

Installing WPoison on a Slash Site

Collin Starkweather

May 26, 2002

Abstract

This document describes the installation and configuration of WPoison on a Slash Code site. It also contains a variety of techniques that can be used to disguise your WPoison installation from detection by a harvesting web crawler even if you are not running a Slash Site. All of the information and links are current as of May 26, 2002.

Contents

1	Introduction	2
2	Installation	3
2.1	Installing Apache	3
2.2	Installing Perl's Template Toolkit	4
2.3	Installing WPoison	4
3	Customization	5
3.1	Edit <code>wpoison.cgi</code>	5
3.1.1	Customize your Environment	5
3.1.2	Edit the HTML Template	6
4	Configuration	6
4.1	Apache Configuration	6
4.2	Slash Site Configuration	7
5	Conclusion	8
A	Requirements	8
B	Necessary Files and Templates	9
B.1	<code>wpoison.cgi</code>	9
B.2	<code>wpoison.shtml</code>	14
B.3	The mainmenu Template	14
B.4	The footer Template	14
C	Compiling this Document	15

1 Introduction

WPoison is a Perl script created by Ronald F. Guilmette which creates a “trap” for unwitting spam bots. The idea is simple but brilliant: A web page is dynamically generated with a variety of randomly generated bogus e-mail addresses. It also includes a number of links which simply take a bot back to another dynamically generated web page with more bogus e-mail addresses. The idea is to fill a spammer’s database with bogus e-mail addresses, adding to the cost and decreasing the effectiveness of spam.

This document explains how I installed a customized version of WPoison (see [1]) on my homepage, which is based on Slash Code (see [2]), including an explanation of how I disguised it from bots and spiders who may otherwise become wise to the fact they have fallen into a trap.

I only detail my own installation. The level of detail should be sufficient that a halfway intelligent monkey should be able to reproduce the results on their own site. For those who do not meet the minimum requirements, lots of references are provided to more detailed materials.

The techniques used may also be used to disguise your script on any Apache instance with the `mod_rewrite` module installed sufficiently that any spammer’s bot or spider will not recognize it as WPoison.

The innovations to WPoison introduced with this document are:

- Many of the advanced techniques mentioned on the WPoison web site are given explicit treatment here. It is meant to serve as a reference to others who may want to create a similar installation. The techniques that are documented here include:
 1. Disguising dynamically-generated WPoison pages to appear to be static web pages with Apache’s `mod_rewrite` module.
 2. Disguising the WPoison logo, placed on the homepage in accordance with WPoison’s terms and conditions, with Apache’s `mod_rewrite` module.
- The WPoison script, `wpoison.pl`, has been modified to
 1. Incorporate Perl’s Template Toolkit, which allows others to more easily customize the pages generated by the script.
 2. Create e-mail addresses that are genuine; however, they are filtered and discarded by the mail server. This prevents spammers from being able to easily filter bogus e-mail addresses from their database.

- WPoison has been seamlessly incorporated into a Slash Site. This document describes all necessary Slash template edits.

The system was created in a Linux environment, and should be similar (if not identical) on any Unix based environment. If you are a Windows user, the ideas would translate fairly easily; however, I am not a Windows user and am not familiar with the details of how to effect such a system in a Windows environment. If anyone successfully installs the system in a Windows environment, I would be interested to know how it was done.

Any errors or omissions are solely my fault and my fault alone. Go ahead. Blame me. See if I care.

If you discover any errors or omissions, please let me know:

`collin.starkweather@collinstarkweather.com`

2 Installation

2.1 Installing Apache

When I first tried to implement this scheme, I discovered I hadn't installed `mod_rewrite`, which is an Apache module which is used to disguise WPoison (by rewriting URLs) from spam bots.

So I rebuilt Apache with the `mod_perl` and `mod_rewrite` modules:

```
tar xvzf apache_1.3.20.tar.gz
tar xvzf mod_perl_1.26.tar.gz
cd mod_perl-1.26
perl Makefile.PL DO_HTTPD=1 USE_APACI=1 EVERYTHING=1 USE_DSO=0
APACHE_PREFIX=/opt/apache APACHE_SRC=./apache_1.3.20 ADD_MODULE=rewrite
make
make test
sudo make install
cd ../apache_1.3.20
sudo make install
```

If you're using dynamically-loaded modules¹ (this is the default in the Apache build), then include the following in `httpd.conf`:

¹I like to statically link mine. Hence the explicit `USE_DSO=0` in the build process.

```
LoadModule mod_rewrite libexec/mod_rewrite.so
```

2.2 Installing Perl's Template Toolkit

The script `wpoison.cgi` uses Perl's Template Toolkit modules, far and away one of the best modules on the CPAN in my humble opinion.

