

Your site is vulnerable.

(really, it is)

Greg

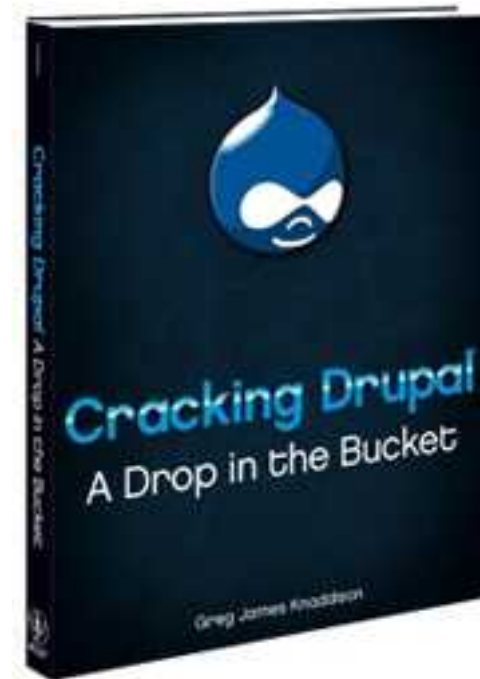
- Drupaler for 4 years
- Drupal Association
- Help with lots of d.o
- 20+ modules
 - Pathauto, token
- Drupal in Colorado
- MasteringDrupal.com
- DrupalDashboard.com



Wrote a book

"Cracking Drupal is probably going to be the first Drupal book I buy."

- Angie 'webchick' Byron



Cracking Drupal: A Dro

by [Greg Knaddison](#) (Author)

★★★★★ [Customer reviews](#)

List Price: ~~\$40.00~~

Price: **\$26.40** & this item

You Save: **\$13.60 (34%)**

In Stock.

Ships from and sold by **Amazon.c**

GVS



- Full service
- Design, development
- Usability reviews
- Community focused
- Progressive

Now....

- Security reviews
(with Ben) →



Worry

Your site is vulnerable.

You can make it safer.

“A site is secure if private data is kept private, the site cannot be forced offline or into a degraded mode by a remote visitor, the site resources are used only for their intended purposes, and the site content can be edited only by appropriate users.”

Some guy – Cracking Drupal chapter 1

- Abusing resources
- Stealing data
- Altering data



Worry in a
prioritized way.



Choose your strategy

Stay ahead of the pack?

Protect valuable assets?

When do attacks occur?

Table 3-1 Exploits

WORM/EXPLOIT	PATCH RELEASE DATE	WORM DATE
Santy*	November 2004	December 2004
Code Red	June 18, 2001	July 13, 2001
SQL Slammer	July 24, 2002	January 25, 2003
Sadmind	December 1999 / October 2000	May 8, 2001

