Securing Mambo Open Source CMS

v.0.4

By Jascha

jascha@localareasecurity.com

http://localareasecurity.com

Table of Contents

Introduction

(From mamboserver.com)
Mambo Open Source is the finest open source Web Content Management System available today. Mambo Open Source makes communicating via the Web easy.

Have you always wanted to have your own site but never understood how? Well Mambo Open Source is just the ticket!

With Mambo Open Source there is no need for HTML, XML or DHTML skills, just enter your content, add a picture and then through the easy to use administrator web-interface ...click Publish!

In this paper I will cover the basics of helping to insure that your install of Mambo Open Source is as secure as possible. Being that there have been recent vulnerabilities published it is best to take extra precautions in light of possible new ones arising. As well as a few notable sites being taken down due to these intrusions. Following some common sense measures to protect your install of Mambo and your host server will go a long way in preventing defacements and compromises.

This paper assumes you are running Mambo on a *nix like server (Linux, FreeBSD, OpenBSD, etc). Using Apache, PHP, and mySQL. I will not cover all the nuances of securing each component involved with running Mambo. But will instead cover some best practices that should be followed. *Some of these could also apply to users running on Windows servers as well, mileage may vary.*

Who Should Read this Paper?
You are installing Mambo or already have Mambo powered sites that need additional security. Readers should have a basic understanding of Apache, PHP, mySQL, Mambo, and a healthy helping of common sense. There are links to additional information in each section for those wishing to get more detail on a chosen topic. This paper also approaches each topic as it applies to both users with a dedicated server and those on shared hosting without shell access. An important topic readers should have a clear understanding of for the purpose of using many of the tips in this paper is Apache '.htaccess' files.

More on .htaccess files:
What is .htaccess? - http://httpd.apache.org/docs/howto/htaccess.html

Why Write This Paper?
After moving my site over to Mambo and taking steps to securing it, I saw that the average user may not take the steps needed to secure their sites. With the aforementioned vulnerabilities recently found in Mambo I saw an even greater need to educate users on how to prevent their sites from being compromised. So I took a few hours and typed up the first public version of this document (0.2). The response was very positive from the Mambo community so I have been adding additional sections as time permits.

What is Cross Site Scripting (XSS)?
When it comes to recent issues with Mambo and many other content management systems. One of the most common references to vulnerabilities is XSS or Cross Site Scripting. Which is actually a very broad term that could apply to a laundry list of issues many sites may have. For the purpose of this paper and the readers' general understanding we will define XSS as this:

"Web pages that can be tricked into displaying web surfer supplied data capable of altering the page for the viewer."

Now the main thing to keep in mind is that even if there exists an XSS vulnerability in a web page it takes a whole lot of creativity on the part of the web surfer to exploit this to their advantage. Being that many XSS issues only produce null results. But depending on the skill level of the web surfer most any XSS hole could lead to bigger issues for webmasters.

More on XSS:
The Cross Site Scripting FAQ - http://www.cgisecurity.com/articles/xss-faq.shtml

What is SQL Injection?
Another vulnerability that has become a buzz word in security and in relation to Mambo and many other dynamic sites is SQL injection attacks. So understanding the problems helps you understand how to prevent them. We will define SQL injection as:

"When user supplied data in SQL queries doesn't strip potentially harmful characters."

SQL injection is easily prevented but often overlooked by developers.

More on SQL injection:
SQL Injection, Are Your Web Applications Vulnerable? -
http://www.spidynamics.com/papers/SQLInjectionWhitePaper.pdf

“You are only as secure as your weakest link.”

Before embarking on this journey I want to clarify for readers that there is no easy way of taking the steps needed to setup a server as securely as possible. Which includes removing unneeded services and packages from your server. Along with checking all permissions are setup correctly. Not to mention firewalling, file integrity checking, and intrusion detection to name a few. It is always the little things that people overlook that lead to their servers being compromised.

I will touch on some of these topics and direct readers in the right direction to gain an understanding of them. It is not required nor my intention to make everyone a security expert. But there is a need for people to know how to defend themselves adequately.

