



渗透测试及漏洞挖掘



第一部分 漏洞类型、特点及攻击实例

第二部分 渗透测试流程



第一部分 漏洞类型、特点及攻击实例



◀ 什么是渗透测试

渗透测试是受信任的第三方进行的一种**评估网络安全**的活动，它通过对企业网络进行**各种手段的攻击**来找出系统存在的漏洞，从而给出网络系统存在安全风险的一种实践活动。通过**模拟现实的网络攻击**，渗透测试证实恶意攻击者有可能获取或破坏企业的数据资产。



◀ 什么是渗透测试

渗透测试为**模拟黑客攻击**测试，但两者也有区别，
渗透测试是“面”的测试，黑客攻击是“深度”测试。
前者讲究广泛度，后者讲究破坏性。

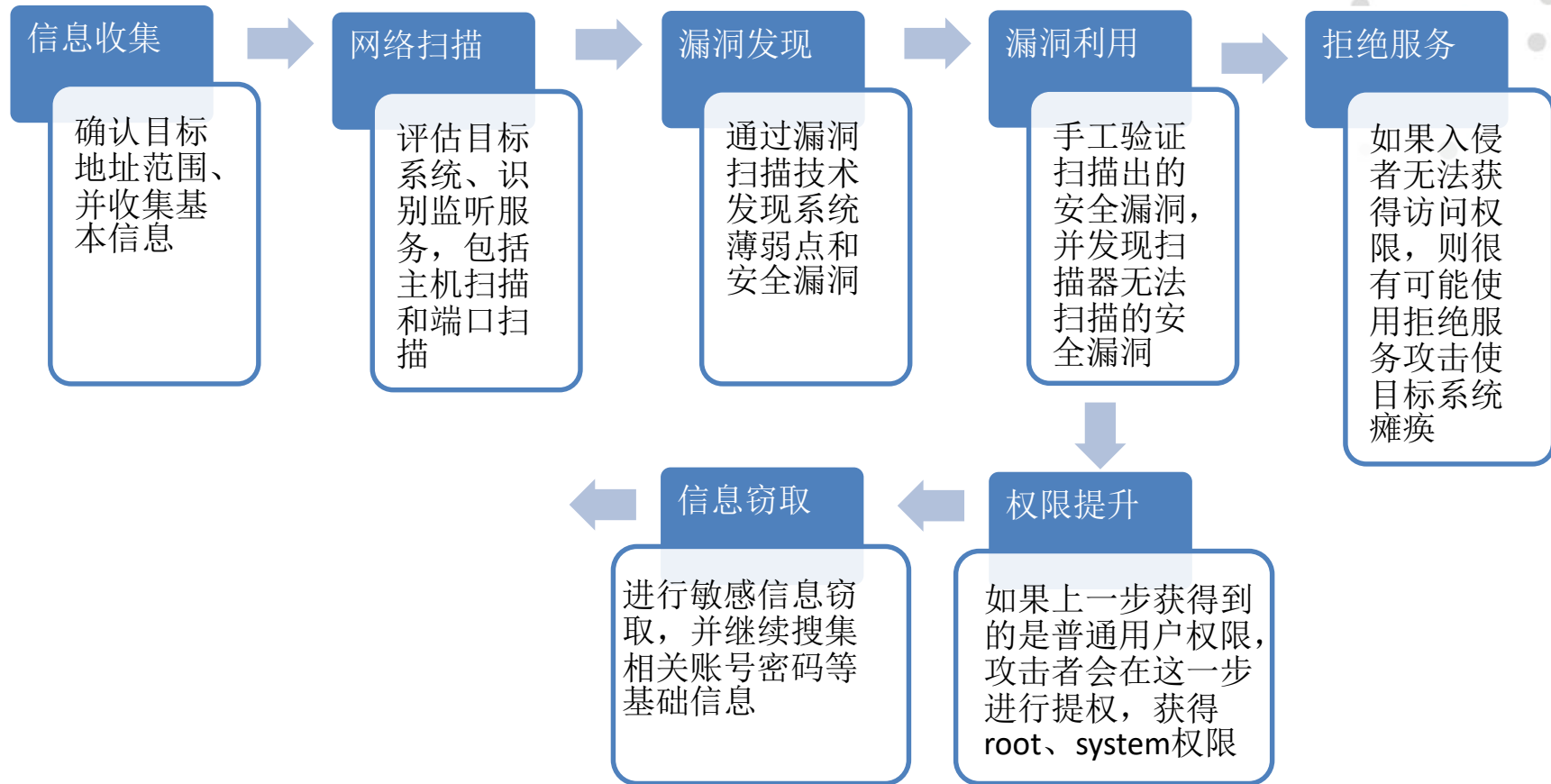


◀◀ 什么是渗透测试

- 1、黑盒测试(Black-box): 渗透测试人员不具备公司网络的任何信息。
- 2、白盒测试(White-box): 渗透测试人员已经具备内部网络完整信息。
- 3、灰盒测试(Gray-box): 测试人员模拟内部雇员，有一个内部网络的账户，并拥有了访问网络的标准方法。



渗透测试流程





OWASP Web应用漏洞Top10



OWASP

The Open Web Application Security Project

A1: Injection

A2: Cross Site Scripting (XSS)

A3: Broken Authentication and Session Management

A4: Insecure Direct Object References

A5: Cross Site Request Forgery (CSRF)

A6: Security Misconfiguration

A7: Failure to Restrict URL Access

A8: Unvalidated Redirects and Forwards

A9: Insecure Cryptographic Storage

A10: Insufficient Transport Layer Protection

A1: 注入

A2: 跨站脚本 (XSS)

A3: 失效的认证和会话管理

A4: 不安全的对象直接引用

A5: 伪造跨站请求 (CSRF)

A6: 安全配置错误

A7: URL访问控制不当

A8: 未验证的重定向和传递

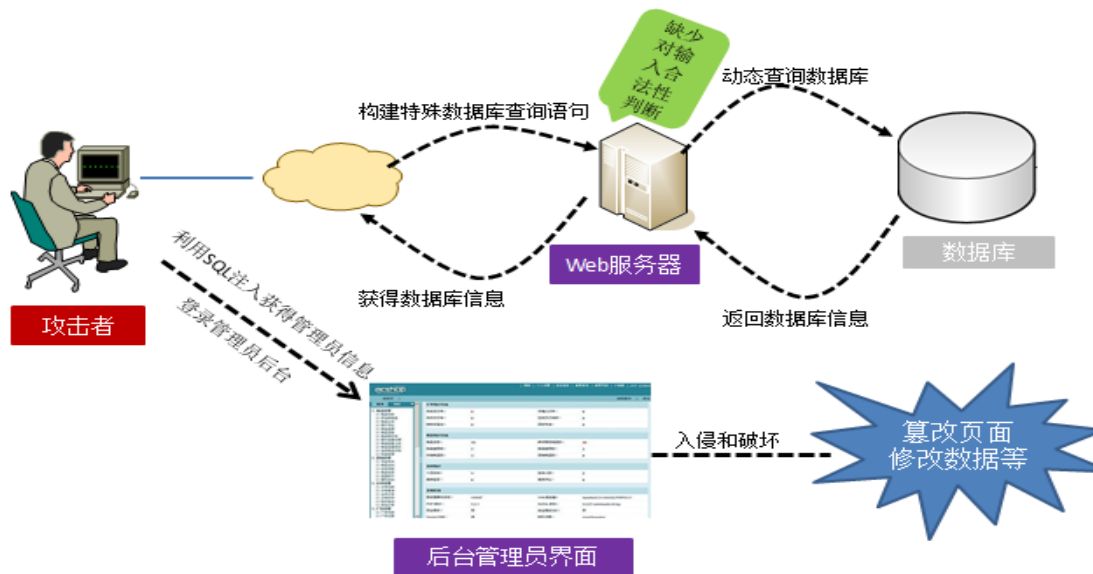
A9: 不安全的加密存储

A10: 不足的传输层保护

OWASP Web应用漏洞Top10-Injection

关于SQL注入

由于程序中对用户输入检查不严格，用户可以提交一段数据库查询代码，根据程序返回的结果，获得某些他想得知的数据，这就是所谓的SQL注入，攻击过程如下：



SQL注入

攻击特点

- 攻击的多样性，有经验的攻击者会**手动调整攻击参数**，致使攻击数据的变种是不可枚举的，这导致传统的**特征匹配检测方法**仅能识别相当少的攻击，难以防范
- 攻击广泛性，由于其利用的是**SQL语法**，并且目前互联网上流行众多的SQL注入攻击工具，攻击者借助这些工具可很快对目标WEB系统实施攻击和破坏
- 攻击的危害大，非法**查取**数据库中的敏感数据，对数据做任意的**修改**，甚至能够获得数据库所在的服务器的系统权限，执行系统命令，破坏力达到及至

影响范围

- 数据库：MS SQL、Oracle、Mysql、DB2、Informix等所有基于SQL语言标准的数据库软件；
- 应用程序：ASP、PHP、JSP、CGI、CFM等所有应用程序；

SQL注入-实例1

系统管理员登陆场景:

Username: admin

Password: admin@123

String query = "SELECT * FROM users WHERE userName = '"

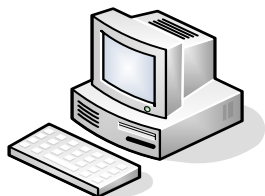
+ 用户名变量 + "' AND password = '"

+ 密码变量 + "'";

ResultSet rs = stmt.execute(query);

字符串
拼接

输入用户名:
admin
输入密码:
admin@123



系统管理员

用户登录

文章管理系统

用户名:

密码:

登录



/login.jsp

登陆
成功!

Select * from users where userName = 'admin' and password = 'admin@123' ;

SQL注入-实例1

系统管理员登陆场景:

Username: admin

Password: admin@123

```
String query = "SELECT * FROM users WHERE userName = '"
```

```
+ 用户名变量 + "' AND password = '"
```

```
+ 密码变量 + "'";
```

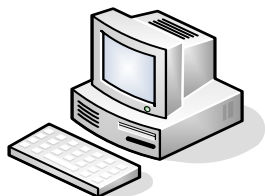
```
ResultSet rs = stmt.execute(query);
```

输入用户名:

' or '1' = '1' ;

--

输入密码: 123



系统管理员

用户登录

文章管理系统

用户名:

密码:

登录

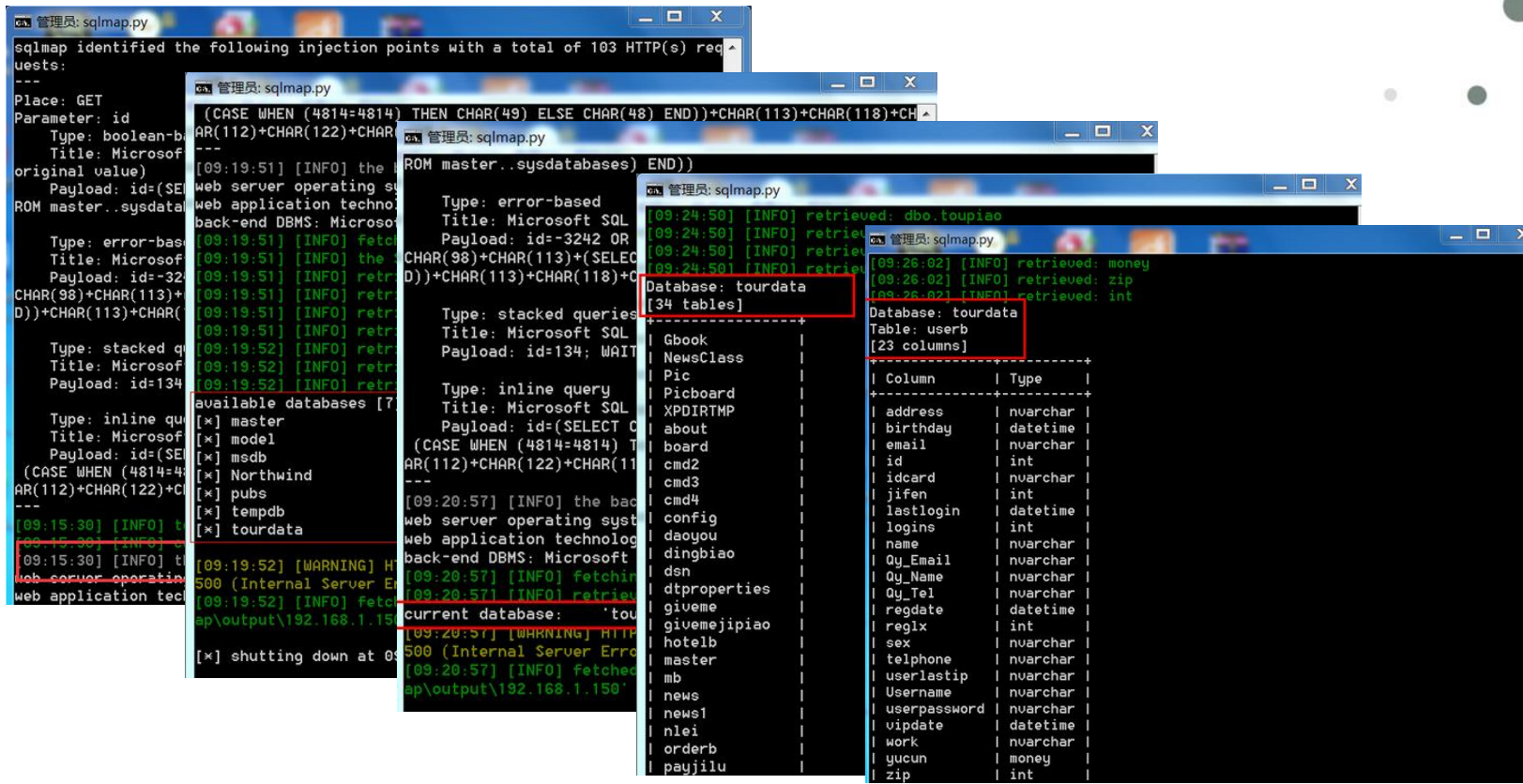


/login.jsp

登陆成功!

```
Select * from users where userName = ' ' or '1' = '1' ;--  
' and password = '123' ;
```

http://192.168.1.150/products.asp?id=134





SQL注入安全防范措施



SQL注入安全防范措施

- 使用参数化查询，将查询逻辑与查询数据分离
- 严格限定参数类型和格式，明确参数检验的边界，必须在服务端正式处理之前对提交的数据的合法性进行检查
- 替换或删除敏感的字符或字符串
- 对所有输入内容进行转义处理
- 验证输入，即白/黑名单验证
- 最小化SQL权限
- 一致的错误消息机制，屏蔽出错信息

OWASP Web应用漏洞Top10-XSS

□ 跨站攻击漏洞 (XSS) :

跨站攻击是指WEB应用程序没有对用户的输入和输出进行严格的过滤和转换而致恶意攻击者往Web页面里插入恶意html代码，如JavaScript, VBScript, ActiveX等恶意脚本，其中最主要的是JavaScript，当用户浏览该页之时，嵌入其中Web里面的html代码会被执行，从而达到到达盗取用户身份、拒绝服务攻击、篡改网页、模拟用户身份发起请求或执行命令等。

□ 跨站攻击分类:

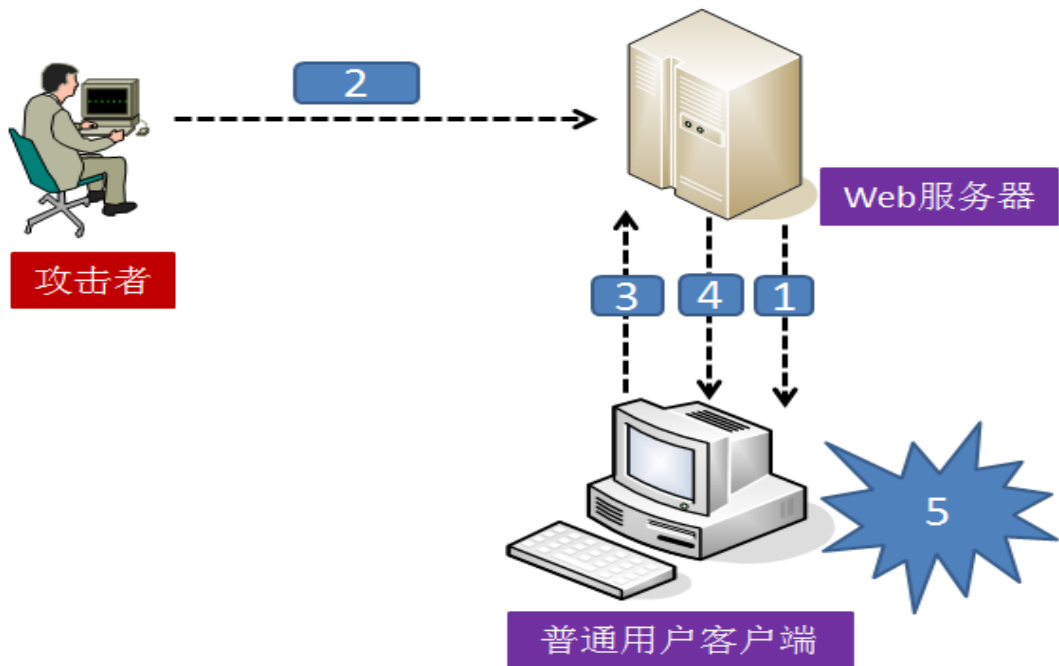
➤持久型XSS (Persistent)，即存储型跨站，它是指通过提交恶意数据到服务器，通过Web应用程序输出恶意数据输出到页面，存储在数据库中的，持久型XSS多出现在微博、留言板、BBS、社区等（比如BBS的某篇帖子中可能就含有恶意代码），存储型跨站危害比较大。

➤非持久型XSS (Non-persistent)，即反射型,它是指那些浏览器每次都要在参数中提交恶意代码才能触发的跨站脚本漏洞。

◀ 存储型跨站XSS

存储型跨站过程如下：

- 1、用户正常浏览信息
- 2、通过发帖向服务器发送存在恶意代码的帖子
- 3、用户查看发帖网页，查看帖子内容
- 4、服务器将恶意的代码发送给用户
- 5、客户端浏览器执行恶意代码



存储型跨站XSS-实例

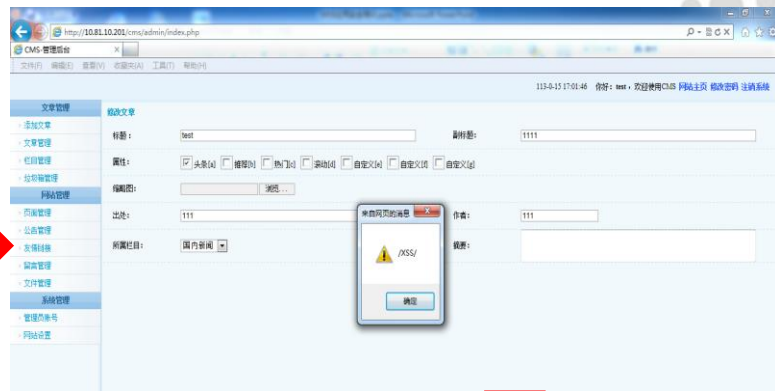
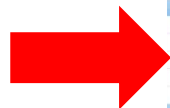
113-0-15 17:04:40 你好: test, 欢迎使用CMS 网站首页 修改密码 注销系统

副标题:

自定义 ☐ 自定义[g]

作者:

摘要:



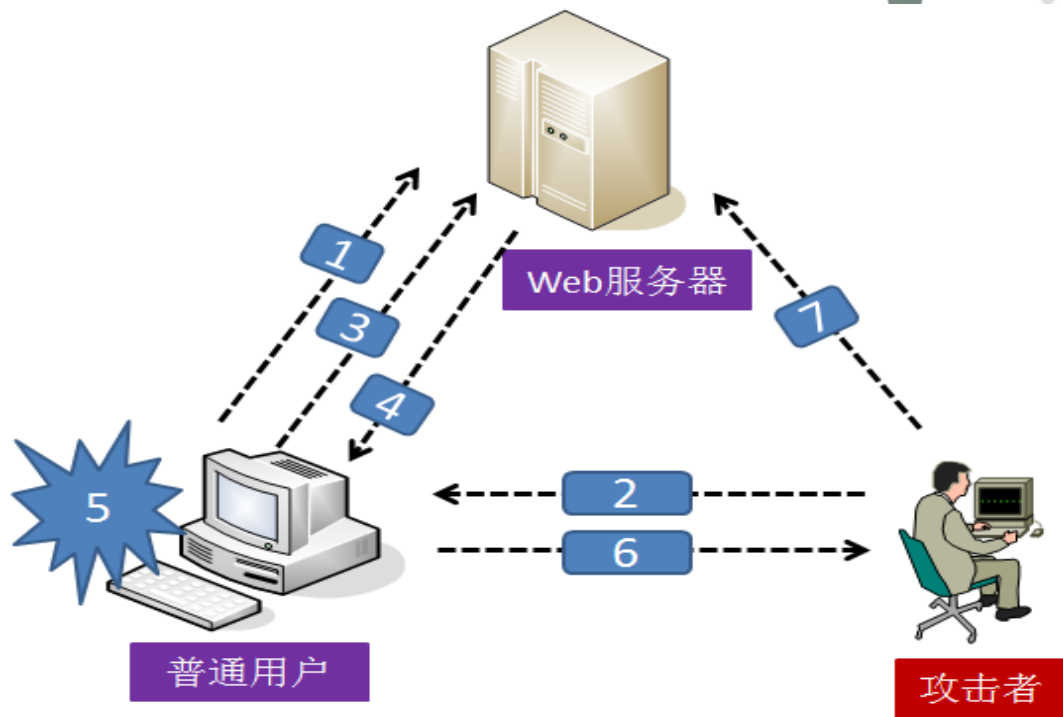
```
<td class="form_list">摘要: </td>
<td class="form_list"><textarea name="resume" class="form" style="width: 90%; height: 50px; overflow: auto;"></textarea></td>
</tr>
<tr>
<td height="40" colspan="4" align="center" class="form_list">
<input type="hidden" id="content" name="content" value="&lt;p&gt;1111&lt;/p&gt;" style="display:none" /><input type="hidden" id="content__Config"
value="" style="display:none" /><iframe id="content__Frame" src=".../include/fckeditor/editor/fckeditor.html?InstanceName=content&Toolbar=MyToolbar"
width="100%" height="350" frameborder="0" scrolling="no"></iframe>
</td>
```

摘要: </textarea> </td> <script>alert(/XSS/) </script>

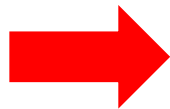
反射性跨站

反射性跨站过程如下：

- 1、用户正常登陆
- 2、攻击者把恶意的URL提交给客户
- 3、用户查请求恶意攻击者的URL
- 4、服务器对攻击者的JS做出回应
- 5、攻击者的JS在客户浏览器执行
- 6、用户的浏览器向攻击者发送会话令牌
- 7、攻击者劫持用户会话



反射型跨站XSS-实例



id=36参数后面添加: `")<script>alert(/XSS/)</script>`

由于URL里面的参数id在传递的过程中没有进行合法性检测, 从而产生了反射性的跨站漏洞。

跨站攻击漏洞的危害

跨站攻击漏洞的危害

- 钓鱼欺骗
- 网站挂马
- 身份盗用
- 盗取网站用户信息
- 垃圾信息发送
- 劫持用户Web行为
- XSS蠕虫：XSS 蠕虫可以用来打广告、刷流量、挂马、恶作剧、破坏网上数据、实施DoS攻击等
- 与其它漏洞结合，修改系统设置，查看系统文件，执行系统命令等



