域名系统安全实验

金舒原
jinshuyuan@mail.sysu.edu.cn
计算机学院

提纲

1. 域名系统(DNS)概述
   - 认识域名-网络中主机的标识
   - 互联网域名空间
   - 域名数据库的内容
     - 资源记录
     - 区域
   - 互联网域名解析
   - DNS协议
2. 域名系统常见攻击、攻击原理及防护措施
   - 缓存污染攻击
   - 拒绝服务攻击
   - DNSSEC
   - 加密DNS协议
3. 实验说明
   - 实验目的和实验环境
   - 实验1: DNS缓存污染攻击
   - 实验2: DNS拒绝服务攻击

从认识域名开始

域名由若干个标签(label)组成，标签之间用点号(.)分隔

- 标签允许包含的字符：英文大小写字母、数字、短横线

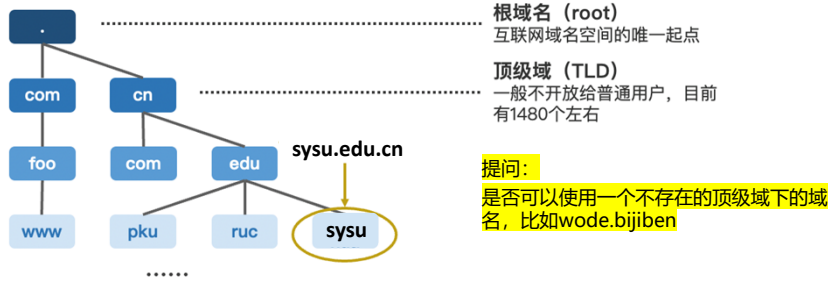
类比：倒过来的邮寄地址 **中山楼.中山大学.海珠区.广州市.中国**

思考： 根据以上特征，可以用一个什么样的数据结构来管理互联网中的所有域名？

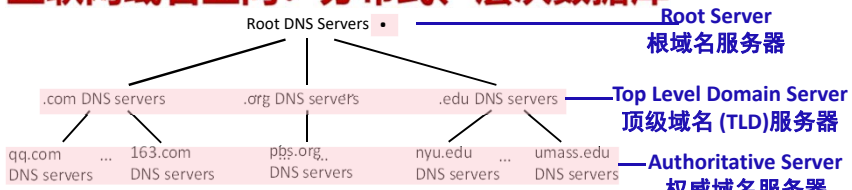
互联网域名空间

互联网域名空间呈现一个树形结构

- 唯一的根结点：根域名(用点号表示，在域名书写中通常省略)
- 确保域名具有排他性，是**互联网主机的命名空间**



互联网域名空间：分布式、层次数据库



根域名服务器：

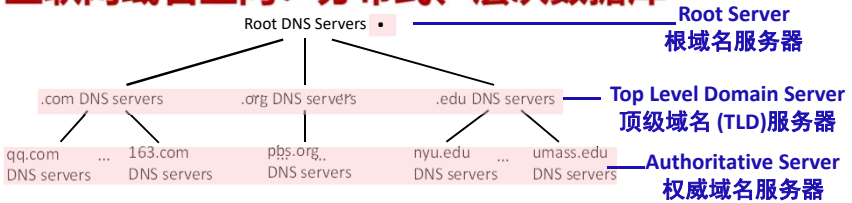
- ICANN(互联网名称与数字地址分配机构)管理根DNS域

顶级域(TLD)服务器：

- 负责.com，.org，.net，.edu，.aero，.jobs，.museums和所有顶级国家/地区域名，例如：.cn，.uk，.fr，.ca，.jp
- 提供权威DNS服务器的IP地址
- 每个顶级域(如.com)都有TLD服务器或者集群
  - Verisign Global Registry Services公司：维护com顶级域的TLD服务器
  - Educause公司：维护edu顶级域的TLD服务器

