www.phparch.com

April 2016
VOLUME 15 - ISSUE 4

php[architect]

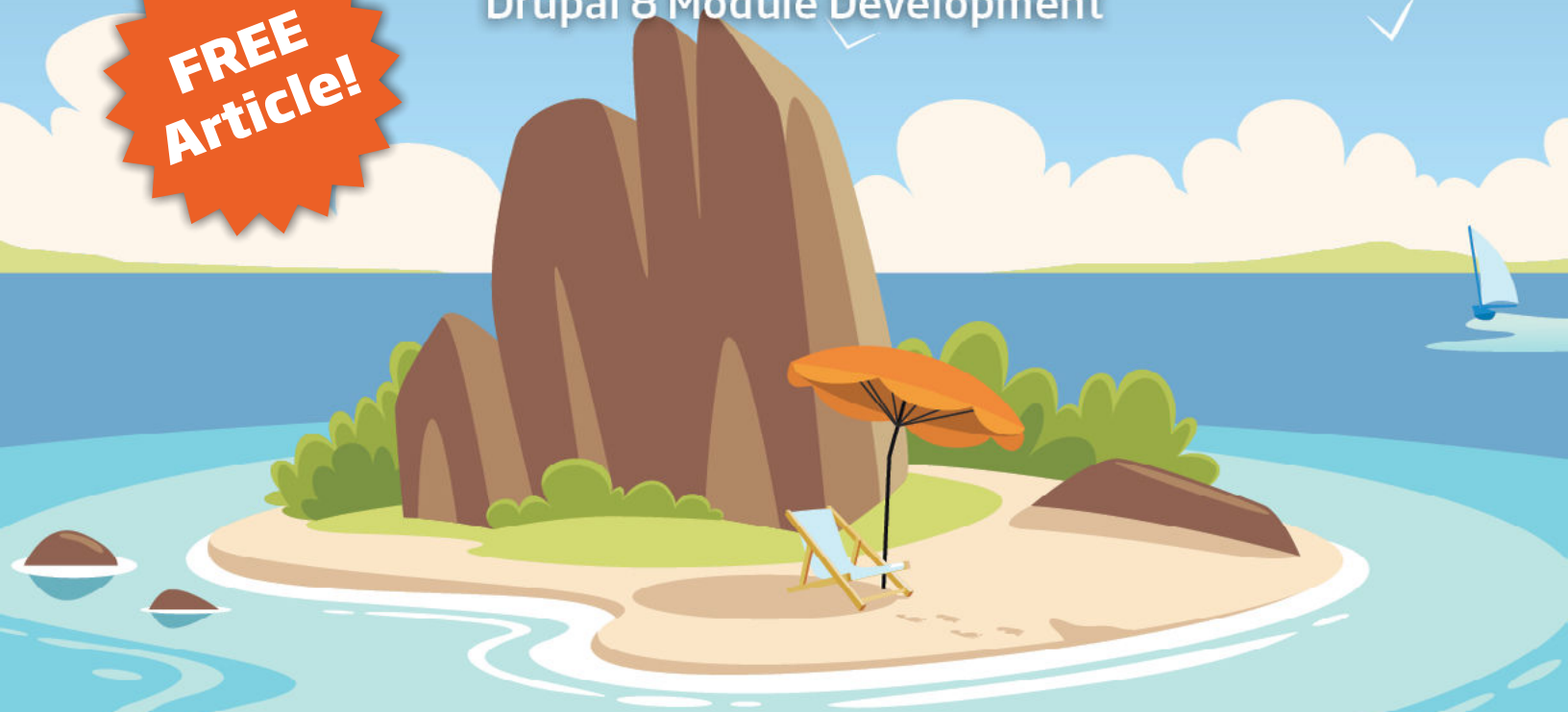
Off the Island: Drupal

Keep Your Drupal Composed

Drupal Security: How Open Source Strengths Manage Software

Drupal 8 Module Development

FREE Article!



ALSO INSIDE

Living Documentation:
Generating Documentation from Source Code

Community Corner:
Community Leaders

Security Corner:
Securing Legacy Applications—Part 2

Education Station:
PhpStorm Intentions for Improved Code Quality

Leveling Up:
Learning a New Framework

finally{}:
Performance vs Scalability

Drupal Security: How Open Source Strengths Manage Software Vulnerabilities

Cathy Theys

It is a frequent topic of discussion whether open-source software (Drupal is under GPL) is not secure because it is open source. Some people worry that if a source is "open," publicly available, and accessible, malicious hackers can find vulnerabilities to exploit. Some think private or closed-source applications would prevent these threats. In this article, I'll review the actions the Drupal project has taken to improve security and handle vulnerabilities.

