

Hacking PHP for Fun and Profit

Why write in PHP when you can write PHP?

By Justin Dearing

President Long Island PHP User Group

[<zippy1981@gmail.com>](mailto:zippy1981@gmail.com)

Hacking PHP for Fun

- If your at a PHP usergroup, you probably consider programming fun.
- Fixing the interpreter/compiler for your language gives you bragging rights

Hacking PHP for Profit

1) Hack PHP

2) ????????

3) PROFIT!!!!

Hacking PHP for indirect profit

- You probably discovered this bug in PHP at your day job
- Bragging rights on your resume
- Increased positive internet visibility
- Networking through speaking engagements

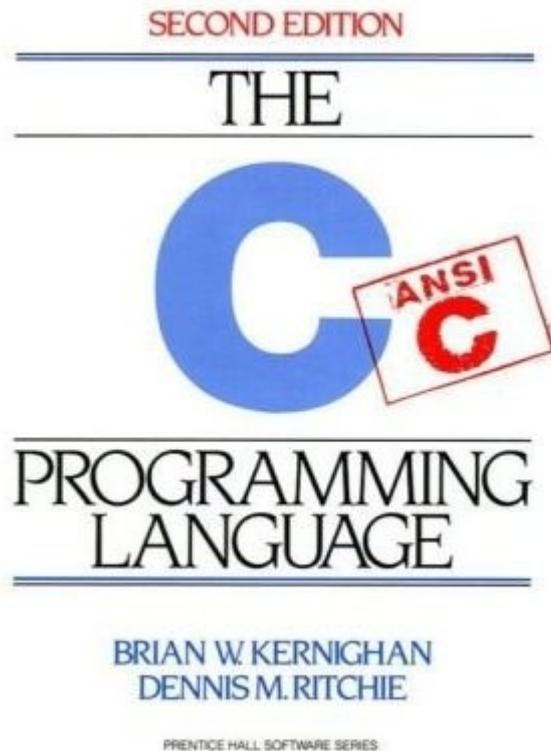
Goals

- Discuss how to file a good bug report or enhancement request
- Demonstrate how to build PHP on both Unix and Windows
- Demonstrate how to write and run a PHP test (in PHP).
- Demonstrate how to submit your test as a patch.
- Case Study of two patches I wrote.

Back story for this talk

- Once upon a time I had trouble consuming a .NET WCF Web Service with PHP.
- I filed [Bug #50698](#) and made a failed attempt to fix it. I then worked around the problem.
- A few months later I was laid off. :(
- I fixed the bug in my copious free time.
- In my still copious free time I asked the PHP devs what grunt work needed to be done.
- I worked on [Bug #47435](#)

For the non-C programmers



- › I can't make you a C programmer in one night.
- › Just being able to write a test, and follow the procedures for bug filing expedites the jobs of those that will fix the bug
- › If you want to learn C use the K&R book.

For the novice PHP programmer

- If you want to report a bug you probably already have code that reproduces the error.
- If you cannot produce test code for a feature request, you probably need to articulate your request to yourself better.
- Learning to write a good bug report is helpful to the process.

SPOILER ALERT!!!

```
Index: php_sdl.c
=====
--- php_sdl.c      (revision 297347)
+++ php_sdl.c      (working copy)
@@ -832,7 +832,12 @@
                                if (strncmp((char*)tmp->children->content,
WSDL_HTTP_TRANSPORT, sizeof(WSDL_HTTP_TRANSPORT)) == 0) {
                                soapBinding->transport =
SOAP_TRANSPORT_HTTP;
                                } else {
-                                soap_error1(E_ERROR, "Parsing WSDL:
PHP-SOAP doesn't support transport '%s'", tmp->children->content);
+                                // Since this is an E_NOTICE severity
message, it will disappear into the ether.
+                                soap_error1(E_NOTICE, "Parsing WSDL:
PHP-SOAP doesn't support transport '%s'", tmp->children->content);
+                                efree(soapBinding);
+                                efree(tmpbinding);
+                                trav = trav->next;
+                                continue;
                                }
                                }
@@ -1128,6 +1133,10 @@
        soap_error0(E_ERROR, "Parsing WSDL: Couldn't bind to service");
    }

+    if (ctx.sdl->bindings == NULL || ctx.sdl->bindings->nNumOfElements == 0) {
+        soap_error0(E_ERROR, "Parsing WSDL: Could not find any usable binding services in
WSDL.");
+    }
+
    zend_hash_destroy(&ctx.messages);
    zend_hash_destroy(&ctx.bindings);
    zend_hash_destroy(&ctx.portTypes);
```

Filing a good bug report

- First read ESR's [How To Ask Questions The Smart Way](#)
- I recommend using stackoverflow.com to verify your problem is a bug.
- Make it as easy as possible to reproduce the bug.
- Use github to store larger example cases.