Here are a few links to helpful general knowledge about securing your server for hosting Mambo:

MySQL

MySQL Security - http://www.mysql.com/doc/en/Security.html
Securing MySQL: step-by-step - http://www.securityfocus.com/infocus/1726
Secure MySQL Database Design - http://www.securityfocus.com/infocus/1667

Apache

Securing Apache: step-by-step - http://www.securityfocus.com/infocus/1694
Apache 2.0 Security - http://httpd.apache.org/docs-2.0/misc/security_tips.html
Apache 1.3 Security - http://httpd.apache.org/docs/misc/security_tips.html

PHP

Securing PHP: step-by-step - http://www.securityfocus.com/infocus/1706

Protecting Your .htaccess File.

The first step to using .htaccess files is to insuring that web surfers can't get their prying eyes on them. It's a simple yet important step to getting started with securing your server.

In all your .htaccess files add the following:

```
<Files .htaccess>
order allow,deny
deny from all
</Files>
```

Now (if available to you) *chmod 644 .htaccess*

This should be done with any .htaccess file you use.

This is an important step to not overlook while setting up .htaccess files!

“Encryption is your friend.”

In order to secure access to certain sections of your site and especially certain parts of your Mambo install (such as /administrator) you will want to use SSL to secure the sessions using encryption. This will insure that administrators use SSL to access the administrator section of Mambo.

First you may need to setup OpenSSL on your server:
http://www.openssl.org

As well as setting up mod_ssl:
http://www.modssl.org/docs/

Firstly you want to insure that users trying to access the /administrator section of Mambo are forced to use an SSL connection. This assumes you have mod_rewrite installed on your server. Add the following lines to your .htaccess file:

```
#/administrator/.htaccess
RewriteEngine On
RewriteRule ^/$ /index.php
RewriteCond %{SERVER_PORT} !443$
RewriteRule ^(.*) https://yourhost.com/administrator/$1 [R=301,L]
```

This will insure you have an encrypted session before you are prompted for the .htaccess and Mambo administrator username and password. Which protects against the possibility if someone is conducting a sniffing attack against you hoping to capture your cleartext usernames and passwords.

More on SSL / Encryption:
What is SSL? - http://en.wikipedia.org/wiki/Secure_Sockets_Layer
How Encryption Works - http://computer.howstuffworks.com/encryption.htm/printable

“When in Doubt, Password Protect.”

Another basic precaution to take is to secure your /administrator directory using .htaccess to add another layer of password protection. Simply add the following lines to your . htaccess file for /administrator.

```
AuthType Basic
AuthName "Private Area (or whatever you wish to call it)"
AuthUserFile /path/to/users/file/
require valid-user
```

You will also need to create a file containing users and passwords with authorization to access /administrator directory. For this example we will call it 'users'. This file should reside somewhere other than your public html directory. You may generate these files using *htpasswd* or by using an online tool such as:
http://www.eftel.com/cgi-bin/user_add.cgi

This will generate usernames and encrypted passwords that look like this:

```
jimmyjoebob:hGYMEhFceht.E
```

Once you have generated these files place them in their respective locations and test to insure insure they are working properly.

More on Password Protection:
Apache 1.3.x mod_auth - http://httpd.apache.org/docs/mod/mod_auth.html
Apache 2 mod_auth - http://httpd.apache.org/docs-2.0/mod/mod_auth.html

“Locking Down Access to /administrator via Domains and IPs”

Now that you have password protected and forced SSL connections to your administrator directory you now want to lock down WHERE administrators can gain access from. By taking a layered security approach even if there were a new Cross Site Scripting (XSS) vulnerability in some part of the administrator section of Mambo. You would be covered being that you have locked down all access to that area.

Your New Best Friend: mod_access

There are many ways to secure a site or directory using mod_access. For the purpose of this paper I will cover it in respect to using an .htaccess file. Depending on how you are using Mambo would dictate which way you would want to use mod_access options. Also first be sure your server has mod_access loaded.

