No photos please
Responsible disclosure, cross-project collaboration, and Drupal 8 security
Drupal + Technology

TRACK SUPPORTED BY

platform.sh
I'm xjm

Drupal 8 release manager
Drupal Security Team member
Code & Community Strategist, Acquia

[drupal.org/u/xjm](drupal.org/u/xjm)
[@xjmdrupal](twitter.com/xjmdrupal)

"Statue" of me, from yched
What is responsible disclosure?

“...A vulnerability is disclosed only after a period of time that allows for the vulnerability to be patched.”

Wikipedia
Drupal security release windows

<table>
<thead>
<tr>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Sunday</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>29</td>
<td>30</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Modern tooling
Semantic versioning, 6-month release cycle

- **8.5.x supported release**
  - Mar 2018
  - Apr
  - May
  - Jun
  - Jul
  - Aug
  - Sep

- **8.6.x supported release**
  - Oct
  - Nov
  - Dec
  - Jan 2019
  - Feb
  - Mar
  - Apr

- **Security**
  - Mar 2018
  - Apr
  - May
  - Jun
  - Jul
  - Aug
  - Sep
  - Oct
  - Nov
  - Dec
  - Jan 2019
  - Feb
  - Mar
  - Apr
Drupal 8 coordinates releases with:

- Drupal 7
- Contributed projects
- Upstream dependencies
- Other OS projects (Backdrop, WordPress...)

Security release challenges and successes

(As illustrated by past Drupal 8 security advisories)
httpoxy & Guzzle

SA-CORE-2016-003
Drupal 8.1.7, July 2016
httpoxy & Guzzle

Fixed in Guzzle 6.2.1

- if ($proxy = getenv('HTTP_PROXY')) {
  - $defaults['proxy']['http'] = $proxy;
+ if (php_sapi_name() == 'cli' && getenv('HTTP_PROXY')) {
+   $defaults['proxy']['http'] = getenv('HTTP_PROXY');
httpoxy & Guzzle

Fixed in Guzzle 6.2.1
httpoxy & Guzzle

Fixed in Guzzle 6.2.1
PHPUnit RCE

SA-CORE-2017-001
Drupal 8.2.7, March 2017
(packaging change December 2016)
Drupal 8 will no longer include dev dependencies in release packages

Posted by xjm on 12 November 2016

As a best practice, development tools should not be deployed on production sites. Accordingly, packaged Drupal 8 stable releases will no longer contain development PHP libraries, because
PHPUnit RCE

Fixed in PHPUnit 4.8.28

```php
- eval('?>' . file_get_contents('php://input'));
+ eval('?>' . file_get_contents('php://stdin'));
```
PHPUnit RCE

CLI functionality

```php
<?php

- eval('?>' . file_get_contents('php://input'));
+ eval('?>' . file_get_contents('php://stdin'));

Compare:

- if ($proxy = getenv('HTTP_PROXY')) {
-   $defaults['proxy']['http'] = $proxy;
+ if (php_sapi_name() == 'cli' && getenv('HTTP_PROXY')) {
+   $defaults['proxy']['http'] = getenv('HTTP_PROXY');
```
PHPUnit RCE

Fixed in PHPUnit 4.8.28
jQuery 2 Ajax XSS

No Drupal 8 SA
Drupal 8.4.0, October 2017
(D7 mitigation in SA-CORE-2018-001)
jQuery 2 Ajax XSS

jQuery 3.0-alpha1 | jQuery 3.0 | jQuery 2.x EOL

2015 | 2016 | 2017 | 2018

jQuery 2 to 3 update issue

Drupal 8.0.0 | Drupal 8.4.0
CKEditor stored XSS (img alt attribute)

SA-CORE-2018-003
Drupal 8.5.2, April 2018

(Thank you mlewand and wwalc!)
REST entity vulnerability #1: Entity access bypass

SA-CORE-2017-002
Drupal 8.3.1 and 8.2.8, April 2017
REST entity vulnerability #1: Entity access bypass

```php
+ if ($operation === 'edit') {
+   if ($field_definition->getName() === $this->entityType->getKey('id')) {
+     return $return_as_object
+       ? AccessResult::forbidden('The entity ID cannot be changed')
+       : FALSE;
+   }
+   elseif ($field_definition->getName() ===
+     $this->entityType->getKey('uuid')) {
+     if ($items && ($entity = $items->getEntity()) && !$entity->isNew()) {
+       return $return_as_object
+         ? AccessResult::forbidden('The entity UUID cannot be changed')
+           ->addCacheableDependency($entity)
+         : FALSE;
+     }
+   }
+ } else if ($field_definition->getName() ===
+   $this->entityType->getKey('uuid')) {
+   if ($items && ($entity = $items->getEntity()) && !$entity->isNew()) {
+     return $return_as_object
+       ? AccessResult::forbidden('The entity UUID cannot be changed')
+         ->addCacheableDependency($entity)
+       : FALSE;
+   }
+ }
+ }
```
REST entity vulnerability #1: Entity access bypass