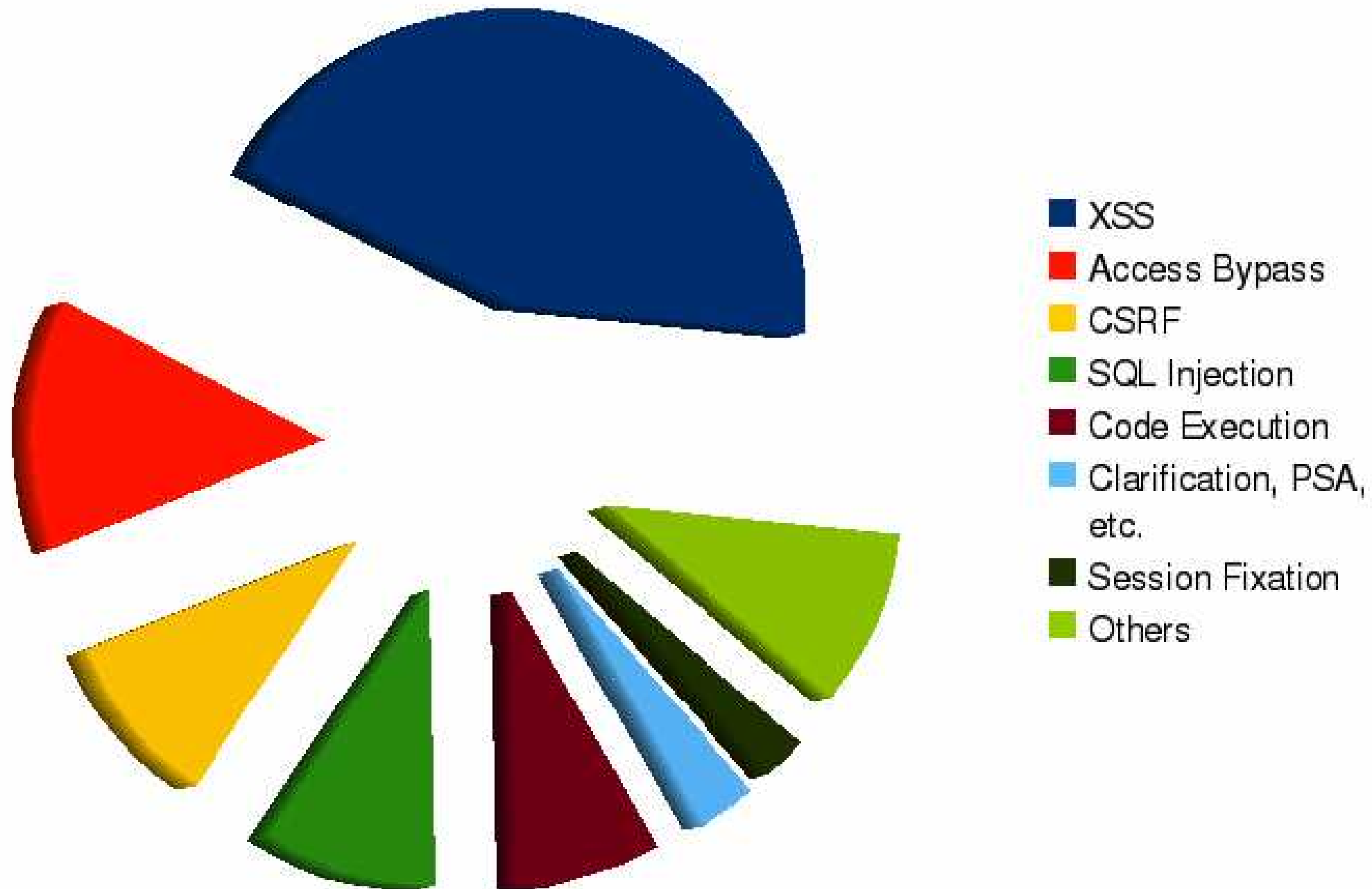
NOTE *Santy was the worm that attacked a site of mine and that first alerted me to the need for attention to security in web applications.

Source: *Cracking Drupal* Chapter 3

Keep up to date

- Know about releases
- Have a method to update your site
- Do it

Vulnerabilities by Type



Protect with configuration

Anything you can do XSS can do (better)

```
jQuery.get(Drupal.settings.basePath + 'user/1/edit',  
function (data, status) {  
  if (status == 'success') {  
    // Extract the token and other required data  
    var matches = data.match(/id="edit-user-profile-form-form-token" value="([a-z0-9])"/);  
    var token = matches[1];  
    // Post the minimum amount of fields. Other fields get their default values.  
    var payload = {  
      "form_id": 'user_profile_form',  
      "form_token": token,  
      "pass[pass1]": 'hacked',  
      "pass[pass2]": 'hacked'  
    };  
    jQuery.post(Drupal.settings.basePath + 'user/1/edit', payload);  
  }  
});
```

<http://crackingdrupal.com/node/8>

Default	Name	Roles	Operations
<input checked="" type="radio"/>	Filtered HTML	All roles may use default format	configure
<input type="radio"/>	Full HTML	No roles may use this format	configure delete

[Set default format](#)

Allowed HTML tags:

```
<a> <em> <strong> <cite> <code> <ul> <ol> <li> <dl> <dt> <dd>
```

If "Strip disallowed tags" is selected, optionally specify tags which should not be stripped. JavaScript event attributes are always stripped.

