DrupalCon
SEATTLE 2019
APRIL 8-12

Coordinated disclosure, cross-project collaboration, and Drupal 8 security

xjm | @xjmdrupal

No photos please

DrupalCon
SEATTLE 2019
APRIL 8-12

"Statue" of me, from yched

I'm xjm

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

drupal.org/u/xjm

@xjmdrupal

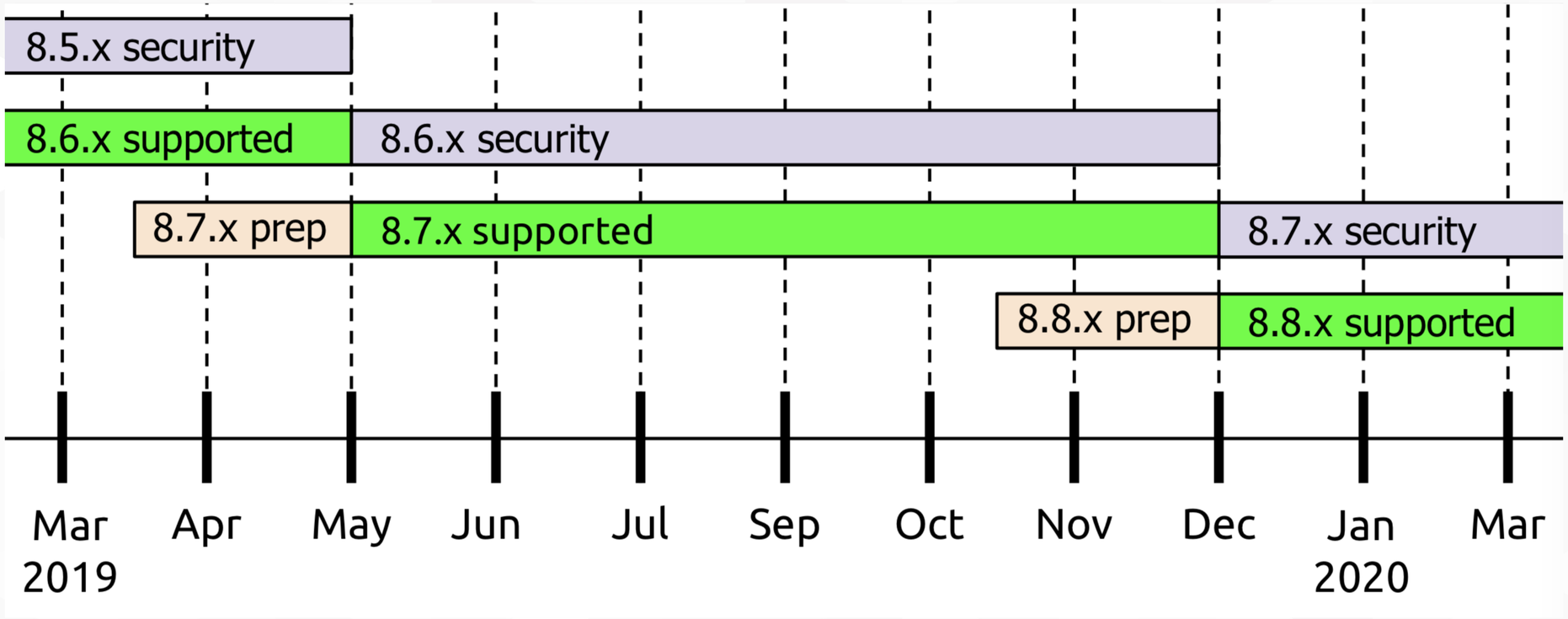
What is coordinated disclosure?

