HTTP sinkholing in a service provider environment

With a short introduction to PHP malware obfuscation

Marco d'Itri

`<md@seeweb.it>`

@rfc1036

Seeweb s.r.l.

Seminario AIPSI - 18 maggio 2023

Content

1. Introduction
2. The implementation
3. The benefits
4. Common PHP obfuscation methods

Who is Marco

Marco d'Itri

- Involved in Internet things in Italy since the mid '90s (we may have met on Usenet...).
- A Debian Developer for 25 years (mutt, inn, ppp, netbase, hotplug, udev, systemd...).
- I also wrote the `whois` command used by all Linux distributions.
- Employed by Seeweb, an italian cloud infrastructure, hosting and colocation provider.
- Designed and manages the Seeweb network (and other services).
- Designed and manages the Seeweb SOC.

What is Seeweb?

A cloud services provider in Italy

- 1998: founded as a pure hosting provider, after an experience as an ISP.
- 2005: opens a second data center in the Via Caldera Campus in Milano.
- 2010: first in Italy to provide cloud infrastructure.
- 2015: creates DHH S.p.A., a company listed on the Milano stock exchange which invests in cloud computing companies in the emerging markets of Europe.

Seeweb owns 4 data centers in Frosinone and Milano.

DHH is also present in Switzerland, Slovenia, Croatia, Serbia and Bulgaria.

seeweb

Content

1. Introduction
2. The implementation
3. The benefits
4. Common PHP obfuscation methods

What is a sinkhole?

Sinkholing: diverting to your own servers the communications from malware to their command and control (C&C) servers.

The Seeweb sinkhole

- HTTP traffic to specific IP addresses.
- Custom answers for specific DNS queries.

HTTP requests from web malware in the Seeweb network are redirected to our server for logging.
And then we send our own answers.

What is being sinkholed?

What kind of request are sinkholed?

- Downloading the second stages of malware.
- Exfiltrating credentials or requesting data from a C&C.

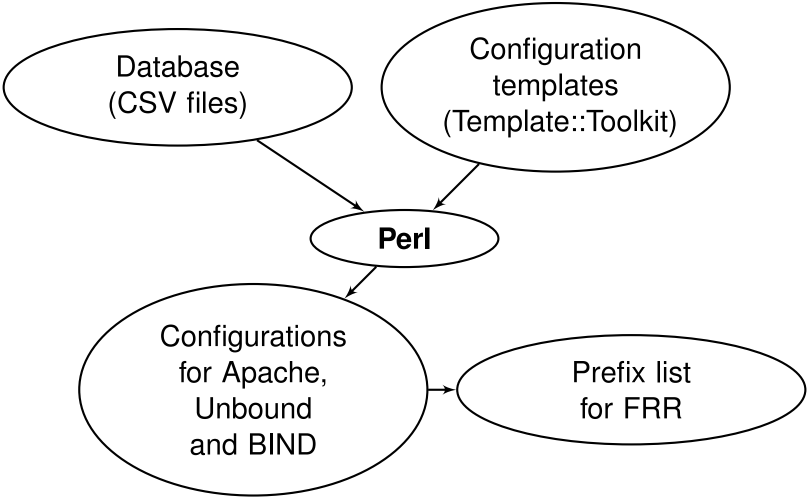
How are the addresses collected?

- Reverse engineering malware.
- Analysis of the resolvers' cache dumps.
- Passive DNS (more subdomains of already known domains).
- DNS enumeration.
- Looking at `/server-status` on large servers.

Content

1. Introduction
2. The implementation
3. The benefits
4. Common PHP obfuscation methods

The components



The database

The database of sinkholed addresses is organized in CSV files.

- IP
- Domain/URL (if known)
- Currently active
- Campaign name
- Type of payload (shell, PHP, Javascript, etc...)
- Expected reply (for validation)
- Notes

Apache configuration (1)

Answer the malware requests in a creative way:

```
<VirtualHost [% http_bind_list %] >
        ServerName sinkhole.seeweb.it

        ErrorLog /var/log/apache2/sinkhole/error.log
        CustomLog /var/log/apache2/sinkhole/[% log_name.$campaign ||          ↪
        log_name.default %].log sinklog

        DocumentRoot /var/www/sinkhole/
```

...