Name

Weight

<input type="checkbox"/> URL filter	0
<input type="checkbox"/> HTML filter	1
<input type="checkbox"/> Line break converter	2
<input type="checkbox"/> HTML corrector	10

Save configuration

Drupal.org <http://drupal.org/node/224921>
Cracking Drupal: Chapter 3

demo time

XSS For Themers / Coders (and reviewers)

Themers

- Read tpl.php and default implementations
- Rely on your module developer for variables

Developers

Where does this text come from?

Is there a way a user can change it?

In what *context* is it being used?

Context

- Mail context
- Database context
- Web context
- Server context

Take an hour:

<http://acko.net/blog/safe-string-theory-for-the-web>

demo time

Resources

- <http://drupal.org/security-team>
- <http://drupal.org/security>
- <http://drupal.org/writing-secure-code>
- <http://drupal.org/security/secure-configuration>
- <http://groups.drupal.org/node/15254> - discussion group
- <http://heine.familiedeelstra.com/>
- Cracking Drupal - <http://crackingdrupal.com>

Click to add title

Your site is vulnerable.

(really, it is)

Maybe you don't know it, but it is.

Greg

- Drupaler for 4 years
- Drupal Association
- Help with lots of d.o
- 20+ modules
 - Pathauto, token
- Drupal in Colorado
- MasteringDrupal.com
- DrupalDashboard.com



I'm pretty awesome.

Wrote a book

"Cracking Drupal is probably going to be the first Drupal book I buy."

- Angie 'webchick' Byron



Cracking Drupal: A Dro

by Greg Knaflitz, Matt Mabe

List Price: ~~\$40.00~~

Price: **\$26.40** on this item

You Save: **\$13.60 (34%)**

In Stock.

Ships from and sold by Amazon.com

It's pretty awesome.

GVS



gvs

- Full service
- Design, development
- Usability reviews
- Community focused
- Progressive

Now...

- Security reviews
(with Ben) →



Damn, they're really awesome.

Cracking Drupal: Webinar 1

Worry Configuration Code Resources

Click to add title

Worry

gvs 5

Your site has some vulnerabilities in it somewhere – given enough time/effort someone could break in.

[Click to add title](#)

Your site is vulnerable.

You can make it safer.

Click to add title

“A site is secure if private data is kept private, the site cannot be forced offline or into a degraded mode by a remote visitor, the site resources are used only for their intended purposes, and the site content can be edited only by appropriate users.”

Some guy – Cracking Drupal chapter 1

Cracking Drupal: Webinar 1

Worry Configuration Code Resources

Click to add title

- Abusing resources
- Stealing data
- Altering data



gvs

DO NOT LIKE

Resources:

- * DDOS
- * Mail forwarding for spam
- * Bot network

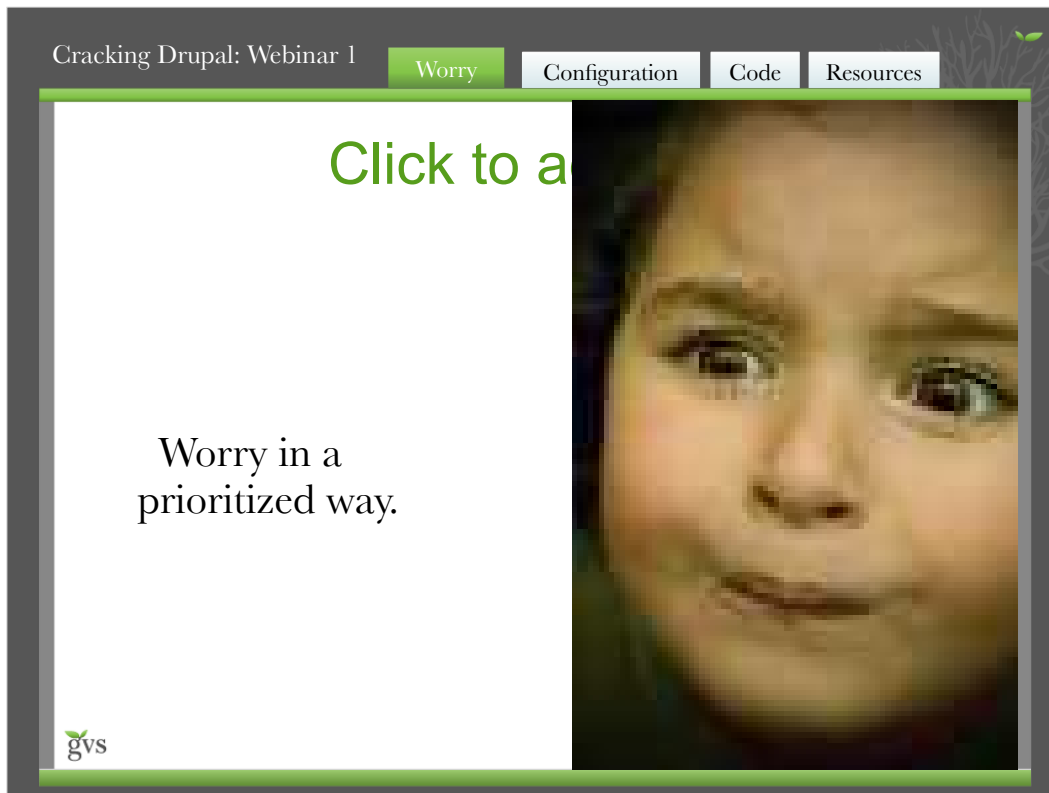
Stealing data

- * E-mails
- * Username/passwords
- * Worse? (credit card, ssn, etc.)

Altering data

- * Homepage defacement
- * Changing prices in e-commerce
- * Deleting content

<http://www.flickr.com/photos/piez/995290158/>



Being educated about security lets you know what to worry about.

<http://www.flickr.com/photos/filipamachado/3249193904/>

Choose your strategy

Stay ahead of the pack?

Protect valuable assets?

If you have a random site, you just need to be more prepared than a typical site. The worms/bots will exploit long forgotten phpnuke installations (they are out there).

If you have a site with significant value of its own, you should be more proactive in what you do.

When do attacks occur?

Table 3-1 Exploits

WORM/EXPLOIT	PATCH RELEASE DATE	WORM DATE
Sentry*	November 2001	December 2001
Code Red	June 10, 2001	July 15, 2001
SQL Slammer	July 24, 2002	January 26, 2003
Sadmind	December 1999 / October 2000	May 8, 2001

NOTE *Sentry was the worm that attacked a site of mine and that first alerted me to the need for attention to security in web applications.

Source: *Cracking Drupal* Chapter 3

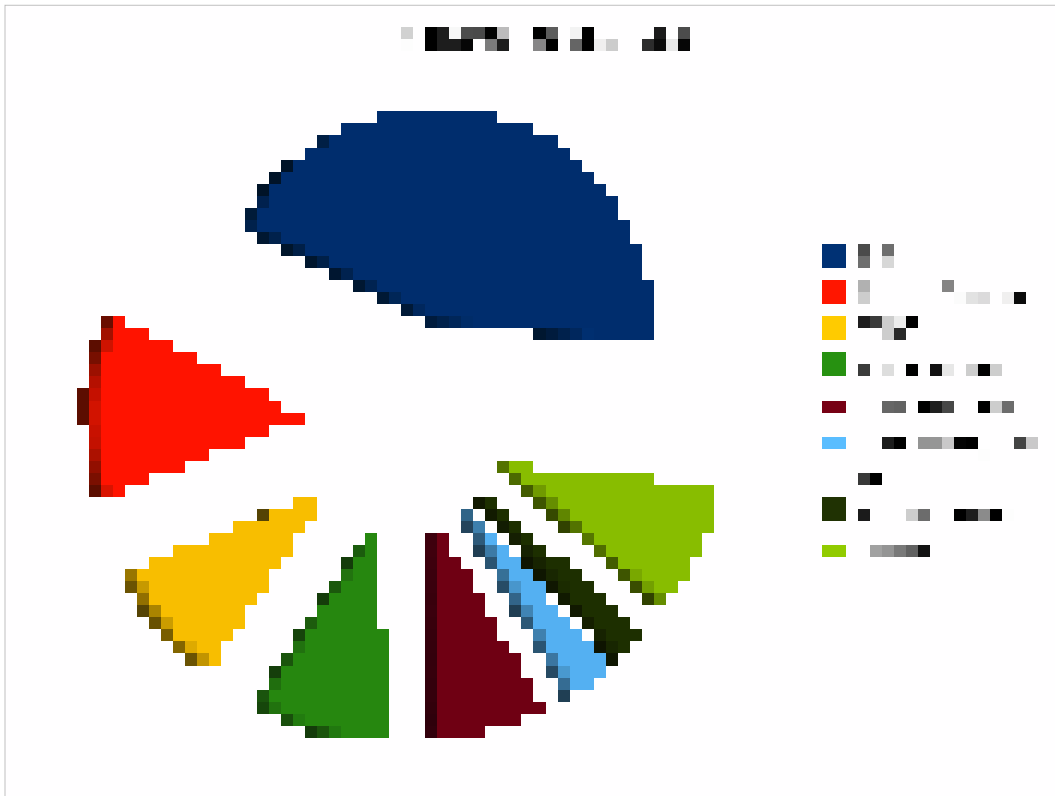
Most major attacks occur after the hole has been patched.