Submitting the bug

- Go to <http://bugs.php.net/report.php>
- Fill out the form completely
- Test case here should not be a formal unit test, you need to “sell” the bug here.

Building PHP

This takes a while, especially on windows.
Make sure your bug is submitted before you
start this.

Building PHP on Windows

Complete directions at

<http://wiki.php.net/internals/windows/stepbystepbuild>

Note

Don't expect help on the mailing list if you deviate from the official build directions too much

Prerequisites

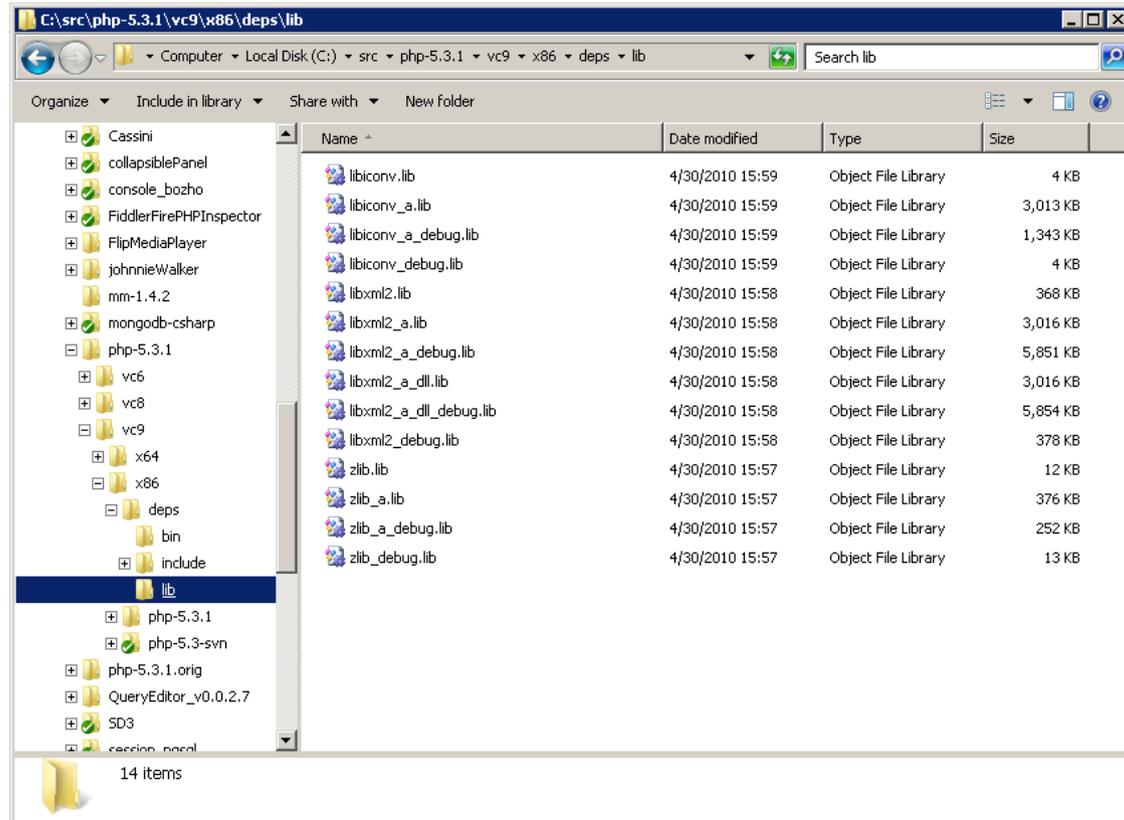
- Windows SDK 6.1 (earlier or later might not work)
- Visual C++ 2008 Express Edition
- [binary-tools.zip](#)
- Some compiled binary libraries from [this url](#)
- SVN
 - Its good to have both the command line version and TortoiseSVN

Build Prep

- Unzip the binary tools to c:\src\php-sdk
- Start the sdk command prompt
 - `setenv /x86 /xp /release`
 - `cd \src\php-sdk`
 - `bin\phpsdk_setvars.bat`
 - `bin\phpsdk_buildtree.bat php53dev`
 - `cd \php53dev\vc9\x86`
 - `start deps`

Build Prep (cont)

- In the explorer window you just opened copy all the dependent DLLs.



