DRUPALCON LONDON

Doing Drupal security right

Presented by Gábor Hojtsy, Acquia

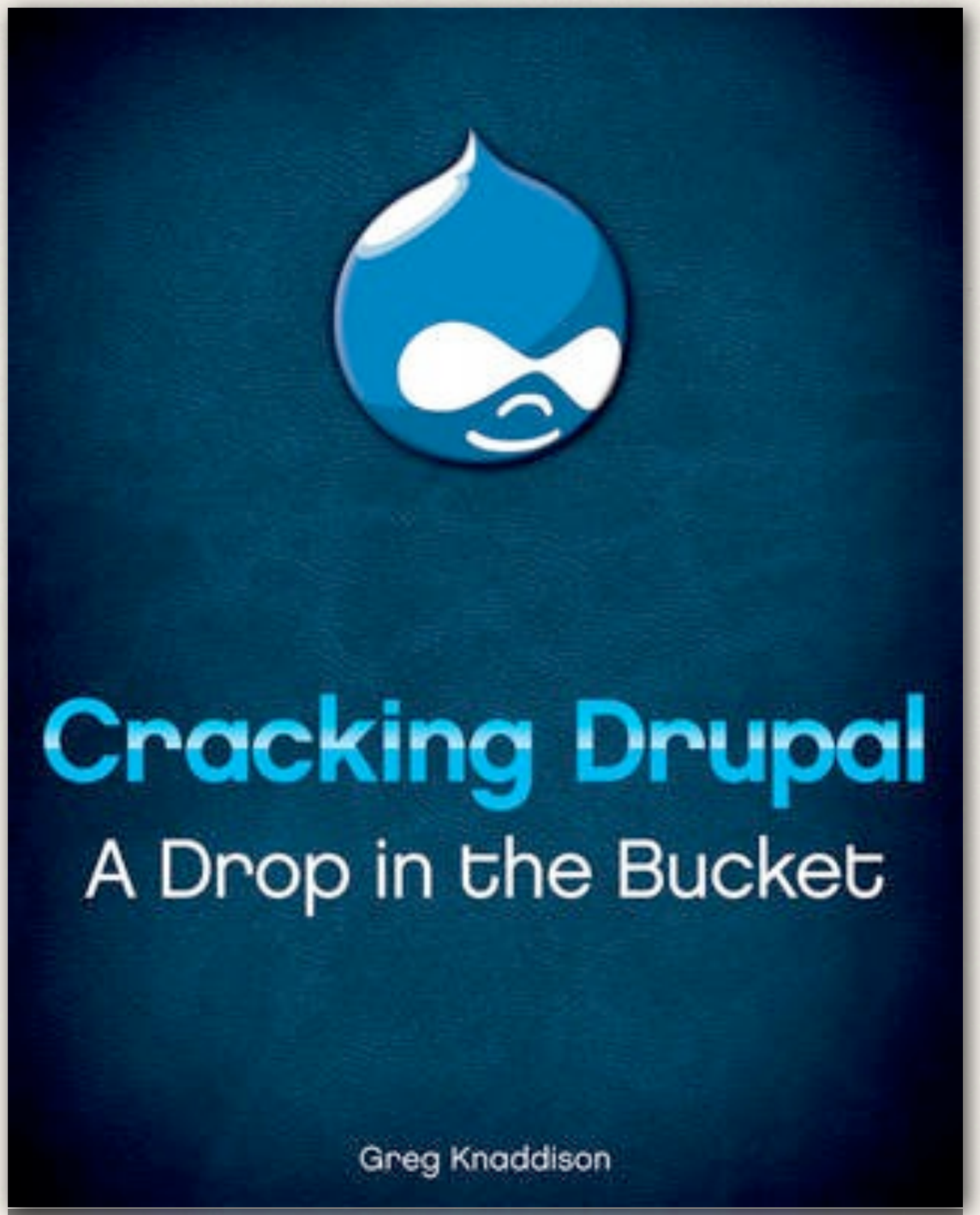
with special thanks to Greg Knaddison, Four Kitchens and Jakub Suchy

Why I’m here?

- Maintainer for Drupal 6
- De-facto member of the security team

Why are you here?