Seriously!

Keep up to date

- Know about releases
- Have a method to update your site
- Do it

- Update status module – can e-mail you!
 - RSS of security updates – there are three of them
 - drupal.org/security
 - Security mailing list
 - Two twitter accounts: [drupalsecurity](#) [drupal_security](#)
 - You get the point...
-
- FTP and tar.gz files
 - CVS direct from drupal.org
 - Acquia's zip file
 - svn from Acquia
 - Remote administration service from Acquia
 - Drush
 - Aegir
 - Hosted Drupal with someone
 - etc.



XSS is the worst!

Only for vulnerabilities fixed from d.o security team process, but anecdotally we know XSS is the worst!

What about real world? Most people don't want to report their weaknesses.

Cracking Drupal: Webinar 1

Worry Configuration Code Resources

Click to add title

Protect with configuration

gvs 14

Defaults are good!

Leave filtered HTML as default

Full HTML is for trusted users only

Anything you can do XSS can do (better)

```
jQuery.get(Drupal.settings.basePath + 'user/1/edit',
function (data, status) {
  if (status == 'success') {
    // Extract the token and other required data
    var matches = data.match(/id="edit-user-profile-form-form-token" value="([a-z0-9])"/);
    var token = matches[1];
    // Post the minimum amount of fields. Other fields get their default values.
    var payload = {
      "form_id": 'user_profile_form',
      "form_token": token,
      "pass[pass1]": 'hacked',
      "pass[pass2]": 'hacked'
    };
    jQuery.post(Drupal.settings.basePath + 'user/1/edit', payload);
  }
});
```

<http://crackingdrupal.com/node/8>

Javascript can do everything that you can do on the site. If you are logged in as an admin user, it can edit any users password, change permissions, etc.

Default	Name	Roles	Operations
<input checked="" type="radio"/>	Filtered HTML	All roles may use default format	configure
<input type="radio"/>	Full HTML	No roles may use this format	configure delete

Set default format

Defaults are good!

Leave filtered HTML as default

Full HTML is for trusted users only

Cracking Drupal: Webinar 1

Worry Configuration Code Resources

Allowed HTML tags:

If "Strip disallowed tags" is selected, optionally specify tags which should not be stripped. JavaScript event attributes are always stripped.

gvs 17

Defaults are good!