The Template Toolkit was created by Andy Wardley (`abw@cre.canon.co.uk`). Send him a thank-you note.

If you are running a Slash Site, you should already have the Template Toolkit installed. (Slash Code relies heavily on it.)

For those who are not running a Slash Site, installing the Template Toolkit is as easy as

```
sudo perl -MCPAN -e 'install Template'
```

RPMs are also available if your Perl distribution is RPM-based.

For more information, see [4].

2.3 Installing WPoison

You will need three WPoison files: `wpoison.cgi`, `wpoison.shtml`, and `logo.gif`.

Obtain the WPoison Perl script `wpoison.cgi` (see [3]) from Collin Starkweather's web site:

```
http://www.collinstarkweather.com/local/wpoison/wpoison-on-slashcode
```

or copy it out of the appendix (section B.1).

Install it in an appropriate directory on your slash site. I like to keep all of the files I generate in a separate directory called `local`:

```
cp wpoison.cgi /usr/local/slash/site/www.collinstarkweather.com/htdocs/local
```

Do likewise with `wpoison.shtml`, which is a wrapper to `wpoison.cgi` that blends the page with the overall look and feel of your slash site. (See appendix B.2.)

```
cp wpoison.shtml /usr/local/slash/site/www.collinstarkweather.com/htdocs/local
```

Finally, put the WPoison logo, which you can obtain from the WPoison web site (see [1]), in a publicly-viewable directory:

```
lynx -source 'http://www.monkeys.com/wpoison/logo.gif' > logo.gif
cp logo.gif /usr/local/slash/site/www.collinstarkweather.com/htdocs/local/images
```

Rename the logo to something generic so as to not tip off the spam bots:

```
cd /usr/local/slash/site/www.collinstarkweather.com/htdocs/local/images
mv logo.gif some-random-image.gif
```

The logo can be found at

<http://www.monkeys.com/wpoison/logo.gif>

3 Customization

Spammers are as crafty as you or I. The more differences there are between WPoison installations the harder it is for a spam bot to tell that you are running a WPoison installation. A little customization to keep their filters from recognizing your site will go a long way.

3.1 Edit `wpoison.cgi`

3.1.1 Customize your Environment

The script in section B.1 is simple and straight-forward Perl. About as simple as it gets. Those with little or no knowledge of Perl should be able to figure out relatively easily how it works and modify it to suit their needs.

If you do introduce innovations into the script, I would be interested in taking a look at your work. Don't worry about making it pretty: It's the idea that I am interested in.

3.1.2 Edit the HTML Template

The HTML template for the page generated by `wpoison.cgi` is in the `__END__` block of the script.

Simply open the script in your text editor of choice, search for `__END__`, and edit the following HTML.

You are free to change anything you like so long as you leave everything between the `<!--% and %-->` tags intact. These templating tags are replaced with the random e-mail addresses and redirects when the WPoison page is generated.

4 Configuration

4.1 Apache Configuration

Slash Code works via `Includes` in your base Apache configuration file (`httpd.conf`) which point to configuration files specific to each virtual host supported.

The included file for my site, `www.collinstarkweather.com`, is found at

```
/usr/local/slash/site/www.collinstarkweather.com/www.collinstarkweather.com.conf
```

I added the following lines into my virtual host definition:

```
<VirtualHost www.collinstarkweather.com:80>
...
    RewriteEngine on
    RewriteMap lowercase int:tolower
    RewriteCond ${lowercase:%{REQUEST_URI}} ^.*motel.*$
    RewriteRule .* http://%{HTTP_HOST}/local/wpoison.shtml
    RewriteCond ${lowercase:%{REQUEST_URI}} ^.*guilmette.*$
    RewriteRule .* http://www.monkeys.com/wpoison/
...
</VirtualHost>
```

The first rewrite rule will take any URL that matches the regular expression `.*motel.*` and redirect the user to `wpoison.shtml`.

