



科来软件 网络分析领军企业

网络分析案例集 2012

Collection of Network Analysis Case Studies

拓展网络视野 精细网络管理

前 言

由CSNA网络分析论坛携手科来软件的共同之力完成的《网络分析案例集2011》，一经刊发即获得多方的好评，成为众多网络管理者在网络分析技术方面学习和提高的必备资料。在此，我们对受到多方的关注与支持表示诚挚的感谢。

科来在为用户进行网络分析服务中发现：现代的网络，管理之复杂、要求之高、难度之大，是很难通过传统的网管技术达到目标的，尤其在面对间歇性网络故障、安全事件的追踪取证、网络容量规划以及网络性能评估方面。这就需要更加有效的手段来解决这些问题。另外，网络运维逐渐成为企业盈利的重要贡献者之一，如何实现运维的价值，也是管理者需要不断思考的问题。

网络分析，尤其是网络的回溯分析，为我们的网络管理提供了新的视野，成为解决上述问题的行之有效的技术手段，并不断为更多的网络管理者所认同与付诸实施。在《科来网络分析案例集2012》书中，就对网络回溯分析的重要应用有较为详尽的案例描述，与大家共同分享学习。更多的参考资料请登录www.colasoft.com.cn查阅，并欢迎加入CSNA网络分析论坛参加讨论。

本案例集的收集整理工作还要感谢CSNA论坛的大力支持与协助，科来软件也将一如既往的为国内技术人员提供高质量的网络分析技术资料，并致力于不断提升国内的整体技术水平。文中不足之处，欢迎大家指正。

科来软件
二零一二年四月

科来软件-网络分析领军企业

中国网络分析领军企业

科来软件成立于2001年，是以网络分析技术为核心的高新技术企业。自创建以来，一直致力于网络分析技术的研究与推动，将研究成果广泛应用于网络安全态势分析、网络故障诊断及网络性能优化等关键领域，并将该项技术的积累经验应用于技术培训和服务。科来已成为中国网络分析领域的第一品牌，并在2010年荣获中国软件协会、中国电子信息产业发展研究院联合颁发的《辉煌十年 网络分析领军企业奖》。

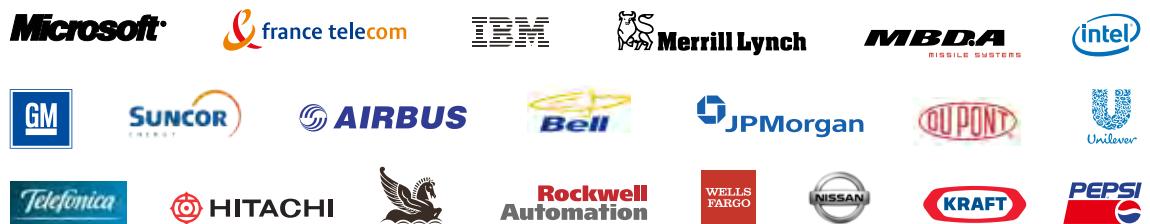
- 国内18万用户正在使用科来的技术
- 为全球5000多家商业客户提供网络分析解决方案
- 科来的技术服务于世界85个国家和地区
- 超过80家财富500强企业选择科来
- 网络分析技术认证培训受到国内高端网络管理人才青睐



国内典型客户

中国科技部	国家电网	广东省电信	中石油集团	北京海关	浙江省农业银行
中国外交部	河南省电力	湖南省电信	中石化集团	沈阳海关	中国人寿保险
中国农业部	四川省电力	湖北省电信	长庆油田	吉林省公安厅	河南中原证券
海关总署	北京华电	广东省移动	开滦煤矿	郑州市公安局	上海湘财证券
国家统计局	安徽省供电	浙江省移动	淮南矿业	焦作市公安局	重庆市商业银行
国家会议中心	江西省供电	四川省移动	神华宁煤	南通市公安局	安徽省农发行
国家电网公司	重庆市电力	江苏省移动	中国海洋石油	鹤壁市公安局	安徽省国税局
国务院新闻办	天津市电力	桂林市电信	昆仑润滑油公司	公安部等保中心	河南省地税局

海外典型客户



目 录

第一章 网络故障分析	1
1. 某银行分行和总行通信问题	1
2. 税务系统问题分析	5
3. 某移动公司BOSS系统故障分析	10
4. 某用户网络问题分析报告	15
5. 抓包验证IOS BUG的案例	22
6. 网络故障分析报告	26
7. 环路分析	33
8. 某电视台故障处理报告	41
9. TCP异常连接分析案例	49
10. 记录两次断网的分析过程	56
11. 某互联网故障分析报告	63
12. 某省天然气分析报告	69
第二章 网络安全分析	86
1. 某证券公司回溯分析案例	86
2. 某电信IDC机房——托管服务器异常行为监控	90
3. 回溯式发现、挖掘、追踪木马通信	96
4. 邮件系统攻击分析	100
5. 飞客蠕虫研究	104
6. 垃圾邮件行为分析	111
7. 一次端口扫描行为的分析案例	119
8. 回溯式异常流量分析	122
9. 黑客攻防技术入门之ARP篇	126
10. 某省某报业分析报告	132
第三章 网络应用分析	135
1. 某单位财务系统与OA系统评估	135
2. IDC出口流量梳理	141
测试申请	146
案例征集	147

欢迎加入CSNA网络分析论坛.....	148
CSNA网络分析认证培训	149
网络分析技术的推动者	149
CSNA网络分析认证培训	149
CSNA网络分析师认证培训	149
CSNA网络分析与家认证培训	149
CSNA安全分析认证培训	149

第一章 网络故障分析

1. 某银行分行和总行通信问题

1.1. 故障描述

1. 故障环境

大家好，今天和大家分享一例银行总行分行之间通信的问题，其中运用了科来 2010 专家版的最新版本。文中也将介绍一些最新版本的功能。

2. 问题描述：

6月初去了浙江一家银行进行测试，正当我们赶到银行的时候，那边网络的负责人正好向我们求助。通过沟通，得知他们一个分部与总行之间的通信总是时断时续。在分行那边没有浏览网页的时候，总行 PING 分行没有问题，一旦分行那边有浏览网页的行为等，就会通信中断。情况已经持续了一天，使用了许多办法，都没有办法定位到发生问题的点和问题的原因。

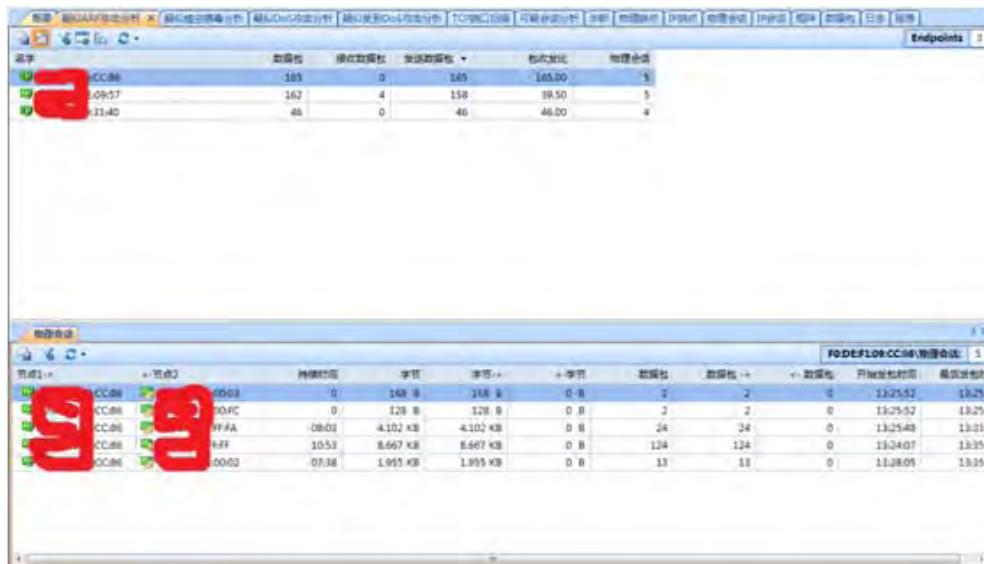
于是在分行出口布置了抓包软件进行抓包，通过邮件将抓下的包发过来，然后使用科来 2010 专家版进行分析。

3. 解决思路：

根据这次问题的特征，观察是否由病毒、攻击造成的影响。是否是应用系统问题，还是网络问题。是否是内网问题，还是外网问题。从简单到复杂，从局部到整体，逐步排除来进行问题的定位。

4. 解决步骤：

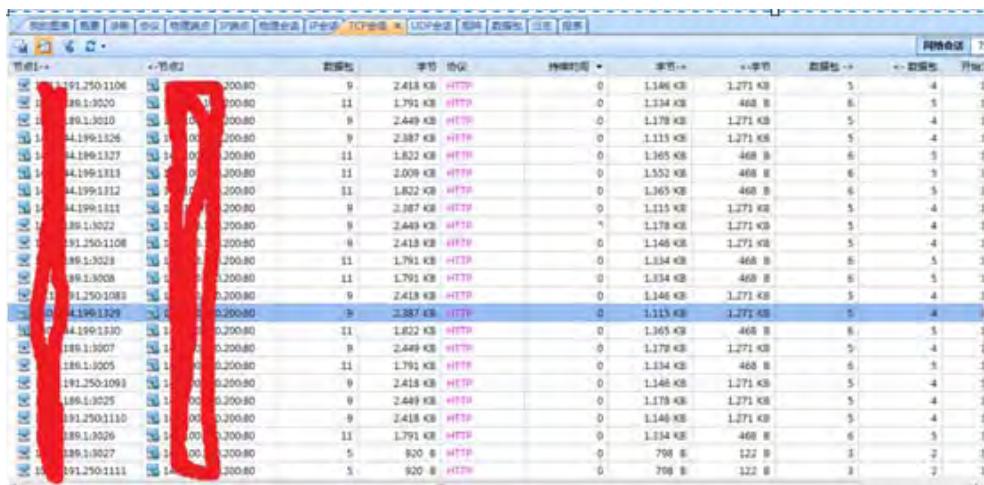
1、首先对安全进行分析。利用科来的安全分析这个分析方案，可以对蠕虫、DOS 攻击、ARP 病毒、TCP 扫描等进行快速定位分析。



通过查找，发现了一些可疑的 ARP 会话，疑似其为 ARP 攻击。经过询问，发现只是一些服务器，并且详细观察数据包，也不符合 ARP 攻击的特征。在其他的一些安全分析中，也没有发现可疑的会话存在。确保在安全问题上，没有影响到本次的故障。

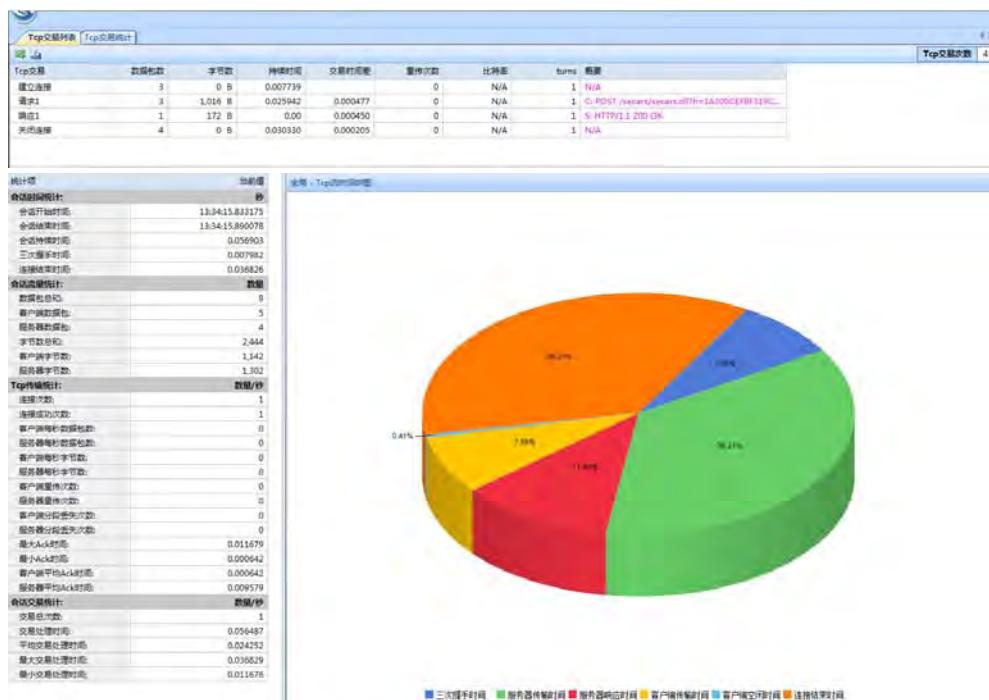
1.2. 分析情况

1. 对应用进行分析



通过对 TCP 会话进行分析，在最新的专家版中，我们可以非常直观地看到整个应用的各个阶段的占用时间。

步骤为：双击该 TCP 会话，选择 TCP 交易统计。可以看到代表三次握手，服务器传输时间，服务器响应时间，客户端传输时间，客户端空闲时间，连接结束时间形成的饼图。

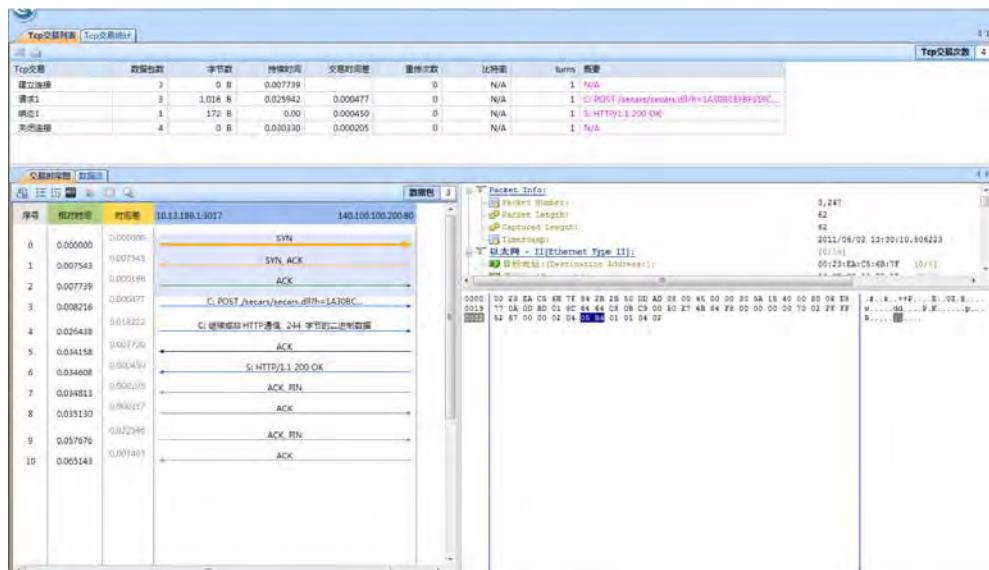


在随机查看了几个 HTTP 会话之后，可以看到服务器端响应时间和客户端空闲时间占据的比例都比较正常，可见在应用程序上应该是没有什么问题的。

2. 查看网络的问题：

点击 TCP 交易列表，可以看到在上面的表格中有建立连接，请求和响应的序列以及关闭连接的相关数据，包括数据包数量，字节数，持续时间，时间差以及重传次数等。

在下面的交易时序图中，可以看到整个会话的过程以及概要，通过对其延迟和丢包重传的分析，可以得出相关的网络情况的结论。



通过随机对数据包的抽查发现，三次握手的时间基本都在十几毫秒到几十毫秒之间没有出现很大的延迟。

通过对重传丢包的分析，可以发现，在网络没有中断的时候，重传为 0，重传出现的时间就是断网的时间，并且都是和外网通信的时候才出现的重传。

The figure consists of three vertically stacked tables from network analysis software. The top table shows a summary of network entities with columns for Name, Physical Address, IP Address, and Quantity. The middle table shows source and destination details for two specific connections. The bottom table shows a list of connections between three specific IP addresses.

名字	物理地址	IP地址	数量
10.12.1.250	00:0C:29:70:B6:43	10.12.1.250	2
10.12.1.7.2	00:29:EA:C5:6B:7F	10.12.1.7.2	2
10.12.1.205	00:22:5A:D5:6B:40	10.12.1.205	6
10.12.1.201.104	00:0C:29:70:B6:40	10.12.1.201.104	6

源IP地址	源物理地址	目标IP地址
10.12.1.250	00:0C:29:70:B6:43	9.90.47.2
10.12.1.250	00:0C:29:70:B6:43	9.90.47.2

源IP地址	源物理地址	目标IP地址
10.12.1.205	00:22:5A:D5:6B:40	10.12.1.201.104
10.12.1.201.104	00:0C:29:70:B6:43	10.12.1.205

1.3. 结论

由此断定，内网通信中并没有问题，在和外网通信的时候才会出现断网的情况，可能是由于运营商提供的链路出现了问题。最后通过将链路进行更换，再进行网页浏览等行为，总行与分行之间的通信也没有中断过。

本次分析充分运用科来强大的分析能力，对整个网络进行透视，从简单到复杂，从局部到整体来进行分析。科来最新版本中，对应用和网络延迟丢包等分析的能力又更上一层楼，更直观表现出应用与网络的问题所在。通过对延迟和响应的分析，快速定位网络与应用的问题，从容排除故障。

2. 税务系统问题分析

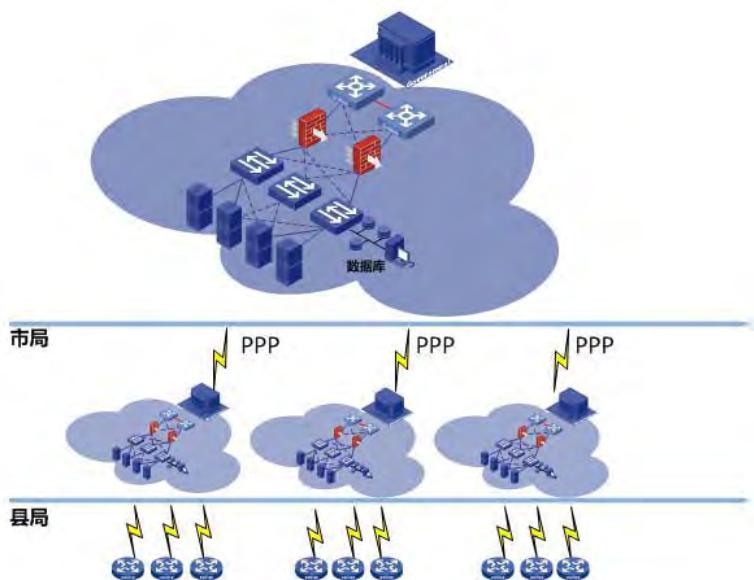
2.1. 故障现象描述

1. 故障现象描述

- 1) 纳税人不能正常购买发票，在购买发票保存时，客户端出现故障“服务器端没有响应提示”或无法显示该项内容。
- 2) 出现问题时间不固定，区域不固定，规律性不强；
- 3) 出现较多的是XX和XX，频率出现较高时期是征缴期。

2. 基本环境描述

用户基本网络拓扑如下：



用户的网络采用内外网隔离方式，基本网络拓扑如上图所示，其中连接地市征缴系统的端口为 100M 带宽。

服务器 IP 地址为 X.X.10.73、X.X.10.74。

2.2. 分析过程

1. 分析目标

- 1) 通过采样分析，了解纳税人购票业务的数据流向及健康情况；
- 2) 确认该故障由于网络原因引起的还是其他原因引起的；

2. 分析方案部署（采用科来软件分布式）

步骤一

此次采样我们到故障现象较为密集的 XXX 地税局，通过科来软件现场抓取网络数据作为捕获地点一；如图我们选择现场的一个（XXXXXX 装饰材料行操作）作为参考，对正在进行的发票购票业务进行数据采集，如下图：

节点1->	<-节点2	数据包	字节	协议	持续时间
192.168.1.26:1459	192.168.1.9:180	127	81.646 KB	HTTP	00:01:27

↓

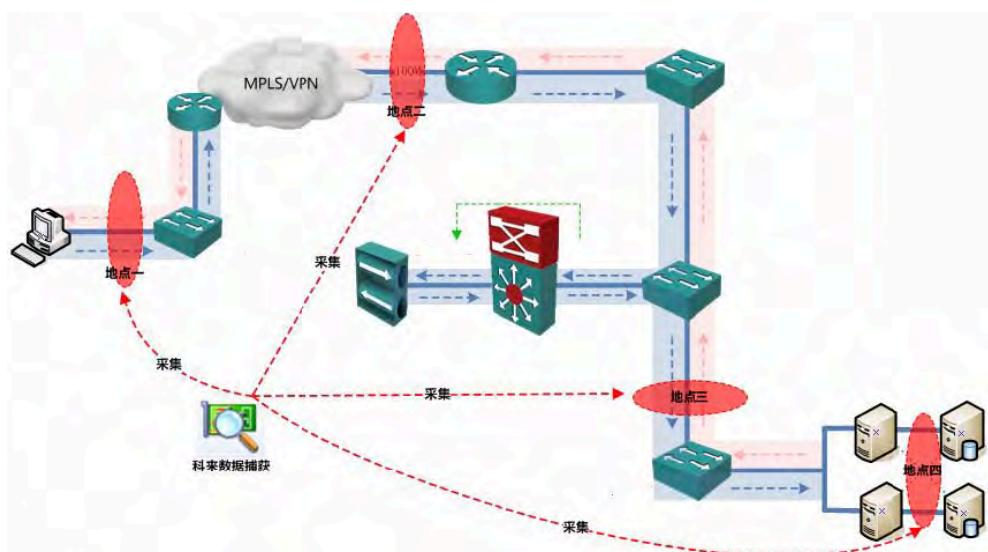
纳税人购票

纳税人识别号: 41020319730916101201
 法定代表人: XXXXXXXXXX
 登记注册类型: 内资个体
 税务登记证号码: 41020319730916101201
 批准日期: 2010-
 经办人证件号码: XXXXXXXXXX
 已申报购票清单

序号	发票名称	发票代码	购买数量	起始号
1	241001030170	50	04034801	
2	文化体育娱乐服务有奖定额发票(10元)	241001000131	50	10707601

步骤二

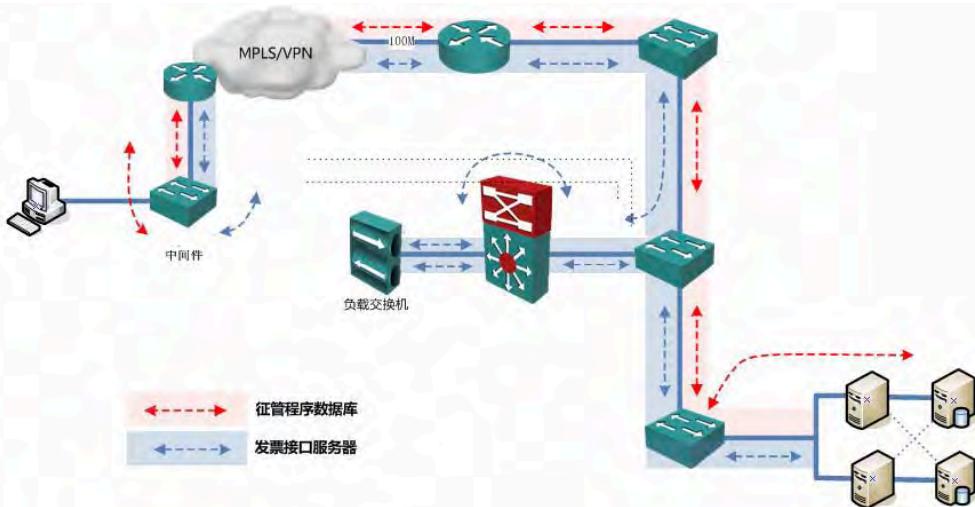
分别在省地税局中心机房路由器与 ATM 接口作为捕获地点二；在交换机与交换机的接口作为捕获地点三；在应用服务器与数据库接口作为捕获地点四；同时采集 XX 地税局（XX 装饰材料行）进出该地点的网络流量，对进出该地点的流量进行分析。整体部署方案下图所示



3. 分析过程

1) 购票业务数据流量分析

利用科来网络分析系统的流量监控分析功能，对网络中采集的数据进行监控分析，确认该业务的数据访问流程。如下图



以当时采集主机 X.X.26.78 (XX 装饰材料行应用) 为例，该主机通过中间件 X.X.9.1 访问省局应用服务器平台 X.X.10.75 与 X.X.10.77;其中 X.X.10.75 与 X.X.10.77 通过三层交换机 8812 转换为 X.X.10.73 与 X.X.10.74 的地址访问该业务系统的业务平台服务器 X.X.10.64 与 X.X.10.63 和数据库进行交互。

4. 性能分析

利用科来网络分析系统的实时网络流量监控分析功能，对网络中采集的流量进行监控分析，确认是否存在网络拥塞导致其他原因。

地点一

流量统计	字节数	数据包数	利用率	每秒位数	每秒包数
总流量	28.016 MB	48,692	2.282%	228 158 Kbps	127
广播流量	26 779 KB	210	0.000%	0 bps	0
多播流量	159 225 KB	470	0.071%	7 072 Kbps	2
平均包长				603.320 字节	

地点二

网络流量	数据包	利用率	每秒位数	平均每秒位数	平均每秒包个数
总共流量	2,452,679	26.899%	26.899 Mbps	15.757 Mbps	4,895.567
发送广播流量	5,230	0.003%	2.560 Kbps	7.015 Kbps	10.439
发送组播流量	3,412	0.004%	3.904 Kbps	10.950 Kbps	6.810

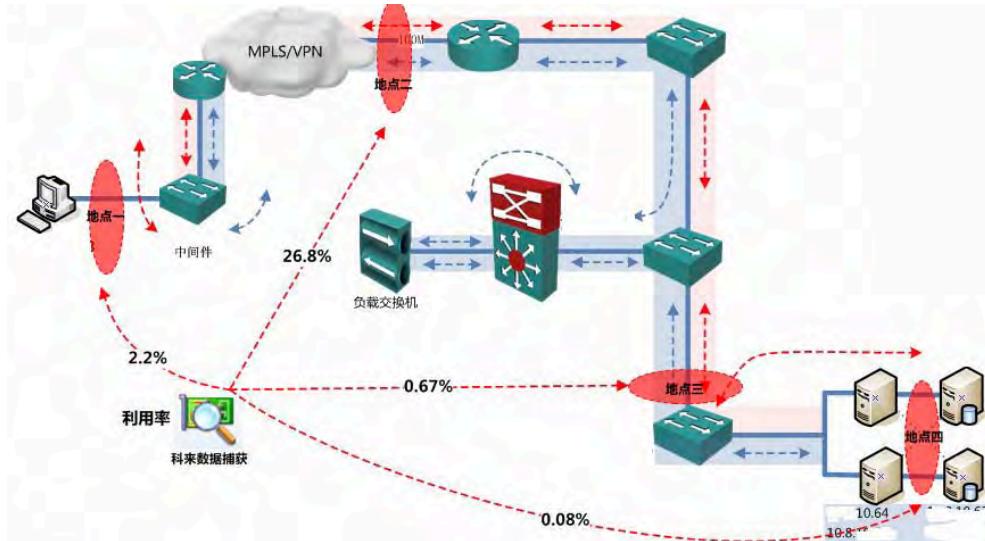
地点三

网络流量	数据包	利用率	每秒位数	平均每秒位数	平均每秒包个数
总共流量	129,997	0.674%	573.953 Kbps	1.057 Mbps	294.778
发送广播流量	2,857	0.004%	4.352 Kbps	4.179 Kbps	6.478
发送组播流量	1,822	0.003%	2.624 Kbps	6.198 Kbps	4.132

地点四

网络流量	数据包	利用率	每秒位数	平均每秒位数	平均每秒包个数
总共流量	255,971	0.084%	338.736 Kbps	833.946 Kbps	271.640
发送广播流量	5,843	0.001%	6.384 Kbps	4.067 Kbps	6.177
发送组播流量	3,757	0.000%	2.400 Kbps	6.012 Kbps	3.971

整体性用率如下图；根据实际采集数据结果所示网内不存在链路拥塞情况；



5. 安全性分析

从此次数据采集情况来看。未发现木马、蠕虫、网络攻击等特征；

6. 异常情况分析

地点一

业务正常处理并打印。

地点二

X.X.248.21 流量正常，无异常。

X.X.17.77 流量正常，无异常

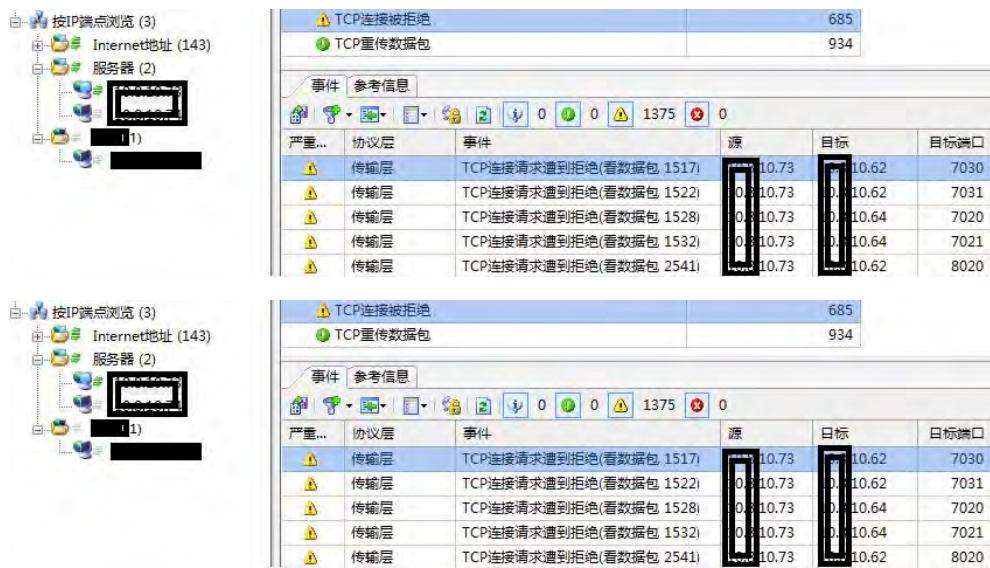
X.X.8.21 流量正常，无异常

地点三

在采集时间 9 分钟 42 秒内出现 X.X.10.73 访问 X.X.10.62、X.X.10.64 的 7030、3031、7020、7021、

8020 端口数据交互时出现大量连接被拒绝情况；

如下图所示



在采集时间 9 分钟 42 秒内出现 X.X.10.74 与 X.X.10.62、X.X.10.64 的 7030、3031、7020、7021、8020 端口数据交互时出现大量连接被拒绝情况；

地点四

数据库与应用平台交换流量正常均未发现异常。

2.3. 分析结论

经过部署科来软件，迅速定位出了故障：

主要出现在 X.X.10.73、X.X.10.74 地址与应用平台 X.X.10.62、X.X.10.64 地址的 7030、3031、7020、7021、8020 端口访问时出现大量连接被拒绝情况；

经过核实 7030、3031、7020、7021、8020 端口为业务和数据库数据交互端口，出现该现象的原因是中间件服务器访问应用平台服务器时出现连接被拒绝，导致应用无法正常运行；

最终通过优化服务端 TCP 通讯连接资源配置使该问题得到解决。

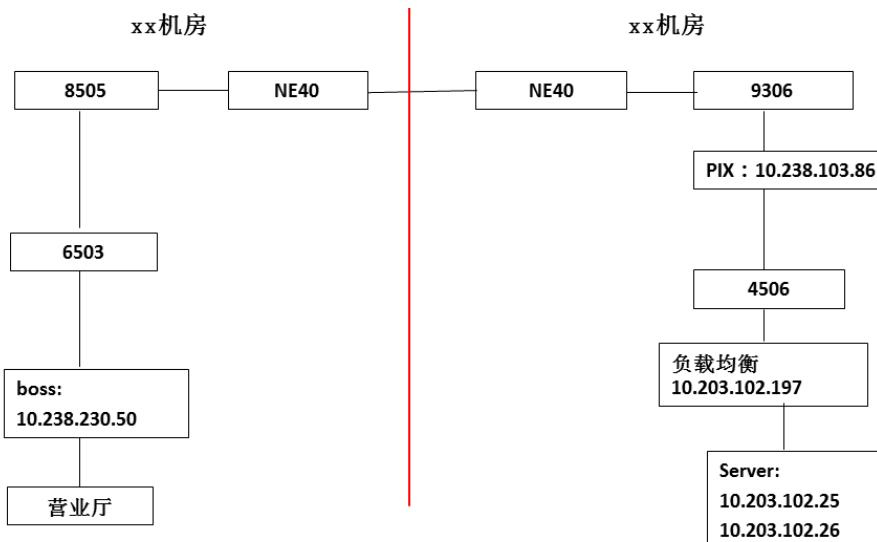
3. 某移动公司BOSS系统故障分析

3.1. 故障描述

1. 故障描述

- 1) Boss系统向服务器提交订单，每天会有 600 个左右不成功的订单，不成功的订单需手工录入，极大的影响工作效率；该现像已持续 2-3 个月；
- 2) 持续ping服务器和boss，未出现任何的丢包现像；
- 3) 应用部门和应用厂商检查应用程序和规则说一切正常；
- 4) 网管人员检查网络设备的性能，设置（MTU、MSS等）一切正常；
- 5) 管理人员说在boss系统上抓取的同步数据包大于在PIX之前抓取的数据包，怀疑有丢包，但其它应用和ping都正常，网络丢包没有说服力。

2. 网络拓扑

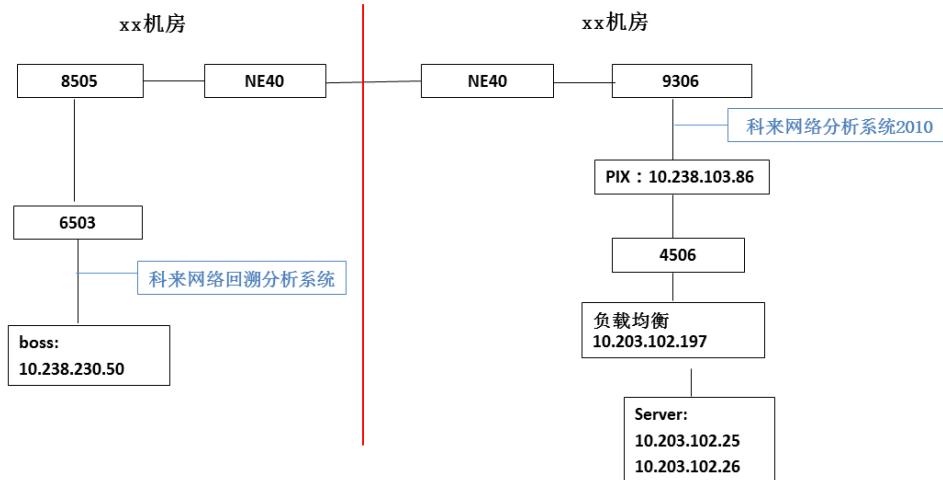


- ◆ Boss系统访问10.238.103.86的80端口；
- ◆ Boss系统收集营业厅的数据传给server，boss在既是客户端又是服务器。
- ◆ 10.238.103.86将80端口映射到10.203.102.197；
- ◆ 将10.203.102.197负载均衡到10.203.102.25和10.203.102.26

3.2. 分析过程

1. 捕获数据包

订单提交不成功则有两种情况，一种是服务端未收到 boss 的请求，另一种则是服务端收到请求后未响应，由于客户说在 boss 和 pix 上抓包不一致，先从这里着手，选择抓包位置，如下图：

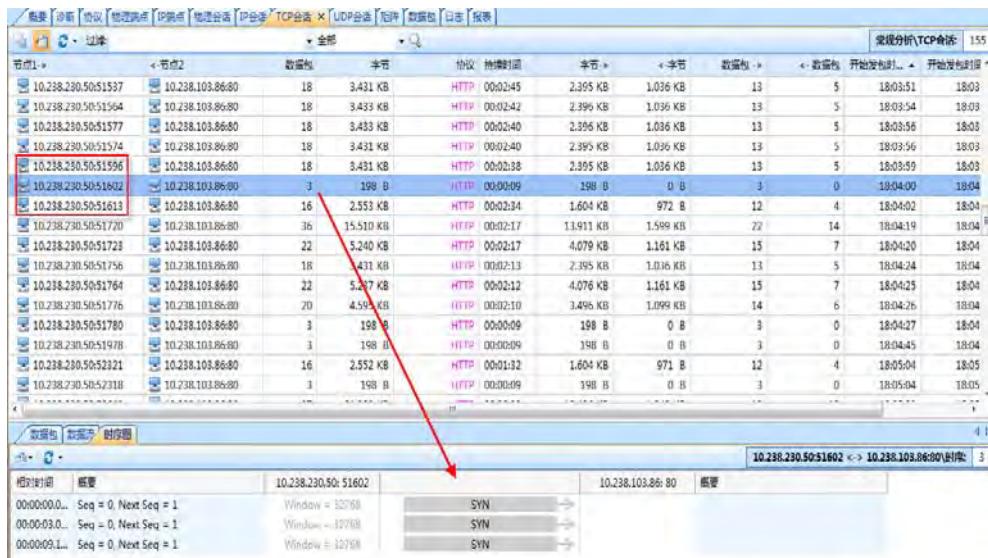


PS：将回溯系统和便携式的时间同步，然后开始抓包；

2. 分析数据包

分别提取回溯系统和便携式上的数据包，进行对比分析。

首先来看在 6503 上抓取的数据，如下图：



在 6503 上捕获到 10.238.230.50 和 10.238.103.86 的会话中，存在多个 syn 包无响应的会话，从而证实确实存在订单提交不成功的问题，而在 PIX 的入口并没有捕获到该会话，也就是说服务端并未收到 boss 的应用请求，所以该现象与服务器端无关。

再来看在 PIX 前抓取的数据，如下图：

节点1->	<-节点2	数据包	半节 协议	持续时间	半节->	<-半节	数据包->	<- 数据包	开始发送时间	耗时
10.238.230.50:51536	10.238.103.86:80	18	3.431 KB HTTP	02:46	2.395 KB	1.036 KB	13	5	18:03:17	
10.238.230.50:51537	10.238.103.86:80	18	3.431 KB HTTP	02:45	2.395 KB	1.036 KB	13	5	18:03:17	
10.238.230.50:51564	10.238.103.86:80	18	3.433 KB HTTP	02:42	2.396 KB	1.036 KB	13	5	18:03:20	
10.238.230.50:51574	10.238.103.86:80	18	3.431 KB HTTP	02:40	2.395 KB	1.036 KB	13	5	18:03:22	
10.238.230.50:51577	10.238.103.86:80	18	3.433 KB HTTP	02:40	2.396 KB	1.036 KB	13	5	18:03:22	
10.238.230.50:51596	10.238.103.86:80	18	3.431 KB HTTP	02:38	2.395 KB	1.036 KB	13	5	18:03:24	
10.238.230.50:51613	10.238.103.86:80	16	2.553 KB HTTP	02:34	1.604 KB	0.972 KB	12	4	18:03:28	
10.238.230.50:51720	10.238.103.86:80	36	15.510 KB HTTP	02:17	13.911 KB	1.599 KB	22	14	18:03:45	
10.238.230.50:51723	10.238.103.86:80	22	5.240 KB HTTP	02:17	4.079 KB	1.161 KB	15	7	18:03:46	
10.238.230.50:51756	10.238.103.86:80	18	3.431 KB HTTP	02:13	2.395 KB	1.036 KB	13	5	18:03:49	
10.238.230.50:51764	10.238.103.86:80	22	5.237 KB HTTP	02:12	4.076 KB	1.161 KB	15	7	18:03:50	
10.238.230.50:51776	10.238.103.86:80	20	4.595 KB HTTP	02:10	3.496 KB	1.009 KB	14	6	18:03:52	
10.238.230.50:52321	10.238.103.86:80	16	2.552 KB HTTP	01:32	1.604 KB	0.971 KB	12	4	18:04:30	
10.238.230.50:52643	10.238.103.86:80	37	21.283 KB HTTP	9	19.436 KB	1.648 KB	19	18	18:04:49	
10.238.230.50:52658	10.238.103.86:80	53	33.239 KB HTTP	8	30.892 KB	2.348 KB	27	26	18:04:50	
10.238.230.50:52673	10.238.103.86:80	19	8.182 KB HTTP	7	6.896 KB	1.285 KB	10	9	18:04:51	
10.238.230.50:52686	10.238.103.86:80	13	4.160 KB HTTP	7	3.063 KB	1.098 KB	7	6	18:04:51	



服务端没有收到包 boss 系统的请求包，是不是由于包被丢弃了呢？从拓扑上看，数据经过的都是路由、交换设备，该包连防火墙都没到，而且该链路上的其它应用一切正常，如前所述，网络丢包没有说服力，继续看数据包，看能不能找到其它线索。

查看“概要”，发现网络中存在大量的 FIN 数据包，4452 个数据包就有 2498 个带包带 FIN 标记：

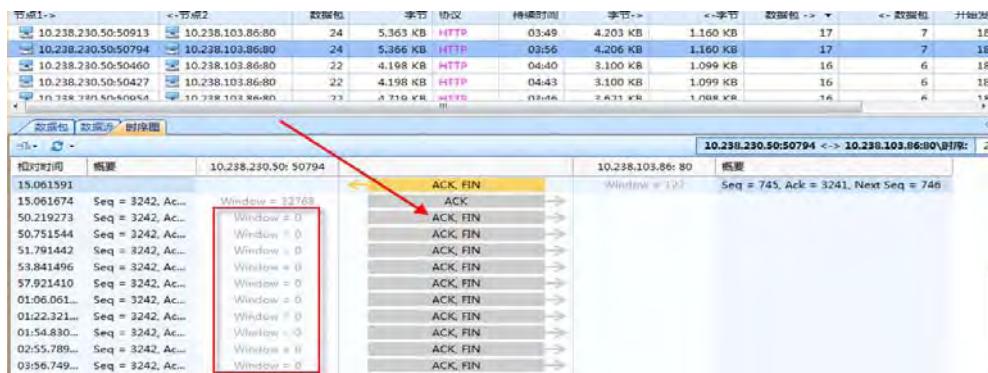
TCP统计	数量
TCP同步发送	202
TCP同步确认发送	182
TCP结束连接发送	2,498
TCP复位发送	0

我的图表	概要	诊断	协议	物理连接	IP会话	物理会话	IP会话	TCP会话	UDP会话	规则	数据包	日期	报告
											4,452		
编号	绝对时间	源	目标										
109	18:01:37.752201	10.238.230.50:50212	10.238.103.86:80										
110	18:01:37.752202	10.238.230.50:50204	10.238.103.86:80										
111	18:01:37.752357	10.238.230.50:50196	10.238.103.86:80										
112	18:01:37.752358	10.238.230.50:50191	10.238.103.86:80										
113	18:01:37.752360	10.238.230.50:50145	10.238.103.86:80										
114	18:01:37.752362	10.238.230.50:50140	10.238.103.86:80										
115	18:01:37.752363	10.238.230.50:50135	10.238.103.86:80										
116	18:01:37.752518	10.238.230.50:50132	10.238.103.86:80										
117	18:01:37.752519	10.238.230.50:50125	10.238.103.86:80										
118	18:01:37.752523	10.238.230.50:50121	10.238.103.86:80										
● 检查点 (Acknowledgment number):													
○ 检查点 (Push Function):													
○ 重置 (Reset the connection):													
○ 同步 (Synchonization sequence):													
○ 同步和确认 (Sync and Ack):													
○ 窗口 (Window):													
○ 检验和 (Checksum):													
○ 序列指针 (Segment point):													
○ TCP选项 (TCP Option):													
○ Extra Data:													
○ Number of Bytes:													

过滤 FIN 数据包，发现绝大部分 FIN 数据包都是由 boss 服务器发出来的：

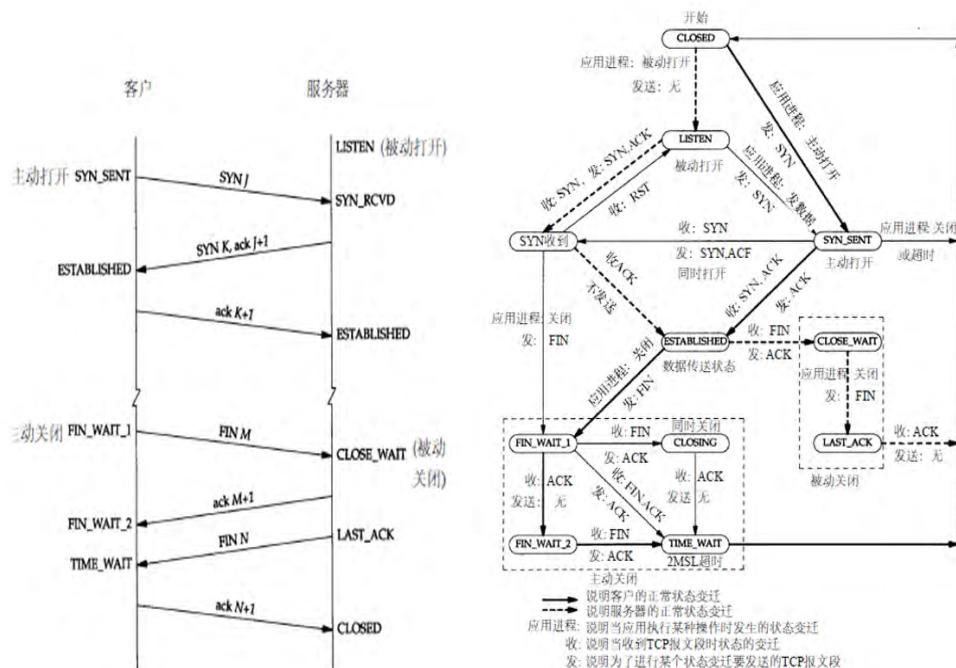
IP会话	TCP会话	UDP会话	规则
			1
10.0.0.8 (1)	10.238.230.50	10.238.103.86	189
			0
			202
			0
			182
			2,317
			181
			0
			0

再定位到 TCP 会话，通过时序图查看，查看会话中 FIN 包的情况：



通过时序图可以看到在一个会话中存在多个 FIN 位置 1 的数据包，而且没收到对端的确认，这表示该会话没有正常关闭。

网络中存在大量的在一个会话中发送大量 FIN+ACK 置 1 且 window 为 0 的数据包的情况（我们知道，在数据传输过程窗口为 0 表示不能接受任何数据，至于关闭连接的 window 为 0 是否表示不能接收任何数据包有待验证，但可以肯定的是不正常的），且这些会话都与 10.238.230.50 有关，这就表示在 10.238.230.50 上有很多未关闭的 TCP 会话，这是不正常的，需要进一步分析原因。先插入一些 TCP 连接建立和关闭的介绍（详细内容请参考《TCP/IP 卷 1》第 18 章）



(TCP 正常建立和关闭对应的状态和状态变迁)

简单的说，在通讯过程中，客户端和服务端的 TCP 状态迁移如下：

客户端 TCP 状态迁移: CLOSED->SYN_SENT->ESTABLISHED->FIN_WAIT_1->FIN_WAIT_2->TIME_WAIT->CLOSED

服务器 TCP 状态迁移: CLOSED->LISTEN->SYN

收到->ESTABLISHED->CLOSE_WAIT->LAST_ACK->CLOSED 登录 10.238.230.50 后台，通过 netstat 查看主机的会话状态：

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	state
tcp	0	0	10.238.230.50.64329	10.238.159.90.9055	CLOSE_WAIT
tcp	0	0	10.238.230.50.65223	10.238.159.90.9055	CLOSE_WAIT
tcp	0	0	10.238.230.50.59545	10.238.159.90.9055	CLOSE_WAIT
tcp	0	1	10.238.230.50.52077	10.238.103.86.80	LAST_ACK
tcp	0	0	10.238.230.50.54172	10.238.159.90.9055	CLOSE_WAIT
tcp	0	1	10.238.230.50.53519	10.238.103.86.80	LAST_ACK
tcp	0	0	10.238.230.50.64251	10.238.159.90.9055	CLOSE_WAIT
tcp	0	1	10.238.230.50.50295	172.16.11.15.8888	LAST_ACK
tcp	0	0	10.238.230.50.60502	172.16.11.15.8888	CLOSE_WAIT
tcp	0	0	10.238.230.50.56242	10.238.159.90.9055	CLOSE_WAIT
tcp	0	1	10.238.230.50.52658	172.16.11.15.8888	FIN_WAIT_1
tcp	0	0	10.238.230.50.59626	10.238.159.90.9055	CLOSE_WAIT
tcp	0	1	10.238.230.50.49929	10.238.103.86.80	LAST_ACK
tcp	0	0	10.238.230.50.64620	10.238.159.90.9055	CLOSE_WAIT
tcp	0	0	10.238.230.50.64077	10.254.126.227.5555	CLOSE_WAIT
tcp	0	0	10.238.230.50.52529	10.254.126.227.5555	CLOSE_WAIT
tcp	0	0	10.238.230.50.54095	10.254.126.227.5555	ESTABLISHED
tcp	0	0	10.238.230.50.55723	10.238.159.90.9055	CLOSE_WAIT
tcp	0	0	10.238.230.50.60608	10.238.159.90.9055	CLOSE_WAIT
tcp	0	0	10.238.230.50.63782	172.16.11.15.8888	CLOSE_WAIT
tcp	0	0	10.238.230.50.49746	10.238.159.90.9055	CLOSE_WAIT
tcp	0	0	10.238.230.50.63481	10.254.126.227.5555	CLOSE_WAIT
tcp	0	0	10.238.230.50.51695	10.254.126.227.5555	CLOSE_WAIT
tcp	0	0	10.238.230.50.49217	172.16.11.15.8888	CLOSE_WAIT
tcp	0	0	10.238.230.50.60260	172.16.11.15.8888	CLOSE_WAIT
tcp	0	0	10.238.230.50.55975	10.254.126.227.5555	CLOSE_WAIT
tcp	0	0	10.238.230.50.53937	172.16.11.15.8888	CLOSE_WAIT
tcp	0	0	10.238.230.50.59979	10.254.126.227.5555	CLOSE_WAIT
tcp	0	0	10.238.230.50.55744	10.254.126.227.5555	CLOSE_WAIT
tcp	0	1	10.238.230.50.51173	10.238.103.86.80	LAST_ACK
tcp	0	0	10.238.230.50.64532	10.238.159.90.9055	CLOSE_WAIT
tcp	0	0	10.238.230.50.54654	10.238.159.90.9055	CLOSE_WAIT
tcp	0	1	10.238.230.50.52183	172.16.11.15.8888	LAST_ACK
tcp	0	0	10.238.230.50.61379	10.254.126.227.5555	CLOSE_WAIT
tcp	0	0	10.238.230.50.52814	10.254.126.227.5555	CLOSE_WAIT

如上图，10.238.230.50 上存在近 5000 个状态为 colse_wait 的连接，会话处于 Colse_wait 表示该连接还没有发 FIN+ACK 数据包。通常情况下，一个 colse_wait 会维持至少 2 个小时的时间，这样，随着时间的增加就会导致不能释放的会话越来越多，直到系统没有资源处理新的连接请求。

3.3. 结论及建议

1. 分析结论

10.238.230.50 上存在大量未释放的 TCP 连接，我们知道，TCP 是有队列限制的，当队列已满时，TCP 将不会处理传入的 SYN，也不会发 RST 应答，因为通常队列已满是由应用程序和操作系统忙造成的原因（详见 TCP/IP 卷 1 第 18 章），这也能够解释为什么服务端没有收到 boss 的 SYN 包了，实际上这些数据包是 boss 系统收到的营业厅的 SYN 包，但由于 boss 系统队列已满或繁忙，则对其不做处理。

10.238.230.50 在与 server 关闭连接的过程中，window 为 0，可能是系统忙于处理 colse_wait 会话所致，从而导致 boss 与 server 的通讯异常。

订单提交不成功的原因是 boss 系统队列已满或繁忙，没有资源对连接请求进行处理，问题出在 boss 系统。建议：检查 10.238.230.50 与 10.254.126.227 和 10.238.159.90 的应用通讯和应用程序（因为 colse_wait 的会话大都数与这两个 IP 有关，而 10.238.230.50 与 server 的连接状态是正常的）；修改 tcp_keepalive_* 的相关参数。

4. 某用户网络问题分析报告

4.1. 故障现象描述

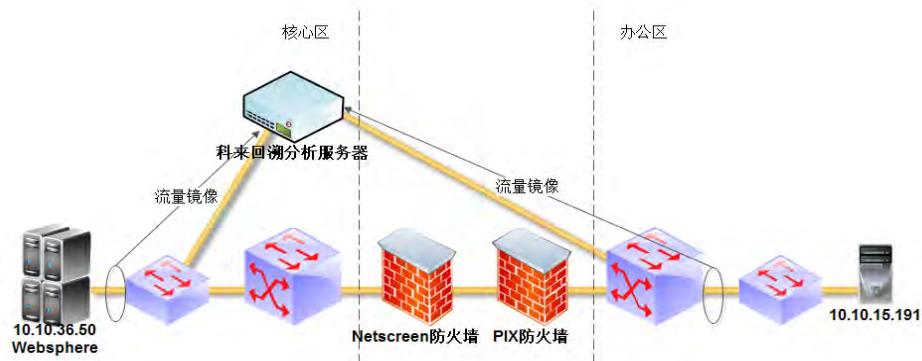
1. 故障现象描述

某公司总部业务内网 IP 电话系统中，一台位于办公区的 IP 电话管理系统主机（vSphere 虚拟机）10.10.15.191 需要定期与位于核心区的一台服务器（也是 vSphere 虚拟机）10.10.36.50 通信，传递 IP 电话状态信息。

10.10.36.50 是一台 WebSphere 应用服务器，在应用服务器的日志中不定期会出现 10.10.15.191 客户端无响应导致会话超时的错误警报。

2. 环境描述

发生问题的两台主机之间的网络逻辑结构示意图如下：



发生问题的客户机与服务器的通信需要经过两道防火墙以及多台网络设备，两台防火墙均未配置内网间 NAT 地址翻译。

4.2. 分析方案设计

1. 分析目标

鉴于发生问题的两台主机间网络设备较多，初步怀疑是防火墙故障阻断了两台主机间的数据传输导致会话超时。需要通过数据包解码分析验证是否中间设备故障导致，找出问题的根源。

2. 分析方法

3. 将科来回溯分析服务器部署在核心区，同时连接服务器接入交换机与办公网汇聚交换机，将服务器接入端口与办公网上行端口的流量镜像到分析服务器。

利用科来回溯分析系统 **7*24** 小时不间断捕获防火墙两端的流量，根据服务器日志产生故障警报的时间回溯当时两台主机间的通信数据包。通过两端流量分析对比，判断防火墙以及中间网络设备是否对两台主机的通信造成影响；如果中间设备没有对会话造成影响，则进一步分析定位造成故障的直接原因。

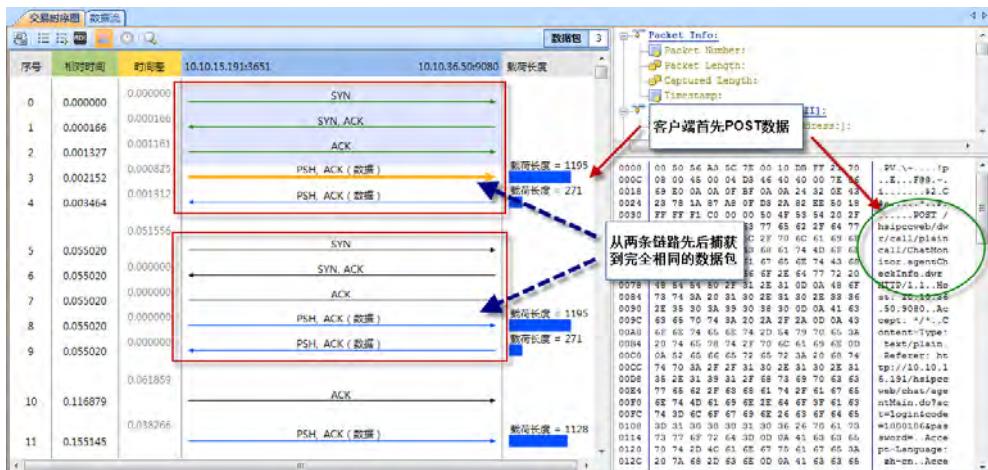
4.3. 分析情况

1. 正常会话行为分析

首先需要对未发生问题时段的正常会话进行解码分析，以建立两台主机间通信的行为模型。下载正常时段 10.10.15.191 与 10.10.36.50 之间 IP 会话的数据包，在科来网络分析模块的 TCP 视图中可以看到两台主机间的会话使用 10.10.36.50 的 TCP 9080 服务端口，会话持续时间、通信数据包数量都不固定，如下图。

