



DRUPALCON
VIENNA



10 Ways Drupal 8 is More Secure

September 28, 2017

Peter Wolanin





The 10 Ways



1. Twig templates used for html generation
2. Removed PHP input filter
3. Site configuration exportable, manageable as code
4. User content entry and filtering improved
5. Hardened user session and session ID handling
6. Automated CSRF tokens via route definitions
7. Trusted host patterns enforced for requests
8. SQL limited to executing single statements
9. Clickjacking protection enabled by default
10. Core JavaScript API Compatible with CSP



Who Am I?

- ❖ Drupal 5, 6, 7, 8 core contributor
drupal.org/user/49851
- ❖ Drupal Security Team
- ❖ BioRAFT Engineering
- ❖ Helped implement several of the Drupal 8 features in this talk
- ❖ DrupalCamp NJ organizer

Photo by amazeelabs, by-nc-sa



<https://www.drupal.org/u/pwolanin>





Open Web Application Security Project (OWASP)

- ❖ <https://www.owasp.org/>
- ❖ Has self-study materials, best practices, and cheat sheets
- ❖ Software tools like the ZAP proxy
- ❖ “The OWASP Top Ten represents a broad consensus about what the most critical web application security flaws are.”
- ❖ Ordered by risk, not just prevalence





OWASP Top Ten (2013)

- 1. Injection (*such as SQL, shell commands, etc*)**
- 2. Broken Authentication and Session Management**
- 3. Cross-Site Scripting (XSS)**
- 4. Insecure Direct Object References**
- 5. Security Misconfiguration**
- 6. Sensitive Data Exposure**
- 7. Missing Function Level Access Control**
- 8. Cross-Site Request Forgery (CSRF)**
- 9. Using Components with Known Vulnerabilities**
- 10. Unvalidated Redirects and Forwards**





OWASP Top Ten (2013)

- ❖ Some of these are very broad - and the order does not match the frequency of Drupal vulnerabilities
- ❖ #5 Security Misconfiguration (including outdated software) - applies to every layer of your stack
- ❖ #9 Using Components with Known Vulnerabilities - applies to every layer of your stack
- ❖ Do you know if there was a vulnerability in some code you got via composer?
<https://github.com/FriendsOfPHP/security-advisories>





#1 Twig templates used for html generation

- ❖ OWASP Top Ten #3: Cross-Site Scripting (XSS)
- ❖ OWASP Top Ten #1: Injection
- ❖ Drupal 8 enables Twig auto-escaping - this feature went on a long and winding cycle of implementation (“string” as object, safe string list)
- ❖ Twig limits the scope of functionality - can’t run SQL or arbitrary PHP in a template
- ❖ Twig is also easier to read/write for people who are not PHP coders (or really, for everyone)



#1 Twig templates used for html generation

```
<div id="<?php print $block_html_id; ?>" class="<?php print $classes; ?>"<?php print $attributes; ?>>
    <?php print render($title_prefix); ?>
    <?php if ($block->subject): ?>
        <h2<?php print $title_attributes; ?>><?php print $block->subject ?></h2>
    <?php endif;?>
    <?php print render($title_suffix); ?>

    <div class="content"<?php print $content_attributes; ?>>
        <?php print $content ?>
    </div>
</div>
```



#1 Twig templates used for html generation

```
<div{{ attributes }}>
  {{ title_prefix }}
  {% if label %}
    <h2{{ title_attributes }}>{{ label }}</h2>
  {% endif %}
  {{ title_suffix }}
  {% block content %}
    {{ content }}
  {% endblock %}
</div>
```



#1 Twig templates used for html generation

- theme() functions deprecated and will be removed in Drupal 9 - don't add new ones.
- Theme autoescape helper function added:
`theme_render_and_autoescape()`

```
<div <?php print $attributes; ?>>
  <h2>
    <?php print $node->title; ?>
  </h2>
</div>
```

```
<article{{ attributes }}>
  <h2>
    {{ node.label }}
  </h2>
</article>
```





#2 Removed PHP input filter and the use of PHP as a configuration import format

- ❖ OWASP Top Ten #1: Injection (*SQL, PHP, etc*)
- ❖ In Drupal 7 getting access to an admin Drupal login is trivially escalated to total control of the site and a server shell
- ❖ For Drupal 7, importing something like a View required importing executable PHP code





#2 Removed PHP input filter and the use of PHP as a configuration import format

- ❖ Other areas where PHP snippets might have been used in Drupal 7 including block visibility, field defaults, etc. have been removed
- ❖ If you need special logic - put it in a module file in git where you can track it!