Within `wpoison.shtml`, a series of bogus links are created such as `/motel/foo.html` which each appear to be unique static pages but which, in fact, merely return another page generated by `wpoison.cgi`.

The second rule will take any URL that matches the regular expression `.*andy.*` and redirect it to the WPoison home page. This is in keeping with the terms and conditions of the WPoison code. The WPoison logo you copied into a publicly viewable HTML directory in section 2.3 will be wrapped in a link that will be redirected to the WPoison home page.

4.2 Slash Site Configuration

Next, you merely need to configure Slash to include a link to the roack motel and a link to the WPoison web site on your front page.

As a Slash Site administrator, you can fully customize your site by editing Slash templates. These templates are designed using Perl's Template Toolkit, the very mentioned in section 2.2. So you've already learned something useful!

After you log in as an administrator, click on the [Template](#) link at the top of the page to edit Slash Site templates.

I put the following link in the `default;misc;footer(108)` template:

```
<A HREF="/guilmette/foo.html">
<IMG ALIGN="bottom" SRC="/local/images/some-random-image.gif" BORDER="0"
      WIDTH="105" HEIGHT="40">
</A>
```

Note that the hyperlink points to an address which will be caught and redirected by the second rewrite rule defined in section 4.1.

The full text of my footer template can be found in section B.4.

I also added the following link in the `default;misc;mainmenu(134)` template:

```
- <A HREF="[ % constants.rootdir %]/motel/foo.html">The Roach Motel</A><BR>
```

This hyperlink points to an address which will be caught and redirected by the first rewrite rule defined in section 4.1.

The full text of my main menu template can be found in section B.3.

5 Conclusion

That's all there is to it! It's a little bit more difficult than falling off a log, but not really all that bad.

I was curious how many spam bots did, in fact, get trapped in my web.

Since all of the e-mail addresses generated by `wpoison.cgi` direct the user to the domain `collinstarkweather.com`, I set up a filter in Evolution (my latest e-mail client of choice) to filter everything to the `collinstarkweather.com` domain *except* `collin.starkweather@collinstarkweather.com` into a special folder.

Please let me know if you have any questions or comments:

`collin.starkweather@collinstarkweather.com`

References

- [1] Ronald F. Guilmette. `wpoison.pl`. <http://www.monkeys.com/wpoison>, May 2002.
- [2] Rob Malda and CowboyNeal. Slash code. <http://www.slashcode.org>, May 2002.
- [3] Collin Starkweather. `wpoison.cgi`. <http://www.collinstarkweather.com>, May 2002.
- [4] Andy Wardley. Template toolkit. <http://search.cpan.org/search?dist=Template-Toolkit>, May 2002.

A Requirements

- Apache
- Apache's `mod_rewrite` module (bundled with the Apache source)
- Perl
- Perl's Template module
- Slash Code

- `wpoison.cgi` (see section B.1)
- `wpoison.shtml` (see section B.2)
- A little ingenuity.

B Necessary Files and Templates

B.1 `wpoison.cgi`

This is a modified version of Rnald F. Gilmette's original `wpoison.pl`. I have slightly modified it to:

- Incorporate templating into the generated document with the Template Toolkit. This is a general improvement which allows a user to more easily customize the output of the script.
- Force all e-mails to be generated for a specific domain. Since spammers can easily filter bogus e-mail addresses, I created filters to accept all e-mails to my domain, then simply discard those which are not valid.