5

互联网域名空间：分布式、层次数据库



权威(域名)服务器：

- 维护“数据库”的实体机器，组织结构与域名空间结构相似
- 由域名所有者自行搭建(或外包)，负责存储区域、响应域名查询

6

域名数据库的内容：资源记录和区域

资源记录(RR)：域名映射关系的表示形式

域名	缓存时间	资源类别	资源类型	资源值
www.edu.cn	3600	IN(默认)	A	202.205.109.203

- www.edu.cn对应的IPv4地址是202.205.109.203
- 这条记录可以被域名服务器缓存1小时

常见的资源类型代号

资源类型	含义	资源类型	含义
A	IPv4地址	AAAA	IPv6地址
NS	权威服务器名称	MX	邮件服务器名称
CNAME	域名别名	SOA	起始授权信息

域名数据库的内容：资源记录和区域

资源记录(RR)：域名映射关系的表示形式

域名	缓存时间	资源类别	资源类型	资源值
www.edu.cn	3600	IN(默认)	A	202.205.109.203

- www.edu.cn对应的IPv4地址是202.205.109.203
- 这条记录可以被域名服务器缓存1小时

常见的资源类型代号

资源类型	含义	资源类型	含义
A	IPv4地址	AAAA	IPv6地址
NS	权威服务器名称	MX	邮件服务器名称
CNAME	域名别名	SOA	起始授权信息

8

域名数据库的内容：资源记录和区域

区域(zone)：一系列资源记录构成的集合

- 某个域名的区域，一般指**和它相关的所有资源记录**(即它的完整“数据库”)，包含它的所有资源记录，也可能包含其子域名的资源记录(如“通往下一级的路径”)

域名edu.cn的区域(部分)				
edu.cn	172800	IN	NS	dns.edu.cn
• edu.cn自己的权威域名服务器名称是dns.edu.cn (edu.cn这个域名并没有指定IP地址，所以没有A/AAAA记录)				
sysu.edu.cn	76255	IN	NS	ns1.sysu.edu.cn
• sysu.edu.cn(下一级)的权威服务器名称是ns1.sysu.edu.cn				
ns1.sysu.edu.cn	75682	IN	A	202.116.64.1
• sysu.edu.cn(下一级)的权威服务器地址是202.116.64.1				

互联网域名的解析

DNS出现之前，互联网上是如何进行计算机名称解析的?

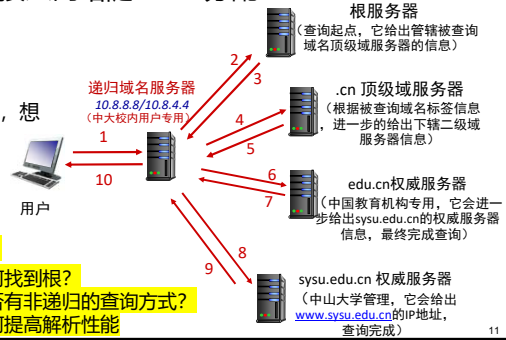
Hosts文件，\Windows\System32\Drivers\etc目录下

互联网域名的解析

- 递归域名服务器：全权代理域名解析操作
  - 何谓“递归”：从根服务器开始，根据权威服务器提供的“线索”，查到最终结果为止
- 递归域名服务器默认情况下由接入网络随DHCP分配
  - 也可由用户自行在操作系统中指定

示例: 校园用户使用自己的设备访问网络，想得到www.sysu.edu.cn的IP地址

- 所联系的服务器回复需要联系的服务器
- “我不知道这个名字，但是问这个服务器”



提问:
1. 如何找到根?
2. 是否有非递归的查询方式?
3. 如何提高解析性能

递归域名服务器 (Recursive Resolver/Server)
通常也是本地域名服务器 (Local DNS Name Server)

- 不严格属于DNS服务器的层次结构中
- 每个ISP(居民ISP，公司，大学)都有一个本地DNS服务器
  - 也称为“默认名称服务器”
- 主机进行DNS查询时，查询将发送到其本地DNS服务器
  - 本地DNS服务器通常邻近本主机
  - 具有名称到地址转换对(name-to-address translation pairs)的本地缓存(但可能已过期！)
  - 充当代理，将查询转发到DNS服务器的层次结构中

提高域名系统的解析性能-缓存DNS信息

- 一旦某个DNS服务器接收到一个DNS回答，它将*缓存*该映射
  - 一段时间(TTL，生存时间，通常是2天)后，缓存条目将被丢弃
  - 通常缓存在本地DNS服务器中
  - 因此，根域名服务器并不经常被访问
  - 缓存未过期前，新的查询请求将直接使用缓存应答；
  - 缓存过期后，缓存条目将被丢弃。
- 缓存的条目可能已**过期**(尽力而为的名称到地址的转换！)
  - 如果主机更改了其IP地址，则在所有TTL都到期之前，可能无法在Internet范围内被知道！
- 更新/通知机制建议的IETF标准
  - RFC 2136