Careful when you tweak these

Cracking Drupal: Webinar 1

Worry Configuration Code Resources

Name	Weight
URL filter	0
HTML filter	1
Line break converter	2
HTML corrector	10

Save configuration

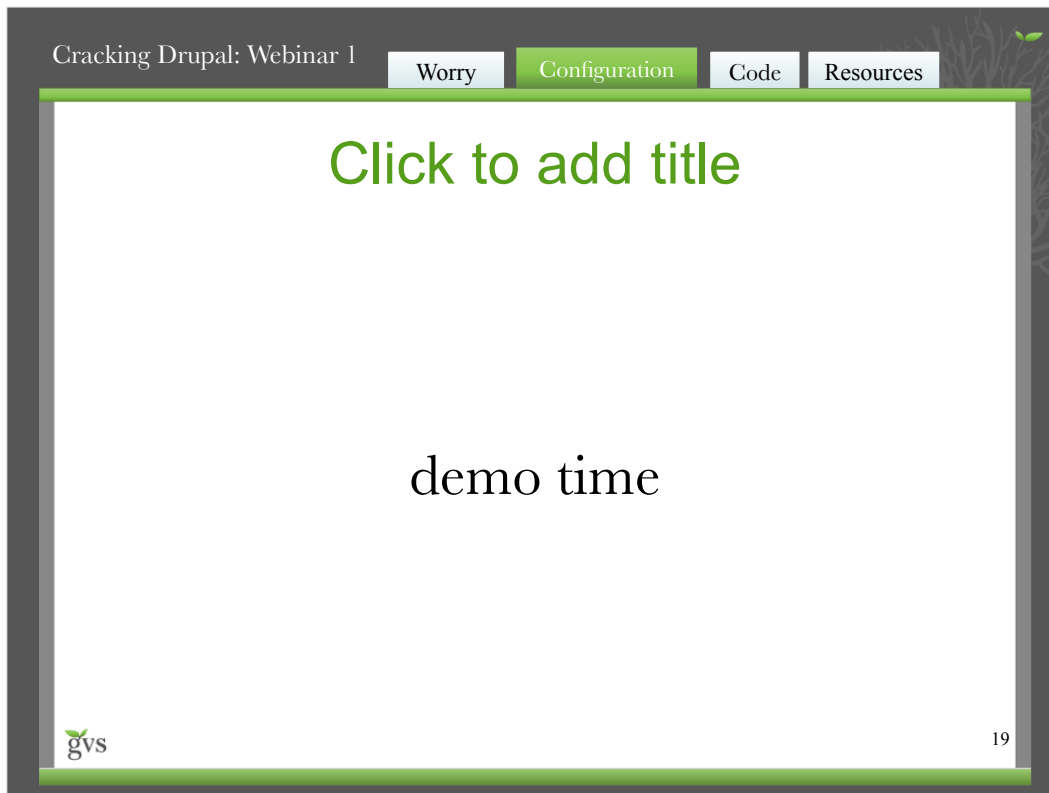
Drupal.org <http://drupal.org/node/224921>
Cracking Drupal: Chapter 3

gvs 18

Defaults are good!

HTML Filter should be after any content altering filters (i.e. markdown, embed filters, etc.)

Heavier “weight” items run later.



Tell folks to pay close attention now.

1. Demonstrate weak configuration.

Make full html default so users can post images and center align their content.

User posts javascript - bad!

Create a node that allows comments and is published to the home page.

Allow anonymous to post comments, post comments without approval, access comments, make "Full HTML" the default input format

Logout

Post a comment as anonymous like:

```
<script>
```

```
$("#h1").hide(4000, function () {
```

```
  $("#header").append("<h1>Greggles got pwned!</h1>");
```

```
});
```

```
</script>
```

Cracking Drupal: Webinar 1

Worry Configuration **Code** Resources

XSS For Themers / Coders (and reviewers)

gvs 20

If you're a themer, you should understand this

If you're a coder, you should really understand this

If you just run a site or are a manager of coders and themers, you should understand it well enough to recognize obvious problems to do QA on your site (downloading new contributed modules, for example).

Themers

- Read tpl.php and default implementations
- Rely on your module developer for variables

tpl.php and default theme_* implementations will show where to use check_plain etc.

Good developers should know when/where/how to filter text, let them worry about it and hand you simple variables via preprocess functions.

Cracking Drupal: Webinar 1

Worry Configuration **Code** Resources

Developers

Where does this text come from?
Is there a way a user can change it?
In what *context* is it being used?

gvs 22

Whenever you deal with assembling bits of text for output consider these three questions.

Answers will determine whether any filtering is required.

User agent in http request to include a file?