Scenario 1 :
Administrator(s) using DSL/Dial-up with DHCP (non-static IP address)

Being that you maintain your site remotely and using a non-static IP you can approach this two ways. The first would be to deny all but the partial domain name of your DSL/Dial-up provider:

```
Order Deny,Allow
Deny from all
Allow from myprovider.com
```

This tells mod_access to deny all others access but those from myprovider.com which helps to lock down the site to all but you. But this would also include the possibility of everyone else using myprovider.com to access you /administrator directory (which may be an issue).

The other option would be to lock the directory down by whatever IP block your DSL provider uses to assign your addresses:

```
Order Deny,Allow
Deny from all
Allow from 10.1
```

This would be the first 1 to 3 bytes of the IP block that you get assigned addresses. You want to be sure that your provider does not revert to another block of IPs if the ones you typically get become exhausted. Being that you will then be unable to access the administrator directory.

Other Allow options include:

```
Allow from 10.1.0.0/255.255.0.0
```

This allows you to still select a IPs as the previous example. But also allows locking down by the addition of netmask which makes for more detailed access restrictions

Scenario 2 :
Administrator(s) using static IP addresses

This is a much more clear cut example and is for obvious reasons the more secure means of locking down access to /administrator. You would simply add:

```
Order Deny,Allow
Deny from all
Allow from 10.1.23.4
```

Where 10.1.23.4 is your static IP address. These directives may be used in conjunction or in multiple entries for more than one administrator etc. It is always a best practice in this situation to use Deny first in Order to insure only addresses/domains you supply have access to /administrator or any other directory you wish to protect.

NOTE: If you are on a network behind a NATed firewall be sure to use you PUBLIC IP address in your entries. Using internal IP addresses would only be used if Mambo is setup to only be used as an intranet site.

More on Apache mod_access:
Apache 1.3.x - http://httpd.apache.org/docs/mod/mod_access.html
Apache 2 - http://httpd.apache.org/docs-2.0/mod/mod_access.html

Putting It All Together

Example 1
Mambo as an Intranet Site

As many of you know Mambo can make for a great intranet site for your organization. Allowing for people to access and write news, articles, and F.A.Q.s. As well as share information in forums and using a document manager.

Important things to keep in mind when setting up Mambo for intranet use is to insure that users outside of your organizations' firewall can not get to your Mambo site. Here are some helpful tips to securing your Mambo intranet site.

- Utilize the forced SSL and .htaccess password controls as detailed earlier. Even though there is no Internet users accessing the site it is better to be overly secure than not secure enough. Most network security issues are caused by internal network users.

- Use the access controls outlined earlier to only allow access to the internal IP addresses your network uses. But instead of placing the .htaccess file in the administrator directory you would place it in the root of the Mambo installation (/).

  An example would be if your internal network used 192.168.1.1 with a netmask of 255.255.255.0 you would add to the .htaccess file these lines:

  ```
  Order Deny,Allow
  Deny from all
  Allow from 192.168.1.0/255.255.255.0
  ```

  This would tell Apache to Deny all other addresses but Allow those with addresses in your internal IP range.

- Add a rule to your firewall to not allow external traffic to access the IP of your intranet web server running Mambo.

- Add a rule to your firewall to not allow traffic originating from your Mambo intranet server out through the firewall to the outside world (Internet).

You now have a secured intranet site for your company or organization.

More on Intranets:
Intranet Definition - http://en.wikipedia.org/wiki/Intranet
"Issues in Intranet Security" - http://www.intranetjournal.com/features/isecurity.shtml

Example 2
Mambo as an Extranet Site

With the ever growing use of telecommuting and distributed projects with contributers living in various countries on separate continents the need for secured extranet sites is ever increasing. You can easily setup Mambo as a secured extranet site in a few simple steps.

- Follow the previously outlined steps to securing your Mambo installation.

- Instead of forcing SSL for only the administrator directory. Instead place the .htaccess file in the root (/) of your Mambo site. Thus forcing ALL access to the site to be over SSL.

- Add the .htaccess authentication to the root of the site as well. Be sure to first force the SSL session prior to the lines for .htaccess authentication to insure the users are entering their passwords over an encrypted channel. Each extranet user will need to be given a username and password to gain access.