Any software—whether open- or closed-source—is at risk of cyber threats, just in different ways. However, the collaborative open-source aspect makes software stronger, more defensive, and able to react to any potential issues.

Drupal 8

I have been heavily involved in Drupal Core development for years and have seen progress in making smart defaults in Drupal 8 to make it more secure, and fixes that increase security hardening.

Compared to Drupal 7, Drupal 8 had a significant amount of code refactoring, and included third-party components. In addition to our usual security efforts, the Drupal 8 pre-release Security Bug Bounty program[1] was launched starting June 2015 to crowdsource the discovery of security bugs. Previous Drupal contributors and people new to Drupal participated. Drupal 8 was released in November 2015.

Keeping a Drupal Site Secure

There are many community procedures in place to help Drupal keep pace with security. For site maintainers, the best practical security advice can be found in Greg Knaddison's (greggles) book: *Cracking Drupal*. Another good resource is the handbook pages on Drupal.org[2].

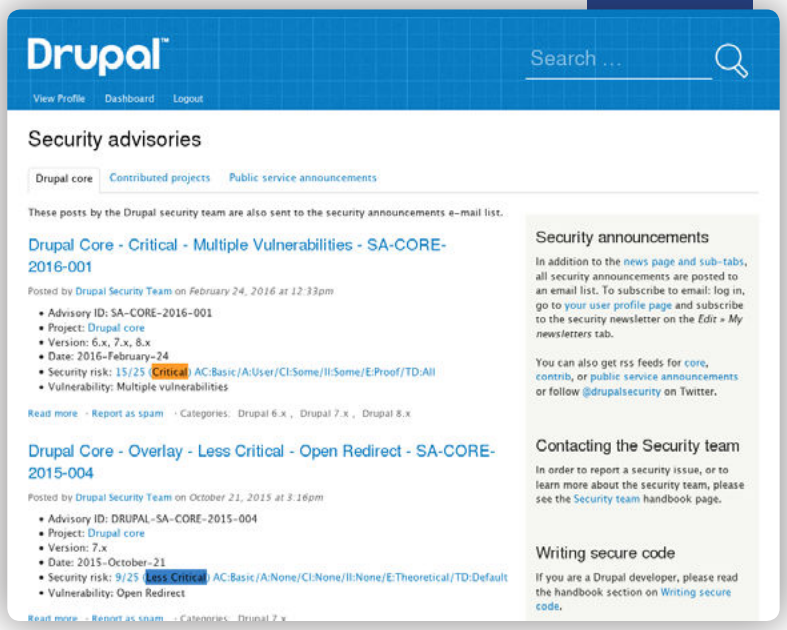
The most important advice is to keep software up to date—both Drupal and your server. *Cracking Drupal* goes into greater depth about common vulnerabilities in custom code, while the book *Drupal Security Best Practices* wisely advises you to write as little custom code as possible. If you do have custom modules or themes, the most common (and very serious) vulnerability is known as Cross-site Scripting (XSS). The most common manifestation of XSS is when user input (such as the title of a piece of content) is printed to the screen without HTML tags being escaped. This could allow a site user to inject JavaScript that would be executed by other visitors of the page. Since JavaScript can cause your browser to take any action you have permission for (create accounts, change settings, etc.), this leads to the site being completely compromised.

Drupal Security Team

The Drupal security team has almost 40 members. I joined the security team as a provisional member July 2015, and became a full member February 2016. The team coordinates reported security issues, makes security advisories, provides assistance for contributed module maintainers in resolving security issues, coordinates with the infrastructure team to keep the drupal.org infrastructure secure, and works to prevent security problems.

To help prevent security problems the security team provides education, including

FIGURE 1



1 *Security Bug Bounty program:* https://www.drupal.org/drupal8-security-bounty

2 *Security Configuration:* https://www.drupal.org/security/secure-configuration

providing documentation on writing secure code[3] and providing documentation on securing sites via the handbook pages mentioned previously.

Drupal projects are made up of *Drupal Core* and also *contributed projects*, referred to as "contrib," that are hosted on Drupal.org. Contributed projects on Drupal.org by first-time contributors are screened for security before the author of the project gets permission to create full projects. The security team facilitates security issues for all full projects with a current 1.0 or greater supported release.

Software Vulnerabilities

All software have bugs, some of which lead to security vulnerabilities. A part of any healthy open-source project is a history of security advisories and fixes. If a project has no advisories, this could indicate security is not getting enough attention. Sometimes the difference between a dangerous security problem and a non-dangerous one is who finds it. If someone finds it and reports it so it can be fixed before it is exploited, it is a better situation. With an open-source project, many people are reporting and fixing things.

