

红队利用中，主要有以下几个板块。

找到漏洞-->利用漏洞-->权限维持-->痕迹清除。

找到漏洞对应的技能是代码审计。

利用漏洞对应的技能是各种实战中利用技巧+绕 waf。

权限维持，抽象来看，就是系统自己启动我的恶意代码，实现上看，往往要和 edr 做对抗。

痕迹清除，这块我研究的不深，日常应付项目反溯源，就是基本的删日志，删 history 等。

本文从代码审计切入，以点概面的来谈这套体系，如果真正理解了，其实会发现其他的东西也是一样的。

代码审计，本质就是阅读理解。

阅读理解大家都做过，无论是语文还是英语还是代码，本质就是一回事。

最开始做阅读理解的时候，这里拿英文举例子可能更有感觉，如果从来没接触过英文，其实大家都看不太懂，再有天赋也看不懂，因为不知道是啥东西。

阅读理解，首先要做拆分，一个是阅读，一个是理解。

阅读是观察具体字符组合，理解是逻辑上达到自洽。

这里先从语言的最小粒度，单词着手开始描述：

笼统的说，单词，就是从特定字符对应到现实中的某样东西，如图所示：



英文单词我们往往是这么理解的，先把英语翻译成中文，然后再进行理解。

因为英语不是我们的第一语言，想办法找同义项替换是自然的第一反应，也是比较高效率的反应。

因为这样可以利用我们已知的东西（中文）来学习未知，相当于已经有基础了，不用再从零开始学习。

翻译成中文之后，我们知道好这个词，马上就理解了，因为我们已经把好这个词，和现实中的某种具体感受联系起来了，比如某些愉悦的感受，站在海边吹着海风，波浪层层递推，万里晴空一望无际，在这个时候，我们就会用，“好”，这个词来描述这种感受，也可以理解为对于现实世界具体事物的一种抽象。

那么单词的学习，本质上就是建立具体词和现实世界对应的事物的一种联系。

可以抽象为以下过程：

看到一个新词-->联系到现实世界某种具体的事物-->建立链接

这种链接其实其实有点像代码中的赋值。

例如，代码中是

`a = 1` --> 把 a 和到内存中的 1 做一个链接

英语中是

Good = Something is pleasing or valuable or useful -->映射到现实的具体事物

汉语中是

好 = 一些让我舒服的东西 --> 映射到现实的具体事物

这里链接的建立是在我们的大脑神经元中建立的，这种连接有强有弱，如果天天熟悉某样特定事物，这种连接就会变强，逐步就会形成长期记忆，然后就会熟悉这门语言。

然而单单会词其实还不够，就像我们背诵了所有的英文单词，但是如果我们要写出一篇优美的英语作文，我们却无从下手一样，因为这里还涉及到单词的组合，需要符合既定的规则，我们称之为语法。

那么有了单词，有了语法，单词+语法，就可以形成单个句子。

然后再把单个句子的逻辑组合起来，在口头表达上，就形成了口语，在作文上，就形成了书面语。

最终，单词+语法+正确的逻辑，就得到了最终的成品。

没有单词，基本的单点事物映射都描述不出来。

没有语法，词语组合一片混乱，单体的简单意思都表达不出来。

没有逻辑，句子组合一片混乱，整体的稍微复杂点的意思就无法表达了。

以上就是任何一门语言的基本性质。

那么通过上面语言学习的原理的描述，这里我们可以推断，要学会一门语言，其实最好的方法就是多用，因为在运用的过程中，会逼迫大脑不断的熟悉特定字符，然后在我们表达具体意思的时候，大脑又会先映射现实事物的具体逻辑，然后再用我们熟悉的字符表达出来，如此往复，不断的训练我们的单词，语法，逻辑，最后我们就能彻底的学会了这门语言。

上面的讲法是通用性的，那么针对代码，方法是一回事，但是具体的技术细节需要调整。

下面以 java 反射举例，这里我想在 java 运行时候调用某个类的方法，方法如下：

```
public void test(String[] arg){
    for (String string : arg) {
        System.out.println(string);
    }
}
```

简单写了个 demo，就是遍历输入的数组，然后打印。

然后我会传入参数：

```
String[] s = new String[]{"fucku", "fucku2"};
```

不出意外，运行代码的人就会被骂。

那么怎么实现呢？

按照我上面的逻辑，一方面是多熟悉单体的词义，例如 java 中的 invoke 反射调用，先知道是干嘛的（单词），上网查了一下，发现是调用方法的（基本链接建立）。

然后尝试写一句话（语法）：

```
Method method = clazz.getMethod("test", String[].class);//获取 test 方法
```

这里我把 test 方法写出来方便理解

```
public void test(String[] arg){  
    for (String string : arg) {  
        System.out.println(string);  
    }  
}
```