Build Prep (cont)

- From the same command prompt
 - svn co <http://svn.php.net/repository/php/php-src/branches/php-5.3-svn>
 - cd php-5.3-svn
 - buildconf
 - configure --disable-all --enable-cli --enable-cgi --enable-filter=shared --enable-libxml=shared --enable-soap=shared

Compiling

- From the same command prompt
 - nmake
 - nmake test

Building PHP on Unix

Build Prep

- Make sure you have gcc and friends installed
- You need autoconf **2.13!!**

Configuring and building

- Go to the shell
 - `PHP_AUTOCONF=autoconf213 ./buildconf`
 - `./configure --disable-all --enable-cli --enable-cgi --enable-filter=shared --enable-libxml=shared --enable-soap=shared`
 - `make`
 - `make test`

Testing

Now that we got all that out of the way.

Writing a good test

- Tests live in `tests/` and `ext/[module_name]/tests/`
- There is a module for everything, even `var_filter()`
- Tests are written in php and have the extension `.phpt`
- A test for a bug is named `bug[number].phpt` or `bug[number]_X.phpt`

My Tests

Adventures in Soap Mocking.

Running your test with run-tests.php

- Go back to your command prompt or shell
 - UNIX: `sapi/cli/php run-tests.php -p sapi/cli/php ext/[module]/bug[number]`
 - WINDOWS: `Release_TS\php run-tests.php -p Release-TS\php.exe ext\[module]\bug[number].phpt`

Submitting your test as a patch

- From your command prompt (adjust slashes for OS):
 - `svn add ext/[name]/tests/bug[number].phpt`
 - `svn diff ext/[name]/tests > bug[number]-tests.patch`
- Add patch to ticket
- Add comment on ticket with any notes

Fixing the Bug

This is where we write in C
Note: Please try this at home!!!

Case Study: Modifying SoapClient

Join my on the journey that I called bug
50698

Notes about Soap

- ext/soap contains SoapClient and SoapServer
- Soap depends on xml, libxml and iconv