You are strongly encouraged to either customize this script or obtain the original script, `wpoison.pl`, and modify customize it. The script itself is extremely straight-forward. Even those with very little knowledge of Perl should be able to understand and customize the code.

```
#!/usr/bin/perl -w

# wpoison.cgi - Version 1.8b
#
# For installation and usage instructions see http://www.monkeys.com/wpoison/
# and http://www.collinstarkweather.com/wpoison
#
# Original idea by Ronald F. Guilmette <rfg@monkeys.com>
# Code implemented by Ronald F. Guilmette <rfg@monkeys.com>
# Adapted to Slash Code by Collin Starkweather <collin.starkweather@collinstarkweather.com>
#
# Copyright (c) 2000,2001 Ronald F. Guilmette; All rights reserved.
#
# Redistribution and use in source form ONLY, with or without modification,
# is permitted provided that the following conditions are met:
#
# 1. Redistributions of source code must retain the above copyright
#    notice, this list of conditions and the following disclaimer.
# 2. All advertising materials mentioning features or use of this software
#    must display the following acknowledgement:
#    This product includes software developed by Ronald F. Guilmette.
# 3. Neither the name of Ronald F. Guilmette nor the names of other con-
#    tributors to this software may be used to endorse or promote products
#    derived from this software without specific prior written permission.
# 4. Either a copy of, or a link to the official Wpoison logo graphic (which
#    may be found at http://www.monkeys.com/wpoison/logo.gif) must be
#    included in, and clearly and legibly visible on the home page of any
#    web site which uses, employs, includes, or makes reference to this
#    software or any derivative or modified version thereof. Also, the
#    official Wpoison logo itself must be include in an HTML hyperlink
#    so that any usser clicking on any part of the logo image will be
```

```

#   directed/linked to the Wpoison home page at:
#
# http://www.monkeys.com/wpoison/
#
#   In order to satisfy this requirement, you may simply include the
#   following HTML code somewhere (anywhere) on your web site home page:
#
# <A HREF="http://www.monkeys.com/wpoison/">
# <IMG SRC="http://www.monkeys.com/wpoison/logo.gif">
# </A>
#
# Disclaimer
# -----
#
# THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS 'AS IS' AND
# ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
# ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
# OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
# LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
# OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
# SUCH DAMAGE.

use strict;
use FileHandle;
use POSIX;
use Template;
use Carp;
use CGI;
use vars qw($pname $tmp_words_file $num_cached_randwords @randwords @tl_domains_1 @tl_domains_2 $config);

$config = {
    POST_CHUNK => 1,
    EVAL_PERL => 0,
    START_TAG => '<!--%',
    END_TAG => '%-->';
};

$pname="wpoison";
$tmp_words_file = "/tmp/wpoison.words";
$num_cached_randwords = 4096;
my $creation_time;
my $age_in_seconds;
my $email_addr;
my $num_addresses;
my $num_links;
my $script_name = getenv ("SCRIPT_NAME") || $0;

my $uri_prefix = 'motel';
my $email_domain = 'foobar.com';

# @tl_domains_1 = (
# "com", "com", "com", "com",
# "net", "net", "net",
# "org", "org",
# "edu", "edu",
# "gov",
# "mil",
# "int");

# @tl_domains_2 = (
# "uk", "su",
# "af", "al", "dz", "as", "ad", "ao", "ai", "aq", "ag", "ar", "am", "aw", "au",
# "at", "az", "bs", "bh", "bd", "bb", "by", "be", "bz", "bj", "bm", "bt", "bo",
# "ba", "bw", "bv", "br", "io", "bn", "bg", "bf", "bi", "kh", "cm", "ca", "cv",
# "ky", "cf", "td", "cl", "cn", "cx", "cc", "co", "km", "cg", "ck", "cr", "ci",
# "hr", "cu", "cy", "cz", "dk", "dj", "dm", "do", "tp", "ec", "eg", "sv", "gq",
# "er", "ee", "et", "fk", "fo", "fj", "fi", "fx", "fx", "gf", "pf", "tf", "ga",
# "gm", "ge", "de", "gh", "gi", "gr", "gl", "gd", "gp", "gu", "gt", "gn", "gw",
# "gy", "ht", "hm", "hn", "hk", "hu", "is", "in", "id", "ir", "iq", "ie", "il",
# "it", "jm", "jp", "jo", "kz", "ke", "ki", "kp", "kr", "kw", "kg", "la", "lv",
# "lb", "ls", "lr", "ly", "li", "lt", "lu", "mo", "mk", "mg", "mw", "my", "mv",
# "ml", "mt", "mh", "mq", "mr", "mu", "yt", "mx", "fm", "md", "mc", "mn", "ms",
# "ma", "mz", "mm", "na", "nr", "np", "nl", "an", "nc", "nz", "ni", "ne", "ng",
# "nu", "nf", "mp", "no", "om", "pk", "pw", "pa", "pg", "py", "pe", "ph", "pn",
# "pl", "pt", "pr", "qa", "re", "ro", "ru", "rw", "kn", "lc", "vc", "ws", "sm",
# "st", "sa", "sn", "sc", "sl", "sg", "sk", "si", "sb", "so", "za", "gs", "es",
# "lk", "sh", "pm", "sd", "sr", "sj", "sz", "se", "ch", "sy", "tw", "tj", "tz",
# "th", "tg", "tk", "to", "tt", "tn", "tr", "tm", "tc", "tv", "ug", "ua", "ae",
# "gb", "us", "um", "uy", "uz", "vu", "va", "ve", "vn", "vg", "vi", "wf", "eh",
# "ye", "yu", "zr", "zm", "zw");

sub
death
{
    my ($message) = @_;

    print STDOUT ("Content-Type: text/html\n",

