Audit Report for *(Client)*

- *(submission date)* -

Content

Agenda

The following Notation is used in the document:

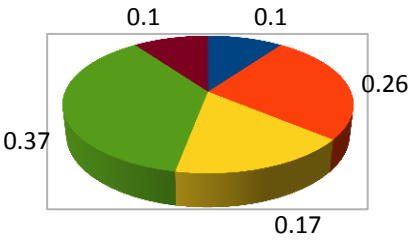
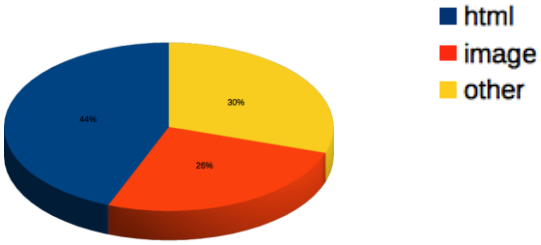
Severity levels: low, average, high, critical

Difficulty levels: low, average, high

I. General information and server configuration

Severity: Critical
Problem: Sensitive files are public. Examples: - http://www.clientwebsite.com/file.....
Recommendations: - move sensitive scripts out of site public root directory
Difficulty: Average

Severity: Critical
Problem: PHP's display_errors is On. On production server display_errors should be Off so user 's could not see occurring errors & sensitive information that could come along them.
Recommendations: - modify back index.php to original 's Magento index.php file.
Difficulty: Low

Severity: Average
Problem: Too many HTTP requests for images. Examples: - **on events page** ~72 image requests on first view ~2 image requests on next views (good client caching) **Requests**  **Requests** 
Recommendations: - transform various background images in CSS sprites - Use CDN for static resources (this will also avoid passing the cookies when make requests to static content – thus improving page load time)
Difficulty: Average

Severity: Low
Problem: VMissing compression for few static files.
Recommendations: Add gzip compression to static files, or use a CDN
Difficulty: Low

Severity: Low
Problem: Various PHP configurations
Recommendations: We recommend changing the following php.ini configurations: - PHP memory limit is set to 384Mb. The recommended value is 512Mb. - consider upgrading PHP to 5.4 (currently 5.3.1). Magento announced recently that Magento Community (version 1.6 and higher) supports PHP 5.4 – and this will improve performance and memory usage.
Difficulty: high (should be done by server administrator after thorough testing on our own server!).

Magento configuration suggestions
In System → Configuration : - Catalog → Google Sitemap – should be configured (this is useful for SEO)
- Change the admin URL from "**admin**" to something that's not default (can be changed from XML configurations).

Our recommendations:

- Enable gzip compression for static resources
- Make sure the images have corresponding dimensions
- Check the admin/php/server configurations we mentioned.
- Use a CDN if possible

II. Module reviews

1. *Community code pool (./app/code/**community**)*

Module **A****** / M********:**
- no comments - ACL for admin controller is not checked...any admin can access the respective url regardless its permissions! - well enforced Magento coding standards

Conclusion: No obvious problems with the module.
Potential for bugs: **very low** (considering the tiny codebase)
Recommendations: - rewrite admin controller and `override _isAllowed()` method to properly check if user has permission to access the respective area / contact the vendor and notify him about the problem in order to fix it in a future release - disable this extension in production environment, use it only in development to check if conflicts occur.

Module **F******* / R**********:**
- some parts are well commented, while some other are not commented at all - very well enforced Magento coding standards - this modules enters in a conflict with another module (R***** / M*****) when rewriting Mage_Review_Block_Form
Conclusion: No obvious problems with the module, overall of good quality.
Potential for bugs: **high**
Recommendations: - resolve the rewrite conflict

Module **L********* / M*********:**
- well commented - very well enforced Magento coding standards
Conclusion: No obvious problems with the module, overall of good quality.
Potential for bugs: **very low** (considering the tiny codebase)
Recommendations: none

Module **S********* / C*********
- no comments - very well enforced Magento coding standards
Conclusion: No obvious problems with the module, overall of good quality.
Potential for bugs: **very low** (considering the tiny codebase)
Recommendations: none

2. *Local code pool (./app/code/**local**)*

Module **A********* / A*********:**
- poorly commented - admin controller does not conform to Magento standards
Conclusion: No obvious problems with the module.
Potential for bugs: **very low** (considering the tiny codebase)
Recommendations: - contact the vendor and notify him about the controller remark in order to fix it in a future release

Module **A********* / O*********:**
- poorly commented - very well enforced Magento coding standards - some php strings are double quoted, instead of single quote for better performance - missing comment attribute value for system config field “description” translation
Conclusion: No obvious problems with the module, overall of good quality.
Potential for bugs: **Low** (considering the large codebase)
Recommendations: none