Drupal 8.2

Nodes vulnerable

REST entity vulnerability #1:
Entity access bypass

Drupal 8.2
Nodes vulnerable

Drupal 8.3
Users vulnerable

REST entity vulnerability #2: Missing file validation

SA-CORE-2017-003
Drupal 8.3.4, June 2017

*Note: Score shown here differs from the published SA
REST entity vulnerability #2: Missing file validation
REST entity vulnerability #2: Missing file validation

```php
+ $create_only_fields = [
+   'uri',
+   'filemime',
+   'filesize',
+ ];
+ $field_name = $field_definition->getName();
+ if ($operation === 'edit' && $items && ($entity = $items->getEntity())
+   && !$entity->isNew()
+   && in_array($field_name, $create_only_fields, TRUE)) {
+   return AccessResult::forbidden();
+ }
```
REST entity vulnerability #3: Comment approval bypass

SA-CORE-2017-004
Drupal 8.3.7, August 2017
REST entity vulnerability #3: Comment approval bypass

```php
if (is_null($this->get('status')->value)) {
    if (!Drupal::currentUser()->hasPermission('skip comment approval')) {
        $this->setPublished();
    }
}
```

```
_restSubmittedFields = {array} [5]
  _restSubmittedFields[cid] = {array} [1]
  _restSubmittedFields[uuid] = {array} [1]
  _restSubmittedFields[langcode] = {array} [1]
  _restSubmittedFields[comment_type] = {array} [1]
  _restSubmittedFields[status] = {array} [1]
    _restSubmittedFields[x-default] = {array} [1]
    _restSubmittedFields[0] = {array} [1]
    value = true
  _restSubmittedFields[pid] = {array} [1]
```
REST entity vulnerability #3: Comment approval bypass

```php
parent::preSave($storage);
- if (is_null($this->get('status')->value)) {
-   if (\Drupal::currentUser()->hasPermission('skip comment approval')) {
-     $this->setPublished();
-   } else {
-     $this->setUnpublished();
-   }
- }
- }
+ $fields['status']->setDefaultValueCallback(
+   'Drupal\comment\Entity\Comment::getDefaultStatus' +);

+ public static function getDefaultStatus() {
+   return \Drupal::currentUser()->hasPermission('skip comment approval')
+     ? CommentInterface::PUBLISHED
+     : CommentInterface::NOT_PUBLISHED;
+ }
```
Highly critical remote code execution in Drupal 7 and Drupal 8

SA-CORE-2018-002
Drupal 8.5.1, 8.4.6, 8.3.9, & 7.58
March 2018 (followup April 2018)

https://www.drupaleurope.org/session/autopsy-vulnerabilities
Highly critical remote code execution in Drupal 7 and Drupal 8

Sites on secure, tagged releases after each SA

First exploits after 2 weeks
First exploits within 5 hours
Lessons

What have we learned?
How can we improve?
Effective coordinated disclosure is hard

OH: Coordinated disclosure is like conducting an orchestra where half the musicians don't show up and the other half are playing a song they never heard before.

Also, some musicians deny they're playing instruments. And the woodwinds section is on fire.

9:03 AM - 28 Feb 2018

70 Retweets  225 Likes
We can't always set the schedule

We must avoid single points of failure. Cross-project relationships are essential.
We have to deal with BC breaks in dependency updates

jQuery3 broke stuff.
Symfony broke more.
We needed both.

https://www.thirdandgrove.com/long-road-drupal-9
We need (secure) automatic updates for security issues

This is not simple to solve.

https://www.drupal.org/initiatives/automatic-updates

New policy: Overlapping security coverage for minor versions

https://www.drupal.org/node/2909665
New vulnerabilities and attack vectors

Drupal 8 APIs are new and evolving. Vulnerabilities evolve along with them.
Become a Drupal contributor
Friday from 9am

- First-time contributor workshop
- Mentored contributions
- General contributions
Thanks to...

- mlhess
- greggles
- samuel.mortenson
- pwolanin
- David_Rothstein
- David Strauss
- dsnopek
- Wim Leers
- Josh Koenig
- Jasu_M
- mlewand and wwalc
- The Boston Drupal Group
- Drupal HackCamp
- Issue reporters
- The Drupal Security Team