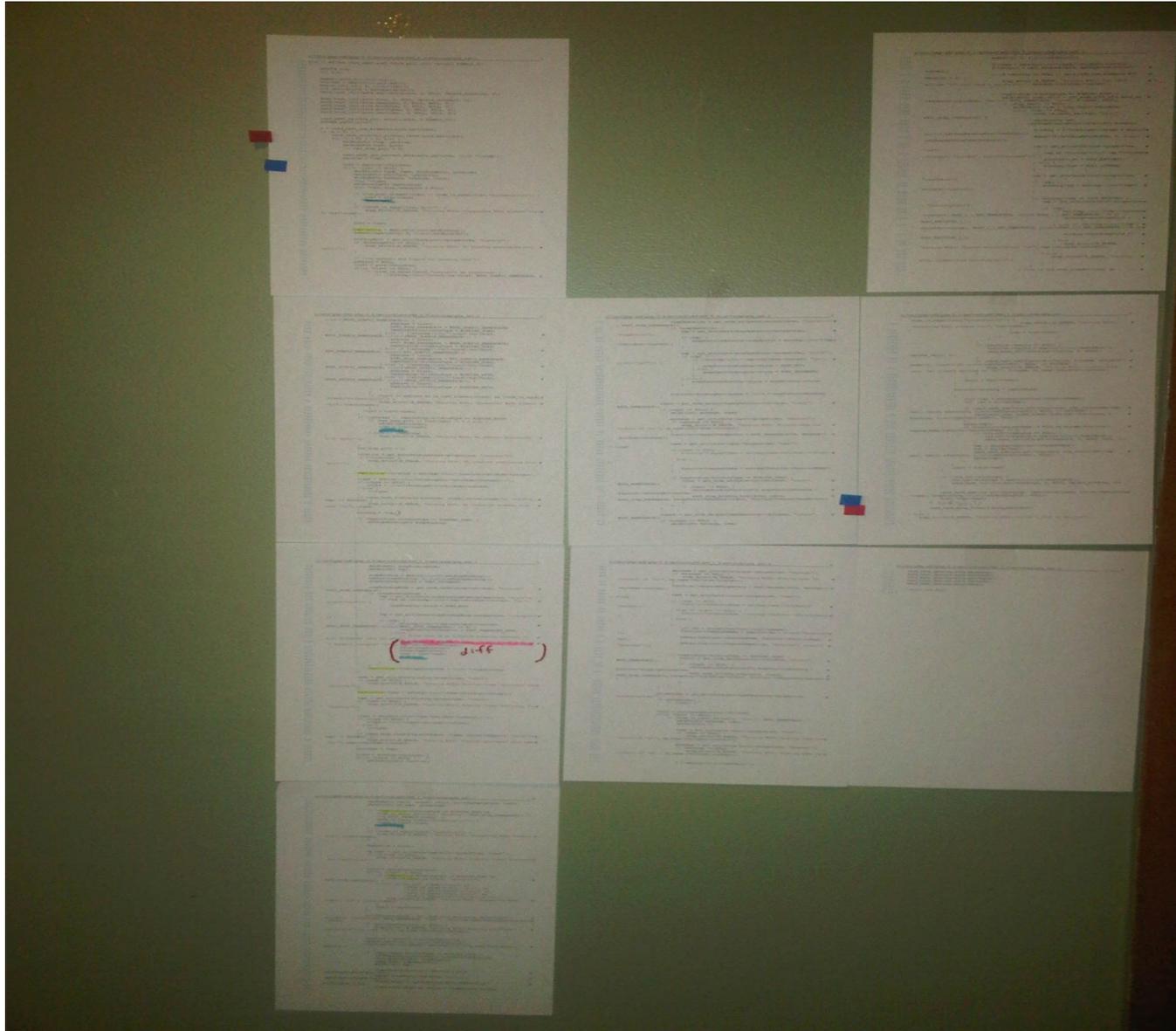
Editing Soap Module in VS

- The Soap module has a Visual Studio 6.0 project that can be imported by Visual Studio 2008
- If you configure the debugger to start php.exe, you can step through the soap code

Editing Soap Module in UNIX

- On UNIX I use GVIM to write C.
- Some files have vim annotations in them
- I don't know how to use gdb, or any unix debugger.

Advanced Debugging Techniques



Advanced debugging techniques (cont)

- I consider myself pretty adept with step through debuggers.
- The function wasn't badly written. Sometimes a subroutine is just that long.
- Comments would have helped
- The resulting patch was less than 10 lines of new code
- YMMV. Code in my second bug fix was smaller.

Recompiling and Retesting

- Just rerun `nmake/make` and `run-tests.php`
- Step through with a debugger if needed
- Rinse and repeat until tests pass

Final Soap Patch

```
Index: php_sdl.c
=====
--- php_sdl.c      (revision 297347)
+++ php_sdl.c      (working copy)
@@ -832,7 +832,12 @@
                                if (strncmp((char*)tmp->children->content,
WSDL_HTTP_TRANSPORT, sizeof(WSDL_HTTP_TRANSPORT)) == 0) {
                                soapBinding->transport =
SOAP_TRANSPORT_HTTP;
                                } else {
-                               soap_error1(E_ERROR, "Parsing WSDL:
PHP-SOAP doesn't support transport '%s'", tmp->children->content);
+                               // Since this is an E_NOTICE severity
message, it will disappear into the ether.
+                               soap_error1(E_NOTICE, "Parsing WSDL:
PHP-SOAP doesn't support transport '%s'", tmp->children->content);
+                               efree(soapBinding);
+                               efree(tmpbinding);
+                               trav = trav->next;
+                               continue;
                                }
                                }
@@ -1128,6 +1133,10 @@
        soap_error0(E_ERROR, "Parsing WSDL: Couldn't bind to service");
    }

+    if (ctx.sdl->bindings == NULL || ctx.sdl->bindings->nNumOfElements == 0) {
+        soap_error0(E_ERROR, "Parsing WSDL: Could not find any usable binding services in
WSDL.");
+    }
+
    zend_hash_destroy(&ctx.messages);
    zend_hash_destroy(&ctx.bindings);
    zend_hash_destroy(&ctx.portTypes);
```