互联网域名的解析

DNS应用层协议：使得主机能够查询分布式数据库的应用层协议、实现名称/IP地址转换

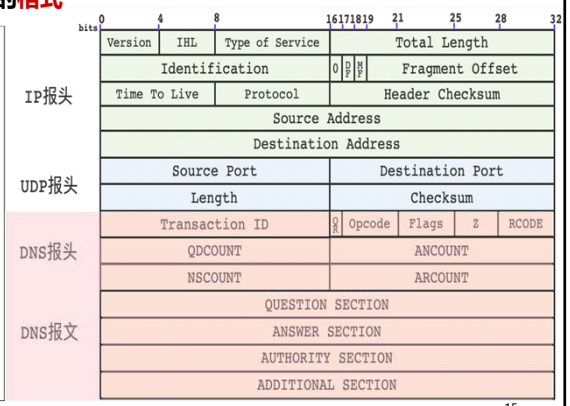
- DNS系统采用客户机/服务器架构，使用的传输层协议为TCP或UDP，服务器端口号53
- DNS服务器之间是TCP，Client与DNS服务器之间是UDP
- DNS服务器通常是运行BIND(Berkeley Internet Name Domain)软件的UNIX机器

DNS报文格式

DNS查询和响应报文，具有相同的格式

报头部分：前12个字节

- **标识字段** (TXID):
  16位＃用于查询，对查询的回答使用相同的＃
- **标志字段** :
  QR(1位): 查询(0)响应(1)；Opcode(4 位)：指示操作类型，如查询或更新等；AA(1 位)：指示权威回答；TC(1 位)：指示报文是否被截断；RD(1 位)：指示是否递归查询；RA(1 位)：指示服务器是否支持递归查询；Z(3 位)：保留字段。RCODE(4 位)：响应码，指示响应的状态。
- **4个关于数量的字段**:
  指示DNS报文中四个部分的资源记录数量：查询、回答、授权、附加部分

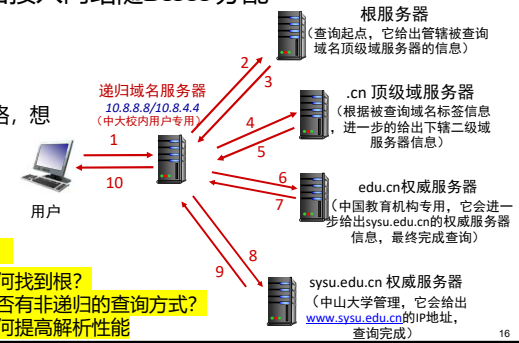


互联网域名的解析

- 递归域名服务器：全权代理域名解析操作
  - 何谓“递归”：从根服务器开始，根据权威服务器提供的“线索”，查到最终结果为止
- 递归域名服务器默认情况下由接入网络随DHCP分配
  - 也可由用户自行在操作系统中指定

示例: 校园用户使用自己的设备访问网络，想得到www.sysu.edu.cn的IP地址

- 所联系的服务器回复需要联系的服务器
- “我不知道这个名字，但是问这个服务器”



提问：
1. 如何找到根？
2. 是否有非递归的查询方式？
3. 如何提高解析性能

域名解析报文格式

如何解读下面这个DNS报文?

提问：这是一个解析请求还是响应？里面包含什么内容?

```
▸ Frame 68: 176 bytes on wire (1408 bits), 176 bytes captured (1408 bits) on interface 0
▸ Ethernet II, Src:                                    , Dst:
▸ Internet Protocol Version 4, Src:            , Dst:
▸ User Datagram Protocol, Src Port: 53, Dst Port: 56693
▾ Domain Name System (response)
    [Request In: 67]
    [Time: 0.010760000 seconds]
    Transaction ID: 0xc3fa
  ▸ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 3
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▸ www.seu.edu.cn: type A, class IN
  ▾ Answers
    ▸ www.seu.edu.cn: type CNAME, class IN, cname www-seu-edu-cn.cname.saaswaf.com
    ▸ www-seu-edu-cn.cname.saaswaf.com: type CNAME, class IN, cname seu-ipv6.cache.saaswaf.com
    ▸ seu-ipv6.cache.saaswaf.com: type A, class IN, addr 121.194.14.142
  ▸ Additional records
```