```

```

"\n",
"<HTML>\n",
"<HEAD><TITLE>WPOISON - Unexpected Error</TITLE></HEAD>\n",
"<BODY bgcolor=\"white\" text=\"#c00000\"><BIG><BIG>\n");

print "$pname: $message\n";

print (</BIG></BIG></BODY>\n</HTML>\n");
exit 0;
}

sub
gen_new_random_words_list
{
my $dictfile;
my $total_words;
my @words;
my %already_taken;

if (-f $tmp_words_file) {
if (not unlink ($tmp_words_file)) {
death ("Error unlinking file \"$tmp_words_file\": $!");
}
}

if (-f "/usr/dict/words") {
$dictfile = "/usr/dict/words";
} elsif (-f "/usr/share/dict/words") {
$dictfile = "/usr/share/dict/words";
} elsif (-f "words") {
$dictfile = "words";
} else {
death ("Cannot find dictionary file!");
}

death ("Error opening dictionary file \"$dictfile\": $!")
unless (open (DICTFILE, "<$dictfile");

$total_words = 0;
while (<DICTFILE>) {
chop;
push @words, $_;
$total_words++;
}

close DICTFILE;

death ("Error opening tmp words file \"$tmp_words_file\": $!")
unless (open (RWORDS, ">$tmp_words_file"));

# We will now pick $num_cached_randwords words at random

srand;

for (1..$num_cached_randwords) {
try_again:
my $rand_index = int (rand $total_words);

goto try_again if (defined $already_taken{$rand_index});
$already_taken{$rand_index} = 1;
print RWORDS "$words[$rand_index]\n";
}

close RWORDS;
}

sub
read_random_words
{
death ("Error opening tmp words file \"$tmp_words_file\": $!")
unless (open (RWORDS, "<$tmp_words_file"));

chop(@randwords = <RWORDS>);

close RWORDS;
}

sub
random_word
{
my $word_index;

$word_index = int (rand $num_cached_randwords);
return $randwords[$word_index];
}

sub
gen_random_words
{
my ($min_words, $max_words) = @_;
my $num_words;