#3 Site configuration exportable, manageable as code, and versionable

- ❖ The Configuration Management Initiative (CMI)
- ❖ Exported YAML files can be managed together with your code in git
- ❖ Auditable history of configuration changes
- ❖ Diff your active config to what's in the codebase



Single export ☆

Synchronize Import **Export**

Full archive **Single item**

[Home](#) » [Administration](#) » [Configuration](#) » [Development](#) » [Synchronize](#)

Choose a configuration item to display its YAML structure.

Configuration type

Simple configuration ▼

Configuration name

user.settings ▼

Here is your configuration:

```
anonymous: Anonymous
verify_mail: true
notify:
  cancel_confirm: true
  password_reset: true
  status_activated: true
  status_blocked: false
  status_canceled: false
  register_admin_created: true
  register_no_approval_required: true
  register_pending_approval: true
register: visitors_admin_approval
cancel_method: user_cancel_block
password_reset_timeout: 86400
password_strength: true
langcode: en
```





#3 Site configuration exportable, manageable as code, and versionable

- ❖ Contributed module allows locking production configuration
drupal.org/project/config_readonly
- ❖ You can also hook into the configuration system to log each change



#4 Filtering Text





#4 User content entry and filtering improved


- ❖ OWASP Top Ten #3: Cross-Site Scripting (XSS)
- ❖ Integration of the editor configuration and the text filter configuration reduces the inclination to grant full HTML access
- ❖ You know full HTML is the same as the ability to hijack your whole site via XSS, right?



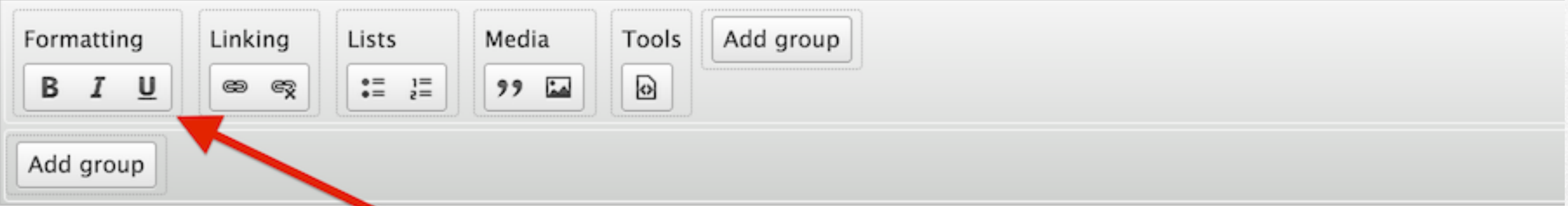
Drag a new button from the available to enabled section in the editor configuration:

TOOLBAR CONFIGURATION
Move a button into the *Active toolbar* to enable it, or into the list of *Available buttons* to disable it. Buttons may be moved with the mouse or keyboard removed upon save.

Available buttons



Active toolbar



The corresponding HTML tag (the U tag) is added to the allowed list:

Filter settings

Limit allowed HTML tags
Enabled

Allowed HTML tags
> <dd> <h4> <h5> <h6> <p>
 <u>

A list of HTML tags that can be used. JavaScript event attributes, JavaScript URLs, and CSS are always stripped.

Based on the text editor configuration, these tags have automatically been added: <u>.

Display basic HTML help in long filter tips

Add rel="nofollow" to all links





#4 User content entry and filtering improved

- ❖ Core text filtering supports limiting users to using only images local to the site
- ❖ Attribute filtering was a late, but important, addition (Issue #2549077, Sep 30, 2015)
- ❖ Attribute filtering is important since it allows you to block various appearance tricks (e.g. SPAM text with a class making it invisible) and ajax hijacking - we blocked some of that in SA-CORE-2015-003





#5 Hardened user session and session ID handling

- ❖ OWASP Top Ten #2: Broken Authentication and Session Management
- ❖ Hashed session IDs in database
- ❖ Mixed-mode session support removed
- ❖ Leading “www.” is no longer stripped from the session cookie domain







#5 Hardened user session and session ID handling

- ❖ Drupal 7: a stolen session ID (sid or ssid) from a database dump can be used to hijack a session
- ❖ Drupal 8: this can't happen (using core DB session handling)

<https://stackoverflow.com/questions/549/the-definitive-guide-to-form-based-website-authentication>

