实验二 ARP与DNS协议分析实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号： | 3-3-1 |  |  |  |  |
| 姓名： | 李云广 | 学号： | 2193712575 | 班级： | 计算机93 |
| 姓名： | 李怀邦 | 学号： | 2193712530 | 班级： | 计算机93 |
|  |  |  |  |  |  |

1. 实验目的

分析ARP协议报文首部格式以及在同一网段内和不同网段间的解析过程，分析DNS协议的工作过程。

1. 实验内容

（1）利用校园网及云服务器搭建内网、外网环境；

（2）用Wireshark截获ARP报文，分析报文结构及ARP协议在同一网段和不同网段间的解析过程；

（3）用Wireshark截取DNS报文，分析DNS工作过程。

1. 实验环境与分组

每2名同学一组，以现有的校园网络环境及云服务器搭建内网、外网网络。

1. 实验网络拓扑皆否

按照实际网络情况绘制拓扑图【标注出内、外地址】。



1. 实验过程及结果分析

【过程记录应当详尽，截图并加以说明。以下过程和表格仅供参考。】

* 1. ARP协议分析

（一）同一网段内IP的ARP协议分析：

步骤1：在计算机终端的命令行窗口执行命令：

执行“arp –a”观察arp缓存；

执行“arp –d”命令清空arp缓存。

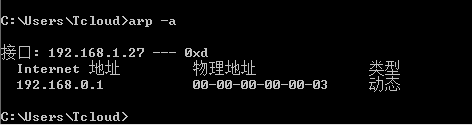
步骤2：在计算机终端上运行Wireshark截获报文，在命令行窗口ping同一网段的另一设备地址。执行完后停止报文截获，筛选出相关的arp和icmp报文进行分析（源IP地址/MAC地址、目的IP地址/MAC地址等）。

步骤3：在命令行窗口执行“arp –a”，记录结果。

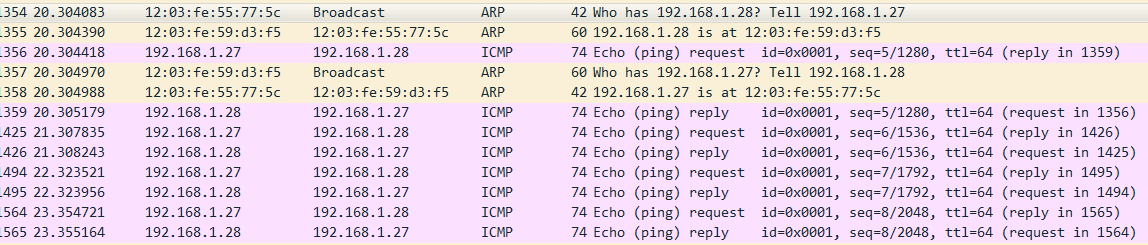
初始arp缓存：

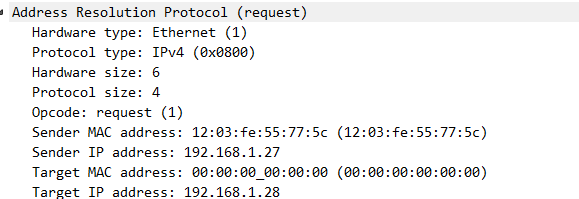
![C:\Users\Tcloud\AppData\Roaming\Tencent\Users\3048096138\QQ\WinTemp\RichOle\2G]HTA7_O9Z]()9XZM`]4QD.png](data:image/png;base64,)