提纲

1. 域名系统(DNS)概述
   - 认识域名-网络中主机的标识
   - 互联网域名空间
   - 域名数据库的内容
     - 资源记录
     - 区域
   - 互联网域名解析
   - DNS协议

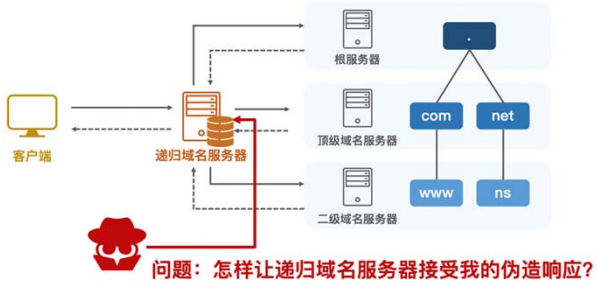
2. 域名系统常见攻击、攻击原理及防护措施
   - 缓存污染攻击
   - 拒绝服务攻击
   - DNSSEC
   - 加密DNS协议

3. 实验说明
   - 实验目的和实验环境
   - 实验1: DNS缓存污染攻击
   - 实验2: DNS拒绝服务攻击

缓存污染攻击

攻击模型：旁路注入(off-path injection)

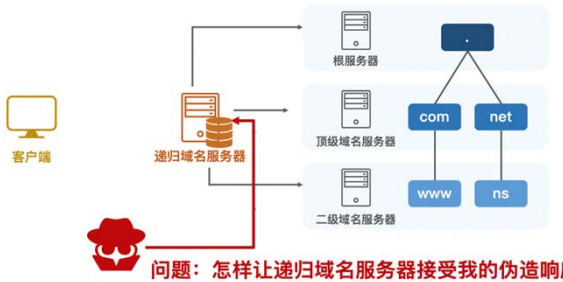
- 攻击者**并不位于域名解析链路上**，无法直接嗅探和修改报文
- 攻击者想要**注入一个伪造的响应**，令递归域名服务器接受并写入缓存



缓存污染攻击

攻击模型：旁路注入(off-path injection)

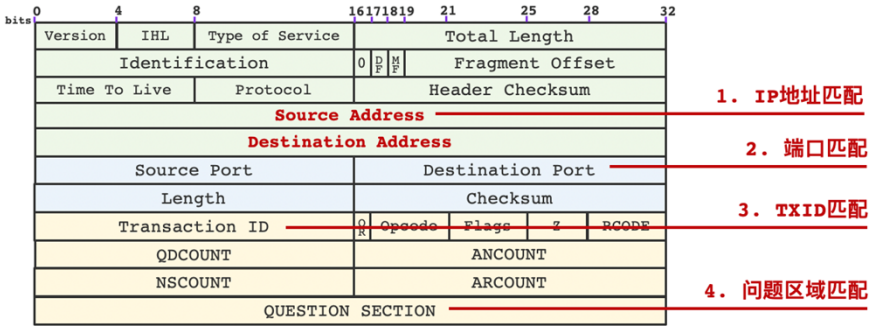
- 攻击者**并不位于域名解析链路上**，无法直接嗅探和修改报文
- 攻击者想要**注入一个伪造的响应**，令递归域名服务器接受并写入缓存



缓存污染攻击

什么样的响应会被递归域名服务器接收?

- 递归域名服务器会做什么检查?



缓存污染攻击

攻击者如何伪造符合上述条件的响应?

条件	备注	是否可控/可预知
IP地址匹配	响应源地址 = 权威服务器地址	是(通过查询实现)
	响应目的地址 = 递归域名服务器	是
端口匹配	响应源端口 = 53(DNS默认服务端口)	是
	响应目的端口 = 请求源端口	**否**
TXID匹配	响应TXID = 请求TXID	**否**
问题区域匹配	响应问题区域 = 请求问题区域	**是(为什么?)**
伪造响应先到达	伪造响应先于真实响应到达	是

缓存污染攻击

通过“自问自答”的方式，使得问题区域匹配

- 需要**猜解请求源端口 & 请求TXID**

思考:
这样的响应检查机制，提供了多大程度的保护?