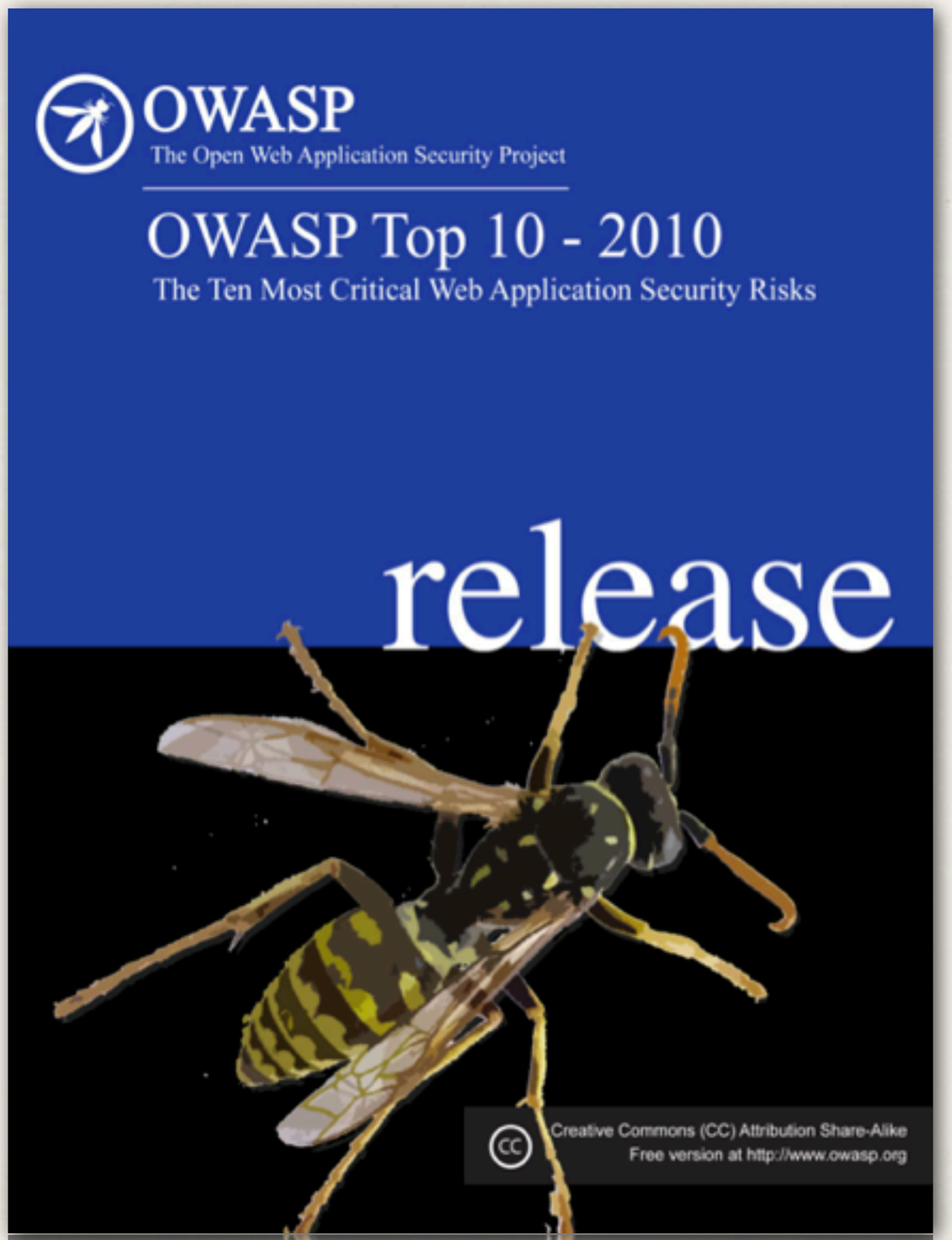
- Managers?
- Site builders?
- Themers?
- Developers?


Cracking Drupal
A Drop in the Bucket
Greg Knaddison

Are you affected?

With relatively simple holes, your administrator user can be taken over.

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project



Security misconfiguration

Heard of the mid-April wordpress.com attack?

Secure server

- Avoid using FTP at all cost, check your client tool
- Who do you share your server with? Are you confident? Run other apps?
- Keep your OS, PHP, SQL server, etc. up to date

Secure Drupal

- Is your admin password “admin”?
- Look at all “administer *” permissions
- “administer filters” can take over a site
- Use update.module, watch the security news (Wednesdays)

Secure Drupal

- Avoid any kind of PHP input, write your own modules instead
- Look into using paranoia.module
- Watch your input formats (you can be googled)
- Check out the security_review module.