Each virtual host listens on up to hundreds of IP addresses.

Apache configuration (2)

...

```
        RewriteCond %{REQUEST_METHOD} =POST
        RewriteRule . /cgi-bin/log-post-data [L,PT]

        RewriteRule \.js$ /logger.js [L]
        RewriteRule \.sh$ /logger [L]

        # probably it will be run in a shell
        RewriteCond %{HTTP_USER_AGENT} ^(curl|Wget)/
        RewriteRule ^/ /logger [L]

        # or else probably it will be executed as php
        RewriteCond %{REQUEST_METHOD} =GET
        RewriteRule . /logger.php [L]
</VirtualHost>
```

The logger

This is the typical reply sent to the malware:

```php
<?php
$msg  = "PROGRAM: php\n";
$msg .= "CWD: " . getcwd() . "\n";
$msg .= "PID: " . getmypid() . "\n";
$msg .= "USER: " . get_current_user() . "\n";
$msg .= "\n";

foreach ($_SERVER as $var => $value) {
          $msg .= "$var=$value\n";
}

mail("soc+magic@seeweb.it", "sinkhole report", $msg);
```

There are also a shell version and other minor variations.

FRR configuration

Announce the sinkholed addresses with BGP (the same script configures them on a local interface):

```
router bgp 12637
 neighbor CORE peer-group
 neighbor CORE remote-as 12637
 neighbor 192.0.2.1 peer-group CORE
 neighbor 192.0.2.2 peer-group CORE
 !
 address-family ipv4 unicast
  redistribute connected route-map CONNECTED-TO-BGP

route-map CONNECTED-TO-BGP permit 200
 match ip address prefix-list CONNECTED-SINKHOLE

ip prefix-list CONNECTED-SINKHOLE permit 192.0.2.42/32
```

Content

1 Introduction

2 The implementation

3 The benefits

4 Common PHP obfuscation methods

Why are we doing this?

- General principle of reporting intrusions to customers!
- Upselling security consulting.

IP sinkholing also benefits transit customers.

Early detection by sinkholing greatly decreased the related security incidents!

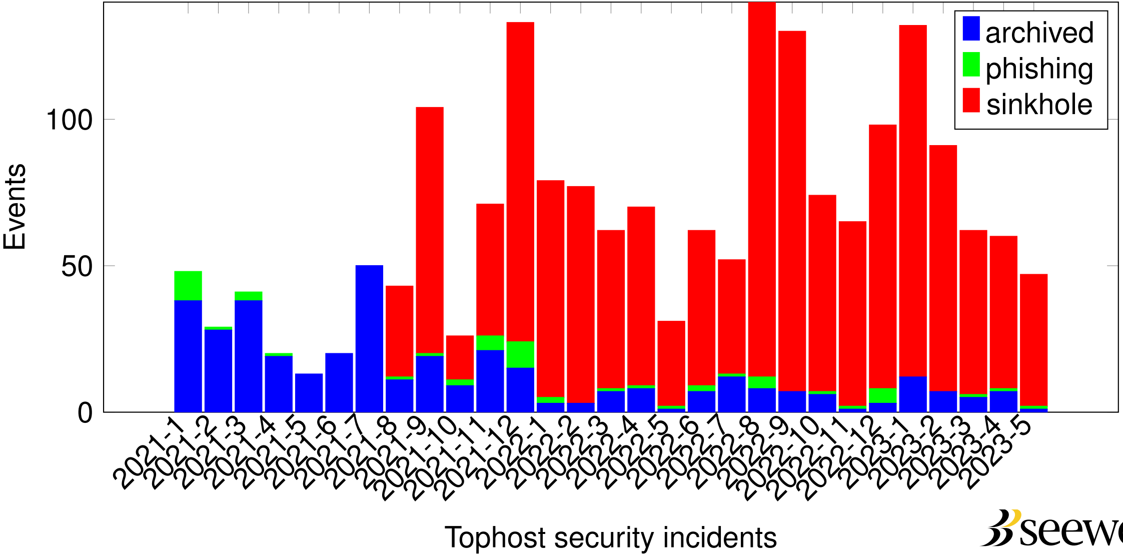
Some statistics

The current database

- 15 "campaigns"
- 1650 sinkholed IP addresses
- 80 sinkholed domains (hosted on Cloudflare)

(One year ago: 11 campaigns, 1010 IPs, 44 domains.)