Dmitry's Version

```
Index: php_sdl.c
=====
--- php_sdl.c (revision 299012)
+++ php_sdl.c (revision 299013)
@@ -832,7 +832,12 @@
                                     if (strcmp((char*)tmp->children->content, WSDL_HTTP_TRANSPORT,
sizeof(WSDL_HTTP_TRANSPORT)) == 0) {
                                         soapBinding->transport = SOAP_TRANSPORT_HTTP;
                                     } else {
-                                     soap_error1(E_ERROR, "Parsing WSDL: PHP-SOAP doesn't support
transport '%s'", tmp->children->content);
+                                     /* try the next binding */
+                                     efree(soapBinding);
+                                     efree(tmpbinding->location);
+                                     efree(tmpbinding);
+                                     trav = trav->next;
+                                     continue;
                                     }
                                 }
@@ -1128,6 +1133,10 @@
         soap_error0(E_ERROR, "Parsing WSDL: Couldn't bind to service");
    }

+    if (ctx.sdl->bindings == NULL || ctx.sdl->bindings->nNumOfElements == 0) {
+        soap_error0(E_ERROR, "Parsing WSDL: Could not find any usable binding services in WSDL.");
+    }
+
    zend_hash_destroy(&ctx.messages);
    zend_hash_destroy(&ctx.bindings);
    zend_hash_destroy(&ctx.portTypes);
```

Submitting your fix

- Submit your fix as a separate patch.
- Add a comment to the bug explaining what needs to be explained.

Case Study Duex: IPv6 filtering

- I was growing impatient with a lack of feedback on my original patch, and still unemployed.
- I asked what grunt-work needed to be done. Bug 47435 was the answer.
- I didn't know anything about Ipv6 so I had a little help from my friends.

Lessons learned from IPv6

- The army is right, never volunteer
- If you ask for grunt-work, you will get what other people consider grunt-work
 - Corollary: If the bug was straightforward, it would have been done already.
- Seek the opinion of SME's you know in real life.
 - Offer to suffer the punishments of their good deeds. They will probably decline.

Conclusions

- Contributing PHP is fun, and good for the resume.
- There's plenty of work to be done for the non C programmer
- You need a lot of tribal knowledge to contribute to PHP. This is the biggest sunk cost to fixing your first PHP bug.

Getting these slides

Go to

<http://github.com/zippy1981/TalkMaterials/tree/master>

Look on #NYPHP on twitter

FIN

Hacking PHP for Fun and Profit

Why write in PHP when you can write PHP?

By Justin Dearing

President Long Island PHP User Group

<zippy1981@gmail.com>

Hacking PHP for Fun

- If your at a PHP usergroup, you probably consider programming fun.
- Fixing the interpreter/compiler for your language gives you bragging rights

Hacking PHP for Profit

- 1) Hack PHP
- 2) ????????
- 3) PROFIT!!!!

Hacking PHP for indirect profit

- You probably discovered this bug in PHP at your day job
- Bragging rights on your resume
- Increased positive internet visibility
- Networking through speaking engagements

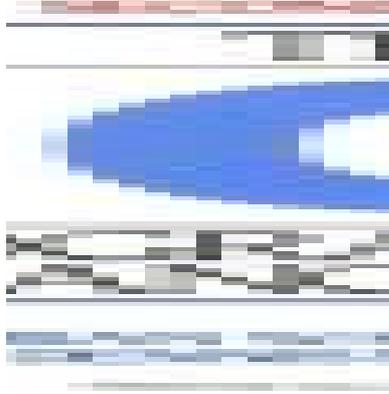
Goals

- Discuss how to file a good bug report or enhancement request
- Demonstrate how to build PHP on both Unix and Windows
- Demonstrate how to write and run a PHP test (in PHP).
- Demonstrate how to submit your test as a patch.
- Case Study of two patches I wrote.

Back story for this talk

- Once upon a time I had trouble consuming a .NET WCF Web Service with PHP.
- I filed [Bug #50698](#) and made a failed attempt to fix it. I then worked around the problem.
- A few months later I was laid off. :(
- I fixed the bug in my copious free time.
- In my still copious free time I asked the PHP devs what grunt work needed to be done.
- I worked on [Bug #47435](#)

For the non-C programmers



- I can't make you a C programmer in one night.
- Just being able to write a test, and follow the procedures for bug filing expedites the jobs of those that will fix the bug
- If you want to learn C use the K&R book.

For the novice PHP programmer

- If you want to report a bug you probably already have code that reproduces the error.
- If you cannot produce test code for a feature request, you probably need to articulate your request to yourself better.
- Learning to write a good bug report is helpful to the process.

Filing a good bug report

- First read ESR's [How To Ask Questions The Smart Way](#)
- I recommend using stackoverflow.com to verify your problem is a bug.
- Make it as easy as possible to reproduce the bug.
- Use github to store larger example cases.

Submitting the bug

- Go to <http://bugs.php.net/report.php>
- Fill out the form completely
- Test case here should not be a formal unit test, you need to “sell” the bug here.

Building PHP

This takes a while, especially on windows.
Make sure your bug is submitted before you
start this.

