Drupal 6 Website Audit

example.com

We performed an in-depth site audit in order to learn about the current state of the site (ie. how much pending maintenance needs to be performed) and to estimate the level of effort necessary to support and maintain the site going forward.

This includes:

- Checking the version of Drupal core and any contrib modules, to see if any security updates need to be performed, or if any unsupported, uncommon or high-risk modules are in use
- Looking for signs that the site has been hacked, or has any insecure configurations that would make it easier to hack
- Evaluating the complexity of the site to determine how difficult it is to make changes (without breaking anything or causing additional problems)
- Reviewing any custom modules or themes to see if they follow best practices, and determining if they'd require an additional bucket of hours in order to maintain
- Searching for anything else that might make supporting or maintaining the site harder

All of the problems identified in this report are things that we'd be happy to fix as part of the maintenance plan if you choose to work with us!

However, we include recommendations that you (or another vendor) could implement to make the site more easily maintainable, even if you don't decide to work with us.

Drupal core, contrib modules and themes

The site runs **Drupal 6.22** (released 2011-05-25). The latest is Drupal 6.37 (released 2015-08-19) - which means **the site is missing 12 security releases,** whose details are known publicly. Attackers can easily check the Drupal core version here.

There are **159 modules enabled** (much higher than average), including:

- 4 unsupported module (which means it hasn't been getting security scrutiny for some time)
    - Google Custom Search Engine 6.x-1.2
    - Flag 6.x-1.3
    - Google Analytics 6.x-3.3
    - ShareBar 6.x-100.1
- 14 modules with unapplied security updates:
    - **DBTNG 6.x-1.0-rc4 ← Drupalgeddon!!**
    - Chaos tool suite (ctools) 6.x-1.8
    - CKEditor - WYSIWYG HTML editor 6.x-1.11
    - Content Construction Kit (CCK) 6.x-2.9
    - Context 6.x-3.0
    - FileField 6.x-3.10
    - Modal Frame API 6.x-1.7
    - Nodewords: D6 Meta Tags 6.x-1.11
    - Printer, email and PDF versions 6.x-1.17
    - Quick Tabs 6.x-2.1
    - Scheduler 6.x-1.8
    - Views 6.x-2.16
    - Views Bulk Operations (VBO) 6.x-1.13
    - Webform 6.x-3.18

If possible, the unsupported modules should be updated to the supported version. It's possible there are security vulnerabilities hiding in them, which haven't been fixed because the Drupal Security Team hasn't been supporting them for some time.

Although, that is less important than making the security updates for Drupal core and the 14 modules that have them, because they *definitely* have security vulnerabilities which are already known publicly (the details are published in the security advisories). This means attackers would know how to start crafting an attack.

The DBTNG one is particularly notable. I don't know if you're familiar with "Drupalgeddon" but it was the worst security vulnerability to affect Drupal ever. It was a vulnerability in the Drupal 7 database layer - so for the most part it doesn't affect Drupal 6. However, DBTNG is a backport of the Drupal 7 database layer to Drupal 6 - so versions 1.0-rc4 and lower have the same vulnerable code! It's harder to exploit on Drupal 6 sites, but the vulnerable code is there.

Also, there were several contrib modules that have been custom patched. Some patching of contrib modules is OK, but it increases the maintenance burden of the site, because every time you update a module, you have to remember to re-apply the patch (and sometimes the patch needs to be changed for the new version too).

This site, unfortunately, includes quite extensive patching, both in the number of modules that have been patched and the extent of some of the changes. Here’s what we found:

- ad
    - Hook and a UI element commented out
- amazon
    - Cron function commented out, settings form options altered
- cdn
    - According to the comment, was patched by Litza for an experimental approach
- date
    - Changes to the settings form
- domain
    - Changes to the domain_settings_ok() function
- domain_meta
    - Changes to the settings form
- flag_note
    - Patched to remove PHP 5.2 warning
- forward
    - Changes to e-mail template
- google_cse
    - Changes to settings form
- link
    - Changes to theme function output
- menutrails
    - Made menutrails configurable per taxonomy vocabulary
- nice_menus
    - Added some Javascript that appears to correct the z-index of the menu
- nodewords
    - Changes to the way that meta tags are output
- print
    - Hard-coded the location of the logo for ReadingGroupGuides.com
    - Some left-over debugging code

Some of these changes could be done without patching the code in the contrib module. For example, any time a settings form is changed, it could have been done in a hook_form_alter() in a custom module, so the code in the contrib module is left clean and updates could be applied with having to worry about losing the custom changes to the module. The same goes for changes the output of theme functions or templates - those could have been done in a custom module or theme, so the contrib module remains clean.

At the very least, it’s recommended that these patches get organized in some way that makes them easier to re-apply when making updates (for example, creating a Drush make file).