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

--Wikipedia

Modern tooling

Semantic versioning, 6-month release cycle



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)

DrupalCon
SEATTLE 2019
APRIL 8-12

httpoxy & Guzzle

SA-CORE-2016-003
Drupal 8.1.7, July 2016

https://httpoxy.org/

httpoxy & Guzzle

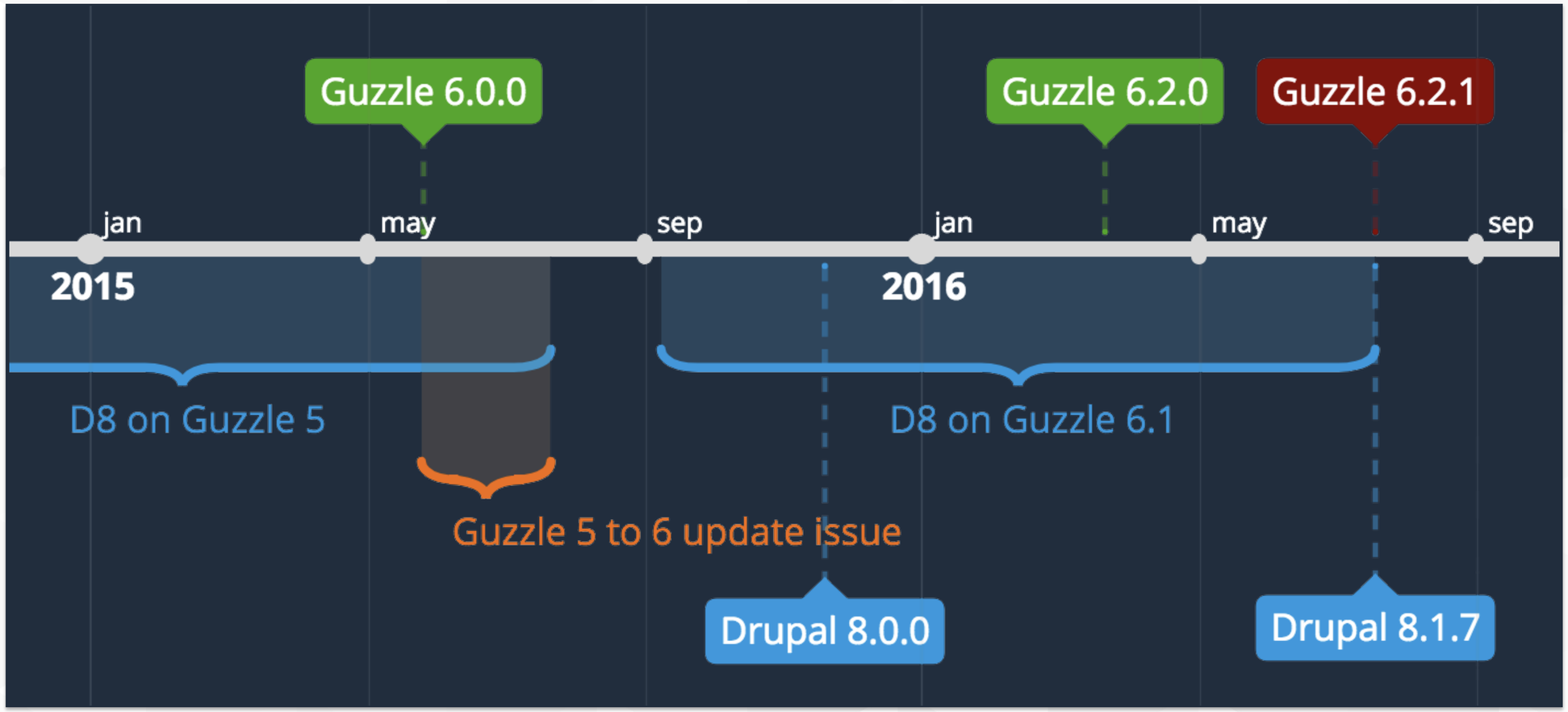
Fixed in Guzzle 6.2.1

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

https://httpoxy.org/

httpoxy & Guzzle

Fixed in Guzzle 6.2.1



httpoxy & Guzzle

Fixed in Guzzle 6.2.1

PHPUnit remote code execution

SA-CORE-2017-001
Drupal 8.2.7, March 2017
(packaging change December 2016)