Building PHP on Windows

Complete directions at

<http://wiki.php.net/internals/windows/stepbystepbuild>

Note

Don't expect help on the mailing list if you deviate from the official build directions too much

Prerequisites

- Windows SDK 6.1 (earlier or later might not work)
- Visual C++ 2008 Express Edition
- [binary-tools.zip](#)
- Some compiled binary libraries from [this url](#)
- SVN
 - Its good to have both the command line version and TortoiseSVN

Build Prep

- Unzip the binary tools to c:\src\php-sdk
- Start the sdk command prompt
 - setenv /x86 /xp /release
 - cd \src\php-sdk
 - bin\phpsdk_setvars.bat
 - bin\phpsdk_buildtree.bat php53dev
 - cd \php53dev\vc9\x86
 - start deps

Build Prep (cont)

- From the same command prompt
 - svn co <http://svn.php.net/repository/php/php-src/branches/php-5.3-svn>
 - cd php-5.3-svn
 - buildconf
 - configure --disable-all --enable-cli --enable-cgi --enable-filter=shared --enable-libxml=shared --enable-soap=shared

Compiling

- From the same command prompt
 - nmake
 - nmake test

Building PHP on Unix

Click to add text

20

I don't go into many details here because unix folks tend to know how to compile their own software. Its a sad fact that many bright talented windows programmers and sysadmins just don't take the time to learn how to navigate a windows command prompt. I think this is changing for many reasons.

A second reason I don't go into many details is because other than the “buildconf” script, its just a matter of “./configure {litany of arguments} && make && make install”

Build Prep

- Make sure you have gcc and friends installed
- You need autoconf **2.13!!**

21

From what I gather from my occasional need to run autoconf, autoconf 2.65 is buggy

On another note, I am by no means an autoconf expert. I made one failed attempt to modify an autoconf script, back when I was young, stupid, and had too much time on my hands. However I am of the ignorant opinion that ERS is absolutely right when he says **autoconf is hopelessly broken**.

Configuring and building

- Go to the shell
 - `PHP_AUTOCONF=autoconf213 ./buildconf`
 - `./configure --disable-all --enable-cli --enable-cgi --enable-filter=shared --enable-libxml=shared --enable-soap=shared`
 - `make`
 - `make test`

Testing

Now that we got all that out of the way.

Writing a good test

- Tests live in tests/ and ext/[module_name]/tests/
- There is a module for everything, even var_filter()
- Tests are written in php and have the extension .phpt
- A test for a bug is named bug[number].phpt or bug[number]_X.phpt

My Tests

Adventures in Soap Mocking.

25

I'm going to break from my tradition of entirely self contained slide decks, and not include my tests here. The tests will be included in a separate file.

Running your test with run-tests.php

- Go back to your command prompt or shell
 - UNIX: `sapi/cli/php run-tests.php -p sapi/cli/php ext/[module]/bug[number]`
 - WINDOWS: `Release_TS\php run-tests.php -p Release-TS\php.exe ext\[module]\bug[number].phpt`

Submitting your test as a patch

- From your command prompt (adjust slashes for OS):
 - `svn add ext/[name]/tests/bug[number].phpt`
 - `svn diff ext/[name]/tests > bug[number]-tests.patch`
- Add patch to ticket
- Add comment on ticket with any notes

Fixing the Bug

This is where we write in C
Note: Please try this at home!!!

Case Study: Modifying SoapClient

Join my on the journey that I called bug
50698

Notes about Soap

- ext/soap contains SoapClient and SoapServer
- Soap depends on xml, libxml and iconv

