// Destroy the old image (no longer needed)
imagedestroy($im);

// Ensure file execution is over
exit;
?>
executable image example #2

You can identify the image to display in a query string.

http://localhost/defcon/show_referenced.php?id=alexispark.jpg

http://localhost/defcon/show_referenced.php?id=riviera.jpg
Executable image example #2

You can identify the image to display in a query string.

http://localhost/defcon/show_referenced.php?id=alexispark.jpg

http://localhost/defcon/show_referenced.php?id=riviera.jpg

Why do this?
Because it’s an executable program!
It mimics the actions of a real image
Executable image example #2

You can identify the image to display in a query string.

http://localhost/defcon/show_referenced.php?id=alexispark.jpg

http://localhost/defcon/show_referenced.php?id=riviera.jpg

Why do this?
Because it’s an executable program!
It mimics the actions of a real image
Executable image example #3

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg( $_GET['id'] );

// Define font and font color
$font = 'arial.ttf';
 setColor = imagecolorallocate ($im, 255, 120, 0);

// Define executable content
$text = date("M d, Y h:m:s A", time());
$angle = rand(0, 90);
imagettftext($im, 20, $angle, 11, 301, $color, $font, $text);

// Create an image from the handle and send to browser
header("Content-type: image/jpeg");
imagejpeg($im);

// Destroy the old image (no longer needed)
imagedestroy($im);
exit;
Executable image example #3

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg( $_GET['id'] );

// Define font and font color
$font = 'arial.ttf';
$color = imagecolorallocate ($im, 255, 120, 0);

// Define executable content
$text = date("M d, Y h:m:s A", time());
$angle = rand(0, 90);
imagettftext($im, 20, $angle, 11, 301, $color, $font, $text);

// Create an image from the handle and send to browser
header("Content-type: image/jpeg");
imagejpeg($im);

// Destroy the old image (no longer needed)
imagedestroy($im);
exit;
// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg( $_GET['id'] );
// Define font and font color
$font = 'arial.ttf';
$color = imagecolorallocate ($im, 255, 120, 0);
// Define executable content
$text = date("M d, Y h:m:s A", time());
$angle = rand(0, 90);
imagettftext($im, 20, $angle, 11, 301, $color, $font, $text);
// Create an image from the handle and send to browser
header("Content-type: image/jpeg");
imagejpeg($im);
// Destroy the old image (no longer needed)
imagedestroy($im);
exit;
Executeable image example #3

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg( $_GET['id'] );
// Define font and font color
$font = 'arial.ttf';
$color = imagecolorallocate ($im, 255, 120, 0);
// Define executable content
$text = date("M d, Y h:m:s A", time());
$angle = rand(0, 90);
imagettftext($im, 20, $angle, 11, 301, $color, $font, $text);
// Create an image from the handle and send to browser
header("Content-type: image/jpeg");
imagejpeg($im);
// Destroy the old image (no longer needed)
imagedestroy($im);
exit;
Executable image example #3

// Create an image handle from an actual JPG image
$im = imagecreatefromjpeg( $_GET['id'] );
// Define font and font color
$font = 'arial.ttf';
$color = imagecolorallocate ($im, 255, 120, 0);
// Define executable content
$text = date("M d, Y h:m:s A", time());
$angle = rand(0, 90);
imagettftext($im, 20, $angle, 11, 301, $color, $font, $text);
// Create an image from the handle and send to browser
header("Content-type: image/jpeg");
imagejpeg($im);
// Destroy the old image (no longer needed)
imagedestroy($im);
exit;
Executable image example #3

Result of executable image #3

• Dynamic Example
Executable images Can:

• Display images stored in databases
• Programmatically select images to display
• Dynamically produce image content
Executable images Can:

- Display images stored in databases
- Programmatically select images to display
- Dynamically produce image content
Executable images *Can:*  
- Display images stored in databases  
- Programmatically select images to display  
- Dynamically produce image content
Executable images Can:

- Do anything a script can do:
  - Read referrer variables,
  - To see the page previous to viewing your image’s page
  - To see the query string on the previous page
Executable images Can:

• Do anything a script can do:
  – Read referrer variables,
    • To see the page previous to viewing you image’s page
    • To see the query string on the previous page
  – Read & write cookies
    • To track individuals
    • Works across domains
**Executable images Can:**

- **Do anything a script can do:**
  - Read referrer variables,
    - To see the page previous to viewing you image’s page
    - To see the query string on the previous page
  - Read & write cookies
    - To track individuals
    - Works across domains
  - **Access databases**
Executable images Can:

- Do anything a script can do:
  - Read referrer variables,
    - To see the page previous to viewing your image’s page
    - To see the query string on the previous page
  - Read & write cookies
    - To track individuals
    - Works across domains
  - Access databases
  - Communicate via email, SMS, etc.
Doing the dirty

// Create an image from the handle and send to browser
header("Content-type: image/jpeg");

// Write Cookie
setcookie("TestCookie", $value);

// Read Cookie
$old_cookie = $HTTP_COOKIE_VARS['TestCookie'];

// Get referer variable
referer = $_SERVER['HTTP_REFERER'];

// Get query strings
$query_string = $_SERVER['QUERY_STRING'];

// Anything else
imagejpeg($im);
Doing the dirty

// Create an image from the handle and send to browser
header("Content-type: image/jpeg");

// Write Cookie
setcookie("TestCookie", $value);

// Read Cookie
$old_cookie = $HTTP_COOKIE_VARS['TestCookie'];

// Get referer variable
referer = $_SERVER['HTTP_REFERER'];

// Get query strings
$query_string = $_SERVER['QUERY_STRING'];

// Anything else
imagejpeg($im);
Doing the dirty

// Create an image from the handle and send to browser
header("Content-type: image/jpeg");

// Write Cookie
setcookie("TestCookie", $value);

// Read Cookie
$old_cookie = $_SERVER['HTTP_COOKIE_VARS']['TestCookie'];

// Get referer variable
$referer = $_SERVER['HTTP_REFERER'];

// Get query strings
$query_string = $_SERVER['QUERY_STRING'];

// Anything else

imagejpeg($im);
Doing the dirty

// Create an image from the handle and send to browser
header("Content-type: image/jpeg");

// Write Cookie
setcookie("TestCookie", $value);

// Read Cookie
$old_cookie = $_HTTP_COOKIE_VARS["TestCookie"];  

// Get referer variable
referer = $_SERVER["HTTP_REFERER"];  

// Get query strings
$query_string = $_SERVER["QUERY_STRING"];  

// Anything else
imagejpeg($im);
Doing the dirty

// Create an image from the handle and send to browser
header("Content-type: image/jpeg");

// Write Cookie
setcookie("TestCookie", $value);

// Read Cookie
$old_cookie = $HTTP_COOKIE_VARS["TestCookie"];

// Get referer variable
referer = $_SERVER["HTTP_REFERER"];

// Get query strings
$query_string = $_SERVER["QUERY_STRING"];

// Anything else

imagejpeg($im);
Doing the dirty

// Create an image from the handle and send to browser
header("Content-type: image/jpeg");

// Write Cookie
setcookie("TestCookie", $value);

// Read Cookie
$old_cookie = $HTTP_COOKIE_VARS["TestCookie"];

// Get referer variable
$referer = $_SERVER['HTTP_REFERER'];

// Get query strings
$query_string = $_SERVER['QUERY_STRING'];

// Anything else
imagejpeg($im);
MySpace doesn't allow them!

<IMG SRC="some_image.jpg"/>

<IMG SRC="some_image.php"/>
Fooling apache to execute .JPGs

In the .htaccess file

```
AddType application/x-httpd-php .jpg
```

Tells apache to parse all files (in this or subsequent directories) with the .jpg extension as though they were PHP scripts!
Fooling apache to execute .JPGs

Once done, you can reference your executable images like this...