节点1->	<-节点2	数据包	字节	协议	持续时间	字节->	<-字节	数据包->	<- 数据包	开始发包...	▲	最后发包时间
10.10.15.191:3644	<-10.10.36.50:9080	536	315,799 KB	TCP - Other	29	250,949 KB	64,840 KB	330	206	14:46:11	14:46:40	
10.10.15.191:3645	<-10.10.36.50:9080	498	284,918 KB	TCP - Other	01:11	226,555 KB	58,363 KB	312	186	14:46:40	14:47:52	
10.10.15.191:3646	<-10.10.36.50:9080	558	320,748 KB	TCP - Other	34	246,906 KB	73,842 KB	346	212	14:46:59	14:47:34	
10.10.15.191:3647	<-10.10.36.50:9080	550	319,287 KB	TCP - Other	41	248,318 KB	70,969 KB	338	212	14:47:34	14:48:15	
10.10.15.191:3648	<-10.10.36.50:9080	544	316,836 KB	TCP - Other	01:26	251,908 KB	64,928 KB	334	210	14:47:52	14:49:18	
10.10.15.191:3649	<-10.10.36.50:9080	522	314,893 KB	TCP - Other	29	250,051 KB	64,842 KB	316	206	14:48:16	14:48:46	
10.10.15.191:3650	<-10.10.36.50:9080	332	197,367 KB	TCP - Other	01:04	156,439 KB	40,928 KB	198	134	14:48:48	14:49:52	
10.10.15.191:3651	<-10.10.36.50:9080	526	315,143 KB	TCP - Other	28	250,303 KB	64,840 KB	320	206	14:49:18	14:49:47	
10.10.15.191:3652	<-10.10.36.50:9080	300	175,717 KB	TCP - Other	02:04	139,299 KB	36,418 KB	180	120	14:49:47	14:51:51	
10.10.15.191:3653	<-10.10.36.50:9080	522	313,717 KB	TCP - Other	01:32	248,916 KB	64,801 KB	314	208	14:49:52	14:51:24	
10.10.15.191:3654	<-10.10.36.50:9080	538	315,852 KB	TCP - Other	36	251,010 KB	64,842 KB	332	206	14:50:16	14:50:52	
10.10.15.191:3655	<-10.10.36.50:9080	550	319,271 KB	TCP - Other	53	249,518 KB	69,754 KB	340	210	14:50:52	14:51:46	
10.10.15.191:3656	<-10.10.36.50:9080	374	219,637 KB	TCP - Other	01:04	174,199 KB	45,438 KB	226	148	14:51:24	14:52:29	
10.10.15.191:3657	<-10.10.36.50:9080	560	316,457 KB	TCP - Other	34	251,820 KB	64,637 KB	352	208	14:51:46	14:52:21	
10.10.15.191:3658	<-10.10.36.50:9080	56	26,141 KB	TCP - Other	37	20,607 KB	5,533 KB	34	22	14:51:51	14:52:29	
10.10.15.191:3659	<-10.10.36.50:9080	542	313,919 KB	TCP - Other	57	250,260 KB	64,805 KB	334	208	14:52:22	14:53:19	
10.10.15.191:3661	<-10.10.36.50:9080	546	316,393 KB	TCP - Other	51	251,428 KB	64,965 KB	338	208	14:52:29	14:53:21	
10.10.15.191:3660	<-10.10.36.50:9080	26	7,377 KB	TCP - Other	01:25	5,711 KB	1,666 KB	16	10	14:52:29	14:53:54	
10.10.15.191:3662	<-10.10.36.50:9080	550	314,500 KB	TCP - Other	02:33	249,725 KB	64,775 KB	342	208	14:53:19	14:55:52	
10.10.15.191:3663	<-10.10.36.50:9080	538	314,605 KB	TCP - Other	39	249,805 KB	64,801 KB	330	208	14:53:21	14:54:00	

从其中一个会话的交易时序图中，可以看到在正常情况下 TCP 三次握手之后，客户端（10.10.15.191）会首先向服务器端（10.10.36.50） POST 数据，如下图。



从图中还可以看出，位于防火墙两端的监控链路先后都捕获到了会话双方向完全相同的数据包，说明

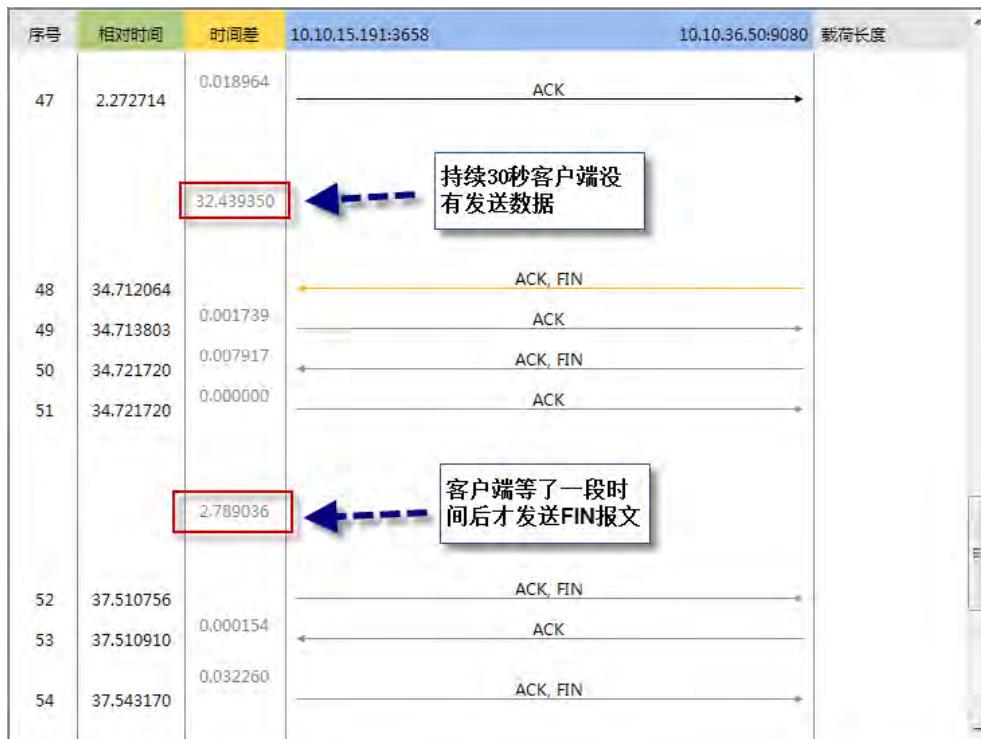
至少在正常时段防火墙并未阻断两台主机的通信。

在会话结束阶段，都是由服务器发送第一个 FIN 报文，客户机应答后向服务器发送 FIN 报文关闭会话，是标准的 TCP 四次握手关闭会话的过程，如下图。



整个会话关闭过程时间非常短，客户机在收到服务器的 FIN 报文后立刻就发送了应答和 FIN 报文。结束阶段的报文我们也是捕获了两次，说明中间设备也未对会话关闭造成影响。

不过在正常时段，我们也发现有个别会话在传输一些数据后，客户端会停止发送数据，服务端在等待 30s 后关闭会话的情况下；而这种情况下，客户机在接收的 FIN 报文并作出 ACK 应答后会等待一段时间再发送 FIN 报文关闭会话。



服务器发送 FIN 报文并接收到客户机的 ACK 之后，如果没有立刻接收到客户端发送的 FIN 报文，则服务器的会话会处在半关闭状态（FIN-WAIT2 状态），如果半关闭状态时间过长超过了服务器的 `tcp-fin-timeout`，服务器就会强行终止会话并在日志中记录会话超时（`tcp-fin-timeout` 因操作系统而异，一般不超过 180 秒）。在上图的会话中客户端 2 秒之后发送的 FIN 报文，因此会话正常关闭，服务器没有会话超时记录。

在上述停顿的时间内，两条监控链路都没有捕获客户端的数据，说明确实是客户端没有发送数据，并非中间设备阻断了客户端数据传输。通常情况客户机如果还有数据需要发送才会暂缓发送 FIN 报文，但从上面的情况看客户机并未再发送任何应用数据，因此怀疑停顿是由客户端应用系统导致的。

2. 异常时段会话行为分析

通过服务日志记录我们调取 12 月 27 日凌晨 3 点左右发送问题时段的数据包，从中发现了明显有问题的会话，如下图。

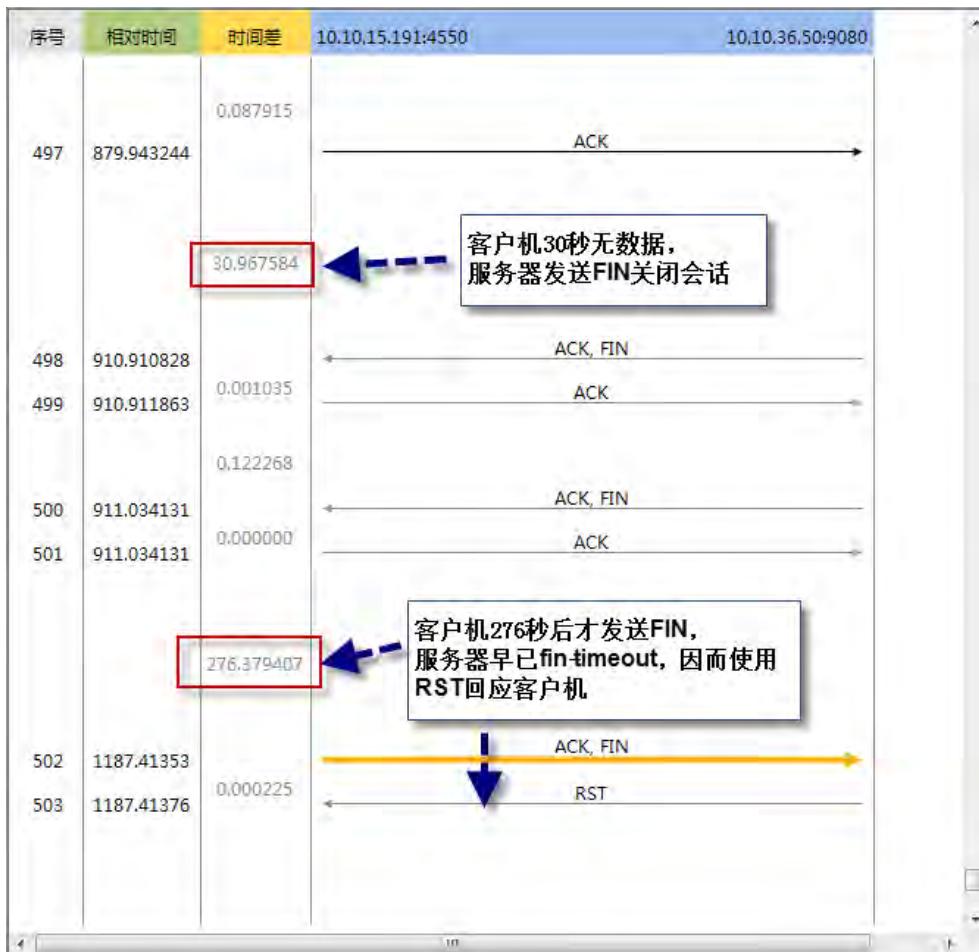
序号	相对时间	时间差	10.10.15.191:4547	10.10.36.50:9080	载荷长度
0	0.000000	0.000000		SYN	
1	0.000178	0.000178	SYN, ACK		
2	0.001983	0.001983	ACK		
3	0.012267	0.010284		SYN	
4	0.012267	0.000000	SYN, ACK		
5	0.012267	0.000000	ACK		
124.297457			<div style="border: 1px solid black; padding: 5px; display: inline-block;"> 会话建立后，客户端 没有发送任何数据 </div>  <div style="border: 1px solid red; padding: 2px; display: inline-block;"> 124.297457 </div> <div style="border: 1px solid red; padding: 2px; display: inline-block;"> 载荷长度 = 149 </div>		
6	124.309724	0.000001	PSH, ACK (数据)		载荷长度 = 149
7	124.309725	0.000001	ACK, FIN		
8	124.309909	0.000184	ACK		
0.091937					
9	124.401846	0.000000	PSH, ACK (数据)		载荷长度 = 149
10	124.401846	0.000000	ACK, FIN		

在三次握手建立 TCP 连接之后，客户机没有发送任何数据，服务器在等待 124 秒之后向客户机发送“Request Timeout”，并立即发送 FIN 报文请求关闭会话，这正是服务器日志记录会话超时的时间。

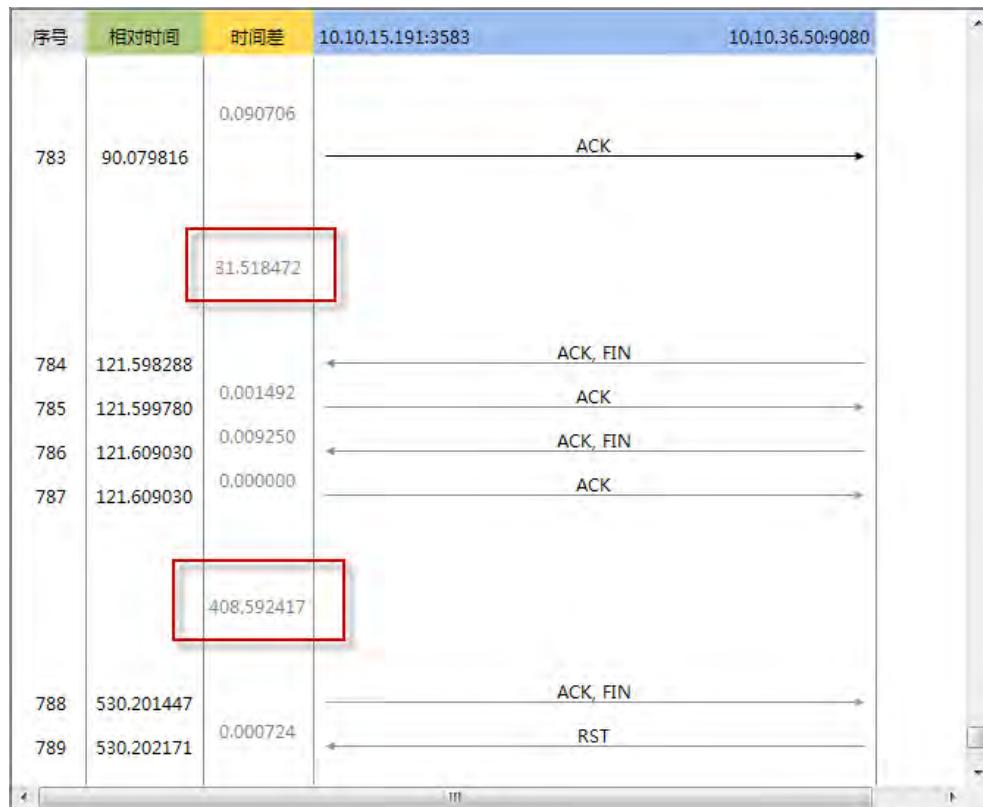
从这个会话中还能够看出客户机在接收到服务器的 FIN 报文后，立刻回应了 ACK 确认，而且两条监控链路都捕获了相关报文，说明在此时刻中间网络通路没有问题并且客户机操作系统的 TCP 协议进程也没有问题；但是客户机回应服务器的 FIN 报文后并没有立即发送 FIN 报文关闭会话，而是在 176 秒后才发送 RST 报文终止会话，在此期间客户机也没有再使用这个会话发送任何数据（如下图），因此可以判断客户机的应用程序对这个会话的处理出现了问题。

序号	相对时间	时间差	10.10.15.191:4547	10.10.36.50:9080	载荷长度
7	124.309725	0.000001	ACK, FIN		
8	124.309909	0.000184	ACK		
0.091937					
9	124.401846	0.000000	PSH, ACK (数据)		载荷长度 = 149
10	124.401846	0.000000	ACK, FIN		
11	124.401846	0.000000	ACK		
176.127853			<div style="border: 1px solid black; padding: 5px; display: inline-block;"> 客户机对服务器的FIN报文做出回应后 没有立即发送FIN结束会话 </div>  <div style="border: 1px solid red; padding: 2px; display: inline-block;"> 176.127853 </div> <div style="border: 1px solid red; padding: 2px; display: inline-block;"> 载荷长度 = 149 </div>		
12	300.529699	0.001394	ACK, RST		
13	300.531093	0.000000	ACK, RST		

除了这个客户机完全没有发送应用数据的会话外，27 日凌晨 3 点左右还有一些会话出现了客户机发送了一些数据后不再继续 POST 数据，导致服务器在 30 秒后发送 FIN 报文关闭会话，而客户机很长时间后向服务器发送 FIN 报文导致服务器出现 tcp-fin-timeout 超时情况，如下图。



服务器出现 tcp-fin-timeout 后也会在日志中记录客户端无响应会话超时的警报。在其他服务器日志出现警报记录的时间我们也发现了类似的情况，如下图 26 日 14:40 分左右的会话。



4.4. 分析结论

通过上述数据包解码分析，我们可以得出以下结论：

首先，排除了网络传输和防火墙导致服务器产生错误警报的可能性，在两条监控链路上捕获到 10.10.15.191 与 10.10.36.50 的会话数据包完全一致，说明中间链路没有阻断两台主机的通信。

网络传输中没有出现重传，三次握手时间很短，说明网络中没有丢包，网络时延也很短，网络服务质量也没问题。

客户机操作系统在接收到服务器的报文后（包括 FIN 报文）很快以 ACK 报文应答，说明客户机操作系统的 TCP 进程没有问题。

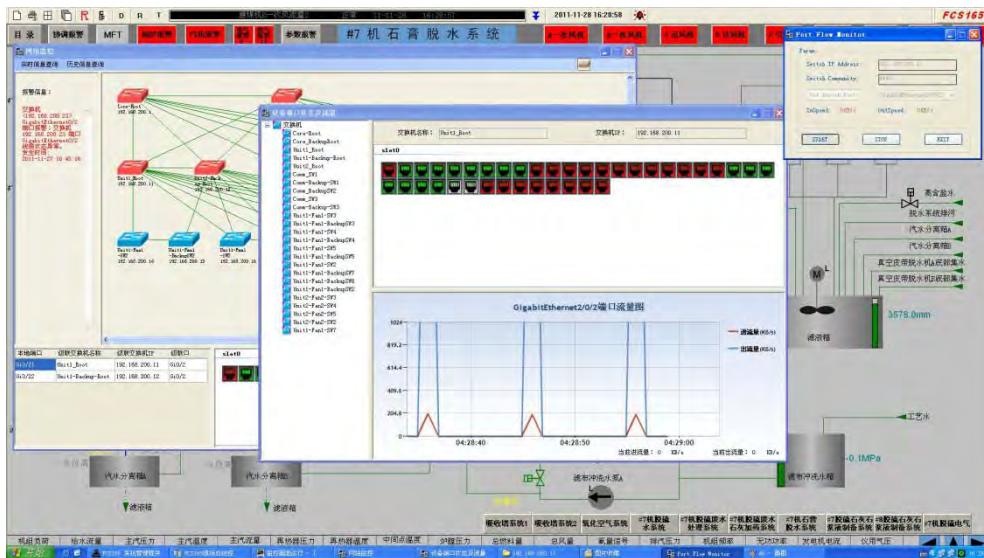
客户机的应用程序会不定期的出现停止传送数据的情况，有时会在接收到服务器关闭会话的 FIN 报文后很长时间才发送客户端 FIN 报文，这是导致服务器出现会话中断警报的主要原因。这很可能是客户机应用程序会话处理功能模块的问题导致，建议与应用系统厂家联系协调优化客户端应用程序。

5. 抓包验证IOS BUG的案例

5.1. 故障描述

1. 问题描述

某电厂生产系统使用 cisco3750 堆叠交换机作为控制数据的转发交换机。在进行业务试运行时发现堆叠交换机的 member 上联到 cisco4500 的端口出现数据间歇性转发的问题：从网管系统上看到该端口每 10 秒集中转发一次数据，而 10 秒空闲内没有任何数据包转发，10 秒后将 10 秒内累积的所有数据包一次性转发，形成了明显的波状数据流。



网管人员登录到该交换机，查看该端口后发现该端口的进出流量数据包统计确实是每 10 秒钟统计一次。而电厂网管人员反映这种情况导致了其中一些处理交互数据的程控机的死机，导致一些业务的运行出现问题，必须尽快解决。

2. 问题思考

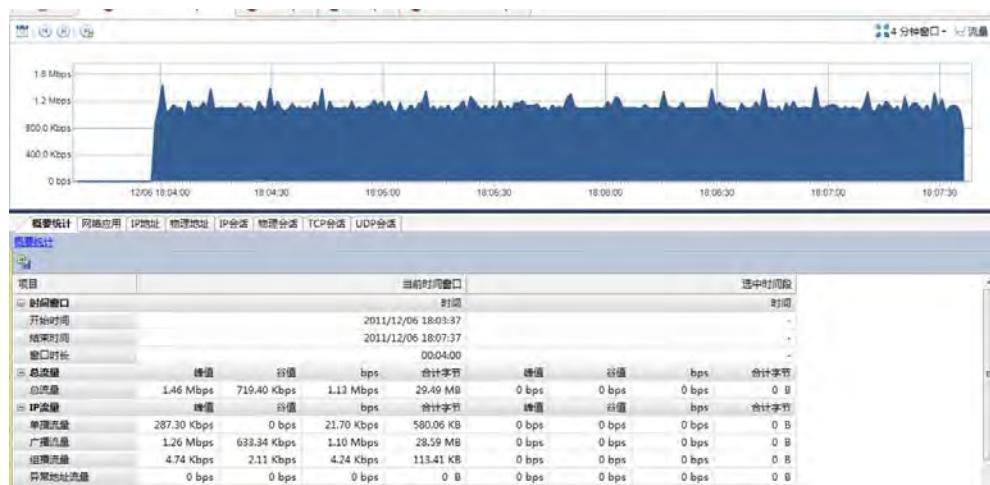
电厂网管人员认为数据处理的程控机死机是因为这种间歇性转发造成。10 秒的流量积累下来，进行短时间内的瞬时转发，使得程控机无法处理大量的数据而造成死机。而管理人员确认，他们的数据交换应该是平滑的，每秒都会有数据，而且每秒钟的数据都相差不大，不会出现这种 10 秒的波峰现象。那么根据反映我们可以初步判断这个可能是一个 IOS BUG。那么究竟是不是 BUG 呢，我们可以使用抓包来验证。看下数据包的流出和流入情况。

3. 抓包验证

12/6 日下午，在该电厂生产网络针对 3750 堆叠问题进行抓包分析。镜像 3750 member 的上联到 cisco4500 的端口。镜像采用全镜像，RX,TX 三种方式进行抓包。

首先在转发正常的 master 进行抓包，看其通信数据转发情况。抓包 5 分钟后，我们发现 master 的转

发比较平滑，没有出现转发的中断和流量突发，如图为科来回溯分析系统看到的流量趋势图：

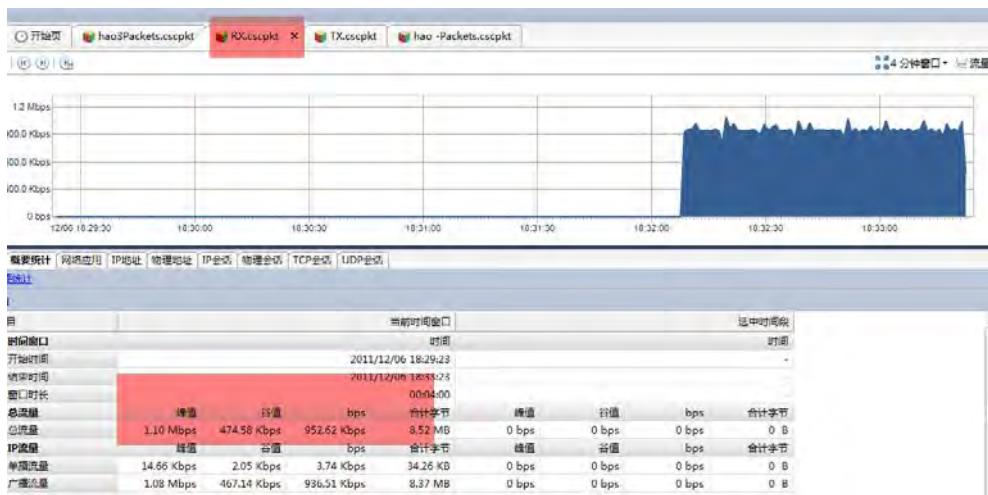


在转发出现异常的 member 上 镜像 member 上联到 cisco4500 端口 采集 both 流量。此时通过网管软件看到的该端口的转发是间歇性的转发，10 秒钟一个峰值，而其他时间为 0。通过登录到交换机查看端口的 out, in 的流量统计也验证了网管软件看到的情况。因此该端口的转发情况存在明显的异常。

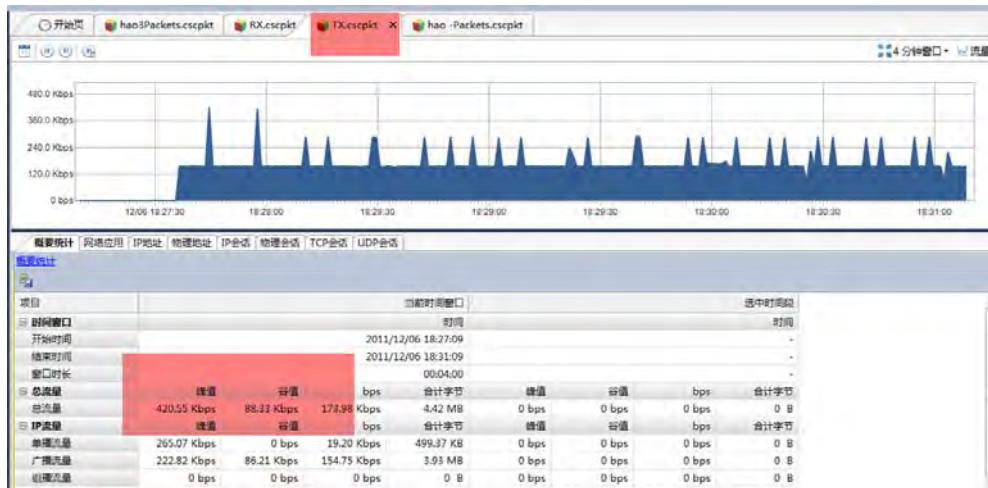
但通过抓包分析我们看到该端口的流量是平滑的，并未有流量的波动。如图可以看到回溯分析的流量趋势图：



此后又再次镜像该端口的 RX, TX 流量。可以看到数据依然是平滑的。



(RX 流量镜像趋势图)



(TX 流量趋势图)

TX 流量出现一些流量的峰值，经过时间对比，和网管人员确认，是正常现象。因为 TX 流量基线比较小，正常只有 120Kbps，而峰值出现也没超过 400kbps。而且并未出现流量为 0 的情况。

综上所述 我们看到无论采用哪种抓包方式，都可以看到流量是平滑的，没有出现过流量的波动较大的情况，而且没有出现流量转发中断的情况。与网管软件和 SW 端口看到的情况完全不一样。出现这种情况我们可以判断这种现象应该是一种 bug。

交换机镜像技术本身是一门很底层的技术，在芯片级别实现的。在流量进出 IOS 处理之前就已经完成了流量的镜像。因此通过镜像我们可以看到流量在进出该端口的底层数据上是平滑的，与交换机的统计完全不一致，那么这种情况肯定是 IOS 的 bug。

此外，本次抓包的流量本身比较怪异，流量没有单播数据。流量的转发全部为广播包，这种现象在网络中比较异常。经过询问得知，控制系统在设计的时候就采用的是全网广播的形式进行数据的交互，因此大量的广播包很正常。

5.2. 事件处理

通过抓包验证了该版本的 IOS 的系统在处理全网广播的数据的时候会出现 bug。后经过与 cisco 沟通，提交了抓包数据和现场情况描述，cisco 确认这是一个 IOS 的计数 bug，即端口的统计数据包不是实时统计，而是 10 秒统计一次，而网管系统读取的是交换机的端口信息，那么同样会出现每 10 秒一次的波形图。

通过抓包了解到数据的转发是正常的，是平滑的，没有出现网管系统中的波形图。随后排除了流量过载影响程控机死机的可能性。至于程控机死机现象由什么引起，还需要网管人员进行其他排查方法。

6. 网络故障分析报告

6.1. 前言

现在的网络要比以前复杂许多，在各行各业中，不断的有新的网络应用加入，这对网络性能要求是非常高的。网络性能评估对网络关键应用能否健康运行有重要意义，通过对网络核心设备的处理能力分析，对网络带宽利用率、网络负载的分析，有助于提高网络整体性能和资源的合理分配，为规划、调整网络提供可靠依据。

科来网络分析系统是非常好的流量分析系统，利用他我们可以实际了解当前网络正在发生的具体流量，并且通过科来网络分析系统的专家系统及进一步对数据包的解码分析，我们可以很快的定位网络故障，确认网络带宽的瓶颈，在故障发生前消除网络隐患，这样能给我们日常的网络维护工作带来很大的方便，也是我们的维护工作处于主动地位，不会再只用接到用户故障投诉后处理故障，这在时间和效率上都有了很大提高。

6.2. 故障描述

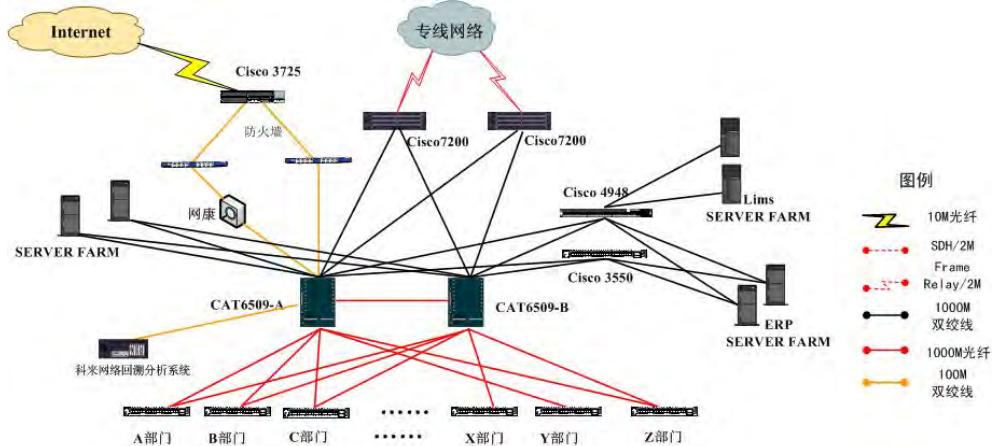
2011年7月8日，某公司网络管理人员通过网管软件发现两台核心网络交换机CPU利用率异常，如下：

1. “核心交换机 6509_A”的 CPU 利用率高达 90%以上。
2. “核心交换机 6509_B”的 CPU 利用率高达 90%以上。

以上问题造成网络延时很高，导致访问内网应用、互联网等速度较慢。

1. 网络拓扑

拓扑图如下：



2. 检测描述

监测软件：科来网络回溯分析系统 3.1

样本文件：Colasoft.pkt

采样时间：2011-7-8 21:30

采样时长：7*24

样本说明：核心交换机 6509 连接部门交换机 3550 的 trunk 链路

6.3. 分析内容

1. 基本分析

首先，我们需要检查是什么进程导致设备 CPU 利用率较高，以提高分析效率。我们分别在两台（A、B）Cisco 6509 交换机上执行 `show process cpu` 命令，查看各进程 CPU 占用情况，如下：

=====6509_A=====

6509_A#sho processes cpu								
CPU utilization for five seconds: 2%/1%; one minute: 75%; five minutes: 87%								
PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	136	4349		31	0.00%	0.00%	0.00%	0 Chunk Manager
2	2276	6300160		0	0.00%	0.00%	0.00%	0 Load Meter
3	1344	437500		3	0.00%	0.00%	0.00%	0 SSH Event handle
4	20	262509		0	0.00%	0.00%	0.00%	0 DHCPD Timer
5	65638348	5312249		12356	0.00%	0.16%	0.20%	0 Check heaps
6	63976	526226		121	0.00%	0.00%	0.00%	0 Pool Manager
7	0	2		0	0.00%	0.00%	0.00%	0 Timers
8	0	2		0	0.00%	0.00%	0.00%	0 Serial Backgroun
9	0	1		0	0.00%	0.00%	0.00%	0 AAA SERVER_DEADT
10	0	2		0	0.00%	0.00%	0.00%	0 AAA high-capacit
11	1533376	16786664		91	0.00%	0.00%	0.00%	0 EnvMon
12	12	525018		0	0.00%	0.00%	0.00%	0 IPC Dynamic Cach
13	0	44		0	0.00%	0.00%	0.00%	0 PF_Split Sync Pr
14	0	1		0	0.00%	0.00%	0.00%	0 IPC BackPressure
15	1824	31499492		0	0.00%	0.00%	0.00%	0 IPC Periodic Tim
16	756	31499495		0	0.00%	0.00%	0.00%	0 IPC Deferred For
17	9892740	10572223		935	0.00%	0.00%	0.00%	0 IPC Seat Manager
18	0	1		0	0.00%	0.00%	0.00%	0 IFS Agent Manag
19	243923900	741583168		328	0.15%	49.19%	58.61%	0 ARP Input
20	0	2		0	0.00%	0.00%	0.00%	0 DDR Timers
21	0	2		0	0.00%	0.00%	0.00%	0 Dialer event
22	184	247		744	0.00%	0.00%	0.00%	0 Entity MIB API
23	220	3150093		0	0.00%	0.00%	0.00%	0 Compute SRP rate

=====6509_B=====

6509_B#sho processes cpu							
PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY Process
1	52	1830	28	0.00%	0.00%	0.00%	0 Chunk Manager
2	1040	6300202	0	0.00%	0.00%	0.00%	0 Load Meter
3	1196	437450	2	0.00%	0.00%	0.00%	0 SSH Event handle
4	8	262511	0	0.00%	0.00%	0.00%	0 DHCPD Timer
5	66295116	5313424	12476	0.00%	0.28%	0.21%	0 Check heaps
6	56432	526332	107	0.00%	0.00%	0.00%	0 Pool Manager
7	0	2	0	0.00%	0.00%	0.00%	0 Timers
8	0	2	0	0.00%	0.00%	0.00%	0 Serial Backgroun
9	0	1	0	0.00%	0.00%	0.00%	0 AAA_SERVER_DEADT
10	0	2	0	0.00%	0.00%	0.00%	0 AAA high-capacit
11	1559808	17837463	87	0.00%	0.00%	0.00%	0 EnvMon
12	4	525022	0	0.00%	0.00%	0.00%	0 IPC Dynamic Cach
13	8	44	181	0.00%	0.00%	0.00%	0 PF_Split Sync Pr
14	0	1	0	0.00%	0.00%	0.00%	0 IPC BackPressure
15	1264	31499636	0	0.00%	0.00%	0.00%	0 IPC Periodic Tim
16	568	31499634	0	0.00%	0.00%	0.00%	0 IPC Deferred Por
17	9042120	13667521	661	0.00%	0.00%	0.00%	0 IPC Seat Manager
18	0	1	0	0.00%	0.00%	0.00%	0 IFS Agent Manage
19	230205000	891130454	258	59.03%	58.80%	58.39%	0 ARP Input
20	0	2	0	0.00%	0.00%	0.00%	0 DDR Timers
21	0	2	0	0.00%	0.00%	0.00%	0 Dialer event
22	116	115	1008	0.00%	0.00%	0.00%	0 Entity MIB API

从上图可以看出两台设备占用 CPU 利用率最高的进程为 ARP 进程，统计结果如下：

设备名称	6509_A	6509_B
ARP 进程占 CPU 利用率 (%)	58.6	58.4

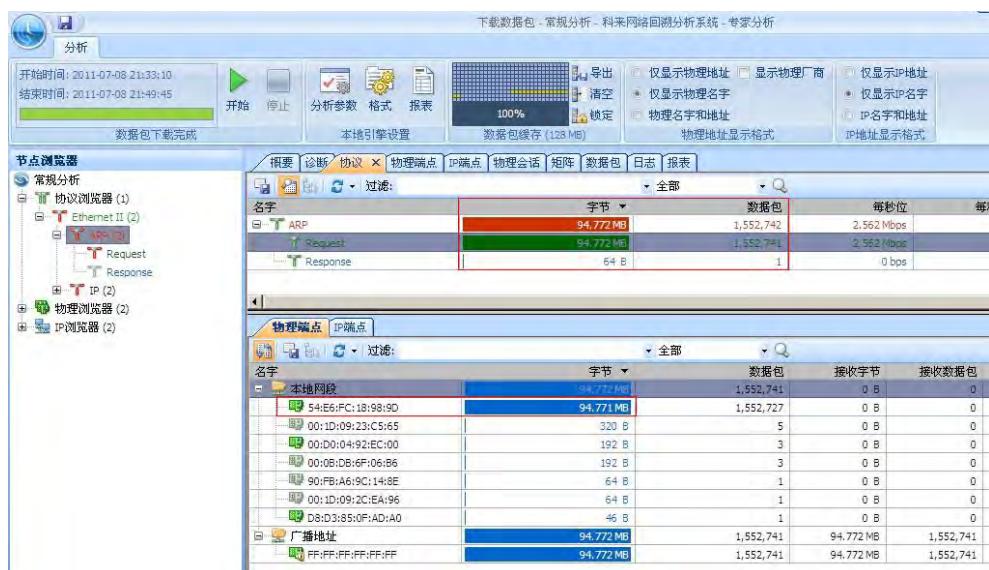
因此，我们推断设备 CPU 利用率较高是由 ARP 流量异常导致的，对于分析 ARP 流量异常，我们需要借助专业的流量分析工具来网络回溯分析系统 3.1。

2. 详细分析

由于用户先前已经部署了科来网络回溯分析系统，并且部署监控点包括核心交换机连接各部门交换机 3550 的端口，因此可以监控到所有 Vlan 的流量。



由于在基本分析的时候我们判断故障原因为 ARP 流量异常造成，因此，我们选择最近时间段的全部流量下载分析即可。下载完成之后，我们定位 ARP 流量进行详细分析，如下图：



在图中，我们可以看到 ARP Request 包的数量远远高于 ARP Response 包的数量，并且这些 ARP Request 包大部分是由 Mac54:E6:FC:18:98:9D 发出来的，现在我们可以直接分析 Mac:54:E6:FC:18:98:9D 所发出的数据包。如下图：



上图显示，Mac 为 54:E6:FC:18:98:9D 的设备发送的数据包为免费 ARP 请求数据包，并且发送频率较高（正常情况下，设备不会发送大量发送免费 ARP 请求）。这种数据包发送到网络当中会导致拥有相同 IP 的主机不停地产生地址冲突提示。

由于已经找到 ARP 报文中的 IP 地址，我们通过 IP 登记记录找到该 IP 10.168.22.215 为一台 IBM 服务器（Server 2003），我们登陆该服务器查看网卡 Mac 为 00:09:6B:A5:19:C4，并且系统不停地提示 IP 地址冲突。我们将该服务器网卡禁用之后，核心交换机设备利用率立即恢复正常，并且 ARP 流量也恢复正常（通过这个现象可以判断非恶意破坏）。

至此，我们确定 ARP 流量异常原因为某台设备配置 IP 地址与一台 IBM 服务器地址冲突，而此设备在 IP 冲突的时候为了抢占该 IP 地址，大量发送免费 ARP 请求，造成网络产生了 ARP 广播风暴，最终导致核心设备 CPU 利用率升高。

6.4. 故障点定位

由于公司网络全部采用的可管理的交换机，因此根据发送 ARP Request 的源 Mac 地址 54:E6:FC:18:98:9D，我们找到该设备具体接在那个交换机端口。通过在核心交换机执行 show mac-address-table | include 54:E6:FC:18:98:9D 找到该 Mac 在某部门的 3550 交换机上，登陆该交换机我们再次执行该命令，最终找到该 Mac 所接交换机端口，如下图：

```
3550-3#show mac-address-table interface fastEthernet 0/25
Mac Address Table

Vlan      Mac Address          Type      Ports
----      -----              -----      -----
  112      54e6.fc18.989d    DYNAMIC   Fa0/25
Total Mac Addresses for this criterion: 1
Yanfa-3550-3#
```

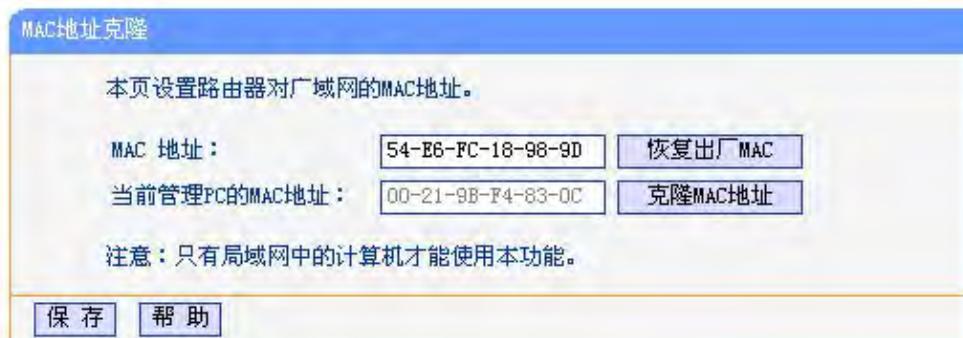
```
3550-3#sh run interface fastEthernet 0/25
Building configuration...

Current configuration : 86 bytes
!
interface FastEthernet0/25
  switchport access vlan 112
  switchport mode access
end

3550-3#
```

从上图可以看到 Mac 地址在该部门交换机的 F0/25 端口上，并且该端口属于 Vlan112。

查阅布线图，我们最终找到了该端口所接设备的具体房间。并且找到了 Mac 为 54:E6:FC:18:98:9D 的设备为一台 Tplink 无线路由器，并且其配置了 IP 地址为 10.168.22.215，如下图：



1. 处理方法

通过科来网络回溯分析系统 3.1，我们快速定位到触发核心交换机 CPU 利用率非常高的原因为某部门擅自使用了一台 Tplink 无线路由器，并且该路由器“Wlan 口”设置的 IP 与同网段的一台 IBM 服务器地址冲突，导致 Tplink 无线路由器快速发送免费 ARP 请求（防护机制），最终导致核心交换机 CPU 利用率升高。

根据以上分析结果，该部门已停用该 Tplink 无线路由器，重新申请了公司购买的企业级无线路由器，同时申请了新的 IP 地址。

2. 处理结果

在停用 Tplink 无线路由之后，网络已经恢复正常，核心交换机 CPU 利用率已经恢复正常，网络延时也恢复到局域网延时水平。

6.5. 分析总结

1. 分析结果

本次通过科来网络回溯分析系统 3.1，我们快速定位本次故障原因为一台非授权 Tplink 无线路由器非法接入网络引起。针对分析结果我们及时采取了处理措施，最终将问题解决。

2. 网络优化建议

针对本次故障，我们看到某些网络设备不适合企业用户使用，同时企业需要采取更为严格网络管理措施防范此类故障的再次发生。

网络中部署准入控制，加强网络设备接入的管理

建议用户使用企业级无线路由器或者交换机，以保障企业网络的稳定运行。

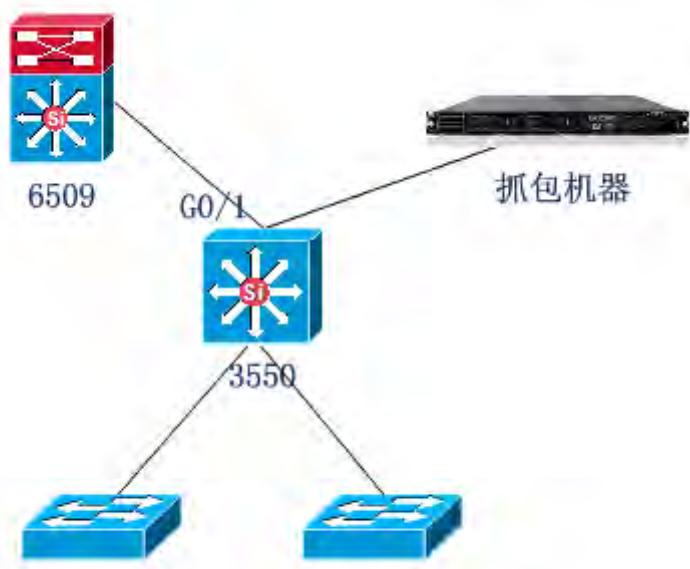
7. 环路分析

7.1. 故障描述

1. 故障背景

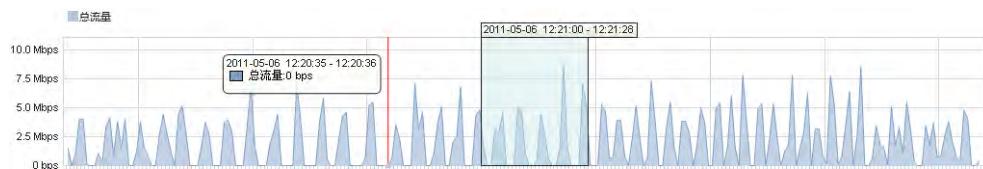
一天，一个客户打电话说他们部门的内网阶段性掉线，而且频率很高，访问外网非常慢，ping 网关，时通时不通，延时较大，部门所有机器都是类似情况！初步判断，有可能是内网异常流量占用，或者广播风暴之类的故障！还好，客户在使用科来回溯产品，可以保存数据，以便于现在回放“录像”。

简单描述一下网络拓扑，比较简单，只是一个部门网络，汇聚交换机——二层交换机——用户。抓包位置，汇聚交换机作镜像，镜像端口接科来回溯分析系统。网关地址：192.168.10.1



选取时间为 28s，总流量 16.312M，速率也是比较大的(如图)！

2. 概要统计



7.2. 下载数据包分析

物理错误						00:00:28
错误包合计						0
CRC错误包						0
对齐错误包						0
过大错误包						0
过小错误包						0
802.3错误						数据包
802.3错误包合计						0
802.3一次冲突						0
802.3多次冲突						0
802.3最大冲突						0
802.3延迟发送						0
网络流量						每秒包数
总共流量	16.312 MB	116,261	77.758%	7.776 Mbps	6,873	
发送广播流量	14.997 MB	94,721	71.540%	7.154 Mbps	5,459	
发送组播流量	1.314 MB	21,530	6.216%	621.568 ...	1,214	
数据包大小分布						每秒包数
<=64	1.314 MB	21,532	6.218%	621.568 ...	1,214	
65-127	5.441 MB	62,572	26.582%	2.658 Mbps	3,638	
128-255	2.098 MB	9,998	10.703%	1.070 Mbps	608	
256-511	7.459 MB	22,159	34.255%	3.426 Mbps	1,213	
512-1023	0 B	0	0.000%	0 bps	0	
1024-1517	0 B	0	0.000%	0 bps	0	
>=1518	0 B	0	0.000%	0 bps	0	
TCP数据包						每秒包数
TCP同步数据包	0 B	0	0.000%	0 bps	0	
TCP结束连接数据包	140 B	2	0.000%	0 bps	0	
TCP复位数据包	128 B	2	0.000%	0 bps	0	

端点视图：



广播和组播流量比较大，几乎占了总流量的 98%。

先前的判断的方向是对的！而且还有两个无效地址 0.0.0.0, 169.254.134.187，没有获取到地址，忘了说了，地址都是自动获得的！

看了协议视图，豁然开朗



是 dhcp 和 netbios 在作怪！

结合 dhcp 和 netbios 的联动性（见注释 1），初步确定是 dhcp 在作怪了，正是由于 dhcp 的缘故，所以出现了 169.254.x.x 和 0.0.0.0 这样的地址。

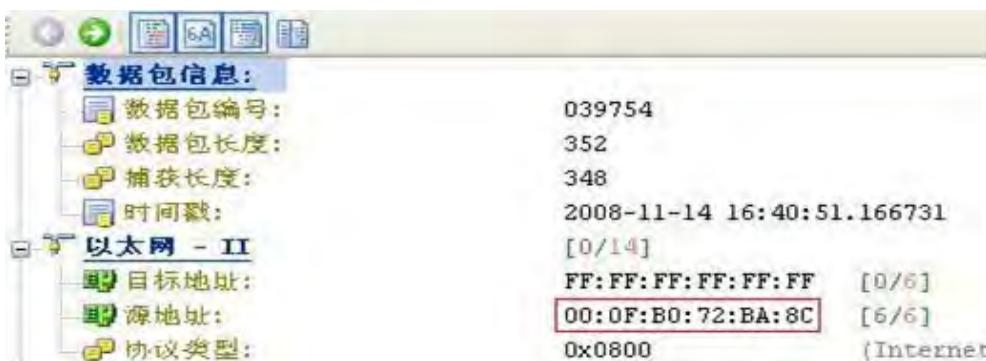
DHCP 的工作原理（见注释 2），现在我们只说第一次登录的时候。

根据客户端是否第一次登录网络，DHCP 的工作形式会有所不同。

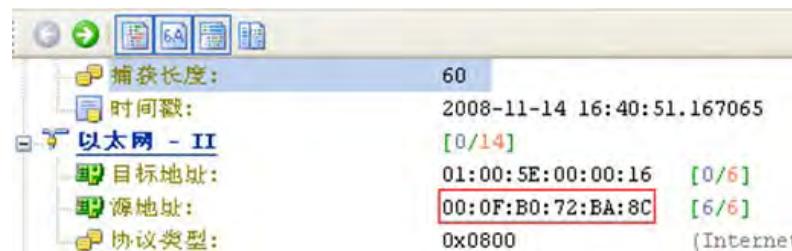
我们只说第一次登录的时候，当 DHCP 客户端第一次登录网络的时候，也就是客户发现本机上没有任何 IP 数据设定，它会向网络发出一个 DHCP discover 封包。因为客户端还不知道自己属于哪一个网络，所以封包的来源地址会为 0.0.0.0，而目的地址则为 255.255.255.255，然后再附上 DHCP discover 的信息，向网络进行广播。在 Windows 的预设情形下，DHCP discover 的等待时间预设为 1 秒，也就是当客户端将第一个 DHCP discover 封包送出去之后，在 1 秒之内没有得到响应的话，就会进行第二次 DHCP discover 广播。若一直得不到响应的情况下，客户端一共有四次 DHCP discover 广播(包括第一次在内)，除了第一次会等待 1 秒之外，其余三次的等待时间分别是 9、13、16 秒。如果都没有得到 DHCP 服务器的响应，客户端则会显示错误信息，宣告 DHCP discover 的失败。

Windows 操作系统预设，客户端如果学不到地址，系统自动会把一个 169.254.x.x 分配给它。之后，基于使用者的选择，系统会继续在 5 分钟之后再重复一次 DHCP discover 的过程。

这是数据包 0.0.0.0 的解码图，是 mac 地址为 00:0f:b0:72:ba:8c 发送的广播包，在进行 dhcp discover



这是数据包 169.254.134.187 的解码图，是 mac 地址为 00:0f:b0:72:ba:8c 发送的组播包



说明是同一个客户端 00:0f:b0:72:ba:8c 发出的数据包，可是为什么没有学到地址呢？又为什么会产生如此大的流量呢？

带着这个疑问我们继续分析：首先定位 DHCP 包，深入分析。

这是 00:0f:b0:72:ba:8c 发出的数据包

BOOTP - 自举协议		
操作码:	1	(请求) [42/1]
硬件类型:	1	(以太网) [43/1]
硬件长度:	6	[44/1]
跳数:	0	[45/1]
事务标识:	2154364198	[46/4]
引导秒数:	7424	[50/2]
标志:	0x8000	(广播) [52/2]
客户端已经知道的IP地址:	0.0.0.0	[54/4]
服务器给客户端的IP地址:	0.0.0.0	[58/4]
服务器IP地址:	0.0.0.0	[62/4]
网关IP地址:	0.0.0.0	[66/4]
客户端硬件地址:	00:0F:B0:72:BA:8C	[70/6]
保留:	[76/10]	
服务器名:	没有给出	[86/64]
引导文件名:	没有给出	[150/128]

DHCP - 动态主机配置协议		
事务ID:	1669485411	[278/4]
DHCP消息类型		
标签:	53	[282/1]
长度:	1	[283/1]
消息类型:	1	(搜索) [284/1]

是 dhcp discover 包，向服务器请求地址，大家看，标签 53 是“dhcp 报文类型”的标签，消息类型为 1 说明 dhcp discover 包，在向服务器请求地址。

再看 192.168.10.1 发的包：

BOOTP - 自举协议		[42/236]
操作码:	2	(应答) [42/1]
硬件类型:	1	(以太网) [43/1]
硬件长度:	6	[44/1]
跳数:	0	[45/1]
事务标识:	3800947217	[46/4]
引导秒数:	0	[50/2]
标志:	0x0000	[52/2]
客户端已经知道的IP地址:	0.0.0.0	[54/4]
服务器给客户端的IP地址:	192.168.10.9	[58/4]
服务器IP地址:	0.0.0.0	[62/4]
网关IP地址:	0.0.0.0	[66/4]
客户端硬件地址:	00:0F:B0:72:BA:8C	[70/6]
保留:	[76/10]	
服务器名:	没有给出	[86/64]
引导文件名:	没有给出	[150/128]
DHCP - 动态主机配置协议		[278/65]
事物ID:	1669485411	[278/4]
DHCP消息类型		[282/3]
标签:	53	[282/1]
长度:	1	[283/1]
消息类型:	2	(提供) [284/1]

大家再看标签类型为 53 的“消息类型”，为 2，说明是 dhcp server 发的 dhcp offer 包，说明服务器给 00:0f:b0:72:ba:8c 分配 192.168.10.9 的地址了，那为什么会有如此多的数据包呢？

我们接着分析！既然服务器为 00:0f:b0:72:ba:8c 提供了地址，说明它收到了 00:0f:b0:72:ba:8c 的 discover 包，那就说明服务器没有问题以及他们之间的网络通信是好的，难道问题出在 00:0f:b0:72:ba:8c 上？

让客户先把 00:0f:b0:72:ba:8c 关了，看看现象。

问题依然存在！看来问题不在它身上！

既然服务器和客户机以及他们之间的链路是畅通的，那问题一定出在中间设备上。

难道是网络中存在环路？看来只能先分析一下数据包再说！

定位到 dhcp 协议，看 dhcp 事物 id 和 ip 的标识 id

dhcp 事物 id 如下：

绝对时间	源	目标	协议	大小	DHCP:事务ID
16:40:51.166731	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	1669485411
16:40:51.167619	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	1669485411
16:40:51.167972	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	1669485411
16:40:51.168818	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	1669485411
16:40:51.169156	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	1669485411
16:40:51.170997	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	1669485411
16:40:51.171472	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	1669485411
16:40:51.172570	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	1669485411
16:40:51.172934	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	1669485411
16:40:51.173760	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	1669485411
16:40:51.174109	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	1669485411
16:40:51.175910	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	1669485411
16:40:51.176349	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	1669485411
16:40:51.177208	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	1669485411
16:40:51.177541	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	1669485411
16:40:51.178400	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	1669485411
16:40:51.178745	00:0F:80:72:BA:8C	255.255.255.255:hnntns	DHCP	352	1669485411

竟然是同一个事物 id，就是说是同一个事件的数据包，看来真的有可能存在环路了。

为了证明以上观点，再看 ip identify id:

绝对时间	源	目标	协议	大小	IP:标识
16:40:51.166731	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	0x002A
16:40:51.167619	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	0x392A
16:40:51.167972	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	0x002A
16:40:51.168818	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	0x392A
16:40:51.169156	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	0x002A
16:40:51.170997	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	0x392A
16:40:51.171472	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	0x002A
16:40:51.172570	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	0x392A
16:40:51.172934	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	0x002A
16:40:51.173760	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	0x392A
16:40:51.174109	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	0x002A
16:40:51.175910	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	0x392A
16:40:51.176349	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	0x002A
16:40:51.177208	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	0x392A
16:40:51.177541	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	0x002A
16:40:51.178400	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	0x392A
16:40:51.178745	00:0F:80:72:BA:8C	255.255.255.255:hnntns	DHCP	352	0x002A

接着再看 TTL，看是否存在路由环路：

绝对时间	源	目标	协议	大小	IP:生存时间
16:40:51.166731	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.167619	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64
16:40:51.167972	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.168818	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64
16:40:51.169156	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.170997	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64
16:40:51.171472	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.172570	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64
16:40:51.172934	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.173760	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64
16:40:51.174109	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.175910	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64
16:40:51.176349	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.177208	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64
16:40:51.177541	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.178400	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64
16:40:51.178745	00:0F:80:72:BA:8C	255.255.255.255:bootps	DHCP	352	128
16:40:51.180204	192.168.10.1:bootps	255.255.255.255:bootpc	DHCP	354	64

TTL 值没有变，说明不是路由环路，那就是交换机的问题了。。。

随后让客户检查交换机。

结论，没有发现环路，这是怎么回事？

思索中。。。

难道是交换机出了故障，让换之测试。

郁闷，问题还存在！

就在我边郁闷边思索时，客户在一个犄角旮旯里发现了一个布满灰尘的 sohu 交换机串在 pc 和交换机之间！

我大喜过望，罪魁祸首出来了。令其去掉该设备，网络正常了。

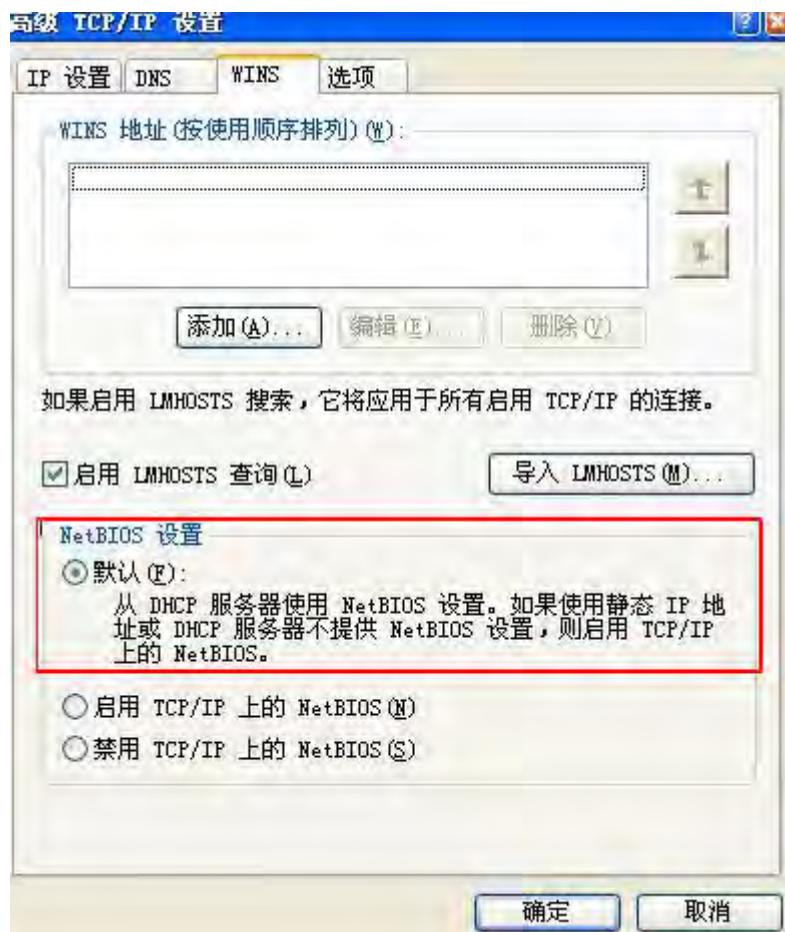
可是客户仔细查看了该 sohu 交换机，并没有打环的迹象！

继续思索着。

突然想起某天某月某日某点某分在 internet 上看到某位高人写到：有时一些 soho 交换机会无缘无辜对收到的数据包进行无限转发。又一次无语了。。。

问题解决了，思路似乎也比较清晰了，希望可以提醒以后遇到这种情况的同志不要忽视类似的小东东。。。

注释 1：dhcp 和 netbios 的联动性



注释 2：dhcp 报文类型（选项 53）取值的含义：

类型为 1：dhcp discover 此为 client 开始 DHCP 过程中的第一个请求报文

类型为 2: **dhcp offer** 此为 **server** 对 **dhcpdiscover** 报文的响应

类型为 3: **dhcp request** 此为 **client** 对 **dhcpooffer** 报文的响应

类型为 4: **dhcp decline** 此为当 **client** 发现 **server** 分配给它的 IP 地址无法使用, 如 IP 地址发生冲突时, 将发出此报文让 **server** 禁止使用这次分配的 IP 地址。

类型为 5: **dhcp ack** 此为 **server** 对 **dhcprequest** 报文的响应, **client** 收到此报文后才真正获得了 IP 地址和相关配置信息。

类型为 6: **dhcp nack** 此为 **server** 对 **client** 的 **dhcprequest** 报文的拒绝响应, **client** 收到此报文后, 一般会重新开始 DHCP 过程。

类型为 7: **dhcp release** 此报文是 **client** 主动释放 IP 地址, 当 **server** 收到此报文后就可以收回 IP 地址分配给其他的 **client**.