Reporting a Drupal Security Issue

The process for reporting a Drupal security issue, potential error, weakness, or threat is to keep it confidential and submit the concern to the Drupal security team[4].

Vulnerabilities are reported from a variety of sources. Sometimes they come from organizations who are performing internal Drupal security audits. Drupal contributors will also report things they notice while working on other issues or tasks for a client. Other open-source projects will sometimes publicly report a vulnerability, and someone will check to see if something similar can happen with Drupal. Other open-source projects will also privately contact the Drupal security team and coordinate security releases when they know Drupal will be affected by something they are also working on.

Sometimes someone might make a public security issue, or a comment on a public issue, if they are not aware of the policy of privately contacting the security team. In those cases, an experienced contributor might notice, unpublish the information, and notify the security team.

Handling Drupal Security Issues

Security issues created (either by going to a project page and using the link "Report a security vulnerability" or by submitting an issue[5] go into a private Drupal security team issue tracker. We gather more info, such as if it effects a current stable release of a project on Drupal.org. People are added to the issue who are not official members of the security team, such as the maintainer, if it is a contributed project. Someone then attempts to reproduce the problem. If it turns out to be an issue that does not need to stay private, a member of the team replies to the reporter and asks them to create a public issue.

Once the issue is verified to be a valid security issue, all the maintainers of the project are also added to the private issue.

The security team and the people added to the issue collaborate to make patches to address the issue. People working on the issue might run tests locally and post test results in the comments on the issue. Once the issue nears consensus, a member of the security team initiates a private full test run on the Drupal CI system and posts the complete test results on the issue.

When consensus is achieved and the test results are good, a release is scheduled, coordinated with contributed project maintainers if it affects contrib projects. And a security team member drafts a Drupal security advisory.

Security Advisory

The Drupal security advisory has an ID, which specifies the project, version, date, and risk level and contains a description of the vulnerability and factors that might mitigate it. An example is https://www.drupal.org/SA-CORE-2015-004, shown in Figure 2.

Part of making the security advisory is using the Drupal Security Risk Calculator. The risk level is calculated using these factors:

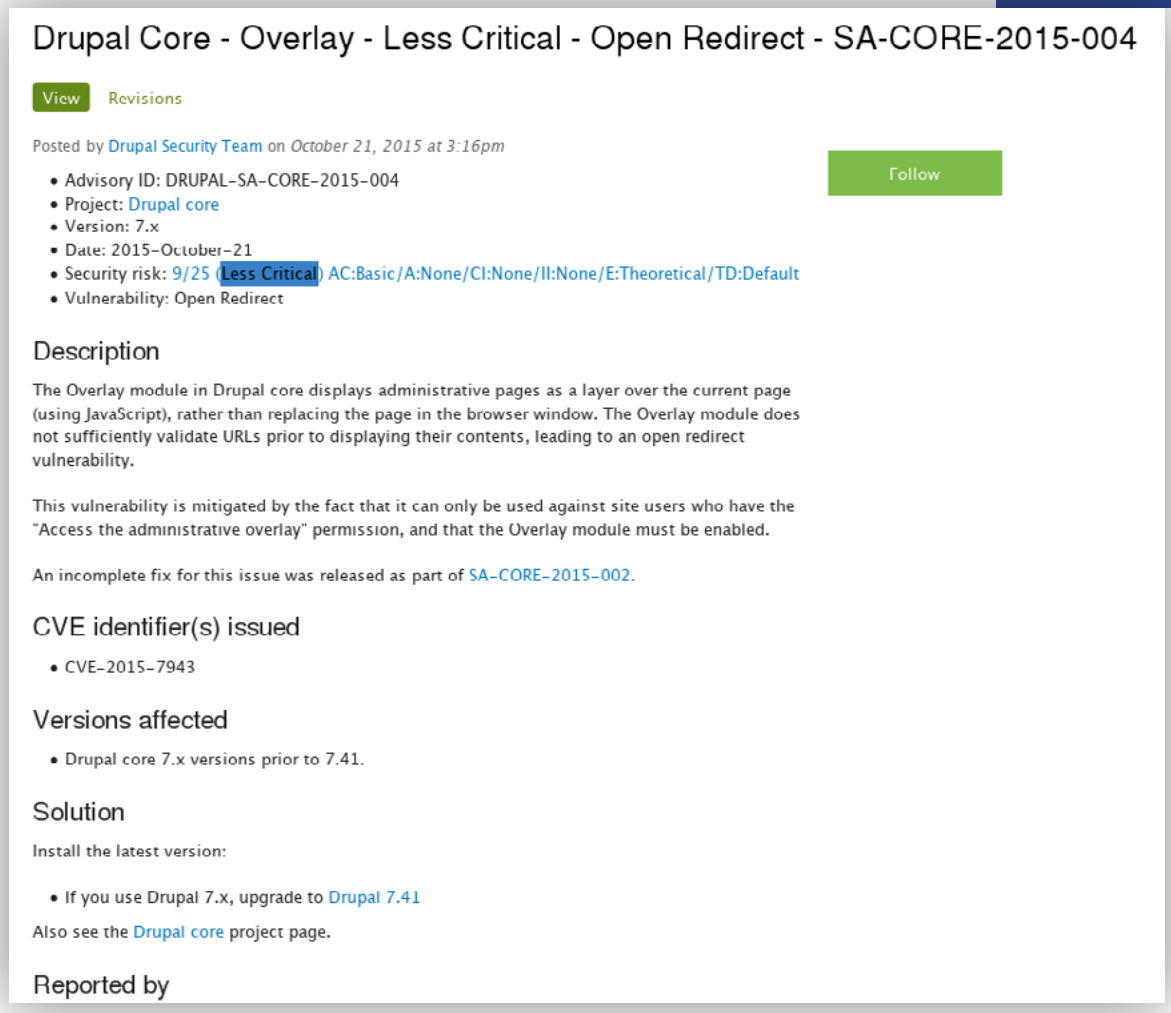
- Access complexity: How difficult is it for the attacker to leverage the vulnerability?
- Authentication: What privilege level is required for an exploit to be successful?
- Confidentiality impact: Does this vulnerability cause non-public data to be accessible?
- Integrity impact: Can this exploit allow system data (or data handled by the system) to be compromised?
- Zero-day impact: Does a known exploit exist?
- Target distribution: What percentage of module users is affected?