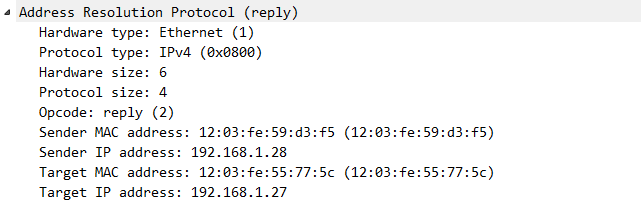
清空缓存区：



Arp报文截获：

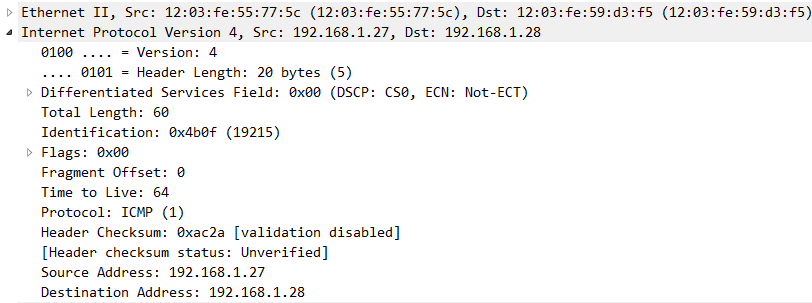


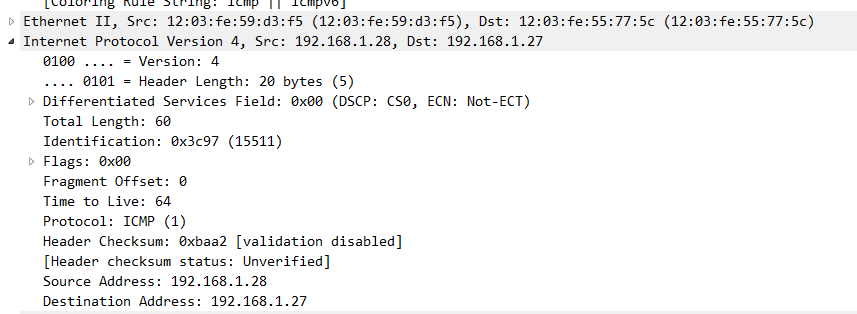




由图可以看出，第一条arp进行广播，寻找192.168.1.28的mac地址，源ip 192.168.27 mac 12：03：fe：55：77：5c，目的ip192.168.1.28 mac00：00：00\_00:00:00.之后192.168.1.28进行答复mac地址12：03：fe：59：d3:f5

Icmp报文分析：

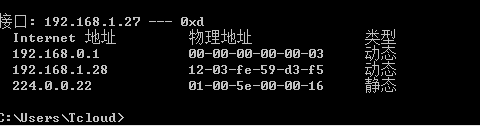




由图可以看出，源ip 192.168.27 mac 12：03：fe：55：77：5c

目的ip192.168.1.28 mac12：03：fe：59：d3:f5

Arp缓存区：



（二）不同网段的ARP协议分析

步骤1：在本地计算机和云服务器执行“arp –d”清空缓存，运行Wireshark捕获报文，在本地计算机ping云服务器地址。执行完后停止报文截获，筛选出相关的arp和icmp报文进行分析（arp与icmp报文的顺序，报文源IP地址/MAC地址、目的IP地址/MAC地址及其对应的主机等）。

【如果网卡自动解析默认网关的MAC地址，可以删除默认网关设置，添加外网路由后再试。参考命令：route delete 0.0.0.0， route add 202.0.0.0 MASK 255.0.0.0 192.168.0.1】

步骤2：执行“arp –a”命令，记录结果。

步骤3：分析捕获的报文，选中第一条ARP请求报文和第一条应答报文，填写2-1表。

表2-1 ARP请求报文和应答报文的字段信息

|  |  |  |
| --- | --- | --- |
| 字段 | 请求报文的值 | 应答报文的值 |
| 以太网链路层Destination项 | Broadcast(ff:ff:ff:ff:ff  ：ff) | 12:03:fe:55:77:5c |
| 以太网链路层Source项 | 12:03:fe:55:77:5c | 00:00:00\_00:00:03 |
| ARP报文发送者硬件地址 | 12:03:fe:55:77:5c | 00:00:00\_00:00:03 |
| ARP报文发送者IP | 192.168.1.27 | 192.168.0.1 |
| ARP报文目标硬件地址 | 00:00:00\_00:00:00 | 12:03:fe:55:77:5c |
| ARP报文目标IP | 192.168.0.1 | 192.168.1.27 |

分析捕获的报文，选中第一条ICMP请求报文和第一条应答报文，填写表2-2。（对应主机填写本机、本地网关、服务器等）

表2-2 ICMP请求报文和应答报文的字段信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 请求报文的值 | 对应主机 | 应答报文的值 | 对应主机 |
| 发送者硬件地址 | 12:03:fe:55:77:5c | 本地主机 | 00:00:00\_00:00:03 | 网关 |
| 发送者IP | 192.168.1.27 | 本地主机 | 39.106.29.224 | 服务器 |
| 目标硬件地址 | 00:00:00\_00:00:03 | 网关 | 12:03:fe:55:77:5c | 本地主机 |
| 目标IP | 39.106.29.224 | 服务器 | 192.168.1.27 | 本地主机 |