如何防止，缓解 DNS缓存污染攻击?

请讨论

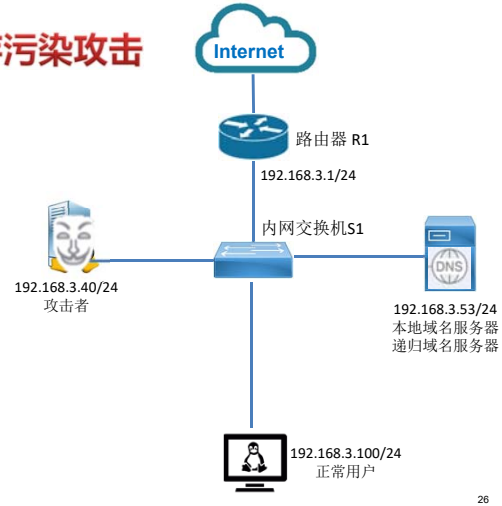
提纲

1. 域名系统(DNS)概述
   - 认识域名-网络中主机的标识
   - 互联网域名空间
   - 域名数据库的内容
     - 资源记录
     - 区域
   - 互联网域名解析
   - DNS协议
2. 域名系统常见攻击、攻击原理及防护措施
   - 缓存污染攻击
   - 拒绝服务攻击
   - DNSSEC
   - 加密DNS协议
3. 实验说明
   - 实验目的和实验环境
   - 实验1: DNS缓存污染攻击
   - 实验2: DNS拒绝服务攻击

实验1(升级版): DNS缓存污染攻击

实验目的

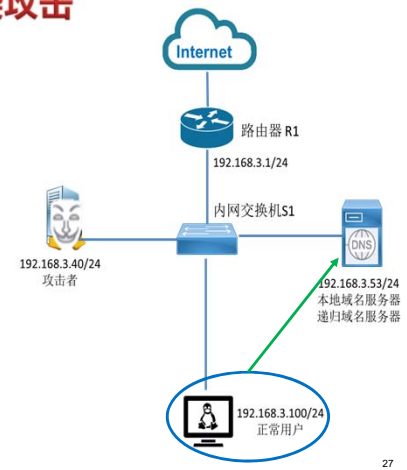
- 掌握DNS解析过程的工作原理及其关键环节
- 学会识别和分析DNS工作过程中可能遇到的攻击点
- 掌握DNS缓存污染攻击原理、执行条件、攻击过程及危害
- 思考并掌握DNS缓存污染攻击的防御策略及技术



实验1(升级版)：DNS缓存污染攻击

攻击过程要点

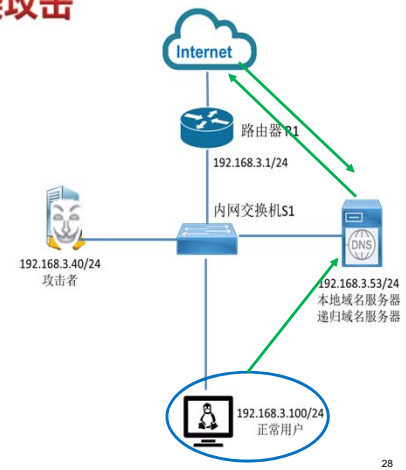
1. 正常用户（即受害者）访问位于互联网中imool.net
2. 本地DNS服务器向imool.net的权威域名服务器发起查询。
3. 攻击者编写DNS缓存污染攻击脚本，发起DNS缓存污染攻击，目的是在正常用户的域名查询响应到达之前，将钓鱼网站的IP地址作为imool.net的域名解析响应插入到本地域名服务器的DNS缓存中。
4. 攻击后结果：正常用户访问的imool.net，显示的却是钓鱼网站的内容。