DrupalCon
SEATTLE 2019
APRIL 8-12

PHPUnit remote code execution

Drupal.org packaging change

Drupal™

Home About Blog

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 remote code execution

Fixed in PHPUnit 4.8.28

```
<?php

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

PHPUnit remote code execution

CLI functionality

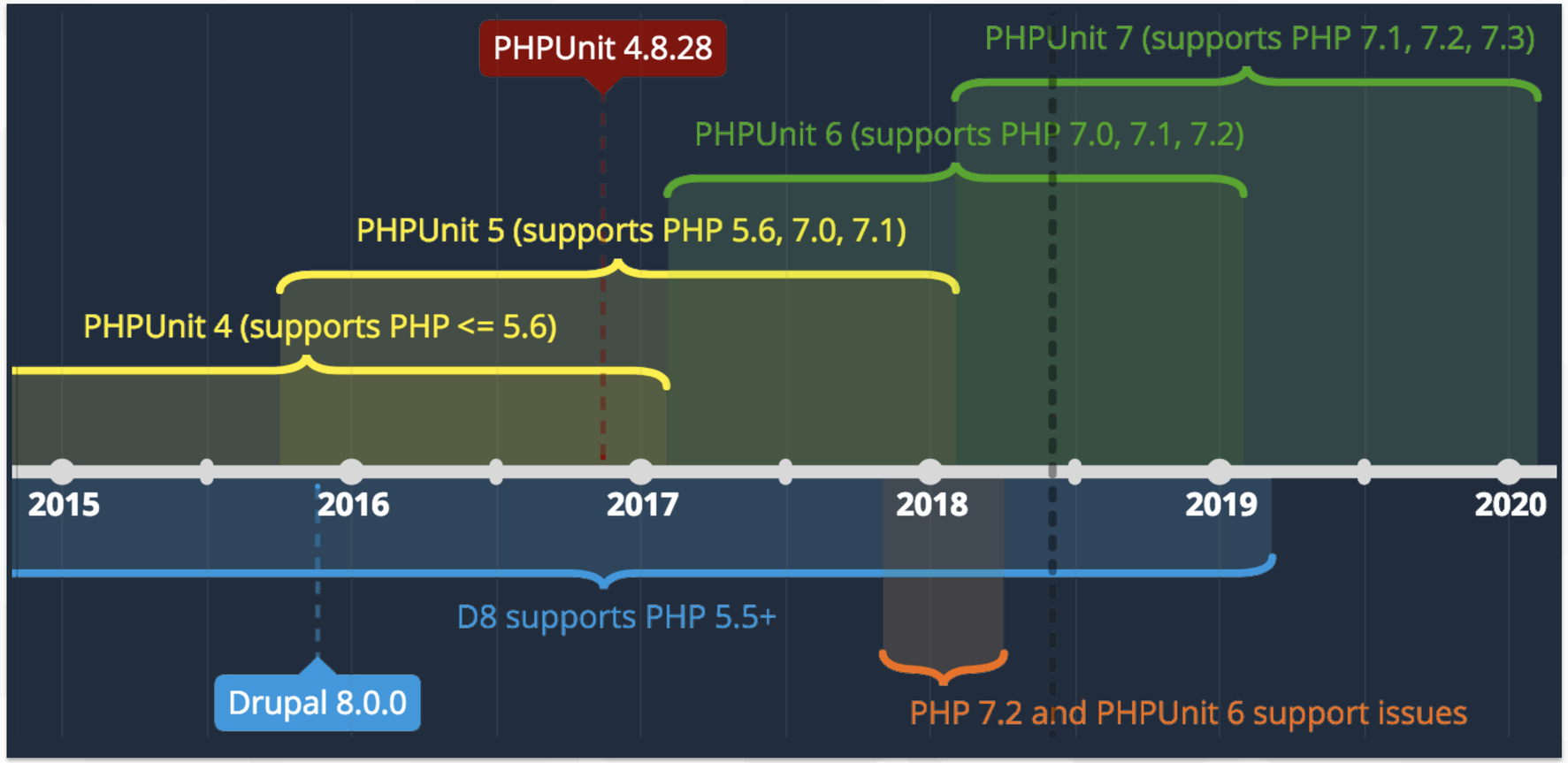
```
<?php