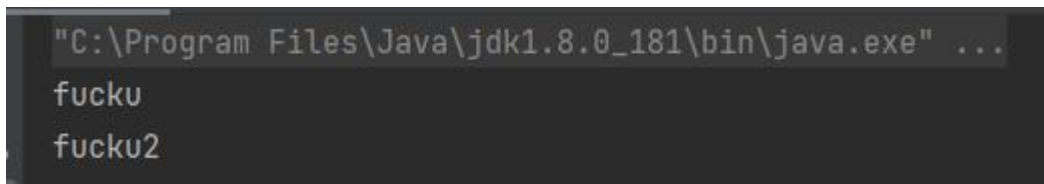
上面这句话的意思就是利用反射获取 test 方法

然后尝试组合逻辑（整体逻辑）：

```
Method method = clazz.getMethod("test", String[].class);//获取 test 方法
```

```
Method.invoke(m,(Object)s)//调用 test 方法
```

最后得到结果，建立代码和现实的映射（fucku fucku2）：



完整代码如下：

```
import java.lang.reflect.InvocationTargetException;  
import java.lang.reflect.Method;  
  
import org.testng.annotations.Test;  
  
public class Msym {  
  
    public void test(String[] arg){  
        for (String string : arg) {  
            System.out.println(string);  
        }  
    }  
  
    public static void main(String[] args) throws NoSuchMethodException, InvocationTargetException, InstantiationException, IllegalAccessException, ClassNotFoundException {  
        Class<Msym> clazz = (Class<Msym>) Class.forName("test.reflect.Msym");  
        //获取一个对象  
        Constructor con = clazz.getConstructor();  
        Msym m = (Msym) con.newInstance();  
        String[] s = new String[]{"fucku", "fucku2"};  
        //获取Method对象  
        Method method = clazz.getMethod("test", String[].class);  
        //调用invoke方法来调用  
        method.invoke(m, (Object) s);  
    }  
}
```

通过以上的描述，应该可以理解，语言学习本质上都是一回事，不管是你是英语也好，中文也好，是 c 语言也好，java 也好，都是大差不差。

其实代码审计挖漏洞，也是一回事。

正如前文提到，代码审计的本质，就是阅读理解。

阅读理解不单单是词，语法，逻辑的组合，还得会做题。

我们做英文的阅读理解，读懂了，是不是得做题，这样考试的时候，考官才能知道你究竟都没读懂。

代码审计也是一样，漏洞，就是阅读理解题。

要做出这些题，单单读懂是不行的，因为你理解的意思，可能和考官想考察你的意思有偏差。

因此这里还需要加上考试技巧，在英文的阅读理解中，也会有各种技巧辅助最终选出正确答案。

代码审计也是一样，理解+考试技巧，才能真正的挖出漏洞。

这里的技巧训练就是阅读历史漏洞，然后总结。

例如想挖 **weblogic**，那么 **weblogic** 的历史漏洞一定要全部看一遍。

每一个组件的代码都是具有个性的，这种个性和开发人员的开发风格和选用的开发套件相关，一个错误，他犯了一次，就可能会犯第二次，然后就会有规律，就会有套路可言。

基于以往的漏洞，往往就能发现新的漏洞。

这个和英语的阅读理解一样，训练英语阅读理解做题，往往我们要做很多题，然后总结题目类型，例如单词题，主旨题，段落理解题等等。

挖漏洞也是一样，单个组件的漏洞无非就是那么几种，例如 **weblogic** 一直在搞反序列化漏洞出来，那么总结以前的漏洞，然后学习源码，掌握规律，然后多熟悉熟悉，挖这种 **web** 组件的 **0day** 并不是难事，只是圈内的人喜欢搞神秘主义，一点东西，喜欢渲染的离奇诡谲，揭开面纱之后，发现本质其实还是相对质朴的。

如果不相信笔者，也可以找其他熟悉的挖 **web** 组件 **0day** 的人问问，这东西真的有那么难吗？一定需要顶尖的天赋吗？

无非就是掌握语言学习的正确方法，多看，多练，多熟悉，仅此而已。

而且挖洞这东西和天赋没啥关系，这又不是体育。

要说打篮球，假如对手两米，**200kg**，而我一米六，**100kg**，这怎么打？那我肯定炸了，别人直接压着我暴扣，我一点脾气没有。

但是人类的智力差别其实根据正态分布来看，并没有体育中体型差别那么离奇，而且又是学语言，可以出去看看有多少中国人学不会中文的，除非人体硬件故障，那确实不行，否则基本都是能学会，能和别人交流。

大部分人搞不出来，无非就是方法不对，或者中途放弃了。

坚持用正确的方法，做正确的事情，在这种技术的追逐上，一般都是能够达到自己想要的目的的。

done