◀◀ XSS防范措施

□ 程序开发者：

- 对用户提交内容进行合法性校验
- 对用户提交内容进行转义处理
- 对用户输入的长度进行限制

□ 普通用户：

- 不要轻易访问别人给你的长链接，它可能包含了转码后的恶意HTML代码。
- 禁止浏览器运行JavaScript和ActiveX代码

需作转义的字 符	字符实体编 码
&	&
<	<
>	>
"	"
'	'
/	/

◀ OWASP Web应用漏洞Top10-BASM

关于身份认证

身份认证最常见的是通过用户名和密码，在安全性要求更高的情况下，有防止密码暴力破解的验证码，基于客户端的证书，物理口令卡等等

关于会话管理

会话管理就是通过cookie信息来识别已建立的连接，而不需要每次都要登陆。

关于失效的身份认证和会话管理

- 与身份认证和会话管理相关的应用程序功能往往得不到正确的实现，这就导致了攻击者破坏密码、密匙、会话令牌或攻击其他的漏洞去**冒充其他用户的身份**
- 用户凭证和Session ID是Web应用中最敏感的部分，也是攻击者最想获取的信息，攻击者会采用网络嗅探、暴力破解、社会工程等手段尝试获取这些信息。

失效的身份认证和会话管理-实例

□ 实例一：

某航空票务网站将用户Session ID包含在URL中：

`http://example.com/sale/saleitems;sessionid=2P0OC2JDPXM0OQSNDLPS
KHCJUN2JV?dest=Hawaii`

一位用户为了让她朋友看到这个促销航班的内容，将上述链接发送给朋友，导致他人可以看到她的会话内容。

□ 实例二：

一位用户在公用电脑上没有登出他访问的网站，导致下一位使用者可以看到他的网站会话内容。

□ 实例三：

登录页面没有进行加密，攻击者通过截取网络包，轻易发现用户登录信息。

□ 实例四：

存储在数据库中的用户密码没有被加密。

失效的身份认证和会话管理-防范措施

身份认证和会话管理的防范措施

- 对用户的登陆信息和cookie进行加密
- 用户密码强度（8字符以上；极其重要：使用多种验证方式）
- 不使用简单或可预期的密码恢复问题
- 登录出错时不给过多提示
- 对多次登录失败的帐号进行短时锁定
- 验证成功后更换Session ID
- 使用128位以上有足够随机性的Session ID
- 设置会话闲置超时（可选会话绝对超时）
- 保护Cookie（Secure flag/HTTPOnlyflag)
- 不在URL中显示Session ID

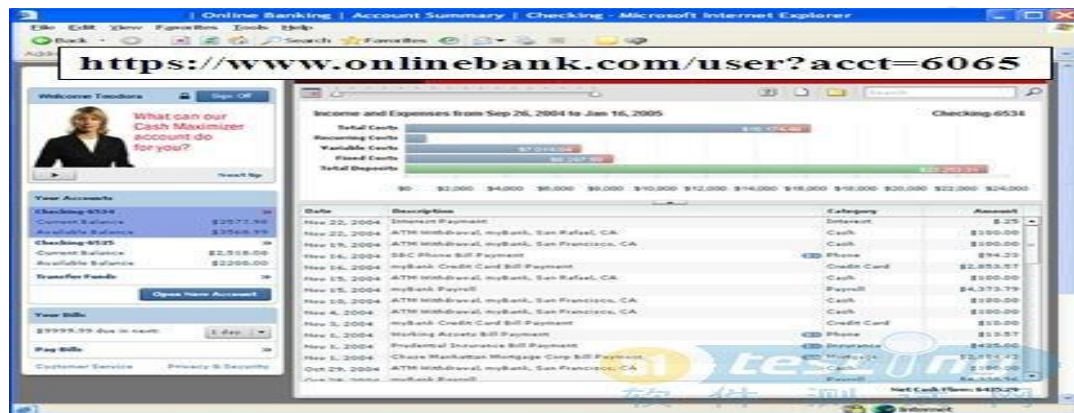
◀ OWASP Web应用漏洞Top10-IDOR

关于不安全的对象直接引用-越权

- 不安全的对象直接访问是指一个已经授权的用户，通过更改访问时的一个参数，从而访问到了原本其并没有得到授权的对象。
- 当开发人员暴露一个对内部实现对象的引用时，例如，一个文件、目录或者数据库密匙，就会产生一个不安全的直接对象引用。在没有访问控制检测或其他保护时，攻击者会操控这些引用去访问未授权数据
- 服务器上具体文件名、路径或数据库关键字等内部资源被暴露在URL或网页中，攻击者可以以此来尝试直接访问其他资源。

不安全的对象直接引用-实例

实例一：



实例二：

深圳航空 Shenzhen Airlines

新客立即体验 繁体 ENGLISH 深航手机平台 深航微博 深航女孩博客 深航订票服务热线: 95080 在线客服

首页 机票预订 集团客户 凤凰知音 深航电粉 深航旅游 深航酒店 关于我们 旅行服务信息

订单管理

我的订单

订单编号: T201203044 订单总金额: ¥ 370

航程	出发城市	目的城市	航班号	航班日期	起飞时间	到达时间	舱位	机型	座位	折扣
去程	南昌	上海浦东	25031	2012-03-07	10:15					¥ 35

旅客信息

航程	旅客类型	姓名	证件类型	证件编号	票价	燃油税	机场建设费	保险	票款总计	票号
去程	成人		身份证		¥ 250	¥ 10	¥ 50	¥ 0	¥ 370	479-0466-0000

订单总金额: ¥ 370

支付信息

支付方式	支付银行	支付流水号	支付时间	订单来源平台	订单状态
网上支付	支付宝	21048999	2012-03-04 19	深航官网	已出票

联系人信息

姓名	联系人手机	固定电话	电子邮箱
	15956		

温馨提示: 【查询2012年11月29日前原票】

www.wooyun.org

◀ 不安全的对象直接引用-实例

□ 实例三：

某网站的新闻检索功能可搜索指定日期的新闻，但其返回的URL中包含了指定日期新闻页面的文件名：

`http://example.com/online/getnews.asp?item=20March2003.html`

攻击者可以尝试不同的目录层次来获得系统文件win.ini:

`http://example.com/online/getnews.asp?item=../../winnt/win.ini`

□ 实例四：

2000年澳大利亚税务局网站曾经发生一位用户通过修改其URL中ABN ID号而直接访问到17000家公司税务信息的事件。

◀ 不安全的对象直接引用-防范措施



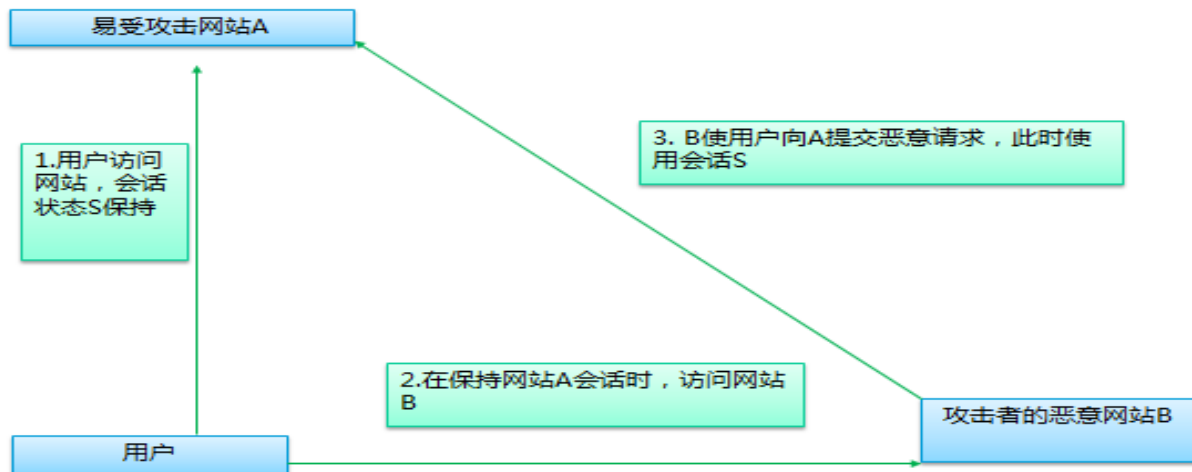
不安全的对象直接引用防范措施

- 避免在URL或网页中直接引用内部文件名或数据库关键字
- 使用非直接的对象引用，可使用自定义的映射名称来取代直接对象名
`http://example.com/online/getnews.asp?item=11`
- 锁定网站服务器上的所有目录和文件夹，设置访问权限
- 验证用户输入和URL请求，拒绝包含./或../的请求

OWASP Web应用漏洞Top10-CSRF

跨站请求伪造（CSRF），是一种对网站的恶意利用，一个跨站请求伪造攻击迫使登录用户的浏览器将伪造的HTTP请求，包括该用户的会话cookie和其他认证信息，发送到一个存在漏洞的web应用程序。这就允许了攻击者迫使用户浏览器向存在漏洞的应用程序发送请求，而这些请求会被应用程序认为是用户的合法请求。

与反射型XSS的主要区别是：反射型XSS的目的是在客户端执行脚本；CSRF的目的是在Web应用中执行操作。



跨站请求伪造CSRF-实例

CSRF构造语句如下:

```
<form name="form1" method="post"
action="http://10.81.10.201/cms/admin/user.
action.php">
  <input name="act" value="add">
    <input name="username"
value="hacker">
    <input name="password"
value="hacker">
    <input name="password2"
value="hacker">
    <input name="userid" value="0">
</form>
<script language="javascript">
document.form1.submit();
</script>
```

文章管理系统

首页 国内新闻 国际新闻 留言板

留言

标题: 文章写得真好!

姓名: 小王 验证码: 5923 5923

性别: ☒ 男 ☐ 女 电话: 15211122233

QQ: 123456789 Email: 123@126.com

地址: 北京市西单

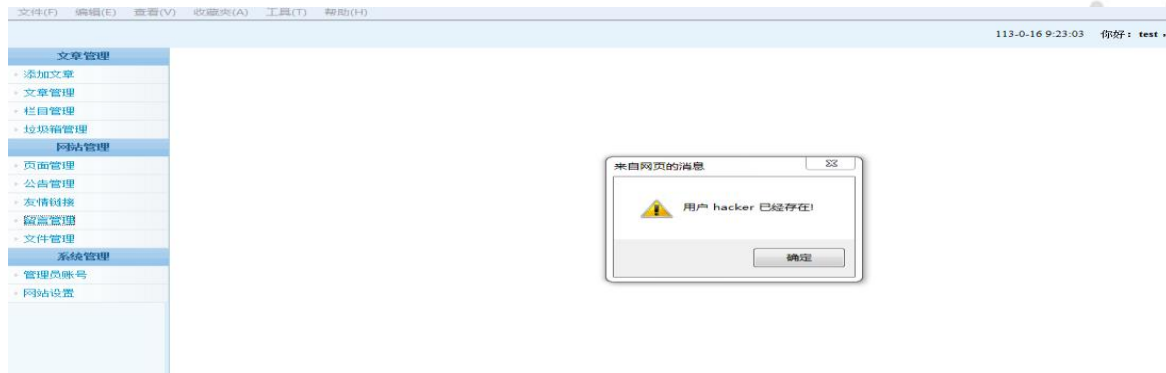
内容: `<form name="form1" method="post"
action="http://10.81.10.201/cms/admin/user.ac`

提交 重置

把构造好的CSRF构造语句
填写在留言板的内容里

跨站请求伪造CSRF-实例

当以管理员的身份登录系统并查看留言时，就会执行恶意攻击者的JS语句去创建用户名和密码都为hacker的用户。



当查看系统管理员时发现已经多出来一个名为hacker的用户，这样恶意攻击者就通过伪造请求的方式创建了hacker这个用户。



◀ 跨站请求伪造CSRF-防范措施



跨站请求伪造CSRF防范措施

- 避免在URL中明文显示特定操作的参数内容
- 使用同步令牌 (Synchronizer Token) ,检查客户端请求是否包含令牌及其有效性
- 检查RefererHeader, 拒绝来自非本网站的直接URL请求