- 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 remote code execution

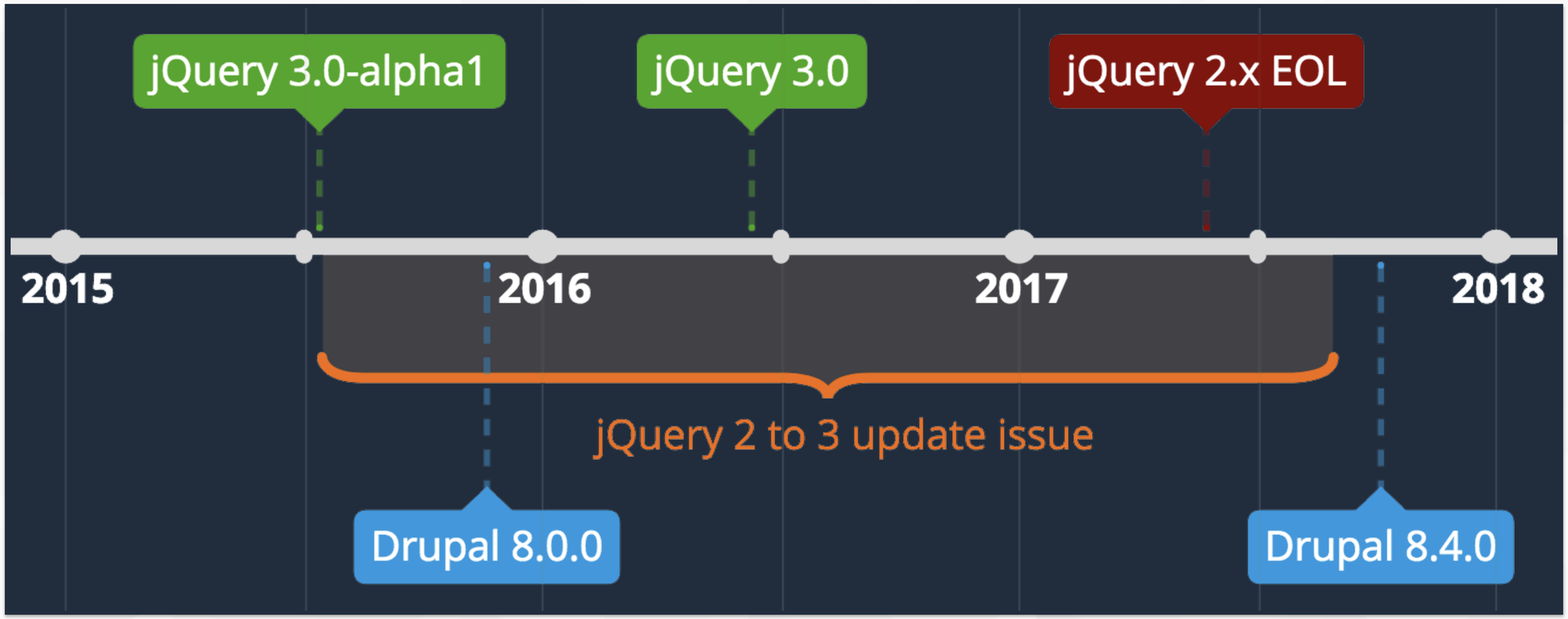
Fixed in PHPUnit 4.8.28



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



CKEditor stored XSS (img alt attribute)

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

(Thank you mlewand and wwalc!)

DrupalCon
SEATTLE 2019
APRIL 8-12

REST entity vulnerability #1: Entity access bypass

SA-CORE-2017-002
Drupal 8.3.1 and 8.2.8, April 2017

http://bit.ly/sam-2017-002

REST entity vulnerability #1: Entity access bypass

```
+ 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;
+     }
+   }
+ }
```