步骤4：比较ARP协议在不同网段和相同网段内解析过程的异同。

* 1. DNS协议分析

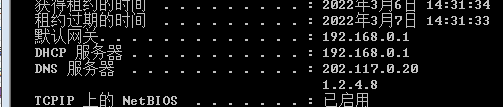
（一）默认DNS域名解析

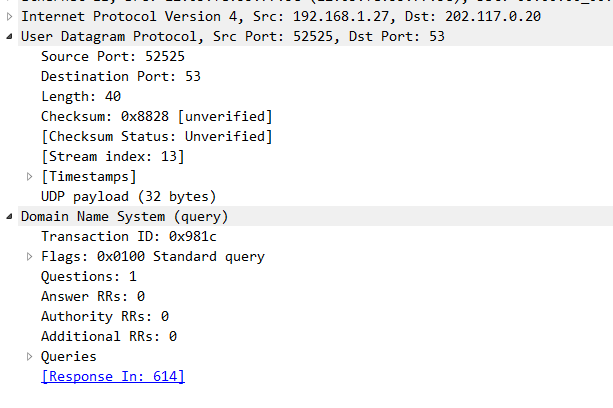
步骤1：在命令窗口执行命令：

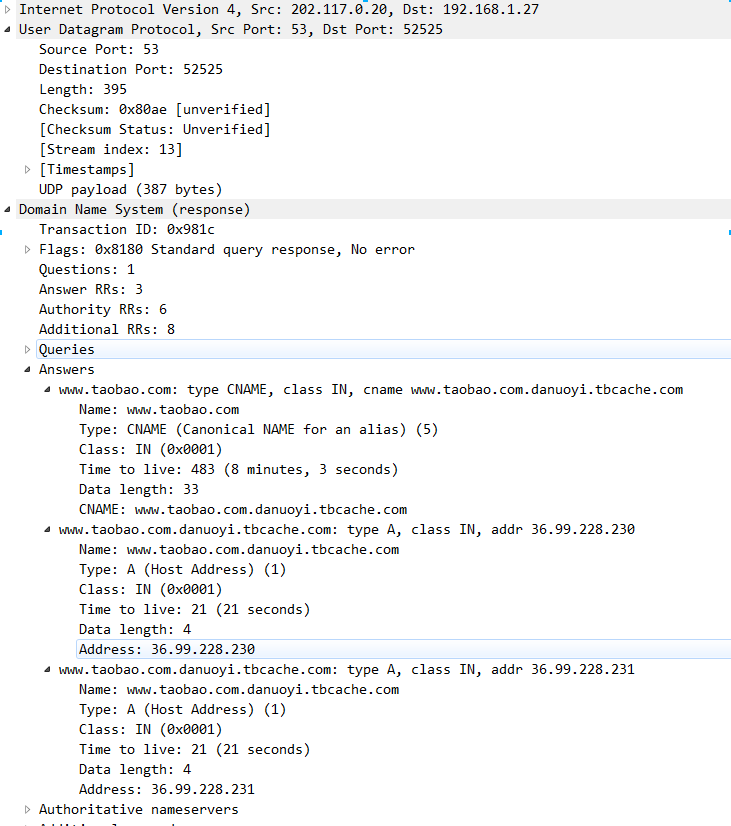
执行“ipconfig /displaydns”观察本地DNS缓存；

执行“ipconfig /flushdns”清除本地DNS缓存。

步骤2：在计算机终端上运行Wireshark截获报文，浏览器访问域名（如http://www.yahoo.com），网站打开后停止报文截获，观察分析DNS查询、回复报文分别包含哪些主要内容（UDP还是TCP、目的地址与本机默认DNS是否相同、源端口和目的端口、域名解析记录类型、解析出的IP地址等）。