类型为 8: **dhcp inform** 此为如果客户通过别的手段获得了网络地址, 它可以使用 **DHCPINFORM** 请求获得其它配置参数, 服务器接收到 **DHCPINFORM** 包, 并建立一个 **DHCPCPACK** 消息, 在其中包括一些合适客户的配置参数, 只是不包括分配网络地址, 检查现有的绑定, 在信息中不填充'yiaddr'字段或租用时间参数。服务器取得 **DHCPINFORM** 包内的'ciaddr'地址, 而返回 **DHCPCPACK** 包。

详情请参照 **DHCP** 协议之 **RFC** 文件

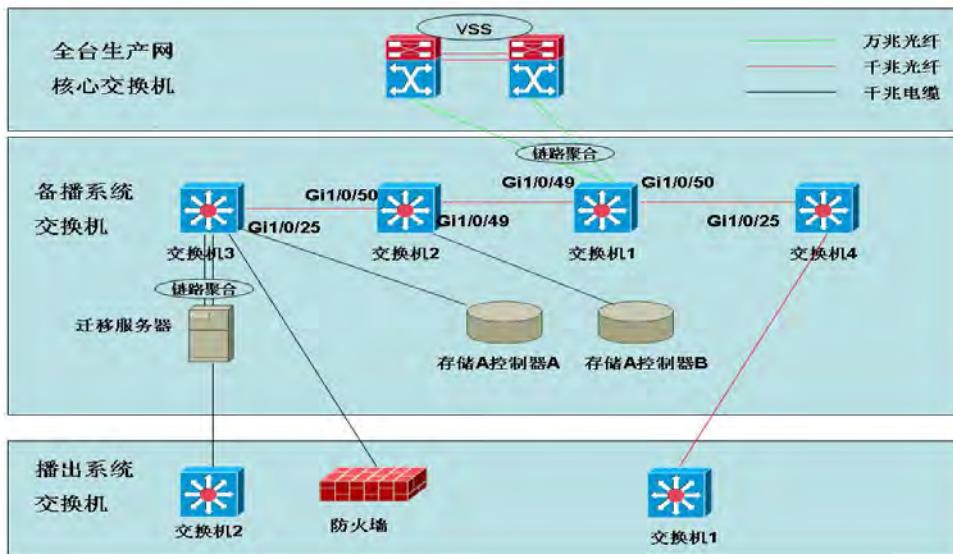
8. 某电视台故障处理报告

8.1. 故障描述:

1. 故障现象

某电视台备播系统存储设备扩容过程中产生瞬时网络风暴，导致重大播出事故，采用应急处理后网络恢复正常，电视台领导责令查找网络风暴产生原因。

2. 基础网络环境



8.2. 分析方案

1. 分析目标

查找备播系统内部产生瞬时网络风暴的原因。

2. 分析思路

第一步：先检查备播网络环境中是否存在干扰因素：如病毒、网络配置不当导致网络环路；

第二步：如果没有干扰因素，则需要重现存储扩容操作过程，查找是否因操作异常导致网络风暴出现；

工具软件：利用科来协议分析仪进行精确定位；

8.3. 排除环境干扰

1. 定位是否为病毒引起

1) 病毒特征分析

根据上面的现象，能够出现这种网络风暴的病毒一般情况是蠕虫或者攻击类的病毒，该病毒有如下几

种特征：

2) 蠕虫类病毒

网络层：同大量的主机会话，大多是发包，每个会话流量很少；

连接层：连接很多，大多是发出的 TCP SYN 包，大部分没有响应或被拒绝；

总体流量：发包远大于收包数量

3) arp 病毒

Arp 扫描病毒：发送大量 arp 请求，扫描本网段内的 mac 地址，消耗交换机资源；

Arp 欺骗攻击：通过主动发送大量的 arp 响应实现地址欺骗，从而获取其它主机通讯信息，

4) 结合现场环境情况分析

网络中的计算机设备被感染该类病毒后会扫描所有网段内的设备，根据故障现象描述，单台被感染的计算机设备扫描网络的流量不会这么大，除非备播系统内部很多的计算机设备被感染，而且是在同一个时间进行这样的操作，否则不会导致备播系统内部 4 台交换机的指示灯全部狂闪。

根据了解系统升级前后运行状态均正常，但是需要查看是否有隐含该类病毒的情况，即是否个别计算机设备存在这类现象，只是数据量比较小，交换机表现不明显；另外根据故障现象，产生网络风暴应该主要是广播风暴，因此通过抓取广播数据包能够检查是否存在该类现象；

2. 确认病毒的方法

部署协议分析仪（笔记本电脑安装科来软件）：

监控对象：在每台交换机上镜像所有端口流量到协议分析仪端口

科来软件抓包结果：

1) 分析软件诊断界面中没有出现频率很高的扫描行为；

2) 诊断界面

出现很多 IP 地址冲突 192.168.0.128，经过了解，该 IP 在所有的计算机上都存在，是 HP 服务器默认管理 IP，经过了解对网络无任何影响，业务系统中也没有使用该网段；

少量 arp 扫描信息，172.27.112.36、172.27.112.6，经过了解这 2 个 IP 扫描属于正常业务情况，这 2 个服务器正在查找几个业务服务器，而这些服务器已经关机；

3. 结果判定

备播系统内正常情况下无异常流量；

4. 定位网络配置

根据上面备播系统网络拓扑图和实际的交换机配置，网络设备物理连接基本为单链路连接，网络设备为 2 层配置，但是和全台网互联为 3 层连接，因此初步排查备播交换机网络设备的配置不会存在网络路由环路，但是需要进一步验证是否受全台网核心交换机的影响。

网络路由环路原理：

同一个数据包在路由器间循环传输最终丢掉；

由于路由实际上是不可达的，IP 包的 TTL 值在传输过程中不断减小直至 1；

路由器在丢掉数据包时会向源地址发送 ICMP 数据包；

网络物理环路原理：

同一个数据包在 2 台设备间无限循环传输，不丢弃；

循环广播报文形成广播风暴（广播报文死循环），导致整个网络阻塞；

部署协议分析仪（笔记本电脑安装科来软件）：

监控对象：镜像备播系统内 1 台服务器访问通信信息到协议分析仪端口

验证思路：

看数据包解码中 3 个参数：IP ID、TTL、ICMP 是否同时出现，并且 3 个参数的特征如下：

IP ID：如果 PID 相同的 tcp/udp 数据包则表示同一个数据包

TTL：即同一个数据包的 TTL 为第一个值逐渐减 1，到最后 TTL 为 1；

检查是否有 ICMP 协议返回给该服务器；

科来分析仪抓包结果：

诊断界面：只有诊断界面中出现很多 IP 地址冲突 192.168.0.128，；

出现很多 IP 地址冲突 192.168.0.128，前面已经介绍过对网络无任何影响；没有异常报警；

少量 arp 扫描信息，172.27.112.36、172.27.112.6，经过了解这 2 个 IP 扫描属于正常业务情况，这 2 个服务器正在查找几个业务服务器，而这些服务器已经关机；

数据包解码界面：3 个参数没有上面的特征

分析结果：没有网络路由环路和物理环路产生；

5. 环境干扰分析总结

根据前面对病毒、网络配置的分析结果，网络中的设备无明显大量发包情况，现场网络环境正常，无任何干扰因素。

8.4. 重现故障

思路：根据前面环境干扰分析总结结果，网络环境正常，而网络风暴是瞬时出现，怀疑是操作过程中操作不当导致，因此让存储扩容人员详细讲解和重现扩容当天的操作过程和线缆连接过程，看是否出现故障现象，并通过科来协议分析仪来详细分析和定位。

1. 恢复连接存储 1 的 B 控制器的数据端口

说明：

存储控制器上有 2 个 RJ45 类型的端口，其中 1 个端口为数据访问端口，另外 1 个端口为存储固件升级专用管理口。

1) 交换机

备播系统 4 台交换机没有出现网络风暴现象；

2) 科来协议

监控对象：交换机上连接存储 1 的 B 控制器的数据端口

抓包时间：15 分钟；

概要界面：广播流量所占比例很低；

诊断界面：没有异常报警信息；

矩阵界面：矩阵连接数量正常，未见明显异常会话连接；

Tcp 会话/udp 会话/IP 会话界面：均正常连接，未见发包数量很多的会话；

2. 连接存储 1 的B控制器的固件专用管理端口

根据存储扩容人员回忆，操作过程中 A 控制器的数据口能够连通，但是 B 控制器的数据口不通，怀疑是端口插错了，于是直接将不通的线缆拔下来插到另外 1 个端口上，然后去做其它的业务操作，之后便出现了交换机所有指示灯全部狂闪的现象（实际情况是该操作人员把 2 个控制器标记给记颠倒了）。

根据其提供的信息我们按照其操作过程演示了一遍：

1) 第一次：网线 1 连接A控制器的数据口，网线 2 连接B控制器的固件管理口

现象：网线 1 能够连通，网线 2 连接后不能 ping 通 172.27.112.201；

交换机现象：备播系统 4 台交换机没有出现网络风暴现象；

部署协议分析仪（笔记本电脑安装科来软件）：

监控对象：镜像交换机连接 B 控制器的固件管理口的通信信息到协议分析仪端口

抓包分析结果：

抓包时间：15 分钟；

诊断界面：没有异常报警信息；

概要界面：广播流量所占比例很低；

矩阵界面：矩阵连接数量正常，未见明显异常会话连接；

Tcp 会话/udp 会话/IP 会话界面：均正常连接；

2) 第二次：网线 1 连接A控制器固件管理口，网线 2 连接B控制器固件管理口

交换机现象：备播系统 4 台交换机立刻出现网络风暴现象；

部署协议分析仪（笔记本电脑安装科来软件）：

监控对象：镜像交换机连接 B 控制器的固件管理口的通信信息到协议分析仪端口

抓包分析结果：

抓包时间：持续；

诊断界面：网卡 2 连通后立刻出现 arp 请求风暴，并且数量不断增加，源 IP 为 2 台服务器 IP；

The screenshot shows the Colasoft NetworkMiner interface. At the top, there's a navigation bar with tabs: 概要 (Summary), 诊断 (Diagnosis) (which is selected and highlighted in yellow), 协议 (Protocols), 物理端点 (Physical Endpoints), IP端点 (IP Endpoints), 物理会话 (Physical Sessions), 矩阵 (Matrix), 数据包 (Data Packets), 日志 (Logs), and 报表 (Reports). Below the navigation bar is a toolbar with icons for search, filter, and export.

诊断条目 (Diagnosis Items):

名字 (Name)	数量 (Count)
所有诊断 (All Diagnoses)	1,198
数据链路层 (Data Link Layer)	845
ARP 请求风暴 (ARP Request Storm)	845

诊断事件 (Diagnosis Events):

严重程度 (Severity)	类型 (Type)	层别 (Layer)	事件描述 (Event Description)	源IP地址 (Source IP Address)	源物理地址 (Source Physical Address)
安全 (Safe)	数据链路层 (Data Link Layer)	ARP请求风暴 (主机的MAC为 50:78:4C:70:F2:DE, IP地址为 172.27.112.36)	172.27.112.36	50:78:4C:70:F2:DE	
安全 (Safe)	数据链路层 (Data Link Layer)	ARP请求风暴 (主机的MAC为 50:78:4C:70:F2:DE, IP地址为 172.27.112.36)	172.27.112.36	50:78:4C:70:F2:DE	
安全 (Safe)	数据链路层 (Data Link Layer)	ARP请求风暴 (主机的MAC为 50:78:4C:70:F2:DE, IP地址为 172.27.112.36)	172.27.112.36	50:78:4C:70:F2:DE	
安全 (Safe)	数据链路层 (Data Link Layer)	ARP请求风暴 (主机的MAC为 50:78:4C:70:F2:DE, IP地址为 172.27.112.36)	172.27.112.36	50:78:4C:70:F2:DE	
安全 (Safe)	数据链路层 (Data Link Layer)	ARP请求风暴 (主机的MAC为 50:78:1C:00:F2:41, IP地址为 172.27.112.6)	172.27.112.6	50:78:1C:00:F2:41	
安全 (Safe)	数据链路层 (Data Link Layer)	ARP请求风暴 (主机的MAC为 50:78:1C:00:F2:41, IP地址为 172.27.112.6)	172.27.112.6	50:78:1C:00:F2:41	
安全 (Safe)	数据链路层 (Data Link Layer)	ARP请求风暴 (主机的MAC为 50:78:1C:00:F2:41, IP地址为 172.27.112.6)	172.27.112.6	50:78:1C:00:F2:41	

概要	诊断	协议	物理端点	IP端点	物理会话	矩阵	数据包	日志	报表	数据包:
编号	时间差	源		目标	协议	大小	解码字段	概要		41,64
1		50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		
2	0.000037	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.210? 告诉 172.27.112.36		
3	0.000016	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.152? 告诉 172.27.112.36		
4	0.003229	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.38? 告诉 172.27.112.36		
13	0.204679	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		
14	0.000927	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		
15	0.000914	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.38? 告诉 172.27.112.36		
16	0.089383	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.38? 告诉 172.27.112.36		
21	0.308830	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.38? 告诉 172.27.112.36		
22	0.100756	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.38? 告诉 172.27.112.36		
23	0.514240	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.38? 告诉 172.27.112.36		
24	0.000045	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.38? 告诉 172.27.112.36		
25	0.000022	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.27.112.38? 告诉 172.27.112.36		
26	0.100142	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		
28	0.307372	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		
29	0.205137	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		
30	0.204580	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		
31	0.204469	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		
32	0.209024	50:78:4C:70:F2:DE		FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.36		

矩阵界面：矩阵连接数量正常，未见明显异常会话连接；

数据包解码：主机 172.27.112.36 和 172.27.112.6 发送 arp 请求的数据包非常频繁；访问的目标 IP 比较固定，经过了解这些 IP 是业务工作站和数据库的 IP，只是当时这些设备关机了；

处理：将连接 B 控制器固件管理口网线 2 拔掉后，网络立刻恢复正常；

3) 第三次：将有问题的 2 台服务器网卡断网，并重新将网线 2 连接 B 控制器固件管理口

网线连通后没过多久，又再次出现下列现象：

交换机现象：备播系统 4 台交换机立刻出现网络风暴现象；

部署协议分析仪（笔记本电脑安装科来软件）：

监控对象：镜像交换机连接 B 控制器的固件管理口的通信信息到协议分析仪端口

抓包分析结果：

抓包时间：持续抓包；

诊断界面：再次出现 arp 请求风暴，并且数量不断增加，源 IP 为其它服务器 IP；

处理：将连接 B 控制器固件管理口网线 2 拔掉后，网络又立刻恢复正常；

至此找到该故障现象，出现交换机网络风暴是人为操作不当导致：网线应该连接到控制器数据口，但是被错误的连接到控制器的固件管理端口上，经过与存储扩容操作人员确认，确实有这种情况；

8.5. 深入分析与结论

1. 存储控制器

第一：2 个存储控制器的固件专用管理口实际上为 1 个物理端口，可能是控制器板卡内部部件将这 2 个固件的管理端口连通，经过与厂商工程师了解得到确认；

第二：由于 2 个存储控制器的固件专用管理口为 1 个端口，因此同时连接 2 个控制器的固件管理口会直接将交换机 2 和交换机 3 物理连通，导致交换机物理环路；



2. 交换机被物理环路的表现特征

向广播地址 `x.x.x.255` 发送的数据包频率很高，在毫秒级；

向广播地址 `x.x.x.255` 发送的数据包的 IP ID 号相同、TTL 值不变；

发生物理环路会出现 arp 请求风暴报警，即网络中同时会伴随大量的 arp 请求数据包出现，不能找到目标 MAC 的 arp 请求数据包被交换机重复转发，造成死循环，最终导致 arp 请求风暴出现；同时如果将出现 arp 请求风暴的设备断网，那么会陆续有其它设备接连出现请求风暴报警；

概要 **诊断** **X** 协议 物理端点 IP 端点 物理会话 矩阵 数据包 日志 报表

诊断条目

名字	数量
所有诊断	1273
数据链路层	948
ARP 请求风暴	948

诊断事件

严重程度	类型	层别	事件描述	源IP地址	源物理地址
安全	数据链路层	ARP请求风暴 (主机的MAC为 88:CT:5D:3B:33:39, IP地址为 172.27.112.37)	172.27.112.37	B8:CT:5D:3B:33:39	
安全	数据链路层	ARP请求风暴 (主机的MAC为 88:CT:5D:3B:33:39, IP地址为 172.27.112.37)	172.27.112.37	B8:CT:5D:3B:33:39	
安全	数据链路层	ARP请求风暴 (主机的MAC为 88:CT:5D:3B:33:39, IP地址为 172.27.112.37)	172.27.112.37	B8:CT:5D:3B:33:39	
安全	数据链路层	ARP请求风暴 (主机的MAC为 88:CT:5D:3B:33:39, IP地址为 172.27.112.37)	172.27.112.37	B8:CT:5D:3B:33:39	
安全	数据链路层	ARP请求风暴 (主机的MAC为 88:CT:5D:3B:33:39, IP地址为 172.27.112.37)	172.27.112.37	B8:CT:5D:3B:33:39	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:17:C4:78:D1:9C, IP地址为 172.27.112.2)	172.27.112.2	00:17:C4:78:D1:9C	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:17:C4:78:D1:9C, IP地址为 172.27.112.2)	172.27.112.2	00:17:C4:78:D1:9C	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:17:C4:78:D1:9C, IP地址为 172.27.112.2)	172.27.112.2	00:17:C4:78:D1:9C	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:17:C4:78:D1:9C, IP地址为 172.27.112.2)	172.27.112.2	00:17:C4:78:D1:9C	
安全	数据链路层	ARP请求风暴 (主机的MAC为 4C:0F:6E:DE:7E:9A, IP地址为 172.27.112.10)	172.27.112.10	4C:0F:6E:DE:7E:9A	
安全	数据链路层	ARP请求风暴 (主机的MAC为 4C:0F:6E:DE:7E:9A, IP地址为 172.27.112.10)	172.27.112.10	4C:0F:6E:DE:7E:9A	
安全	数据链路层	ARP请求风暴 (主机的MAC为 4C:0F:6E:DE:7E:9A, IP地址为 172.27.112.10)	172.27.112.10	4C:0F:6E:DE:7E:9A	
安全	数据链路层	ARP请求风暴 (主机的MAC为 4C:0F:6E:DE:7E:9A, IP地址为 172.27.112.10)	172.27.112.10	4C:0F:6E:DE:7E:9A	
安全	数据链路层	ARP请求风暴 (主机的MAC为 4C:0F:6E:DE:7E:9A, IP地址为 172.27.112.10)	172.27.112.10	4C:0F:6E:DE:7E:9A	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:16:17:B7:55:3F, IP地址为 172.27.112.17)	172.27.112.17	00:16:17:B7:55:3F	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:16:17:B7:55:3F, IP地址为 172.27.112.17)	172.27.112.17	00:16:17:B7:55:3F	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:16:17:B7:55:3F, IP地址为 172.27.112.17)	172.27.112.17	00:16:17:B7:55:3F	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:16:17:B7:55:3F, IP地址为 172.27.112.17)	172.27.112.17	00:16:17:B7:55:3F	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:16:17:B7:55:3F, IP地址为 172.27.112.17)	172.27.112.17	00:16:17:B7:55:3F	
安全	数据链路层	ARP请求风暴 (主机的MAC为 00:16:17:B7:55:3F, IP地址为 172.27.112.17)	172.27.112.17	00:16:17:B7:55:3F	

时间差	源	目标	协议	大小	解码字段	摘要
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000344	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000501	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000503	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000499	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000503	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000502	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000500	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17
0.000000	00:16:17.B7:55:3F	FF:FF:FF:FF:FF:FF	ARP	64		谁是 172.127.112.1? 告诉 172.27.112.17

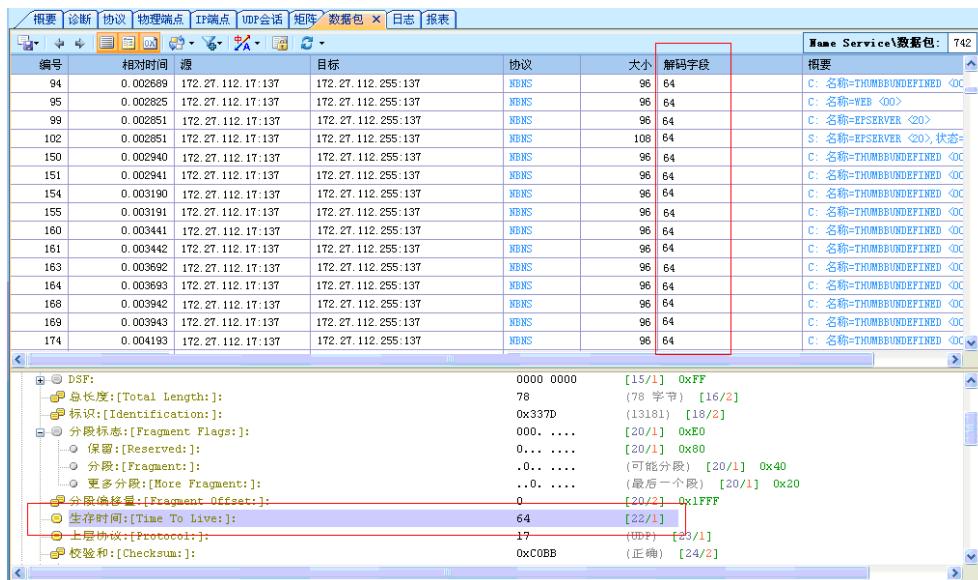
编号	时间差	源	目标	协议	大小	解码字段	摘要
1	0.000001	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
2	0.000001	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
6	0.000249	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
7	0.000001	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
10	0.000018	172.27.112.17:137	172.27.112.255:137	NBNS	114		C: 名称=LT <0>
11	0.000000	172.27.112.17:137	172.27.112.255:137	NBNS	114		C: 名称=A043 <0>
14	0.000232	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
15	0.000001	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
20	0.00250	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
21	0.000000	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
24	0.000250	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
25	0.000001	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
28	0.000250	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
29	0.000000	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
32	0.000118	172.27.112.17:137	172.27.112.255:137	NBNS	114		C: 名称=LT <0>
33	0.000000	172.27.112.17:137	172.27.112.255:137	NBNS	114		C: 名称=B045 <0>
35	0.000133	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
36	0.000000	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
41	0.000250	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
42	0.000001	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
46	0.000250	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
47	0.000000	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>
54	0.000253	172.27.112.17:137	172.27.112.255:137	NBNS	96		C: 名称=THUMBUNDEFINED <0>

编号	相对时间	源	目标	协议	大小	解码字段	摘要
94	0.002699	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
95	0.002825	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=WEB <0>
99	0.002851	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=EP SERVER <20>
102	0.002851	172.27.112.17:137	172.27.112.255:137	NBNS	108	0x337D	S: 名称=EP SERVER <20>, 状态=
150	0.002940	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
151	0.002941	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
154	0.003190	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
155	0.003191	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
160	0.003441	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
161	0.003442	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
183	0.003692	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
164	0.003693	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
168	0.003942	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
169	0.003943	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>
174	0.004193	172.27.112.17:137	172.27.112.255:137	NBNS	96	0x337D	C: 名称=THUMBUNDEFINED <0>

展开地址:[Source Address:] 00:16:17:B1:4D:E0 [6/6]
 展开协议类型:[Protocol:] 0x0800 (Internet IP(IPv4)) [12/2] [14/20]

展开版本:[Version:] 4 [14/1] 0xF0
 展开头部长度:[Header Length:] 5 (20 字节) [14/1] 0x0F
 展开总长度:[Total Length:] 0000 0000 [15/1] 0xFF
 展开生存时间:[TTL:] 78 (78 字节) [16/] [16/]

展开标识:[Identification:] 0x337D [13181] [18/2]
 展开分段标志:[Fragment Flags:] 000. [20/1] 0xE0
 展开保留:[Reserved:] 0... [20/1] 0x80
 展开片段:[Fragment:] .0... (可能分段) [20/1] 0x40



3. 交换机配置

经过检查交换机上的生成树 spanning-tree 协议没有启用，在交换机存在物理环路的情况下没有阻塞其中 1 条链路，从而导致数据包被重复转发。

8.6. 故障解决

交换机上启用生成树 spanning-tree 协议：4 台 H3C 5600 交换机: stp enable

验证：再将 2 个存储控制器的固件专用管理口同时连接，没有网络风暴现象出现。

8.7. 案例自评

- 故障简单，但是处理过程中重现该故障具有一定的操作难度；
- 在没有原始数据包的情况下，查找产生瞬时网络风暴思路处理过程清晰；
- 物理环路特征总结：IP ID相同、TTL值相同、伴随arp请求风暴报警；

9. TCP异常连接分析案例

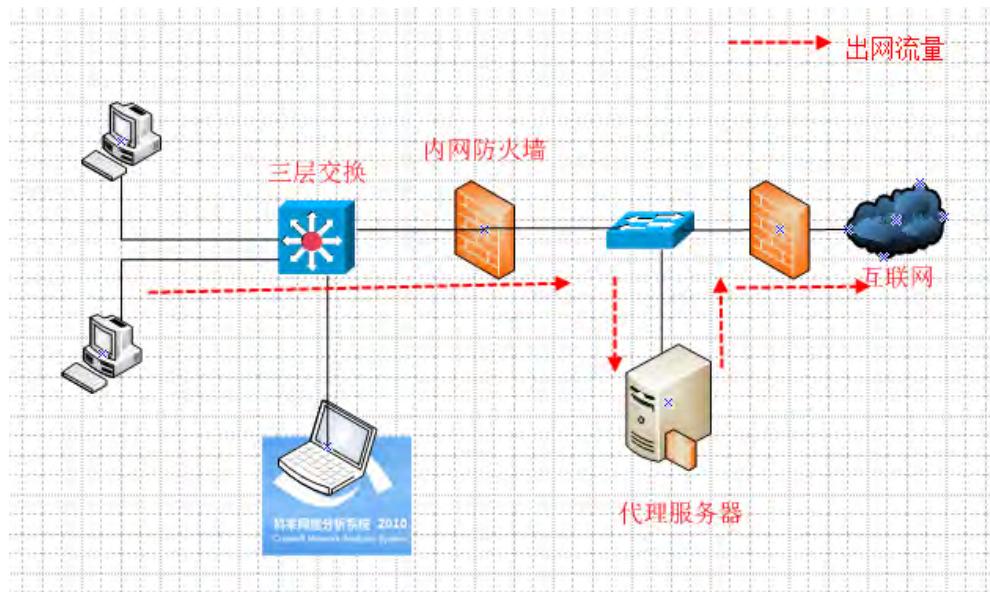
9.1. 故障现象描述

1. 故障现象描述

一大型国企总部的某区域网络，在内网防火墙上发现大量 TCP 半连接，疑似 DOS 攻击，暂时未对内网服务器造成破坏性问题。由于大量 TCP 产生原因未明，调用了应急人员来解决。

2. 基本环境描述

基本网络拓扑如下：



图表 1 网络基本拓扑

内网用户访问互联网流量，必须通过代理服务器中转访问。流量走向如上图红色标出线条，用户访问代理服务器需要经过内网三层交换、内网防火墙，其中在三层交换上旁路了大容量存储的网络分析设备，实现了对流量的回溯分析能力。就是在内网的这台防火墙上发现了大量的 TCP 连接。

9.2. 分析方案设计

1. 分析目标

找出产生大量 TCP 的原因，定位到具体主机。需要重点分析的是 TCP 会话部分。

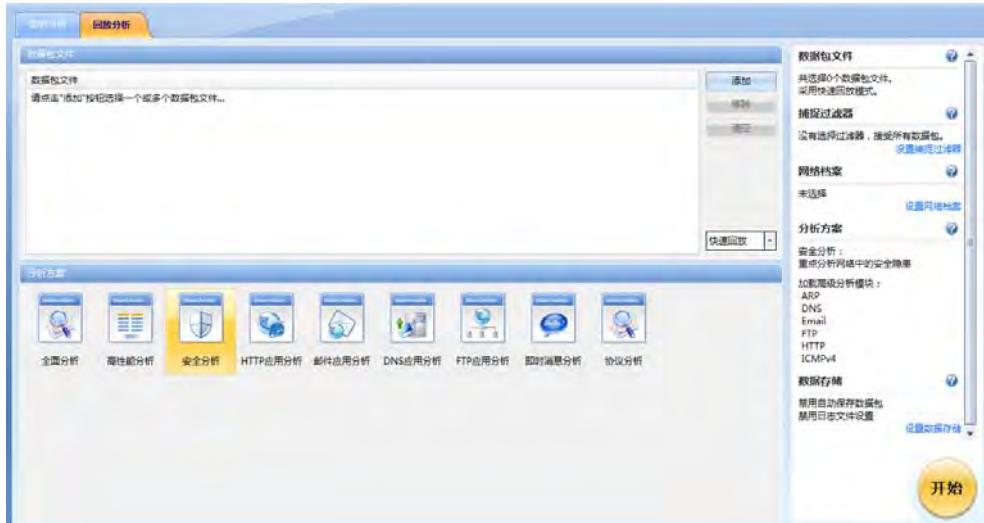
2. 分析设备部署

由于网络中已经部署了回溯的流量分析设备，只需要把故障出现时的流量分析即可，通过科来的网络分析系统 2010 回放已保存的数据包。

9.3. 分析情况

1. 基本流量分析

首先利用科来网络分析系统的安全分析方案，对网络中的 DOS 攻击、蠕虫病毒、TCP 扫描等进行智能分析。



图表 2

此数据包共选取了 3 分 31s 的数据包，总流量为 2.389GB。

流量统计	字节数	数据包数	利用率	每秒位数	每秒包数
总流量	2.389 GB	4,796,920	100.000%	960.502 Mbps	219,956
广播流量	0 B	0	0.000%	0 bps	0
多播流量	7.133 KB	66	0.004%	3.712 Kbps	4

图表 3 总流量

数据包大小分布分析，65-127 的小包为 2.683.048 个，明显较多。

数据包大小分布	字节数	数据包数	利用率	每秒位数	每秒包数
<=64	0 B	0	0.000%	0 bps	0
65-127	183.937 MB	2,683,048	71.255%	71.255 Mbps	124,034
128-255	52.151 MB	282,181	19.775%	19.775 Mbps	12,725
256-511	36.921 MB	101,340	12.321%	12.321 Mbps	3,996
512-1023	228.837 MB	366,253	65.928%	65.928 Mbps	12,915
1024-1517	293.517 MB	223,773	100.000%	137.074 Mbps	12,420
>=1518	1.612 GB	1,140,325	100.000%	654.149 Mbps	53,866

图表 4 数据包大小分布

TCP 会话分析，TCP 连接数过多，并且存在大量 TCP 复位包。

TCP统计	数量
TCP同步发送	305,419
TCP同步确认发送	216,137
TCP结束连接发送	364,327
TCP复位发送	60,838

图表 5 TCP 统计

DNS 分析，查询数量明显大于回应数据，有大量 DNS 请求没有得到回应。

DNS分析		数量
DNS查询		5,893
DNS回应		686

图表 6 DNS 分析

安全问题诊断：存在疑似 DOS 攻击问题，需针对性分析。

安全分析统计		数量
疑似蠕虫病毒发生地址		191
疑似发起DoS攻击地址		56
疑似受到DoS攻击地址		6
可疑会话发生地址		0
TCP端口扫描发生地址		0
ARP攻击发生地址		0

图表 7 安全分析统计

2. TCP 异常分析

在对基本流量分析后，对网络中主机 TCP 同步发送数据进行排名分析。

安全分析									
名字	字节	数据包	接收数据包	TCP同步发送	TCP同步确认接收	发送数据包	发送/接收(数据包)比率	TCP会话	
10.78.238.228	2,327 MB	34,687	11,178	3,367	2,556	23,509	2.10	3,625	
10.78.178.87	1,520 MB	22,858	9,748	3,291	3,265	13,110	1.34	3,320	
10.78.178.15	1,467 MB	22,065	9,427	3,156	3,146	12,638	1.34	3,199	
10.59.4.12	1,189 MB	17,886	7,631	2,566	2,541	10,255	1.34	2,584	
10.78.176.171	1,130 MB	16,998	7,272	2,418	2,432	9,726	1.34	2,461	
10.78.80.210	1,011 MB	15,204	6,502	2,173	2,166	8,702	1.34	2,201	
10.78.59.219	1,018,981 KB	14,965	6,391	2,153	2,139	8,574	1.34	2,174	
10.78.65.2	948,065 KB	13,936	5,376	2,056	1,829	8,560	1.59	2,208	
10.59.19.166	820,089 KB	11,377	3,263	2,006	865	8,114	2.49	2,206	
10.59.10.93	1,050 MB	15,839	7,907	1,986	1,979	7,932	1.00	2,002	
10.78.210.41	1,020,200 KB	12,405	6,762	1,021	1,010	7,722	1.24	1,023	

IP会话									
节点1->	<-节点2	数据包	字节	协议	持续时间	字节->	<-字节	数据包	
10.78.238.228:3635	10.22.16.20:8080	12	834 B	HTTP Proxy	0	556 B	278 B		10.78.238
10.78.238.228:1192	220.181.125.208:80	1	74 B	HTTP	0	74 B	0 B		
10.78.238.228:24796	10.22.16.20:8080	10	698 B	HTTP Proxy	0	420 B	278 B		
10.78.238.228:24797	10.22.16.20:8080	10	776 B	HTTP Proxy	0	468 B	308 B		
10.78.238.228:50987	10.22.16.20:8080	5	352 B	HTTP Proxy	0	210 B	142 B		
10.78.238.228:3656	10.22.16.20:8080	12	834 B	HTTP Proxy	0	556 B	278 B		
10.78.238.228:2069	10.22.16.20:8080	3	247 B	HTTP Proxy	0	179 B	68 B		
10.78.238.228:3769	10.22.16.20:8080	11	766 B	HTTP Proxy	0	420 B	346 B		
10.78.238.228:3657	10.22.16.20:8080	12	834 B	HTTP Proxy	0	556 B	278 B		

图表 8 TCP 同步发送排名

对 TCP 连接较高的 IP：10.78.178.87 进行针对性分析。

安全分析									
名字	字节	数据包	接收数据包	TCP同步发送	TCP同步确认接收	发送数据包	发送/接收(数据包)比率	TCP会话	
10.78.238.228	2,327 MB	34,687	11,178	3,367	2,556	23,509	2.10	3,625	
10.78.178.87	1,520 MB	22,858	9,748	3,291	3,265	13,110	1.34	3,320	
10.78.178.15	1,467 MB	22,065	9,427	3,156	3,146	12,638	1.34	3,199	
10.59.4.12	1,189 MB	17,886	7,631	2,566	2,541	10,255	1.34	2,584	
10.78.176.171	1,130 MB	16,998	7,272	2,418	2,432	9,726	1.34	2,461	
10.78.80.210	1,011 MB	15,204	6,502	2,173	2,166	8,702	1.34	2,201	
10.78.59.219	1,018,981 KB	14,965	6,391	2,153	2,139	8,574	1.34	2,174	

IP会话									
节点1->	<-节点2	数据包	字节	持续时间	字节->	<-字节	数据包	数据包->	<- 数据包
10.78.178.87	10.22.16.20	14	1,087 MB	636,574 KB	476,727 KB	16,351	9,378	6,973	10.78.178.87

图表 9

在 IP 会话中只有跟 10.22.16.20 的会话，10.22.16.20 为其中的一台代理服务器。

继续分析 TCP 会话，其中本地使用的端口从 9135 进行递增，并且每个会话的数据包个数都为 7 个，

存在 TCP DOS 攻击嫌疑，也验证了在安全分析的统计结果。

节点1->	<-节点2	数据包	字节	协议	持续时间	字节->
10.78.178.87:9135	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9136	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9137	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9138	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9139	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9140	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9141	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9142	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9143	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9144	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9145	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9146	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B
10.78.178.87:9147	10.22.16.20:8080	7	488 B	HTTP Proxy	0	278 B

图表 10

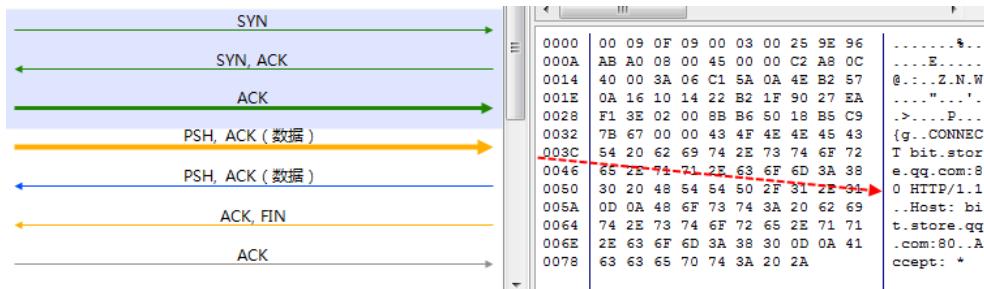
选中任意 TCP 会话，对其进行 TCP 流分析。会话中都包含了这样 9 个数据包，如下图：



图表 11 TCP 时序图分析

其中包括前三个包为 tcp 三次握手建立连接，最后四个包用于关闭连接，中间两个数据包是应用层数据，包含了一个请求和一个响应。

定位到数据包解码中，查看应用层的详细信息，如下图：



图表 12 应用层请求

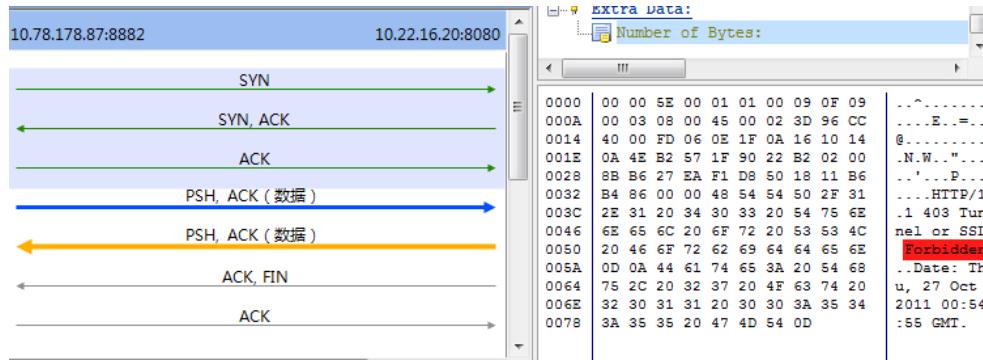
在客户端发起的应用请求为一 CONNECT 类型的 HTTP 请求数据，请求内容为：

g..CONNECT bit.store.qq.com:80 HTTP/1.1

Host: bit.store.qq.com:80

Accept: *

服务器对此请求直接给予 **Forbidden** 响应，如下图：



图表 13 应用层响应

应用层响应内容，如下：

HTTP/1.1 403 Tunnel or SSL Forbidden

Date: Thu, 27 Oct 2011 00:54:55 GMT

对如上的 QQ 请求进行了解发现，此请求为开启腾讯 QQ 旋风后，发出的请求。代理服务器由于不支持这种 connect 方式的 http 请求，收到后则直接给予拒绝。而关于此 URL 请求，最近在网上已经有多例导致网络拥塞现象的故障。

对网络进一步了解后，通过 TTL 值分析，代理服务器与捕获点位置正好经过两跳，的确是从代理服务器发出。

编号	绝对时间	源	目标	协议	大小	解码字段
24700	08:54:56.169181	10.78.178.87:8881	10.22.16.20:8080	HTTP Proxy	68	58
24701	08:54:56.169181	10.22.16.20:8080	10.78.178.87:8881	HTTP Proxy	68	253
24702	08:54:56.171182	10.78.178.87:8882	10.22.16.20:8080	HTTP Proxy	74	58
24703	08:54:56.171182	10.22.16.20:8080	10.78.178.87:8882	HTTP Proxy	74	253
24704	08:54:56.171182	10.78.178.87:8882	10.22.16.20:8080	HTTP Proxy	68	58
24705	08:54:56.172181	10.78.178.87:8882	10.22.16.20:8080	HTTP Proxy	212	58
24706	08:54:56.173181	10.22.16.20:8080	10.78.178.87:8882	HTTP Proxy	591	253
24707	08:54:56.173181	10.22.16.20:8080	10.78.178.87:8882	HTTP Proxy	68	253
24708	08:54:56.174182	10.78.178.87:8882	10.22.16.20:8080	HTTP Proxy	68	58
24709	08:54:56.174182	10.78.178.87:8882	10.22.16.20:8080	HTTP Proxy	68	58

图表 14 TTI 值分析

客户端请求被拒绝后，却并未停止发起连接，在 QQ 旋风运行期间一直保持高速的 TCP 请求状态。

绝对时间	时间差	源	目标	协议	大小	源端口	解码字段
08:54:37.378914		10.78.178.87:6065	10.22.16.20:8080	HTTP Proxy	74	6065	52
08:54:37.379914	0.001000	10.22.16.20:8080	10.78.178.87:6065	HTTP Proxy	74	8080	52
08:54:37.379914	0.000000	10.78.178.87:6065	10.22.16.20:8080	HTTP Proxy	68	6065	40
08:54:37.380915	0.001001	10.78.178.87:6065	10.22.16.20:8080	HTTP Proxy	212	6065	194
08:54:37.380915	0.000000	10.22.16.20:8080	10.78.178.87:6065	HTTP Proxy	591	8080	573
08:54:37.380915	0.000000	10.22.16.20:8080	10.78.178.87:6065	HTTP Proxy	68	8080	40
08:54:37.381914	0.000999	10.78.178.87:6065	10.22.16.20:8080	HTTP Proxy	68	6065	40
08:54:37.381914	0.000000	10.78.178.87:6065	10.22.16.20:8080	HTTP Proxy	68	6065	40
08:54:37.382914	0.001000	10.22.16.20:8080	10.78.178.87:6065	HTTP Proxy	68	8080	40
08:54:37.384914	0.002000	10.78.178.87:6066	10.22.16.20:8080	HTTP Proxy	74	6066	52
08:54:37.384914	0.000000	10.22.16.20:8080	10.78.178.87:6066	HTTP Proxy	74	8080	52
08:54:37.385914	0.001000	10.78.178.87:6066	10.22.16.20:8080	HTTP Proxy	68	6066	40
08:54:37.385914	0.000000	10.78.178.87:6066	10.22.16.20:8080	HTTP Proxy	212	6066	194
08:54:37.386915	0.001001	10.22.16.20:8080	10.78.178.87:6066	HTTP Proxy	591	8080	573
08:54:37.386915	0.000000	10.22.16.20:8080	10.78.178.87:6066	HTTP Proxy	68	8080	40
08:54:37.387914	0.000999	10.78.178.87:6066	10.22.16.20:8080	HTTP Proxy	68	6066	40
08:54:37.388914	0.001000	10.78.178.87:6066	10.22.16.20:8080	HTTP Proxy	68	6066	40
08:54:37.388914	0.000000	10.22.16.20:8080	10.78.178.87:6066	HTTP Proxy	68	8080	40
08:54:37.390914	0.002000	10.78.178.87:6067	10.22.16.20:8080	HTTP Proxy	74	6067	52

图表 15 新连接间隔分析

通过上图可以清晰的看到，客户端在的第一个 TCP 会话结束后，2ms 后发起了第二次连接，并且在第二次连接被拒绝后，仍以 2ms 的间隔发起了新一次的连接。红色标示部分 0.002s 即 2ms。

而在短短的 3 分 31s 内，共发起了 2966 个 TCP 连接，也就是每秒 14 个 TCP 连接。

名字	字节	数据包	接收数据包	发送数据包	包收发比	TCP会话	TCP同步发送 ▾
10.78.238.228	2.175 MB	32,430	10,505	21,925	2.09	3,370	3,133
10.78.178.87	1.370 MB	20,610	8,795	11,815	1.34	2,990	2,966
10.78.178.15	1.344 MB	20,216	8,636	11,580	1.34	2,930	2,889
10.59.4.12	1.095 MB	16,473	7,031	9,442	1.34	2,379	2,365
10.78.176.171	1.039 MB	15,618	6,680	8,938	1.34	2,260	2,220
10.78.80.210	940.706 KB	13,815	5,905	7,910	1.34	1,998	1,975

图表 16 TCP 连接数

而在概要中我们也可以看到内网主机数有 40744 台之多，如果大量内网主机同时在线运行 QQ 旋风，则会造成灾难性故障。

□ 地址统计	数量
物理地址数	8
IP地址数	53,490
本地IP地址数	40,744
远程IP地址数	12,746

图表 17 地址统计

这种连接则会一直持续下去，直到用户关掉 QQ 旋风程序为止。（一旦把 QQ 旋风最小化，运行到主机关机，那后果则会更严重）

而目前由于代理服务器前部署了负载均衡设备，对内部服务器暂未造成影响，而在服务器之前的防火墙则吃不下了，在短时间内则出现了大量 TCP 会话，影响了防火墙的正常运行。

9.4. 分析结论

当前大量 TCP 连接爆发，主要为 QQ 旋风程序发起，并且在 HTTP 的 CONNECT 请求被拒绝后，仍会以 2ms 的间隔重现发起新的连接。网络规模庞大，QQ 旋风同时在线的主机数较多，导致了类似 DOS 效果的攻击现象。由于服务器区部署了负载均衡设备，代理服务器暂未受到影响，而在内网的防火墙则明显性能下降，需要处理大量 TCP 会话。

建议：在防火墙或上网行为管理设备上对 QQ 旋风进行限制，或在代理服务器上做相应调整；另外建议跟腾讯方联系，修改其 QQ 旋风的连接机制。

10. 记录两次断网的分析过程

10.1. DHCP服务器

10.8 国庆后的第一个工作日上午，某用户部分链路出现断网的情况，通过远程连接，科来工程师对该网络进行了分析。

名称	字节数	数据包	每秒流量	每秒数据包
未知TCP应用	115.9 GB	436,927,296	5.97 MB	8,091
HTTP	62.69 GB	114,120,478	1.57 MB	2,113
HTTP Proxy	42.86 GB	80,871,560	832.28 KB	1,497
BOOTP	23.10 GB	72,201,994	448.50 KB	1,337
MSRDP	20.09 GB	44,599,988	390.17 KB	825
CIFS	13.85 GB	38,769,974	268.87 KB	717
Oracle	5.15 GB	14,117,766	95.99 KB	261
未知UDP应用	3.91 GB	11,296,533	76.00 KB	209
HTTPS	1.73 GB	3,930,624	32.54 KB	72
FTP	1.07 GB	1,123,752	20.70 KB	20
MGCP	484.60 MB	458,705	8.81 KB	8
NetBIOS Name Service	192.62 MB	1,979,861	3.63 KB	36
DNS	152.48 MB	1,287,937	2.89 KB	23
NetBIOS Session Service	130.73 MB	1,082,071	2.48 KB	20
HSRP	111.16 MB	1,774,939	2.12 KB	32
POP3	85.91 MB	110,323	1.69 KB	2
LDAP	80.29 MB	253,777	1.52 KB	4
NetBIOS Datagram Service	71.45 MB	307,632	1.35 KB	5
Kerberos	68.52 MB	156,599	1.30 KB	2
SNMP	65.18 MB	356,807	1.24 KB	6

通过对网络应用的观察发现，BOOTP 占据了很大一部分的流量。BOOTP 是动态分配给工作站 IP 的协议，通过横向对比发现，平时 BOOTP 的流量都很小。但是在 10.7 晚上 7 点左右，突然开始了流量的爆发。于是我们对 BOOTP 协议进行了挖掘：

地址	字节数	数据包	每秒流量	每秒数据包
18.3	23.03 GB	72,012,998	479.21 KB	1
106.202	22.92 GB	71,664,757	476.80 KB	1
30.9	53.77 MB	158,592	1.09 KB	
18.112	31.42 MB	91,198	653 B	
255.255.255	10.14 MB	30,072	210 B	
141.254	7.07 MB	21,006	147 B	
26.43	5.01 MB	15,020	104 B	
26.251	4.75 MB	13,870	98 B	
26.252	4.75 MB	13,862	98 B	
121.254	4.19 MB	12,200	87 B	
115.252	3.47 MB	10,505	72 B	
115.253	3.46 MB	10,485	72 B	
106.253	3.45 MB	10,161	71 B	
106.252	3.42 MB	10,077	71 B	
131.254	2.95 MB	8,679	61 B	
31.21	2.86 MB	8,678	59 B	
112.253	2.59 MB	7,860	53 B	
112.252	2.54 MB	7,686	52 B	
38.252	2.36 MB	6,848	49 B	



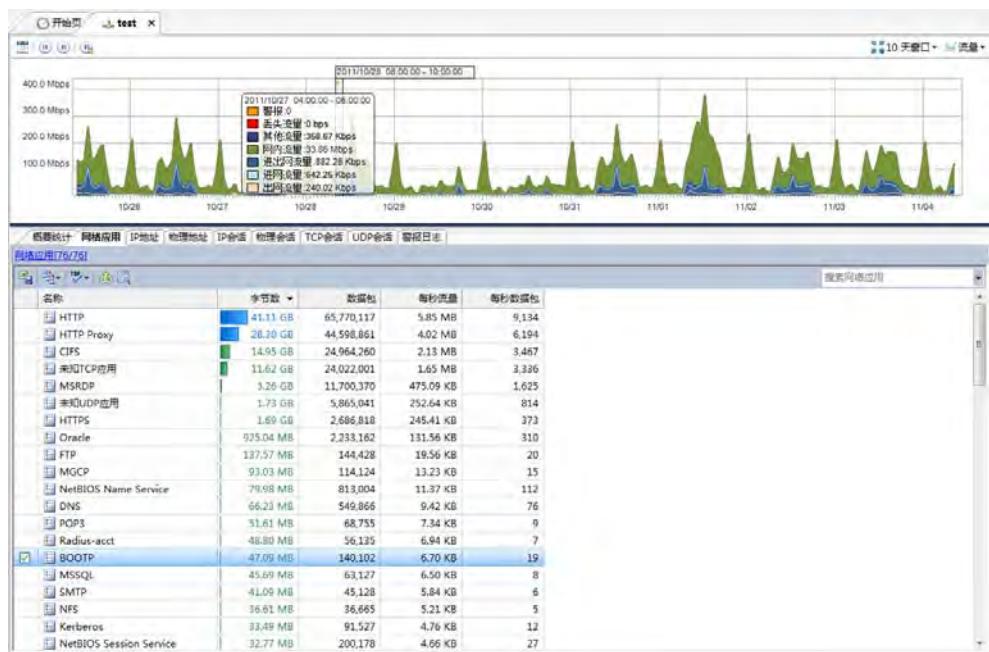
看到，X.X.18.3 和 X.X.106.202 之间的通信流量占了绝大部分。那么他们之间的行为究竟是什么呢？我们可以通过将其数据包下载下来进行分析。

IP会话		TCP会话	UDP会话	搜索IP会话	
地址1 ->	<- 地址2		字节数	数据包	
<input checked="" type="checkbox"/> 192.168.1.106.202	192.168.1.18.3		22.92 GB	71,664,757	
	挖掘	192.168.1.18.3	53.77 MB	158,586	
	自定义列	192.168.1.18.3	4.65 MB	13,558	
	复制	Ctrl+C 192.168.1.18.3	3.52 MB	10,466	
	复制列	192.168.1.18.3	3.32 MB	9,685	
	下载数据包到文件	0.26.43	2.90 MB	8,788	
	下载并分析数据包	192.168.1.18.3	2.37 MB	6,917	
	过滤器	192.168.1.18.3	2.16 MB	6,544	