Module **G********* / A*********:**
- code is well commented - Magento coding standards are generally respected
Conclusion: No obvious problems with the module.
Potential for bugs: **very low** (considering the large codebase)
Recommendations: none

Module **M********* / C*********
- Parts of the code are not commented at all (while other parts are well commented) - disabled using a not advisable method (deleting activation file) - Potential for conflicts because it rewrites a lot of Magento core classes: Mage_Adminhtml_Block_Catalog_Product_Attribute_Edit_Tabs …... …... Mage_Catalog_Model_Resource_Eav_Mysql4_Attribute
Conclusion: No obvious problems with the module
Potential for bugs: **low** (considering the large codebase).
Recommendations: - best practice is to delete module files if the extension is out of use (using a code version control system the module can be added when needed. We recommend to use GIT as version control system for your code)

Module **W********* / C*********:**
- well commented - well enforced Magento coding standards
Conclusion: No obvious problems with the module
Potential for bugs: **very low** (considering the tiny codebase).
Recommendations: none

Upgrades are available for the following extensions, if upgrading to Magento CE 1.9:

- E*********_M********* - client 's version: 1.1.15, currently available: 1.1.25
- …....
- …...
- F*********_G********* – client 's version: 0.13.7, currently available: 0.15.12

These upgrades can add features, improve security and fix potential bugs.

Conclusions:

- The installed modules come from the following vendors (in community and local): (...).
- A rewrite conflict was found that needs to be resolved: (…) module rewrites Reviews Form block.
- Modules (…) (…) use jQuery (a javascript library that's not included in magento). This can potentially cause conflicts, if jQuery is included multiple times by various modules, or have dependencies of different jQuery versions. There seem to be no conflicts with the currently installed modules – as they are made by the same company, however this should be taken into consideration when installing new modules
- There are 2 modules that are unused. They should be deleted and thus cleanup the code a little bit.
    - (…)
    - (...)

  There seem to be differences between normal modules and “old” modules.
  If the “old” versions are just and old version of extension from the vendor there should not be any problems deleting them.
  If the non “old” versions contain modifications made directly by the client then this will cause issues on further extensions upgrade and is not a good practice to modify vendor ‘s extension directly. The custom modifications should be made using standard Magento rewrite functionality.

III. Magento core files modifications

Magento core files modified directly. These changes will be overwritten if an upgrade is made, and need to be saved before making an upgrade:

./app/code/core/Mage/clientwebsite/controllers/**IndexController.php**
- Line #371
Original code:
Modified code: **added** `$session = Mage::getSingleton('customer/session');`
- Lines #341, #402
Original code:
Modified code: **added** `$message = $this->__('Your information has been updated.');` `$session->addSuccess($message);`

Local <Mage> files.

There are 3 files in `local/Mage` that is used in *System -> Import / Export -> Dataflow - Advanced Profiles*. Usually in `local/Mage` reside files are not overwritten using the Magento rewrite functionality, the file is copied, and then the modifications are made. This is a bad practice anyway and can cause upgrade problems:

- `Mage_Catalog_Model_(...)`
- `Mage_Catalog_Model_(...)`
- `Mage_Catalog_Model_(...)`

Recommendations: Put the three files in custom module and update through an installer the dataflow advanced profiles according to the new models belonging to the new package.

Front-end modifications:

The package used is (...) and the theme is **default**. The site **should have its own theme** and there the current modified files/customizations should reside, not in the default theme. In the default should reside all original layouts and templates (from third party/community extensions for example – now they are spread all over `base/default` and `default/default` themes)

Admin modifications:

Magento has a default package and theme. (`default/default`). The change of custom package/theme for admin takes a little more work in admin area than on front-end, it requires an additional module with following settings in its `etc/config.xml` file:

```
<stores>
      <admin>
            <design>
                  <...>
                    <...>...</...>
            …
            </design>
      </admin>
</stores>
```

This is a good practice for leaving default admin theme unmodified so further upgrades go easily. The new third-party or local code/modifications should be placed in the new custom package/theme.

Conclusions:

- There are few core modifications that aren't done according to best practices, these are minor and easy to fix, and on an upgrade they should be taken into account/fix.

- Design package/themes should be reorganized because they don't really apply Magento's best practices.

- Enabling Magento profiler on production environment has a negative performance impact.

Difficulty: Average

IV. Audit conclusions

1. There are a few custom modules and modifications. Although some modules have minor problems, the majority of them follows Magento coding standards and is actively maintained.

2. The server configurations are overall appropriate. There are some tweaks that can (and should) be made, that would improve the site load speed, and in certain situations these improvements could be significant.

3. **An upgrade is recommended** to keep up with all the recent security updates, optimizations and features Magento has to offer. The main concern here is the directly core modifications that have been made and not losing third party's design due to current package/themes organization.

4. Site‘s Magento version is (...) and at the time of writing the last available **Magento CE is 1.9.1.0**