REST entity vulnerability #1: Entity access bypass

Drupal 8.2

Nodes vulnerable

http://bit.ly/sam-2017-002

REST entity vulnerability #1: Entity access bypass

Drupal 8.2

Nodes vulnerable

Drupal 8.3

Users vulnerable

http://bit.ly/sam-2017-002

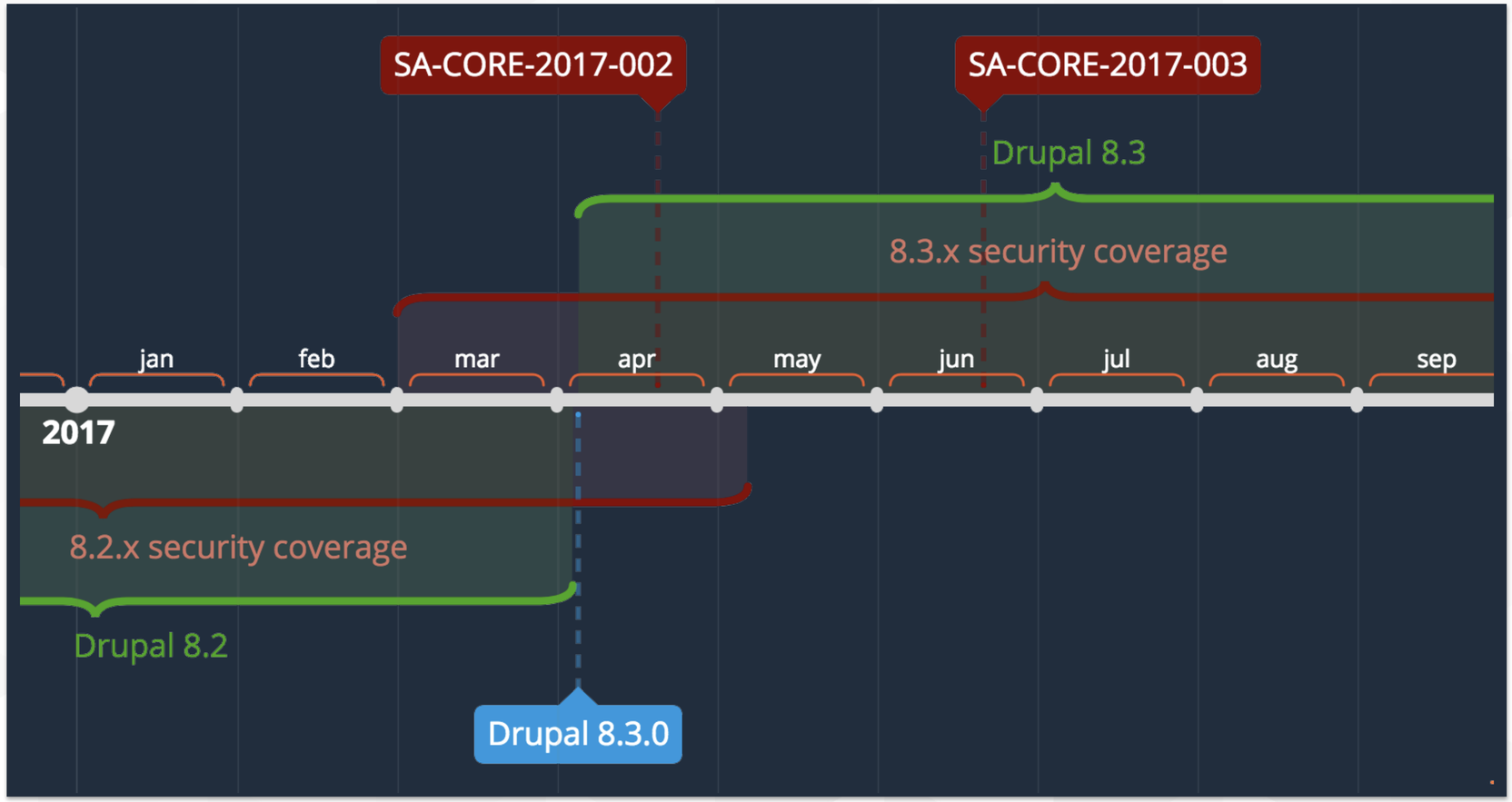
DrupalCon
SEATTLE 2019
APRIL 8-12

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

```
+ $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();
+ }
```