```

```

my $word_index;
my $i;

my $return;
$num_words = $min_words + (rand ($max_words - $min_words));
for $i (1..$num_words) {
    $word_index = int (rand $num_cached_randwords);
    $return .= $randwords[$word_index];
    $return .= " " if ($i < $num_words);
}
return $return;
}

sub
gen_random_color
{
    my $red_code = int (rand 256);
    my $green_code = int (rand 256);
    my $blue_code = int (rand 256);

    return "%02x%02x%02x", $red_code, $green_code, $blue_code;
}

sub
random_letter
{
    return chr (unpack ("%c", 'a') + int (rand 26));
}

sub
random_domain
{
    my $rindex;

    if (int (rand 4) == 0) {
        $rindex = int (rand ($#t1_domains_2 + 1));
        return $t1_domains_2[$rindex];
    } else {
        $rindex = int (rand ($#t1_domains_1 + 1));
        return $t1_domains_1[$rindex];
    }
}

#autoflush STDOUT 1;

if (not -r $tmp_words_file) {
    gen_new_random_words_list ();
} else {
    $creation_time = (stat $tmp_words_file)[9];
    $age_in_seconds = time - $creation_time;
    gen_new_random_words_list () if ($age_in_seconds > (30 * 60));
}

read_random_words ();

my $vars = { death => 0 };

$vars->{'random_title'} = gen_random_words (1, 3);
$vars->{'random_words'} = gen_random_words (30, 60);

$vars->{'bgcolor'} = gen_random_color ();
$vars->{'text_color'} = gen_random_color ();
$vars->{'link_color'} = gen_random_color ();
$vars->{'vlink_color'} = gen_random_color ();

$vars->{'more_random_words'} = gen_random_words (10,30);

my @addresses;
$num_addresses = 2 + int (rand 16);
for (1..$num_addresses) {
    $email_addr = random_word ();
    $email_addr .= "@";
    $email_addr .= $email_domain;
    push @addresses, qq{<A HREF="mailto:$email_addr">$email_addr</A><BR>};
}

$vars->{'e_mail'} = join "\n", @addresses;

$vars->{'yet_more_random_words'} = gen_random_words (2, 20);

my @links;
$num_links = 1 + int (rand 16);
for (1..$num_links) {
    my $random_words = gen_random_words(1,9);
    $random_words =~ s/^\s+//;
    $random_words =~ s/\s+$//;
    my $random_links = join " ", map { qq{<A HREF="/$uri_prefix/${_}.html">${_}</A>} } split /\s+/, $random_words;
    $random_links .= "<BR>\n";
    print STDERR "Added [$random_links] to \@links ..\n";
    push @links, $random_links;
}

```

```

$vars->{'links'} = join "\n", @links;

my $random_word = random_word();
$vars->{'image_link'} = "/andy/${random_word}.html";
$vars->{'image_file'} = "/local/images/roaches.gif";

# Sleep for one second when finishing each page to avoid server overload.

sleep (1);

my $template = new Template ($config) ;
my $return;
my $html = join '', <DATA>;
$template->process(\$html,$vars,\$return)
|| confess $template->error();

my $cgi = new CGI;
print $cgi->header();
print $return;

__END__

<!--% IF death %--> <!-- Generate the error page ... -->

<BIG>
<BIG>
<!--% pname %--><!--% message %-->
</BIG>
</BIG>

<!--% ELSE %--> <!-- Generate a normal page ... -->

<FONT COLOR="#AA00AA"><!--% random_words %--></FONT>

<P>

<i>Welcome to the roach motel. This page will generate a variety of bogus
e-mail addresses which will then be picked up by spiders and bots used
by spammers to collect e-mail addresses from web pages.</i>

<P>

<i>Now some more random words to make this seem like a different page
each time it is loaded ...</i>

<P>

<FONT COLOR="#AA00AA"><!--% more_random_words %--></FONT>

<P>

<i>And now some bogus e-mail addresses:</i>

<P>

<!--% e_mail %-->

<P>

<i>And now some links that will merely redirect a user back
to another page containing bogus e-mail addresses. All the
e-mail addresses are valid (that is, they do not generate
bounces) but all of the e-mail delivered to them is discarded:</i>

<P>

<!--% links %-->

<P>

<FONT COLOR="#AA00AA"><!--% more_random_words %--></FONT>

<P>

<i>That is all there is to it!</i>

<P>

<FONT COLOR="#AA00AA"><!--% yet_more_random_words %--></FONT>

<P>

<!--% END %-->

<P>

```

B.2 wpoison.shtml

```
<HTML>
<HEAD>
<META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
<!--#include virtual="/slashhead.inc"-->
<LINK HREF="/local/styles/default.css" REL="StyleSheet" TYPE="text/css">
<!--#include virtual="/local/wpoison.cgi"-->
<!--#include virtual="/slashfoot.inc"-->
```

B.3 The mainmenu Template

```
- <A HREF="[ % constants.rootdir %]/about.shtml">About</A><BR>
- <A HREF="[ % constants.rootdir %]/faq.shtml">FAQ</A><BR>
- <A HREF="[ % constants.rootdir %]/comments.pl">Discussions</A><BR>
- <A HREF="[ % constants.rootdir %]/journal.pl">Journals</A><BR>
- <A HREF="[ % constants.rootdir %]/messages.pl">Messages</A><BR>
- <A HREF="[ % constants.rootdir %]/topics.pl">Topics</A><BR>
- <A HREF="[ % constants.rootdir %]/authors.pl">Authors</A><BR>
<HR NOSHADE SIZE="1">
- <A HREF="[ % constants.rootdir %]/users.pl?op=edithome">Preferences</A><BR>
- <A HREF="[ % constants.rootdir %]/search.pl">Older Stuff</A><BR>
- <A HREF="[ % constants.rootdir %]/pollBooth.pl">Past Polls</A><BR>
- <A HREF="[ % constants.rootdir %]/submit.pl">Submit Story</A><BR>
- <A HREF="[ % constants.rootdir %]/motel/foo.html">The Roach Motel</A><BR>
```