<https://utcc.utoronto.ca/~cks/space/blog/web/HashYourSessionIDs>





```
--  
-- Dumping data for table `sessions`  
--
```


```
LOCK TABLES `sessions` WRITE;  
/*!40000 ALTER TABLE `sessions` DISABLE KEYS */;
```

```
INSERT INTO `sessions`  
VALUES  
(1, 'lNeHVJs6XmKq0vew4gizoAo-_B18LA-1G_EcABK8KaI',  
'', '127.0.0.1', 1466174035, 0, '');
```

```
INSERT INTO `sessions`  
VALUES  
(130, 'PdV0vPyj0hOahcTq3eJQOZ1WBA-0n8BZVsxBYwbkMgE',  
'', '127.0.0.1', 1466174490, 0, '');
```



← → ↻ drupal-7.dd:8083

 drupal-7.local

Home

🔍

Clone of MMMMM asdasd Article
Submitted by peter on Tue, 2015-11-17 15:20
asdasdas
Tags: [sdfdf](#) [sdsdfsdf](#) [kkk/asdasd](#)
[Read more](#) [Log in or register](#) to post comments

Navigation

- [Donate Immediately via iframe](#)
- [Event attendees](#)
- ▶ [Not the original of Not promoted asdasd Article](#)

User login

Username *

Password *

MMMMM asdasd Article
Submitted by peter on Tue, 2015-11-17 15:19
asdasdas
Tags: [sdfdf](#) [sdsdfsdf](#) [kkk/asdasd](#)
[Read more](#) [Log in or register](#) to post comments

🏠 📄 | Elements Console Sources Network Timeline Profiles Resources Security Audits | ⋮ ✕

🚫 🗑 top ▼ Preserve log

```
> document.cookie="SESS23e0a551c47e1621e7ef70e8b29cd83c=PdV0vPyj0h0ahcTq3eJQ0Z1WBA-0n8BZVsxBywbkMgE";|
```



drupal-7.dd:8083

Dashboard Content Structure Appearance People Modules Configuration Reports Help

Hello yyy Log out

Add content Find content User account shortcuts

My account Log out

drupal-7.local

Home

Clone of MMMMM asdasd Article

Submitted by peter on Tue, 2015-11-17 15:20

asdasdas

Tags:

sdfdf sdsdfsdf kkk/asdasd

[Read more](#)

Navigation

- ▶ Add content
- Donate Immediately via iframe
- Event attendees
- ▶ Not the original of Not promoted asdasd Article

MMMMM asdasd Article

Submitted by peter on Tue, 2015-11-17 15:19

asdasdas

drupal-7.dd:8083/user

Resources									
	Name	Value	Do...	Path	Expires /...	...	H...	Sec...	Sa...
▶ Frames	Drupal.toolbar.collaps...	0	dru...	/	2116-05-...	...			
▶ Web SQL	SESS23e0a551c47e1...	PdV0vPyj0hOahcTq3eJQOZ1W...	dru...	/	Session	...			
▶ IndexedDB	has_js	1	dru...	/	Session	7			
▶ Local Storage									





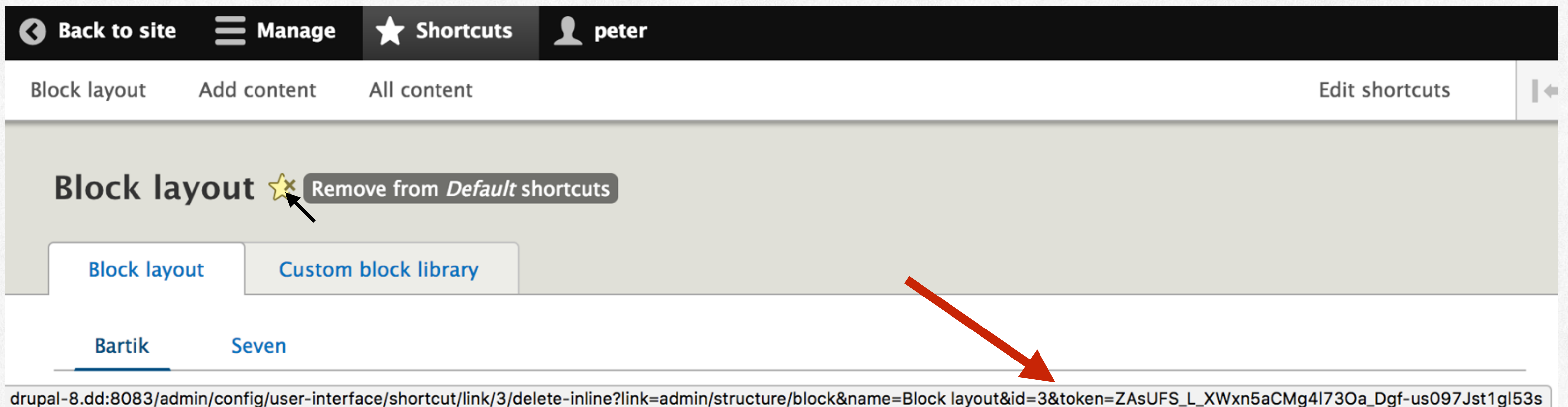
#6 Automated CSRF token protection in route definitions