Some statistics



Content

1. Introduction
2. The implementation
3. The benefits
4. Common PHP obfuscation methods

Obfuscation: eval and base64_decode

```php
<?php
eval(base64_decode('ZWNobyAiaGVsbG9cbiIK'));

eval(base64_decode(str_rot13('MJAbolNvnTIfoT9povVX')));

eval(gzuncompress(base64_decode(str_rot13(('...')))));

eval(gzuncompress(base64_decode(some_decrypter(('...')))));
```

Deobfuscation:

```
s/eval/echo/
```

Obfuscation: variables as function names

```
<?php
$f='str_rot13';
eval($f('rpub "uryyb jbeyq!\a";'));
```

Deobfuscation:

```
echo
```

Obfuscation: (hidden) create_function

```php
<?php
$f = 'create_function';
$a = '$foo';
$c = 'echo str_rot13($foo);';
$ff = $f($a, $c);
$ff("uryyb jbeyq!\n");
```

Deobfuscation:

echo

Obfuscation: printf escaping

```
<?php                                                                          ↪
${"\x47\x4c\x4f\x42AL\x53"}["x\x64\x72\x75\x74\x6a\x6ff\x72\x68"]="rg\x78";   ↪
${"\x47L\x4fB\x41L\x53"}["a\x65\x67o\x76\x68\x6a\x79i"]="\x6e\x65w";           ↪
${"GLO\x42ALS"}["\x65\x72b\x6f\x62\x6a\x68\x75k\x69\x72"]="\x6fl\x64\x5fb";   ↪
${"\x47\x4c\x4fB\x41L\x53"}["\x6bpeuu\x72\x75\x6a\x66\x67"]                    ↪
="\x6f\x6c\x64\x5fa";                                                          ↪
${"\x47\x4cOB\x41\x4c\x53"}["\x67mb\x73iv\x6b\x61\x79\x6bjx"]                  ↪
="m\x79\x73e\x6c\x66";                                                         ↪
${"\x47\x4cOB\x41LS"}["\x72\x66j\x74\x75\x63\x69\x77\x68\x6eo"]                ↪
="\x6ee\x77\x5f\x62";
```

Deobfuscation:

Filter with `printf`.

Obfuscation: junk comments

```
<?php
eval/*a*/(/*dru*//*dnh*/(/*y*/rawurldecode/*u*/(/*up9s*/$_pbtfh1/*z*/)/*9*/     ↪
^ substr/*a3vs2*/(/*1*/str_repeat/*uq7r*/(/*4k*/$_elijqf,                       ↪
/*9y*/(/*g3*/strlen/*muyx*/(/*6yk8*/$_pbtfh1/*th8vc*/)/*vdgp*//strlen/*vx0*/    ↪
(/*21g*/$_elijqf/*8xa9*/)/*c5*//*er*/)/*cf3dz*/ + 1/*en*/)/*8*/, 0,             ↪
strlen/*oit01*/(/*e*/$_pbtfh1/*pr8*/)/*dcf51*//*n*/)/*7rtw*//*6*/)/*y*//*w*/    ↪
)/*c0su*/;
```

Deobfuscation:

```
php -w
```

Obfuscation: arrays

```
<?php
$OOOOOO="%71%77%65%72%74%79%75%69%6f%70...%21%2a%7c%2b%2c";
$O=urldecode($OOOOOO);
${$O{18}.$O{7}.$O{24}.$O{2}.$O{50}.$O{8}}="2336";

if (!preg_match($O{63}.$O{79}.$O{15}.$O{4}.$O{4}.$O{9}.                ↪
$O{83}.$O{62}.$O{83}.$O{63}.$O{83}.$O{63}.$O{63}.$O{11}. $O{7}, $OOOoooOO))   ↪
{

function oooOOoOOoOOoO($OOOoooOO, $OOoooO = 1, $OOOOoooO = NULL, $OOOOoooOOO =   ↪
array()) {
```