Injection

```
index.php?id=12
```

```
mysql_query(“UPDATE mytable
SET value = ‘”. $value .”’
WHERE id = ”. $_GET[‘id’]);
```

Drupal approach

- db_query(“UPDATE {mytable} SET value = :value WHERE id = :id”, array(‘:value’ => $value, ‘:id’ => $id);
- If you need to include dynamic table or column names in your query, see db_escape_table()

2

Cross Site Scripting (XSS)

```
index.php?id=12
print $_GET['id'];
```

```
$output .= $node->title;
```

Giving full HTML access.

Unsafe tags in other formats.

64%

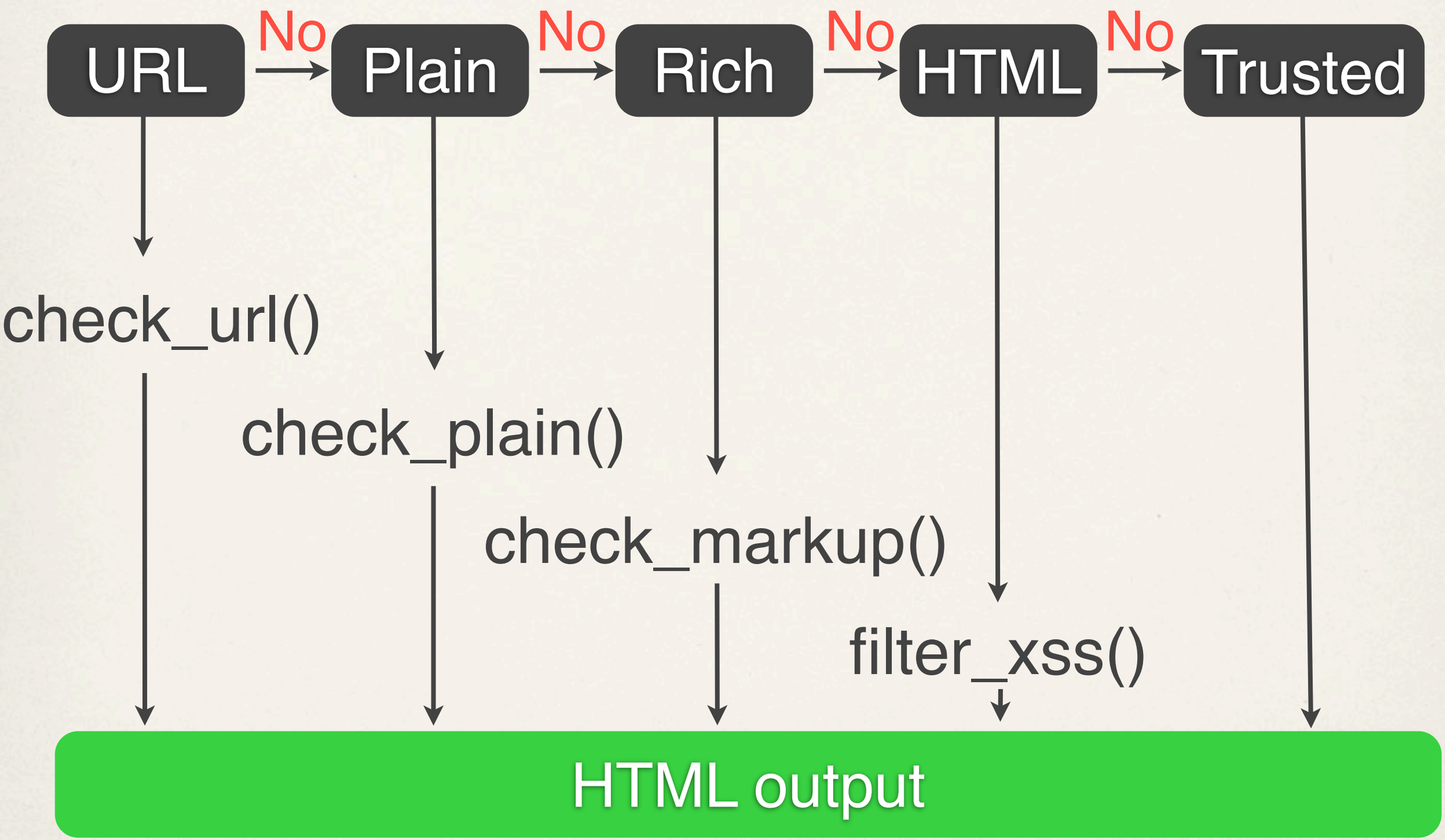
likelihood a website has a Cross site scripting issue

https://www.whitehatsec.com/assets/presentations/11PPT/PPT_topwebvulns_030311.pdf

```
jQuery.get('/user/1/edit',
  function (data, status) {
    if (status == 'success') {
      var p = /id="edit-user-edit-form-token"
value="([a-z0-9]*)"/;
      var matches = data.match(p);
      var token = matches[1];
      var payload = {
        "form_id": 'user_edit',
        "form_token": token,
        "pass[pass1]": 'hacked',
        "pass[pass2]": 'hacked'
      };
      jQuery.post('/user/1/edit', payload);
    }
  }
);
```