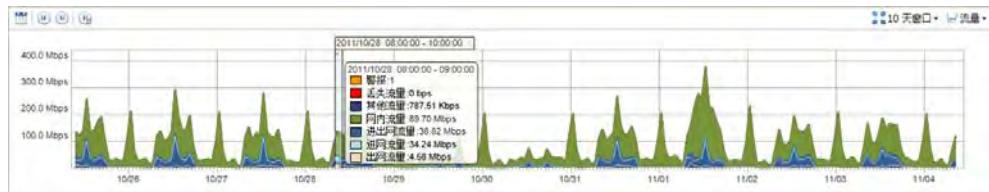
分析其数据包中详细的信息：

在深入的分析中，我们发现 106.202 这台主机一直在向 18.3 发送请求，几乎每 1 毫秒都有一
次请求。可以确定是这个工作站 BOOTP 请求机制出现了问题。

编号	绝对时间	时间间隔	源	目标	协议	大小	报文字节	摘要
27	19:04:58.548734	0.000170	18.3.67	106.202.68	DHCP	348	0	0.0.0.0.106.202
28	19:04:58.349927	0.001193	106.202.68	18.3.67	DHCP	341	106.202	0.0.0.0.106.202
29	19:04:58.350093	0.000166	18.3.67	106.202.68	DHCP	348	0	0.0.0.0.106.202
40	19:04:58.351408	0.001213	106.202.68	18.3.67	DHCP	341	106.202	0.0.0.0.106.202
41	19:04:58.311578	0.000170	18.3.67	106.202.68	DHCP	348	0	0.0.0.0.106.202
42	19:04:58.312767	0.001186	106.202.68	18.3.67	DHCP	341	106.202	0.0.0.0.106.202
43	19:04:58.352962	0.000165	18.3.67	106.202.68	DHCP	348	0	0.0.0.0.106.202
44	19:04:58.353879	0.000947	106.202.68	18.3.67	DHCP	341	106.202	0.0.0.0.106.202
45	19:04:58.354044	0.000165	18.3.67	106.202.68	DHCP	348	0	0.0.0.0.106.202
46	19:04:58.355238	0.001194	106.202.68	18.3.67	DHCP	341	106.202	0.0.0.0.106.202
47	19:04:58.348444	0.000933	18.3.67	106.202.68	DHCP	348	0	0.0.0.0.106.202
操作: 展开/折叠 (Hardware Length(1))								
操作: 展开/折叠 (Length(1))								
操作: 展开/折叠 (Transmission ID(1))								
操作: 展开/折叠 (Type(1))								
操作: 展开/折叠 (硬件地址 (16 bit) (LL Address (12 byte)))								
操作: 展开/折叠 (广播地址 (16 bit) (LL Broadcast Address (12 byte)))								
操作: 展开/折叠 (逻辑地址 (16 bit) (IP Address (4 byte)))								
操作: 展开/折叠 (广播地址 (4 byte) (IP Broadcast Address (4 byte)))								
操作: 展开/折叠 (源端口 (16 bit))								
操作: 展开/折叠 (目的端口 (16 bit))								
操作: 展开/折叠 (源 MAC 地址 (6 byte))								
操作: 展开/折叠 (目的 MAC 地址 (6 byte))								
操作: 展开/折叠 (源 IP 地址 (4 byte))								
操作: 展开/折叠 (目的 IP 地址 (4 byte))								
操作: 展开/折叠 (TOS (3 bit))								
操作: 展开/折叠 (T1 (1 bit))								
操作: 展开/折叠 (T2 (1 bit))								
操作: 展开/折叠 (T3 (1 bit))								
操作: 展开/折叠 (T4 (1 bit))								
操作: 展开/折叠 (T5 (1 bit))								
操作: 展开/折叠 (T6 (1 bit))								
操作: 展开/折叠 (T7 (1 bit))								
操作: 展开/折叠 (T8 (1 bit))								
操作: 展开/折叠 (T9 (1 bit))								
操作: 展开/折叠 (T10 (1 bit))								
操作: 展开/折叠 (T11 (1 bit))								
操作: 展开/折叠 (T12 (1 bit))								
操作: 展开/折叠 (T13 (1 bit))								
操作: 展开/折叠 (T14 (1 bit))								
操作: 展开/折叠 (T15 (1 bit))								
操作: 展开/折叠 (T16 (1 bit))								
操作: 展开/折叠 (T17 (1 bit))								
操作: 展开/折叠 (T18 (1 bit))								
操作: 展开/折叠 (T19 (1 bit))								
操作: 展开/折叠 (T20 (1 bit))								
操作: 展开/折叠 (T21 (1 bit))								
操作: 展开/折叠 (T22 (1 bit))								
操作: 展开/折叠 (T23 (1 bit))								
操作: 展开/折叠 (T24 (1 bit))								
操作: 展开/折叠 (T25 (1 bit))								
操作: 展开/折叠 (T26 (1 bit))								
操作: 展开/折叠 (T27 (1 bit))								
操作: 展开/折叠 (T28 (1 bit))								
操作: 展开/折叠 (T29 (1 bit))								
操作: 展开/折叠 (T30 (1 bit))								
操作: 展开/折叠 (T31 (1 bit))								
操作: 展开/折叠 (T32 (1 bit))								
操作: 展开/折叠 (T33 (1 bit))								
操作: 展开/折叠 (T34 (1 bit))								
操作: 展开/折叠 (T35 (1 bit))								
操作: 展开/折叠 (T36 (1 bit))								
操作: 展开/折叠 (T37 (1 bit))								
操作: 展开/折叠 (T38 (1 bit))								
操作: 展开/折叠 (T39 (1 bit))								
操作: 展开/折叠 (T40 (1 bit))								
操作: 展开/折叠 (T41 (1 bit))								
操作: 展开/折叠 (T42 (1 bit))								
操作: 展开/折叠 (T43 (1 bit))								
操作: 展开/折叠 (T44 (1 bit))								
操作: 展开/折叠 (T45 (1 bit))								
操作: 展开/折叠 (T46 (1 bit))								
操作: 展开/折叠 (T47 (1 bit))								
操作: 展开/折叠 (T48 (1 bit))								
操作: 展开/折叠 (T49 (1 bit))								
操作: 展开/折叠 (T50 (1 bit))								
操作: 展开/折叠 (T51 (1 bit))								
操作: 展开/折叠 (T52 (1 bit))								
操作: 展开/折叠 (T53 (1 bit))								
操作: 展开/折叠 (T54 (1 bit))								
操作: 展开/折叠 (T55 (1 bit))								
操作: 展开/折叠 (T56 (1 bit))								
操作: 展开/折叠 (T57 (1 bit))								
操作: 展开/折叠 (T58 (1 bit))								
操作: 展开/折叠 (T59 (1 bit))								
操作: 展开/折叠 (T60 (1 bit))								
操作: 展开/折叠 (T61 (1 bit))								
操作: 展开/折叠 (T62 (1 bit))								
操作: 展开/折叠 (T63 (1 bit))								
操作: 展开/折叠 (T64 (1 bit))								
操作: 展开/折叠 (T65 (1 bit))								
操作: 展开/折叠 (T66 (1 bit))								
操作: 展开/折叠 (T67 (1 bit))								
操作: 展开/折叠 (T68 (1 bit))								
操作: 展开/折叠 (T69 (1 bit))								
操作: 展开/折叠 (T70 (1 bit))								
操作: 展开/折叠 (T71 (1 bit))								
操作: 展开/折叠 (T72 (1 bit))								
操作: 展开/折叠 (T73 (1 bit))								
操作: 展开/折叠 (T74 (1 bit))								
操作: 展开/折叠 (T75 (1 bit))								
操作: 展开/折叠 (T76 (1 bit))								
操作: 展开/折叠 (T77 (1 bit))								
操作: 展开/折叠 (T78 (1 bit))								
操作: 展开/折叠 (T79 (1 bit))								
操作: 展开/折叠 (T80 (1 bit))								
操作: 展开/折叠 (T81 (1 bit))								
操作: 展开/折叠 (T82 (1 bit))								
操作: 展开/折叠 (T83 (1 bit))								
操作: 展开/折叠 (T84 (1 bit))								
操作: 展开/折叠 (T85 (1 bit))								
操作: 展开/折叠 (T86 (1 bit))								
操作: 展开/折叠 (T87 (1 bit))								
操作: 展开/折叠 (T88 (1 bit))								
操作: 展开/折叠 (T89 (1 bit))								



可以看到，在8点到10点之间，BOOTP的流量只有47MB，在正常范围之内，那么究竟是什么原因导致的这次断网的发生？



我们在趋势图中发现了警报，

This screenshot shows the '报警' (Alarms) configuration table. It lists various monitoring rules with their status (启用 - Enabled/Disabled), name, entry conditions, entry actions, release conditions, and release actions.

启用	名称	进入条件	进入动作	解除条件	解除动作
<input type="checkbox"/>	总利用率	值:>=80%;持续:5秒		值:<60%;持续:5秒	
<input type="checkbox"/>	出网利用率	值:>=80%;持续:5秒		值:<60%;持续:5秒	
<input type="checkbox"/>	进网利用率	值:>=80%;持续:5秒		值:<60%;持续:5秒	
<input type="checkbox"/>	总每秒包数	值:>=150000;持续:5秒		值:<120000;持续:5秒	
<input type="checkbox"/>	出网每秒包数	值:>=80000;持续:5秒		值:<60000;持续:5秒	
<input type="checkbox"/>	进网每秒包数	值:>=80000;持续:5秒		值:<60000;持续:5秒	
<input checked="" type="checkbox"/>	TCP同步包	值:>=1000;持续:5秒		值:<6;持续:5秒	
<input type="checkbox"/>	TCP同步重置包	值:>=1;持续:5秒		值:<0;持续:5秒	

在测试的时候，也对 TCP 同步包设置了相关的警报，一旦同步包过多，那么就会有报警的产生。

那么，如何判断是不是出现了 SYN FLOOD 攻击呢？有一个简单的方法，就是观察 IP 的数量。



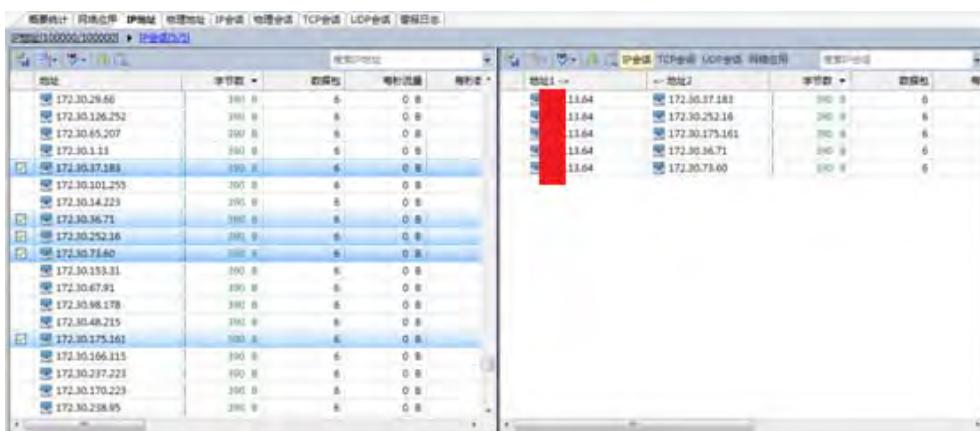
地址	字节数	数据包	每秒流量	每秒数据包数	发送字节数	接收字节数	发包	接收	发送每秒字节数	接收每秒字节数
109.500.10.20	20,091,190	4,264 KB	6,820	216.4 KB	4,34 GB	27,820,243	21,178,947	2,053 KB	1,653 KB	912.6
109.500.10.20	45,529,164	3,87 MB	6,295	1,94 GB	212.6 KB	20,306,542	25,027,822	574.03 KB	574.03 KB	1.3
109.500.10.20	23,438,141	2,10 MB	3,295	6.03 GB	8.73 GB	18,127,259	18,127,259	1,131.03 KB	1,131.03 KB	1.3
109.500.10.20	7,622,075	581.95 KB	1,078	8.51 GB	252.32 MB	4,841,740	2,792,295	948.07 KB	548.07 KB	55.6
109.500.10.20	6,745,179	674.54 KB	1,049	1,006.41 MB	1.85 GB	3,515,361	3,986,978	141.11 KB	141.11 KB	51.6
109.500.10.20	6,692,279	102.52 KB	927	5.89 MB	768.68 MB	3,771,596	2,908,649	551.20 KB	199.7	199.7
109.500.10.20	3,881,075	1,270.27 KB	454	75.94 MB	2.79 GB	1,127,480	2,162,813	12.80 KB	405.6	405.6
109.500.10.20	4,456,402	402.51 KB	618	2.45 MB	118.29 MB	2,618,958	1,837,444	357.25 KB	452	452
109.500.10.20	6,998,824	885.12 KB	1,393	2.26 GB	112.52 MB	5,027,476	4,381,548	426.67 KB	44.4	44.4
109.500.10.20	3,159,281	357.38 KB	438	50.51 MB	2.40 GB	747,051	2,412,242	7.18 KB	350	350
109.500.10.20	2,403,018	189.81 KB	267	19.69 MB	2.48 GB	130,520	1,791,317	2.86 KB	347.6	347.6
109.500.10.20	1,921,426	129.95 KB	623	92.53 MB	1.29 GB	2,137,028	2,351,783	132.61 KB	187.5	187.5
109.500.10.20	1,618,675	219.05 KB	224	2.19 GB	0.90 KB	1,616,571	154	319.05 KB	319.05 KB	319.05 KB
109.500.10.20	1,539,031	305.82 KB	629	1.52 GB	576.17 MB	2,441,856	2,088,177	223.35 KB	83.8	83.8
109.500.10.20	1,589,042	284.97 KB	262	1.94 GB	15.88 MB	1,337,914	302,168	281.08 KB	5.1	5.1
109.500.10.20	2,306,388	238.79 KB	120	1.17 GB	476.72 MB	1,076,818	1,227,570	176.88 KB	87.8	87.8
109.500.10.20	1,581,458	226.62 KB	231	1.53 GB	23.91 MB	1,320,866	342,592	221.08 KB	3.2	3.2
109.500.10.20	1,019,241	220.09 KB	418	548.40 MB	993.28 MB	1,430,378	1,575,882	77.99 KB	142.1	142.1
109.500.10.20	1,344,111	198.48 KB	270	1.03 GB	777.11 MB	1,046,882	897,220	152.07 KB	99.4	99.4

在选择显示全部后，发现还没有办法将所有的 IP 都显示出来（大于等于 10 万个 IP）。而通常来说不太可能会出现如此多的 IP。我们可以观察一下那些 IP 发送了多少流量。



地址	字节数	数据包	每秒流量	每秒数据包数	发送字节数	接收字节数	发包	接收	发送每秒字节数	接收每秒字节数
172.30.29.66	390 B	6	0 B	0	192 B	198 B	2	3	0 B	0 B
172.30.126.252	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.65.207	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.1.11	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.37.183	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.101.255	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.14.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.36.71	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.252.16	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.71.60	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.153.31	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.67.91	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.98.178	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.48.215	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.175.161	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.166.115	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.237.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.170.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.238.95	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.73.60	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.153.31	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.67.91	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.98.178	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.48.215	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.175.161	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.166.115	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.237.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.170.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.238.95	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B

问了一下管理人员，这些 IP 都不是该用户的，而其数据包都是 6 个，再来观察一下它们都在和谁通信。

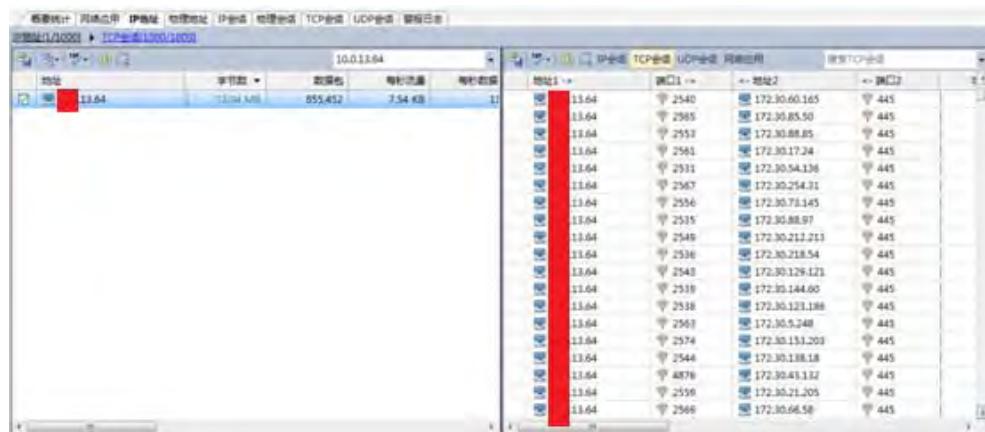


地址	字节数	数据包	每秒流量	每秒数据包数	发送字节数	接收字节数	发包	接收	发送每秒字节数	接收每秒字节数
172.30.29.66	390 B	6	0 B	0	192 B	198 B	2	3	0 B	0 B
172.30.126.252	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.65.207	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.1.11	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.37.183	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.101.255	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.14.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.36.71	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.252.16	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.71.60	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.153.31	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.67.91	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.98.178	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.48.215	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.175.161	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.166.115	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.237.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.170.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.238.95	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.71.60	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.153.31	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.67.91	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.98.178	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.48.215	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.175.161	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.166.115	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.237.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.170.223	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B
172.30.238.95	390 B	6	0 B	0	192 B	198 B	3	3	0 B	0 B

对多个 IP 进行挖掘之后，我们找到了其通信对端：X.X.13.64。



通过搜索功能，我们定位到了这台主机。然后对其行为进行详细的分析。



直接挖掘其 TCP 会话，可以看到他正在用随机端口和各个部存在的主机的 445 端口进行通信。我们将数据包下载下来并进行进一步分析。



可以看到这个主机在向各 IP 发送 SYN 包，一直被各 IP 拒绝

进入数据包进行取证，由下图可以看到 13.64 作为源头时，其发送的确实都是 SYN 包，并且一直被拒绝。

编号	绝对时间	源	目标	协议	大小	解码字段	摘要
6	08:00:15.039686	172.30.111.67:443	13.64.2832	CIFS	64	—_—	序列号=0000000000, 链路号=0238054832
7	08:00:15.044276	13.64.2835	172.30.6.54:445	CIFS	66	_—_—	序列号=1111111111, 链路号=0000000000
8	08:00:15.047232	13.64.2836	172.30.59.168:445	CIFS	66	_—_—	序列号=1421703408, 链路号=0000000000
9	08:00:15.048242	13.64.2837	172.30.210.68:445	CIFS	66	_—_—	序列号=1511144811, 链路号=0000000000
10	08:00:15.050171	13.64.2838	172.30.111.153:445	CIFS	66	_—_—	序列号=1591880221, 链路号=0122962000
11	08:00:15.052055	13.64.2839	172.30.242.119:445	CIFS	66	_—_—	序列号=1612287611, 链路号=0000000000
12	08:00:15.059264	172.30.79.81:445	13.64.2834	CIFS	64	—_—	序列号=163000000000, 链路号=0072997511
13	08:00:15.061508	172.30.6.54:445	13.64.2835	CIFS	64	—_—	序列号=163000000000, 链路号=1711111111
14	08:00:15.062402	172.30.59.168:445	13.64.2836	CIFS	64	—_—	序列号=163000000000, 链路号=1421703408
15	08:00:15.062847	172.30.210.68:445	13.64.2837	CIFS	64	—_—	序列号=163000000000, 链路号=1511144811
16	08:00:15.063266	172.30.111.153:445	13.64.2838	CIFS	64	—_—	序列号=163000000000, 链路号=1591880221

编号	绝对时间	源	目标	协议	大小	解码字段	摘要
6	08:00:15.039686	172.30.111.67:443	13.64.2832	CIFS	64	—_—	序列号=0000000000, 链路号=0238054832
7	08:00:15.044276	13.64.2835	172.30.6.54:445	CIFS	66	_—_—	序列号=1111111111, 链路号=0000000000
8	08:00:15.047232	13.64.2836	172.30.59.168:445	CIFS	66	_—_—	序列号=1421703408, 链路号=0000000000
9	08:00:15.048242	13.64.2837	172.30.210.68:445	CIFS	66	_—_—	序列号=1511144811, 链路号=0000000000
10	08:00:15.050171	13.64.2838	172.30.111.153:445	CIFS	66	_—_—	序列号=1591880221, 链路号=0122962000
11	08:00:15.052055	13.64.2839	172.30.242.119:445	CIFS	66	_—_—	序列号=1612287611, 链路号=0000000000
12	08:00:15.059264	172.30.79.81:445	13.64.2834	CIFS	64	—_—	序列号=163000000000, 链路号=0072997511
13	08:00:15.061508	172.30.6.54:445	13.64.2835	CIFS	64	—_—	序列号=163000000000, 链路号=1711111111
14	08:00:15.062402	172.30.59.168:445	13.64.2836	CIFS	64	—_—	序列号=163000000000, 链路号=1421703408
15	08:00:15.062847	172.30.210.68:445	13.64.2837	CIFS	64	—_—	序列号=163000000000, 链路号=1511144811
16	08:00:15.063266	172.30.111.153:445	13.64.2838	CIFS	64	—_—	序列号=163000000000, 链路号=1591880221

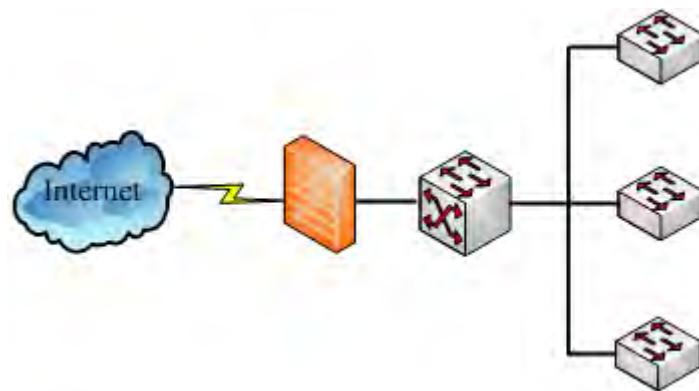
由此得出结论 13.64 存在 SYN FLOOD 攻击行为，建议立刻进行病毒查杀并隔离

11. 某互联网故障分析报告

11.1. 故障环境

1. 故障描述

某互联网的结构比较简单，通过核心和防火墙直接相连，中间没有任何的网络监控、管理设备，示意
图如下所示：



某互联网的防火墙走的是路由模式，内部员工上网通过防火墙进行地址转化，转换成公网地址。

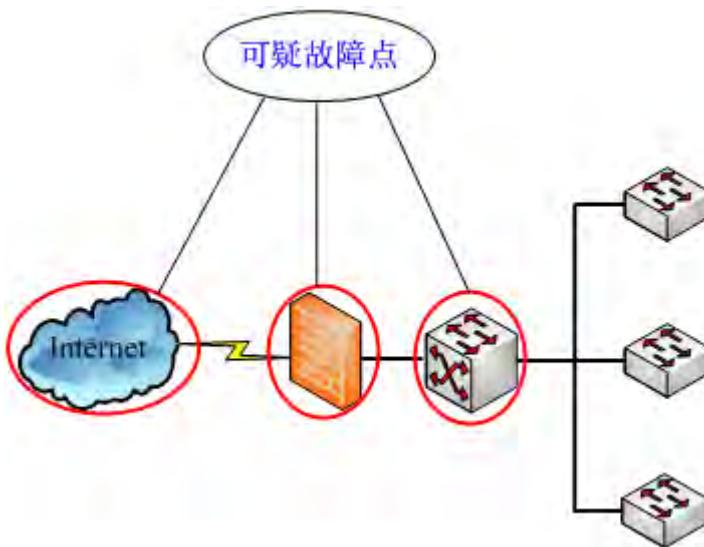
2. 故障现象

打开网页速度比较慢，但是下载的速度很快，而且利用 web 页面上传邮件附件也很慢，几兆的附件经
常上传不上去。同时利用 WEB 页面登陆防火墙的速度也很慢。

11.2. 故障分析

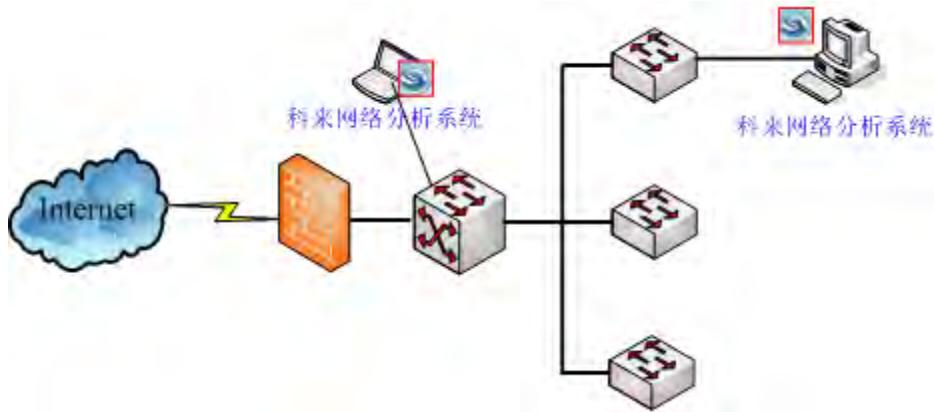
1. 确认故障点

通过上面的介绍，我们可以发现在上网慢时只经过了核心交换机和防火墙，所以可疑的故障点有以下
几个位置：



2. 部署科来网络分析系统

为了找到故障的具体位置，我们找一台慢的主机装上科来网络分析系统，然后在核心交换和防火墙之间部署上科来网络分析系统，通过端口镜像来捕获通信的数据包：



3. 数据包分析

1) 分析防火墙有无丢包、延时。

同样的，我们做故障还原测试，并利用防火墙自带的抓包功能来抓取网络通信的数据包，同时在客户端进行抓包。

通过测试得到如下的数据包：

时间差	源	目标	协议	大小	摘要
0.040789	:1991	:1991	HTTP	66	序列号=2919377234, 确认号=0000000000, 标志=A....S., 长度= 0, 窗口=65535
0.000066	:1991	:1991	HTTP	66	序列号=0877273928, 确认号=2919377235, 标志=A..S., 长度= 0, 窗口= 5840
0.001163	:1991	:1991	HTTP	58	序列号=2919377235, 确认号=0877273929, 标志=A...., 长度= 0, 窗口=65535, 校验和错误
0.000046	:1991	:1991	HTTP	1,418	C: POST /s3/compose/upload.jsp?sid=DBZ0YeqMogrQLhy1pMmhwZumEMqo!type=flash&
0.039540	:1991	:1991	HTTP	341	C: HTTP流还有283字节的数据
0.000043	:1991	:1991	HTTP	70	序列号=0877273929, 确认号=2919377235, 标志=A...., 长度= 0, 窗口= 5840, 校验和错误
0.042551	:1991	:1991	HTTP	1,458	C: 继续或非HTTP通信, 1400 字节的二进制数据
0.000050	:1991	:1991	HTTP	70	序列号=0877273929, 确认号=2919377235, 标志=A...., 长度= 0, 窗口= 5840, 校验和错误
2.932629	:1991	:1991	HTTP	1,418	C: POST /
6.035142	:1991	:1991	HTTP	1,458	C: POST /
11.970487	:1991	:1991	HTTP	1,458	C: POST /
24.039202	:1991	:1991	HTTP	1,458	C: POST /
0.051027	:1991	:1991	HTTP	70	序列号=0877273929, 确认号=2919380278, 标志=A...., 长度= 0, 窗口= 8400, 校验和错误
0.000064	:1991	:1991	HTTP	1,458	C: 继续或非HTTP通信, 1400 字节的二进制数据
0.000019	:1991	:1991	HTTP	1,458	C: 继续或非HTTP通信, 1400 字节的二进制数据
47.928279	:1991	:1991	HTTP	1,458	C: 继续或非HTTP通信, 1400 字节的二进制数据
0.059156	:1991	:1991	HTTP	64	序列号=0877273929, 确认号=2919381678, 标志=A...., 长度= 0, 窗口=11200
0.000039	:1991	:1991	HTTP	1,458	[重传] C: 继续或非HTTP通信, 1400 字节的二进制数据
0.000016	:1991	:1991	HTTP	1,458	C: 继续或非HTTP通信, 1400 字节的二进制数据

客户端数据包

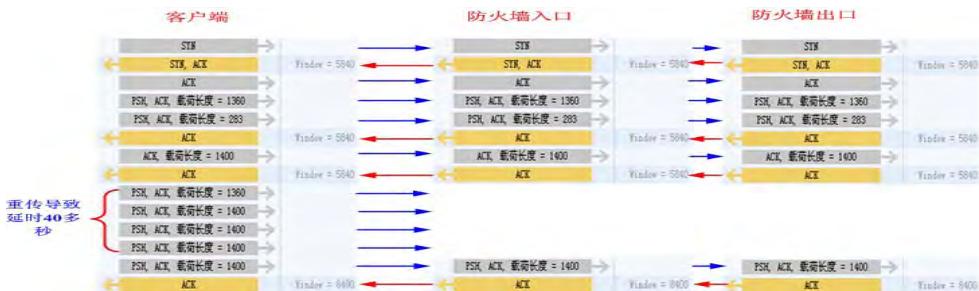
```
早阳烟草.system# tcpdump -i eth2 host 10.■■■■■ and host 220■■■■■
tcpdump: WARNING: eth2: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 68 bytes
08:01:52.960067 R0eth2 IP b■■■■■.1991 > m12-■■■■■.com.80: S 2919377234:2919377234(0) win 65535 <mss 1460,nop,nop,sackOK>
08:01:52.998297 X0eth2 IP m12-■■■■■.com.80 > b■■■■■.1991: S 877273928:877273928(0) ack 2919377235 win 5840 <mss 1400,nop,nop,sackOK>
08:01:53.000049 R0eth2 IP b■■■■■.1991 > m12-■■■■■.com.80: . ack 1 win 65535
08:01:53.000532 R0eth2 IP b■■■■■.1991 > m12-■■■■■.com.80: P 1361:1644(283) ack 1 win 65535
08:01:53.039231 X0eth2 IP m12-■■■■■.com.80 > b■■■■■.1991: . ack 1 win 5840 <nop,nop,sack 1 {1361:1644}>
08:01:53.042806 R0eth2 IP b■■■■■.1991 > m12-■■■■■.com.80: . 1644:3044(1400) ack 1 win 65535
08:01:53.081827 X0eth2 IP m12-■■■■■.com.80 > b■■■■■.1991: . ack 1 win 5840 <nop,nop,sack 1 {1361:3044}>
08:02:13.721728 X0eth2 IP m12-■■■■■.com.80 > b■■■■■.1993: P 652719328:652719614(286) ack 3121251095 win 32767 <nop,nop,sack 1 {2801:4201}>
08:02:38.069703 R0eth2 IP b■■■■■.1991 > m12-■■■■■.com.80: P 1:1401<1400> ack 1 win 65535
08:02:38.100287 X0eth2 IP m12-■■■■■.com.80 > b■■■■■.1991: . ack 3044 win 8400 <nop,nop,sack 1 {1361:1401}>
```

防火墙入口数据包

```
早阳烟草系统# tcpdump -i eth1 host 2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 64 bytes
08:01:52.960108 X@eth1 IP 228.1.1991 > m12.1991: S 2919377234: .2919377234(0) win 65535 <nop,nop,sackOK>
08:01:52.998293 R@eth1 IP m12.1991 > 228.1.1991: E 877273928: 877273928(0) ack 2919377235 win 5840 <nop,nop,sackOK>
08:01:53.000043 X@eth1 IP 228.1.1991 > m12.1991: . ack 1 win 65535
08:01:53.000534 X@eth1 IP 228.1.1991 > m12.1991: . com.80: P 1361:16440: 2832 ack 1 win 65535
08:01:53.039228 R@eth1 IP m12.1991 > 228.1.1991: . ack 1 win 5840 <nop,nop,sack 1 (1361:16440)>
08:01:53.042892 X@eth1 IP 228.1.1991 > m12.1991: . 1644:30440: 1400) ack 1 win 65535
08:01:53.081823 R@eth1 IP m12.1991 > 228.1.1991: . ack 1 win 5840 <nop,nop,sack 1 (1361:30440)>
08:02:13.721717 R@eth1 IP m12.1991 > 228.1.1991: P 652719320: 652719614(2060) ack 3121251095 win 32767 <nop,nop,sack 1 (2801:4201)>
08:02:38.069711 X@eth1 IP 228.1.1991 > m12.1991: . 1:1401<1400) ack 1 win 65535
08:02:38.100282 R@eth1 IP m12.1991 > 228.1.1991: . ack 3044 v in 1400 <nop,nop,sack 1 (1361:1401)>
```

防火墙出口数据包

通过上面的数据包我们可以得出一个数据包交互图：



通过比较数据包，我们可以发现在防火墙入口处有数据包丢失！造成这种丢失的最大可能性就是防火墙和中间设备之间有其他网络设备导致数据包丢失，但是在防火墙和客户端之间没有其他设备，只有一个核心交换机，为了能够准确找出故障点，我们再捕获防火墙和核心交换之间的数据包与客户端发送的数据包进行比较。

2) 检测核心交换和客户端之间有没有丢包、延时。

部署好分析软件后，我们做故障还原测试，并同时开启在客户端和核心交换处的抓包工具，得到如下数据包：

概要								诊断		协议		IP端点		IP会话		TCP会话		UDP会话		矩阵		数据包		日志		报表		
节点1->	<-节点2	数据包	字节	协议	持续时间	字节->	<字节																					
10.	3524	220.	80	14	3.157 KB	HTTP	00:00:25	2.285 KB	893 B																			
10.	3536	220.	80	12	2.749 KB	HTTP	00:00:25	1.805 KB	967 B																			
10.	3541	220.	80	21	5.255 KB	HTTP	00:00:22	4.117 KB	1.138 KB																			
10.	3542	220.	80	927	808.376 KB	HTTP	00:01:53	785.104 KB	23.271 KB																			
10.	3543	220.	80	12	4.125 KB	HTTP	00:00:27	3.172 KB	979 B																			

数据包								数据流		时序图		10.50.0.168 <-> 220.181.12.208\局域网															
编号	时间差	源	目标					协议	大小	解... 序列号	摘要																
1118		USER-3	3524	twebmai	80			HTTP	62		序列号=0994174																
1120	0.039887	twebmai	3524	3...	USER-30			HTTP	62		序列号=0066719																
1121	0.000024	USER-3	3524	twebmai	80			HTTP	54		序列号=0994172																
1122	0.000342	USER-3	3524	twebmai	80			HTTP	1,454		C: POST /js3/																
1123	0.000006	USER-3	3524	twebmai	80			HTTP	59		C: HTTP流还有待续																
1124	0.000076	USER-3	3524	twebmai	80			HTTP	549		C: HTTP流还有待续																
1125	0.038571	twebmai	3524	3...	USER-30			HTTP	60		序列号=0066719																
1126	0.000009	twebmai	3524	3...	USER-30			HTTP	60		序列号=0066719																
1127	0.000006	twebmai	3524	3...	USER-30			HTTP	60		序列号=0066719																
1128	0.056830	twebmai	3524	3...	USER-30			HTTP	591		S: HTTP/1.0 200																
1129	0.141008	USER-3	3524	twebmai	80			HTTP	54		序列号=0994174																
1336	23.956147	twebmai	3524	3...	USER-30			HTTP	60		序列号=0066719																
1337	0.000028	USER-3	3524	twebmai	80			HTTP	54		序列号=0994174																
1340	0.902166	USER-3	3524	twebmai	80			HTTP	54		序列号=0994174																

客户端数据包

概要								诊断		协议		IP端点		IP会话		TCP会话		UDP会话		矩阵		数据包		日志		报表		
节点1->	<-节点2	数据包	字节	协议	持续时间	字节->	<字节																					
10.	3524	220.	80	14	3.236 KB	HTTP	00:00:25	2.341 KB	917 B																			
10.	3536	220.	80	12	2.819 KB	HTTP	00:00:25	1.855 KB	987 B																			
10.	3541	220.	80	21	5.360 KB	HTTP	00:00:22	4.184 KB	1.177 KB																			
10.	3542	220.	80	926	810.579 KB	HTTP	00:01:37	785.858 KB	24.721 KB																			
10.	3543	220.	80	12	4.192 KB	HTTP	00:00:27	3.217 KB	999 B																			

数据包								数据流		时序图		10.50.0.168 <-> 220.181.12.208\局域网															
编号	时间差	源	目标					协议	大小	解... 序列号	摘要																
561		USER-3	3543	twebmai	80			HTTP	66		序列号=1886281																
562	0.037422	twebmai	3543	3...	USER-3			HTTP	66		序列号=2581509																
563	0.000068	USER-3	3543	twebmai	80			HTTP	64		序列号=1886281																
564	0.000367	USER-3	3543	twebmai	80			HTTP	1,446		C: POST /js3/																
565	0.000021	USER-3	3543	twebmai	80			HTTP	132		C: HTTP流还有待续																
566	0.038475	twebmai	3543	3...	USER-3			HTTP	70		序列号=2581509																
639	3.002640	USER-3	3543	twebmai	80			HTTP	1,458		C: POST /js3/																
640	0.038309	twebmai	3543	3...	USER-3			HTTP	70		序列号=2581509																
641	0.009747	twebmai	3543	3...	USER-3			HTTP	729		S: HTTP/1.0 200																
647	0.248741	USER-3	3543	twebmai	80			HTTP	64		序列号=1886282																
1110	24.618643	twebmai	3543	3...	USER-3			HTTP	64		序列号=2581510																
1111	0.000130	USER-3	3543	twebmai	80			HTTP	64		序列号=1886282																

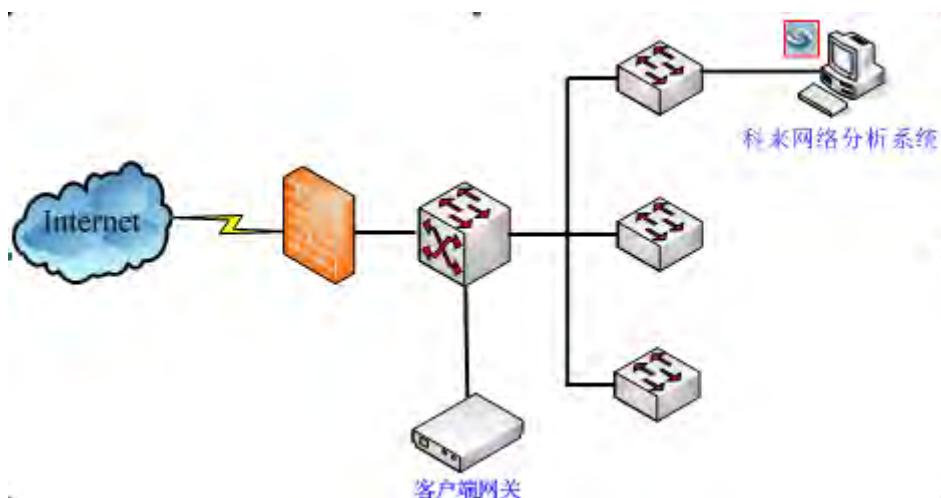
核心交换处数据包

从上面的数据包对比可以看到，凡是客户端发送的数据包都转发到了核心交换处，核心交换处捕获的数据包和在客户端捕获的数据包是一样的，所以可以得出在核心交换到客户端之间没有数据包丢失。

11.3. 分析结论

通过上面的分析我们可以知道，客户端和核心交换机并没有丢包，但是在防火墙的入口处抓包发现确实有丢包，所以该故障的可能原因如下：

1. 防火墙故障，防火墙丢弃了数据包交互的数据包，在入口处没有捕获到是因为防火墙收到数据包后的处理流程导致，因为防火墙有很多的匹配模块，如果丢包的匹配模块在Tcpdump之前，我们使用Tcpdump抓包就抓不到数据包；
2. 路由设置有问题，客户端的网关地址不是核心交换的VLAN地址或是防火墙的内网口地址，而是接在核心交换机上的另外的设备，如下图所示：



而在核心交换到网关设备之间，或是网管到防火墙之间有设备丢包。（可以通过 TTL 来判断，但是发现 TTL 在防火墙和客户端是一样的！）

3. 交换和防火墙之间有设备，只是没有发现！需要认真的查一下设备。

11.4. 总结

我们在上面的分析中可以看到，在防火墙上开启抓包功能时，发现凡是到防火墙入口的数据包都转发到防火墙出口处了，从这看故障应该和防火墙没有关系，丢包应该是防火墙前面的设备造成的！

但是在核心交换机处看数据包并没有丢包，数据包应该都到达防火墙了（因为防火墙和核心交换机之间没有其它设备了）！

那为什么会出现上面防火墙抓包的现象呢？需要具体的联系防火墙厂家查看具体的原因。

12. 某省天然气分析报告

12.1. 测试描述

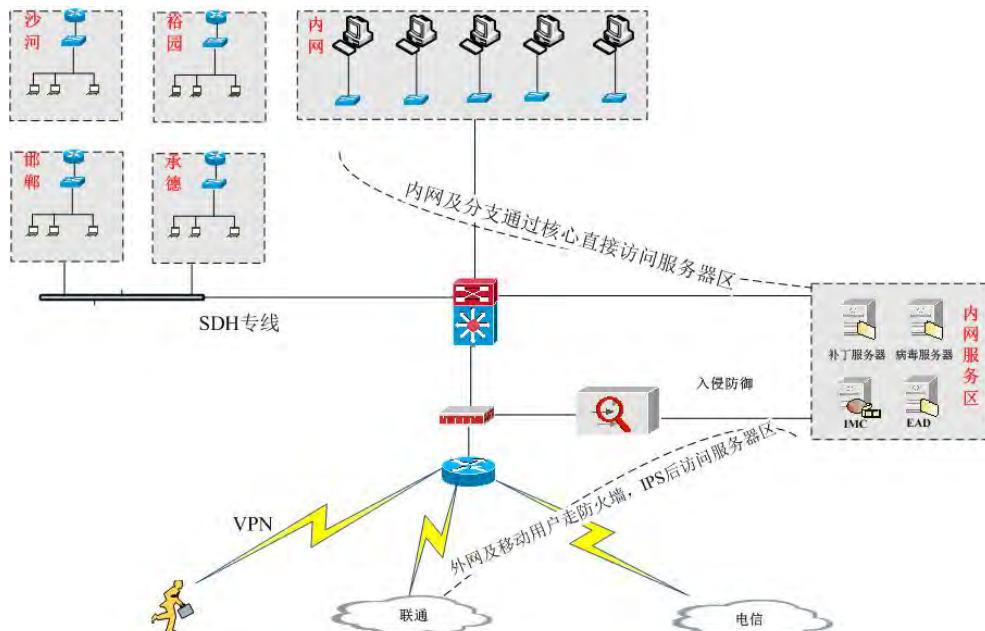
客户反映：网络有延时，ping 内网部分服务器延时高。网络环境中有上网行为产品，出部分特殊主机外均做有流策略。

检测软件：科来网络分析系统 2010

部署网络节点：联通，典型 Internet 出口链路。承德，邯郸，沙河，裕园等 分支 SDH 专线。

采样日期：13: 30-16: 30

技术支持：科来某办事处技术支持中心



如上网络拓扑，外网及移动办公用户走防火墙和 IPS 进入服务器区，内网用户和分支机构用户直接走专线从核心访问 DMZ 区。

12.2. 基本流量带宽占用率分析

1. 联通流量状况：



联通带宽为 10M，采样中多次出现利用率 100% 的情况，部分峰值数据流达到 14.2M 以上，网络拥塞

严重。

2. 电信流量状况:



电信带宽为 10M。采样中利用率 30%左右，峰值数据流在 3.0M 左右，流量状况无异常。

3. 地级市一流量状况:



地级市一专线带宽为 2M, 采样中利用率多次达到 100%。采样峰值曾达到 2.6M, 网络拥塞现象明显。

4. 地级市二流量状况:



地级市二专线带宽为 2M, 采样中利用率多次达到 100%。采样峰值曾达到 3.6M, 网络拥塞现象明显。

5. 地级市三流量状况:



地级市三专线为 2M, 采样显示虽然有部分突发流量达到 1.9M 左右, 但大部分时间数据流并不是很大。

6. 地级市四等专线:



地级市四采样时网络利用率 25%左右，流量峰值在 1.2M 左右，专线带宽 2M，流量状况显示暂无异常。

宁晋，客服等专线采样流量跟承德类似，利用率不是很高。流量显示暂无异常。

12.3. 各专线网络评估分析

1. 联通专线

1) 概要统计

诊断统计		数量			
信息类诊断		28,921			
注意类诊断		1,806			
警告类诊断		26,175			
错误类诊断		0			
流量统计		字节数	数据包数	利用率	每秒位数
总流量		544.467 MB	1,019,512	79.572%	7.957 Mbps
广播流量		124.319 KB	2,072	0.014%	1.392 Kbps
多播流量		1.259 KB	9	0.000%	0 bps
平均包长		559.988 字节			
数据包大小分布		字节数	数据包数	利用率	每秒位数
<=64		25.811 MB	423,008	2.154%	215.408 Kbps
65-127		12.124 MB	150,870	2.002%	200.184 Kbps
128-255		6.407 MB	37,876	0.975%	97.536 Kbps
256-511		14.316 MB	40,336	2.131%	213.136 Kbps
512-1023		22.574 MB	33,252	4.274%	427.352 Kbps
1024-1517		149.433 MB	117,407	23.832%	2.383 Mbps
>=1518		313.803 MB	216,763	44.204%	4.420 Mbps
地址统计		数量			
物理地址数		6			
IP地址数		4,025			
本地IP地址数		8			

在概要统计中，我们可以看出采样的联通数据流概要信息显示，在本图表中，我们看到总流量为 544M，峰值利用率多次达到 100%。线路拥塞严重。

在数据包大小分布上图中，正常的平均包长应该在 512-800 之间。平均包长在正常范围内，但偏向小包应用较多，仅小于 64 的数据包达到每秒 421 个。当小包过多时，会对网络速度产生相应的影响。

广播包和组播包，正常应该每秒 20 个以内。本次采样 3 个，属正常范围。

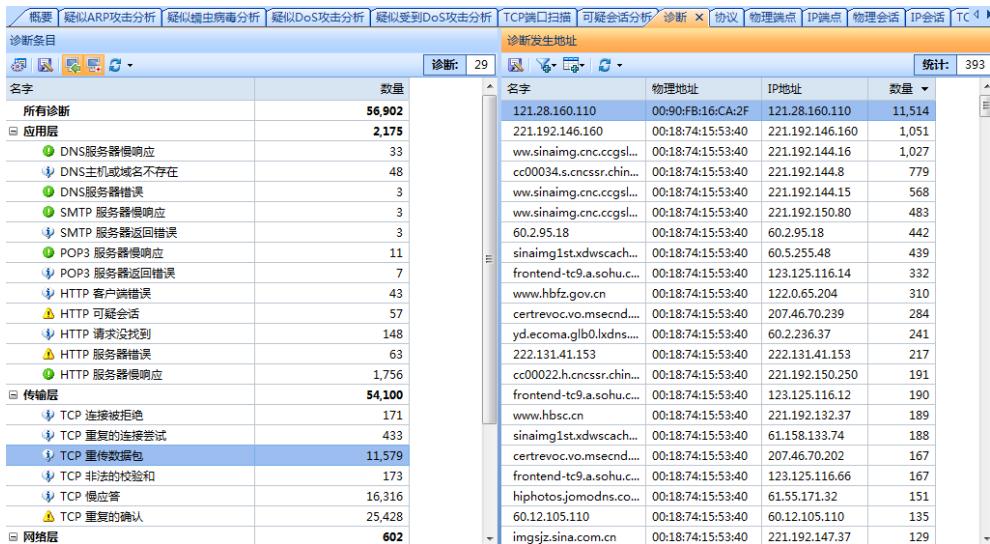
数据流统计		数量	
物理会话		5	
IP会话		2,125	
TCP会话		7,208	
UDP会话		2,152	

正常情况下 TCP 应大于 UDP 回话，本采样正常。

TCP统计		数量	
TCP同步发送		6,821	
TCP同步确认发送		6,872	
TCP结束连接发送		12,708	
TCP复位发送		1,189	

正常网络通讯中，同步发送基本等于同步确认发送。本采样正常。

2) 诊断异常分析



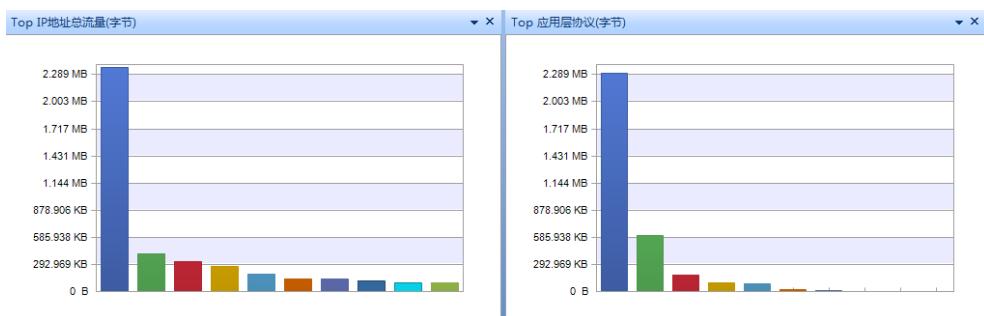
从自动诊断信息上看，我们发现网络中存在大量 TCP 重传、重复连接请求、慢应答现象，这是较典型的网络传输质量差丢包导致的传输异常现象。

正式由于网络传输质量差，连带着出现大量 HTTP 服务器慢应答，DNS 服务慢应答等应用层的诊断报告。

同时在网络层出现大量的 ICMP 网络、主机、端口不可达警告，这主要是很多的目标地址无法访问，一般是由于扫描或者是 P2P 应用大量连接外部主机引起的。

3) 流量异常分析

利用科来网络分析系统的协议和端点分析视图，对影响网络较大的应用和主机进行分析，主要分析其网络行为。



流量最大的主机和应用

从传输层应用上看，HTTP 应用流量占据了 80% 以上网络带宽。其后依次为 HTTP-OTHER,HTTPS,UDP-OTHER,RTSR, PPTP 等。

对 UDP-Other 流量进一步分析。

以下为 NetworkMiner 工具截取的 UDP 会话数据：

节点1->	<-节点2	持续时间	字节	字节->	<-字节	数据包	数据包
121.28.160.110:65449	cnchub5u.sandai.net:8000	00:00:00	227 B	227 B	0 B	1	
121.28.160.110:46825	client.stat.xunlei.com:80	00:00:00	505 B	351 B	154 B	2	
121.28.160.110:16255	221.208.127.58:18484	00:00:00	150 B	150 B	0 B	1	
121.28.160.110:27972	c0913.sandai.net:80	00:00:00	276 B	202 B	74 B	2	
121.28.160.110:51216	cncj.phn.p2p.baofeng.net:8000	00:00:00	330 B	81 B	249 B	2	
121.28.160.110:43326	202.99.160.68:53	00:00:00	225 B	85 B	140 B	2	
121.28.160.110:46526	202.99.160.68:53	00:00:00	241 B	85 B	156 B	2	
121.28.160.110:61897	cncsz.phn.p2p.baofeng.net:8000	00:00:00	277 B	80 B	197 B	2	
202.110.65.10:30956	121.28.160.110:3614	00:00:00	150 B	150 B	0 B	1	
121.28.160.110:50296	202.99.160.68:53	00:00:00	414 B	91 B	323 B	2	
121.28.160.110:4450	202.99.160.68:53	00:00:00	450 B	106 B	344 B	2	
121.28.160.110:40338	curl.qlb.com:53	00:00:00	846 B	718 B	128 B	4	
121.28.160.110:18708	119.114.4.124:1122	00:00:00	76 B	76 B	0 B	1	

针对 UDP-Other 的端点分析

通过对 UDP-Other 协议流量的端点分析，我们发现 UDP-Other 流量主要由 121.28.90.110 产生，与他建立的 UDP 连接中，有多个是 P2P 下载的站的连接，如暴风影音的资源站点。通过进一步分析，这些主机都是采用 P2P 软件下载或安装了 P2P 软件自动上传，上传大量流量，占据网络带宽，从而造成一定的网络拥塞。

同时，经管理员协助，得知网络中存在视频会议系统，这些正常的视频流量也是走的 UDP 协议。

网络中 HTTP 访问的均为 80 端口，访问的网站以 360 的团购网，汽车网，新浪网等居多。

网络中存在大量的 HTTPS 协议，并由此产生了大量的 64 位小包。高于正常情况。但经咨询河北天然气网络管理人员得知，此 443 端口的访问为员工访公司 OA 系统，为正常的访问。

RTSR 实时流协议。

PPTP 隧道协议。用于公司与分公司建建立 VPN 隧道。属于 2 层隧道协议，没有加密。若用于传输机要文件，则存在一定安全隐患。建议采用 ipsec vpn 或 ssl vpn 模式建立隧道连接。

4) 连接异常分析

主要针对总体分析中发现的异常进行进一步分析。

组播包分析

以下为 NetworkMiner 工具截取的组播包数据：

源	目标	协议	大小	帧数/包	摘要
16:54:47.445889	00:90:FB:16:C4:2F	ARP	64	1	发送 121.28.160.110>广播 121.28.160.110
16:54:47.445711	00:90:FB:16:C4:2F	ARP	64	1	发送 121.28.160.110>广播 121.28.160.110
16:54:48.145793	00:90:FB:16:C4:2F	ARP	64	1	发送 121.28.160.110>广播 121.28.160.110
16:54:48.282798	192.168.1.242:37	TCP	96	1	C: 客户=PC-2011081720409.1<<
16:54:48.445571	00:90:FB:16:C4:2F	ARP	64	1	发送 121.28.160.110>广播 121.28.160.110
16:54:49.031719	192.168.1.242:37	TCP	96	1	C: 客户=PC-2011081720409.1<<
16:54:49.145729	00:90:FB:16:C4:2F	ARP	64	1	发送 121.28.160.110>广播 121.28.160.110
16:54:49.445503	00:90:FB:16:C4:2F	ARP	64	1	发送 121.28.160.110>广播 121.28.160.110
16:54:49.669806	192.168.1.242:37	TCP	96	1	C: 客户=WWW.RAIIPI.COM <<
16:54:49.781809	192.168.1.242:37	TCP	96	1	C: 客户=PC-2011081720409.1<<

经过分析，发现多播包和广播包为多台主机发送，没有单台主机发送大量多播包或广播包的情况，主要是路由信息、热备主机心跳信息及网关发送，没有发现异常数据包。

物理地址数量大现象分析

名字	字节	数据包	每秒位	接收字节	接收数据包	发送字节
00:1B:38:A4:39:38	2.099 KB	26	0 bps	1.130 KB	11	992 B
01:03:D8:CB:57:6D	2.046 KB	26	0 bps	0 B	0	2.046 KB
01:03:D8:CB:57:6C	2.046 KB	26	0 bps	0 B	0	2.046 KB
01:03:D8:CB:57:66	2.045 KB	26	0 bps	0 B	0	2.045 KB
01:03:D8:CB:57:6A	1.892 KB	24	0 bps	0 B	0	1.892 KB
01:03:D8:CB:57:6F	1.892 KB	24	0 bps	0 B	0	1.892 KB
01:03:D8:CB:57:6E	1.892 KB	24	0 bps	0 B	0	1.892 KB
01:03:D8:CB:57:65	1.891 KB	24	0 bps	0 B	0	1.891 KB
01:03:D8:CB:57:69	1.891 KB	24	0 bps	0 B	0	1.891 KB
01:03:D8:CB:57:67	1.891 KB	24	0 bps	0 B	0	1.891 KB
01:03:D8:CB:57:68	1.891 KB	24	0 bps	0 B	0	1.891 KB
00:1C:25:9D:4B:1E	1.752 KB	20	3.952 Kbps	394 B	5	1.367 KB
01:03:D8:CB:57:64	1.621 KB	25	0 bps	0 B	0	1.621 KB
01:03:D8:CB:57:6B	1.621 KB	25	0 bps	0 B	0	1.621 KB
00:1A:64:6D:5D:73	1.449 KB	20	0 bps	1.262 KB	17	192 B
00:1B:38:A4:39:3E	1.329 KB	20	0 bps	1.011 KB	4	229 B

大量地址为交换机或路由设备地址，流量也为正常流量。

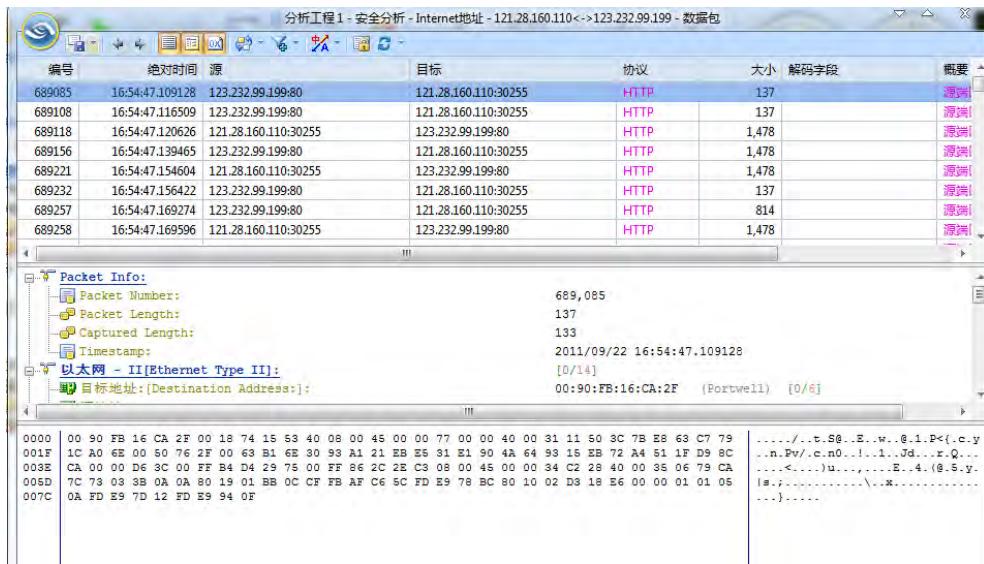
UDP 会话分析

经分析，大量的 UDP 会话主要是由于 P2P 软件及视频会议流量产生。

TCP 连接请求分析

Internet地址IP会话 4,109								
节点1->	<-节点2	持续时间	字节	字节->	<-字节	数据包	数据包-> ▾	<- 数据包 开始△
121.28.160.110	tuan-s.360.cn	00:03:06	5.366 MB	2.672 MB	2.694 MB	87,535	43,747	43,788 1
121.28.81.218	121.28.160.110	00:18:27	33.680 MB	32.793 MB	909.212 KB	46,264	32,019	14,245 1
121.28.160.110	123.232.99.199	00:18:28	27.895 MB	14.690 MB	13.205 MB	52,179	28,799	23,380 1
221.192.146.160	121.28.160.110	00:04:14	22.023 MB	21.527 MB	508.000 KB	22,998	14,870	8,128 1
221.192.144.8	121.28.160.110	00:09:02	13.571 MB	12.631 MB	962.963 KB	19,295	10,957	8,338 1
121.28.160.110	www.sinaimg.cnc.ccg...	00:17:55	20.446 MB	698.883 KB	19.764 MB	24,276	9,864	14,412 1
sinaimg1st.dwdasca...	121.28.160.110	00:18:27	13.028 MB	12.448 MB	593.913 KB	16,332	9,375	6,957 1
123.147.190.5	121.28.160.110	00:03:31	13.342 MB	12.852 MB	501.375 KB	16,900	8,878	8,022 1
121.28.160.110	www.hbfz.gov.cn	00:17:42	15.269 MB	1.044 MB	14.224 MB	18,800	7,969	10,831 1
121.28.160.110	182.118.6.115	00:00:25	22.395 MB	448.482 KB	21.957 MB	22,339	7,168	15,171 1
123.138.26.12	121.28.160.110	00:18:27	9.756 MB	9.500 MB	262.981 KB	10,695	6,816	3,879 1
121.28.160.110	a855.g.akamai.net	00:16:38	11.576 MB	353.129 KB	11.231 MB	13,333	5,540	7,793 1
121.28.160.110	ide.buimin.com	00:19:26	1.620 MB	670.699 KB	670.699 KB	0,020	6,409	2,671 1

通过端点视图分析，发现网络中 121.28.160.110、221.192.146.160 发送的 tcp 同步发送数量远远高于同步确认接收数量，我们进行进一步分析。



TCP 发送连接请求走的都是 80 端口，且都得到回应，经查询 IP 地址发现为公司内部服务器间进行数据的传输。

5) 安全异常分析

ARP 攻击分析

利用安全分析方案中的 ARP 攻击分析是否存在 ARP 攻击情况。

The screenshot shows the 'Diagnosis Items' and 'Diagnosis Events' sections of the network analysis tool.