- ❖ OWASP Top Ten #8: Cross-Site Request Forgery (CSRF)
- ❖ Very common Drupal vulnerability - a menu callback (route) does an action like unpublish or delete or comment approval on GET
- ❖ Drupal 7 required custom code to add and validate a token - Drupal 8 makes it easy




#6 Automated CSRF token protection in route definitions

```
entity.shortcut.link_delete_inline:  
  path: '/admin/config/user-interface/shortcut/link/{shortcut}/delete-inline'  
  defaults:  
    _controller: 'Drupal\shortcut\Controller\ShortcutController::deleteShortcutLinkInline'  
  requirements:  
    _entity_access: 'shortcut.delete'  
    _csrf_token: 'TRUE'
```



Back to site Manage Shortcuts peter

Block layout Add content All content Edit shortcuts

Block layout  Remove from *Default* shortcuts

Block layout Custom block library

Bartik Seven

drupal-8.dd:8083/admin/config/user-interface/shortcut/link/3/delete-inline?link=admin/structure/block&name=Block layout&id=3&token=ZAsUFS_L_XWxn5aCMg4l73Oa_Dgf-us097Jst1gl53s



#6 Automated CSRF token protection in route definitions

```
entity.shortcut.link_delete_inline:  
  path: '/admin/config/user-interface/shortcut/link/{shortcut}/delete-inline'  
  defaults:  
    _controller: 'Drupal\shortcut\Controller\ShortcutController::deleteShortcutLinkInline'  
  requirements:  
    _entity_access: 'shortcut.delete'  
    _csrf_token: 'TRUE'
```

Back to site Manage Shortcuts peter

Block layout Add content All content Edit shortcuts

Block layout Remove from *Default* shortcuts

Block layout Custom block library

Bartik Seven

drupal-8.dd:8083/admin/config/user-interface/shortcut/link/3/delete-inline?link=admin/structure/block&name=Block layout&id=3&token=ZAsUFS_L_XWxn5aCMg4l73Oa_Dgf-us097Jst1gl53s





#7 Trusted host patterns enforced for requests

- ❖ OWASP Top Ten #5: Security Misconfiguration
- ❖ Handbook page on host header spoofing:
drupal.org/node/1992030
- ❖ In settings.php you need to define a set of patterns and only matching hostnames are allowed when bootstrapping Drupal



#7 Trusted host patterns enforced for requests

The screenshot shows the Drupal 8 admin interface at the URL `drupal-8.dd:8083/admin/reports/status`. The user is logged in as 'peter'. The navigation menu includes 'Content', 'Structure', 'Appearance', 'Extend', 'Configuration', 'People', 'Reports', and 'Help'. The 'Reports' section is active, displaying system information:

OpenSSL/ 1.0.1n mod_php/ 7.0.4 (more information)	5.5.48-57.8-10g
Memory limit 256M	System MySQL, MariaDB, Percona Server, or equivalent

Below the system information, the 'Errors found' section displays a critical error:

✘ TRUSTED HOST SETTINGS **Not enabled**
The `trusted_host_patterns` setting is not configured in `settings.php`. This can lead to security vulnerabilities. It is **highly recommended** that you configure this. See [Protecting against HTTP HOST Header attacks](#) for more information.





#8 SQL limited to executing single statements

- ❖ OWASP Top Ten #1: Injection (*SQL, PHP, etc*)
- ❖ Drupal 6 used the PHP mysqli driver - this only allows a single statement to be sent to the DB server in each call
- ❖ Drupal 7 and 8 use PDO MySQL - this allowed unlimited statements in each call to the DB server - who knew?





#8 SQL limited to executing single statements

- ❖ Why was SA-CORE-2014-05 so bad?
- ❖ Multiple vectors accessible to anonymous users
- ❖ A single read query (e.g. looking up a username) could be converted into a read plus one or more inserts or updates - multiple SQL statements
- ❖ This means Drupal 7 on MySQL was actually a lot more vulnerable to SQL injection than Drupal 6!