<img src="www.yourdomain.com/image.jpg">

Example: Dynamic JPG
Applications

Can be used on many (web 2.0) websites that let you post comments.
- Craigs List
- Ebay
- MySpace
- Fark
- PayPal (payment page)

Also on non-web environments
- Newsgroups (NNTP)
- Email
Applications

Can be used on many (web 2.0) websites that let you post comments.
– Craig's List
– Ebay
– MySpace
– Fark
– PayPal (payment page)

Also on non-web environments
– Newsgroups (NNTP)
– Email
Tracking people on MySpace

Add an inline (executable) image in a MySpace comment
Tracking people on MySpace

When one checks new messages or new comments

and the comment/message contains a executable image...

The userID is in \$_SERVER['HTTP_REFERER'];
Tracking people on MySpace

The userID lets you associate a cookie with their identity:

http://profile.myspace.com/index.cfm?fuseaction=user.viewprofile&friendid=userID

Anytime they revisit, you can track them.
Other MySpace fun

You can write an application that shows the viewing habits of all your friends by sending them each a message that contains a executable image.
Other MySpace fun

You can show one set of pictures to your MySpace friends, and another to set of images to non-friends.
Other MySpace fun

You can use these same cookies to track people’s movement on other sites (eBay, Craigslist, etc).

Since your cookies all belong to the domain that your executable image is on, your cookies will “appear” to function across domains.
Spanning domains

Your Browser

eBay Cookie

eBay.com
Spanning domains

Your Browser
  eBay Cookie
  Myspace cookie

eBay.com

MySpace.com
Spanning domains

Your Browser
- eBay Cookie
- Your Cookie
- Myspace cookie

eBay.com
- ex img

MySpace.com
- ex img

Your Server
Third Party Cookies

First party cookie
Third Party Cookies

First party cookie

Third party cookie
The Executable Image Exploit

www.schrenk.com

DEFCON XV Las Vegas Nevada
mike@schrenk.com

![Firefox Preferences](image)
<?php
header("P3P: policyref=\"http://www.yourDomain.com/w3c/p3p.xml\", CP=\"CAO DSP COR\"');
?>
How to Deploy P3P Privacy Policies on Your Web Site

This overview explains the steps to take when deploying a Platform for Privacy Preferences (P3P) privacy scheme on a Web site beginning with the Web site's natural language privacy policy. The examples are oversimplified and used only to illustrate the steps to deployment. For details on content, vocabulary, and structure of P3P privacy policies and alternate deployment options, see the Platform for Privacy Preferences (P3P) specification.

- The Steps for Deploying P3P
- Related Topics

The Steps for Deploying P3P

The following graphic shows the steps for deploying P3P.
Other Web 2.0 Fun

Show high quality images to members of your site and poor quality images to everyone else
Other Web 2.0 Fun

Embed identifying watermarks in images to track unauthorized use
Other Web 2.0 Fun

Create eBay auctions with images that change as you near the end of the auction
Other Web 2.0 Fun

Show different images in your eBay auction after people see your similar ad on Craig's List
Other Web 2.0 Fun

Evaluate websites you want to advertise on
Other Web 2.0 Fun

Receive an acknowledgement when an email is read

nonrepudiation
Other Web 2.0 Fun

Develop images with expiration dates
Getting ideas

Focus on applications where images can be loaded from your server (think across domains)

Use:
• Cookies
• Referrer variables to catch query strings

Images are easy to manipulate with PHP & GD
Getting ideas

Focus on applications where images can be loaded from your server (think across domains).

Use:
• Cookies
• Referrer variables to catch query strings

Images are easy to manipulate with PHP & GD
Getting ideas

Focus on applications where images can be loaded from your server (think across domains).

Use:
• Cookies
• Referrer variables to catch query strings

Images are easy to manipulate with PHP & GD
Defences

Watch what you put in query strings
• Sessions may be stolen if all of the session variable is in a query string

Allow people to upload images instead of referencing them
• Takes more server space & bandwidth
• Removes the “executable” from images.
Defences

Watch what you put in query strings
• Sessions may be stolen if all of the session variable is in a query string

Allow people to upload images instead of referencing them
• Takes more server space & bandwidth
• Removes the “executable” from images.
Thank You!

The Executable Image Exploit
DEFCON XV
Michael Schrenk