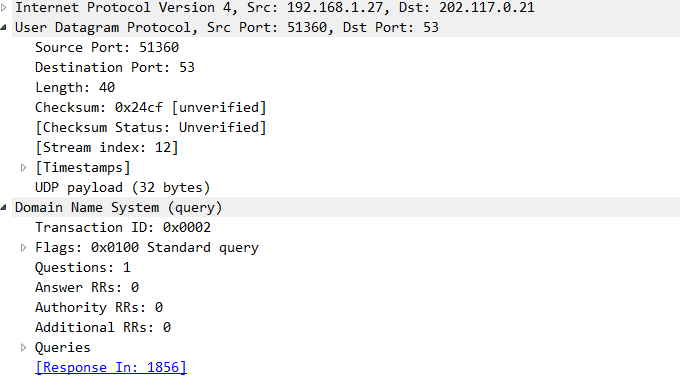
（二）指定DNS域名解析

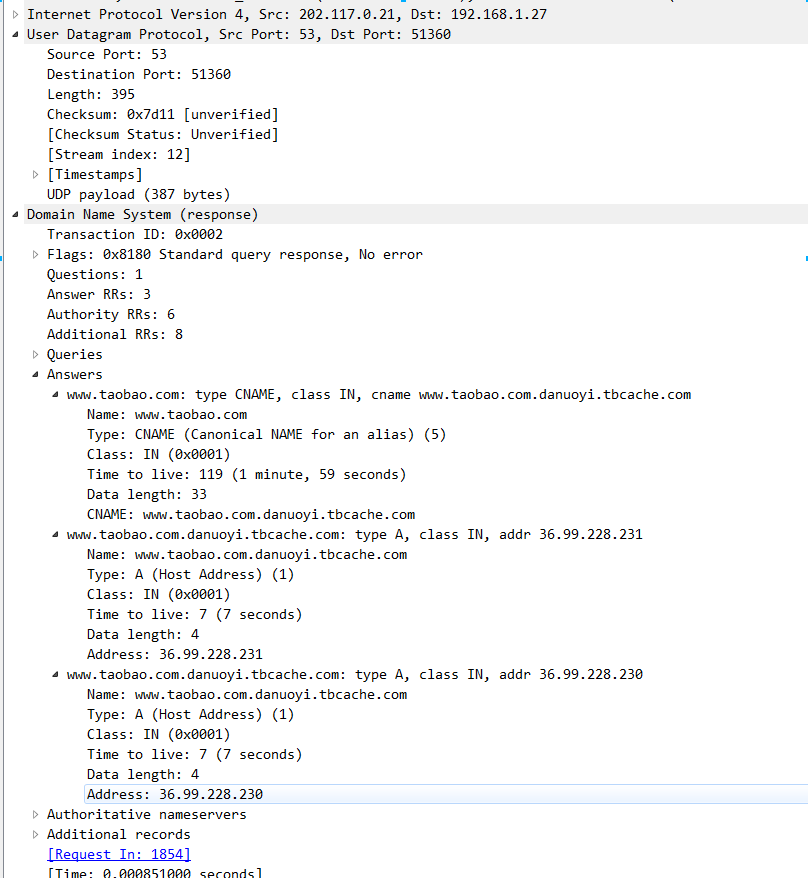
步骤1：在命令窗口执行命令：

执行“ipconfig /displaydns”观察本地DNS缓存；

执行“ipconfig /flushdns”清除本地DNS缓存。

步骤2：在计算机终端上运行Wireshark截获报文，在命令窗口执行指定DNS服务器解析域名命令（如nslookup www.synlogictx.com 223.6.6.6），解析完毕后停止报文截获，观察分析DNS查询、回复报文分别包含哪些主要内容（UDP还是TCP、目的地址与本机默认DNS是否相同、源端口和目的端口、域名解析记录类型、解析出的IP地址等）。





![C:\Users\Tcloud\AppData\Roaming\Tencent\Users\3048096138\QQ\WinTemp\RichOle\@9X9F9XF)$C{DK]U](QVWZV.png](data:image/png;base64,)

* 1. 互动讨论主题

（1）发送方与接收方ARP与ICMP报文出现的次序成因；

（2）ARP的安全性问题；

（3）DNS的欺骗带来的安全性问题；

* 1. \*进阶自设计

Scapy是一个 Python程序，它允许用户发送、嗅探、分析和伪造网络包。这种能力允许构建能够探测、扫描或攻击网络的工具。换句话说，Scapy是一个强大的交互式包操作程序。它能够伪造或解码大量协议的数据包，在网络上发送它们，捕获它们，匹配请求和响应，等等。Scapy可以轻松地处理大多数经典任务，如扫描、跟踪、探测、单元测试、攻击或网络发现。它可以代替hping、arpsoof、arp-sk、arping、p0f甚至Nmap、tcpdump和tshark的某些部分。

（1）使用scapy在Linux下写程序来模拟完成一个简单的ARP欺骗。

（2）使用scapy在Linux下写程序来模拟完成一个简单的DNS欺骗。完整的攻击实现工作量和难度都很大。为了降低难度，可以不实现中间人攻击，而是直接让受害者把DNS服务器修改为欺骗者的地址。