◀ OWASP Web应用漏洞Top10-SM

关于安全配置错误

- 安全配置可能存在于Web应用的各个层次
- 需要对应用程序、框架、应用程序服务器、web服务器、数据库服务器和平台，定义和执行安全配置。
- 由于许多设置的默认值并不是安全的，因此，必须定义、实施和维护所有这些设置。这包含了对所有的软件保持及时地更新，包括所有应用程序的库文件
- 管理员在服务器安全配置上的疏忽，通常会导致攻击者非法获取信息、篡改内容，甚至控制整个系统

◀ 安全配置错误-实例

安全配置错误实例

- 服务器没有及时安装补丁
- 网站没有禁止目录浏览功能
- 网站允许匿名用户直接上传文件
- 服务器上文件夹没有设置足够权限要求，允许匿名用户写入文件
- 网站安装并运行并不需要的服务，比如FTP或SMTP
- 出错页面向用户提供太过具体的错误信息，比如call stack
- Web应用直接以SQL SA帐号进行连接，并且SA帐号使用默认密码
- SQL服务器没有限制系统存储过程的使用，比如xp_cmdshell

◀◀ 安全配置错误-防范措施



安全错误配置防范措施

- 可使用自动化的安全配置向导
- 必须覆盖整个平台和系统
- 对所有组件都必须保证安装了最新的补丁
- 完善分析变更带来的安全影响
- 对所有做过的安全配置进行记录
- 使用自动化扫描工具定期对系统进行验证

◀ OWASP Web应用漏洞Top10-FRUA

关于URL访问控制不当

URL访问控制不当，与认证相关的，正常情况是web应用程序在显示受保护的链接和按钮之前会检测URL访问权限,但是如果URL访问控制不当攻击者能够很容易的就伪造请求直接访问未被授权的页面或隐藏的页面。

URL访问控制不当-实例

- 攻击者发现他自己的访问地址为 /user/getAccounts;
- 修改访问地址为 /admin/getAccounts或 /manager/getAccounts;
- 这样攻击者就能够查看到更多的账户信息了



◀ URL访问控制不当-防范措施



URL访问控制不当防范措施

- 对于网站内的所有内容（不论公开的还是未公开的），都要进行访问控制检查
- 只允许用户访问特定的文件类型，比如.html、.asp、.php等，禁止对其他文件类型的访问

◀ OWASP Web应用漏洞Top10-URF

关于未验证的重定向和传递

- Web应用程序经常将用户重定向和转发到其他网页和网站，并且利用不可信的数据去判定目的页面。如果没有得到适当验证，攻击者可以重定向受害用户到钓鱼软件或恶意网站，或者使用转发去访问未授权的页面
- 攻击者可能利用未经验证的重定向目标来实现钓鱼欺骗，诱骗用户访问恶意站点。
- 攻击者可能利用未经验证的跳转目标来绕过网站的访问控制检查。

未验证的重定向和传递-实例

- 服务器端程序根据用户输入的URL决定跳转的目的地, JSP代码如下:

```
<form method= "get" action= "/user" >
```

```
Destination URL: <input type= "text" name= "target" >
```

```
</form>
```

- 如果攻击者直接将URL修改为:

```
http://www.host.com/redir.jsp?target=http://www.malicious.com/index.jsp
```

用户看到这个连接指向了一个可信的网站, 没有注意到重定向机制可能将用户转到恶意网站, 攻击者可能成功诱使用户点击该请求

- 而且攻击者还可能将上面的链接进行编码

```
http://www.host.com/redir.jsp?target=http://
```

```
%77%77%77%2E%6D%61%6C%69%63%69%6F%75%73%2E%63%6F%6D%  
2F%69%6E%64%65%78%2E%6A%73%70%0D%0A
```

这时不能直接看到重定向的目的URL, 从而是这个危险链接更具有隐蔽性

◀ 未验证的重定向和传递-防范措施



未验证的重定向和传递防范措施

- 尽可能的避免使用重定向和转发机制
- 如果使用了，那么在定义目标URL的时候不要包含用户参数
- 如果一定要包含用户的参数，那么对每个参数都必须进行验证以确保它的正确性和合法性；或是在服务器端提供映射机制，将用户的选择参数转变为真正的目标页面

◀◀ OWASP Web应用漏洞Top10-ICS

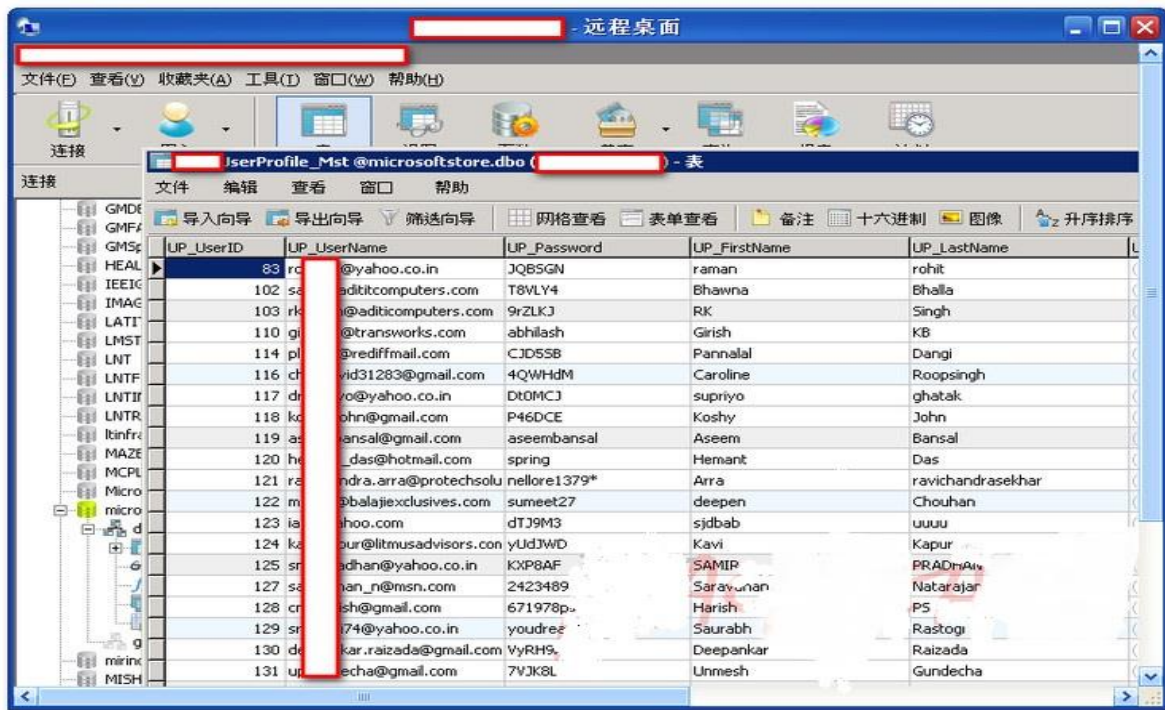
关于不安全的加密存储

- 许多web应用程序并没有使用恰当的加密措施或Hash算法保护敏感数据，比如信用卡、社会安全号码(SSN)、身份认证证书等等。攻击者可能利用这种弱保护数据实行身份盗窃、信用卡诈骗或其他犯罪
- 对重要信息不进行加密处理或加密强度不够，或者没有安全的存储加密信息，都会导致攻击者获得这些信息。

不安全的加密存储-实例

实例一：

对于重要信息，比如银行卡号、密码等，直接以明文写入数据库



UP_UserID	UP_UserName	UP_Password	UP_FirstName	UP_LastName
83	rc@yahoo.co.in	JQ85GN	raman	rohit
102	seaditcomputers.com	T8VL4	Bhawna	Bhalla
103	rk@aditcomputers.com	9rZLKJ	RK	Singh
110	gl@transworks.com	abhilash	Girish	KB
114	pl@rediffmail.com	CJD55B	Pannalal	Dangi
116	clvid31283@gmail.com	4QWHdM	Caroline	Roopsingh
117	dyo@yahoo.co.in	Dt0MCJ	supriyo	ghatak
118	kdohn@gmail.com	P46DCE	Koshy	John
119	aseembansal@gmail.com	aseembansal	Aseem	Bansal
120	hldas@hotmail.com	spring	Hemant	Das
121	raandra.arra@protechsolu	nellore1379*	Arra	ravichandrasekhar
122	mbalajixclusives.com	sumeet27	deepen	Chouhan
123	ia@yahoo.com	dtJ9M3	sjdbab	uuuu
124	kaour@litmusadvisors.com	yUdJWD	Kavi	Kapur
125	sradhan@yahoo.co.in	KXP8AF	SAMIR	PRADnAiv
127	san_n@msn.com	2423489	Sarayan	Natarajar
128	crsh@gmail.com	671978p	Harish	PS
129	sr74@yahoo.co.in	youdree	Saurabh	Rastogi
130	dkar.raizada@gmail.com	VyRH9	Deepankar	Raizada
131	unecha@gmail.com	7VJK8L	Unmesh	Gundecha

◀ 不安全的加密存储-实例

□ 实例二：

使用自己编写的加密算法进行简单加密

□ 实例三：

使用MD5, SHA-1等低强度的算法

□ 实例四：

将加密信息和密钥存放在一起

不安全的加密存储-防护措施



不安全的加密存储防范措施

- 对所有重要信息进行加密
- 仅使用足够强度的加密算法，比如AES、RSA
- 存储密码时，用SHA-256等健壮哈希算法进行处理
- 采用Salt技术来防范rainbow表攻击
- 产生的密钥不能与加密信息一起存放
- 严格控制对加密存储的访问

◀ OWASP Web应用漏洞Top10-ITLP

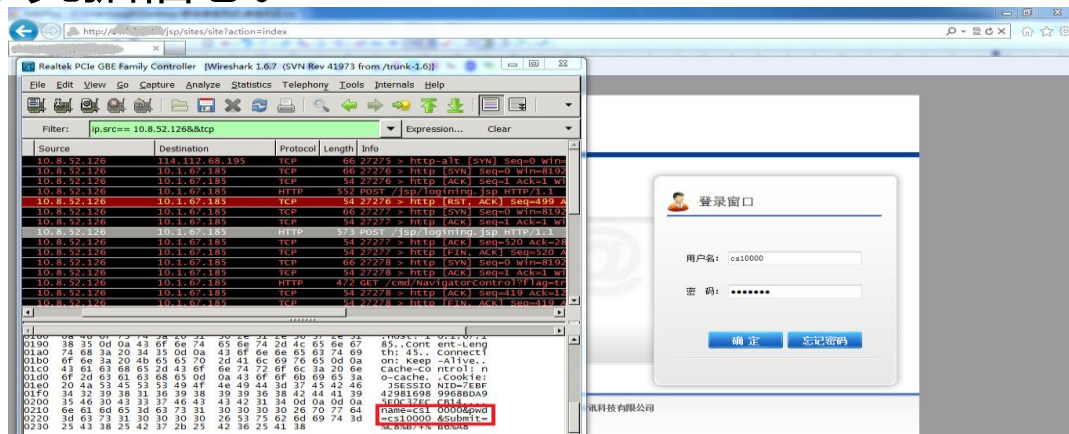
关于不足的传输层保护

- 应用程序时常没有进行身份认证，加密措施，甚至没有保护敏感网络数据的保密性和完整性。而当进行保护时，应用程序有时采用弱算法、使用过期或无效的证书，或不正确地使用这些技术
- 网络窃听（Sniffer、WireShark等工具）可以捕获网络中流过的敏感信息，如密码，Cookie字段等。高级窃听者还可以进行ARP欺骗，中间人攻击。