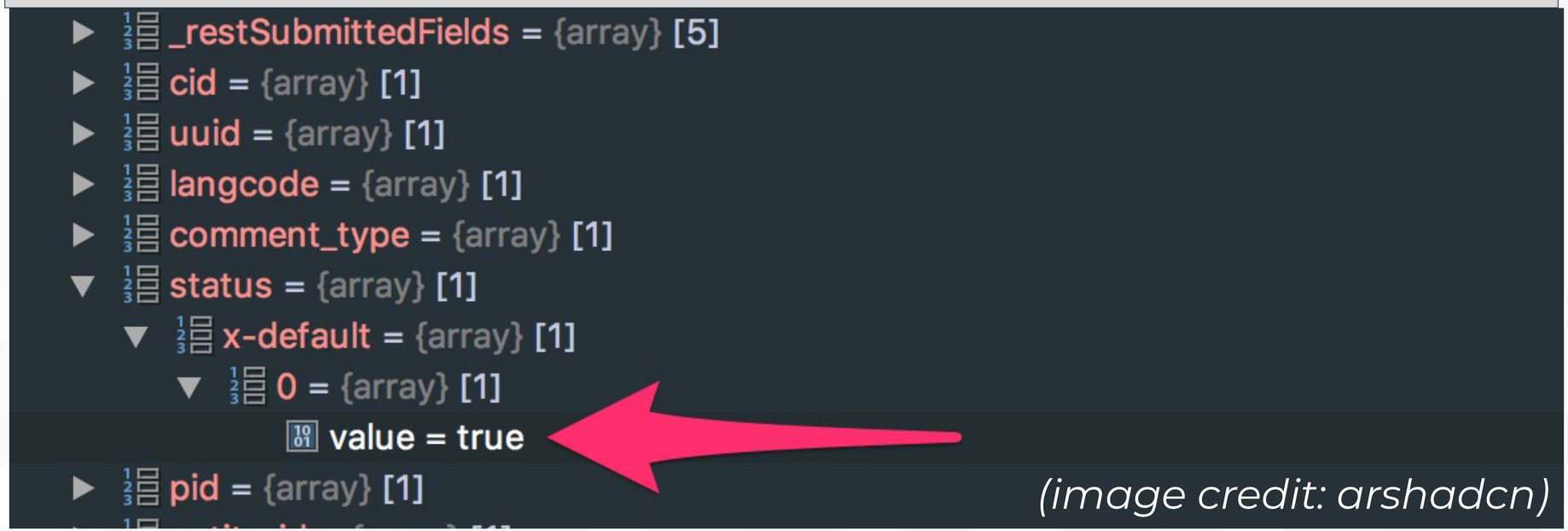
DrupalCon
SEATTLE 2019
APRIL 8-12

REST entity vulnerability #3: Comment approval bypass

SA-CORE-2017-004
Drupal 8.3.7, August 2017

REST entity vulnerability #3: Comment approval bypass

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



(image credit: arshadcn)

REST entity vulnerability #3: Comment approval bypass

```
   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;
+  }
```

DrupalCon
SEATTLE 2019
APRIL 8-12

REST entity vulnerability #4: Remote code execution

"Potatobug"
SA-CORE-2019-003
Drupal 8.6.10 and 8.5.11, February 2019

REST entity vulnerability #4: Remote code execution

```
+ $this->checkForSerializedStrings($data, $class, $field_item);
```

```
  if (is_string($values['options'])) {
-   $values['options'] = unserialize($values['options']);
+   if (version_compare(PHP_VERSION, '7.0.0', '>=')) {
+     $values['options'] =
+       unserialize($values['options'], ['allowed_classes' => FALSE]);
+   }
+   else {
+     $values['options'] = unserialize($values['options']);
+   }
  }
```

https://paper.seebug.org/821/
https://www.ambionics.io/blog/drupal8-rce

REST entity vulnerability #4: Remote code execution

Coordinated releases

Drupal 8

- JSON:API
- Translation Management Tool
- Paragraphs
- Metatag
- Font Awesome Icons
- Video