诊断条目 (Diagnosis Items):

名字	数量
TCP 偏应答	16,316
TCP 重复的确认	25,428
网路层	
ICMP 目的不可达	70
ICMP 网络不可达	1
ICMP 主机不可达	8
ICMP 端口不可达	523
数据链路层	
ARP 扫描	20
ARP 太多的主动应答	5

诊断发生地址 (Diagnosis Occurred Address):

名字	物理地址	IP 地址	数量
00:1D:72:C1:05:A5	00:1D:72:C1:05:A5	-	20
FF:FF:FF:FF:FF:FF	FF:FF:FF:FF:FF:FF	-	20

诊断事件 (Diagnosis Events):

事件描述	源 IP 地址	源物理地址	目标 IP 地址	目标物理地址
ARP 地址扫描(主机的 MAC 为 00:1D:72:C1:05:A5, IP 地址为 192.168.1.242)	192.168.1.242	00:1D:72:C1:05:A5	192.168.1.107	FF:FF:FF:FF:FF:FF
ARP 地址扫描(主机的 MAC 为 00:1D:72:C1:05:A5, IP 地址为 192.168.1.242)	192.168.1.242	00:1D:72:C1:05:A5	192.168.1.107	FF:FF:FF:FF:FF:FF
ARP 地址扫描(主机的 MAC 为 00:1D:72:C1:05:A5, IP 地址为 192.168.1.242)	192.168.1.242	00:1D:72:C1:05:A5	192.168.1.107	FF:FF:FF:FF:FF:FF
ARP 地址扫描(主机的 MAC 为 00:1D:72:C1:05:A5, IP 地址为 192.168.1.242)	192.168.1.242	00:1D:72:C1:05:A5	192.168.1.107	FF:FF:FF:FF:FF:FF
ARP 地址扫描(主机的 MAC 为 00:1D:72:C1:05:A5, IP 地址为 192.168.1.242)	192.168.1.242	00:1D:72:C1:05:A5	192.168.1.107	FF:FF:FF:FF:FF:FF

发现地址为 192.168.1.242 的主机疑似进行 arp 扫描，且目的地址单一为 192.168.1.107，请检查链路联通状态。

TCP 端口扫描分析

利用安全分析方案中的 TCP 端口扫描分析来分析网络中是否存在 TCP 端口扫描的主机。

The screenshot shows the 'TCP端口扫描' (TCP Port Scan) section of the network analysis tool.

安全分析 (TCP端口扫描):

名字	数据包	每秒字节	每秒数据包	接收数据包	发送数据包	发送/接收 C 数据...	IP会话	TCP会话	UDP会话
没有可显示的列表项目。									

IP会话 (IP Session):

节点1->	<-节点2	持续时间	字节	字节->	<-字节	数据包	数据包->	<- 数据包	开始发包时间	最后发
没有可显示的列表项目。										

未发现网络中有 TCP 端口扫描的主机。

其他安全状态检测也均未出现异常状况。

2. 电信专线

1) 概要统计

流量统计					
	字节数	数据包数	利用率	每秒位数	每秒包数
总流量	61.216 MB	86,824	10.448%	1.045 Mbps	203
广播流量	60.303 KB	1,021	0.022%	2.160 Kbps	4
多播流量	805 B	5	0.000%	0 bps	0
平均包长				739.310 字节	
数据包大小分布					
<=64	812.754 KB	13,071	0.290%	29.040 Kbps	57
65-127	2.114 MB	26,921	0.214%	21.408 Kbps	34
128-255	194.292 KB	1,222	0.040%	4.024 Kbps	3
256-511	864.508 KB	3,037	0.190%	19.016 Kbps	6
512-1023	1.390 MB	2,430	0.1780%	177.984 Kbps	37
1024-1517	36.780 MB	26,946	1.376%	137.568 Kbps	12
>=1518	19.105 MB	13,197	6.558%	655.776 Kbps	54
地址统计					
物理地址数				数量	
物理地址数				5	

在电信专线中，并无异常流量。平均包场为 739.310 字节，在正常包长范围内。利用率不告，无网络拥塞现象。

2) 网络中的异常诊断信息

名字	数量
HTTP 服务器错误	2
HTTP 服务器慢响应	27
传输层	3,229
TCP 连接被拒绝	2
TCP 重复的连接尝试	26
TCP 重传数据包	116
TCP 慢应答	2,564
TCP 重复的确认	521
网络层	169
IP TTL太小	1
ICMP 跳口不可达	168

严重程度	类型	层别	事件描述	源IP地址	源物理地址	目标IP地址
性能	性能	传输层	太慢的TCP应答(数据包[107]与数据包[77]相隔263毫秒)	219.148.43.118	00:90:FB:16:CA:2C	61.187.123.142
性能	性能	传输层	太慢的TCP应答(数据包[110]与数据包[83]相隔404毫秒)	219.148.43.118	00:90:FB:16:CA:2C	21tb-dx2.cneln.net
性能	性能	传输层	太慢的TCP应答(数据包[111]与数据包[83]相隔404毫秒)	219.148.43.118	00:90:FB:16:CA:2C	21tb-dx2.cneln.net
性能	性能	传输层	太慢的TCP应答(数据包[173]与数据包[109]相隔395毫秒)	219.148.43.118	00:90:FB:16:CA:2C	61.187.123.142
性能	性能	传输层	太慢的TCP应答(数据包[233]与数据包[72]相隔1743毫秒)	219.148.43.118	00:90:FB:16:CA:2C	61.187.123.142
性能	性能	传输层	太慢的TCP应答(数据包[237]与数据包[126]相隔63毫秒)	219.148.43.118	00:90:FB:16:CA:2C	61.187.123.142
性能	性能	传输层	太慢的TCP应答(数据包[265]与数据包[240]相隔263毫秒)	219.148.43.118	00:90:FB:16:CA:2C	21tb-dx2.cneln.net
性能	性能	传输层	太慢的TCP应答(数据包[266]与数据包[240]相隔263毫秒)	219.148.43.118	00:90:FB:16:CA:2C	21tb-dx2.cneln.net
性能	性能	传输层	太慢的TCP应答(数据包[303]与数据包[236]相隔547毫秒)	219.148.43.118	00:90:FB:16:CA:2C	61.187.123.142
性能	性能	传输层	太慢的TCP应答(数据包[322]与数据包[242]相隔594毫秒)	219.148.43.118	00:90:FB:16:CA:2C	61.187.123.142

在本专线中，不存在网络拥塞现象。但是在采样数据流中，我们发现有大量的 TCP 慢应答。主要集中在 219.148.43.118, 61.164.142.204, 61.187.123.142 及 oa 系统见出现慢相应。建议检查路由器硬件负载状况，链路下端与服务器间的其他防护设备（如防火墙，IPS 等），路由设置以确定是否为最佳路由。

采样中，电信链路并未出现其他异常流量现象。

3. 地级市一专线：

1) 概要统计

流量统计		字节数	数据包数	利用率	每秒位数	每秒包数
总流量	180.880 MB	350,639	25.185%	2.519 Mbps	523	
广播流量	774.256 KB	10,022	0.049%	4.864 Kbps	9	
多播流量	335.004 KB	2,517	0.388%	38.800 Kbps	13	
平均包长					540.915 字节	
数据包大小分布		字节数	数据包数	利用率	每秒位数	每秒包数
<=64	5.096 MB	83,489	0.430%	43.008 Kbps	84	
65-127	7.586 MB	100,331	1.180%	118.024 Kbps	185	
128-255	3.849 MB	21,888	0.414%	41.432 Kbps	28	
256-511	8.125 MB	25,675	0.661%	66.088 Kbps	23	
512-1023	9.942 MB	13,377	0.739%	73.896 Kbps	15	
1024-1517	71.854 MB	54,467	17.025%	1.702 Mbps	149	
>=1518	74.428 MB	51,412	4.736%	473.616 Kbps	39	

平均包长在 540.915，属于正常范围。但网络利用率在采样过程中多次达到 100%，平均每秒包数大于 500 个，而专线的带宽为 2M，拥塞现象明显。

小包大包的数量基本持平，小包稍多。但由于网络中有大量的正常的小包应用。所以尚属正常范围。

数据流统计		数量
物理会话		226
IP 会话		1,854
TCP 会话		3,882
UDP 会话		3,414
TCP 统计		数量
TCP 同步发送		3,878
TCP 同步确认发送		3,624
TCP 结束连接发送		5,353
TCP 复位发送		2,035
DNS 分析		数量
DNS 查询		1,031
DNS 回应		827

在本次采样中，存在大量的 UDP 会话。TCP 连接复位发送较多，说明有大量的 TCP 连接没正常结束连接。需进一步核查。DNS 查询大于回应，网络中存在 DNS 查询失败错误，大量的执行 DNS 查询，也会造成网络中小包变多。

2) 异常诊断分析

诊断条目		诊断	24	诊断发生地址	统计:
名字	数量				
所有诊断	23,945				
应用层	891				
DNS 服务器慢响应	10				
DNS 主机或域名不存在	99				
DNS 服务器错误	104				
POP3 服务器慢响应	4				
POP3 服务器返回错误	3				
HTTP 客户端错误	2				
HTTP 可疑会话	2				
HTTP 请求没找到	21				
HTTP 服务器错误	3				
HTTP 服务器慢响应	643				
传输层	18,990				
TCP 重复的连接尝试	211				
TCP 重传数据包	432				
TCP 慢应答	17,792				
TCP 重复的确认	555				
网络层	3,734				
IP 地址冲突	3,430				
ICMP 主机不可达	9				
ICMP 端口不可达	295				
数据链路层	330				
ARP 请求风暴	8				
ARP 扫描	322				
诊断事件					

从自动诊断信息上看，我们发现网络中存在大量 TCP 重传、重复连接请求、TCP 慢应答现象，这是较典型的网络传输质量差丢包导致的传输异常现象。

正由于网络传输质量差，连带着出现大量 HTTP 服务器慢应答，DNS 服务慢应答等应用层的诊断报告。

诊断条目		诊断发生地址			
名字	数量	名字	物理地址	IP地址	数量
所有诊断	23,945	10.69.20.253	00:1E:49:BE:CC:C0	10.69.20.253	102
应用层	891	hebgnc.com	00:21:D8:5D:8A:C0	192.168.0.248	51
DNS服务器慢响应	10	10.69.20.13	00:0F:E1:A3:8B:57	10.69.20.13	2
DNS主机或域名不存在	99	hebgnc.com	00:21:D8:5D:8A:C0	192.168.0.249	53
DNS服务器错误	104				
POP3 服务器慢响应	4				
POP3 服务器返回错误	3				
HTTP 客户端错误	2				
HTTP 可疑会话	2				
HTTP 请求未找到	21				
HTTP 服务端错误	3				
HTTP 超时禁用	612				

诊断事件					
严重程度	类型	层别	事件描述	源IP地址	源物理地址
故障	应用层	服务器失败, 数据包编号为 215267	hebgnc.com	00:21:D8:5D:8A:C0	10.69.20.253
故障	应用层	服务器失败, 数据包编号为 215275	hebgnc.com	00:21:D8:5D:8A:C0	10.69.20.253
故障	应用层	服务器失败, 数据包编号为 215281	hebgnc.com	00:21:D8:5D:8A:C0	10.69.20.253
故障	应用层	服务器失败, 数据包编号为 215284	hebgnc.com	00:21:D8:5D:8A:C0	10.69.20.253

在上图中，我们可以看到网络中存在大量的 DNS 服务器错误及 DNS 主机域名不存在。经检查发现，多为 hebgnc.com 查询失败。请检查 DNS 主机客户端请求是否正确，正确配置 DNS 地址。

同时在网络层出现大量的 ICMP 网络、主机、端口不可达警告，这主要是很多的目标地址无法访问，一般是由于扫描，P2P 应用，或者是防火墙阻断引起的。

诊断条目						诊断发生地址			
名字	数量	名字	物理地址	IP地址	数量				
TCP 重复的连接尝试	211	FF:FF:FF:FF:FF:FF	-	-	3,430				
TCP 重传数据包	432	52:54:4C:F0:D4:D0	52:54:4C:F0:D4:D0	-	829				
TCP 慢应答	17,792	52:54:4C:F0:5F:30	52:54:4C:F0:5F:30	-	493				
TCP 重复的确认	555	52:54:4C:F0:EB:D9	52:54:4C:F0:EB:D9	-	461				
TCP 重复的连接尝试	3,734	52:54:4C:F0:D2:7B	52:54:4C:F0:D2:7B	-	449				
IP 地址冲突	3,430	52:54:4C:F0:ED:67	52:54:4C:F0:ED:67	-	430				
ICMP 主机不可达	9	52:54:4C:F0:E9:89	52:54:4C:F0:E9:89	-	425				
ICMP 端口不可达	295	52:54:4C:F0:5F:11	52:54:4C:F0:5F:11	-	338				
数据链路层	330	58:1F:AA:8F:8A:76	58:1F:AA:8F:8A:76	-	2				
ARP 请求风暴	8	00:24:7E:68:0D:43	00:24:7E:68:0D:43	-	1				
ARP 扫描	322	00:0F:EA:1A:8B:57	00:0F:EA:1A:8B:57	-	1				

诊断事件						诊断事件	
严重程度	类型	层别	事件描述	源IP地址	源物理地址	目标IP地址	目标物理地址
安全	网络层	数据包 212284 与 212364 发生地址冲突	170.151.24.203	52:54:4C:F0:EB:D9	100.100.30.21	FF:FF:FF:FF:FF:FF	-
安全	网络层	数据包 214132 与 214180 发生地址冲突	170.151.24.203	52:54:4C:F0:EB:D9	100.100.30.21	FF:FF:FF:FF:FF:FF	-
安全	网络层	数据包 214506 与 214522 发生地址冲突	170.151.24.203	52:54:4C:F0:EB:D9	100.100.30.21	FF:FF:FF:FF:FF:FF	-
安全	网络层	数据包 215114 与 215355 发生地址冲突	170.151.24.203	52:54:4C:F0:FR:D9	100.100.30.21	FF:FF:FF:FF:FF:FF	-

在上图中，我们可以看到网络中存在大量的 IP 地址冲突。这些会造成网络广播包，小包增多，影响网络质量。建议绑定 IP 地址，便于故障主机的定位与排查。

诊断事件						诊断事件	
严重程度	类型	层别	事件描述	源IP地址	源物理地址	目标IP地址	目标物理地址
安全	数据链路层	ARP地址扫描(主机的MAC为 52:54:4C:F0:5F:11, IP地址为 170.151.24.203)	170.151.24.203	52:54:4C:F0:5F:11	-	10	FF:FF:FF:FF:FF:FF
安全	数据链路层	ARP地址扫描(主机的MAC为 52:54:4C:F0:E9:89, IP地址为 170.151.24.203)	170.151.24.203	52:54:4C:F0:E9:89	-	10	FF:FF:FF:FF:FF:FF
安全	数据链路层	ARP地址扫描(主机的MAC为 52:54:4C:F0:5F:30, IP地址为 170.151.24.203)	170.151.24.203	52:54:4C:F0:5F:30	-	10	FF:FF:FF:FF:FF:FF
安全	数据链路层	ARP地址扫描(主机的MAC为 52:54:4C:F0:D4:0B, IP地址为 203.24.151.170)	203.24.151.170	52:54:4C:F0:D4:0B	-	10	FF:FF:FF:FF:FF:FF
安全	数据链路层	ARP地址扫描(主机的MAC为 52:54:4C:F0:EB:09, IP地址为 170.151.24.203)	170.151.24.203	52:54:4C:F0:EB:09	-	10	FF:FF:FF:FF:FF:FF
安全	数据链路层	ARP地址扫描(主机的MAC为 52:54:4C:F0:FR:67, IP地址为 170.151.24.203)	170.151.24.203	52:54:4C:F0:FR:67	-	10	FF:FF:FF:FF:FF:FF

在自动诊断中，我们还发现了大量的 ARP 扫描，经进一步分析，我们发现造成 ARP 扫描的地址主要为 170.151.24.203 及 203.24.151.170，其扫描对象也集中在 100.100.30.0 网段的服务器，而在上面的 IP 地址冲突检测中，也正是这两个 IP 主机出现地址冲突现象，并引发大量的广播包。建议检查此两个主机安全状况，并绑定其 IP 地址。

3) 流量应用分析

Top 10 IP地址统计

名字	接收百分比	发送百分比	字节	数据包
GAOQY-PC.hebngc.com	5.339%		94.661%	63.541 MB 127,133
10.69.21.132	97.743%		2.257%	37.010 MB 39,604
21tb-dx2.cneln.net	2.208%		97.792%	36.963 MB 39,427
oa.hebngc.com	29.618%		70.382%	21.065 MB 34,501
10.69.21.130	70.931%		29.069%	12.355 MB 20,897
123.172.151.150	97.476%		2.524%	8.065 MB 8,542
U8	30.761%		69.239%	7.811 MB 20,394
10.69.20.153	82.805%		17.195%	7.693 MB 11,695
10.69.20.14	77.748%		22.252%	7.200 MB 18,018
10.69.21.8	69.193%		30.807%	6.330 MB 16,145

返回

在 IP 地址的 TOP 前十名我们可见到排名最高的 ip 主机。

Top 10 应用层协议

名字	百分比	字节	数据包
HTTP		46.698%	84.467 MB 130,064
UDP - Other		34.818%	62.979 MB 126,075
HTTPS		8.382%	15.161 MB 29,045
MSSQL		4.343%	7.856 MB 21,217
TCP - Other		3.049%	5.515 MB 14,061
RTP		1.114%	2.016 MB 3,729
CIFS		0.191%	353.330 KB 1,417
HTTP Proxy		0.147%	271.854 KB 1,722
Kerberos		0.140%	259.973 KB 1,136
DNS		0.128%	236.676 KB 1,866

返回

在应用协议排名中,我们可以看到最高的依然是 HTTP, UDP-OTHER, HTTPS, MSSQL, TCP-OTHER。



在 UDP-OTHER 中, 我们经分析得知, UDP 的应用主要在 GAOQY-PC.Hebngc.com,进行数据的传输。

4) 安全异常分析



在安全分析的疑似蠕虫病毒分析中，我们可以看到 GAOQY-PC.Hebngc.com 的身影。但是经分析，我们发现与此主机同的 IP 地址没有规律性，且端口不固定，主要集中在 8080, 443 等常用端口上。并非是蠕虫病毒。而从矩阵视图可看到，发送字节 60M，接收却只有 3M 多。但是右侧的连接分布却从另一面验证我们的猜测。

建议用户在使用科来系统时，根据网路的实际状况，适当调整安全模块的判定阀值，以提高工作效率。

4. 地级市二专线

1) 概要统计

流量统计	字节数	数据包数	利用率	每秒位数	每秒包数
总流量	138,669 MB	300,068	5.625%	112.504 Kbps	110
广播流量	1.019 MB	13,086	0.552%	11.040 Kbps	10
多播流量	384.106 KB	3,116	0.068%	1.360 Kbps	1
平均包长				484.572 字节	
数据包大小分布	字节数	数据包数	利用率	每秒位数	每秒包数
<=64	6.022 MB	98,673	0.845%	16.896 Kbps	33
65-127	5.918 MB	74,749	1.828%	36.560 Kbps	54
128-255	3.619 MB	22,293	1.003%	20.056 Kbps	17
256-511	4.404 MB	12,899	0.347%	6.944 Kbps	2
512-1023	7.929 MB	11,498	0.256%	5.120 Kbps	1
1024-1517	33.631 MB	26,667	1.346%	26.928 Kbps	3
>=1518	77.145 MB	53,289	0.000%	0 bps	0

平均包长 482.572 字节，比正常范围偏小。网络利用率多次达到 100%，并在中间持续很长一段时间，网络拥塞严重。小包数量明显大于大包数量，偏多，需进一步排查。

2) 诊断异常分析

诊断条目		诊断发生地址			
名字	数量	名字	物理地址	IP地址	数量
所有诊断	22,214	ad1.hebngc.com	00:21:D8:5D:8A:C6	192.168.0.249	165
应用层	726	ad2.hebngc.com	00:21:D8:5D:8A:C2	192.168.0.248	88
DNS服务器慢响应	17	10.69.40.109	00:1E:90:78:45:FA	10.69.40.109	66
DNS主机或域名不存在	253	10.69.40.63	00:15:58:DC:B2:09	10.69.40.63	46
DNS服务器错误	20	10.69.40.142	00:1E:90:78:4D:BE	10.69.40.142	44
POP3 服务器慢响应	4	10.69.40.8	00:1E:90:78:4E:EC	10.69.40.8	44
POP3 服务器返回错误	4	10.69.40.22	00:1B:B9:D5:CE:61	10.69.40.22	44
HTTP 客户端错误	7	10.69.42.22	00:19:21:AE:78:EE	10.69.42.22	2
HTTP 可疑会话	3	10.69.40.14	00:1E:90:76:9D:2B	10.69.40.14	2
HTTP 请求没找到	38	10.69.40.73	44:37:E6:47:C1:A1	10.69.40.73	2
HTTP 服务器错误	7	10.69.40.91	44:37:E6:47:C1:F9	10.69.40.91	1
HTTP 服务器慢响应	373	10.69.40.12	00:1E:90:78:42:46	10.69.40.12	1
TCP 重复的连接尝试	16,257	10.69.40.3	00:1E:90:78:4E:D9	10.69.40.3	1
TCP 重传数据包	582				
TCP 慢应答	472				
TCP 重复的确认	14,642				
TCP 重复的扫描	560				
TCP 端口扫描	1				

在本专线中，HTTP 服务器相应慢，TCP 慢应答，TCP 重传，等进一步验证我们刚才对本专线网络拥塞的判断。同时，DNS 主机域名不存在错误报告，在右侧的诊断发证地址中，我们找到 ad1.hebngc.com 等地址。建议查看这些地址的 DNS 设置是否正确。

诊断条目		诊断发生地址			
名字	数量	名字	物理地址	IP地址	数量
TCP 端口扫描	1	52:54:4C:F0:ED:67	52:54:4C:F0:ED:67	-	496
IP 非法的校验和	4,816	52:54:4C:F0:5F:11	52:54:4C:F0:5F:11	-	503
IP TTL 太小	5	52:54:4C:F0:5F:30	52:54:4C:F0:5F:30	-	520
IP 地址冲突	4,389	52:54:4C:F0:EB:09	52:54:4C:F0:EB:09	-	602
ICMP 目的不可达	19	52:54:4C:F0:D2:7B	52:54:4C:F0:D2:7B	-	607
ICMP 端口不可达	402	52:54:4C:F0:D4:0D	52:54:4C:F0:D4:0D	-	1,059
数据链路层	415	52:54:4C:F0:E9:89	52:54:4C:F0:E9:89	-	602
ARP 请求风暴	1	FF:FF:FF:FF:FF:FF	FF:FF:FF:FF:FF:FF	-	4,389
ARP 扫描	413				
ARP 太多的主动应答	1				
诊断事件					
严重程度	类型	层别	事件描述	源IP地址	源物理地址
安全	网络安全	数据包 181415 与 181581	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67
安全	网络安全	数据包 182876 与 183043	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67
安全	网络安全	数据包 184594 与 184661	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67
安全	网络安全	数据包 186273 与 186435	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67
安全	网络安全	数据包 187838 与 187890	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67
安全	网络安全	数据包 189305 与 189448	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67
安全	网络安全	数据包 191270 与 191464	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67
安全	网络安全	数据包 193288 与 193416	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67
安全	网络安全	数据包 195294 与 195408	发生地址冲突	170.151.24.203	52:54:4C:F0:ED:67

同地级市一类似，地级市二专线也存在大量的 IP 地址冲突和 ARP 扫描。且地址主要为 170.151.24.203

及 203.24.151.170。建议对此两台主机进行仔细检测并绑定 IP。

3) 流量异常分析

Top 10 应用层协议

名字	百分比	字节	数据包
HTTP	57.550%	79.804 MB	113,587
UDP - Other	17.166%	23.804 MB	54,763
TCP - Other	15.501%	21.495 MB	58,263
MSSQL	4.376%	6.068 MB	17,953
HTTPS	1.583%	2.195 MB	6,070
Yahoo Messenger	0.524%	744.460 KB	9,232
CIFS	0.510%	724.015 KB	4,038
Kerberos	0.449%	637.611 KB	3,642
QQ	0.402%	570.843 KB	4,841
NetBIOS	0.262%	371.707 KB	3,847

返回

在应用协议排名中，我们可以看到最高的依然是 HTTP, UDP-OTHER, HTTPS, MSSQL, TCP-OTHER。



在上面的协议分析视图中，我们发现了 BT 及 PPLIVE 协议。这些协议均用于 P2P 下载。无论是上传还是下载，P2P 都会占用大量带宽，影响网络传输质量。

Top 10 本地IP地址统计

名字	接收百分比	发送百分比	字节	数据包
ZHANGYUWEN-PC.hebngc.com	82.185%	17.815%	61.857 MB	78,067
10.69.40.126	7.228%	92.772%	22.166 MB	44,580
10.69.40.105	26.616%	73.384%	17.589 MB	39,314
U8	37.960%	62.040%	6.377 MB	18,953
10.69.11.83	63.331%	36.669%	6.311 MB	18,670
10.69.40.8	75.999%	24.001%	5.637 MB	14,376
10.69.40.63	89.992%	10.008%	5.147 MB	8,710
192.168.0.242	45.063%	54.937%	4.591 MB	10,569
ad2.hebngc.com	51.159%	48.841%	3.177 MB	26,986
10.69.42.22	52.607%	47.393%	2.812 MB	10,018

返回

上图显示的是流量最高的本地主机及他们的收发比。

4) 安全异常分析

疑似蠕虫病毒分析

名字	数据包	每秒字节	每秒数据包	接收数据包	发送数据包	包收发比	IP会话
ZHANGYUWEN-PC.hebngc....	78,067	4,086 Kbps	12	46,910	31,157	0.66	184
10.69.40.105	39,314	242 Bps	3	14,629	24,685	1.69	220
10.69.40.60	12,907	64 Bps	1	2,717	10,190	3.75	195

IP会话

节点1->	<-节点2	持续时间	字节	字节->	<-字节	数据包	数据包->	开始发包时
110.231.118.55	ZHANGYUWEN-PC....	00:00:09	704 B	704 B	0 B	11	11	15:26
ZHANGYUWEN-PC.hebngc....	fds.pps24.com	00:00:00	630 B	64 B	566 B	2	1	15:30
ZHANGYUWEN-PC.hebngc....	fds.pps24.net	00:00:00	630 B	64 B	566 B	2	1	15:30
ZHANGYUWEN-PC.hebngc....	239.255.255.250	00:00:01	34,863 KB	34,863 KB	0 B	200	200	15:30
ZHANGYUWEN-PC.hebngc....	mail.hebngc.com	00:00:01	320 B	320 B	0 B	4	4	15:30
ZHANGYUWEN-PC.hebngc....	10.69.40.1	00:00:00	1,654 KB	1,582 KB	74 B	10	9	15:30
ZHANGYUWEN-PC.hebngc....	110.188.2.184	00:00:00	159 B	159 B	0 B	1	1	15:30
ZHANGYUWEN-PC.hebngc....	110.188.2.187	00:00:00	150 B	150 B	0 B	1	1	15:30
ZHANGYUWEN-PC.hebngc....	119.188.40.17	00:00:00	301 B	158 B	143 B	2	1	15:30
ZHANGYUWEN-PC.hebngc....	125.46.40.6	00:00:00	302 B	159 B	143 B	2	1	15:30
ZHANGYUWEN-PC.hebngc....	125.46.40.11	00:00:00	293 B	150 B	143 B	2	1	15:30
ZHANGYUWEN-PC.hebngc....	125.46.40.37	00:00:00	301 B	158 B	143 B	2	1	15:30
ZHANGYUWEN-PC.hebngc....	119.188.40.49	00:00:00	304 B	161 B	143 B	2	1	15:30
ZHANGYUWEN-PC.hebngc....	119.147.144.169	00:00:00	304 B	161 B	143 B	2	1	15:30
ZHANGYUWEN-PC.hebngc....	119.147.144.137	00:00:00	301 B	158 B	143 B	2	1	15:30

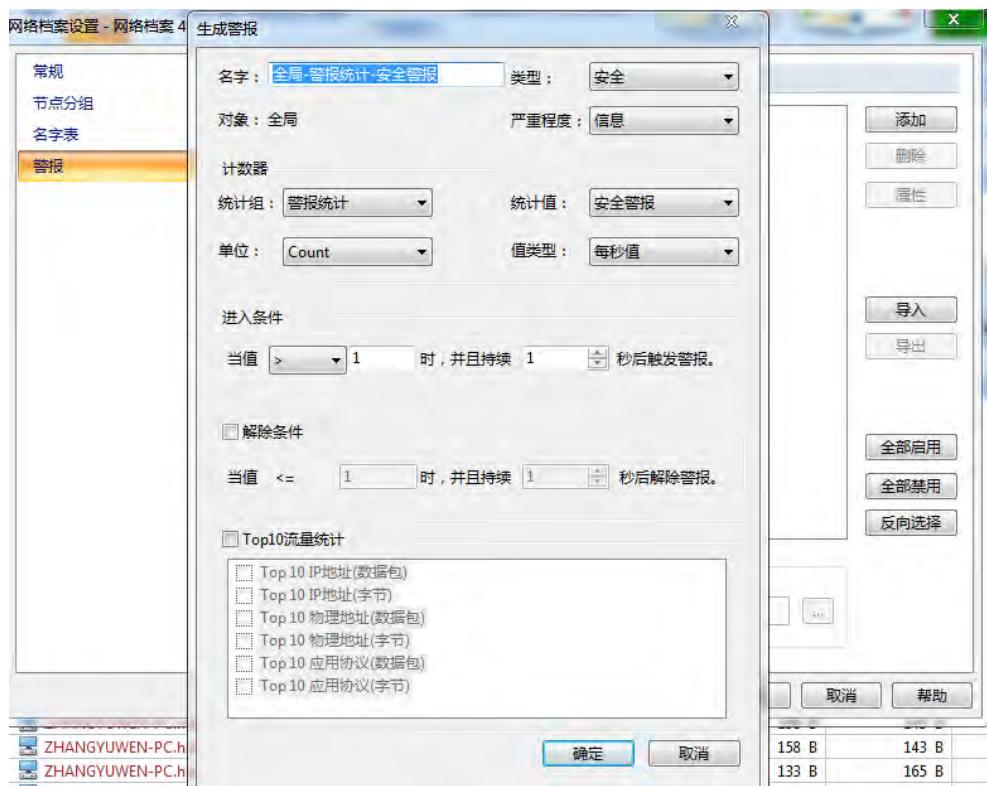
疑似发起DoS攻击

名字	数据包	广播数据包	多播数据包	接收数据包	发送数据包	包收发比	IP会话
ZHANGYUWEN-PC.hebngc....	78,067	26	313	46,910	31,157	0.66	184
10.69.40.105	39,314	4	239	14,629	24,685	1.69	220

IP会话

节点1->	<-节点2	持续时间	字节	字节->	<-字节	数据包	数据包->	开始发包时
110.231.118.55	ZHANGYUWEN-PC....	00:00:09	704 B	704 B	0 B	11	11	15:26
ZHANGYUWEN-PC.hebngc....	fds.pps24.com	00:00:00	630 B	64 B	566 B	2	1	15:30:0
ZHANGYUWEN-PC.hebngc....	fds.pps24.net	00:00:00	630 B	64 B	566 B	2	1	15:30:0
ZHANGYUWEN-PC.hebngc....	239.255.255.250	00:00:01	34,863 KB	34,863 KB	0 B	200	200	15:30:0
ZHANGYUWEN-PC.hebngc....	mail.hebngc.com	00:00:01	320 B	320 B	0 B	4	4	15:30:0
ZHANGYUWEN-PC.hebngc....	10.69.40.1	00:00:00	1,654 KB	1,582 KB	74 B	10	9	15:30:0
ZHANGYUWEN-PC.hebngc....	110.188.2.184	00:00:00	159 B	159 B	0 B	1	1	15:30:2
ZHANGYUWEN-PC.hebngc....	110.188.2.187	00:00:00	150 B	150 B	0 B	1	1	15:30:2
ZHANGYUWEN-PC.hebngc....	119.188.40.17	00:00:00	301 B	158 B	143 B	2	1	15:30:2
ZHANGYUWEN-PC.hebngc....	125.46.40.6	00:00:00	302 B	159 B	143 B	2	1	15:30:2
ZHANGYUWEN-PC.hebngc....	125.46.40.11	00:00:00	293 B	150 B	143 B	2	1	15:30:2
ZHANGYUWEN-PC.hebngc....	125.46.40.37	00:00:00	301 B	158 B	143 B	2	1	15:30:2
ZHANGYUWEN-PC.hebngc....	119.188.40.49	00:00:00	304 B	161 B	143 B	2	1	15:30:2
ZHANGYUWEN-PC.hebngc....	119.147.144.169	00:00:00	304 B	161 B	143 B	2	1	15:30:2
ZHANGYUWEN-PC.hebngc....	119.147.144.137	00:00:00	301 B	158 B	143 B	2	1	15:30:2

经分析发现，与地级市一的情况类似，都是由于数据的传输，造成首发比失调，进而触发报警。建议更改报警阀值。

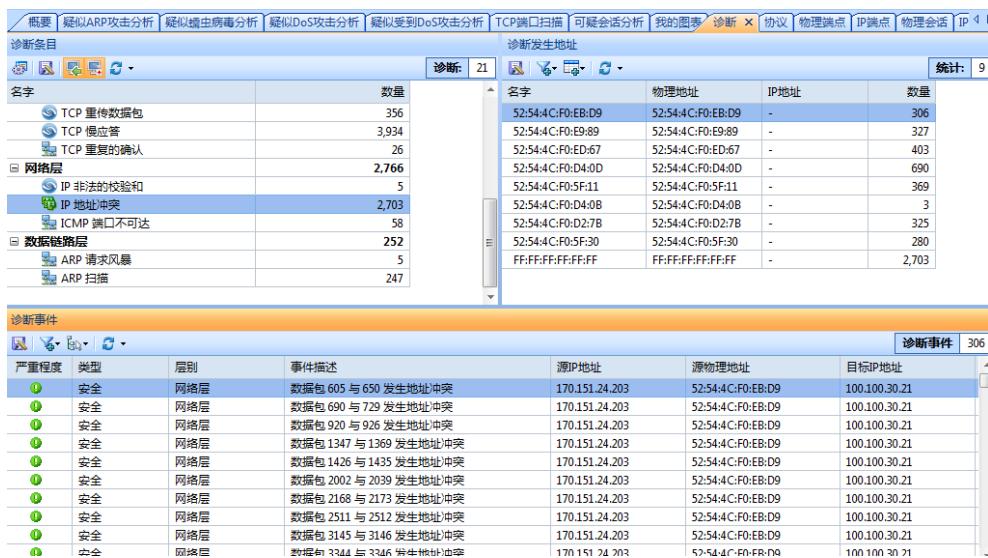


5. 地级市三等专线

1) 安全分析

总连接统计		子口数	数据包数	利用率	每秒位数	每秒包数
总流量		50,664 MB	81,174	1.263%	25.264 Kbps	16
广播流量		577.623 KB	8,376	0.154%	3.072 Kbps	6
多播流量		108.729 KB	871	0.026%	528 bps	1
平均包长					654.454 字节	
数据包大小分布		字节数	数据包数	利用率	每秒位数	每秒包数
<=64		2.120 MB	34,733	0.256%	5.120 Kbps	10
65-127		568.186 KB	7,051	0.112%	2.248 Kbps	3
128-255		435.508 KB	2,487	0.000%	0 bps	0
256-511		1.172 MB	3,462	0.160%	3.192 Kbps	1
512-1023		1.392 MB	1,976	0.735%	14.704 Kbps	2
1024-1517		2.360 MB	2,011	0.000%	0 bps	0
>=1518		42.640 MB	29,454	0.000%	0 bps	0

地级市三专线包长利用率基本正常。但偶尔高峰期利用率可达 90%以上，一般在 30%左右。建议多次采样，以判断地级市三专线利用率常规阀值。



地级市三专线也存在 IP 地址冲突的问题。且地址主要为 170.151.24.203 及 203.24.151.170。建议排查线路，弄清此两个 IP 的真实应用及网络拓扑。

这几条专线，数据量不大，网络流量异常反映不明显，暂无数据异常。建议长期检测，以逐个分清各专线数据流高峰峰值，及其他网络异常状况。

12.4. 测试总结

通过科来网络分析系统对 Internet 出口及各专线进行的评估分析，我们对该链路运行情况有了深入的了解，对此次测试结果总结如下。

联通 Internet 出口，沙河，裕园专线链路拥塞，传输质量较差

该链路在测试过程中一直处于非常拥塞的状态，流量很大，传输质量较差，丢包严重。

沙河及其他少数用户利用 P2P 软件下载挤占网络带宽

少数用户占据大量网络带宽，这些用户利用 P2P 软件下载，是造成网络拥塞的一定原因。

HTTP,HTTPS,HTTP-OTHER,UDP-OTHER 应用挤占大量带宽

由于这些都是正常办公时使用的协议，建议增加相应专线带宽，以缓解网络拥塞的情况。

危害安全的网络行为

此次测试中发现了一些危害网络安全的行为如蠕虫、ARP 攻击等网络异常行为，需针对异常主机进行进一步的分析。

IP 地址冲突

采样中 170.151.24.203 及 203.24.151.170。建议排查主机，线路，弄清此两个 IP 的真实应用及网络拓扑。

由 Internet 访问内部服务器的慢应答

对于电信专线中，出现的慢应答显现，建议检测中间网络传输设备的负载状况及防火墙，IPS 等安全设备的正确配置。

PPTP 隧道协议

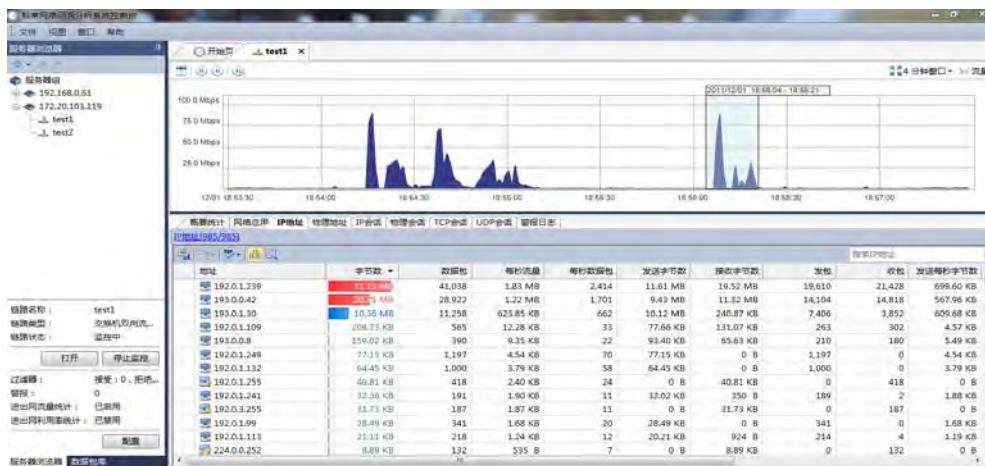
采样中，PPTP 隧道协议的数量大于 ssl vpn 隧道协议的数量。由于 pptp 是不加密的隧道协议，存在一定的安全隐患，虽然走的是专线，但在联通或电信的机房内仍有泄密的可能。建议采用专业的 VPN 设备进行加密传输。

第二章 网络安全分析

1. 某证券公司回溯分析案例

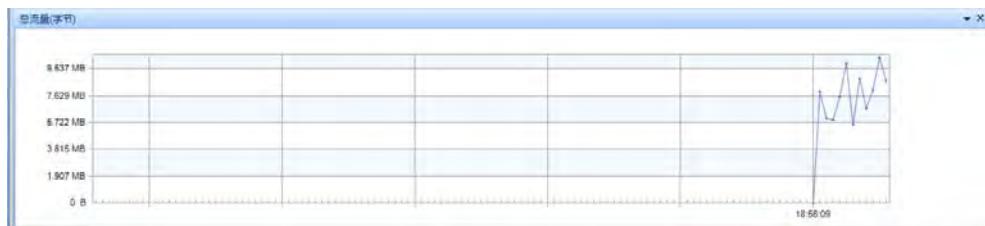
1.1. 分析情况

采用科来回溯分析系统来查看和分析服务应用中一些数据流和各服务器的流量情况，并通过控制台选择一部分流量异常的数据包并下载分析。



1. 带宽利用率满载

某证券公司为办公网所划分的带宽总体为 20Mbps，其中 10Mbps 为电信，另一条 10Mbps 是联通线路。经过测试发现某证券公司办公网利用率时刻处于满载状态。



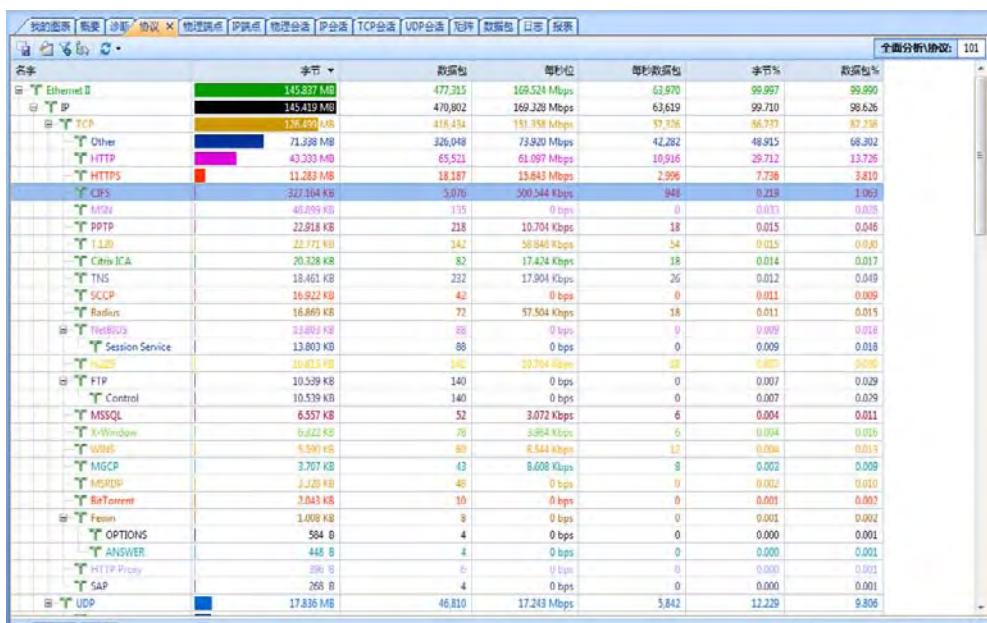
流量统计	字节数	数据包数	利用率	每秒位数	每秒包数
总流量	98.138 MB	183,776	100.000%	72.546 Mbps	16,506
广播流量	79.305 KB	833	0.054%	21.504 Kbps	42
多播流量	17.080 KB	226	0.000%	0 bps	0
平均包长					559.949 字节

对问题主机进行定位，发现 192.168.0.79 等节点发现 UDP 包大量传输，经鉴定是下载行为。因为这些主机的下载，导致了带宽利用率被占满，其他业务和应用变慢、或网络时断时续。

经过分析后，某证券公司技术人员进入路由器管理，对响应的节点进行了控制和带宽上的限制，在行为上添加了一些策略。最后，问题得到了解决。

2. 蠕虫病毒

对某证券公司的互联网（交易网）进行整体检测，该网络总体带宽为 300 兆，承载公司所有的互联网业务。打开“协议”页面，发现 CIFS 协议的数据包相对较多。



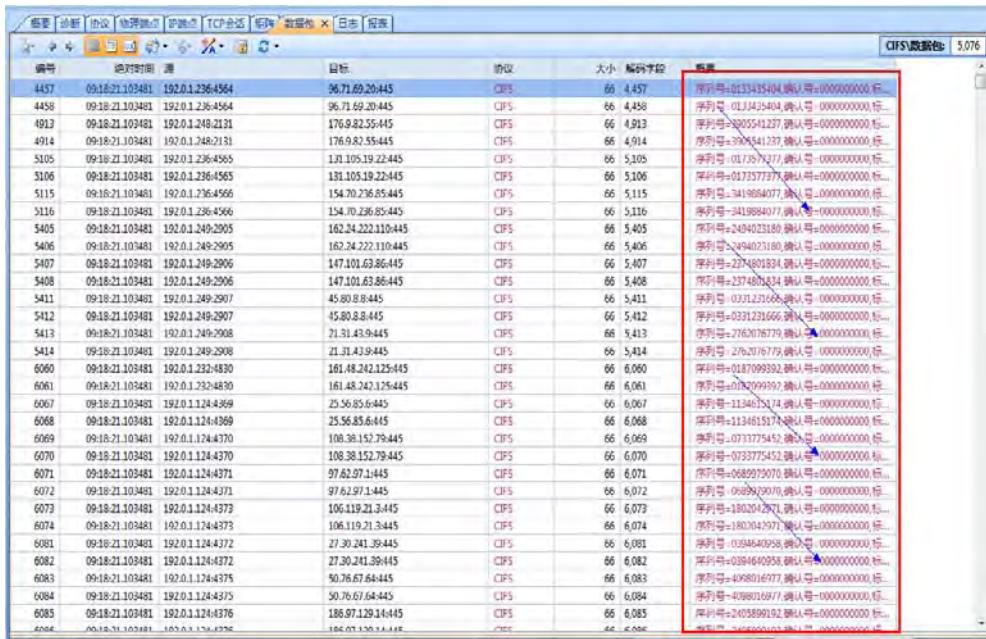
定位到 CIFS 协议发现数据包都是 192.0.1.0/24 这个网段发送的，CIFS 协议是 WINDOWS 主机之间进行网络文件共享一种互联网协议。

名字	字节 ▾	数据包	接收字节	接收数据包	发送字节	发送数据包	CIFS会话 1.73	
							组内字节	组内数据包
Internet地址	327,164 KB	5,076	0 B	0	1,934 KB	2,960	136,383 KB	2,111
美国	22,746 KB	5,076	0 B	0	39,703 KB	616	0 B	0
192.0.1.244	616	0 B	0	0	39,703 KB	600	0 B	0
192.0.1.250	600	0 B	0	0	38,672 KB	600	0 B	0
192.0.1.121	600	0 B	0	0	38,672 KB	600	0 B	0
192.0.1.236	598	0 B	0	0	38,543 KB	598	0 B	0
192.0.1.124	586	0 B	0	0	37,770 KB	586	0 B	0
192.0.1.248	582	0 B	0	0	37,512 KB	582	0 B	0
192.0.1.232	544	0 B	0	0	35,063 KB	544	0 B	0
192.0.1.249	514	0 B	0	0	33,129 KB	514	0 B	0
192.0.1.123	436	0 B	0	0	28,102 KB	436	0 B	0
28.115.111.109	264 B	4	264 B	4	0 B	0	0 B	0
208.12.229.126	264 B	4	264 B	4	0 B	0	0 B	0
199.109.7.34	264 B	4	264 B	4	0 B	0	0 B	0
67.116.139.96	264 B	4	264 B	4	0 B	0	0 B	0
64.118.94.14	264 B	4	264 B	4	0 B	0	0 B	0
26.88.229.124	264 B	4	264 B	4	0 B	0	0 B	0
34.89.108.48	264 B	4	264 B	4	0 B	0	0 B	0
98.84.161.77	264 B	4	264 B	4	0 B	0	0 B	0
152.8.4.81	264 B	4	264 B	4	0 B	0	0 B	0
214.308.135.121	264 B	4	264 B	4	0 B	0	0 B	0
71.27.240.124	264 B	4	264 B	4	0 B	0	0 B	0
131.105.19.22	264 B	4	264 B	4	0 B	0	0 B	0
156.85.95.57	264 B	4	264 B	4	0 B	0	0 B	0
63.91.200.84	264 B	4	264 B	4	0 B	0	0 B	0
170.48.192.65	264 B	4	264 B	4	0 B	0	0 B	0
55.110.205.117	264 B	4	264 B	4	0 B	0	0 B	0
164.94.176.7	264 B	4	264 B	4	0 B	0	0 B	0
209.21.92.123	264 B	4	264 B	4	0 B	0	0 B	0
11.126.147.96	264 B	4	264 B	4	0 B	0	0 B	0
73.78.27.8	264 B	4	264 B	4	0 B	0	0 B	0

192.0.1.0/24 这个网段不断的发送数据包，只发送、不接收，端口有顺序的变换，数据包大小相同，目的地址端口号都是 445。

节点 ▾	<- 字节 ▾	数据包	字符 ▾	协议	持续时间	字符 ▾	CIFS会话 871	
							<- 字节	数据包 ▾
192.0.1.244-4477	191.22.227.80:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.232-4987	34.103.16.74:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.236-4668	87.115.248.123:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.124-4558	198.29.40.107:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.121-4442	78.67.243.119:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.124-4559	85.113.130.26:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.249-3036	191.3.112.108:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.121-1443	79.7.196.83:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.121-1444	113.70.51.91:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.173.4214	23.50.160.46:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.123-4215	57.82.0.57:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.128-4216	223.102.141.92:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.124-4560	143.14.189.14:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.249-3037	144.25.238.8:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3147	186.100.119.0:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3148	119.7.2.146:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3149	121.37.138.90:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3150	109.57.64.12:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3151	180.9.5.17:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3152	118.90.43.10:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3153	33.3.197.62:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3154	165.112.4.36:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3155	147.10.1.183.77:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3156	147.124.134.54:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3157	169.23.176.117:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3158	222.19.67.105:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3159	122.108.109.22:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3160	151.29.165.15:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3161	216.30.219.10:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3162	91.66.182.92:445	2	132 B	CIFS	00:00:00	132 B	0 B	2
192.0.1.250-3163	222.110.222.41:445	2	132 B	CIFS	00:00:00	132 B	0 B	2

随后打开“数据包”页面，发现所有 CIFS 包在概要里的确认号都是 000000000000



编号	绝对时间	源	目标	协议	大小	编码字符	摘要
4457	09:18:21.103481	192.0.1.236.4564	96.71.69.20445	CIFS	66	4,457	序列号=0133435404, 源IP号=0000000000 等...
4458	09:18:21.103481	192.0.1.236.4564	96.71.69.20445	CIFS	66	4,458	序列号=0133435404, 源IP号=0000000000 等...
4913	09:18:21.103481	192.0.1.248.2131	176.9.82.55.445	CIFS	66	4,913	序列号=23005541237, 源IP号=0000000000 等...
4914	09:18:21.103481	192.0.1.248.2131	176.9.82.55.445	CIFS	66	4,914	序列号=23005541237, 源IP号=0000000000 等...
5105	09:18:21.103481	192.0.1.236.4565	131.105.19.22445	CIFS	66	5,205	序列号=0117917417, 源IP号=0000000000 等...
5106	09:18:21.103481	192.0.1.236.4565	131.105.19.22445	CIFS	66	5,106	序列号=0117917417, 源IP号=0000000000 等...
5115	09:18:21.103481	192.0.1.236.4565	154.70.236.85.445	CIFS	66	5,115	序列号=3419884077, 源IP号=0000000000 等...
5116	09:18:21.103481	192.0.1.236.4566	154.70.236.85.445	CIFS	66	5,116	序列号=3419884077, 源IP号=0000000000 等...
5405	09:18:21.103481	192.0.1.249.2905	162.24.222.110.445	CIFS	66	5,405	序列号=2404023168, 源IP号=0000000000 等...
5406	09:18:21.103481	192.0.1.249.2905	162.24.222.110.445	CIFS	66	5,406	序列号=2404023168, 源IP号=0000000000 等...
5407	09:18:21.103481	192.0.1.249.2906	147.101.63.86.445	CIFS	66	5,407	序列号=227401834, 源IP号=0000000000 等...
5408	09:18:21.103481	192.0.1.249.2906	147.101.63.86.445	CIFS	66	5,408	序列号=227401834, 源IP号=0000000000 等...
5411	09:18:21.103481	192.0.1.249.2907	45.89.8.845	CIFS	66	5,411	序列号=031231668, 源IP号=0000000000 等...
5412	09:18:21.103481	192.0.1.249.2907	45.89.8.845	CIFS	66	5,412	序列号=031231668, 源IP号=0000000000 等...
5413	09:18:21.103481	192.0.1.249.2908	21.31.43.9445	CIFS	66	5,413	序列号=27620176779, 源IP号=0000000000 等...
5414	09:18:21.103481	192.0.1.249.2908	21.31.43.9445	CIFS	66	5,414	序列号=27620176779, 源IP号=0000000000 等...
6060	09:18:21.103481	192.0.1.232.4830	161.48.242.125.445	CIFS	66	6,060	序列号=018709932, 源IP号=0000000000 等...
6061	09:18:21.103481	192.0.1.232.4830	161.48.242.125.445	CIFS	66	6,061	序列号=018709932, 源IP号=0000000000 等...
6067	09:18:21.103481	192.0.1.124.4369	25.56.85.6445	CIFS	66	6,067	序列号=1134615174, 源IP号=0000000000 等...
6068	09:18:21.103481	192.0.1.124.4369	25.56.85.6445	CIFS	66	6,068	序列号=1134615174, 源IP号=0000000000 等...
6069	09:18:21.103481	192.0.1.174.4370	106.38.152.79.445	CIFS	66	6,069	序列号=073375452, 源IP号=0000000000 等...
6070	09:18:21.103481	192.0.1.174.4370	106.38.152.79.445	CIFS	66	6,070	序列号=073375452, 源IP号=0000000000 等...
6071	09:18:21.103481	192.0.1.174.4371	97.62.97.1145	CIFS	66	6,071	序列号=0682970070, 源IP号=0000000000 等...
6072	09:18:21.103481	192.0.1.174.4371	97.62.97.1145	CIFS	66	6,072	序列号=0682970070, 源IP号=0000000000 等...
6073	09:18:21.103481	192.0.1.174.4373	106.419.21.3-445	CIFS	66	6,073	序列号=180201291, 源IP号=0000000000 等...
6074	09:18:21.103481	192.0.1.174.4373	106.119.21.3-445	CIFS	66	6,074	序列号=180201291, 源IP号=0000000000 等...
6081	09:18:21.103481	192.0.1.174.4372	27.30.241.39.445	CIFS	66	6,081	序列号=0304640538, 源IP号=0000000000 等...
6082	09:18:21.103481	192.0.1.174.4372	27.30.241.39.445	CIFS	66	6,082	序列号=0304640538, 源IP号=0000000000 等...
6083	09:18:21.103481	192.0.1.174.4375	50.76.67.64.445	CIFS	66	6,083	序列号=4098016977, 源IP号=0000000000 等...
6084	09:18:21.103481	192.0.1.174.4375	50.76.67.64.445	CIFS	66	6,084	序列号=4098016977, 源IP号=0000000000 等...
6085	09:18:21.103481	192.0.1.174.4376	186.97.129.14.445	CIFS	66	6,085	序列号=2405899192, 源IP号=0000000000 等...
6086	09:18:21.103481	192.0.1.174.4376	186.97.129.14.445	CIFS	66	6,086	序列号=2405899192, 源IP号=0000000000 等...