◀ 不足的传输层保护-实例

□ 实例一：

某网站的登录页面没有进行加密，攻击者在截取网络包后，可以获得用户的登录凭据信息。



□ 实例二：

某网站的HTTPS网页内容中还包含一些HTTP网页的引用。攻击者在截取网络包后，可以从HTTP请求中发现客户端的Session ID。

◀ 不足的传输层保护-防范措施



不足的传输层安全防范措施

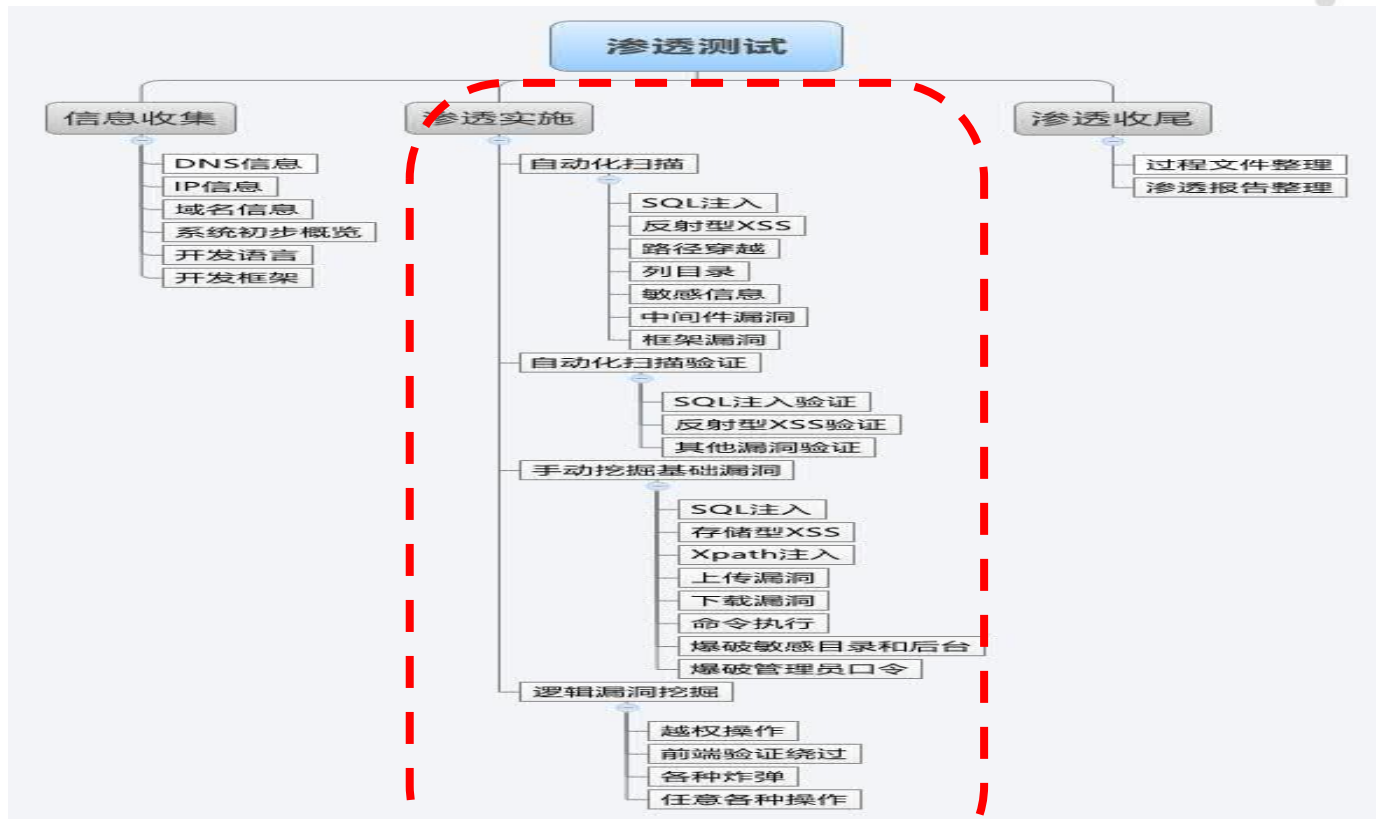
- 对所有验证页面都使用SSL或TLS加密
- 对所有敏感信息的传输都使用SSL/TLS加密
- 在网页中不要混杂HTTP和HTTPS内容
- 对Cookie使用Secure标签
- 只允许SSL 3.0或TLS 1.0以上版本协议
- 有需要的情况下，要求客户端证书

第二部分 渗透测试流程





标准渗透测试流程





信息收集

信息收集

DNS信息

信息收集

```
C:\Users>nslookup
默认服务器: Broadcom.Home
Address: 192.168.1.1
```

```
> set type=A
> testfire.net
服务器: Broadcom.Home
Address: 192.168.1.1
```

```
非权威应答:
名称: testfire.net
Address: 65.61.137.117
```

```
> set type=MX
> testfire.net
服务器: Broadcom.Home
Address: 192.168.1.1
```

```
testfire.net
primary name server = asia3.akam.net
responsible mail addr = hostmaster.akamai.
serial = 1366025603
refresh = 43200 (12 hours)
retry = 7200 (2 hours)
expire = 604800 (7 days)
default TTL = 86400 (1 day)
```

Domain Name: testfire.net

Registry Domain ID: 8363973_DOMAIN_NET-VRSN

Registrar WHOIS Server: whois.corporatedomains.com

Registrar URL: www.cscprotectsbrands.com

Updated Date: 2016-12-04T11:23:27Z

Creation Date: 1999-07-23T13:52:32Z

Registrar Registration Expiration Date: 2017-07-23T13:52:32Z

Registrar: CSC CORPORATE DOMAINS, INC.

Registrar IANA ID: 299

Registrar Abuse Contact Email: domainabuse@cscglobal.com

Registrar Abuse Contact Phone: +1.8887802723

Domain Status: clientTransferProhibited <http://www.icann.org/epp#clientTransferProhibited>

Registry Registrant ID:

Registrant Name:

Registrant Organization:

Registrant Street:

Registrant City:

Registrant State/Province:

Registrant Postal Code:

Registrant Country:

Registrant Phone:

Registrant Phone Ext:

Registrant Fax:

Registrant Fax Ext:

Registrant Email:

Registry Admin ID:

Admin Name: IBM DN

Admin Organization:

Admin Street: New

Admin City: Armonk

Admin State/Province:

Admin Postal Code:

Admin Country: US

Admin Phone: +1.91

Admin Phone Ext:

Admin Fax: +1.9147654370

科技业务管理系统

申报单位人员登录

账号登录

证书登录

组织机构代码
(或社会信用代码)

组织机构代码 (或社会信用代码)

不知道机构代码?
点击 查询

证件号码

证件号码

登陆密码

登陆密码

验证码

验证码

u05#A

登录

忘记密码

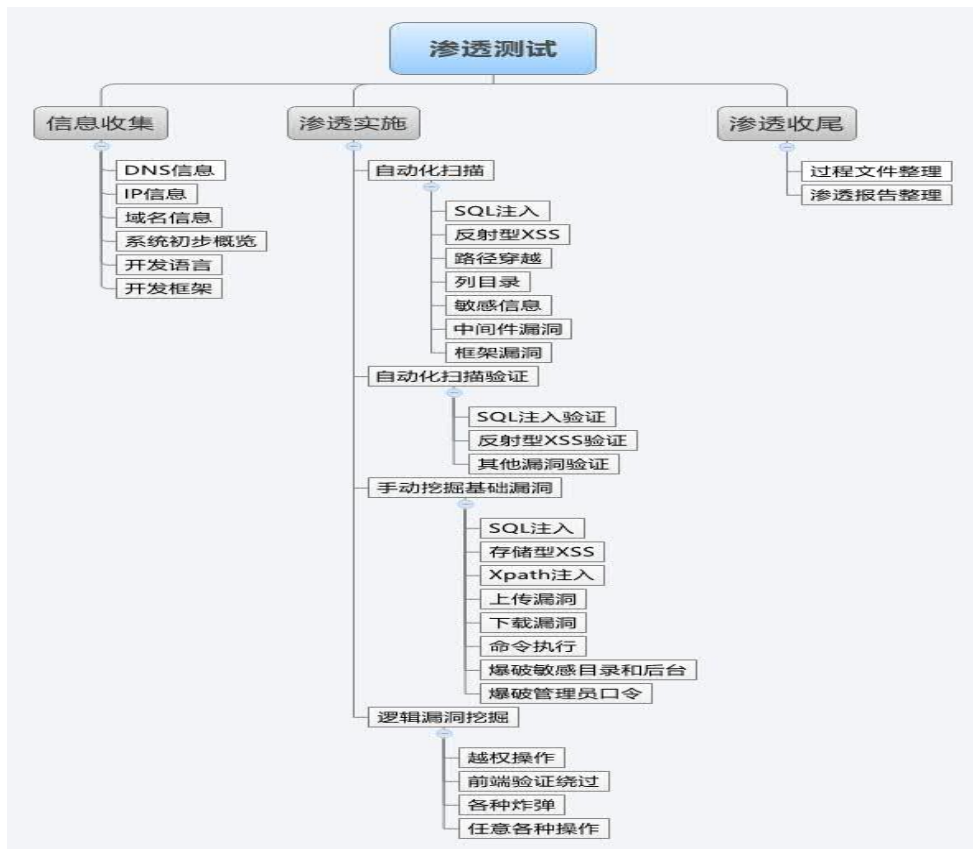
没有帐号, 点击单位注册

科研人员 (专家) 登录

业务处室人员登录



渗透实施

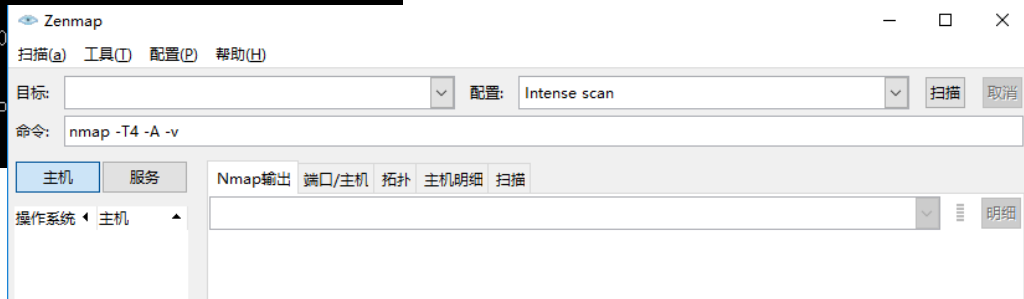


渗透实施——端口扫描

目的：发现已经打开的端口，对于映射全IP的服务，了解其系统情况和开启的服务。

实施方法：nmap

```
C:\Users>nmap
Nmap 7.50 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0.0/8
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
```





渗透实施——详细的WEB结构&信息收集

目的：了解系统结构，初步探明开发框架和开发语言。

实施方法：爬虫、手动浏览、**session&csrf&后缀**等标识查看

Scan Results	
editor	Not Found (404)
filemanager	Not Found (404)
images	Not Found (404)
js	Not Found (404)
plugins	Not Found (404)
+	Not Found (404)
fckeditor	Not Found (404)
editor	Not Found (404)
+	Not Found (404)
htaccess.txt	Ok (200)
images	Not Found (404)
main	Not Found (404)
+	Not Found (404)
scripts	Not Found (404)
+	Not Found (404)
calendar-setup.js	Ok (200)
calendar.js	Ok (200)
cn_utf8.js	Ok (200)
common.js	Ok (200)
date.js	Ok (200)
OpenModelessDialog.js	Ok (200)
prototype.js	Ok (200)
qswHG82312.js	Ok (200)
TabPage.js	Ok (200)
util.js	Ok (200)
wforms_core.js	Ok (200)
wforms_validation.js	Ok (200)
styles	Not Found (404)