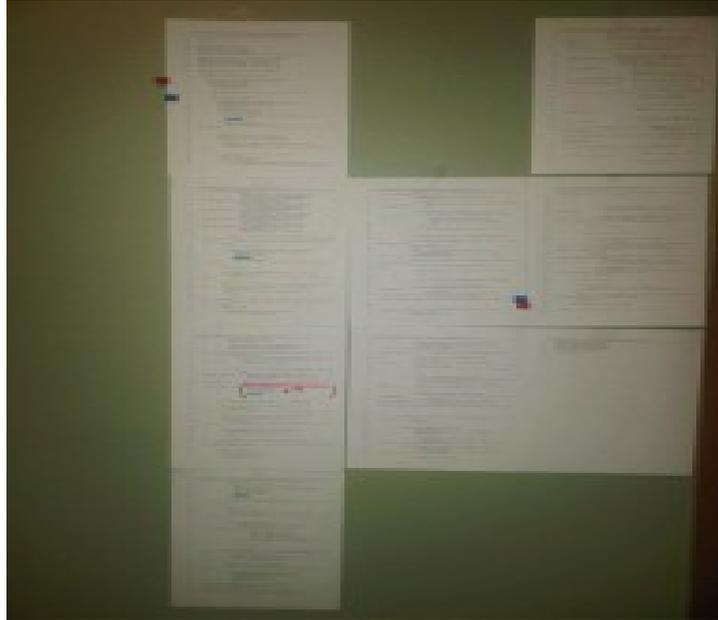
Editing Soap Module in VS

- The Soap module has a Visual Studio 6.0 project that can be imported by Visual Studio 2008
- If you configure the debugger to start php.exe, you can step through the soap code

Editing Soap Module in UNIX

- On UNIX I use GVIM to write C.
- Some files have vim annotations in them
- I don't know how to use gdb, or any unix debugger.

Advanced Debugging Techniques



These 9 pages on my fiancee's apartment wall are one C function. Much of it seems to be validating the WSDL in ways that XSD validation would do.

I needed to do this to know where to put extra error handling for the new possible error conditions I was introducing.

Since this is C I had to clean up memory so I also had to see what variables were dynamically allocated so I could free them.

Advanced debugging techniques (cont)

- I consider myself pretty adept with step through debuggers.
- The function wasn't badly written. Sometimes a subroutine is just that long.
- Comments would have helped
- The resulting patch was less than 10 lines of new code
- YMMV. Code in my second bug fix was smaller.

Recompiling and Retesting

- Just rerun nmake/make and run-tests.php
- Step through with a debugger if needed
- Rinse and repeat until tests pass

Final Soap Patch

```
Index: php_sdl.c
-----
--- php_sdl.c      (revision 297347)
+++ php_sdl.c      (working copy)
@@ -832,7 +832,12 @@
WSDL_HTTP_TRANSPORT, sizeof(WSDL_HTTP_TRANSPORT)) == 0) {
    soapBinding->transport =
SOAP_TRANSPORT_HTTP;
    } else {
        soap_error1(E_ERROR, "Parsing WSDL:
- PHP-SOAP doesn't support transport '%s'", tmp->children->content);
+ // Since this is an E_NOTICE severity
+ message, it will disappear into the ether.
        soap_error1(E_NOTICE, "Parsing WSDL:
+ PHP-SOAP doesn't support transport '%s'", tmp->children->content);
+
+         efree(soapBinding);
+         efree(tmpbinding);
+         trav = trav->next;
+         continue;
    }
}
@@ -1128,6 +1133,10 @@
    soap_error0(E_ERROR, "Parsing WSDL: Couldn't bind to service");
}
+
+ if (ctx.sdl->bindings == NULL || ctx.sdl->bindings->nNumOfElements == 0) {
+     soap_error0(E_ERROR, "Parsing WSDL: Could not find any usable binding services in
WSDL.");
+ }
+
zend_hash_destroy(&ctx.messages);
zend_hash_destroy(&ctx.bindings);
zend_hash_destroy(&ctx.portTypes);
```

Dmitry's Version

```
Index: php_sdl.c
-----
--- php_sdl.c (revision 299012)
+++ php_sdl.c (revision 299013)
@@ -832,7 +832,12 @@
sizeof(WSDL_HTTP_TRANSPORT) == 0) {
    soapBinding->transport = SOAP_TRANSPORT_HTTP;
    } else {
-      soap_error1(E_ERROR, "Parsing WSDL: PHP-SOAP doesn't support
transport '%s'", tmp->children->content);
+      /* try the next binding */
+      efree(soapBinding);
+      efree(tmpbinding->location);
+      efree(tmpbinding);
+      trav = trav->next;
+      continue;
    }
    }
}
@@ -1128,6 +1133,10 @@
    soap_error0(E_ERROR, "Parsing WSDL: Couldn't bind to service");
}
+
+ if (ctx.sdl->bindings == NULL || ctx.sdl->bindings->numOfElements == 0) {
+   soap_error0(E_ERROR, "Parsing WSDL: Could not find any usable binding services in WSDL.");
+ }
+
zend_hash_destroy(&ctx.messages);
zend_hash_destroy(&ctx.bindings);
zend_hash_destroy(&ctx.portTypes);
```

Submitting your fix

- Submit your fix as a separate patch.
- Add a comment to the bug explaining what needs to be explained.