- You may also use the tips from above to only allow the users to access the extranet site from specific IPs or domains if you wish. Depending on the number of users and other factors this my become an administration nightmare.

- Be sure to lock down your firewall rules to insure stringent access controls to your internal network. It is assumed your web server would be located in a DMZ*.

By securing the entire Mambo site using SSL and authentication you can create a secured extranet site in minutes. Now your organization can easily share information with partners, clients, and more! I will not get into too much detail in terms of securing your internal network in order to allow business partners etc. to access information being that each case is always a little different. Extranets are very useful for businesses needing to share catalogs and information with partners and clients. As well as for such things as e-learning, online collaboration, etc.

More on Extranets:
Extranet Definition -
http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci212089,00.html
**DMZ Definition* - http://en.wikipedia.org/wiki/DMZ

“Keeping the Spambots, Leeches, and Kiddies a Bay”

One of the biggest problems with hosting a site is malicious people who do everything from harmless annoyances to series disruptions of your site. There are a few easy ways to block users by IP address or domain name following the same example of using mod_access as outlined earlier. Instead of allowing and denying all for the administration section. You are wanting to allow all and deny certain trouble makers.

Deny by IP Using mod_access in .htaccess

The following are simple examples:

```
order allow,deny
deny from 10.15.20.2                    #specific banned IP
deny from .badguysdomain.com            #ban all users originating from this domain
allow from all
```

You are telling Apache to allow everyone but those in the list.

Denying Access Using PHP

Another way of blocking users is by using the power of PHP. By using a simple script like this:

http://localareasecurity.com/scripts/ipblocker-1.1.tar.gz (Download)

You can easily edit this script to grep a list of IPs you gather from another program etc. This script already has the optional ability to have a custom message for each offending IP address.

“Spambots and Rippers Ate My Bandwidth!”

One big issue most any webmaster faces is the constant crawling of their sites by Spambots looking for fresh email addresses. You can use the power of mod_rewrite to help thwart the arachnid from eating up your bandwidth and Spamming yours and visitors email addresses.

First we will look at a list of some **Spambots**:

```
RewriteEngine on
RewriteBase /
RewriteCond %{HTTP_USER_AGENT} ^EmailSiphon [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailWolf [OR]
RewriteCond %{HTTP_USER_AGENT} ^ExtractorPro [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mozilla.*NEWT [OR]
RewriteCond %{HTTP_USER_AGENT} ^Crescent [OR]
RewriteCond %{HTTP_USER_AGENT} ^CherryPicker [OR]
RewriteCond %{HTTP_USER_AGENT} ^[Ww]eb[Bb]andit [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebEMailExtrac.* [OR]
RewriteCond %{HTTP_USER_AGENT} ^NICErsPRO [OR]
RewriteCond %{HTTP_USER_AGENT} ^Teleport [OR]
RewriteCond %{HTTP_USER_AGENT} ^Zeus.*Webster [OR]
RewriteCond %{HTTP_USER_AGENT} ^Microsoft.URL [OR]
RewriteCond %{HTTP_USER_AGENT} ^Wget [OR]
RewriteCond %{HTTP_USER_AGENT} ^LinkWalker [OR]
RewriteCond %{HTTP_USER_AGENT} ^sitecheck.internetseer.com [OR]
RewriteCond %{HTTP_USER_AGENT} ^ia_archiver [OR]
RewriteCond %{HTTP_USER_AGENT} ^DIIbot [OR]
RewriteCond %{HTTP_USER_AGENT} ^psbot [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailCollector
RewriteRule ^.* - [F]
RewriteCond %{HTTP_REFERER} ^http://www.iaea.org$
RewriteRule !^http://[^/.]\.yourdomain.com.* - [F]
```

Be sure to change 'yourdomain.com'. This will block many known Spambots that crawl sites for email addresses.

Now for a FULL integrated **Spambot** and **Ripper** list (too long to list here):

http://localareasecurity.com/scripts/spambots-rippers-for-htaccess.txt

The difference other than being a **much** longer list is that there is the addition of the last line: RewriteRule ^.*$ middlefinger.php [L] Which will redirect all of the list above to a page called 'middlefinger.php'. You can remove this line and edit with what you have learned in previous examples to tweak your configuration.