Example from Heine Deelstra, Drupal Security team lead
http://heine.familiedeelstra.com/change-password-xss
Technique (with code changes) works up to Drupal 6

Drupal approach



Drupal approach

- t(), format_plural() placeholders: %name, @url, !insecure

  t(‘%name has a blog at @url’, array(‘@url’ => valid_url($user->profile_blog), ‘%name’ => $user->name));

- Use Drupal.t(), Drupal.formatPlural() in JS.

Not all output is HTML

Authentication & sessions

- Weak password storage and account management
- Session hijacking / fixation
- Lack of session timeout / logout

Drupal approach

- Passwords are stored hashed
- Session IDs changed when permissions change
- Drupal works with Apache's SSL transport
- Modules to set certain URLs to use SSL

Insecure direct object references

index.php?id=12

```
db_query(“SELECT * FROM {node}
WHERE nid = :id”, array(‘:id’
=> $_GET[‘id’]));
```

Drupal approach

- Menu system handles permission checking
- user_access(‘administer nodes’, $account)
- node_access(‘edit’, $node, $account);
- $select->addtag(‘node_access’);
- Form API checks for data validity

Cross Site Request Forgery (CSRF)

```
<img src="http://example.com/
user/logout" />
```

```
http://example.com/index.php?
delete=12
```

```
<img src="http://example.com/
index.php?delete=12" />
```

Drupal approach

- Form API works with POST submissions by default (makes it harder)
- Form API includes form tokens, requires form retrieval before submission, checks valid values
- drupal_valid_token() provided to generate/validate tokens for GET requests

7

Insecure cryptographic storage

Drupal approach

- Drupal stores user passwords hashed with a one-way hash
- Different randomly generated private key is provided on each site, which can be used to do reversible encryption
- Modules exist to help encrypt more data
- Up to you to ensure backups are properly protected

Failure to restrict URL access

Drupal approach

- Menu system uses access callback and access arguments
- Continually review permissions

Insufficient transport protection

Heard of Firesheep?

Drupal approach

- Run Drupal on top of full SSL
- Use securepages and securepages_prevent_hijack to wall your important pages
- http://drupalscout.com/knowledge-base/drupal-and-ssl-multiple-recipes-possible-solutions-https
- Use a valid certificate