Drupal 7

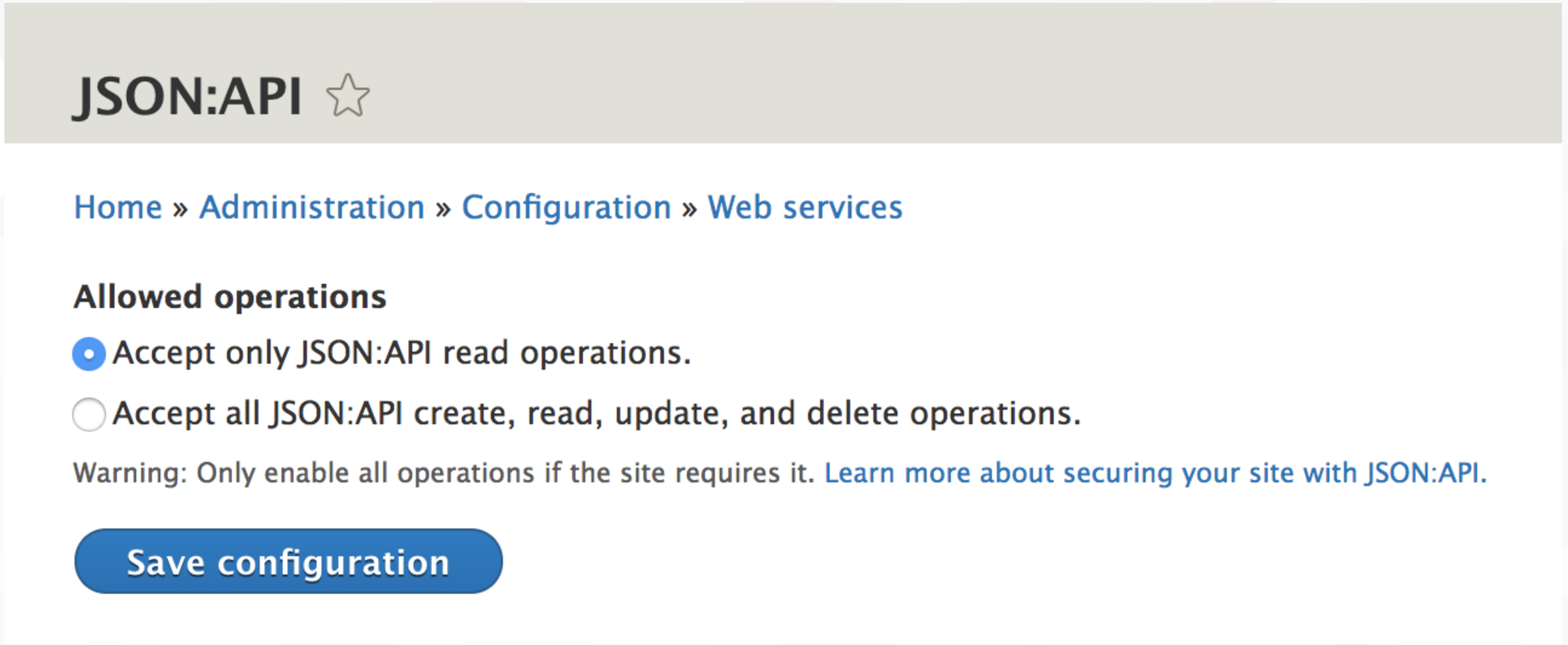
- RestWS
- RESTful *(released later)*
- Link

REST entity vulnerability #4: Remote code execution

```
rest.resource.entity.node.yml:

  methods:
    - GET
    - POST
    - PATCH
    - DELETE
  formats:
    - hal_json
  authentication:
    - basic_auth
```