You may also be interested in a small perl script for directing Spambots to never ending lists of randomly generated emails **(Use at your won risk!)**:
http://localareasecurity.com/scripts/Infinospam2.pl.txt

“Protecting Content from Leeches and Linkers”

An easy but very effective way to prevent users from linking to your files directly is using the power of mod_rewrite. For our first example we will block external sites from linking to our images:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?yourdomain.com/.*$ [NC]
RewriteRule \.(gif|jpg)$ - [F]
```

As you see we would place this in the .htaccess file for a directory containing images we want to protect. Or place this in the root of your site to protect all images. Additionally you can add other images types such as bmp, png, etc. depending on your needs.

Alternatively you can also use the above to protect linking to PDFs etc. by changing
`RewriteRule \.(gif|jpg)$ - [F]`

Another fun trick is to have external image links all point to one image of your choice. It could also be an HTML page with a warning or whatever you wish. The following is an example of pointing to an alternate image:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?yourdomain.com/.*$ [NC]
RewriteCond %{HTTP_REFERER} !^http://(www\.)?another-allowed-domain.com/.*$ [NC]
RewriteRule \.(gif|jpg)$ http://www.yourdomain.com/middle-finger.gif [R,L]
```

So all external image links would default to 'middle-finger.gif' in this example.

More on Spambots, Leeching, etc.
Stopping Spambots - http://www.neilgunton.com/spambot_trap/
The Web Robots Page - http://www.robotstxt.org/wc/robots.html
Robotcop - http://www.robotcop.org/
Using Apache to Stop Bad Robots -
http://www.evolt.org/article/Using_Apache_to_stop_bad_robots/18/15126/index.html

Conclusion

Mambo Open Source is a great and user friendly content management system good for use as Internet, intranet, and extranet sites. It's ease of use allows for easy collaboration and content management. By taking a few extra common sense steps and more fully using the powerful server software it runs on users can secure their own sites easily. As a good webmaster you must do all you can o prevent malicious people from gaining access o your site and server. So along with the recommendations outlined in this paper it is also good practice to review your log files and other general steps to monitoring your site and server. Some examples of best practices would be:

- Monitor log files or use tools such as LogWatch (http://www.logwatch.org) to monitor your logs for you. There are many other similar tools available as well.

- Keep your server up to date with the latest patches. All vendors (RedHat, Debian, etc.) have free mailing lists you can subscribe to in order to receive emails of vulnerabilities and patches available for your server (or allows you to keep track of your hosting company).

- Disable all unused or unneeded services on your server. As an example, if you are running BIND (named) but aren't using your server for DNS then turn off that service (and firewall that port). Many Linux distributions and other operating systems have many unused or dangerous services turned on by default.

- Setup **at least** a host based firewall. If you are running a Linux server you can use a tool such as Firewall Builder (http://www.fwbuilder.org/) to create a rule set to allow for what you need and deny the rest. You can find plenty of documentation on your systems firewall by using Google (http://www.google.com)

- Backup your Mambo directory and database on a regular basis. This will insure you always have a copy in case your server fails or is compromised. Be sure to do these backups to another computer and not just on the server itself.

By following these steps and using common sense you can go a long way to securing your server and Mambo site to the fullest. Many of the above won't apply to you if you are using shared hosting. In that case be sure to wisely pick your hosting provider and don't be scared to ask questions about their security practices. You are paying them after all to host something I am sure you work very hard on.

Common Issues

“My server doesn't recognize my .htaccess file.”

If the .htaccess file doesn't work be sure to look in your httpd.conf file and look for 'AllowOverride' to insure it is set to enable the use of .htaccess files. Or if you are using shared hosting you may need to contact your support representative to troubleshoot this issue.

Please contact me (jascha[at]localareasecurity.com) with any changes or suggestions to this paper. Or use the bug trackers on the project page to report issues with this paper. You can always find the latest version of this paper at: http://mosforge.net/projects/mossecurity/