Cracking Drupal: Webinar 1

Worry Configuration **Code** Resources

Context

- Mail context
- Database context
- Web context
- Server context

Take an hour:
<http://acko.net/blog/safe-string-theory-for-the-web>

gvs 23

Steven Wittens wrote this up way better than anyone else.

If you don't have an hour and don't have a themer or developer...use the cheat sheet

Creator:inkscape 0.46

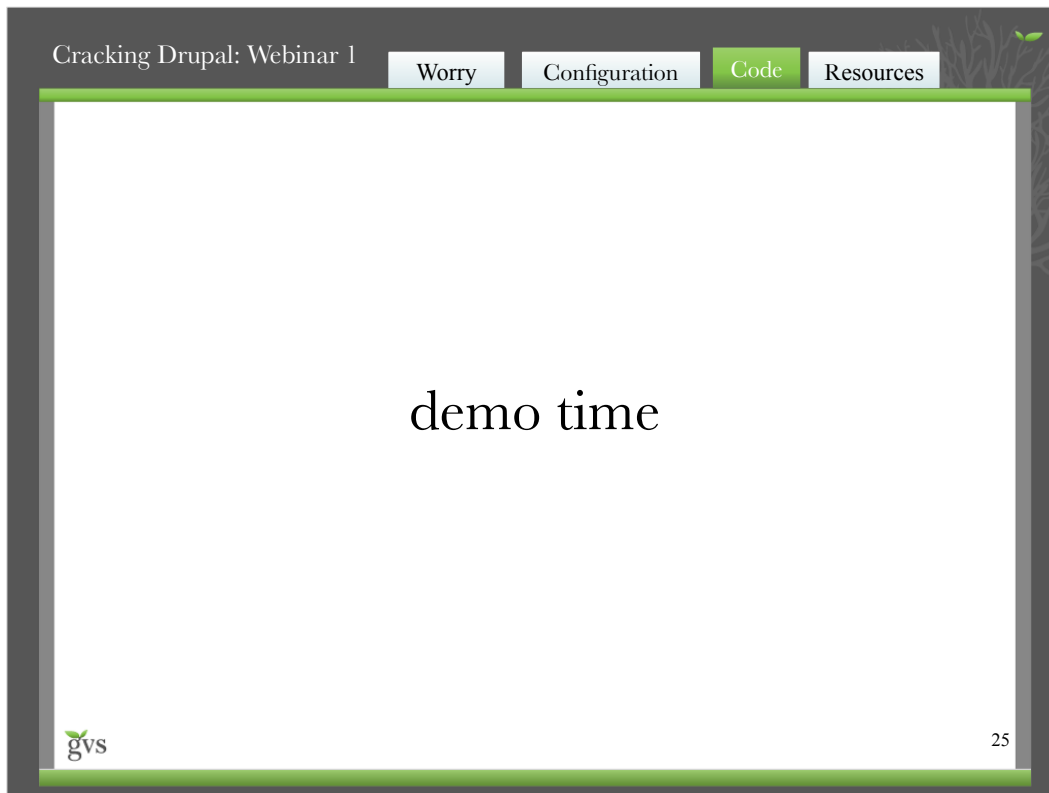
You deal with a string

Input comes in, any yes answer drops down, HTML output at the bottom.

Sometimes we use the underlying function like `check_url` via a convenience function like `l()`. Ditto `check_plain` via `t()`.

Rich text may contain html, may contain Wiki formatting.

Trusted text is way less than 1% of the text on a site.



2. Demonstrate weak code.

XSS is from “user input” - ALL user input! Including browser user agent!

SETUP NOTES:

1. Install browscap and monitor user agents
2. Setup firefox useragent switcher with a useragent like `<script>$("body").replaceWith("<h1>now what</h1>?");</script>`

Resources

- <http://drupal.org/security-team>
- <http://drupal.org/security>
- <http://drupal.org/writing-secure-code>
- <http://drupal.org/security/secure-configuration>
- <http://groups.drupal.org/node/15254> - discussion group
- <http://heine.familiedeelstra.com/>
- Cracking Drupal - <http://crackingdrupal.com>