The answers help the team determine if the risk level is *Highly Critical*, *Critical*, *Moderately Critical*, *Less Critical*, or *Not Critical*.

The security advisory credits the original reporter and the people

3 *Writing Secure Code:* https://www.drupal.org/writing-secure-code
4 *Reporting Issues:* https://www.drupal.org/security-team/report-issue
5 *Drupal Security, Submit Issue:* https://security.drupal.org/node/add/project-issue

FIGURE 2



Drupal Core - Overlay - Less Critical - Open Redirect - SA-CORE-2015-004

View Revisions

Posted by Drupal Security Team on October 21, 2015 at 3:16pm

Follow

- Advisory ID: DRUPAL-SA-CORE-2015-004
- Project: Drupal core
- Version: 7.x
- Date: 2015-October-21
- Security risk: 9/25 (Less Critical) AC:Basic/A:None/CI:None/II:None/E:Theoretical/TD:Default
- Vulnerability: Open Redirect

Description

The Overlay module in Drupal core displays administrative pages as a layer over the current page (using JavaScript), rather than replacing the page in the browser window. The Overlay module does not sufficiently validate URLs prior to displaying their contents, leading to an open redirect vulnerability.

This vulnerability is mitigated by the fact that it can only be used against site users who have the "Access the administrative overlay" permission, and that the Overlay module must be enabled.

An incomplete fix for this issue was released as part of SA-CORE-2015-002.

CVE identifier(s) issued

- CVE-2015-7943

Versions affected

- Drupal core 7.x versions prior to 7.41.

Solution

Install the latest version:

- If you use Drupal 7.x, upgrade to Drupal 7.41

Also see the Drupal core project page.

Reported by

who reviewed and worked on the fix. On the day of the release, the fix is committed, and the security advisory is published. After an advisory is published, a CVE (Common Vulnerabilities and Exposures)[6] ID is applied for.

Core security advisories are listed on the Drupal.org security page, https://www.drupal.org/security, and security advisories for contrib projects are listed at https://www.drupal.org/security/contrib.

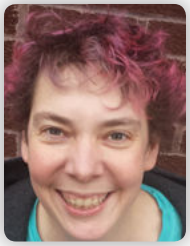
Some issues do not get security advisories. Only problems affecting stable releases get advisories. Advisories are not issued for development releases: dev, alpha, beta, RCs, or sandboxes. If an exploit requires the use of elevated permissions, then there also is no advisory. For example, if a user has to have the "administer users" permission to exploit a vulnerability, there would be no advisory, since someone with advanced permission could already take over a site. The decision to have an advisory or not is made according to the security advisory policy[7].