B.4 The footer Template

```
[% USE Slash %]
</TD></TR></TABLE><P>
<TABLE WIDTH="100%" BORDER="0" CELLPADDING="0" CELLSPACING="0">
<TR BGCOLOR="[ % user.bg.3 %]">
<TD COLSPAN="3"><IMG SRC="[ % constants.imagedir %]/pix.gif" WIDTH="1" HEIGHT="1"></TD>
</TR>
<TR BGCOLOR="[ % user.bg.1 %]">
<TD ALIGN="LEFT"><IMG SRC="[ % constants.imagedir %]/660000.gif" WIDTH="1" HEIGHT="60" ALT=""></TD>
<TD WIDTH="100%">
<TABLE WIDTH="100%" BORDER="0" CELLPADDING="5" CELLSPACING="0"><TR>
<!--
<TD>
<A HREF="http://slashdot.org/">
<IMG SRC="[ % constants.imagedir %]/sdlogo.gif" BORDER="0">
</A>
</TD>
-->
<TD WIDTH="30%">
<A HREF="/andy/random.html">
<IMG ALIGN="bottom" SRC="/local/images/some-random-image.gif" BORDER="0"
WIDTH="105" HEIGHT="40">
</A>
</TD>
<TD ALIGN="CENTER">
[% IF user.is_admin %]
[% Slash.currentAdminUsers() %]
[% ELSE %]
<FONT SIZE="2" FACE="[ % constants.mainfontface %]"><I>[% PROCESS motd %]</I></FONT>
[% END %]
</TD>
<!--
<TD>
<A HREF="http://slashcode.com/">
<IMG SRC="[ % constants.imagedir %]/slashlogo.gif" BORDER="0">
</A>
```

```

</TD>
-->
<TD ALIGN="RIGHT" WIDTH="30%">
  <A HREF="http://www.linux.com/">
    <IMG SRC="[ % constants.imagedir %]/damn-powered.gif" BORDER="0">
  </A>
</TD>
</TR></TABLE>
<TD>
<TD ALIGN="RIGHT"><IMG SRC="[ % constants.imagedir %]/660000.gif" WIDTH="1" HEIGHT="60" ALT=""></TD>
</TR>
<TR BGCOLOR="[ % user.bg.3 %]">
<TD COLSPAN="3"><IMG SRC="[ % constants.imagedir %]/pix.gif" WIDTH="1" HEIGHT="1"></TD>
</TR>
</TABLE>

<CENTER>
<FONT SIZE=2 COLOR="[ % user.bg.3 %]"> [
<A HREF="[ % constants.rootdir %]"/>home</A> |
<A HREF="[ % constants.rootdir %]/submit.pl">contribute story</A> |
<A HREF="[ % constants.rootdir %]/search.pl">older articles</A> |
<A HREF="[ % constants.rootdir %]/pollBooth.pl">past polls</A> |
<A HREF="[ % constants.rootdir %]/faq.shtml">faq</A> |
<A HREF="[ % constants.rootdir %]/authors.pl">authors</A> |
<A HREF="[ % constants.rootdir %]/users.pl">preferences</A> ]
</FONT>
</CENTER>
</BODY>
</HTML>

```

C Compiling this Document

```

latex wpoison-on-slashcode
latex wpoison-on-slashcode
bibtex wpoison-on-slashcode
latex wpoison-on-slashcode
# Create a PDF version of this document
pdfelatex wpoison-on-slashcode
# Create an HTML version of this document
latex2html -split 0 wpoison-on-slashcode.tex
# Create a text version of this document
lynx -dump wpoison-on-slashcode/wpoison-on-slashcode.html > wpoison-on-slashcode.txt

```