INFORMATION SECURITY GUIDELINE

Securing Drupal

Introduction

1. Drupal is a popular open source content management system and is frequently targeted for attacks; this hardening guide is meant to further enhance the level of security for Drupal by reducing the exposed attack surface by providing configuration guidance.
2. This guideline has been issued by the Chief Information Officer to supplement the Vulnerability Management standard. Compliance with this guideline is recommended, but not mandatory. Questions about this guideline may be referred to information.security@ubc.ca.

Best Practices for Protecting the Application Platform

3. Once installed, ensure that Drupal's "Update" module is installed and notification for security updates is enabled. Subscribe to Drupal's Security Advisory email list, or Twitter account.
4. Input filters & WSIWYG
    a. configure input filters for different user-levels;
    b. turn on WSIWYG Filter and configure to allow specific tags/classes/etc. only; and
    c. disallow "Full HTML" (potential for privilege-escalation attacks).
5. Logging & errors
    a. log (but don't show) errors; and
    b. set login logging ("Login Security" module, even in D7, to log incorrect logins and set better flood protection).
6. If you have comments enabled, consider installing comment moderation software like Mollom and a captcha solution like ReCaptcha.
7. Only enable modules you intend to use, leave unused modules deactivated—or uninstall and remove from your server.
    a. for each module, look at its issue queue on Drupal.org: consider the number of active installs versus the number of open and closed issues;
    b. make sure the module page says "Actively maintained"; and
    c. never enable FTP module updating; always update modules through the command line (FTP updating requires giving the webserver write access to PHP files, which is a dangerously insecure site configuration).
8. To enhance overall security it is also important to harden the operating system that will be hosting the Drupal installation:
    a. look at file permissions, make sure that the webserver can never write into a directory with executable PHP; and
    b. ensure .htaccess files exist in all Drupal files directories (sites/*/files)
9. Permissions & Roles:
    a. create roles to add privileges to users, either by group (e.g., a "faculty" role) or function (a "webform results" role) – permissions granted by roles are additive, so a mix of both is most useful;
    b. set permissions to add/delete/update content as granularly as possible; and
    c. audit roles and permissions annually

10. If you have control over the server, and it is solely for a Drupal site, consider custom rewrites in the Apache configuration (at the <VirtualHost> or <Directory> level) http://www.makina-corpus.org/blog/better-rewriterules-drupal

Recommended Sites

11. The following sites provide additional information on securing/hardening Drupal

Topic Area	Site
Securing your site	https://www.drupal.org/security/secure-configuration
Drupal Security Best Practices	http://openconcept.ca/sites/openconcept/files/drupal_security_best_practices_v1.1_-_2015-09-14.pdf
Hardening Drupal 7 Websites	https://bput4all.wordpress.com/2012/02/01/hardening-drupal-websites/
Securing Drupal 7	http://www.madirish.net/242
Security Group	https://groups.drupal.org/security
Security Advisories	https://www.drupal.org/security

Related Documents

Policy 104, Acceptable Use and Security of UBC Electronic Information and Systems

Vulnerability Management standard

UBC Systems and Application Hardening Guides guideline