The Drupal Security Team Welcomes New Members

Members of the security team coordinate with security researchers who deliver reports and project maintainers responsible for fixing security issues. They follow Drupal's internal process for creating security announcements. Members provide advice to project maintainers as they work through security issues, and educate the Drupal community on security topics to improve the overall security stance of the project. Members also use their experience to identify vulnerabilities and make enhancements related to security in Drupal Core and contributed projects. The team is open to new members: https://www.drupal.org/node/1760866.

Open Source

When the source is open, more people can identify issues and privately report them so they can be fixed before they are exploited. We work together, sometimes with other projects, to make sure we handle issues in a responsible way.



Cathy Theys is the Drupal Community Liaison for BlackMesh, a FedRAMP-moderate PaaS certified managed hosting and solutions provider. In her role, Cathy works as a member of the Drupal Security team, contributes to Drupal 8 Core, participates in and presents at Drupal conferences, and organizes the Drupal Mentoring program. You can find Cathy online as @YesCT.

6 CVE: https://cve.mitre.org

7 Security Advisory Policy: https://www.drupal.org/security-advisory-policy

BLACKMESH

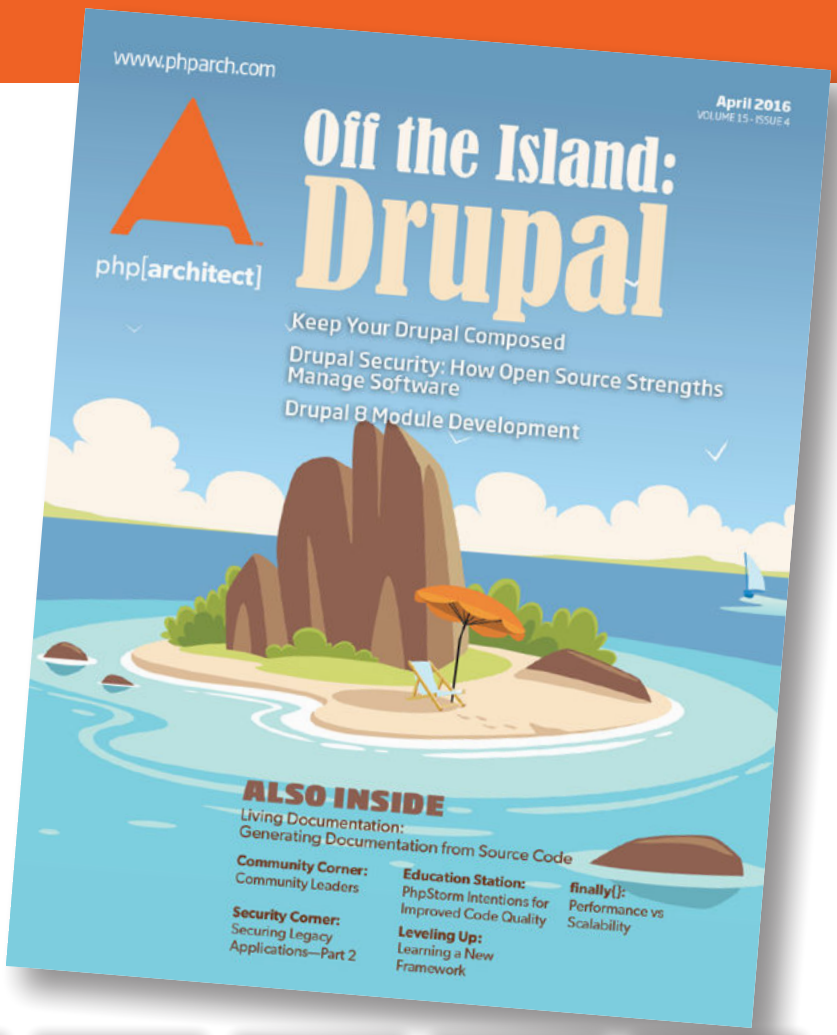
How secure is your Drupal site?

Optimizing your Drupal infrastructure with BlackMesh allows you to easily deploy and manage site content, boost application quality, and eliminate the stress of the underlying infrastructure operations.

With high performance technologies, managed integration and advanced security assessments, BlackMesh is committed to keeping your data protected.

Contact us to schedule a free solution consultation and security assessment.
888-473-0854
blackmesh.com
@blackmesh

Want more articles like this one?



Keep your skills current and stay on top of the latest PHP news and best practices by reading each new issue of php[architect], jam-packed with articles.

Learn more every month about frameworks, security, ecommerce, databases, scalability, migration, API integration, devops, cloud services, business development, content management systems, and the PHP community.

php[architect]

magazine

books

conferences

training

www.phparch.com

Get the complete issue for only $6!

We also offer digital and print+digital subscriptions starting at $49/year.