#8 SQL limited to executing single statements

- ❖ PDO MySQL limited to executing single statements via PHP flag in $\geq 5.6.5$ or $5.5.21$
- ❖ Good news - that's also in $7.40+$
- ❖ Required changing PHP's PDO MySQL driver
- ❖ Delimiter checking added for all Drupal 8 SQL drivers





#8 SQL limited to executing single statements

- ❖ SQL injection is still very dangerous, however - a UNION query can be used to exfiltrate data like hashed passwords or the values of variables





#9 Clickjacking protection enabled by default

- ❖ `X-Frame-Options: SAMEORIGIN`
- ❖ Prevents the site from being served inside an iframe
- ❖ This blocks so-called click-jacking attacks
- ❖ Prevents content hijacking via iframing
- ❖ Not in OWASP Top Ten, but a favorite of independent security researchers





#10 Core JavaScript API Compatible with CSP

- ❖ OWASP Top Ten #3: Cross-Site Scripting (XSS)
- ❖ Content Security Policy v2:
<https://www.w3.org/TR/CSP2/>
- ❖ Drupal 8 JS settings added to page content as JSON, not a script that's executed
- ❖ There is no inline JS in core (not supported), so all inline JS can be blocked by CSP greatly reducing the possible XSS attack surface





#10 Core JavaScript API Compatible with CSP

- ❖ Drupal 7 settings - inline JS in the HEAD

```
<script type="text/javascript" src="http://drupal-7.dd:8083/modules/toolbar/toolbar.js?oumtxv">
</script>
<script type="text/javascript">
<!--//--><![CDATA[//><!--
jQuery.extend(Drupal.settings, {"basePath":"\/","pathPrefix":"","ajaxPageState":{"theme":"bartik",
{...} }, "urlIsAjaxTrusted":{"\/user":true}, "tableHeaderOffset":"Drupal.toolbar.height"});
//--><![ ]>
</script>
</head>
```



#10 Core JavaScript API Compatible with CSP

- ❖ Drupal 8 settings - JSON content near the end of the body

```
<script type="application/json" data-drupal-selector="drupal-settings-json">{"path":  
{"baseUrl":"\/","scriptPath":null,"pathPrefix":"","currentPath":"user\/1","currentPathIsAdmin":false,  
"isFront":false,"currentLanguage":"en"}, {...} ,"user":  
{"uid":"1","permissionsHash":"00553a620ea00b34d24cc527b46085135f27161dc58389eeaeebdc17c2245e7f"}}  
</script>  
<script src="/sites/drupal-8.dd/files/js/js_BKcMdIbOMdbTdLn9dkUq3KCJfIKKo2SvKoQ1AnB8D-g.js"></script>  
  
<!--[if lte IE 9]>  
<script src="/sites/drupal-8.dd/files/js/js_VhqXmo4azheUjYC30rijnR_Dddo0WjWkF27k5gTL8S4.js"></script>  
<![endif]-->  
<script src="/sites/drupal-8.dd/files/js/js_oXzIkyIiCKTCMCHwNQZx12hxXkaMyZFZxJRXXwm_vxY.js"></script>  
  
</body>  
</html>
```





General Take-aways for PHP Devs

- ❖ Study the OWASP Top 10 in the PHP context
- ❖ Limit SQLi damage with mysql single statements
- ❖ If using Twig, enable auto-escaping
- ❖ Don't store raw session IDs in the database/files
- ❖ Enable CSP and block inline JS
- ❖ Use web server or PHP to limit allowed hostnames
- ❖ Always beware user input!



Final Thoughts

- ❖ Drupal 8 is more secure than Drupal 7 and many of the security features actually enhance DX or user experience
- ❖ Drupal 8 does have possible new risks due to the inclusion of 3rd party libraries
- ❖ Extensive refactoring of code to a more OO style and to use new APIs may also have introduced bugs with security impact

This presentation is © 2017

This is a derivative work of a presentation I gave at Drupal North 2016

Licensed:

<http://creativecommons.org/licenses/by-nc-sa/2.0/>





JOIN US FOR
CONTRIBUTION SPRINT
Friday, September 29, 2017

Mentored
Core Spint
9:00-12:00
Room:
Stolz 2

First time
Sprinter Workshop
9:00-12:00
Room:
Lehgar 1 - Lehar 2

General sprint
9:00-12:00
Room:
Mall

#drupalsprints





WHAT DID YOU THINK?

Please evaluate this session:

<https://events.drupal.org/node/19328>

Take the DrupalCon survey!

<https://www.surveymonkey.com/r/drupalconvienna>



THANK YOU!