实验1(升级版)：DNS缓存污染攻击

攻击过程要点

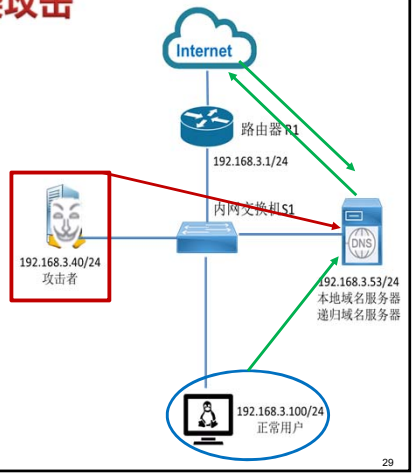
1. 正常用户（即受害者）访问位于互联网中imool.net
2. 本地DNS服务器向imool.net的权威域名服务器发起查询。
3. 攻击者编写DNS缓存污染攻击脚本，发起DNS缓存污染攻击，目的是在正常用户的域名查询响应到达之前，将钓鱼网站的IP地址作为imool.net的域名解析响应插入到本地域名服务器的DNS缓存中。
4. 攻击后结果：正常用户访问的imool.net，显示的却是钓鱼网站的内容。



实验1(升级版)：DNS缓存污染攻击

攻击过程要点

1. 正常用户（即受害者）访问位于互联网中imool.net
2. 本地DNS服务器向imool.net的权威域名服务器发起查询。
3. 攻击者编写DNS缓存污染攻击脚本，发起DNS缓存污染攻击，目的是在正常用户的域名查询响应到达之前，将钓鱼网站的IP地址作为imool.net的域名解析响应插入到本地域名服务器的DNS缓存中。
4. 攻击后结果：正常用户访问的imool.net，显示的却是钓鱼网站的内容。

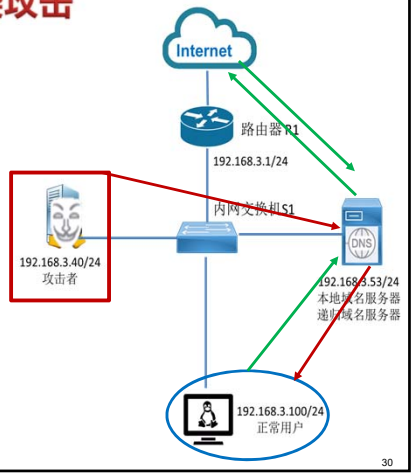


29

实验1(升级版)：DNS缓存污染攻击

攻击过程要点

1. 正常用户（即受害者）访问位于互联网中imool.net
2. 本地DNS服务器向imool.net的权威域名服务器发起查询。
3. 攻击者编写DNS缓存污染攻击脚本，发起DNS缓存污染攻击，目的是在正常用户的域名查询响应到达之前，将钓鱼网站的IP地址作为imool.net的域名解析响应插入到本地域名服务器的DNS缓存中。
4. 攻击后结果：正常用户访问的imool.net，显示的却是钓鱼网站的内容。

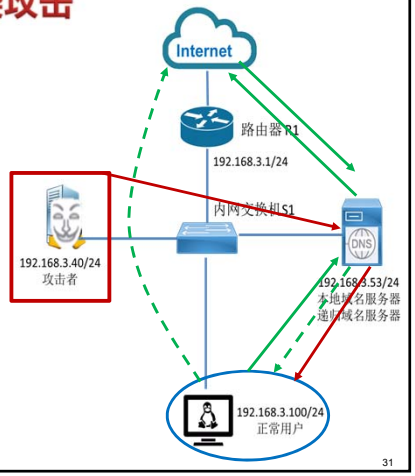


30

实验1(升级版)：DNS缓存污染攻击

攻击过程要点

1. 正常用户（即受害者）访问位于互联网中imool.net
2. 本地DNS服务器向imool.net的权威域名服务器发起查询。
3. 攻击者编写DNS缓存污染攻击脚本，发起DNS缓存污染攻击，目的是在正常用户的域名查询响应到达之前，将钓鱼网站的IP地址作为imool.net的域名解析响应插入到本地域名服务器的DNS缓存中。
4. 攻击后结果：正常用户访问的imool.net，显示的却是钓鱼网站的内容。