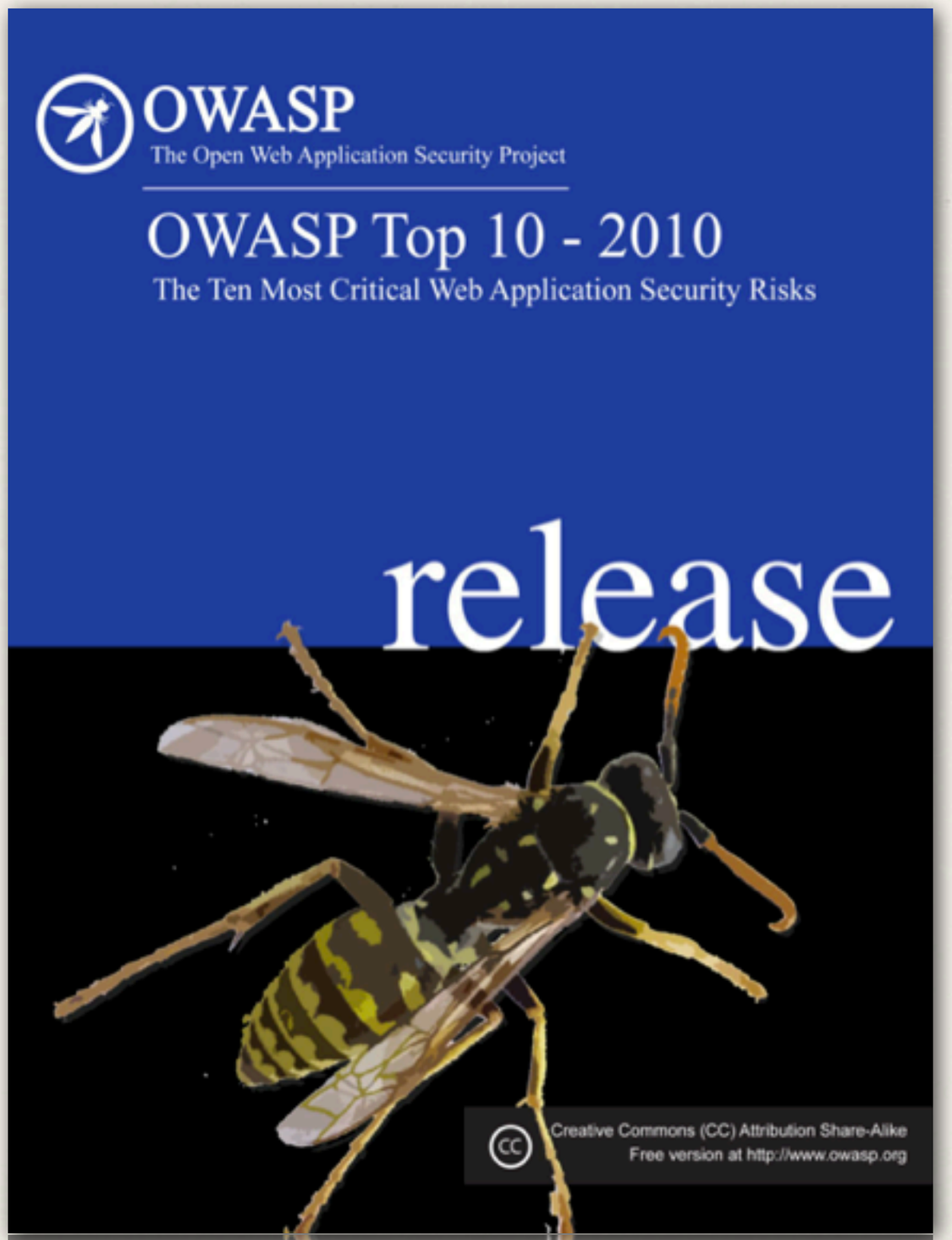
Unvalidated redirects

http://example.com/index.php?
target=evil.com

Drupal approach

- Drupal has various internal redirections, which use local paths and generate URLs based on them
- Look for use of drupal_goto() and Form API #redirect instances in your modules to validate their compliance

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project



Is Open Source secure?

“Open Source is secure”

- Open Source makes people look at it
- Popularity gets more eyes
- There are always more smart people to find and fix problems

“Open Source is insecure”

- People can equally find holes
- Some people (inadvertently) disclose issues in the public
- Fix becomes public and can / will be reviewed

Is Drupal secure?

Developers and users

- Drupal APIs are designed to be secure
- It is eventually up to programmers to use them that way
- http://drupal.org/writing-secure-code
- Tools designed for security can still be misconfigured

Drupal security team

A team of volunteers working to ensure best security of Drupal and thousands of contributed modules