1.2. 分析总结

由此可以推断，192.0.1.0/24 这个网段中的某主机，很可能中了蠕虫，在对网络进行实时的扫描。发现这一问题后，网管们比较重视，对该网段进行了检查，在交换机上对几个地址进行了跟踪和查询。最终确定是蠕虫感染并彻底的检查和病毒查杀。

2. 某电信IDC机房——托管服务器异常行为监控

2.1. 案例背景

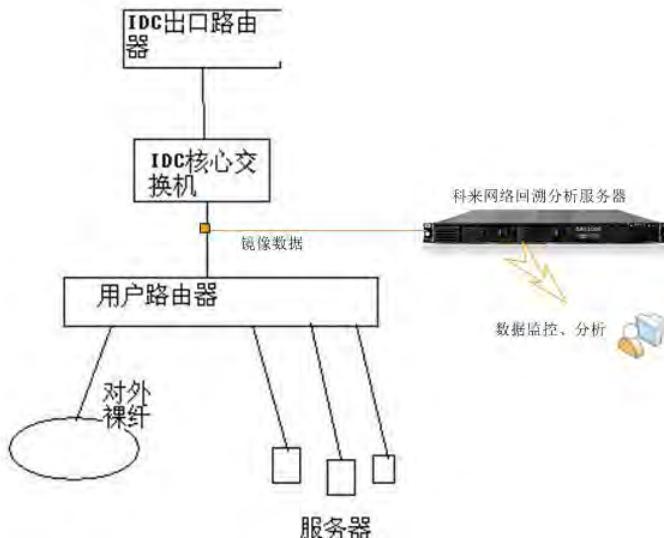
某电信 IDC 机房为数千家政企用户提供服务器托管业务，有些用户却利用托管服务器为跳板从事一些非业务流量，比如代理上网、视频或下载，这不仅增加了 IDC 出口的负担，也为 IDC 内部网络的安全带来了隐患。

而 IDC 机房现有的网管及分析工具，对用户的这种用户非业务行为往往是事后才能发现，而且缺乏足够的历史数据对过去发生的网络事件进行回溯，无法快速的通过量化的证据对这些用户的行为进行通告和规范。

科来网络分析系统通过数据包层面的分析，能够对用户的网络行为进行可视化的监控、分析，并支持 7*24 小时不间断的长期分析，掌握接入流量的负载及每日、每周的流量基准、变化趋势，为用户异常行为发现提供基准，从而及时的发现网络的流量突发与拥塞，快速的定位到造成异常流量的 IP 源头，并对其行为进行深入分析，提供历史事件的证据，及时发现危害网络的病毒、攻击行为，避免网络瘫痪。

2.2. 设备部署

在交换机上启动镜像功能，由科来网络回溯分析服务器采集用户接入 IDC 机房网络的流量，部署示意图如下：



通过网络分析对网络进行可视性分析和专家级的诊断，对 VPN 接入用户的网络行为进行 7*24 小时不间断的实时监控与分析，快速的发现异常行为的用户，并对潜在的安全隐患进行预警，并将关键的网络参数指标及原始数据包保存、记录，同时提供方便快捷的数据挖掘功能快速的对历史事件进行追溯，并能够通过智能专家诊断系统，快速的发现问题。

2.3. 托管服务器异常行为监控

IDC 机房主要业务为托管 web 服务器等业务，其他代理上网、P2P 下载都是非业务行为，而且将消耗出口带宽资源，影响正常业务的效率。通过科来网络回溯分析系统，我们可以快速的定位异常主机，通过其流量行为特征判断是否在从事非业务行为；通过一段时间的流量监控后，可以获取各种网络参数的监控基准，从而设置合理的告警阀值，实现对网络异常主动预警。

1. 快速的发现异常主机

我们可以找出任意时间段所有主机的流量统计，统计参数包括流量大小、数据包量、收发流量及 TCP 层关键参数，如下图所示：



通过上图，我们把流量最大的四台机器几个关键参数提取出来，如下所示：

IP 主机	流量	流量发收比	发 TCP 同步包	收 TCP 同步包
..211.66	180.23MB	0.46	1,044	283
..69.53	136.16MB	2.82	0	934
..69.32	71.57MB	28.52	0	475
..211.47	60.38MB	2.37	2	60

2. 关键参数解读：

流量：传输数据的大小，直接影响到总出口带宽，通过该参数，我们能够快速的发现对 IDC 出口影响最大的 IP 主机

流量发收比：上传流量和下载流量的比值，托管机房里的服务器，一般都是提供网络服务供他人下载，正常情况下，上传流量应该大于下载流量，而“流量发收比” = 上传流量/下载流量，该比值应该大于 1。流量最大的四台主机中，*.*.69.53、*.*.69.32 和*.*.211.47 的“流量发收比”都大于 1，符合服务器特征，而流量最大的*.*.211.66 该参数为 0.46，下载流量大于上传流量，可能存在非业务流量

“发 TCP 同步包”与“收 TCP 同步包”：网页、邮件等基于 TCP 传输的网络应用，在数据传输前必须先建立 TCP 连接，TCP 连接的第一步便是有客户端发送“TCP 同步包”给服务器，IDC 机房内的托管设备作为服务器，“收到 TCP 同步包”的数量应该远远大于“发 TCP 同步包”的数量。主机*.*.69.53、*.*.69.32 和*.*.211.47 “发 TCP 同步包”的数量都很少，“收 TCP 同步包”的数量较多，符合托管服务器的业务特征；而主机*.*.211.66 “发 TCP 同步包”的数量达到 1044，远超过“收 TCP 同步包” 283 的数量，这说明该主机向外发起大量的访问请求，不大符合托管服务器的特征。

综合以上几个关键参数，我们可以确认主机`*.*.211.66` 的行为可疑，极可能存在非业务流量，并且消耗了 IDC 出口较高的带宽。科来网络回溯分析系统保存了该主机通信的原始数据包，我们可以通过专家分析系统，对主机`*.*.211.66` 的行为进行深度的分析，进一步确认其是否异常。

3. 异常主机行为深度分析

1) 流量负载分析

发现主机`*.*.211.66` 存在异常后，将进一步对其行为进行深度的挖掘、分析：

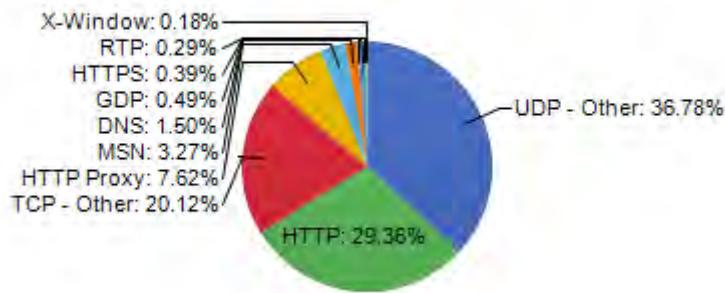
8月24号一天，该主机总共传输了 23.09G 字节的流量，“流量发送比”只有 0.38，“发 TCP 同步包”的数量超过 26 万个，明显对外发起了大量请求

地址	字节数	数据包	流量发收比	发TCP同步包	每秒流量
14.211.66	23.09 GB	59,457,871	0.38	265,544	281.42 KB

从流量分布特征来看，该主机流量主要集中在每天的白天上班时间短，消耗带宽接近 1.5Mbps 左右：



2) 流量成分分析：



该主机 top n 的流量成分如上图所示：

UDP-other: 36.78%

HTTP: 29.36%

TCP-other: 20.12%

HTTP-Proxy: 7.62

流量成分中存在大量的的 UDP-other 和 TCP-other，该主机可能存在大量的视频、下载等非业务流量。

4. 上网行为分析

DNS 日志分析：我们查看 11 月 25 号 56 这一分钟该主机的 DNS 记录，可以看到，主机 *.*.211.66 在一分钟内，总共进行了 124 次域名解析请求，请求的域名包括：“gamer sky”、“起点中文网”、“新浪微博”、“酷 6 视频”、“QQ 空间”、“人人网”、“谷歌”……

很明显，*.*.211.66 如果是托管服务器，不可能同时对外发起这么多、而且不重复的 DNS 请求，有很多人通过 *.*.211.66 作为跳板上网。

全面分析\日志 124					
时间	客户端	客户端端口	服务器	服务器端口	查询
2011/11/25 16:56:00	211.66	59145	20.0.0.199.8	53	www.gamersky.com
2011/11/25 16:56:01	211.66	53094	20.0.0.199.8	53	pic.snyu.com
2011/11/25 16:56:03	211.66	51000	20.0.0.199.8	53	rm.api.weibo.com
2011/11/25 16:56:05	211.66	56526	20.0.0.199.8	53	urs.microsoft.com
2011/11/25 16:56:06	211.66	63342	20.0.0.199.8	53	img.ku6.com
2011/11/25 16:56:06	211.66	54898	20.0.0.199.8	53	qidian.i.adsame.com
2011/11/25 16:56:07	211.66	60616	20.0.0.199.8	53	jumpmyku6.ku6.com
2011/11/25 16:56:08	211.66	57505	20.0.0.199.8	53	dwttracking.sdo.com
2011/11/25 16:56:08	211.66	63350	20.0.0.199.8	53	script.cmfu.com
2011/11/25 16:56:08	211.66	65306	20.0.0.199.8	53	jingpin.qidian.com
2011/11/25 16:56:08	211.66	63475	20.0.0.199.8	53	junshi.qidian.com
2011/11/25 16:56:09	211.66	61898	20.0.0.199.8	53	top.qidian.com
2011/11/25 16:56:09	211.66	65046	20.0.0.199.8	53	www.qdwenzxue.com
2011/11/25 16:56:09	211.66	62879	20.0.0.199.8	53	www.qidian.cn
2011/11/25 16:56:09	211.66	55749	20.0.0.199.8	53	youxi.qidian.com
2011/11/25 16:56:10	211.66	50707	20.0.0.199.8	53	kankan.dl.meituan.com
2011/11/25 16:56:11	211.66	60464	20.0.0.199.8	53	uedas.qidian.com
2011/11/25 16:56:16	211.66	61667	20.0.0.199.8	53	open.qzone.qq.com
2011/11/25 16:56:18	211.66	58351	20.0.0.199.8	53	hzs7.cnzz.com
2011/11/25 16:56:18	211.66	50133	20.0.0.199.8	53	api.renren.com
2011/11/25 16:56:25	211.66	61538	20.0.0.196.68	53	seupdate.360safe.com
2011/11/25 16:56:27	211.66	55300	20.0.0.199.8	53	hzvs1.cnzz.com
2011/11/25 16:56:28	211.66	64179	20.0.0.199.8	53	seupdate.360safe.com
2011/11/25 16:56:28	211.66	63799	20.0.0.199.8	53	3gqq.qq.com
2011/11/25 16:56:28	211.66	64972	20.0.0.199.8	53	qz.qq.com

HTTP 日志分析：*.*.211.66 在 12 月 25 号 56 分这一分钟，总共有 789 条上网记录，这同样不可能是托管服务器的行为：

5. 下载、视频行为分析

从 DNS 记录中，可以发现大量的视频、下载门户网站查询：

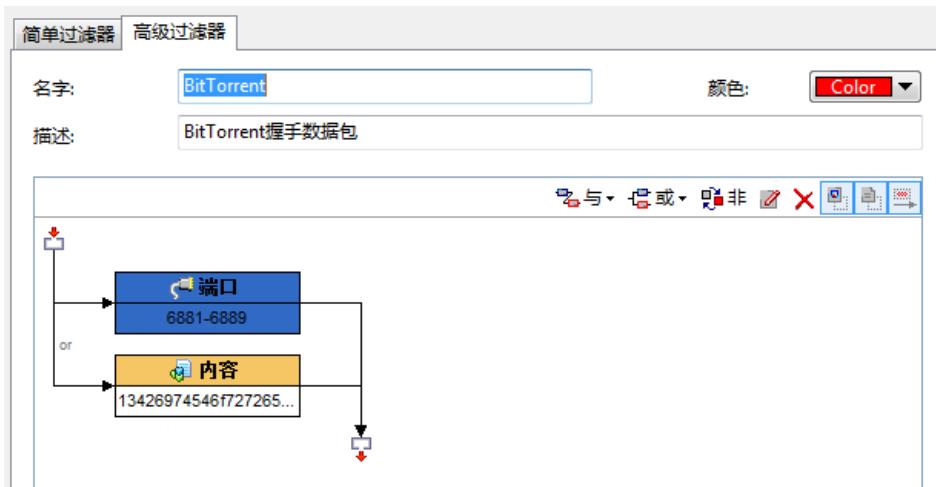
客户端	DNS 服务器	查询内容	对应网站
..211.66	*.*.199.8	xmp.kankan.xunlei.com	迅雷看看
..211.66	*.*.199.8	live.v2.pplive.com	PPLIVE
..211.66	*.*.199.8	ku6.com	酷六视频
..211.66	*.*.199.8	so.tudou.com	土豆网
..211.66	*.*.199.8	www.tom365.com	TOM365 免费电影
..211.66	*.*.199.8	www.haidaowan.net	海盗湾电影下载
..211.66	*.*.199.8	local_p2p.qq.com	QQ 旋风下载

通过 HTTP 日志记录，同样可以发现大量的 P2P 链接请

客户端	服务 器	查询内容	下载内容
..211.66	61.16 4.118.217	http://61.164.118.217/FTP/fen gyun/20110628/1/ZSZZ_Setup1.9. 0.105.exe	游戏“诸神之战”客 户端, 1GB
..211.66	down. qq.com	http://down.qq.com/dnf/Patch/ DNF_SEASON2_V5.236_Pack.ex e	游戏“银魂”客户端, 600MB
..211.66	down. qq.com	QQR2_v3.2.1_FULL.exe	QQ 音速游戏, 3G
..211.66	yxdow n.com	http://52.yxdown.com/COD8.r ar	游戏, 14GB
..211.66	*.*.19 9.8	http://down.qq.com/cf/full/Cro ssFire_OBV134_Full.exe	游戏, 700MB

此外，我们还可以通过数据包特征码，找出存在下载行为的主机。

比如，我们定义了 BT 握手的数据包特征的过滤器



执行过滤功能之后，便能找到符合 BT 下载行为的通信主机：

节点1->	<-节点2	数据包	字节 ->	协议	持续时间	字节->	<-字节	数据包 ->
192.168.11.66:62495	121.149.146.129:6881	6,779	4,443 MB	BitTorrent	02:34:27	4,443 MB	0 B	6,779
192.168.11.66:2994	121.149.144.65:6881	11,957	3,774 MB	BitTorrent	02:21:01	2,306 MB	1,468 MB	6,381
192.168.11.66:49905	121.149.146.209:6881	1,542	1,867 MB	BitTorrent	01:18:36	1,867 MB	0 B	1,542
192.168.11.66:66881	121.149.141.211:65348	4,762	1,387 MB	BitTorrent	02:26:32	1,387 MB	0 B	4,762
192.168.11.66:1407	121.149.146.242:6881	1,452	1,150 MB	BitTorrent	23:36	1,150 MB	0 B	1,452
192.168.11.66:62267	121.149.146.56:6881	727	913.834 KB	BitTorrent	14:30	913.834 KB	0 B	727
192.168.11.66:1456	121.149.146.487:6881	1,181	872.741 KB	BitTorrent	11:15	39.409 KB	833.332 KB	51
192.168.11.66:63855	121.149.146.404:6881	983	833.914 KB	BitTorrent	22:50	806.775 KB	27.139 KB	65
192.168.11.66:72681	121.149.146.110:62264	1,506	743.637 KB	BitTorrent	01:23:11	389.768 KB	353.869 KB	681
192.168.11.66:55273	121.149.146.558:6881	602	742.066 KB	BitTorrent	15:47	742.066 KB	0 B	602
192.168.11.66:59208	121.149.146.539:6881	589	741.114 KB	BitTorrent	19:32	741.114 KB	0 B	589
192.168.11.66:51596	121.149.146.162:6881	1,011	731.994 KB	BitTorrent	44:51	731.994 KB	0 B	1,011
192.168.11.66:21176681	121.149.146.211:61822	2,003	697.165 KB	BitTorrent	03:23:59	697.165 KB	0 B	2,003
192.168.11.66:61090	121.149.146.165:536881	556	684.329 KB	BitTorrent	08:51	684.329 KB	0 B	556
192.168.11.66:26681	121.149.146.211:6657061	790	664.262 KB	BitTorrent	13:32	21.081 KB	643.181 KB	27
192.168.11.66:54357	121.149.146.188:6881	761	614.217 KB	BitTorrent	08:51	614.217 KB	0 B	761
192.168.11.66:50783	121.149.146.100:4376881	443	599.809 KB	BitTorrent	10:47	599.809 KB	0 B	443

2.4. 托管服务器异常行为总结

通过对异常主机*.211.66 的行为分析后，我们发现该主机存在大量不符合托管服务器特征的行为，包括：

对外发起了大量请求，下载流量远大于上传流量

每个时刻都有大量的 DNS、HTTP 请求，这是多人同时上网才能引起的现象

访问了许多视频网站

存在很多 BT 下载行为

以上行为特征不应该是一台托管服务器所具有的，管理人员需要确认一下该 IP 地址的用途，如果不是对外网关的地址，而是托管服务器的地址，那么，该托管服务器已经被设置成对外上网的代理。

其他存在异常行为服务器汇总：

通过“流量发收比”、“发 TCP 同步包”等参数能够快速的找出不符合托管服务器特征的 IP 主机，那么，我们通过这两个参数的排序，便能找出比较异常的托管服务器：

2.5. “发TCP同步包”异常主机汇总：

地址	字节数	数据包	每秒流量	发TCP同步包	收TCP同步包	流量发收比
121.11.251.24	2.12 GB	3,427,215	175.18 KB	69,256	34,077	0.17
121.11.211.66	9.45 GB	19,068,064	782.87 KB	62,618	18,793	0.40
121.11.251.10	2.51 GB	4,268,717	208.00 KB	52,309	98,811	0.22
121.11.251.11	1.72 GB	3,093,280	142.69 KB	49,458	178,122	0.28
121.11.251.18	1.68 GB	2,729,782	139.27 KB	41,407	91,686	0.24
121.11.251.27	1.76 GB	2,835,536	145.51 KB	38,994	38,405	0.29
121.11.251.14	1.37 GB	2,333,430	113.86 KB	38,915	71,327	0.18
121.11.251.9	2.03 GB	3,620,818	168.23 KB	37,855	24,574	0.31
121.11.251.6	1.87 GB	3,133,227	155.16 KB	37,495	120,549	0.22
121.11.251.29	2.18 GB	3,637,021	180.69 KB	37,321	98,010	0.28
121.11.251.16	1.26 GB	2,188,667	104.51 KB	35,472	98,343	0.24
121.11.251.28	1.56 GB	2,725,648	128.91 KB	33,995	63,868	0.28
121.11.251.3	1.46 GB	2,565,507	120.61 KB	30,974	145,441	0.29
121.11.251.2	1.93 GB	3,533,901	159.74 KB	30,874	37,943	0.30
121.11.251.22	1.14 GB	1,817,321	94.56 KB	30,805	49,257	0.33
121.11.251.12	1.44 GB	2,306,100	119.08 KB	30,261	14,079	0.46
121.11.251.9.30	117.84 MB	327,479	9.53 KB	28,671	4,882	5.33
121.11.251.19	1.55 GB	2,747,793	128.44 KB	28,290	51,662	0.38
121.11.251.5	1.28 GB	2,206,958	105.74 KB	28,241	86,254	0.23
121.11.251.15	2.10 GB	3,938,431	173.87 KB	27,172	87,525	0.34
121.11.251.26	1.50 GB	2,625,801	124.50 KB	25,010	65,155	0.35
121.11.211.2	1.93 GB	4,435,165	159.59 KB	24,701	32,425	0.42
121.11.211.6	2.30 GB	5,414,048	190.44 KB	23,657	2,761	0.44
121.11.251.25	1.35 GB	2,379,118	111.71 KB	23,356	5,326	0.37

以上 IP 主机都是对外发送了大量的 TCP 同步包，而且流量发收比都很小，不符合托管服务器的特征。

综上所述，科来网络回溯分析服务器记录了各种关键网路参数指标，通过这些关键参数排序，可以快速的找到异常主机。

3. 回溯式发现、挖掘、追踪木马通信

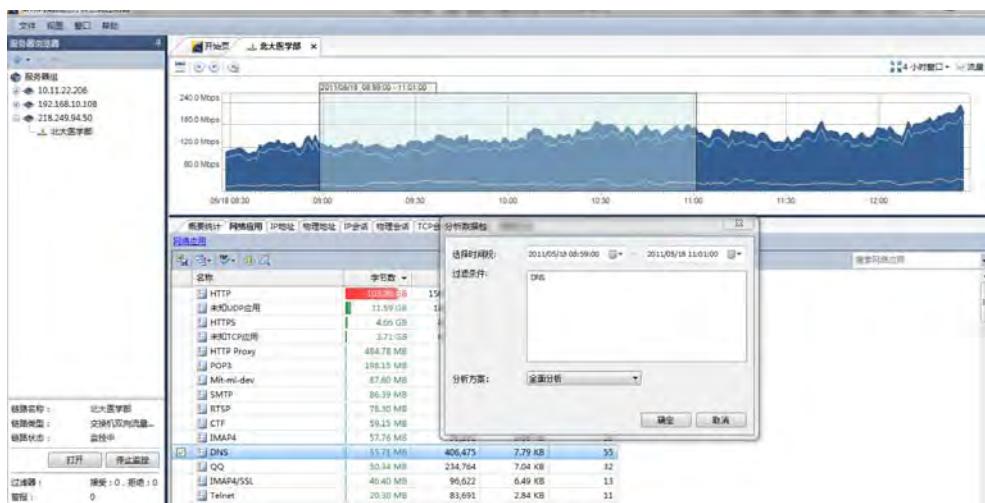
3.1. 案例背景

客户：某知名大学医学部

部署位置：互联网出口

知识前瞻：当今木马多采用反弹式建立连接。中木马的主机由内向外发起连接，绕开防火墙和安全设备的检测，更加隐蔽。而木马在建立连接之初 总要通过动态域名，blog 等形式进行查询，以找到控制端的 IP 与服务端口。因此根据其特点进行 DNS 的日志分析是找出反弹木马重要手段。

木马发现：该大学医学部于 5-17 日部署了科来回溯式硬件，以便了解自己网络流量构成，异常流量，以及木马网络攻击等行为。5-18 日，使用控制台将其 上午 9:00—11:00 的 DNS 日志下载下来进行分析。



选择下载并分析数据包后，对该两个小时的 DNS 日志进行搜索，在结果统计中关键字 3322.(3322.org 是黑客常用动态域名，著名的希网解析服务，此类域名开头部分随意，结尾多为 *.3322.org;*.7766.org;*.2288.org 等)。

如图：

DNS 日志							
	客户端	客户端端口	服务器	服务器端口	查询	状态	归属
全局日志	8.09:35:40	218.249.94.16	37242	202.106.0.20	53	img3.pplive.cn	成功 CNAME=webcdn.pptv.com.cacheon
DNS日志	8.09:35:40	218.249.94.16	21847	202.106.0.20	53	img1.pplive.cn	成功 CNAME=webcdn.pptv.com.cacheon
Email信息	8.09:35:40	218.249.94.16	32694	202.106.0.20	53	img2.pplive.cn	成功 CNAME=webcdn.pptv.com.cacheon
FTP 传输	8.09:35:40	218.249.94.16	367	在 DNS 日志中查找	23	in	成功 CNAME=webcdn.pptv.com.cacheon
HTTP 请求日志	8.09:35:40	218.249.94.16	112	查找内容: 3322	查找下一个	in	成功 CNAME=webcdn.pptv.com.cacheon
MSN 日志	8.09:35:40	218.249.94.16	523	直接忽略	结果	关闭	失败 CNAME=webcdn.pptv.com.cacheon
Yahoo 日志	8.09:35:40	218.249.94.30	287	是否分大小写	选择全部	统计全部	失败 搜索响应: 服务端繁忙!!
	8.09:35:40	218.249.94.30	287	失败	失败	失败	失败 搜索响应: 服务端繁忙!!
	8.09:35:40	218.249.94.30	463	失败	失败	失败	失败 搜索响应: 服务端繁忙!!
	8.09:35:40	218.249.94.21	9621	www.adm66.cn	53	www.adm66.cn	成功 CNAME=quip.f360.cn; CNAME=quip
	8.09:35:40	218.249.94.21	22812	231.161.46.85	53	conf.f360.cn	成功 CNAME=quip.f360.cn; CNAME=quip
	8.09:35:40	218.249.94.30	28737	210.238.149.5	53	in	失败 搜索响应: 服务端繁忙!!
	8.09:35:40	218.249.94.30	28737	210.238.149.5	53	in	失败 搜索响应: 服务端繁忙!!
	8.09:35:40	218.249.94.20	17710	119.84.84.11	53	xjmir.7766.org	成功 A=218.6.11.7.17.1; Renz2.3322.net; B
	8.09:35:40	218.249.94.30	28737	210.238.149.5	53	in	失败 搜索响应: 服务端繁忙!!
	8.09:35:40	218.249.94.30	28737	210.238.149.5	53	in	失败 搜索响应: 服务端繁忙!!
	8.09:35:40	218.249.94.34	44994	202.106.0.20	53	androidportal.hu	成功 A=195.56.55.57
	8.09:35:41	218.249.94.16	23224	202.106.0.20	53	img9.pplive.cn	成功 CNAME=webcdn.pptv.com.cacheon
	8.09:35:41	218.249.94.16	11326	202.106.0.20	53	img5.pplive.cn	成功 CNAME=webcdn.pptv.com.cacheon
	8.09:35:41	218.249.94.16	27603	202.106.0.20	53	img6.pplive.cn	成功 CNAME=webcdn.pptv.com.cacheon
	8.09:35:41	218.249.94.34	44994	202.106.0.20	53	www.newsmit.net	成功 CNAME=www.k.newsmit.net; CNAME

通过查找我们发现两个域名：mayi.7766.org 和域名 xjmir.7766.org. 而且该域名都是由 IP 218.249.94.20 进行解析。两个域名在 1 小时内被解析过 40 次，解析较为频繁。

试着通过 google 和百度来搜索该域名的信息发现这两个域名为木马控制端的域名。

2、本周恶意域名处理情况

依据《中国互联网域名管理办法》和《木马和僵尸网络监测与处置机制》等相关法律法规的规定，本周 ANVA 在万网、希网等域名注册服务机构的配合和支持下，对 139 个在中国大陆注册的、传播网络病毒的恶意域名采取了暂停解析的处置措施。详细列表如下所示。

处置域名列表	处置原因
schastlivieiveselierebyta0009.com、stvpn.com、sd3721.com.cn、sdguancheng.cn、weifeng.com.cn、xiongrulin.cn、xiuwap.cn、yj123.cn、mapeak.com、tjqln.com、xyz68.com、lqzf.net、www.adm66.cn、360vs2B.3322.org、6070.9966.org、aaahsgz11.8866.org、adsjcxzb22.8866.org、afpm.3322.org、dddjah1.8800.org、dfsferaw112.3322.org、guidady.9966.org、guisb.9966.org、guitou.9966.org、jczhhass33.7766.org、jxzcnhas11.8866.org、nimadc190.3322.org、nimakfc272.3322.org、sdjfggag11.7766.org、vcmfc003.3322.org、vcmfc004.3322.org、vsdfscv890.8866.org、woail68.3322.org、xjmir.7766.org、1235633.3322.org、123654654.3322.org、1452werwer.3322.org、lqwas42.3322.org、lqwas43.3322.org、8899liao.3322.org、9090z.3322.org、akddos.3322.org、as8877.3322.org、benbenwan.3322.org、bjbj888.3322.org、btfans.3322.org、bytuzi.3322.org、cean.3322.org、ddosfuwu.3322.org、dksk.3322.org、dongqilail23.3322.org、dr112212.3322.org、feng7020.3322.org、fengbaogege.3322.org、fjlh.3322.org、fsjidian.3322.org、fuqy.3322.org、gdmore.3322.org、gm5a.3322.org、otczvlvl3322.org、hevanos520.3322.org、hk12000.3322.org、	传播恶意代码

显然，内网 PC 218.249.94.20 确实是中了木马了。

木马挖掘：中了木马后的主机，向控制端传送过什么数据？有哪些通信行为？网内是否还有人中了该木马？这就要对中了马的主机进行细致分析了。

3.2. 分析总结

首先我们确定木马控制端是什么 IP，其实在 DNS 日志中，已经有答案了。我们看到 xjmir.7766.org 该域名的公网 IP 是:218.61.17.171。mayi.7766.org 的 IP 是 61.147.116.75。我们选择 5-18 日一天的流量（时间窗口选择 24 小时），在 IP 地址里去搜索这两个 IP（注意：木马通信一般流量不大，而控制台每次默认选择流量最大的 1000 台 PC 进行展现，所以我们要选择“显示全部 IP”）

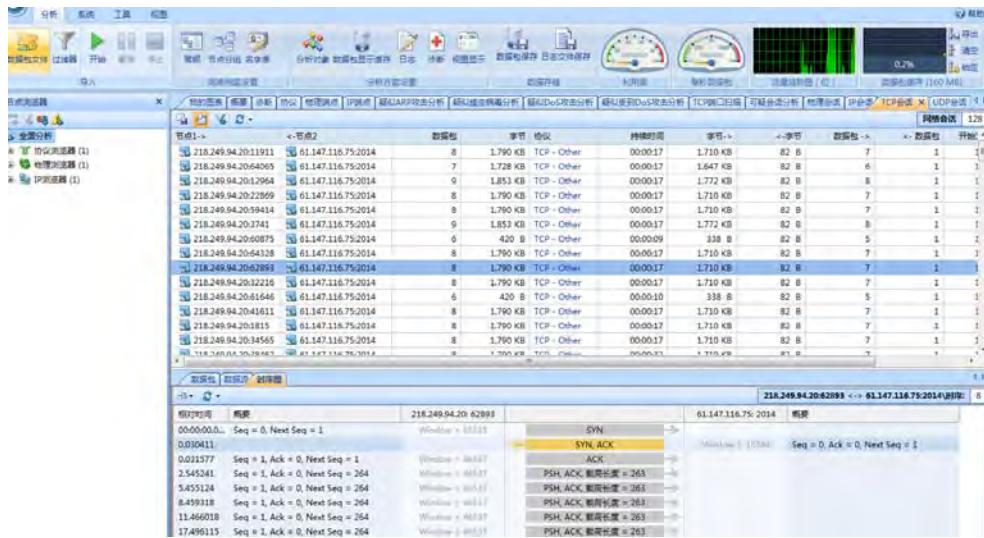
地址	字节数	数据包	每秒流
100	101.29 GB	142,864,072	1.43 M
500	64.44 GB	90,896,904	933.77 M
1000	62.88 GB	90,676,401	911.21 M
2000	61.35 GB	98,183,202	889.02 M
5000	56.54 GB	84,657,928	819.38 M
10000	55.93 GB	76,993,828	810.54 M
20000	54.88 GB	91,387,580	795.24 M
	显示全部		
218.249.94.42	54.34 GB	79,169,186	787.47 M
	52.40 GB	73,913,091	759.30 M

通过搜索 IP 找出了木马控制端 IP，并发现其与 20 主机有通信。

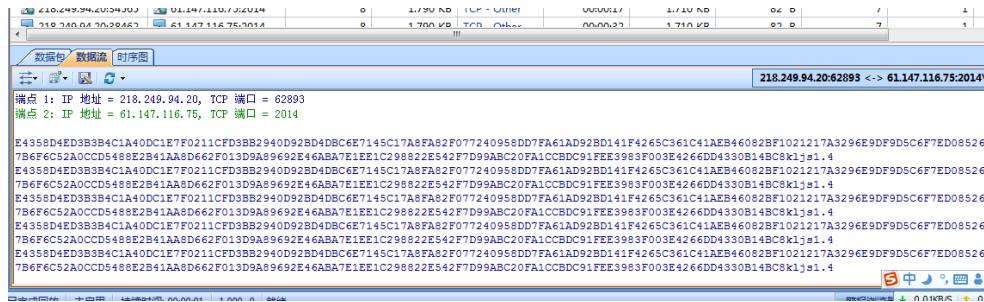
概要统计	网络应用	IP地址	物理地址	IP会话	物理会话	TCP会话	UDP会话	警报日志
地址								
								61.147.116.75
地址	字节数	数据包	每秒流量	发包	收包	流量发收比	数据包发收比	发TCP同步包
61.147.116.75	195.44 KB	1,137	20 B	118	1,019	0.05	0.11	0

公网 IP 218.61.17.171 未与肉鸡进行通信，估计是控制端不在线，或木马处于潜伏期。

下载木马通信的数据流。使用控制台集成的数据包级分析工具进行分析。



如图我们看到木马通信使用的是 TCP 2014 端口，但该木马通信内容部分是加密过的，数据流看到内容有限



而且木马通信的质量也很不好，出现大量的重传数据包。查看控制端的 IP:

ip138.com IP查询(搜索IP地址的地理位置)

您查询的IP:61.147.116.75

本站主数据: 江苏省扬州市 电信
参考数据一: 江苏省扬州市 电信

4. 邮件系统攻击分析

4.1. 案例背景

邮件系统是企业单位最经常使用的网络应用之一。邮件系统中一般有客户的关键信息，一旦邮件系统瘫痪或被黑客掌控，那么就会给企业带来重大损失。本文两个案例都是针对邮件服务器的攻击案例，希望通过这些实际网络案例给大家有所帮助。

客户环境说明

客户为某大型保险公司，邮件系统是该单位使用最为频繁的系统之一。该单位邮件系统分为两种：WEB 登录方式，和使用标准的 SMTP POP3 协议收发方式。科来回溯式分析服务器部署在数据中心的核心交换机上，通过 span 将 DMZ 区的所有服务器流量引入回溯服务器进行分析。

4.2. 针对邮件系统的暴力破解

9-20 日在进行分析时发现分公司的一些 IP 在进行针对邮件服务器的暴力破解攻击。我们选择 2 天时间窗口，然后选择 9/19 的上午的数据进行分析。点击“发 tcp 同步包”选项进行排名，我们发现 IP 10.94.200.66 的流量只有 9.35MB 但“tcp 发送同步包”却排名第 3 位，达到了 20592 个。这种 TCP 会话很多，流量又特别小的 IP 通常是比较异常的。我们选择下载分析该 IP 数据包，进行深入分析。



下载该 IP 的通信数据后我们发现，该 IP 在 9/19 日上午对邮件服务器发起近超过 2 万次 TCP 请求，而且密集时候每秒能发送 100 多个 TCP 同步包。



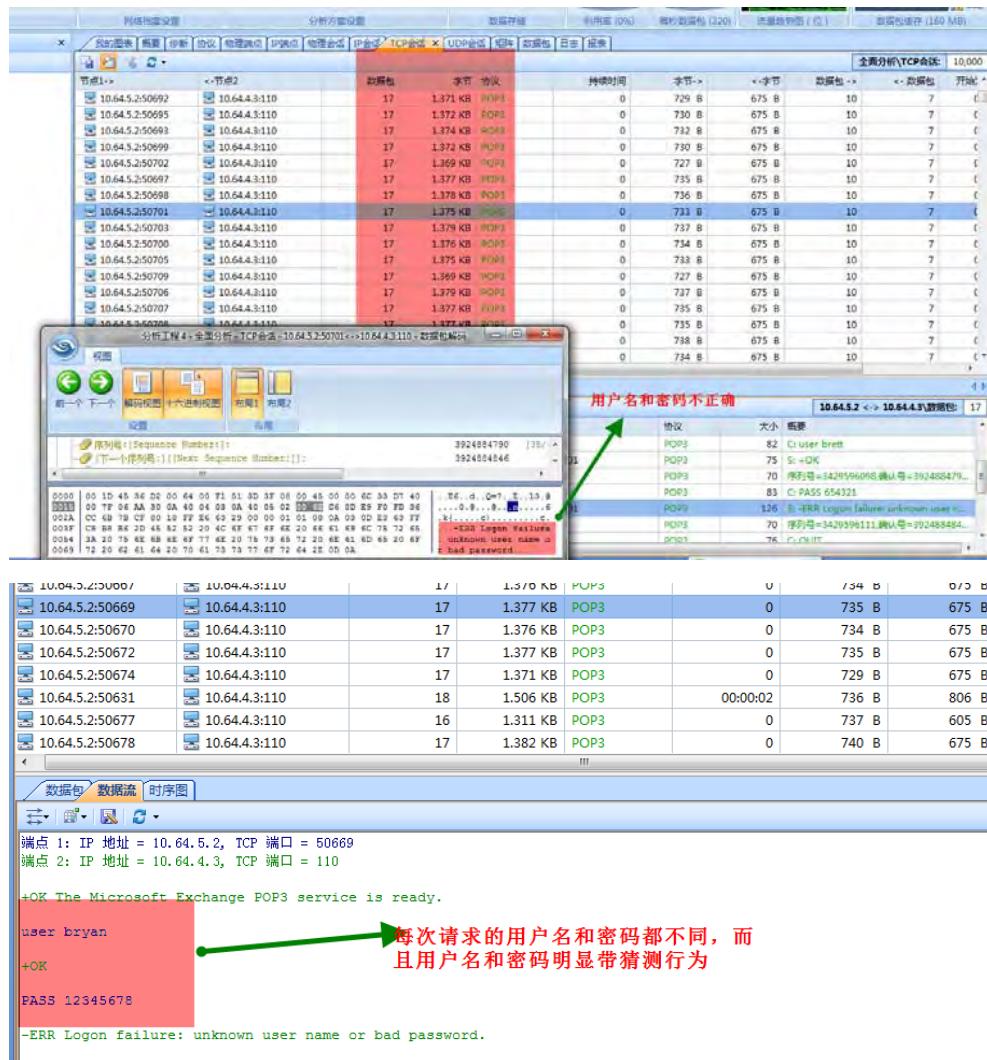
如上图，我们看到 IP10.94.200.66 在很短时间内向 mail 服务器 10.64.4.3 做了多次重复的会话，从行为上来看，10.94.200.66 在向 mail 服务器进行请求，但又始终不发送三次握手中最后的 ACK 数据包，这样导致它与服务器的 TCP 会话始终无法建立，而且服务器为了等待 200.66 回送 ACK 会消耗一定的系统资源，这样高的频率的不正常的请求访问，就造成了对 mail 服务器的 DOS 攻击。

而 200.66 在与服务器建立的成功的会话中也是较大异常的，通过“**HTTP 日志**”分析我们可以看到以下不正常现象，如图：



我们看到 200.66 的每次访问的 URL 是一模一样的，而且出现每秒钟多达 10 次以上的访问，从该频率看不是人为访问，应该是病毒程序自动访问导致。分析这个 URL 发现打开后是 mail 服务器的 WEB 登录界面。因此我们可以认为这种行为应该是在进行密码尝试。

同样的本次分析发现服务器段的 IP 10.64.5.2 也在想 mail 服务器进行密码尝试行为，如图：



通过以上针对 mail 服务器的分析我们发现，网络中存在很多针对 mail 服务器的不正常会话，这些会话对 mail 服务器形成了攻击。攻击以 DOS 和用户名密码的猜测较多，属于渗透攻击。

这些攻击猜测行为一旦被取得真实的用户名和密码后，就能够对 mail 服务器做数据偷窃，那么每封 mail 的信息将会没有秘密可言。（例如，黑客攻击得到了 mail 服务器的用户名和密码后可以潜伏到网络中侦听他想要的信息，造成信息窃密的发生，对公司业务造成损失）

建议加强 mail 服务器的防护，并对攻击者强制杀毒，并在防火墙上做一些 TCP 会话的强制会话时间限制（例如：在防火墙上做策略，使 mail 每次 TCP 会话空闲时间不超过 2 秒，如果 2 秒得不到 ACK 回应则重置会话）

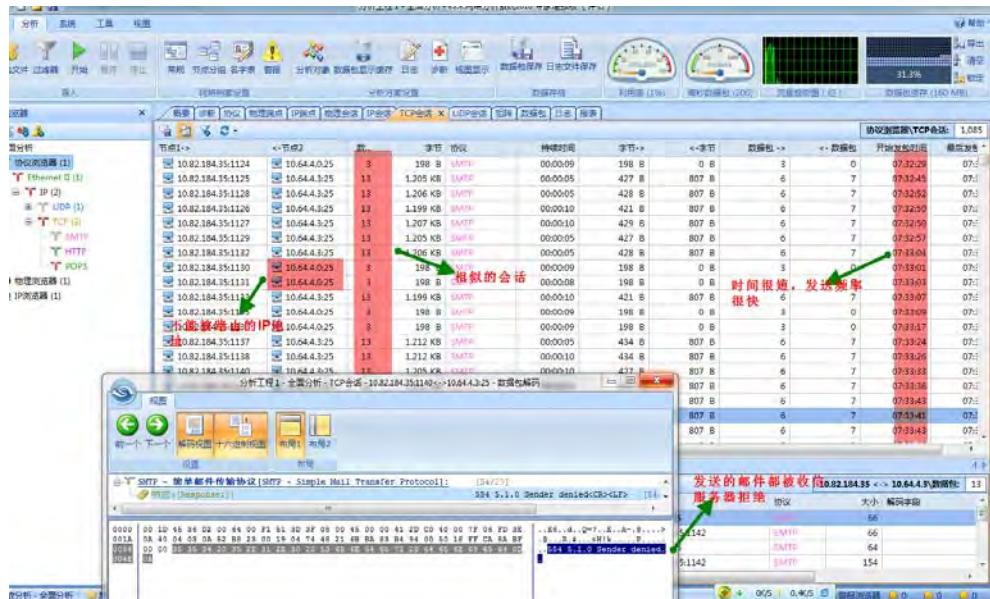
4.3. 邮件蠕虫攻击

通过以上分析我们发现网络中的邮件服务器的状况不太安全。那么还有没有其他问题呢？

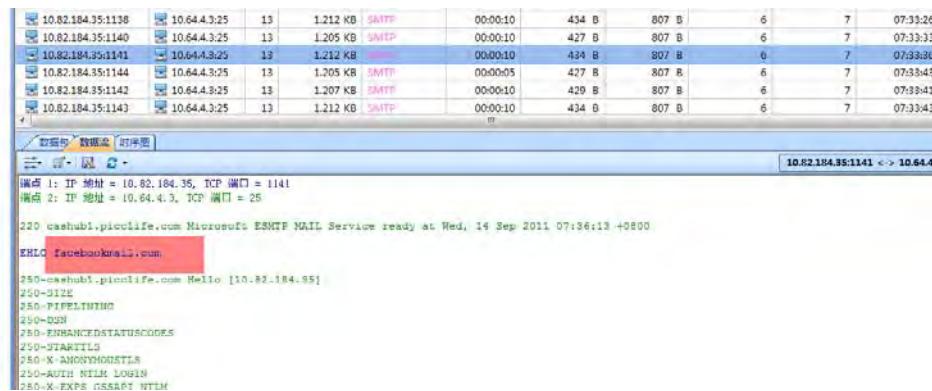
由于邮件服务器的数据量很大，每天有超过 10GB 的流量，因此我们决定使用采样分析的方法对邮件服务器进行数据采样分析。我们选择上午 9-10 点之间数据（该单位 9 点上班，邮件系统比较繁忙）。然后

选择网络应用中的 SMTP 进行挖掘分析，在查看会话时我们发现 IP10.82.184.35 的会话数很多，在近 1 小时内该 IP 的 SMTP 会话到达几百个。明显的异常现象。由于我们选择将该 IP 的一上午时间的数据包全部下载分析。

首先我们打开“tcp 会话”看到最多的是 10.82.184.35 和 mail 服务器 10.64.4.3 之间的 13 个数据包的会话。



而且该 IP 还向 10.64.4.0 发起请求，但显然这种 IP 是不会存在的，所以只有三次 SYN 包，但没有任何回应。该 IP 在 1 分钟内就能发送近 10 封内容相差不多的邮件，而且这种邮件收信者多是比较大的门户网站。如图：



统计发现，该主机在一上午时间内发送了超过 2000 封类似的邮件。而这么高频率的发送显然不是人工所为。这种现象应该是该主机中了僵尸程序，然后僵尸程序自动向其他网站发送大量的垃圾邮件所致。

建议：对该主机进行杀毒后再接入网络。

5. 飞客蠕虫研究

5.1. 案例背景

飞客蠕虫是近两年感染率最高的蠕虫，没有之一。在本人实际工作中发现过多起感染飞客蠕虫的事件，现在将对该蠕虫进行比较细致介绍和分析一下。

飞客蠕虫是英文 conficker 的中文发音。该蠕虫最早发现于 2008 年 12 月，利用微软的利用微软 MS08-067 漏洞发起攻击。自从发现该蠕虫病毒后，在 1 年的时间内出现了 5 个主要变种，功能和隐蔽性比原始程序得到很大提升。其主要版本演进如下：



飞客蠕虫感染最多的就是中国大陆地区，网上看到一则新闻，在 2010 年 10 月份的平均每个月内就有 1800 万 PC 感染了飞客蠕虫。最高时，根据国家计算机应急指挥中心公布的数据，全国有近 10% 的 PC 感染了该蠕虫。由此我们也可以看出国内的 PC 安全意识很淡薄，其实简单的更新下系统，打上相关补丁，就不会感染该蠕虫。

“飞客蠕虫”依旧活跃每月感染1800万用户

2010-10-22 17:42:33 来源：新华网(广州) 跟贴 0 条 手机看新闻

新华网北京 10 月 22 日电 据瑞星“云安全”系统统计，本周共截获 29 万个挂马网址。瑞星安全专家介绍，利用微软 MS08-067 漏洞发起攻击的恶性病毒“飞客蠕虫”依然在互联网中活跃，平均每月感染 1800 万用户。

本周关注的被挂马网站：“杭州摄影网”“EIC 启德教育”“中华网”等网站的部分页面曾被黑客挂马，黑客利用微软 IE 最新漏洞和服务器不安全设置进行入侵。用户访问这些页面后，可能会感染恶性“后门”。

研究任何病毒木马最好的方式就是主动感染该病毒木马，然后跟踪其行为。本文也是以这种思路来进行对飞客蠕虫的研究。研究之前的工作如下：

准备一台没有打 MS08-067 补丁的 XP 系统的主机，在此我使用较早版本的 XP SP2 版本，然后安装到一台虚拟机上。

从互联网上寻找飞客蠕虫的各版本，主要是 C,D,E 这三个版本为主，因为这些后来版本功能较强大，通信行为也很复杂。这项准备工作最为困难，因为找到合适的样本是比较不容易的，在此我推荐一个网站：<http://www.offensivecomputing.net> 该网站是国外比较专业的木马病毒样本网站，不过需要较严格的用户注册条件。

在虚拟机上安装科来 2010 旗舰版，准备做好抓包工作。

准备完毕后，将下载的样本在本机上运行。然后进行抓包，设置好过滤器保证抓到的数据是比较纯净的，不要使用该 XP 系统做任何网络操作，然后进行长时间的抓包。在连续抓包几个小时后停止抓包，进行数据分析。

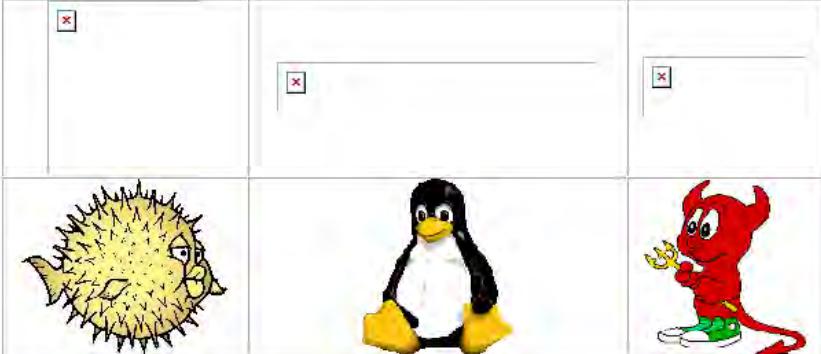
首先，感染飞客蠕虫后的主机会禁止系统访问各种杀毒厂商的网站和相关安全信息内容。所以根据这个属性我们可以检查一下我们是否中了飞客蠕虫。点击链接：<http://www.joestewart.org/cfeyechar.html> 我们看到如下的图：

址 ② http://bigdouya.blogbus.com/logs/37519623.html

Conficker Eye Chart 原理：Conficker的一大特性是主动屏蔽杀毒软件的网站，这个测试就是针对这一特点设计的。所以当只是第一排中的几大杀软的主页Logo不能读取而下面的一排正常时，就说明你可能是中招了。

Conficker Eye Chart 使用方法：

1、用浏览器完成对下图Logo的读取后，观察下图显示情况

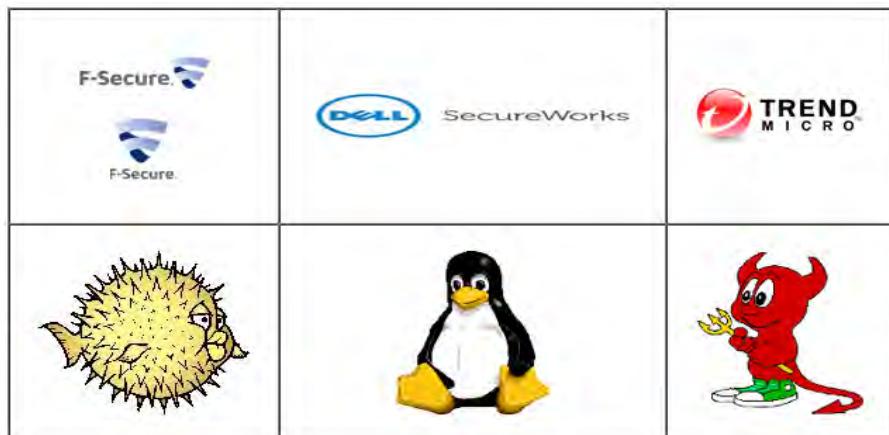


2、根据上图Logo显示的情况，判断是否中了Conficker

这的确是个简单检测Conficker的方法，怀疑感染的试试吧~~~但此方法只是帮助判断是否感染Conficker，实际上还是用Conficker专杀工具更彻底。

原文链接：<http://www.joestewart.org/?p=26>

如图，这是本机中了飞客蠕虫的表现，排名上面的一行是各安全厂商的图片链接。



对抓包的数据进行分析，可以了解其行为：

首先，蠕虫成功运行后会向指定的域名发送请求，该蠕虫每天尝试从 50000 个域名中随机挑选 500 个域名以试图与恶意软件制造者通信，因此会产生大量的奇怪的域名解析如图：

全面分析日志 1,084						
	客户端	客户端端口	服务器	服务器端口	查询	状态
金明 日志	2011/10/10 14:02:28	192.168.10.90	1059	202.106.0.20	53	dnndwdnqj.li 成功 A=202.106.199.37
DNS 日志	2011/10/10 14:05:59	192.168.10.90	1059	202.106.0.20	53	qwszzzbqyukz 成功 A=202.106.195.30
Email 日志	2011/10/10 14:03:38	192.168.10.90	1059	202.106.0.20	53	oddm.com.do 成功 A=202.106.199.30
FTP 日志	2011/10/10 14:04:21	192.168.10.90	1059	202.106.0.20	53	bgwvqenqk.lv 成功 A=202.106.199.39
HTTP 请求日志	2011/10/10 14:04:41	192.168.10.90	1059	202.106.0.20	53	aerjovnae 成功 A=202.106.199.37
HTTP 请求日志	2011/10/10 14:05:24	192.168.10.90	1059	202.106.0.20	53	kgnr.no 成功 A=202.106.199.39
HTTP 请求日志	2011/10/10 14:06:01	192.168.10.90	1059	202.106.0.20	53	ptvppn.hk 成功 A=202.106.199.36
HTTP 请求日志	2011/10/10 14:06:21	192.168.10.90	1059	202.106.0.20	53	efqtmslc.com.do 成功 A=202.106.199.39
HTTP 请求日志	2011/10/10 14:06:53	192.168.10.90	1059	202.106.0.20	53	qehs.co.nz 成功 A=202.106.199.30
HTTP 请求日志	2011/10/10 14:06:53	192.168.10.90	1059	202.106.0.20	53	turvniz.yq 成功 A=202.106.199.37
Yahoo 日志	2011/10/10 14:07:26	192.168.10.90	1059	202.106.0.20	53	plfyruzcz.com.do 成功 A=202.106.199.39
Yahoo 日志	2011/10/10 14:07:43	192.168.10.90	1059	202.106.0.20	53	gryterdt.com.ar 成功 A=202.106.195.30
Yahoo 日志	2011/10/10 14:08:13	192.168.10.90	1059	202.106.0.20	53	wkmnqdy.ie 成功 A=202.106.199.39
Yahoo 日志	2011/10/10 14:08:50	192.168.10.90	1059	202.106.0.20	53	jabebruhn.de 成功 A=202.106.199.36
Yahoo 日志	2011/10/10 14:09:30	192.168.10.90	1059	202.106.0.20	53	qpgqrlie.qs 成功 A=202.106.199.36
Yahoo 日志	2011/10/10 14:09:53	192.168.10.90	1059	202.106.0.20	53	nbowtive.sc 成功 A=202.106.199.39
Yahoo 日志	2011/10/10 14:10:41	192.168.10.90	1059	202.106.0.20	53	pvsae.com.ua 成功 A=202.106.199.36
Yahoo 日志	2011/10/10 14:11:09	192.168.10.90	1059	202.106.0.20	53	gfrogxqq.as 成功 A=202.106.195.30
Yahoo 日志	2011/10/10 14:11:57	192.168.10.90	1059	202.106.0.20	53	lypon.tl 成功 A=202.106.199.36
Yahoo 日志	2011/10/10 14:12:27	192.168.10.90	1059	202.106.0.20	53	ulfj.mn 成功 A=202.106.199.36
Yahoo 日志	2011/10/10 14:12:54	192.168.10.90	1059	202.106.0.20	53	tdice.com.tt 成功 A=202.106.195.30
Yahoo 日志	2011/10/10 14:13:55	192.168.10.90	1059	202.106.0.20	53	evuti.gr 成功 A=202.106.199.36
Yahoo 日志	2011/10/10 14:14:07	192.168.10.90	1059	202.106.0.20	53	elhusktw.sc 成功 A=202.106.199.39
Yahoo 日志	2011/10/10 14:14:50	192.168.10.90	1059	202.106.0.20	53	uabwqoro.co.kr 成功 A=202.106.195.30
Yahoo 日志	2011/10/10 14:15:25	192.168.10.90	1059	202.106.0.20	53	ukchbzwidipl.com 成功 A=202.106.195.30
Yahoo 日志	2011/10/10 14:16:16	192.168.10.90	1059	202.106.0.20	53	uuuqgysew.cn 成功 A=221.8.69.25
Yahoo 日志	2011/10/10 14:17:54	192.168.10.90	1059	202.106.0.20	53	vwwr.mn 成功 A=202.106.195.30
Yahoo 日志	2011/10/10 14:18:29	192.168.10.90	1059	202.106.0.20	53	aiea.es 成功 A=202.106.199.39