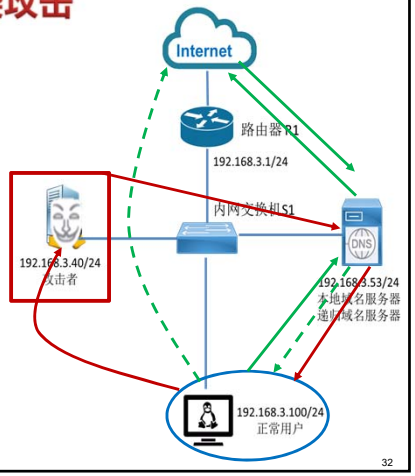


31

实验1(升级版)：DNS缓存污染攻击

攻击过程要点

1. 正常用户（即受害者）访问位于互联网中imool.net
2. 本地DNS服务器向imool.net的权威域名服务器发起查询。
3. 攻击者编写DNS缓存污染攻击脚本，发起DNS缓存污染攻击，目的是在正常用户的域名查询响应到达之前，将钓鱼网站的IP地址作为imool.net的域名解析响应插入到本地域名服务器的DNS缓存中。
4. 攻击后结果：正常用户访问的imool.net，显示的却是钓鱼网站的内容。

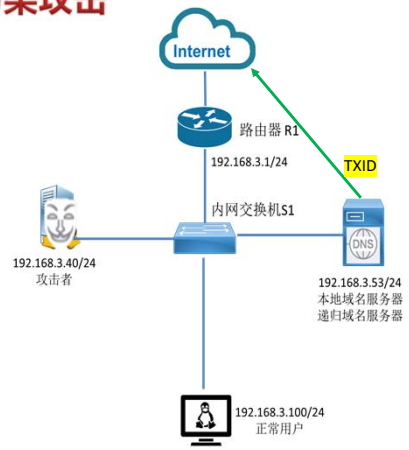


32

实验1（升级版）：DNS缓存污染攻击

攻击者能够攻击成功的关键：

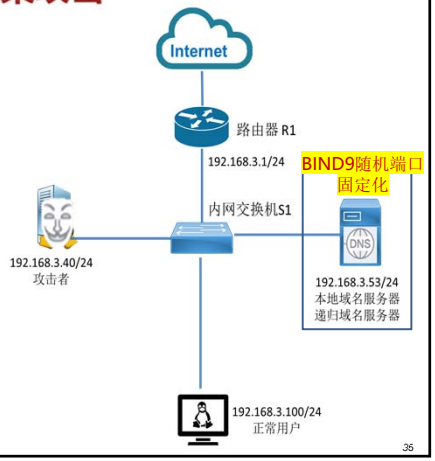
1. 在imool.net的权威域名服务器的DNS解析响应返回前，破解出正常域名的本地DNS服务器发出的查询报文的TXID字段
2. 伪造DNS 响应报文，该响应报文中的回答部分是钓鱼网站的IP 地址
3. 将响应报文插入到本地域名服务器缓存



实验1（升级版）：DNS缓存污染攻击

简化实验难度的操作：

1. 配置正常用户的本地DNS服务器，使BIND9对外发起DNS解析请求的随机端口固定化
2. 互联网中的imool.net权威域名服务器已设置了响应时间延迟，以增加攻击者在正常响应到达之前污染本地DNS服务器缓存的可用时间段长度。
3. 互联网中的imool.net相关资源记录的TTL值已设置为10秒，以便若本地域名服务器已缓存正常的RR记录的失效时间更快。
4. imool.net域名已开启了泛域名解析，可以自行选定子域名作为攻击对象。如dnslab.imool.net，sysu.imool.net 等等。



实验内容： DNS缓存污染攻击(升级版)

1. 在攻击者主机上，查询imool.net权威域名服务器地址，确定攻击脚本需用到的权威域名服务器源IP。
2. 在本地域名服务器机器上，修改本地域名服务器配置，将BIND9随机化端口设置为固定端口，并重新启动BIND9。
3. 在攻击者主机上，使用scapy编写代码，实现对DNS 响应报文的伪造。
4. 猜测DNS响应报文中的TXID字段。
5. 发起攻击。在攻击脚本中，提高发包速度。
6. 检查攻击是否成功。
7. 配置钓鱼网站。
8. 验证DNS缓存污染攻击后的钓鱼效果。
9. 将重要过程和结果截图，完成实验报告。
10. 思考：
    （1）如何验证攻击者构造的DNS响应报文是正确的?
    （2）怎样提高DNS缓存污染攻击的成功率?
    （3）怎样防范DNS缓存污染攻击?