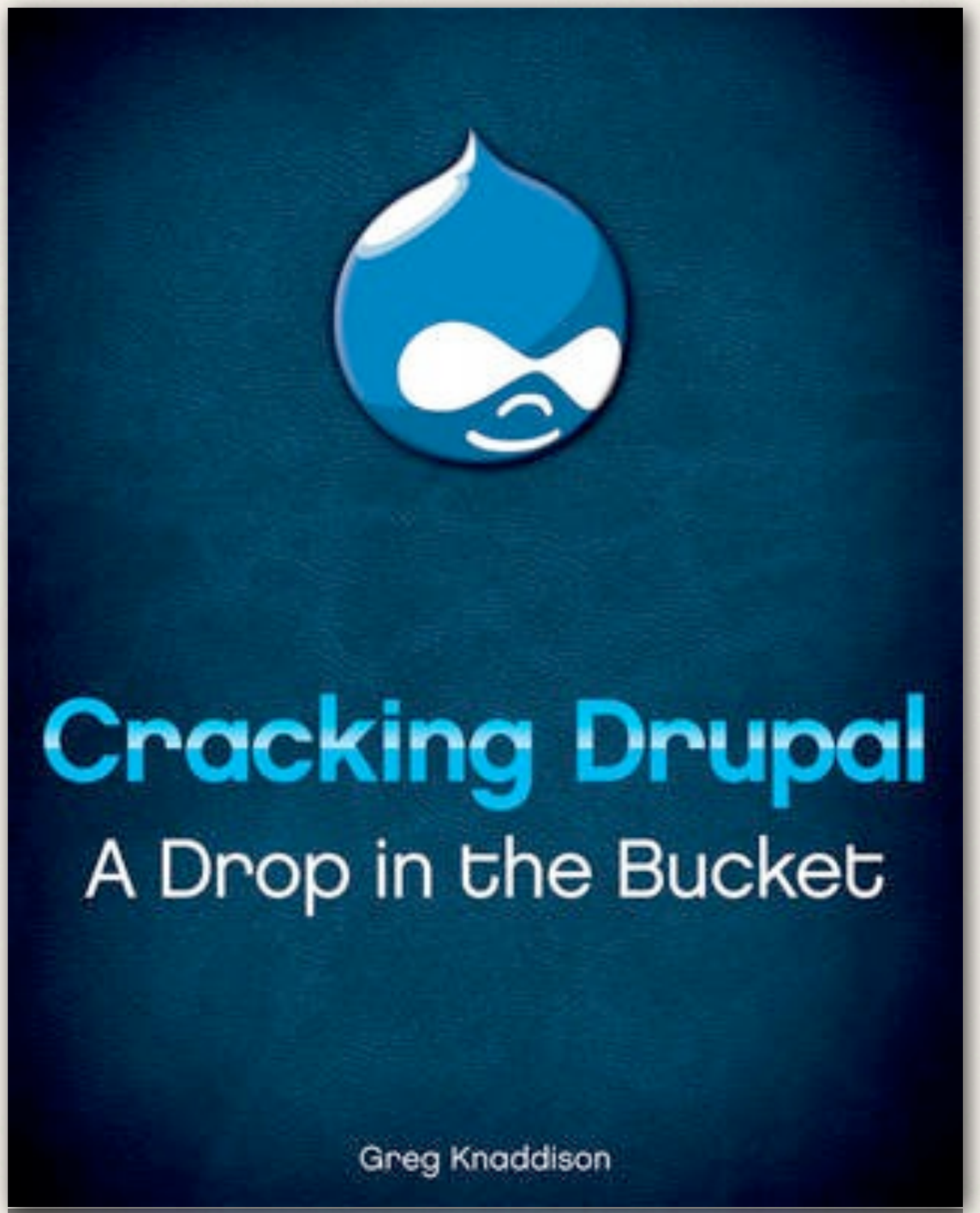
Design. Educate. Fix.

What's supported?

- Drupal core and all(!) contributed projects on drupal.org
- Stable releases (development versions only for very popular modules)
- Not actively looking for vulnerabilities in contributed modules
- Only current and one earlier versions are supported: now 7.x and 6.x

Points of contact

- Releases at http://drupal.org/security
- Reporting issues: http://drupal.org/node/101494
- Reporting cracked sites: http://drupal.org/node/213320
- Discuss general issues: http://groups.drupal.org/best-practices-drupal-security


Cracking Drupal
A Drop in the Bucket
Greg Knaddison

These slides are (CC)

Images used:

http://www.flickr.com/photos/rtv/2398561954/

http://www.flickr.com/photos/jonk/19422564/

http://www.flickr.com/photos/duncan/2693141693/

http://www.flickr.com/photos/duncan/2742371814

http://www.flickr.com/photos/jontintinjordan/3736095793/

http://www.flickr.com/photos/djbrady/2304740173/

http://www.flickr.com/photos/inkytwist/2654071573/

http://www.flickr.com/photos/duncan/2741594585/

http://www.flickr.com/photos/shellysblogger/2924699161/

http://www.flickr.com/photos/blogumentary/434097609/

http://www.flickr.com/photos/glamhag/2214986176/

http://www.flickr.com/photos/duncan/2693140217/

This presentation created by Gábor Hojtsy

Questions?

What did you think?

What did you think?

Locate this session on the DrupalCon London website:

http://london2011.drupal.org/conference/schedule

DRUPALCON LONDON

What did you think?

Locate this session on the DrupalCon London website:

http://london2011.drupal.org/conference/schedule

Click the “Take the survey” link

What did you think?

Locate this session on the DrupalCon London website:

http://london2011.drupal.org/conference/schedule

Click the “Take the survey” link

THANK YOU!