https://paper.seebug.org/821/
https://www.ambionics.io/blog/drupal8-rce

REST entity vulnerability #4: Remote code execution

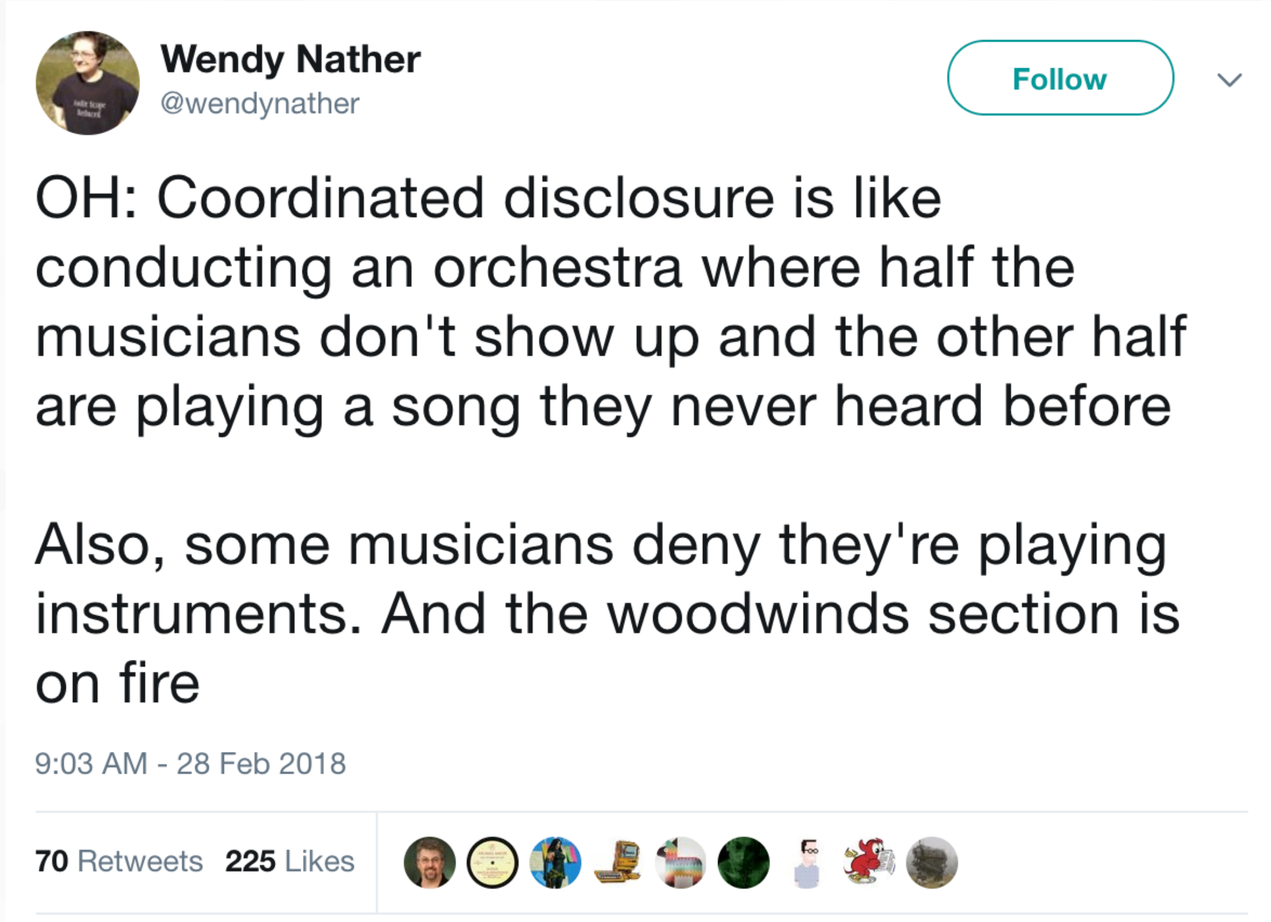


https://www.drupal.org/project/jsonapi_extras

Lessons learned

What have we learned?
How can we improve?

Effective coordinated disclosure is *hard*



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.

Dependency release schedules are important

Drupal 8 will be end-of-life in November 2021,
to avoid another Symfony BC break.

Drupal 9 will be released in 2020 with Symfony 4 (or 5) support.

We need (secure) automatic updates for security issues

This is not simple to solve.

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

http://bit.ly/hacking-wordpress-autoupdate

New vulnerabilities and attack vectors

Drupal 8 APIs are new and evolving.
Vulnerabilities evolve along with them.