通过爬虫，查看系统具有哪些目录

目录名会透漏使用的插件或框架等信息

如图：fckeditor代表使用了fckeditor文本编辑器，其存在大量漏洞，可以在具体的实施中尝试确认



渗透实施——详细的WEB结构&信息收集

目的：了解系统结构，初步探明开发框架和开发语言。

实施方法：爬虫、手动浏览、session&csrf&后缀等标识查看

JSESSIONID —— Java

PHPSESSID —— Php

Sessoid —— python或其他

Aspsessionid —— asp

目录也可以用于判断框架

大部分扫描器其中带有开发语言识别功能，遇到不熟悉的系统或不熟悉的开发框架可以使用扫描器的类似功能

一些例外：

Name	Value	Domain
bmxy_auth	5159YH%2FRp4ahWUA9QabkCFZs4wnE6...	.baimaoxueyuan

Name	Value
Hm_lpvt_9cd262d07f6b9b86b429e...	1473561055
Hm_lvt_9cd262d07f6b9b86b429e...	1473561010
PengingID	5322D88C95...210E6A6D
PortallD	59a0708f5ae61a21aefaa6756c10bcb8
_qdda	3-1.1
_qddab	3-59v4kf.isy06hkk
_qddamta_4009886888	3-0
serviceurl	""
tencentSig	5761124352

渗透实施——Web扫描

目的：发现系统存在的中间件漏洞、部分框架漏洞、SQL注入、反射式跨站、列目录等问题。

实施方法：扫描器自动化扫描+人工介入验证

The screenshot displays the Nessus web interface. The top navigation bar includes 'Scans' and 'Policies'. The main content area is titled 'Scans' and shows a table of scan results. The table has columns for 'Name', 'Schedule', and 'Last Modified'. There are four rows of scan results, each with a checkbox, a name, a schedule of 'On Demand', and a date. The dates are July 3, June 14, June 9, and June 1. The interface also includes a sidebar with navigation links like 'Dashboard', 'Targets', 'Vulnerabilities', 'Scans', 'Reports', and 'Settings'.

Name	Schedule	Last Modified
[Redacted]	On Demand	July 3
[Redacted]	On Demand	June 14
[Redacted]	On Demand	June 9
[Redacted]	On Demand	June 1



渗透实施——SQL注入漏洞验证和遗漏查找

目的：验证扫描器发现的SQL注入漏洞，发现扫描器没有发现的SQL注入漏洞。

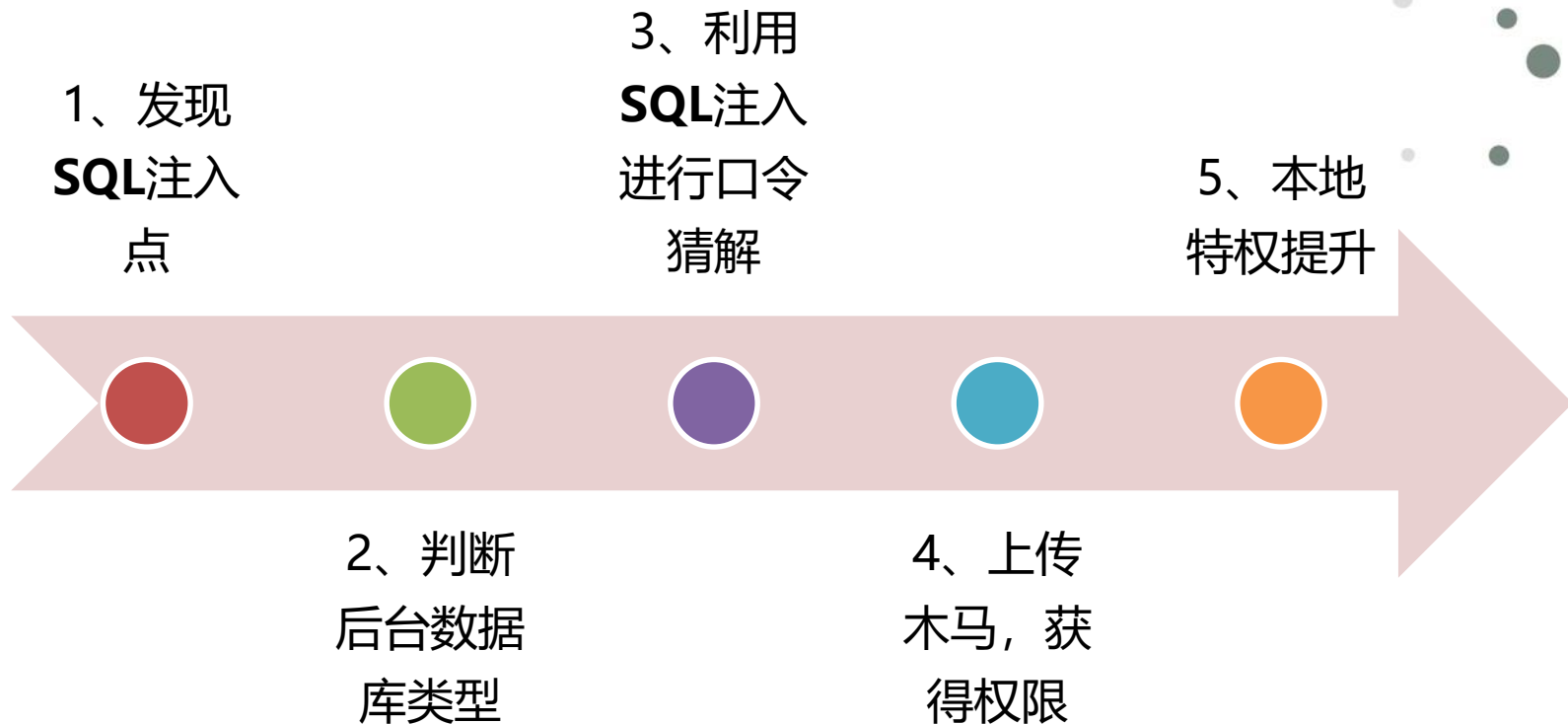
实施方法：人工验证

验证工具：SQLMAP -- 加载http内容和 参数

可选择具有验证、或者自动化爬虫无法爬到的页面，将页面的请求保存下来。从而避免扫描器遗漏。

对于无法使用sqlmap的手动输入注入语句查找。

SQL注入攻击步骤



◀ SQL注入攻击步骤-1

检测注入点的方法：

- ❑ 手工检查，通过手工查找URL参数、搜索框等
- ❑ 自动化扫描，如绿盟WEB应用漏洞扫描系统WVSS、AppScan、WVS等
- ❑ Google Hacking，目标关键字 inurl:*.asp?id=XX

SQL注入攻击步骤-1

验证注入点:

□ 整数型参数

- `http://host/test.php?id=100'`
- `http://host/test.php?id=100 and 1=1`
- `http://host/test.php?id=100 and 1=2`

SQL错误
返回正常
返回 不正常

□ 字符型参数

- `http://host/test.php?name=rainman '`
- `http://host/test.php?name=rainman ' and '1' = '1`
- `http://host/test.php?name=rainman ' and '1' = '2`

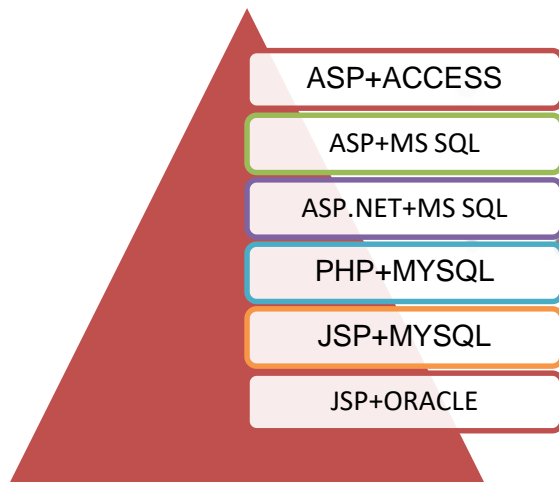
SQL错误
返回正常
返回不正常

SQL注入攻击步骤-2

□ 通过常见构架判断数据库类型:

□ 使用报错信息判断数据库类型:

- **Oracle:** **ORA**-01756: 括号内的字符串没有正确结束
- **Mysql:** ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your **MySQL** server version for the right syntax to use near
- **MS SQL:** [**Microsoft**][ODBC SQL Server Driver][SQL Server]字符串' '之前有未闭合的引号



◀ SQL注入攻击步骤-2

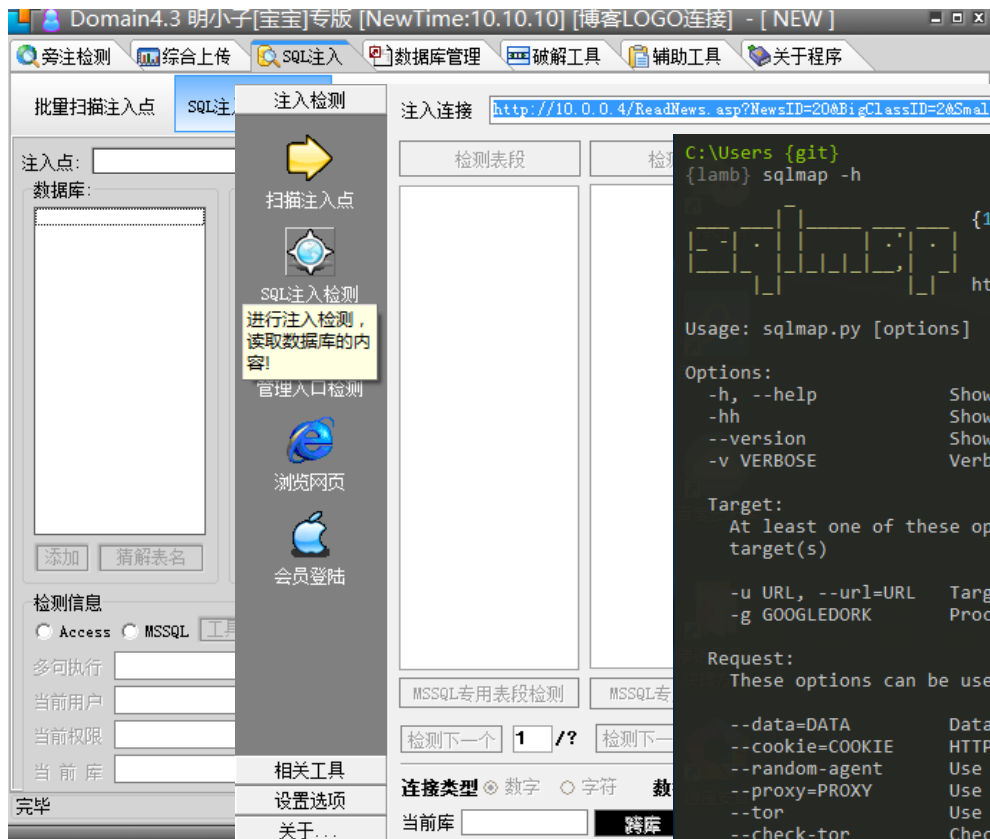
□ 后台管理界面:

- 猜测或者利用自动化工具找出系统的管理页面
- 利用猜测的用户名和密码登陆系统管理后台。
- 利用提供的上传/下载文件等功能上传ASP后门

□ 进一步本地权限提升:

- 利用系统或某些特权应用服务(如**Serv-U**)安全漏洞
- 利用系统配置不当提升系统权限

自动化SQL注入攻击工具



```
C:\Users\{git}
{lambda} sqlmap -h

{1.0.6.28#dev}

http://sqlmap.org

Usage: sqlmap.py [options]

Options:
-h, --help          Show basic help message and exit
-hh                Show advanced help message and exit
--version          Show program's version number and exit
-v VERBOSE         Verbosity level: 0-6 (default 1)

Target:
At least one of these options has to be provided to define the
target(s)

-u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
-g GOOGLEDORK       Process Google dork results as target URLs

Request:
These options can be used to specify how to connect to the target URL

--data=DATA         Data string to be sent through POST
--cookie=COOKIE     HTTP Cookie header value
--random-agent      Use randomly selected HTTP User-Agent header value
--proxy=PROXY       Use a proxy to connect to the target URL
--tor               Use Tor anonymity network
--check-tor         Check to see if Tor is used properly
```

