# AGI + PHP

"Making phones jump through fiery hoops of death."

Rob Peck
dealnews.com
PHP Appalachia 2008

# Introduction to VoIP

- At its simplest, sending telephone calls over an IP network.
- No magic. “It’s just software.”
- Voice packets are treated just like any other packet on a network. Broken down, transmitted, and reassembled.

# Do It Open Source!

- Asterisk is an open-source (GPL) software PBX (Private Branch Exchange) that was created by Mark Spencer.
- Basic Asterisk software includes many features available in proprietary PBX systems.
- Runs on ordinary PC hardware.

# How Asterisk Works

- extensions.conf is the config file that controls how Asterisk handles calls. It is called the "dialplan."

- Commands are sequentially executed based on the status from the previous command.

```
exten => _1XXX,1,Answer()
exten => _1XXX,n,Wait(2)
```

# I Need Something Else!

- Asterisk can be extended to do almost anything using AGI, or the Asterisk Gateway Interface.
- You can write scripts for AGI in any language (Perl, Python, C, etc). My favorite, of course, is PHP.

# Anatomy of an AGI call

1. Dialplan calls AGI
2. AGI reads variables from Asterisk
3. AGI sends commands to Asterisk
4. AGI reads response from Asterisk
5. Repeat 3-4 as necessary.
6. ???
7. Profit!
8. AGI exits

# How does AGI work?

- AGI reads data from and writes data to Asterisk, much the same way CGI would do with a web server.
- When Asterisk starts an AGI script, it feeds the channel variables to the script on standard input...

# Reading From Asterisk

- Every time an AGI script executes, Asterisk passes a number (about 20) values to the script.
- These AGI headers take the form of "key: value", one per line separated with a line feed, concluding with a blank line.

# Look! Some Code!

```
1.  ob_implicit_flush(true);
2.  set_time_limit(6);
3.  $in = fopen("php://stdin","r");
4.  $stdlog = fopen("php://stderr", "w");
5.
6.  while ($env=read()) {
7.      $s = split(": ",$env);
8.      $key = str_replace("agi_","",$s[0]);
9.      $value = trim($s[1]);
10.     $_AGI[$key] = $value;
11.     if (($env == "") || ($env == "\n")) {
12.         break;
13.     }
14. }
15.
16. function read() {
17.     global $in, $debug, $stdlog;
18.     $input = str_replace("\n", "", fgets($in, 4096));
19.     return $input;
20. }
```

# What can I do with it?

- You now have data from the server to interact with. This may include caller ID, channel, extension, etc, in an $_AGI array.
- With this you can interact with databases, write logs, route calls. Pretty much anything you can think of.

# Interacting

- Writing data back to the channel.
- Sending commands to stdout in the form of AGI commands.
- There are around 30 native AGI commands, and hundreds of apps.
- Commands return data in the form:
  200 result=x

# AGI Commands

- ANSWER - Answers a channel
- GET DATA - Gets digits
- SAY * - Reads data back to the channel.
- SET * - Sets various options on a channel.
- VERBOSE -  Sends data to the CLI.
- EXEC - Executes a dialplan app.

# Commands vs. Apps

- Applications ("apps") are functions that can be run in the Asterisk dialplan itself, or in an AGI via EXEC.
- Commands are AGI specific actions that can only be executed from within an AGI.

# Interacting

```
1.  function write($line) {
2.      global $debug, $stdlog;
3.      echo $line."\n";
4.  }
5.
6.  function execute($command) {
7.      global $in, $out, $debug, $stdlog;
8.      write($command);
9.      $data = fgets($in, 4096);
10.     if (preg_match("/^([0-9]{1,3}) (.*)/", $data, $matches)) {
11.         if (preg_match('/^result=([0-9a-zA-Z]*)( ?\((.*)\))?$/',
    $matches[2], $match)) {
12.             $arr['code'] = $matches[1];
13.             $arr['result'] = $match[1];
14.             if (isset($match[3]) && $match[3]) {
15.                 $arr['data'] = $match[3];
16.             }
17.             return $arr;
18.         } else return 0;
19.     } else return -1;
20. }
```

# Simple AGI

```
1.      #!/usr/bin/php
2.      <?php
3.      include "agi.php";
4.
5.      execute("SAY DATETIME #");
6.
7.      ?>
```

# More Complex

```
1.    #!/usr/bin/php
2.    <?php
3.
4.    include "agi.php" ;
5.
6.    $db=mysql_connect('redacted', 'redacted', 'redacted');
7.
8.    if(!$db) {
9.        verbose("Could not connect to DB!");
10.       exit(1);
11.   }
12.
13.   if(!mysql_select_db('redacted', $db)) {
14.       verbose("Could not use DB!");
15.       exit(1);
16.   }
17.
18.   $res=mysql_query(sprintf("select substitution_name from cid_substitution where
      from_number='%s'",mysql_escape_string($_AGI['callerid'])));
19.
20.   $result=mysql_fetch_row($res);
21.
22.   if(!empty($result)) {
23.       execute(sprintf("SET CALLERID \"%s <%s>\"",$result[0],$_AGI['callerid']));
24.   }
25.
26.   mysql_close($db);
27.
28.   ?>
```

# “Calling” your AGI

```
1.exten => 1000,1,Wait(2)
2.exten => 1000,n,AGI(test.php)
```

# AGI Execution

- AGI() - simple AGI execution.
- EAGI() - provides access to the audio channel.
- FastAGI() - Allows AGI execution on remote server(s).
- DeadAGI() - Allows AGI access to a “dead” (hungup) channel.

# Passing Variables

- Two ways to pass AGI variables to the script: channel variables and command-line arguments.
- Channel variables set from the dialplan.

```
exten => 1000,n,Set(VAR=1)
```

- Read it from within the script:

```
$var = execute("get variable VAR");
```

# Passing Variables

- Command line variables called from the AGI command.

  exten => 1000,n,AGI(test.php|test)

- These are available via $argv just like a shell script.

# Passing Variables

- AGI scripts can pass variables back to Asterisk for use in the dialplan.
- This is accomplished by calling the SET VARIABLE command to set the variable on the channel.
- Variables can then be read in dialplan applications.

# Common Problems

- AGI scripts live in /var/lib/asterisk/agi-bin
- PHP script must be executable by the user Asterisk is running as (usually “asterisk”).

# Debugging your AGI

- No way around it, debugging an AGI is a pain.
- One shortcut is to use AGI's "verbose" command to output text to the Asterisk CLI. You can then connect to the Asterisk CLI and watch calls proceed.
- Softphones are a lifesaver.

# AGI Debugging

- “AGI DEBUG” within the CLI will show what is sent and received from each AGI command.

# Beyond AGI

- Asterisk Manager Interface (AMI) allows you to interact with Asterisk outside of a call.
- Callfiles allow you to easily originate a call by dropping a specially formatted file into a directory.

# Summary

- AGI allows you to extend Asterisk using any language of your choice.
- AGI scripts may read and write only once, or many times. Read from stdin, write to stdout.
- Allows you to construct complicated actions that interact with outside data.

# Slides, Notes, Etc.

- Slides (will be on my blog): http://codelemur.wordpress.com
- Asterisk: http://www.asterisk.org
- Lots of info ("the wiki"): http://www.voip-info.org