这些域名 google 和百度的搜索结果都是零，也就是说，是极冷僻的域名。

Google search results for the query "plfyruzcz.com.do":

找不到和您的查询 "plfyruzcz.com.do" 相符的内容或信息。

Suggestions:

- 请检查输入字词有无错误。
- 请尝试其他的查询词
- 请改用较常见的字词。

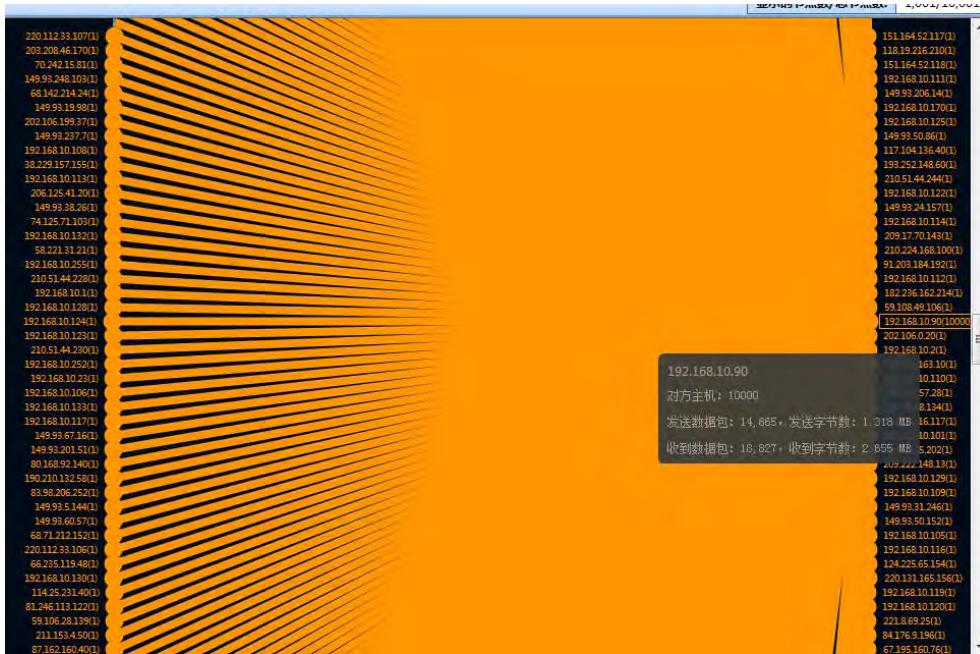
Links:

- 图片
- 地图
- 视频
- 新闻
- 购物
- 更多

成功解析了这些域名后，肉鸡开始像这些解析后的地址发起 HTTP 请求，下载最新的蠕虫更新程序，以避免被杀毒软件清除。然后还会下载各种病毒，如键盘记录软件，远程控制软件，密码收集软件等，这样 PC 就会被各种病毒木马所占领。

日志	日期时间	客户端地址	请求URL	IP过滤器/日本	方法
全局日志	2011/10/10 13:54:00	192.168.10.90	http://static.wowza.com/side_images/2011/08/15/34147498.gif		GET
	2011/10/10 13:54:00	192.168.10.90	http://static.wowza.com/site_images/2011/06/21/3577215.gif		GET
DNS日志	2011/10/10 13:53:51	192.168.10.90	http://public.blogbus.com/imgs/picobox/picobox_beauty/popup_heading.gif		GET
	2011/10/10 13:53:59	192.168.10.90	http://public.blogbus.com/imgs/picobox/picobox_beauty/rounded_lb.png		GET
Email日志	2011/10/10 14:13:32	192.168.10.90	http://149.93.721.92/		GET
	2011/10/10 15:09:16	192.168.10.90	http://149.93.60.57/		GET
FTP 传输	2011/10/10 15:44:55	192.168.10.90	http://149.93.206.14/		GET
	2011/10/10 15:46:34	192.168.10.90	http://yakura.ne.jp/		GET
HTTP请求日志	2011/10/10 15:51:26	192.168.10.90	http://linkbucks.com/		GET
	2011/10/10 16:02:39	192.168.10.90	http://tiny.cc/		GET
MSN 日志	2011/10/10 15:55:51	192.168.10.90	http://149.93.248.103/		GET
	2011/10/10 16:41:15	192.168.10.90	http://alegro.pl/		GET
Yahoo 日志	2011/10/10 16:23:40	192.168.10.90	http://megedick.com/		GET
	2011/10/10 16:25:51	192.168.10.90	http://149.93.67.16/		GET
	2011/10/10 16:35:16	192.168.10.90	http://149.93.201.51/		GET
	2011/10/10 16:53:42	192.168.10.90	http://149.93.19.98/		GET
	2011/10/10 16:53:55	192.168.10.90	http://38.229.157.155/		GET
	2011/10/10 17:26:55	192.168.10.90	http://skyrock.com/		GET
	2011/10/10 11:46:16	192.168.10.90	http://location.app.msn.com.cn/Location/GetLocation?callback=jsonCallback		GET
	2011/10/10 11:46:11	192.168.10.90	http://cmsgame.com/c.cgi?rid=4C071B8804164056095AD1EAD429F07D&ctt=1182187708634d+361p+3723sp+706/		GET
	2011/10/10 11:46:16	192.168.10.90	http://msn-allies.com/main/adfshow?user=MSNHome_Page!homepage_small_button1#b=msnborder#llocal#sysjs		GET
	2011/10/10 11:46:16	192.168.10.90	http://msn-allies.com/main/adfshow?user=MSNHome_Page!Homepage_small_button2#b=msnborder#llocal#sysjs		GET
	2011/10/10 11:46:16	192.168.10.90	http://msn-allies.com/main/adfshow?user=MSNHome_Page!CNNHOME_PAGE_BAN728x90_12debu		GET
	2011/10/10 11:46:16	192.168.10.90	http://simpeng1.s-msn.com/mangongai/hp/2011/10/08/480new4d-ed68-11a7-8a45-16d7f15a09af.jpg		GET

此外，本次使用的 D 样本，还采用了点对点（P2P）机制，使它能够从其他已经感染 Conficker.D 计算机中分配和接收命令。因此我们可以看到大量的向互联网上中了蠕虫病毒的 PC 发起的连接，如图矩阵视图：



如此，我们可以看到飞客蠕虫的威力，可以从 50000 个不同的域名随机选取 500 个进行解析，取得与黑客控制者的连接及下载其他节点信息。然后向其他中了飞客蠕虫的节点进行 P2P 连接，取得版本信息及黑客指令，这样不仅能够隐藏控制者的 IP，而且还使飞客蠕虫建立起来的僵尸网络更加健壮（P2P 方式的连接方式不会因为失去一个控制者而使整个僵尸网络瓦解）。

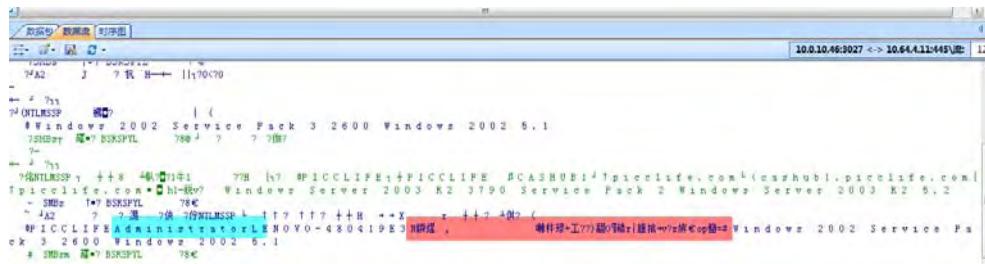
飞客蠕虫的传播性极强，如果内网中有一台 PC 因为访问含病毒网站，或含木马邮件等方式感染了飞客蠕虫，那么局域网其他有漏洞的主机感染的概率基本是 100%。飞客蠕虫对内网其他 PC 的入侵和感染主要是通过网络共享方式进行。而且飞客蠕虫可以通过其他硬件载体，比如 U 盘，移动硬盘等方式进行传播。

网络共享是 windows 比较方便的一个利用网络共享文件的方式，但给用户带来便利的同时也带来了很

大的隐患，局域网内的两大攻击手段就是 arp 感染和网络共享破解。飞客蠕虫是使用网络共享破解最多的蠕虫病毒之一，我们可以从其他的攻击数据上面进行验证。

首先，我们会发现网络中存在大量的 CIFS 和 netBIOS 协议，而且这些协议流量不是很大，但 TCP 会话数却很大。

这种会话具有明显的暴力破解行为，可以通过“数据流”选项来仔细对比，我们随机选择两个 12 个数据包的 CIFS 协议的数据流进行对比发现，其内容部分只有密码部分在改变如图：



根据其他安全资料，可以找出其尝试密码的列表，通过列表我们可以发现，这些密码都是比较常见的密码，因此也可以看到设置一个比较复杂的随机的密码的重要性。

The Conficker.B worm and weak passwords

by ASEEM123 on JANUARY 22, 2009

All admins should read the Microsoft Malware Protection Center's [analysis of the Win32/Conficker.B worm](#). It references the weak passwords the worm attempts to use. I've finally found a [list of these weak passwords on the Analysis tab](#). Here they are, make sure none are in use in your network!

```

123
1234
12345
123456
1234567
12345678
123456789
1234567890
123123
12321
123321
123abc
123qwe
123asd
123abcd
1234qwer
1q2w3e
a1b2c3
admin
Admin
administrator
  
```

如果局域网内某台有漏洞的 PC 感染了飞客蠕虫，在传播过程中碰到其他 PC 是弱密码或根本没有密码，那么被入侵就是迟早的事情了。

1. 飞客蠕虫的危害：

导致个人机密信息被窃取，如 QQ 密码，银行账号，个人或公司保密文件等。

感染局域网内其他主机，强大的传播性使其他虚弱的 PC 受到感染，造成大面积的感染。

大面积感染后会使大量消耗防火墙等设备的网络并发连接，影响其他 PC 的正常业务访问（firewall 的并发连接数是固定的，如果几十台 PC 感染了飞客蠕虫，导致的并发连接可能会消耗掉防火墙大多数连接数，从而导致正常用户也受到影响，这种就会出现，带宽足够但访问也会比较慢的情况产生。）

2. 防护：

规范用户的上网行为，对陌生邮件和危险网站进行禁止打开和访问。

个人 PC 要及时升级微软更新补丁，所谓“苍蝇不叮无缝蛋”就是这种道理。飞客蠕虫就是针对微软的一个漏洞，如果打了补丁修补了该漏洞，那么感染飞客蠕虫的概率就会很低了。

个人用户出现网络慢，或发现比较多的网络连接时，或发现自己的杀毒软件无法正常工作的时候可以使用测试页面测试以下是否感染了飞客蠕虫：<http://www.joestewart.org/cfeyechart.html>

一旦感染确定感染了飞客蠕虫后，要进行断网隔离，下载专杀工具进行查杀。

5.2. 参考文档：

<http://mtc.sri.com/Conficker/>

<http://www.confickerworkinggroup.org/wiki/> （飞客蠕虫工作组）

国家计算机应急指挥中心年度报告

<http://www.malwaredomains.com/>

6. 垃圾邮件行为分析

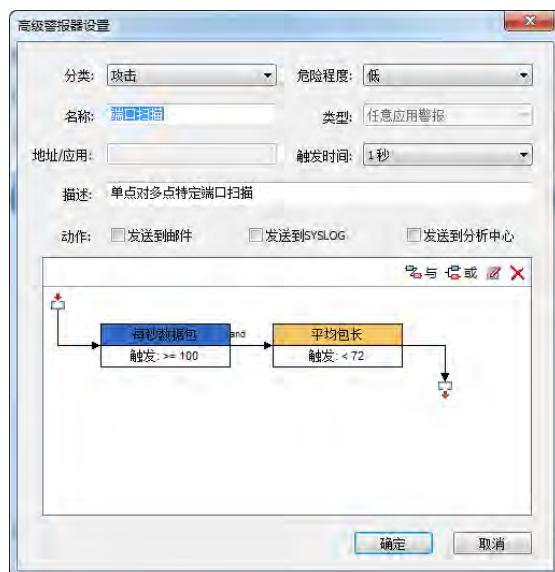
6.1. 案例背景

科来回溯分析系统最新的 3.1 版增加了很多新功能，丰富的实时警报功能就是其中之一。3.1 版的实时警报功能与 3.0 版相比可以说是一次质的飞跃，新版的警报功能即可以基于字节数、数据包数量、平均包长、TCP 特征统计等流量统计信息设置警报，还可以设置邮件敏感字、可疑域名检测以及报文特征值的警报。利用这些灵活的警报功能可以让网管人员及时发现各种故障和安全隐患。

本文就是一个利用科来回溯分析系统 3.1 版流量警报功能发现内网主机发送垃圾邮件的实例。

背景介绍

本文的网络环境是一家中国教育网用户的网络，内网使用公有 IP 地址，在其互联网出口部署科来回溯分析服务器，7*24 小时捕获互联网入出站流量。由于内网使用公有 IP 地址没有做 NAT，因此内网主机会直接面对来自互联网的各种威胁，端口扫描就是其中较常见的行为之一。为了及时监测端口扫描的行为，我们在分析服务器上设置了旨在发现特定端口的主机扫描行为的警报，如下图。



科来回溯分析系统 3.1 版可以灵活的利用与或逻辑关系设置复杂的警报触发条件。这个警报就是监测网络中任意应用，如果某应用 1 秒钟内数据包数量超过 100 个，并且平均包长小于 72 字节则触发警报。

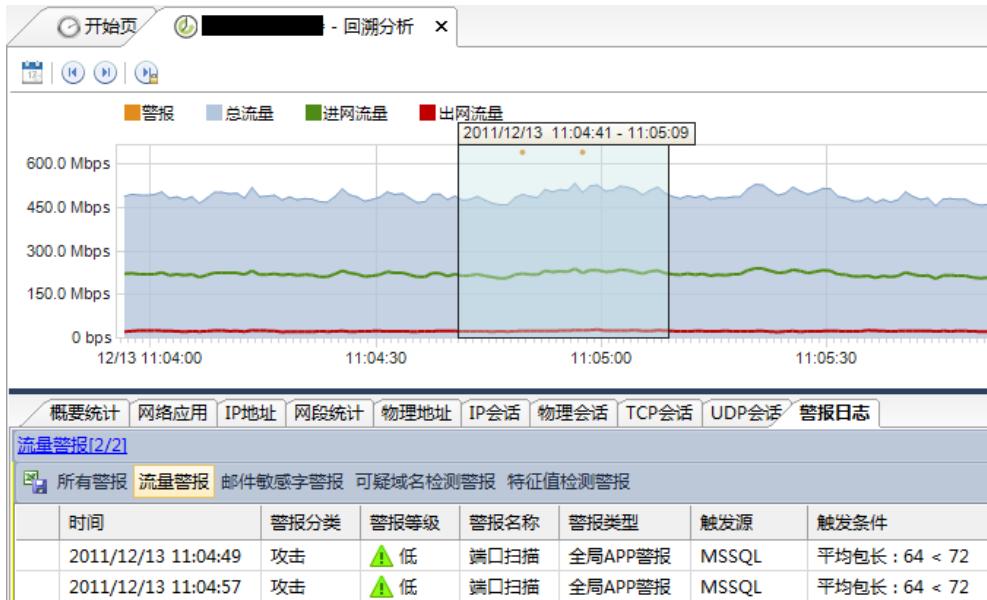
通常来自互联网主机扫描会针对特定服务端口（如 MSSQL 1433 端口），短时间内向一个网段内每个 IP 发送连接请求，如果发现有某主机有 TCP 同步确认回应则与该主机建立 TCP 连接，而后进一步尝试漏洞攻击或弱口令尝试。由于 TCP 同步包和同步确认包都没有上层数据，因此这种主机扫描行为的数据包都很小，一般不会超过 72 字节。

设置这个警报的初衷虽然是发现主机扫描行为，但是在实际使用时意外的发现某台内网主机在发送垃圾邮件时触发了这个警报。

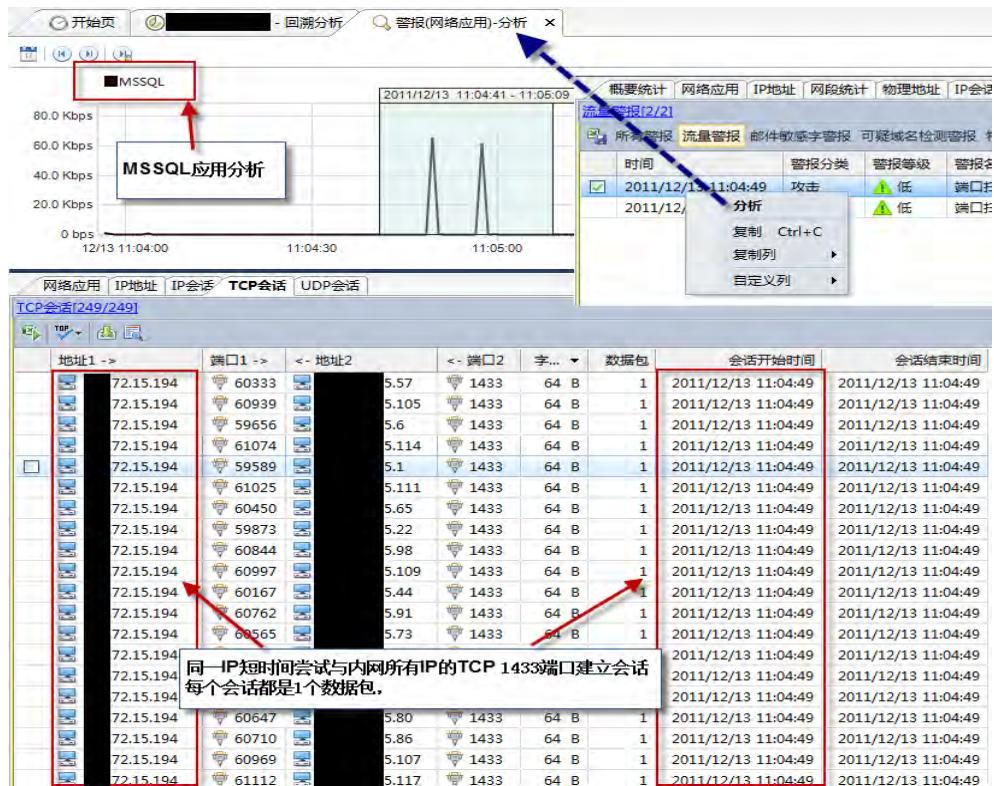
6.2. 主机扫描警报的基本效果

在设置案例中的主机扫描警报之前，我们要发现主机扫描行为通常是在“TCP 分析”趋势图中找 TCP 同步包的异常峰值，但是如果流量较大的网络中短短 1~3 秒的主机扫描行为所触发的 TCP 同步包增加往往会被忽略。例如案例中的网络工作时段 TCP 同步包量在每秒 400~600 之间，偶尔每秒增加 100 个并不是非常明显，因而很多主机扫描行为没有被及时发现。

在设置了案例中的警报之后，我们可以在控制台的趋势图中直观的看到每一次主机扫描的警报，同时在警报日志视图看到主机扫描所针对的应用服务，甚至不用切换到“TCP 分析”趋势图，如下图所示。



从上图中可以看到选中时段内有两次针对 MSSQL 应用的疑似主机扫描行为。3.1 系统还增加了针对选中对象的分析功能（如选中某应用或某 IP 地址），在这里我们选中其中某警报日志条目，点击鼠标右键，选择“分析”菜单项，就可以针对选中时段的 MSSQL 应用进行单独分析，这样可以快速判断警报是否误报，如下图。

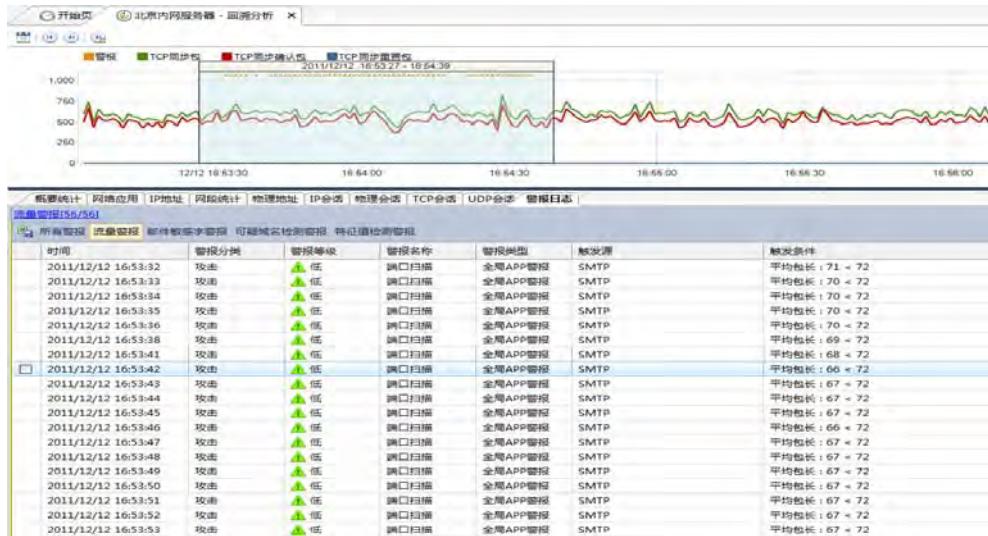


从上图中可以看出，警报发生的时段某外网 IP 在短时间内尝试与内网所有主机的 TCP 1433 端口建立连接，由于内网主机都没有安装 SQL Server，所以每个连接请求都没有得到应答，每个会话都只有 1 个数据包。可以断定这个外网 IP 在做全网段的 MSSQL 服务主机扫描。

在案例中的网络中，我们利用这个自定义的主机扫描警报发现了大量的类似主机扫描行为。这些行为的共同特点是发生时间很短（整个扫描过程一般在 3 秒内完成），一次扫描会触发 1~3 次警报。扫描针对的应用主要集中在 MSSQL、MySQL、Oracle 等数据库端口以及 CIFS、NetBios 等共享端口，通常这些端口会容易受到漏洞攻击或弱口令攻击。这些行为在使用 3.1 版本之前需要非常仔细的观察和分析才能发现，现在我们可以及时的发现并在边缘设备上针对这些扫描的端口或 IP 地址进行过滤，避免更大的安全问题发生。

6.3. 意外的SMTP主机扫描警报

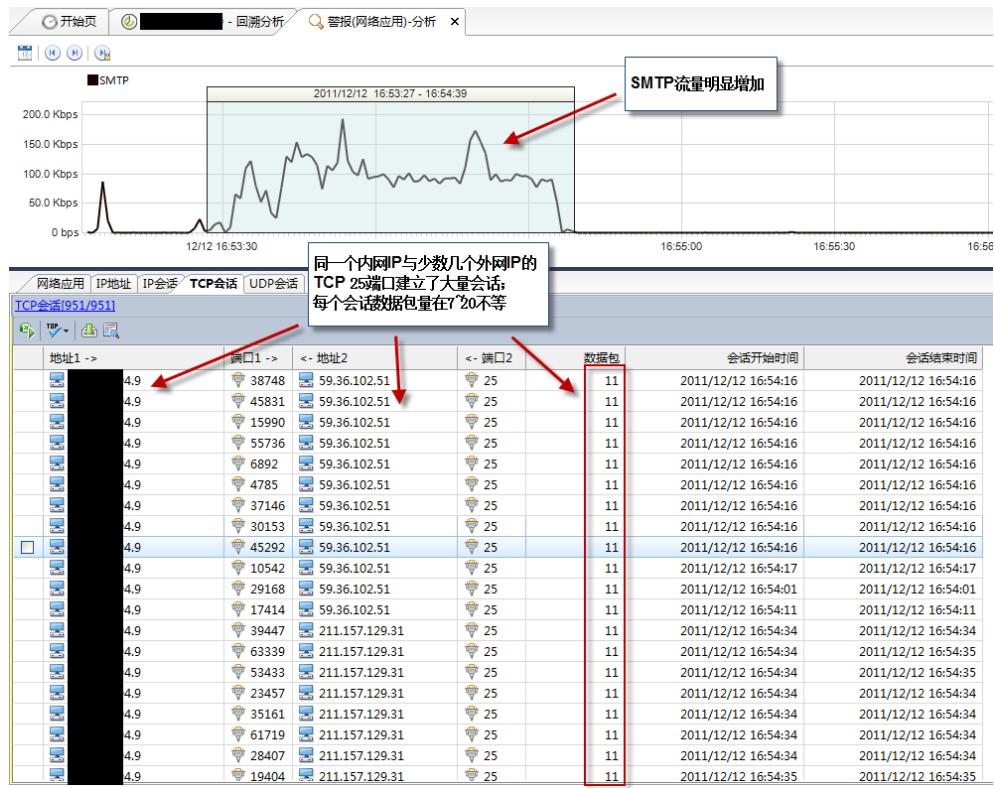
在配置了自定义主机扫描警报之后，偶然发现某个时段有大量的针对 SMTP 应用的主机扫描报警，报警的频繁程度明显超过了其他的主机扫描行为。



这次意外事件在 1 分多种的时间里触发了 56 次主机扫描警报，这明显与其他时段发生的主机扫描行为有区别。通常主机扫描者不会针对一个应用端口持续很长时间反复扫描。一般情况下，针对 SMTP 端口的 SYN flood 攻击才有可能持续触发我们定制的主机扫描警报，然而这种 SYN flood 攻击往往会在“TCP 分析”趋势图上看到明显的 TCP 同步包数量增加，而从上图的“TCP 分析”趋势图上却看不到这一现象。

为了进一步分析判断这一事件的原因，我们使用 3.1 系统的应用统计分析功能，对这一时段的 SMTP 会话进行了统计分析。

6.4. SMTP会话统计分析



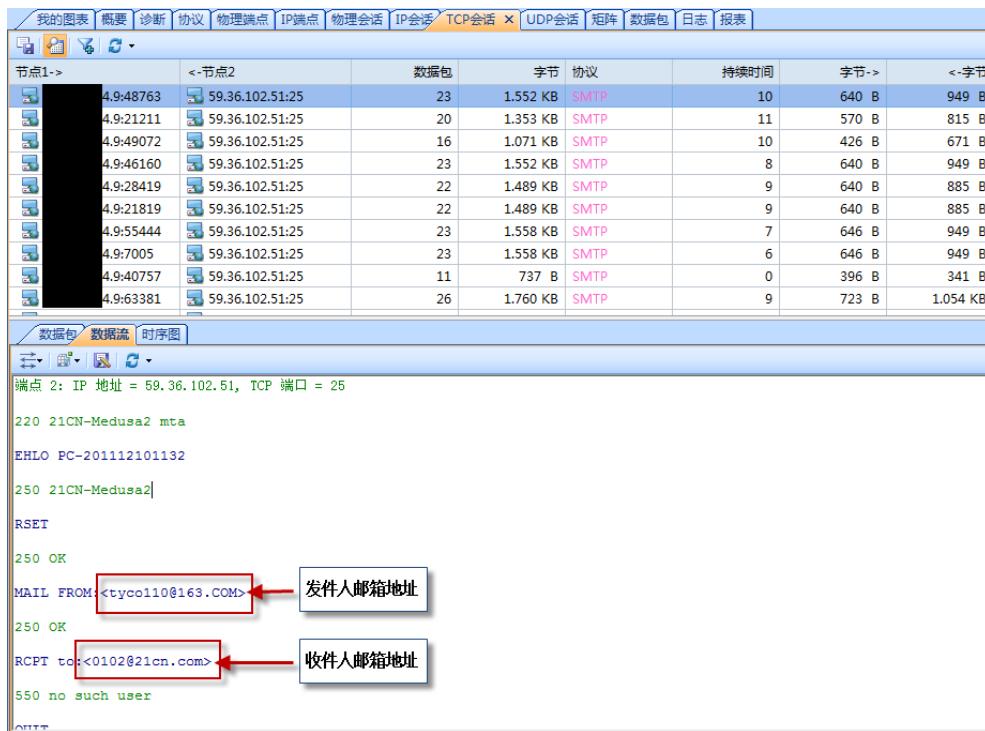
从上图中的流量趋势图上明显看到 SMTP 流量在警报发生的时段内有明显增加，最大流量超过 150Kbps；在 TCP 会话视图中，我们看到一个内网 IP 在短时间内与若干个外网 IP 的 TCP 25 端口建立了很多 TCP 会话，这些会话并不像主机扫描行为那样只有很少量的数据包，而是每个会话有几个到几十个不等（截图中碰巧都是 11 个数据包）。

至此基本排除了这些警报是主机扫描行为的可能性，但可以判断这些 TCP 会话不是正常的邮件发送，因为正常的邮件发送不会产生如此多的会话，而且正常邮件发送的平均数据包长度不会小于 72 字节。

要了解些异常会话的真正作用，就需要对这些会话进行数据包级解码分析，于是我们将这一时段的 SMTP 应用的数据包下载到控制台，利用控制台自带的科来网络分析模块进行解码分析。

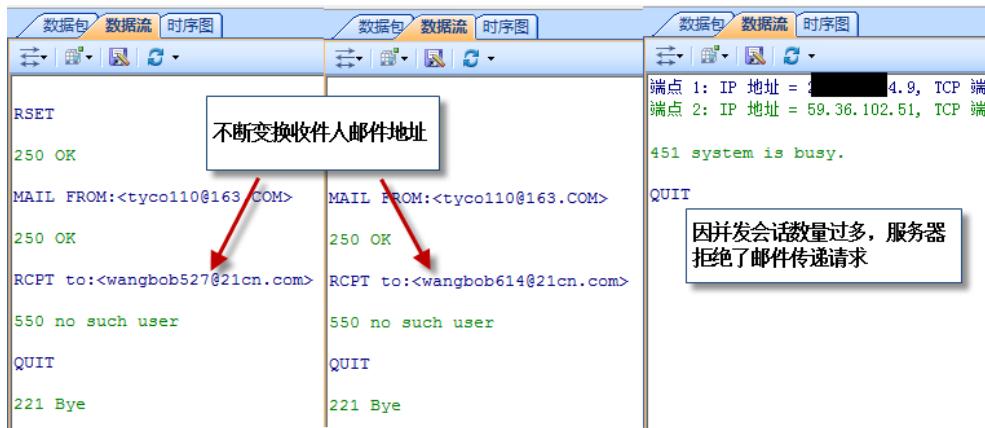
6.5. SMTP数据流解码分析

下载 SMTP 数据包后，定位到“TCP 会话”视图，并且选中某个会话，查看其“数据流”信息，能够完整展现一个 TCP 会话的应用层数据交互信息。



从数据流信息中我们看到 59.36.102.51 这个外网 IP 有可能是 21CN 的邮件服务器，触发警报的内网 IP 在尝试向 21cn.com 的某个不存在的用户发送邮件，21CN 的邮件服务器拒绝了这次邮件发送。

继续查看其他与 21CN 的 TCP 会话，发现每个会话的发件人都是“tyco110@163.com”，收件人邮箱地址在不断变化，每个会话的收件人都不相同但邮件后缀都是“@21cn.com”。说明这个内网 IP 的主机的邮件发送程序并不知道收件人的真实信息，而是在不断变换邮件前缀尝试向 21CN 的用户发送邮件。



由于该内网 IP 在短时间内向 21CN 的邮件服务器发起了大量 SMTP 会话，一段时间后 21CN 的邮件服务器拒绝了该 IP 的邮件发送请求。

在该内网 IP 与其他外网 IP 的 SMTP 会话中，我们看到了与 21CN 的会话相似的行为，这些外网 IP 包括“263.net”、“126.com”、“世纪互联”等多家邮件服务提供商或 IDC 的邮件服务器地址，还包括一些中小型 ICP 的邮件服务器地址。

在下载的全部 4000 多个 SMTP 会话中，成功发送邮件的会话不到 10 个，看来这个垃圾邮件发送程序的效率并不是很高，或者是内置的用户列表已经过时了。在几个成功发送的邮件会话中我们可以看到明显的垃圾邮件内容，如下图。

```
From: tyco110@163.COM
Subject: =?GB2312?B?xO01xNDFz6LK1bW9LsnMzvHNxrnsO/W+sTj?=
To: webmaster@16167.com
Date: Mon, 12 Dec 2011 17:14:38 +0800
X-Priority: 3
X-Mailer: CSM2.8

你的信息收到.商务推广帮助你只需7天，就可以提升网站排名。让你的信息遍布互联网，在十大搜索引擎(关键词不限)百度 Google雅虎 新浪搜狐 MSN YOKTOM QQ等著名引擎及数万个搜信信息发布站点，都逐步排名第一。具有投资小(同样效果的广告数十万元)、效果好等特点，是企业网站推广及产品信息发布的最佳选择。在百度或Google搜索到所发布的信息和时间记录，花费极低-效果却如此立竿见影。登录过程及结果完全可视化透明运作，结果真实可靠。座在电脑“钱”看订单.咨询电话:全球通1: [REDACTED]
```

至此，我们可以确定一系列的 SMTP 主机扫描警报是这个持续的效率不是很高的垃圾邮件发送行为引起的。

6.6. 案例总结

通过这个案例的分析过程，我们可以总结一下几点经验：

科来回溯分析系统 3.1 版强化的警报功能可以更加灵活的用来及时发现多种故障隐患和安全隐患，例如可以更加方便的发现主机扫描行为；

警报功能本质上是模式比对，这一点与 IDS 的警报相似，由于存在行为模式相似的网络行为，这使得精心设计警报出现误报（包括 IDS 警报）。与 IDS 等常规安全管理产品相比回溯分析系统的优点在于可以对每一次警报进行深入挖掘和分析，以验证警报的真实性，避免误报或漏报对网络管理带来的影响。

尤其是在遇到警报出现频率有明显变化时，对警报相关的流量进行深入分析，往往能够及时发现安全或故障隐患。

在数据包解码分析过程中，我们发现不同的邮件服务器对此次垃圾邮件的处理方式也不尽相同，有些邮件服务器由于配置了 SPF（Sender Policy Framework）能够在发送方提交发送者邮件地址时对其 IP 有效性进行验证，从而更及时的阻止垃圾邮件发送行为，减小这些行为对其邮件服务器带来的性能影响，如下图。

```
EHLO PC-201112101132
250-XFORWARD NAME ADDR PROTO HELO SOURCE
250-263.net
250-SIZE 47185920
250-ETRN
250 8BITMIME

RSET

250 Ok

MAIL FROM:<tyco110@163.COM>

520 ip and spf record not match

QUIT
```

7. 一次端口扫描行为的分析案例

7.1. 案例背景

端口扫描是网络中常见的行为之一。网络管理员利用端口扫描可以检测自己网络的健康状况，用以修补漏洞、制定完善的安全策略；黑客利用端口扫描可以发现目标网络/主机中存在的漏洞，为后续的进一步入侵做准备。本案例是一次典型的针对网络中 Windows 系统和 MS SQL Server 数据库服务器漏洞的端口扫描行为。

1. 环境说明

本案例为某实验性网络，内部主机使用公有 IP 地址。核心交换机上部署了科来回溯式分析服务器，通过端口镜像将内部网络的流量导入回溯式分析服务器。

2. 案例分析

某日上午在分析系统控制台上将趋势图表设置为“TCP 分析”状态时，偶然发现有两个时刻网络中 TCP 同步包数量明显增多。TCP 同步包最多时达到了 TCP 同步确认包的两倍还多，而通常情况下 TCP 同步包数量只会略多于 TCP 同步确认包。于是我选取了其中一个峰值约 15 秒的时间段，在“IP 地址”浏览页面按照“发 TCP 同步包”进行排名，发现某 IP 地址 15 秒内发送了 773 个 TCP 同步包排名第一，而该 IP 总发包量才 868 个。这显然不是一个正常现象。



于是下载该 IP 的数据包进行深入分析。在 IP 会话列表中看到该 IP 地址与内网的每一个 IP 都有 IP 会话，这是对网段内所有主机进行扫描的典型特征。

源IP	目的IP	端口	协议	状态	连接数	发送字节数	接收字节数	平均速率	丢包率	重传次数	建立时间	最后发送时间
123.76.75.116	123.76.75.116	239	TCP	建立	0	192 B	0 B	3 B/s	0 %	0	09:14:40	09:14:40
123.76.75.116	123.76.75.116	240	TCP	建立	0	192 B	0 B	3 B/s	0 %	0	09:14:40	09:14:40
123.76.75.116	123.76.75.116	241	TCP	建立	0	192 B	192 B	0 B/s	1 %	5	09:14:40	09:14:40
123.76.75.116	123.76.75.116	242	TCP	建立	0	192 B	192 B	0 B/s	1 %	6	09:14:40	09:14:40
123.76.75.116	123.76.75.116	243	TCP	建立	0	192 B	192 B	0 B/s	1 %	1	09:14:40	09:14:40
123.76.75.116	123.76.75.116	244	TCP	建立	0	192 B	192 B	0 B/s	1 %	5	09:14:40	09:14:40
123.76.75.116	123.76.75.116	245	TCP	建立	0	192 B	192 B	0 B/s	1 %	3	09:14:40	09:14:40
123.76.75.116	123.76.75.116	246	TCP	建立	0	192 B	192 B	0 B/s	1 %	3	09:14:40	09:14:40
123.76.75.116	123.76.75.116	247	TCP	建立	0	192 B	192 B	0 B/s	1 %	3	09:14:40	09:14:40
123.76.75.116	123.76.75.116	248	TCP	建立	0	192 B	192 B	0 B/s	1 %	3	09:14:40	09:14:40
123.76.75.116	123.76.75.116	250	TCP	建立	0	192 B	192 B	0 B/s	1 %	3	09:14:40	09:14:40
123.76.75.116	123.76.75.116	251	TCP	建立	0	192 B	192 B	0 B/s	1 %	8	09:14:40	09:14:40
123.76.75.116	123.76.75.116	252	TCP	建立	0	450 B	192 B	258 B	6 %	9	09:14:40	09:14:40
123.76.75.116	123.76.75.116	253	TCP	建立	0	384 B	192 B	192 B	6 %	3	09:14:40	09:14:40
123.76.75.116	123.76.75.116	255	TCP	建立	0	102 B	102 B	0 B/s	1 %	3	09:14:40	09:14:40
123.76.75.116	123.76.75.116	256	TCP	建立	0	102 B	102 B	0 B/s	1 %	5	09:14:40	09:14:40
123.76.75.116	123.76.75.116	203	TCP	建立	7	10,128 KB	5,962 KB	4,166 KB	91 %	51	09:14:45	09:14:45
123.76.75.116	123.76.75.116	201	TCP	建立	7	5,218 KB	3,322 KB	1,896 KB	41 %	23	09:14:45	09:14:45
123.76.75.116	123.76.75.116	249	TCP	建立	6	8,827 KB	5,674 KB	3,153 KB	81 %	47	09:14:40	09:14:40

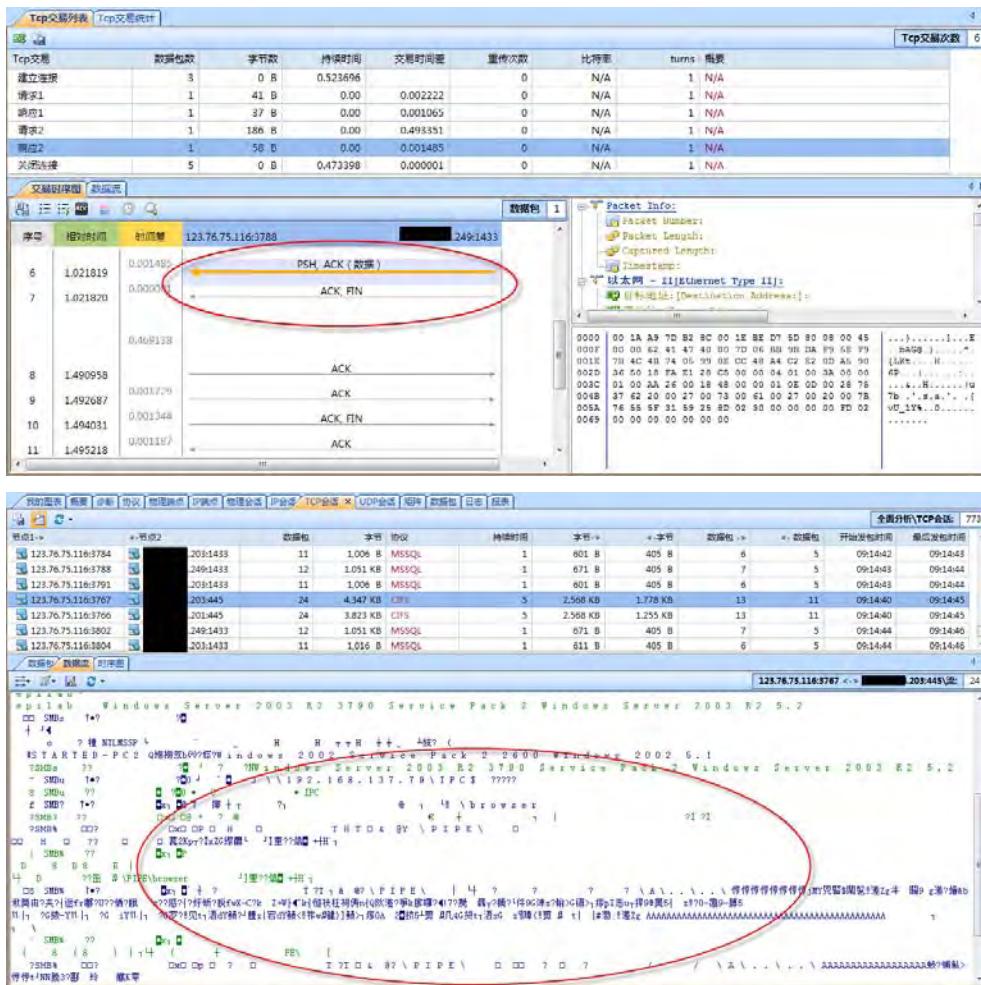
从上图中可以看出攻击者对每台主机发送了至少 3 个 TCP 同步报文，目标端口是每台主机的 RPC（135）、MSSQL（1433）和 CIFS（445）。图中数据包总量为 3 的是主机不存在或未作响应；数据包为 6 的是主机对扫描着回应了 TCP 重置或 ICMP 目标不可达消息，表示攻击者访问的端口没有开放；而有几台主机与攻击者交换了几十个数据包，说明在这几台主机上的上述 3 个端口有一个或多个可以访问，攻击者对这几台主机进行了深入的漏洞扫描。

TCP 135 和 445 是用于 Windows 远程过程调用和文件共享的端口。在 Windows 2003 SP1 之前的系统中这两个服务存在比较大的漏洞，常被一些蠕虫病毒利用，如早年的冲击波和震荡波，攻击者也会利用这两个端口对系统进行入侵。

TCP 1433 是 SQL Server 的服务端口。黑客可以利用它进行弱口令尝试，如果成功就可能获得目标主机的系统权限。

为了看到攻击者对那几台开放端口的主机做了什么，我把界面切换到“TCP 会话”，深入分析攻击者进行漏洞扫描的行为。

此次针对 SQL Server 的扫描每次会话为 11 到 12 个报文不等，选取其中某个会话在数据流页面中能够明显看到攻击者在尝试 sa 口令，上图中服务器的回应数据我看不太明白，只是从服务器在回应口令尝试后立刻终止了会话来推测此次尝试并未成功。



从针对 CIFS 的扫描会话的数据流视图中能够明显的看出 IPC\$连接请求，和命名管道的访问请求，可以看出这是针对 Windows 2003 SP1 以前版本"\pipe\browser"命名管道漏洞的攻击尝试。被扫描系统是 Windows Server 2003 R2 SP2 并不会受到影响。

建议：此次端口扫描过程时间不长，但类似的行为曾多次出现，并且在其他时段发现了数个不同的源 IP 地址在对内网进行扫描。虽然都没有造成实质的破坏，但还是建议在边缘设备上过滤不必要的 TCP 端口访问，尤其是 135、445、1433 等通常只对内网提供服务的端口。

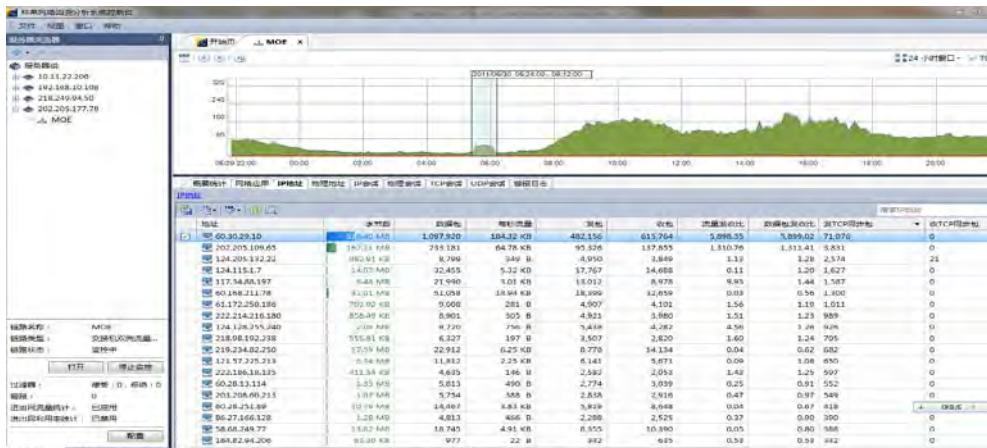
8. 回溯式异常流量分析

异常流量指的是区别于正常流量的通信，通常表现为一些流量，数据包，TCP 同步，TCP 重置等参数，在一段时间内突然爆发式的增长，有别于正常情况流量。引起异常流量的原因通常是一些攻击，下载，或扫描等不正常的网络行为。科来回溯式分析系统能够记录下网络通信的数据包，控制台直观的趋势图可以很快发现网络异常流量，对异常流量进行快速的定位和分析，找出异常流量的根源。

8.1. 流量突起分析

某单位部署了科来回溯式分析系统 RSA3000，镜像本单位的服务器流量出口。该网络出口流量为 50M，主要提供 WEB, FTP, Mail, 和各种查询等服务。在没有部署回溯式之前，防火墙和入侵检测发现过多次告警日志，但因为没有告警时数据包，而且告警次数每天有几百次造成无法及时处理。

在一次例行的检查中，我们从流量趋势图中发现该单位在凌晨 5 点左右有一个流量突起，而这个时段应该没有什么业务流量。如图：

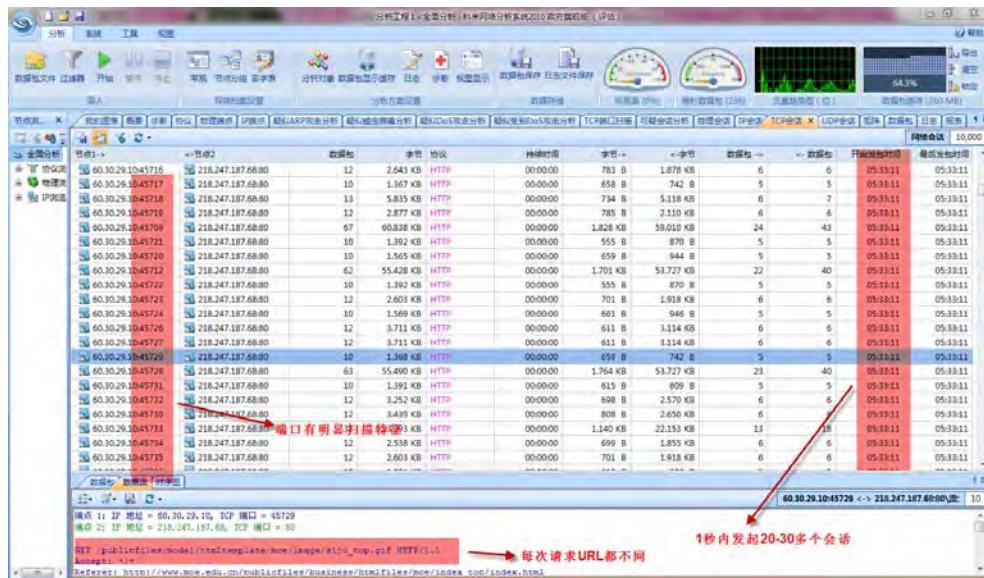


我们看到在突起发生的时间段位 6-30 日 5:24---6:12。选中该段时间进行分析，从 IP 地址栏可以看到，60.30.29.10 IP 流量达到 528MB 远远超过其他 IP 应该是引起此处流量突起的原因。而 IP 202.205.109.65 是该网络的服务器 IP。对 60.30.29.10 进行挖掘分析发现该 IP 的大多数 TCP 会话都是与 202.205.109.65 通信产生的。

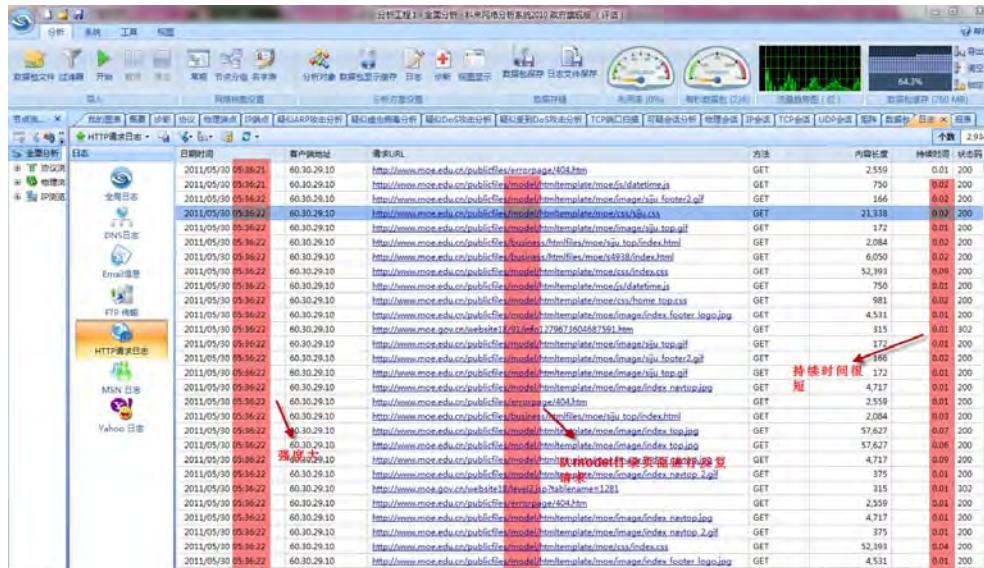
对 60.30.29.10 的数据进行下载分析，我们看到该 IP 在突起时间段内产生了 12431 次 TCP 会话，产生了 149MB 的流量。如图：



打开“TCP 会话”看到该 IP 的通信会话都很整齐，每次会话数据包多为 10-30 个，每次请求的页面都不同。而会话时间很短，每秒钟能进行 20 多次这种会话



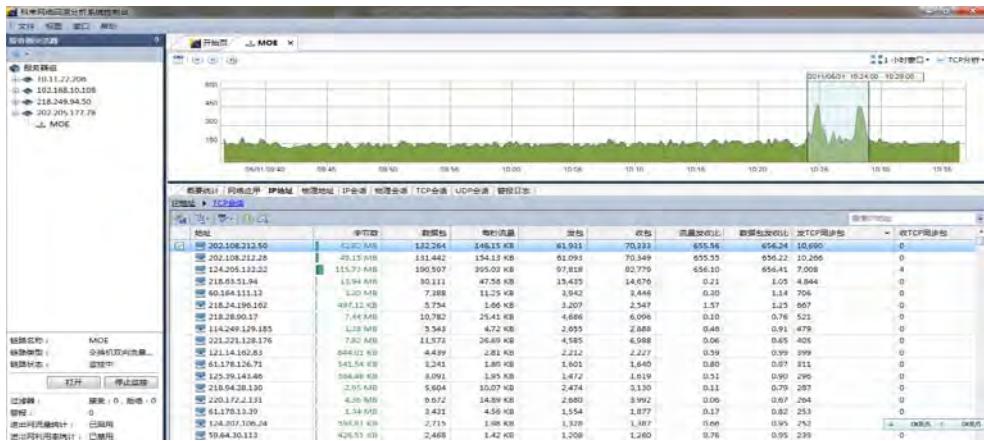
查看 HTTP 日志也会看到很详细的访问情况，如下图：



综合日志，TCP 会话，和其流量行为我们判断 60IP 是在对该单位的 WEB 服务器做扫描渗透攻击，使用攻击软件对 WEB 服务器进行扫描，企图找到 WEB 漏洞，然后进行入侵和提权。该行为造成了流量突起的产生。

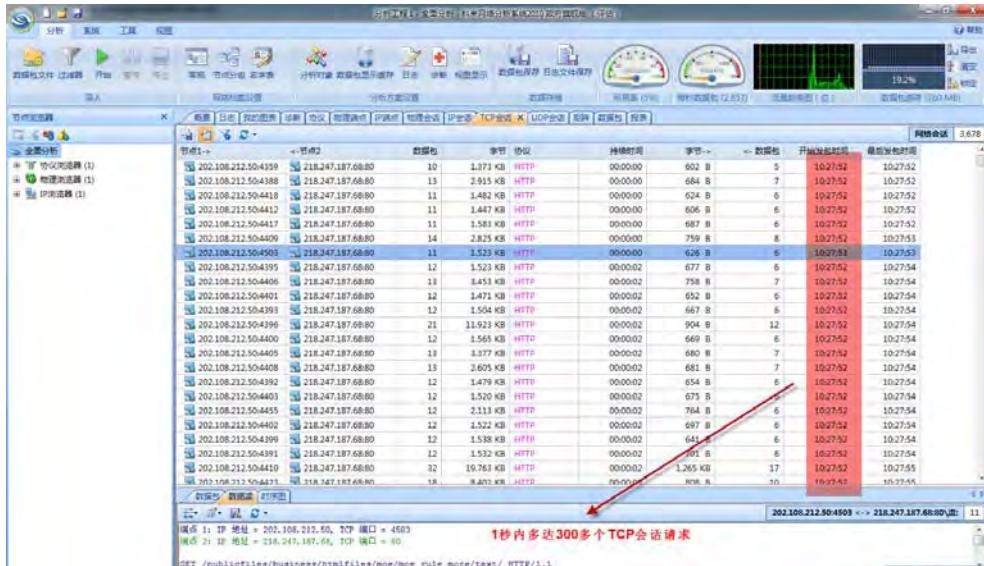
8.2. 案例 2 TCP 请求异常分析

某单位在网络中部署了科来回溯式分析系统，用以日常的安全检测和事件分析，为网络管理人员提供详实准确的网络流量信息。某次在分析数据时发现网络中有一些流量异常的 TCP 请求峰值如图：