自动化SQL注入攻击工具--Sqlmap

- `sqlmap -u "http://url/news?id=1" --current-user` #获取当前用户名称
- `sqlmap -u "http://url/news?id=1" --current-db` #获取当前数据库名称
- `sqlmap -u "http://url/news?id=1" --tables -D "db_name"` #列表名
- `sqlmap -u "http://url/news?id=1" --columns -T "tablename" -D "db_name"` #列字段
- `sqlmap -u "http://url/news?id=1" --dump -C "column_name" -T "table_name" -D "db_name"` #获取字段内容
- `sqlmap -u "http://url/news?id=1" --dbms "Mysql"` #指定数据库类型
- `sqlmap -u "http://url/news?id=1" --dbs` #列数据库
- `sqlmap -u "http://url/news?id=1" --privileges` #查看权限
- `sqlmap -u "http://url/news?id=1" --roles` #枚举数据库用户角色
- `sqlmap -u "http://url/news?id=1" --os-cmd=whoami` #执行系统命令
- `sqlmap -u "http://url/news?id=1" --sql-shell` #执行指定sql命令
- `sqlmap -u "http://url/news?id=1" -b` #获取banner信息
- `sqlmap -u "http://url/news?id=1" --reg-read` #读取win系统注册表

◀ 渗透实施——反射型XSS确认

目的：验证扫描器发现的反射型XSS漏洞。

实施方法：人工验证

验证工具：老式浏览器

查看漏洞扫描结果。

将扫描结果中的特殊字符带入到特定的参数中，并在返回的页面中查找。如果< > & ' " 等需要转义的字符没有被转义可以初步确认存在。

自行根据html上下文编辑代码带入到特定的参数中，查看是否可以弹框等。



渗透实施——跨路径、列目录、文件下载确认

目的：验证扫描器发现的跨路径和列目录等漏洞。

实施方法：人工验证

验证工具：浏览器

查看漏洞扫描结果。

在浏览器中输入结果中的跨路径特征字符，查看能否下载或打开系统文件等。

如：../../../etc/passwd等

典型的跨路径和列目录

← → ↻ [tiku.huatu.com/index.php?act=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2F](#)

```

root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/r
sync:x:5:0:sync:/sbin:/bin/sync mail:x:8:8:mail:/var/mail:/bin/mailx n games
uucp:x:10:14:uucp:/var/spool/uucp:/bin/rsh n games
ftp:x:14:50:FTP User:/var/ftp:/bin/rsh :81:Sys
saslauth:x:499:76:"Saslauth"::/etc/passwd:/bin/rsh /spool/p
ntp:x:38:38::/etc/ntp:/bin/rsh n mysql
  
```

← → ↻ [ebs.midea.com.cn/web/plugin/Excel/](#)

Index of /web/plugin/Excel/

Name	Last Modified	Size
Parent Directory		
User Import 1.xls	18-10-2013 15:59	142k
User Import 10056.xls	27-12-2011 15:07	52k
User Import 10137.xls	12-12-2012 14:24	14k
User Import 10165.xls	17-02-2012 14:13	19k
User Import 10178.xls	27-03-2013 14:46	91k
User Import 10183.xls	28-03-2012 10:45	25k
User Import 10184.xls	21-02-2012 17:04	158k
User Import 10203.xls	25-09-2012 14:16	13k
User Import 10229.xls	30-08-2012 10:33	13k
User Import 10376.xls	16-02-2012 15:25	35k
User Import 1039.xls	31-07-2013 13:18	14k
User Import 1042.xls	31-07-2013 13:29	14k
User Import 10479.xls	21-03-2012 15:19	14k
User Import 10489.xls	08-08-2012 17:25	14k
User Import 10491.xls	31-12-2011 11:46	13k
User Import 10546.xls	13-12-2012 11:46	13k
User Import 10551.xls	16-07-2013 08:29	58k
User Import 10594.xls	15-07-2013 08:10	68k
User Import 10602.xls	24-08-2012 13:19	43k
User Import 10658.xls	16-07-2013 09:37	23k
User Import 10688.xls	12-03-2012 13:48	13k
User Import 1078.xls	06-09-2013 11:44	15k
User Import 10845.xls	05-04-2012 11:31	13k
User Import 110.xls	05-01-2012 13:49	111k
User Import 1100.xls	20-04-2012 15:07	21k
User Import 1103.xls	10-04-2012 11:21	19k
User Import 11040.xls	27-09-2012 17:16	21k
User Import 11110.xls	17-02-2012 08:58	17k
User Import 11111.xls	09-01-2012 18:09	341k
User Import 11122.xls	07-04-2013 10:52	25k
User Import 11257.xls	27-12-2011 17:07	153k
User Import 11257.xls	18-07-2013 12:45	24k
User Import 11316.xls	09-01-2012 17:13	23k
User Import 1136.xls	20-09-2012 18:11	13k
User Import 11391.xls	29-12-2011 15:09	17k
User Import 11414.xls	11-07-2012 15:34	12k

www.wooyun.org

◀ 渗透实施——框架漏洞确认

目的：验证扫描器发现的框架漏洞。

实施方法：人工验证

验证工具：特定的验证工具

WebLogic Unserialization GetShell&CMD Exploit by rebeyond

Host: Port:

FilePath:

CMD:

Result: OK!!!!
如需上传文件，此文本框填写上传文件内容。



渗透实施——中间件漏洞确认

目的：验证扫描器发现的框架漏洞。

实施方法：人工验证

验证工具：特定的验证工具和验证方式

常见的中间件漏洞：特殊字符限制绕过、文件解析漏洞、

- IIS6文件夹解析漏洞 /test.asp/111.jpg（忽略/后的部分）
- IIS6文件名解析漏洞 /test.asp;1.jpg（忽略;后的部分）
- Apache文件名解析漏洞 /test.php.xxx（忽略.后的部分）
- Nginx cgi模式文件名解析漏洞 /test.jpg/a.php
 - 注：攻击者在/test.jpg后加上a.php就可以实现让nginx以php来解析任何类型的文件了。



渗透实施——敏感信息泄漏漏洞确认

目的：验证扫描器发现的敏感信息泄漏漏洞。

实施方法：人工验证

验证工具：浏览器，根据扫描器结果反馈查看浏览器中的错误信息是否属于敏感信息



渗透实施——人工查找的基础漏洞

- 命令执行
- 文件上传
- 后台爆破
- 弱口令爆破
- 错误处理不当等

◀◀ 渗透实施——命令执行漏洞

目的：挖掘命令执行漏洞。

实施方法：人工实施

验证工具：浏览器

除了框架和插件漏洞外，命令执行漏洞常见于一些可以输入系统目录、站点目录、用于系统执行相关命令的IP&域名、用于系统执行相关命令的用户名等处



典型的跨路径和列目录



网络设置

无线设置

NAT

防火墙

QoS

系统管理

系统

系统管理

工作模式

固件升级

设置管理

系统重启

系统工具:

选择命令:

PING

PING 包次数:

4

(4-20)

主机名或 IP 地址:

8.8.8.8|cat /etc/passwd

admin:/nZTJ3XW2S6Yo:0:0:admin:./bin/sh



渗透实施——文件上传

目的：挖掘命令执行漏洞。

实施方法：人工实施

验证工具：浏览器

利用插件、框架和富文本编辑器等实现文件上传时，使用了存在漏洞的插件版本

自行实现的上传功能没有对文件类型、MIME以及特殊字符如“../”等进行限制和过滤



典型的上传漏洞



◀◀ 文件上传漏洞

- 客户端检测绕过(javascript 检测)
- 服务端检测绕过(MIME 类型检测)
- 服务端检测绕过(文件扩展名检测)
- 服务端检测绕过(目录路径检测)
- 服务端检测绕过(文件内容检测)
- 解析攻击

◀◀ 渗透实施——管理员弱口令爆破

- 目的：发现可能存在的后台。
- 实施方法：人工实施
- 实施工具：burpsuite、X-scan、流光等
- 对于不存在验证码等措施的登陆点，使用弱口令字典进行爆破



口令爆破--Burpsuite

Intruder attack 2

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length	histor...	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	1608	<input checked="" type="checkbox"/>	baseline request
1	administrator	joekoe	200	<input type="checkbox"/>	<input type="checkbox"/>	1608	<input checked="" type="checkbox"/>	
2	sa	joekoe	200	<input type="checkbox"/>	<input type="checkbox"/>	1606	<input checked="" type="checkbox"/>	
3	joekoe	joekoe	302	<input type="checkbox"/>	<input type="checkbox"/>	520	<input type="checkbox"/>	
4	tomcat	joekoe	200	<input type="checkbox"/>	<input type="checkbox"/>	1606	<input checked="" type="checkbox"/>	
5	root	joekoe	200	<input type="checkbox"/>	<input type="checkbox"/>	1606	<input checked="" type="checkbox"/>	
6	administrator	nsfocus2014	200	<input type="checkbox"/>	<input type="checkbox"/>	1606	<input checked="" type="checkbox"/>	
7	sa	nsfocus2014	200	<input type="checkbox"/>	<input type="checkbox"/>	1606	<input checked="" type="checkbox"/>	
8	joekoe	nsfocus2014	200	<input type="checkbox"/>	<input type="checkbox"/>	1606	<input checked="" type="checkbox"/>	

Request Response

Raw Params Headers Hex

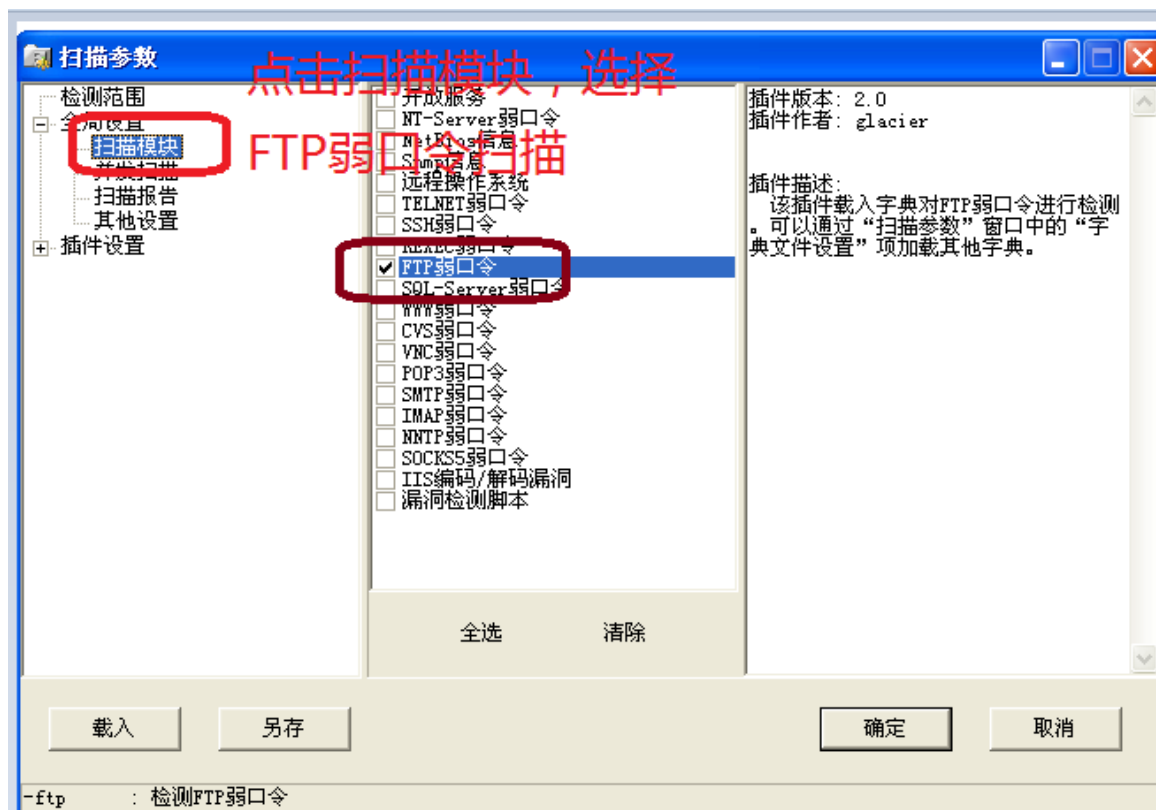
POST /ceshi/admins_login.asp? HTTP/1.1
Host: 10.84.20.172