Also, since these changes are custom code, it's effectively making maintenance of these particular modules more like maintaining a custom module than just a standard contrib module.

Security

We weren't able to find any obvious signs that the site has been hacked!

There were some configuration things, however, that make the site less secure:

1. **The PHP module is enabled.** Even if only administrator users are allowed to use the PHP filter, attackers could use an XSS vulnerability (the most common vulnerability found in Drupal 6 and 7 contrib modules) to use the website as an administrator user, and then escalate their access via the PHP filter. It's recommended to disable this module on production sites. Unfortunately, there are three pages that use the PHP filter (but they could be replaced with code in a custom module):
    a. http://example.com/newsletters
    b. http://example.com/connect
    c. http://example.com/connect-with-us
2. **The "Filtered HTML" allows unsafe HTML,** including the <script> tag, which means that anyone who has access to edit content can execute an XSS attack. For a little more background, an XSS vulnerability allows a user to add code to a page that will execute as any user who visits that page and perform any operations they can. So, if an administrator user visits a special page created by an attacker, that attacker will be able to perform actions the administrator is able to! Unfortunately, there are some pages that use <script> tags in content (but they could be replaced with code in the custom theme):
    a. http://example.com/faithful-surfer
    b. http://example.com/faithful-surfer-study-guides
    c. http://example.com/surfer-awards
3. **Password included in user emails.** Drupal offers a '!password' token that can be included in email templates, but it should not be used because it can be stolen. The '!password' token is used two e-mail templates.

Since your site appears to only allow staff to login and created content, some of the risk of the above is mitigated, but not completely! You're dependent on how secure your users passwords are, and since your site doesn't use HTTPS, someone running a password sniffer on the wifi at a public coffeeshop could conceivably pick up the password for a staff member when they login.

Those are more remote possibilities, but for due diligence we felt they were important to note!

Custom modules and themes

There are several custom modules:

- 'krc_scraper'
- 'trc_scraper'
- 'brc_nodewords'
- 'rgg_packets'
- 'brc_search'
- 'brc_quotes'
- 'rgg_scraper'
- 'brc_delete'
- 'brc_connect'
- 'brc'
- 'frc_scraper'
- 'brc_scraper'
- 'rgg_custom'
- 'brc_tokens'

We weren't able to evaluate all of this custom code in-depth, because there is quite a bit (more than average). Unfortunately, it doesn't totally follow Drupal coding standards, is relatively dense (ie. lots and lots of code crammed together) and doesn't include much in the way of comments. That said, there didn't appear to be any major violations of the Drupal APIs from a quick skim through the code.

I think this, along with all the contrib modules that were patched, adds to the maintenance burden of this site more than anything.

There is the custom theme 'mytheme' based on the Adaptivetheme base theme. It acts as the base theme for the other themes: 'c20', 'gnr', and 'rgg'. These child themes are relatively simple and straight-forward, pretty much just having template and CSS customizations.

The 'mytheme' theme itself is quite complex, though, and does add slightly to the maintenance burden of this site. Unfortunately, it violates one Drupal best practice by including a couple database queries in the theme - these would be better in a custom module, with only the appearance generated by the theme. However, they are simple and relatively harmless - you just remember they're there.

Conclusion

There's a number of pending maintenance tasks, including: Drupal core and module updates, hardening configuration for security, and organizing the way modules are patched. However, this definitely isn't out of the ordinary - most sites have similar needs, just not to this extent. In any case, these are things that we'd help you take care of in your first month if you signed up for any of our plans!

The main concern with this site is the large amount of custom code. So, while the choice of plan is ultimately up to you, **we'd recommend the "Enterprise" plan ($1250/mo).** This includes a small bucket of hours (5hrs) to allow us to account for custom code when applying updates, digging into any issues on the site, fixing bugs or addressing security.

If you didn't want any support for your custom code, **you could also select the "Standard" plan ($625/mo)**, which includes the following (also included in the Enterprise plan):

- Making security fixes
- Fixing bugs that prevent users from performing critical use-cases
- Performing basic one-off maintenance and support tasks on request
- Getting your site back online in case of an outage
- Remediation if your site has been hacked

But the "Standard" plan would only cover Drupal core and (unpatched) contrib modules - we wouldn't be responsible for your custom code, including even testing if a security update to Drupal core or a contrib module would break your custom code. This is why, given what we've learned about your site, that we'd recommend the "Enterprise" plan for your needs.

Theoretically, the **Basic plan ($125/mo)** could also be an option if the site was moved to Pantheon and if you only wanted security support for the "popular" contrib modules used on the site (defined as being in the top 200 on Drupal.org). But the site is using a large number of modules, many of which aren't "popular", and of course, this also wouldn't include support for the custom code. That's why we're not including it in our recommendation.

Please let us know if you have any questions!