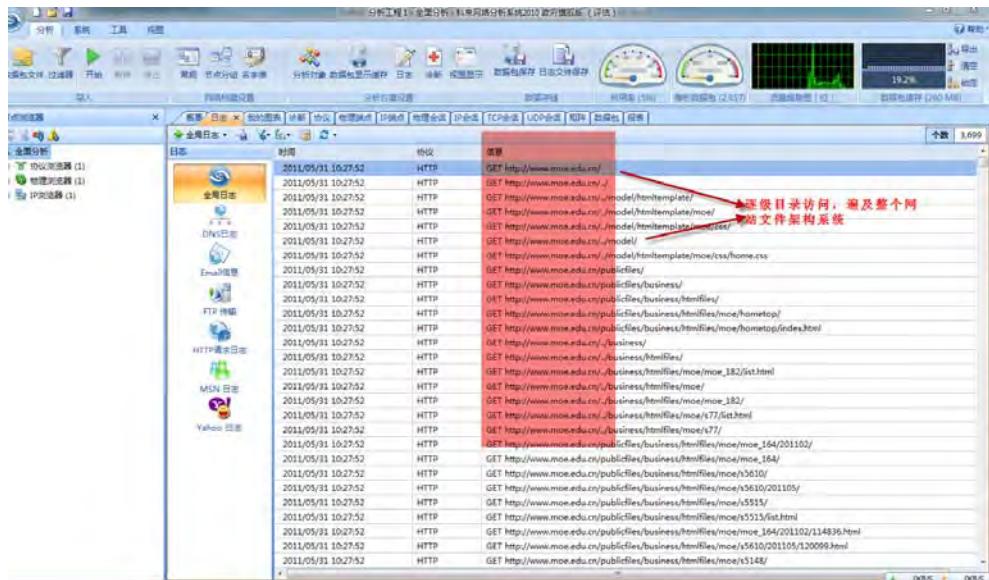


我们可以看到，选中 1 小时分析窗口，在 10:24—10:29 这 5 分钟产生了两次 TCP 请求的峰值，峰值时 TCP 同步请求达到了 450 个，而平时 TCP 同步为 150 个。在控制台选择 IP 地址，然后对 TCP 同步发送选项进行排名，我们发现两个 IP 202.108.212.50 和 202.108.212.28 在这 5 分钟内 TCP 同步发送达到了 10690 和 10266 个。

我们下载这段数据包首先看其 TCP 会话，我们发现外网 IP 202.108.212.50 和 202.108.212.28 在对该单位网站 218.247.187.68 进行高频率的“访问”。而其频率之高绝对不是人工点击网站所能形成，我们看到在 1 秒时间内会话请求就达到 300 多个



而这种会话具有明显的扫描特征，我们查看 HTTP 日志可以看出 202.108.212.50 对该单位网站请求有明显按照目录逐级访问的特征如图



8.3. 结论

如此，我们可以确定，TCP 峰值的产生是由于 202.108.212.50, 28 对网站进行扫描所致。这种扫描对网站产生一些威胁，而且也对服务器造成很大压力，影响其他用户的正常访问。

从两个案例我们看到回溯式分析对网络安全的重要意义，通过对异常流量的分析快速定位引起异常流量的原因，了解网络中存在的攻击现象，从而对网管人员指定反制措施，加固网络有了指导意义。

9. 黑客攻防技术入门之ARP篇

ARP (Address Resolution Protocol)，即地址解析协议。所谓地址解析，也就是将 IP 地址转换为 MAC 地址的过程。在局域网中，任何两台计算机之间进行通信前，都必须知道对方的 MAC 地址，所以 ARP 这个协议就非常重要。但如果该协议被恶意用于对网络进行攻击，会对局域网产生重大影响，甚至导致网络瘫痪。下面就将对 ARP 攻击的原理，类型以及如何用科来对 ARP 攻击进行定位，排除等。

ARP 欺骗攻击的类型大致分为两种：一种是对路由器 ARP 表的欺骗，另一种是对内网电脑的网关进行欺骗。

对路由器进行 ARP 表的欺骗：对路由器发送一系列错误的内网 MAC 地址，长时间不断发送，冲刷路由器的 ARP 表，使得路由器 ARP 表中都是错误信息，转发数据的时候都发向错误的 MAC 地址，造成正常的电脑无法收取信息，而假冒者可以窃取相关的信息。

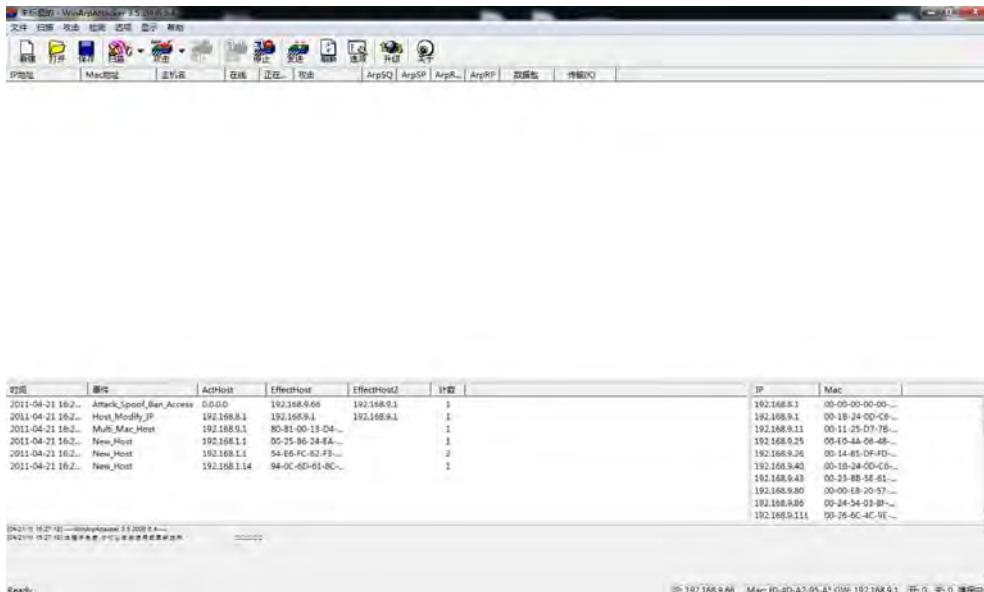
对内网电脑的网关进行欺骗：主要是通过建立假网关，让被他欺骗的电脑向这个假网管发送数据，而不是通过正常的路由器途径上网，这种欺骗造成的结果就是网络掉线。

那么攻击者是如何来进行 ARP 攻击的呢？

在这里向大家介绍一款软件，名称为 WinArpAttacker，我们可以用这个来做一下实验。

在使用 WinArpAttacker 之前，首先需要安装 Winpcap，用于为应用程序提供访问网络低层的能力的软件。

然后我们打开 winarpattacker。



其界面如此。

在实行攻击之前，我们首先要对整个局域网内的计算机进行扫描，以确定要攻击的主机。

IP地址	MAC地址	主机名	租期	正在.../续租	ARP[SQ]	ARP[ST]	ARP[RT]	数据包	传输(RQ)
192.168.9.1	00-0C-0E-1D-C...	192.168.9.1	在租	Nor... Normal	0	7	0	23	0
192.168.9.11	00-13-25-07-7...	BW	在租	Nor... Normal	0	0	0	18	0
192.168.9.25	00-03-04-09-4...	PC_20110325	在租	Nor... Normal	0	0	0	18	0
192.168.9.26	00-1A-05-0F-F...	MICROSOFT...	在租	Nor... Normal	0	0	0	18	0
192.168.9.40	00-1B-24-0D-L...	192.168.9.40	在租	Nor... Normal	0	0	0	18	0
192.168.9.43	00-23-0B-5E-6...	CHINA-BITS...	在租	Nor... Normal	0	0	0	18	0
192.168.9.66	00-4D-A2-95-A...	COLASOFT-PC	在租	Nor... Normal	252	6	4	47	0
192.168.9.66	00-00-E8-20-S...	PC_20100608	在租	Nor... Normal	0	0	0	18	0
192.168.9.66	00-00-E8-20-S...	192.168.9.66	在租	Nor... Normal	0	0	0	18	0
192.168.9.111	00-00-06-09-8...	192.168.9.111	在租	Nor... Normal	0	0	0	18	0
192.168.9.112	00-05-0C-AC-A...	192.168.9.112	在租	Nor... Normal	0	0	0	18	0
192.168.9.113	00-05-0C-AC-A...	192.168.9.113	在租	Nor... Normal	0	0	0	18	0
192.168.9.115	00-04-09-0D-L...	PC_20100715	在租	Nor... Normal	0	0	0	18	0
192.168.9.125	00-04-0E-08-F...	KM-007	在租	Nor... Normal	0	0	0	18	0
192.168.9.128	00-30-08-46-C...	192.168.9.128	在租	Nor... Normal	0	0	0	18	0
192.168.9.133	00-00-4E-59-L...	WWW-SJ095...	在租	Nor... Normal	0	0	0	18	0
192.168.9.134	00-30-08-10-2...	985571D0995...	在租	Nor... Normal	0	0	0	18	0
192.168.9.134	00-30-08-10-2...	192.168.9.134	在租	Nor... Normal	0	0	0	18	0
192.168.9.140	00-1C-04-88-E...	PC_20110904	在租	Nor... Normal	0	0	0	18	0
192.168.9.158	1C-06-05-ED-N...	GUEST	在租	Nor... Normal	1	0	0	18	0
192.168.9.166	00-1B-01-CD-A...	FORMY-PC	在租	Nor... Normal	1	0	0	18	0
192.168.9.175	00-0C-C4-CD-9...	WWW-40446...	在租	Nor... Normal	0	0	0	18	0
192.168.9.176	00-0D-47-99-5...	AEFE6080511...	在租	Nor... Normal	0	0	0	18	0
192.168.9.184	88-8E-10-8D-D...	192.168.9.184	在租	Nor... Normal	0	0	0	18	0

[04/21/11 08:26:25] Scanning ports online status warning.
[04/21/11 08:26:25] Start to print because of Local_Ac_Entry_Change error
[04/21/11 08:26:25] Start to print because of Local_Ac_Entry_Change error
[04/21/11 08:26:25] Start to print because of Local_Ac_Entry_Change error

Ready

IP: 192.168.9.66 Mac: F0-4D-A2-95-A1 GW: 192.168.9.1 If: 25 Unit: 0

单击扫描以后，我们可以看到，整个局域网 192.168.9.0/24 内的所有计算机的 IP 地址、主机名和 MAC 地址都显示出来了。

在扫描的时候，打开科来软件……我们瞬间捕捉下来了扫描的过程，见下图：

The screenshot shows the Colasoft Network Monitor interface. On the left, the 'Diagnostic Items' tab is selected, showing a table with columns: Name, Count, and Type. It lists various diagnostic categories with their counts: All Diagnostics (17), Transmission Layer (2), Network Layer (14), and Data Link Layer (1). The Data Link Layer section has a warning icon for 'ARP Request Storm'. On the right, the 'Diagnose Occurrence Address' tab is selected, showing a table with columns: Name, Physical Address, IP Address, and Count. It lists two entries: F0:4D:A2:95:A5:F7 (Physical address FF:FF:FF:FF:FF:FF) and FF:FF:FF:FF:FF:FF (Physical address FF:FF:FF:FF:FF:FF), both with a count of 1.

The screenshot shows the 'Diagnostic Events' tab. A single event is listed: 'ARP Request Storm (The host's MAC address is F0:4D:A2:95:A5:F7, IP address is 192.168.9.66)'. The event details show it occurred at 00:18:FC-FD:AF on 2011-04-21 08:26:25, with a severity of 'Warning' and type 'Data Link Layer'. The source IP is 192.168.9.66 and the source physical address is F0:4D:A2:95:A5:F7.

双击打开诊断事件，我们观察一下数据包的内容

The screenshot shows a detailed view of an ARP request packet. The packet information table includes columns: Time, Date, Source, Destination, Protocol, Port, Status, Size, Throughput, and Duration. The packet details pane shows the raw hex and ASCII data. The packet is an ARP request from 192.168.9.1957 to 192.168.9.66, type 0x0806, and operation 0x0003. The packet content shows the source MAC as F0:4D:A2:95:A5:F7 and destination MAC as FF:FF:FF:FF:FF:FF. The packet bytes are displayed in hex and ASCII format, with some fields like 'Source MAC' and 'Destination MAC' highlighted.

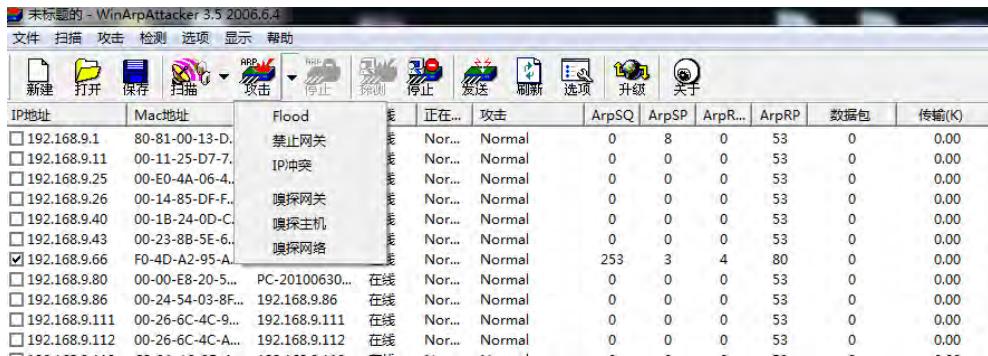
其中 9.66 就是我试验用的主机，可以很明显看到，时间差几乎在 1 微秒，大量持续的扫描

数据包	数据包 ->	<- 数据包	开始发包时间	最后发包时间
254	254	0	16:28:56	16:28:56

在一秒不到的时间内，发送了 254 个 ARP 包进行扫描，最终得出了整个局域网的情况。

在平时，我们在诊断中发现有 ARP 请求风暴的时候，可能就是黑客正在利用 ARP 扫描来进行对局域网内主机信息的分析，我们就要当心，及时做好防范措施。

在扫描完了之后，我们可以选中其中一台主机，来进行攻击。



在这里，笔者设定 flood 是 1000 次，进行攻击后，用科来软件进行分析



几乎是在瞬间，完成了攻击，这样的攻击，如果硬件性能不够好，可能会在瞬间崩溃，直至宕机等严重的情况。



可以看到，攻击用的 MAC 地址完全是假冒的 01:01:01:01:01:01，在科来软件的诊断中，就出现了 ARP 格式违规。

查看源 IP 地址

源IP地址
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66
192.168.9.66

就可以找出攻击来源。

再尝试禁止网关的攻击。我们可以在数据包中分析：

192.168.9.1 在 80:81:00:13:D4:45
192.168.9.1 在 00:1B:24:0D:C6:33

由于不断在误导计算机错误的网关，导致了该计算机的数据转发向了一个虚假的地址，最终导致无法正常上网。

最后，我们来尝试下 IP 地址冲突的攻击。从科来软件的诊断功能中，我们可以看到：

诊断条目		诊断发生地址		
名字	数量	名字	物理地址	IP地址
所有诊断	79	01:01:01:01:01:01	01:01:01:01:01:01	-
传输层	16	00:1B:24:0D:C6:33	00:1B:24:0D:C6:33	-
TCP 重复的连接尝试	14	00:25:86:24:EA:40	00:25:86:24:EA:40	-
TCP 慢应答	2	FF:FF:FF:FF:FF:FF	FF:FF:FF:FF:FF:FF	-
网络层	57	80:81:00:13:D4:45	80:81:00:13:D4:45	-
IP 地址冲突	57	F0:4D:A2:95:A5:F7	F0:4D:A2:95:A5:F7	-
数据链路层	6			
ARP 格式违规	1			
ARP 扫描	4			
ARP 太多的主动应答	1			

诊断事件				
严重程度	类型	层别	事件描述	诊断事件
安全	网络层	数据包 80 与 80 发生地址冲突	源IP地址 192.168.9.66 源物理地址 01:01:01:01:01:01	1

在几秒钟之内，就出现了 57 次的 IP 地址冲突，在诊断事件中打开详细的数据包内容：

192.168.9.66 在 01:01:01:01:01:01
192.168.8.1 在 80:81:00:13:D4:45
192.168.9.66 在 F0:4D:A2:95:A5:F7

我们可以通过解码看到，9.66 不停地再宣告自己在 01:01:01:01:01:01 和 F0:4D:A2:95:A5:F7，很明

显产生了冲突，造成了原本发送向 9.66 的数据可能会被丢弃的这种情况，用户就会感受到网速变慢，甚至无法上网等。更有甚者可以用自己的 MAC 地址来假冒正常用户的 MAC 地址，以达到窃取信息的目的。

通过前面的分析，我们大致上了解了，如何用科来软件对 ARP 攻击做出快速的定位分析。那么，当我们查出了攻击来源，受到攻击的机群，我们如何来防御 ARP 攻击呢？

我们可以使用 ARP 病毒专杀工具来对 ARP 病毒进行查杀。

使用 ARP 防火墙，这也是在企业中比较常用的软件。

绑定 MAC 地址。

前面两种方法都是通过自动的方式进行的，而绑定 MAC 地址则需要手动来进行，在本机上绑定，我们可以使用 **arp -s IP MAC** 的命令来进行。而在路由器上的设置则比较简单，大家可以自行去路由器上进行尝试。

本文主要介绍了一种入门级的攻击方式，而黑客在进行攻击的时候，会有一整套的流程进行，如何应对他们的攻击，需要我们再深入研究他们的攻击手段，科来软件是一个非常有帮助的工具，通过它来进行分析，能够快速定位到攻击源头以及受害机群，可以及时对问题进行排除，从而避免了大量的损失。

10. 某省某报业分析报告

10.1. 测试概述

1. 测试描述

测试软件：科来网络分析系统 2010

测试部署：核心交换机上做镜像

采样日期：2011-11-24

技术支持：科来软件某省办事处

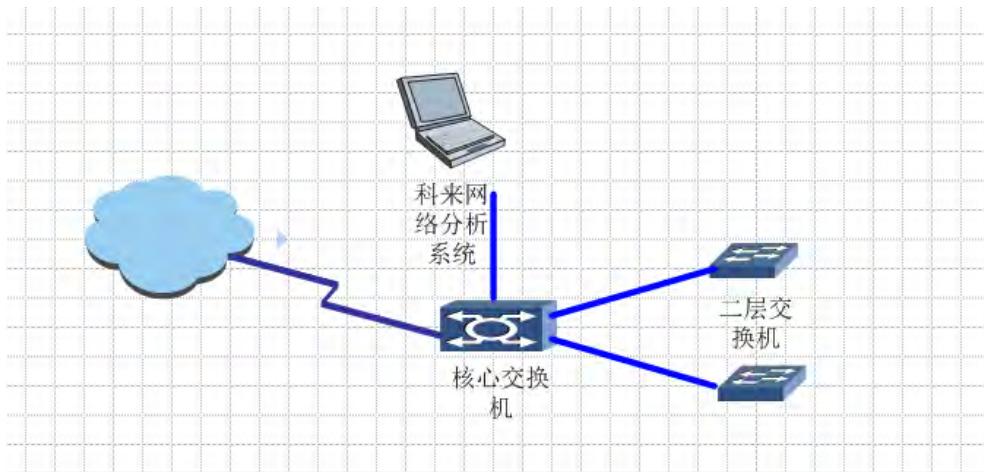
2. 故障现象描述

- 1) 外访问外网很慢，在交换机上查看交换机路由器的使用率也不高

某单位是网络结构很普通，规模不大，核心交换机上联一台路由器上外网，下面是两台普通交换机。最近一周当内网机器访问外网时变的很慢，通过查看交换机路由器的使用率等发现并不高，管理员在内网杀了杀毒也没发现病毒。

最后管理员和我们联系，我们对其进行一次抓包分析。

2) 网络拓扑图



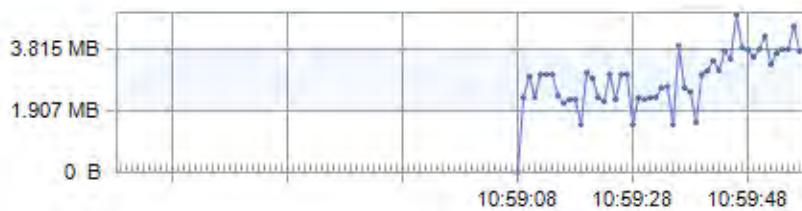
我们在核心交换机上做全端口的景象，在科来网络分析工具在核心交换机上抓取数据包。

10.2. 分析情况

1. 网络中的异常诊断信息

在抓取一段时间的数据包后我们可以看到几个重要的参数：

1) 网络实时流量



网络最高时流量 4MB/S 左右，并不是太高，我们的带宽并没有被占满。

2) 全局利用率



观察全局利用率我们看到网络的利用率最高百分之四十，还在我们的承受范围内，这样的利用率虽然对我们的网络有一定的影响，但是不应该是造成现在这种情况的主要原因，我们接着分析。

3) 我们在协议模块看到 UDP 的流量在这么短的时间内很大，平时我们网络中的业务流量不会这么大

通过协议浏览器，打开 UDP 的流量

Ethernet II	167.510 MB	261		
IP	167.509 MB	261		
UDP	98.093 MB	165		
Other	89.132 MB	151		
MSN	5.827 MB	7		
RTP	2.199 MB	3		
Audio	730.247 KB	1		
G.729	144.171 KB			
G.728	121.680 KB			
10:59:21.948383	13.1.12.1.2.4:1704	12.1.13.5:35244	UDP	1,098
10:59:21.948392	13.1.12.1.2.4:1704	12.1.13.5:35244	UDP	1,098
10:59:21.948406	1.5.6.88.205:81	12.1.125.2:80	UDP	119
10:59:21.948413	1.2.1.13.65:15000	5.240.17.18.14980	UDP	94
10:59:21.948416	12.1.22.10.55876	20.0.17.2.1.122.11850	UDP	1,107
10:59:21.948421	1.1.1.8.1.12.12587	18.16.6.244.12643	UDP	1,439
10:59:21.948429	1.1.1.19.132.12424	14.3.10.6.251.14153	UDP	1,483
10:59:21.948433	1.2.1.22.24.8604	18.4.186.186.21381	UDP	93
10:59:21.948437	1.2.1.17.48.8114	9.59.48.8.25558	UDP	93
10:59:21.948655	2.1.8.19.132.12424	14.3.10.6.251.14153	UDP	1,486
10:59:21.948661	1.5.2.8.30.35.11359	20.1.8.19.183.8183	UDP	1,486

打开可以看到，虽然这些主机用的端口号都很大，通过观察可以发现，这些端口号大部分都是随即的，观察会话数特别大的机器，可以确定是几台机器之间在传输大量的数据，通过 IP 地址我们找到这几台机器，原来是他们在看网络电视（最近有部电视剧很火）。

- 4) 管理员让他们关掉网络电视以后网络的整体流量没那么高，但是访问外网时还是不怎么流畅，

接着分析，在诊断中我们发现很不少的ARP扫描（定位一台机器），打开如下视图：

绝对时间	源	目标	协议	大小	解	摘要
14:21:13.098717	00:21:5C:64:EF:E9	B0:48:7A:73:7B:EE	ARP	46		谁是 192.168.1.1? 告诉 192.168.1.101
14:21:13.099594	B0:48:7A:73:7B:EE	00:21:5C:64:EF:E9	ARP	46		192.168.1.1 在 B0:48:7A:73:7B:EE
14:21:14.618759	90:FB:A6:42:24:68	FF:FF:FF:FF:FF:FF	ARP	64		谁是 192.168.1.70? 告诉 192.168.1.122
14:21:58.099308	00:21:5C:64:EF:E9	B0:48:7A:73:7B:EE	ARP	46		谁是 192.168.1.1? 告诉 192.168.1.101
14:21:58.100490	B0:48:7A:73:7B:EE	00:21:5C:64:EF:E9	ARP	46		192.168.1.1 在 B0:48:7A:73:7B:EE
14:22:25.102895	00:21:5C:64:EF:E9	B0:48:7A:73:7B:EE	ARP	46		谁是 192.168.1.1? 告诉 192.168.1.101
14:22:25.103518	B0:48:7A:73:7B:EE	00:21:5C:64:EF:E9	ARP	46		192.168.1.1 在 B0:48:7A:73:7B:EE
14:22:41.116133	B0:48:7A:73:7B:EE	00:21:5C:64:EF:E9	ARP	46		谁是 192.168.1.101? 告诉 192.168.1.1
14:22:41.116172	00:21:5C:64:EF:E9	B0:48:7A:73:7B:EE	ARP	46		192.168.1.101 在 00:21:5C:64:EF:E9
14:23:14.500558	00:21:5C:64:EF:E9	B0:48:7A:73:7B:EE	ARP	46		谁是 192.168.1.12 生活 192.168.1.101

从上图我们看到 1.101 不停地寻找网关，而回应的机器并不是真正的网关，我们发现这个现象在局域网很多，我们通过下载专门的查杀 ARP 病毒的工具，确实发现几台可疑的机器，但是用杀毒软件并不能查杀出来，无奈之下管理员将这几台机器关机，果然网络中的 ARP 扫描少了很多，开开机器这种现象有出现了，最后不得没办法格机重装系统。

10.3. 测试总结

1. 通过用科来抓包发现了很多潜在的问题，如未知的ARP扫描。如果单纯用杀毒软件查杀我们的网络可能会觉得我们的网络很正常。
2. 在排除问题的时候我们最好能将一些下载，网络电视应用关掉，这些大流量的应用往往会造成一些排除问题的故障。

第三章 网络应用分析

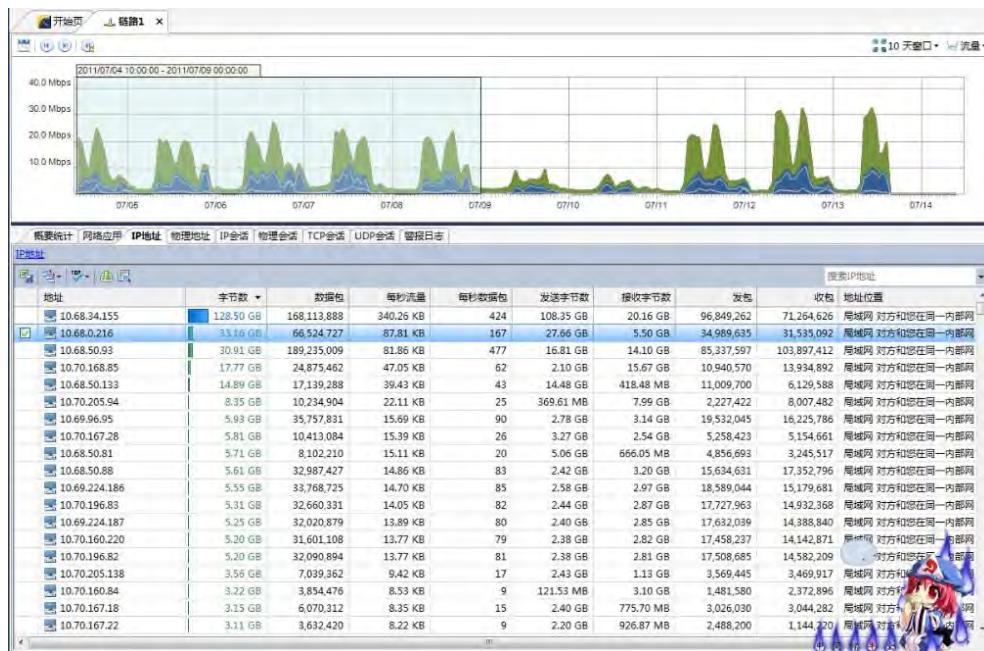
1. 某单位财务系统与OA系统评估

1.1. 故障描述

大家好，这次笔者去武汉出差，带来了最新的科来网络回溯分析系统的应用案例，与大家共同分享。

该单位目前出现的问题如下：用户经常反映，财务系统时常会出现响应慢的情况，最长的时间要等到半分钟左右，管理人员苦于没有好的分析工具，正好本次测试了科来网络回溯分析系统，于是对财务系统和OA系统进行了一次摸底评估。

测试的链路为各分处与总部服务器群连接的链路上的交换机处做了镜像。



简单介绍一下科来网络回溯分析系统，比起先前的软件，这套硬件设备功能可谓相当强大，本次使用的是 4T 的存储，千兆的抓包网卡。相对于原来的软件，硬件在使用上，多了一些数据挖掘的步骤，通过对 IP 地址、会话信息、网络应用（可自行添加）等条件，进行逐步挖掘，最后进行数据包级别的分析。

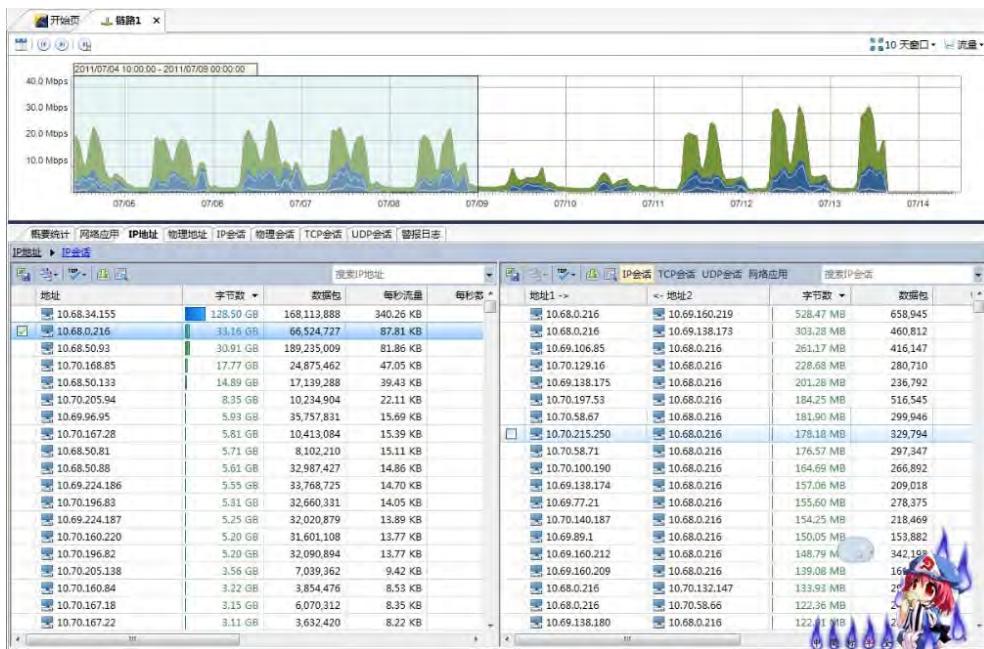
上图中是对该单位一周流量情况的分析，其中 34.155 即是 OA 系统的服务器，0.216 就是财务系统的服务器。

首先看一下财务系统，看看究竟是什么原因导致了财务系统响应慢的问题。

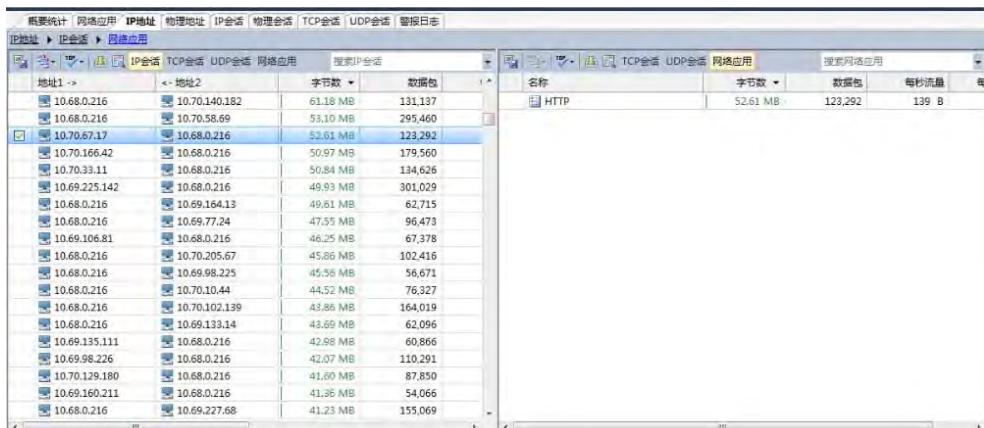
1.2. 分析情况

1. 通信对分析

通过数据挖掘的功能，我们挖掘与财务系统 10.68.0.216 进行通信的主机如下图所示：



再通过数据挖掘，对其中某一通信对的网络应用进行挖掘，发现是 HTTP 的应用，属于正常的业务应用。



2. 财务系统网络状况分析

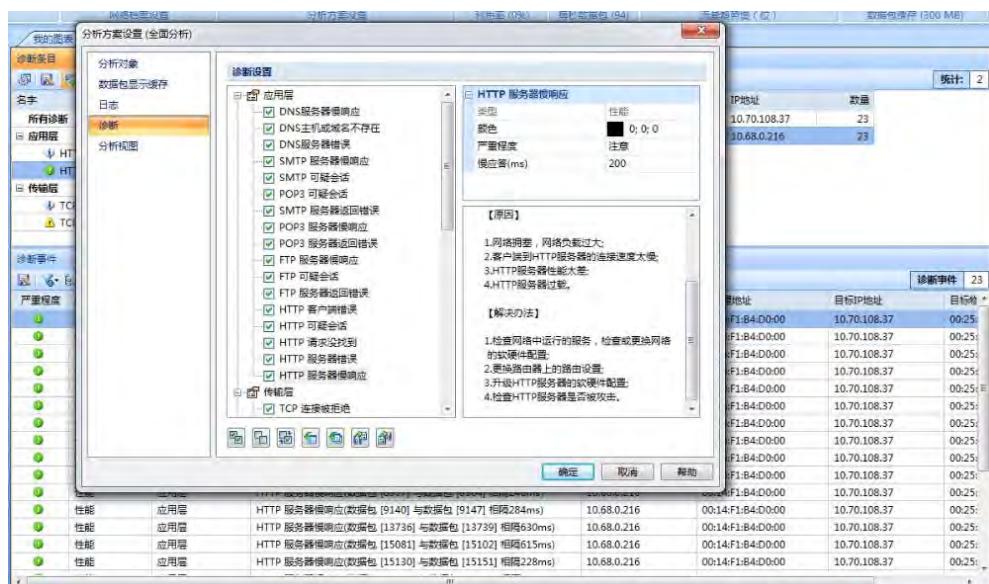
在应用交易中，判断网络状况的方法包含应用交易的三次握手时延和应用交易过程中的丢包率：



我们发现，在该通信对中，应用交易的时延基本在 5-10 毫秒之间，在诊断中也没有出现丢包的情况，可以断定，应用交易中的网络状况十分良好。

3. 专家诊断

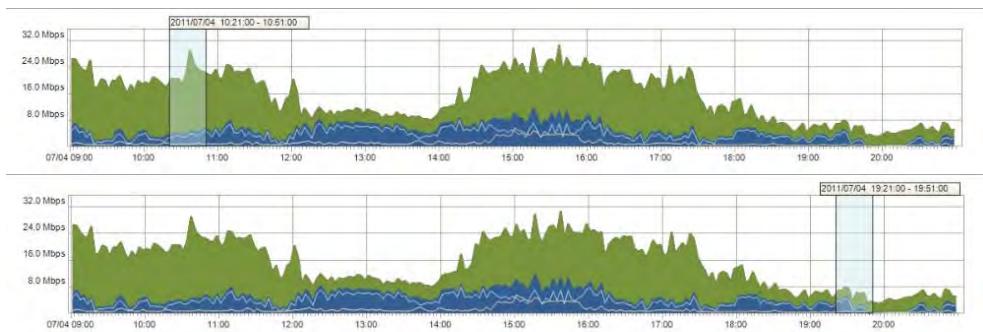
通常影响整个应用正常运行的情况有 2 中，一是网络状况，二就是应用系统服务器的状况，如何去判断服务器响应问题？科来网络分析系统提供了强大的专家诊断功能，轻松快速定位到相应的服务器响应问题，并告诉管理人员出现问题的原因和解决办法：



在上图的诊断中可以看到，整个应用交易中，有很多服务器慢相应的问题存在，可以初步判定，整个财务系统的响应慢，是由于服务器的原因造成的。

4. 繁忙与空闲时段的分析

然后我们分别在繁忙时段和空闲时段对采集的数据包进行了回溯分析：



分别在繁忙时段和空闲时段进行了网络状况的分析：

在繁忙时段的时延：

相对时间		摘要	10.69.77.23: 56423	10.68.0.216: 80	相对时间	摘要
00:00:00.0...	Seq = 0, Next Seq = 1	Window = 8192	SYN	Window = 65536	Seq = 0, Ack = 0, Next Seq = 1	
0.000078			SYN, ACK			
0.006180	Seq = 1, Ack = 0, Next Seq = 1	Window = 64240	ACK			

在空闲时段的时延：

相对时间	摘要	10.69.160.219: 2263	10.68.0.216: 80	相对时间	摘要
00:00:00.0...	Seq = 0, Next Seq = 1	Window = 65535	SYN	Window = 65535	Seq = 0, Ack = 0, Next Seq = 1
0.000125			SYN, ACK		
0.006576	Seq = 1, Ack = 0, Next Seq = 1	Window = 65535	ACK		

可以发现，时延无论在繁忙时段还是空闲时段，基本都保持在 5-10 毫秒之间。所以可以证明，网络中并不存在问题。

再来看一下服务器响应的问题：

在繁忙时段：

应用层		5
HTTP 可疑会话		1
HTTP 服务器慢响应		4

空闲时段：

应用层		3
HTTP 服务器慢响应		3

服务器慢响应却是始终存在的。由此可以断定，财务系统响应慢的问题，肯定在服务器上。推荐更换服务器硬件或更新财务系统程序。

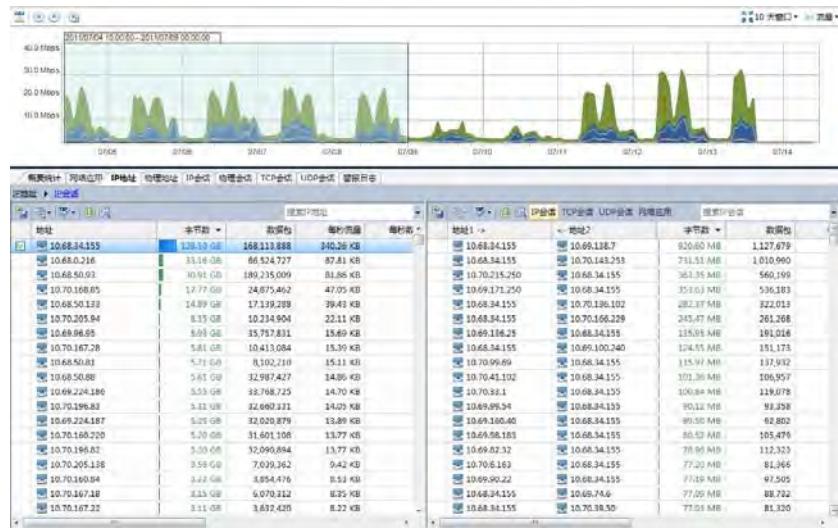
5. 小结

对财务系统的分析，主要还是通过科来网络回溯分析系统进行数据的挖掘。然后经由诊断来判断大概问题的原因。可能影响财务系统的原因有两点，一是网络问题，二是应用系统的问题。通过时延和丢包率证明了网络没有问题之后，在诊断中发现了服务器响应慢的问题。为了进一步证明是服务器响应慢造成的问题，通过在繁忙时段和空闲时段分别回溯分析，来证明确实是由于服务器的问题导致了整个财务系统的问题，建议对服务器进行硬件、软件的升级。

在分析完了财务系统之后，我们对 OA 系统也来进行一次分析：

6. 通信对分析

同样我们通过数据挖掘的功能，在该周内对访问 34.155 的主机进行了挖掘：



7. 时延分析

在随机抽取了其中的一些通信对进行分析后，我们通过 TCP 会话分析可以看到整个 OA 系统的网络延迟通常都在 100 毫秒以上，比起之前财务系统，明显要大了很多，可能是由于带宽不够等情况造成的：



8. 专家诊断

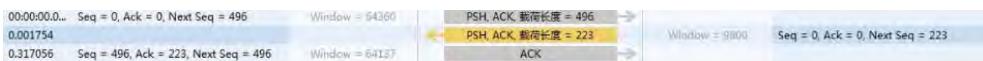
下图中可以看到，在诊断中，存在着 10 次重传，也就是 10 次丢包，并且有 2500 多次 TCP 慢应答，几乎占了整个通信的 1/10，网络中存在的明显的拥塞，可能是由于网络带宽不够造成的。HTTP 服务器慢响应也有 48 次，相应的时间基本在 200 毫秒左右，而不像之前的财务系统，相应时间最长的达到了 19 秒！所以对于用户来说，可能只是感觉稍稍有些慢，而不会觉得是不是系统出现了问题。



9. 繁忙与空闲时段的分析

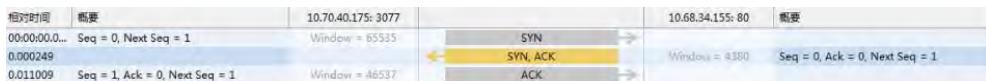
为了进一步证明网络中存在瓶颈，可能宽带不足引起的时延，我们依然通过繁忙时段与空闲时段的对比来进行分析：

在繁忙时段的时延：



已经达到了 300 多毫秒！是比较严重的延迟了。

在空闲时段的时延：



只有 11 毫秒。与财务系统不同，财务系统的链路无论在繁忙还是空闲时段，时延都在 5-10 毫秒，而 OA 系统在不同时段，时延相差了好几十倍！

再来看一下繁忙时段的诊断信息：



空闲时段的诊断信息：



通过对比，很明显就能发现，确实是网络存在着一定的问题，但是对于用户的体验来说，只是稍微感觉有一些延迟，不影响操作。

1.3. 小结

与财务系统类似，我们依然通过时延、丢包与服务器响应的判断，确定了在 OA 系统中，网络存在着瓶颈，建议管理人员对带宽进行一次升级，以降低网络时延和丢包率。

2. IDC出口流量梳理

2.1. 故障背景

某单位有一套独立的生产系统，服务器集中在 IDC 机房，各分支部门通过专线访问生产业务。IDC 所有服务器一直以来都采用相同的安全策略，为了提高网络的安全性，领导决定重新划分网络安全区域。

在网络安全区域划分之前，需要对 IDC 出口流量进行梳理，调查清楚每个应用对应的服务器 IP、服务端口号及用户源头等各种信息，为防火墙策略提供依据。这些信息虽然可以通过应用部门获取，但网络部门依然需要在实际网络流量中验证这些信息是否正确、完整，避免错误的安全规则造成生产业务访问失败。

一开始，网络部同事通过各种便携式的数据包分析工具，在 IDC 出口捕获数据包，手工分析服务器 IP、端口号及用户源头等信息，在耗费了大量的精力后，工作进度却非常的缓慢。用便携式的数据包分析工具在大流量环境中进行流量梳理，显得相当困难：

流量大，每次都能分析短时间的数据，而流量梳理至少需要一个月的周期才有说服力

效率低，手动分析，一次性只能分析一个应用，而 IDC 有上百种不同业务

信息量大，每个业务系统都存在大量的通信信息，人工处理相当困难，而且容易遗漏

这时候，我们就发现“科来网络回溯分析系统”的价值，回溯分析系统是专用的流量分析硬件设备，不仅仅能抓包，还能统计、存储网络中各种关键指标，并提供快速检索：

7*24 小时不间断监控，高性能数据采集

提供海量存储空间，记录每个时间点数据包、数据流和网络会话等信息，能够保存几天、几周甚至更长时间的数据

快速的数据检索能力，能够对过去任意时间点网络中发生的时间进行快速的回溯分析

内置智能专家分析系统，对网络异常行为可进行深入分析

这些功能，可以帮助网络部门的同事高效、准确的进行流量梳理，大大节省管理人员的精力。

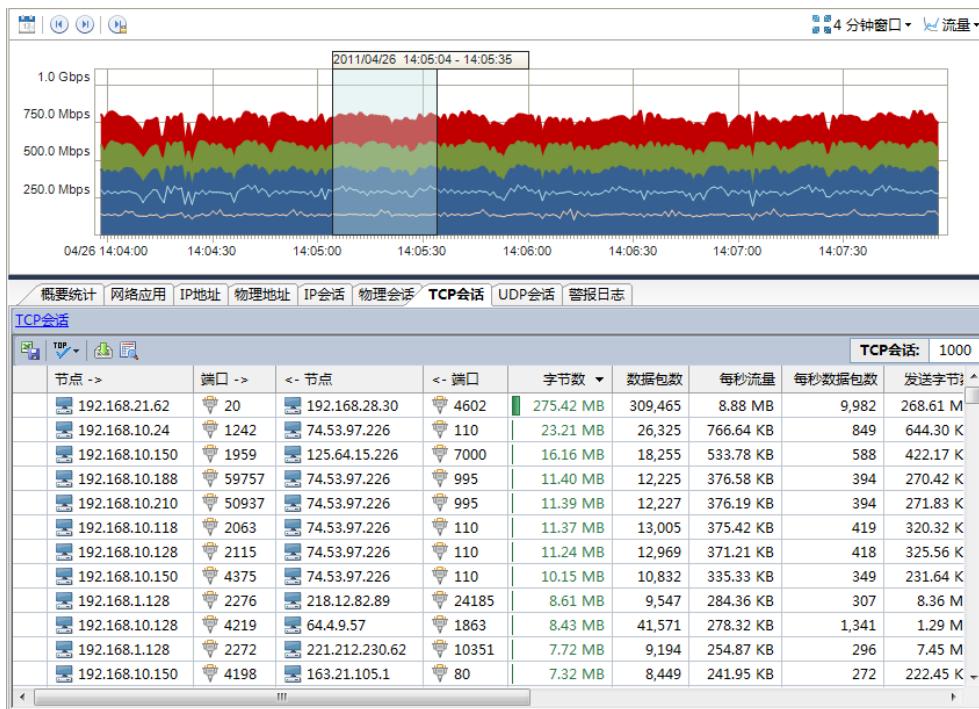
2.2. IDC出口流量梳理

1. 设备部署

在 IDC 出口设备上做镜像，将镜像流量接到科来网络回溯分析系统，简单易用。

2. 快捷历史数据回溯

部署科来网络回溯分析系统后，可以分析任意时间段的流量概要统计、网络应用、IP 地址、物理地址、IP 会话、物理会话、TCP 会话、UDP 会话等信息，并且可以逐层追溯分析、设置告警：



3. 流量信息验证

网络部门同事已经从应用部门获取应用服务器 IP 地址等信息，那么我们如何通过科来网络分析系统进行确认呢？

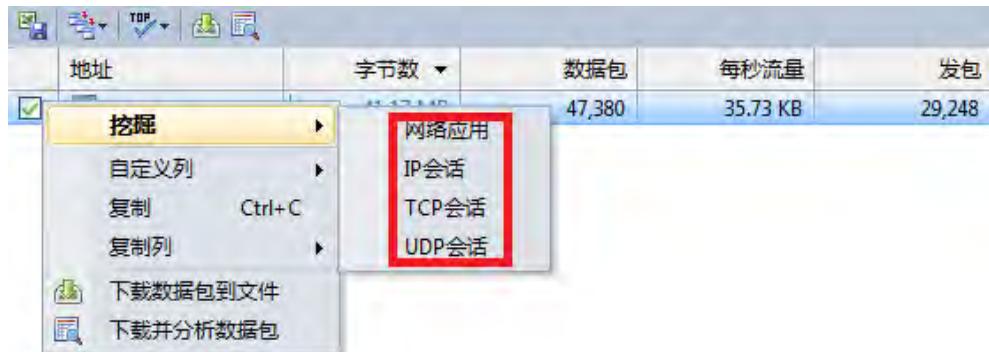
我们可以一次性查看一天、一周甚至更长时间的“IP 地址”信息，就可以获取网络中所有 IP 地址的流量信息，包括通信流量大小、数据包量、进出流量、流量收发比、发送 TCP 同步包数量、接收 TCP 同步包数量等各种信息，通过这些信息我们可以快速的判断每个 IP 地址是否为服务器、负载大小等信息：



在所有的 IP 地址中，我们可以通过“地址检索”功能快速定位到我们想找的地址，假如服务器“*.*.10”通过 TCP443 端口提供了“财务”系统的业务，我们可以在地址检索中输入“*.*.10”这个地址，就可以获取该服务器的流量信息：



从上图可以看到，地址检索之后，“IP 地址”列表中就只剩下我们想要的那台服务器地址的信息，同时我们看到该服务器的“收到 TCP 同步包”、“发 TCP 同步确认包”这两列的数量均为 9，这就证明该 IP 地址为服务器，在这段时间内收到 9 次连接请求，并且都成功响应了。接下来，我们还可以通过数据“挖掘”功能，获取该服务器更多的信息：



比如，挖掘该服务器的“TCP 会话”：

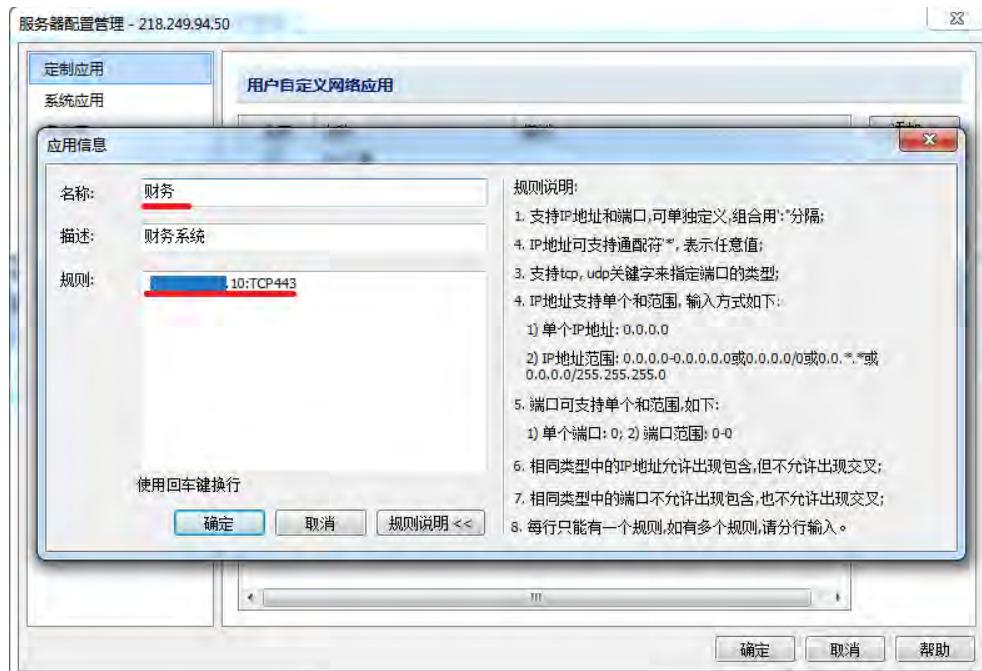


由上图可见，服务器“*.**.10”提供服务的端口号是“TCP443”，而且只有“TCP443”端口，于是，我们可以很轻松的判断，应用部门提供的信息：“财务系统：*.**.10: TCP443”是正确的。

4. 业务主动监控

通过以上的步骤，我们能够轻松的检验业务系统的各种信息，我们能不能在科来网络回溯分析系统中更智能的记录我们验证过的信息，更主动的对业务系统进行监控，甚至主动的发现异常流量或者新上线的业务流量？

答案是肯定的！科来网络分析系统提供“定制应用”功能，可以根据服务器 IP 地址、IP 地址段、TCP/UDP 端口号、TCP/UDP 端口端及 IP 地址与端口号结合等各种方式进行自定义业务的定制。比如，我们可以将“*.**.10: TCP443”定义为“财务”：



那么,我们在“网络应用”这里,就可以看到“财务”系统的流量信息:

名称	字节数	数据包	每秒流量	每秒数据包
主页	678.13 KB	1,337	581 B	1
知识库	22.03 KB	192	18 B	0
财务	57.31 MB	64,548	49.11 KB	54
ACENT	76 B	1	0 B	0
AIM	206 B	3	0 B	0
BOOTP	8.11 KB	24	6 B	0
BitTorrent	317.50 KB	1,169	272 B	0
CIFS	290.31 KB	4,178	248 B	3
Cisco-fna	188 B	2	0 B	0
Citrix ICA	398 B	5	0 B	0
CTF	29.23 MB	35,866	25.05 KB	30
DNS	9.56 MB	70,065	8.19 KB	58
Daytime	188 B	2	0 B	0

“网络应用”这里同样支持应用检索、回溯分析功能,可以方便、快捷、主动的对业务系统进行分析、监控:

地址1 ->	端口1 ->	<- 地址2	<- 端口2	字节数
10	61773	10	443	6.41 MB
10	4520	10	443	5.71 MB
10	54105	10	443	5.50 MB
10	59990	10	443	5.42 MB
10	57555	10	443	5.22 MB
10	47234	10	443	5.14 MB
10	38964	10	443	5.11 MB
10	53592	10	443	5.09 MB
10	60476	10	443	4.41 MB
10	35056	10	443	3.67 MB
10	62404	10	443	3.10 MB
10	58191	10	443	2.39 MB
10	32304	10	443	139.23 KB

如果我们将 100 多种业务系统都定制在科来网络回溯分析系统中，那么当“网络应用”中出现任何其他应用，那不是异常流量就是新上线的业务系统，这是不是一劳永逸的监控方式呢？

2.3. 小结

以上的所有操作，都无需涉及到手动数据包解码，相当轻松。由此可见，科来网络回溯分析系统能极大的提高网管人员的工作效率，在节省大量的人力、精力的同时，还提供更准确、更主动的网络监控方式。

测试申请

如果您有关于《科来网络回溯分析系统》以及《科来网络分析系统》产品的测试需求，请您拨打
400-6869-069 我们将尽快为您安排相关测试工作。相关产品信息请参见：<http://colasoft.com.cn/products/>

案例征集

科来软件将不断为广大网络管理人员及技术爱好者提供更有价值与借鉴意义的分析案例，同时也希望大家踊跃投稿，共同参与网络分析技术的研究与讨论，为提升国内的网络分析技术水平而共同努力。如果您有好的案例可以通过在 CSNA 论坛（www.csna.cn）进行分享，或者将案例直接发信到 support@colasoft.com.cn 邮箱。

欢迎加入 CSNA 网络分析论坛

CSNA 论坛是国内最大最专业的网络分析技术论坛，致力于为国内用户提供优秀的案例、前沿的网络分析技术知识以及提供网络问题的研究与讨论的平台。截止到 2012 年初共有 18 万的用户通过该平台接触和学习网络分析技术。欢迎加入到 CSNA 网络分析论坛！论坛网址：<http://www.csna.cn/>

CSNA 网络分析认证培训

网络分析技术的推动者

科来积极推动网络分析技术的发展，在官网推出免费版本的软件供广大技术爱好者交流使用，同时有针对性的提供大量技术学习文档及视频资料，并组织力量在相关技术论坛解答用户的疑问及分享资源，旨在提高国人在网络分析领域的技术水平，帮助用户透过网络现象看本质，从而真正的驾驭自己的网络。

除了上面所述，科来公司还通过 CSNA 网络分析认证培训让更多的用户掌握网络分析技术，提高对网络问题的解决能力，促进网络管理水平。科来在国内外大量客户的网络分析案例中，积累了丰富的经验，使得科来网络分析技术培训有很强的实用性及针对性。

CSNA网络分析认证培训



▣ CSNA 网络分析师认证培训

通过培训，能够掌握基本的网络分析知识，能够熟练的使用科来网络分析系统，能够熟练的利用网络分析系统对网络进行日常分析，定位常见问题。

▣ CSNA 网络分析与家认证培训

通过培训，能够掌握高级的网络分析知识，能够根据网络和应用故障现象独立的制定分析方案，定位网络或应用问题点，分析网络和应用问题。

▣ CSNA 安全分析认证培训

通过培训，能够掌握各种威胁网络安全的蠕虫、攻击行为、木马的工作原理、网络行为模式特点以及分析方法。

科来软件官网 www.colasoft.com.cn
网络分析论坛 www.csna.cn
意见与建议反馈 support@colasoft.com.cn

拓展网络视野 精细网络管理