? < + > Type a search term 0 matches

Finished

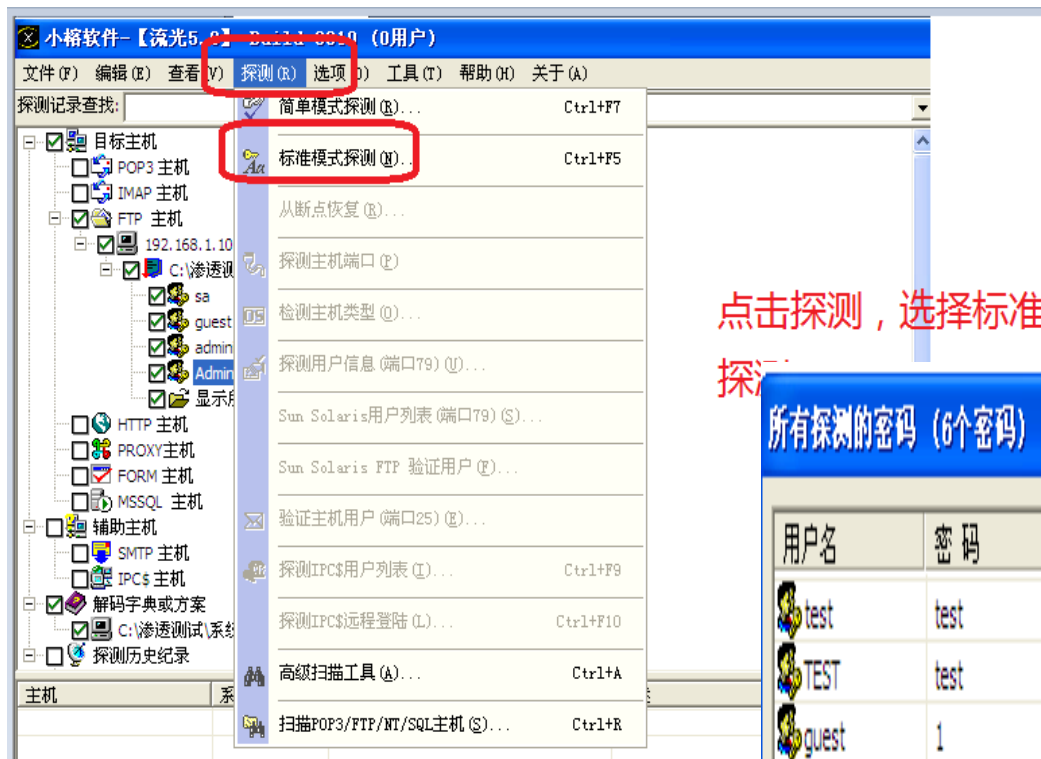


口令爆破—X-scan





口令爆破—流光



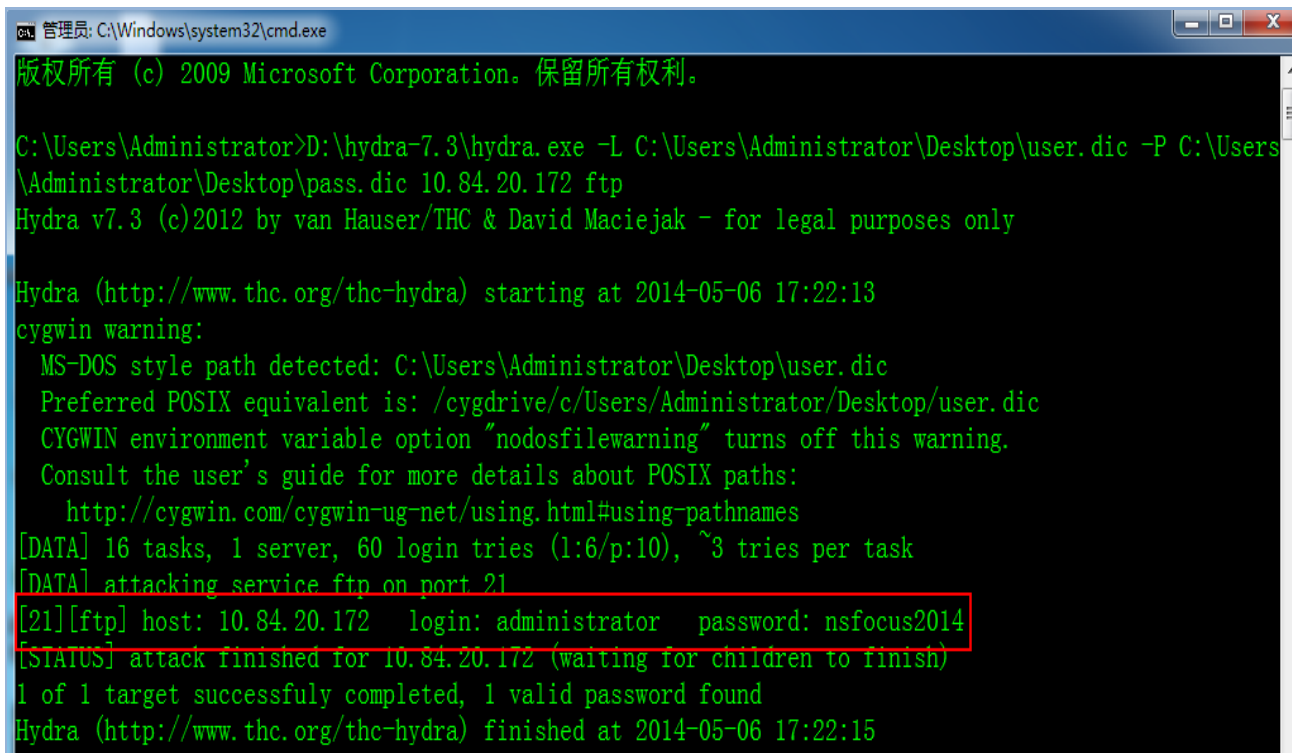
点击探测，选择标准
探测

所有探测的密码 (6个密码)

用户名	密码	主机	端口
test	test	192.168.1.107	FTP
TEST	test	192.168.1.107	FTP
guest	1	192.168.1.107	FTP
administrator	hellonetwork	192.168.1.107	FTP

◀◀ 口令爆破—Hydra

- Hydra -L user.dic -P pass.dic ip service



```
管理员: C:\Windows\system32\cmd.exe
版权所有 (c) 2009 Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>D:\hydra-7.3\hydra.exe -L C:\Users\Administrator\Desktop\user.dic -P C:\Users\Administrator\Desktop\pass.dic 10.84.20.172 ftp
Hydra v7.3 (c)2012 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2014-05-06 17:22:13
cygwin warning:
  MS-DOS style path detected: C:\Users\Administrator\Desktop\user.dic
  Preferred POSIX equivalent is: /cygdrive/c/Users/Administrator/Desktop/user.dic
  CYGWIN environment variable option "nodosfilewarning" turns off this warning.
  Consult the user's guide for more details about POSIX paths:
    http://cygwin.com/cygwin-ug-net/using.html#using-pathnames
[DATA] 16 tasks, 1 server, 60 login tries (1:6/p:10), ~3 tries per task
[DATA] attacking service ftp on port 21
[21][ftp] host: 10.84.20.172 login: administrator password: nsfocus2014
[STATUS] attack finished for 10.84.20.172 (waiting for children to finish)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2014-05-06 17:22:15
```



口令爆破—fast RDP Brute

Fast RDP Brute GUI v2.0 by Stas'M | rdpthread by ROleg

Start scan

Max. threads: 1000 Scan ports: 3389 Enter IP ranges to scan: 10.84.20.172

Scan timeout: 2000 Thread timeout: 60000

Statistics

- 1 Hosts left
- 0 Good
- 0 Bad
- 0 Error

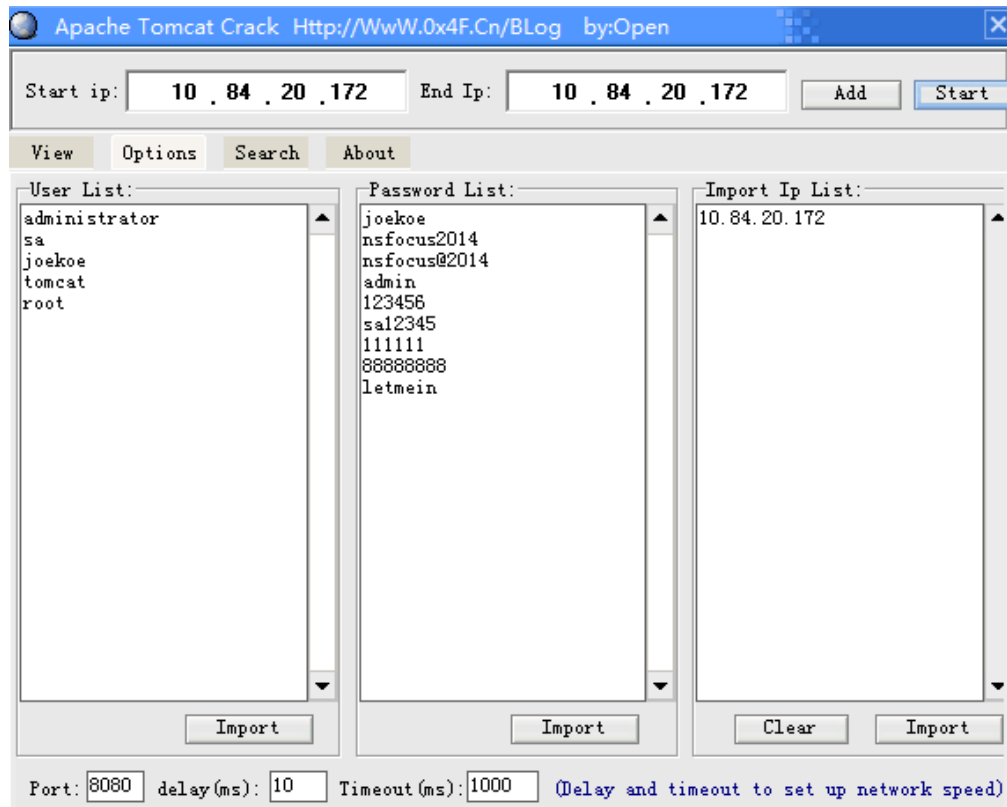
PPS: 0

Copyright © Stas'M Corp. 2012
<http://stascorp.com>

IP-address	Port	Status	User	Password
10.84.20.172	3389	Good pair found!	administrator	nsfocus2014



口令爆破—Apache Tomcat Crack



◀◀ 渗透实施——错误处理不当

- 目的：发现可能存在的不合理的错误处理。
- 实施方法：人工实施
- 实施工具：浏览器
- 错误处理可能性较多，需根据实际情况来判断和处理



渗透实施——人工查找的逻辑漏洞

- 越权操作
- 前端验证绕过
- 短信炸弹、各种炸弹
- 任意注册、任意重置、任意登录

◀◀ 渗透收尾——整理报告





谢谢!