Deobfuscation:

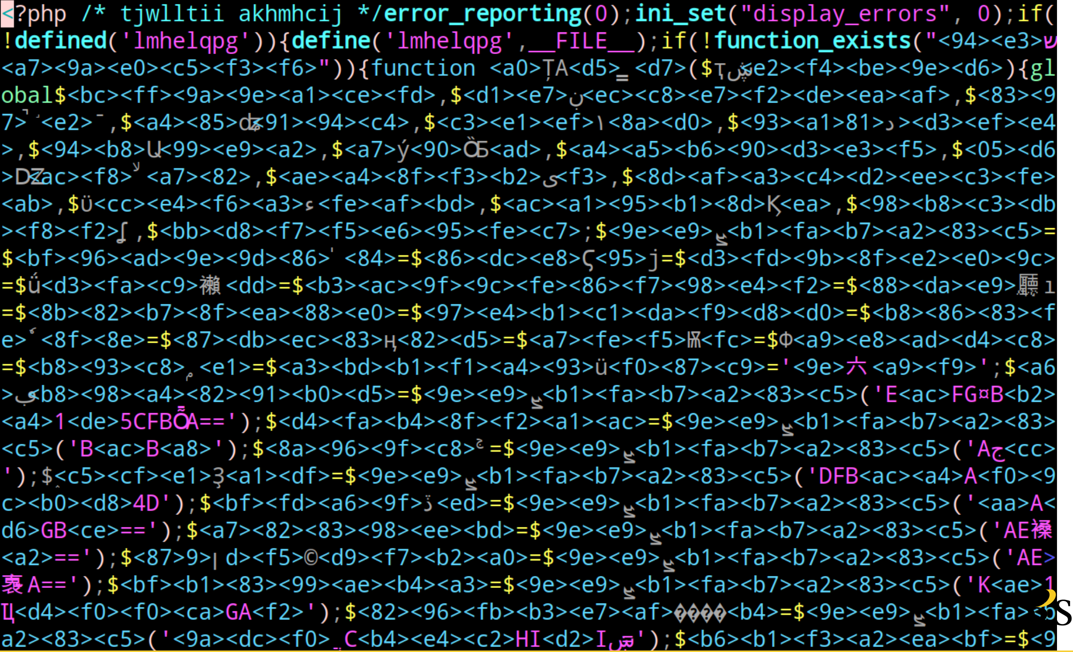
```
echo, var_dump
```

Obfuscation: goto

```
<?php
goto pn_FT; AHvKp: goto a5HmB; Wf8RV: function kk2qW($nujoa) { goto RS2q9;      ↪
PAhG4: goto n2edC; goto X8pnw; ctNPy: return $G4JcL; goto ov1Qt; B875x:         ↪
$cW69A = 0; goto HWHHd; d4dHM: ogkPy: goto fil8P; fil8P: $cW69A += 2; goto      ↪
PAhG4; O1mPS: $G4JcL .= pack("C", hexdec(substr($nujoa, $cW69A, 2))); goto      ↪
d4dHM; HWHHd: n2edC: goto psCVH; RS2q9: $PE3gP = strlen(trim($nujoa)); goto     ↪
sW_Kh; X8pnw: iChD_: goto ctNPy; sW_Kh: $G4JcL = ''; goto B875x; psCVH: if      ↪
(!($cW69A < $PE3gP)) { goto iChD_; } goto O1mPS; ov1Qt: } goto AHvKp; pn_FT:    ↪
error_reporting(0); goto Wf8RV; a5HmB: echo(kK2qw("65766..."));
```

Deobfuscation:

Reorder...



```
<?php /* tjwlltii akhmhcij */error_reporting(0);ini_set("display_errors", 0);if(
!defined('lmhelqpg')){define('lmhelqpg',__FILE__);if(!function_exists("<94><e3>ש
<a7><9a><e0><c5><f3><f6>")){function <a0>ŢA<d5>_ <d7>($ҭڜ<e2><f4><be><9e><d6>){gl
obal$<bc><ff><9a><9e><a1><ce><fd>,$<d1><e7>ڹ<ec><c8><e7><f2><de><ea><af>,$<83><9
7>ˀ <e2>ˉ,$<a4><85>ʣ91><94><c4>,$<c3><e1><ef>ا<8a><d0>,$<93><a1>81>د><d3><ef><e4
>,$<94><b8>Ա<99><e9><a2>,$<a7>ý<90>ŎБ<ad>,$<a4><a5><b6><90><d3><e3><f5>,$<05><d6
>DŽac><f8>ٰ <a7><82>,$<ae><a4><8f><f3><b2>ی<f3>,$<8d><af><a3><c4><d2><ee><c3><fe>
<ab>,$ü<cc><e4><f6><a3>ء<fe><af><bd>,$<ac><a1><95><b1><8d>Ҟ<ea>,$<98><b8><c3><db
><f8><f2>ʆ,$<bb><d8><f7><f5><e6><95><fe><c7>;$<9e><e9>ۓ<b1><fa><b7><a2><83><c5>=
$<bf><96><ad><9e><9d><86>ˈ <84>=$<86><dc><e8>Ç<95>j=$<d3><fd><9b><8f><e2><e0><9c>
=$ǘ<d3><fa><c9>瀨<dd>=$<b3><ac><9f><9c><fe><86><f7><98><e4><f2>=$<88><da><e9>歷ı
=$<8b><82><b7><8f><ea><88><e0>=$<97><e4><b1><c1><da><f9><d8><d0>=$<b8><86><83><f
e>ˁ <8f><8e>=$<87><db><ec><83>ң<82><d5>=$<a7><fe><f5>琢<fc>=$Ф<a9><e8><ad><d4><c8>
=$<b8><93><c8>ِ <e1>=$<a3><bd><b1><f1><a4><93>ü<f0><87><c9>='<9e>六 <a9><f9>';$<a6
>ڀ<b8><98><a4><82><91><b0><d5>=$<9e><e9>ۓ<b1><fa><b7><a2><83><c5>('E<ac>FG¤B<b2>
<a4>1<de>5CFBŐ̃A==');$<d4><fa><b4><8f><f2><a1><ac>=$<9e><e9>ۓ<b1><fa><b7><a2><83>
<c5>('B<ac>B<a8>');$<8a><96><9f><c8>ۡ =$<9e><e9>ۓ<b1><fa><b7><a2><83><c5>('Aج<cc>
');$<c5><cf><e1>ȝ<a1><df>=$<9e><e9>ۓ<b1><fa><b7><a2><83><c5>('DFB<ac><a4>A<f0><9
c><b0><d8>4D');$<bf><fd><a6><9f>ڗ<ed>=$<9e><e9>ۓ<b1><fa><b7><a2><83><c5>('<aa>A<
d6>GB<ce>==');$<a7><82><83><98><ee><bd>=$<9e><e9>ۓ<b1><fa><b7><a2><83><c5>('AE褖
<a2>==');$<87>9>ן d><f5>©<d9><f7><b2><a0>=$<9e><e9>ۓ<b1><fa><b7><a2><83><c5>('AE>
裹A==');$<bf><b1><83><99><ae><b4><a3>=$<9e><e9>ۓ<b1><fa><b7><a2><83><c5>('K<ae>1
Џ<d4><f0><f0><ca>GA<f2>');$<82><96><fb><b3><e7><af>����<b4>=$<9e><e9>ۓ<b1><fa><b
a2><83><c5>('<9a><dc><f0>_C<b4><e4><c2>HI<d2>Iػ');$<b6><b1><f3><a2><ea><bf>=$<9
```

Any questions?

`https://www.linux.it/~md/text/sinkholing-aipsi.pdf`

(Google . . . Marco d'Itri . . . I feel lucky)