Case Study Duex: IPv6 filtering

- I was growing impatient with a lack of feedback on my original patch, and still unemployed.
- I asked what grunt-work needed to be done. Bug 47435 was the answer.
- I didn't know anything about Ipv6 so I had a little help from my friends.

Lessons learned from IPv6

- The army is right, never volunteer
- If you ask for grunt-work, you will get what other people consider grunt-work
 - Corollary: If the bug was straightforward, it would have been done already.
- Seek the opinion of SME's you know in real life.
 - Offer to suffer the punishments of their good deeds. They will probably decline.

40

It took me a while to understand exactly what I had to fix for this bug. This is mainly because I have never used Ipv6. My approach here was that of a consultant serving a client. I was given a set of requirements, clarified them, and executed them.

Partial Ipv6 patch

```
Index: extfilter/logical_filters.c
-----
+++ extfilter/logical_filters.c (revision 297627)
+++ extfilter/logical_filters.c (working copy)
@@ -452,4 +552,7 @@
+
+ int          s[4];
+ int          mode;
+ int          value_string;
+
+ if (memcmp(Z_STRVAL_P(value), "", Z_STRELEN_P(value))) {
+     mode = FORMAT_IPV6;
+
+     } Check flags ?'
+     if (flags & FILTER_FLAG_NO_PRIV_RANGE) {
+         if (Z_STRELEN_P(value) == 2 && !memcmp("FC", Z_STRVAL_P(value), 2) || !memcmp("D0", Z_STRVAL_P(value), 2)) {
+             /* matches with the value "FC00::" as a private range.
+             */
+             if (Z_STRELEN_P(value) == 2 &&&
+                 !memcmp("FC", Z_STRVAL_P(value), 2) ||
+                 !memcmp("D0", Z_STRVAL_P(value), 2))
+             )
+             {
+                 RETURN_VALIDATION_FAILED;
+             }
+         }
+     }
+     if (flags & FILTER_FLAG_NO_RES_RANGE) {
+         if (
+             /* documentation address (RFC2001)
+             */
+             (Z_STRELEN_P(value) == 2 &&& !memcmp("FC", Z_STRVAL_P(value), 2)) ||
+             /* IPv6 address (RFC2001)
+             */
+             (Z_STRELEN_P(value) == 6 &&& !memcmp("::", Z_STRVAL_P(value), 6)) ||
+             /* IPv6 address (RFC2001)
+             */
+             (Z_STRELEN_P(value) == 3 &&& !memcmp("::1", Z_STRVAL_P(value), 3)) ||
+             /* IPv6 address (RFC2001)
+             */
+             (Z_STRELEN_P(value) == 7 &&& !memcmp("::128", Z_STRVAL_P(value), 7)) ||
+             /* IPv6 link local unicast (RFC4291)
+             */
+             (Z_STRELEN_P(value) == 5 &&&
+                 !memcmp("fe8", Z_STRVAL_P(value), 3) ||
+                 !memcmp("fdb", Z_STRVAL_P(value), 3) ||
+                 !memcmp("fdb7", Z_STRVAL_P(value), 3) ||
+                 !memcmp("fdb7", Z_STRVAL_P(value), 3))
+             )
+         )
+         {
+             /* documentation address (RFC2001)
+             */
+             (Z_STRELEN_P(value) == 3 &&& !memcmp("2001:0db8", Z_STRVAL_P(value), 9)) ||
+             /* Ebone
+             */
+             (Z_STRELEN_P(value) == 2 &&& !memcmp("2F", Z_STRVAL_P(value), 2)) ||
+             (Z_STRELEN_P(value) == 4 &&& !memcmp("309", Z_STRVAL_P(value), 4)) ||
+             /* TODO IPv4 Mapping (RFC2001)
+             */
+             (Z_STRELEN_P(value) == 8 &&& !memcmp("2001:001", Z_STRVAL_P(value), 8)) ||
+             /* IPv6 unicast route address
+             */
+             (Z_STRELEN_P(value) == 4 &&& !memcmp("::", Z_STRVAL_P(value), 4)) ?
+             )
+         {
+             RETURN_VALIDATION_FAILED;
+         }
+     }
+     }
+     break;
+ }
+ }
```

Conclusions

- Contributing PHP is fun, and good for the resume.
- There's plenty of work to be done for the non C programmer
- You need a lot of tribal knowledge to contribute to PHP. This is the biggest sunk cost to fixing your first PHP bug.

Getting these slides

Go to

<http://github.com/zippy1981/TalkMaterials/tree/master>

Look on #NYPHP on twitter

FIN