NAT资料收集

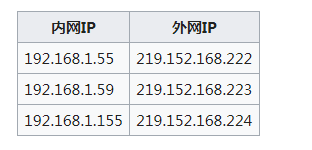
NAT(Network Address Translation),是一种对IP数据报的源IP或者目的IP进行重写的技术，该技术是由IETF组织制定的一定标准，用于暂时解决公网Ipv4地址资源匮乏问题的过渡技术。（可以解释下为什么Ipv4 公网地址紧缺：量少,分布不均匀）

NAT的方式：

1. Basic NAT

NAT服务器会维护一张内网IP与外网IP映射关系的表，将内网IP映射到特定的外网IP,该方式不会修改端口号。

映射表类似于：



因为修改了IP地址，所以IP数据包的校验和字段需要修改。

过程：

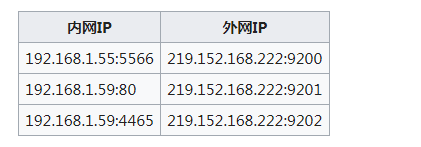
192.168.1.55往114.114.114.114主机发送一个IP数据包，这个IP数据包的源IP为192.168.1.55，目的IP为114.114.114.114 。经过NAT服务器的时候，NAT服务器通过查表，把这个IP报的目的IP改成219.152.168.222,然后发出去。

114.114.114.114收到IP数据包后，就会有一个回复数据包。这个回复包的目的IP是不是219.152.168.222。当这个包到达NAT服务器时，它通过查表发现这个包其实是发给192.168.1.55，于是将目的IP改成192.168.1.55。

根据这张映射关系表是否会变化，又可以将NAT分为静态和动态之分。

1. NAPT 网络地址端口转换

NAT服务器会维护一张内网IP/端口和外网IP/端口之间的映射关系表。该表类似于：

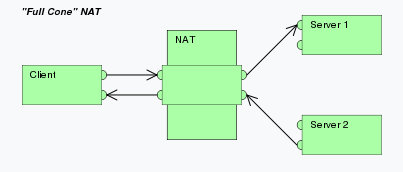


NAPT分两类：源地址转换和目的地址转换。即一种是修改源地址，那一种是修改目的地址，分别对应由内网经NAT网关到外网以及由外网经NAT网关到内网。

其中NAPT又分以下几种工作方式：

1. 完全圆锥型NAT,即一对一NAT

特点：一旦一个内部地址（iAddr:iport）映射到外部地址(eAddr:eport),所有发自iAddr:iport的包都经由eAddr:eport向外发送。任意外部主机都能通过访问eAddr:port来访问iAddr:port。

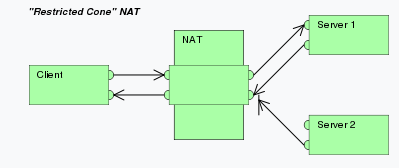


1. 受限圆锥型NAT

特点：

内部地址（iaddr:iport）映射到外部地址(eAddr:eport),所有发自iAddr:iport的包都经由eAddr:eport向外发送;外部主机hostAddr能够通过eAddr:eport访问iAddr:iport当且仅当内部地址先向该外部主机hostAddr发送过数据包。

受限的含义：外部主机是受限的，并非所有外部主机都可以通过iAddr:iport访问到内部服务。



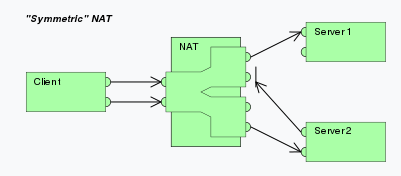
1. 端口受限圆锥型NAT

特点：内部地址（iaddr:iport）映射到外部地址(eAddr:eport),所有发自iAddr:iport的包都经由eAddr:eport向外发送;只有iaddr:iport向特定外部主机(hostAddr:hostport)发送过数据，（hostAddr:hostport）才能通过eaddr:eport访问iaddr:iport.

受限的含义：受限圆锥型NAT的基础上加上对端口的限制。

1. 对称型NAT

特点：每一个来自相同内部IP与端口，到一个特定目的地地址和端口的请求，都映射到一个独特的外部IP地址和端口。同一内部IP与端口发到不同的目的地和端口的信息包，都使用不同的映射



对称型与圆锥型NAT的区别：

圆锥型：iaddr:iport经过NAT后映射到固定的eaddr:eport,之后所有的从iaddr:iport出来的数据包都会映射到eaddr:eport

对称型：iaddr:iport经过NAT后映射到不同的eaddr:eport

举例：

假设：

NAT内的主机A: IP为A,使用端口1000

NAT网关： 对外IP为NAT,NAT的端口池为(5001-5999)

公网的主机B:IP为B,使用端口2000

公网的主机C:IP为C,使用端口3000

假设A先访问了B：2000，但是没有访问C

完全圆锥型NAT:

任意地址发往 NAT （ 5001 ）的数据包都可以到达 A （ 1000 ）

==========================================================

B （ 2000 ） —— > NAT （ 5001 ） ——— > A （ 1000 ）

B （ 3000 ） —— > NAT （ 5001 ） ——— > A （ 1000 ）

C （ 3000 ） —— > NAT （ 5001 ） ——— > A （ 1000 ）

受限圆锥型NAT:

只要是从 B 主机发往 NAT （ 5001 ）的数据包都可以到达 A （ 1000 ）

==========================================================

B （ 2000 ） —— > NAT （ 5001 ） ——— > A （ 1000 ）

B （ 3000 ） —— > NAT （ 5001 ） ——— > A （ 1000 ）

C （ 2000 ） —— > NAT （ 5001 ） — X — > A （ 1000 ）

端口受限：

只有 B （ 2000 ）发往 NAT （ 5001 ）的数据包可以到达 A （ 1000 ）

===========================================================

B （ 2000 ） —— > NAT （ 5001 ） ——— > A （ 1000 ）

B （ 3000 ） —— > NAT （ 5001 ） — X — > A （ 1000 ）

C （ 2000 ） —— > NAT （ 5001 ） — X — > A （ 1000 ）

对称型NAT:

假设A先访问B,再访问C,则：

可能：

A （ 1000 ） —— > NAT （ 5001 ）—— > B （ 2000 ）

A （ 1000 ） —— > NAT （ 5002 ）—— > C （ 3000 ）

而如果是锥型：

A （ 1000 ） —— > NAT （ 5001 ）—— > B （ 2000 ）

A （ 1000 ） —— > NAT （ 5001 ）—— > C （ 3000 ）

NAT的缺陷：

1. 破坏IP对等通信的原则;IP对等通信是指,所有IP地位一致。而NAT的加入会使在未建立映射关系前，内网可以访问外网与外网无法对等的访问内网。
2. NAT使IP会话的保持时效变短。这主要是因为NAT需要维护一个内外地址的映射关系表。而这个地址表不应该是一成不变的，而应该有自动更新、清理的功能。于是NAT需要某种策略来清理一些过时的映射关系。由于IP协议的不可靠性，不同主机对整个网络的感知是不同的，于是NAT很难采取一种特别完美的策略来清理映射关系。对于TCP协议这种面向连接的协议，NAT可以跟踪链接的状态。但是对于UDP协议来说，何时清理“过时”的NAT就比较难了。
3. NAT将网络环境变的复杂，使得依赖IP进行主机跟踪的机制失效。
4. NAT会修改IP数据报的字段，这会妨碍一些安全协议，例如IPSec协议等。

NAT的穿透：

使用STUN+TRUN协议穿透,使用Upnp打浻

也有使用构造好的ICMP进行打洞。

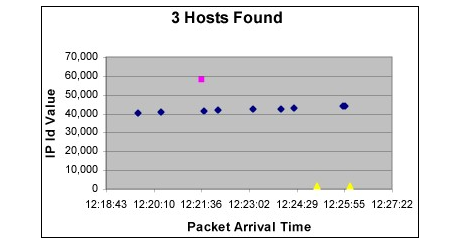
NAT的检测：

STUN用于主动检测当前主机所处的网络环境。

Martin Grill [1]提出使用HTTP 访问记录，使用SVM分类器进行被动检测。统计一定时间段内的8个属性：不同IP地址数;不同User-Agent数目;不同的OS数目;持续的流数目;上传字节数;下载字节数;http请求数。生成数据集的方法，先收集明确是非nat主机的访问记录;然后将不同端的访问记录组合在一起，人工构建nat数据;这是因为若干个主机将经nat后，就相当于把这8个向量整合在一起。

Rui[3]提出一种使用SVM进行检测NAT流量的方法。他生成数据集的方式：1台主机不NAT.2～5台主机nat。使用的特征向量有：出包数量;入包数量;UDP报文数量;TCP报文数量;DNS查询数量;FIN数量;RST数量;SYN数量。为了过滤掉一些在线率不高的主机,通过某种方式把休眠的主机过滤了。

Bellovin 提出通过观察IP 报文中的ID字段来发现NAT设备后面的主机数量.[3]这种方法的实现是基于当主机每发现一个IP数据包则IP ID增加一定数值，且每个主机IP ID的初值不同。如果在NAT后面存在多个主机则这些IP ID的值将会发离开来。可以发现三条不同的直线，因此有三个主机。但是这种方式不适用于BSD系统，因为BSD系统有IP ID是随机的。



Beverly 提出使用TCP/IP报文里面的TTL ,DF,窗口大小以及SYN初始值来区分NAT和主机的方法。[4] 其实就是基于OS的协议栈指纹，来统计有多少不同的指纹。

Kohno提出使用TCP的时间戳字段来当作OS的指纹。[5]

白雪提出基于统计一定时间段内访问同一网站Cookie ID数量的方式来判断NAT以及NAT后面的主机数量。还有提出统计一段时间内IM数据包内的QQ号码数量来判断是不是NAT,以及NAT后面有多少主机。

上面两种方法的缺点在于如果NAT后的系统是一致的，则出现错误的判断。而且是依赖于特定的报文字段，这些字段有很大的局限性。

Sophon Mongkolluskamee [6] 提出通过观察IPID,TCP 序列号,TCP 源端口的方式来计算Nated后的主机数目。他将这三者关联起来，可以识别Windows,linux,freeBSD,mac os ,openbsd系统以及对应的数量。他的实验方法是通过nat流量合成，实际办公室ap抓包，以及把这个用在一个公开流量数据集中计算2001-2010每年的nat比例和nated规模。

基于流量特征的：

与传统单主机的流量相比，经过NAT后的主机流量应该是具有以下特征：更多byte,更多连接;更多DNS查询;更多协议类型等等。

想法：使用流量特征来快速发现NAT流量，然后使用IP ID,TCP sequence Number,Source Port 来识别有多少个主机，有哪些主机;

如何采集数据：前期，使用虚拟机构建网络，6台NAT模式的下的主机（3台Windows 7，2台linux,1台 Mac OS）5台没有nat的，让他们运行脚本访问网络资源（注意随机）。主要用来得到Windows,Linux,Mac OS的IP ID,TCP sequence,Source Port的模式，然后基于这些数据构造数据集，训练分类器（SVM，CNN）;在实验室抓取数据包，抓无线网络的，主要验证（需要组里提供设备）;在大的数据集里面计算一波，计算nat比例，nat后的主机数比例，主机类型比例;

[1] Tom´aˇs Kom´arek, Martin Grill, Tom´aˇs Pevn´y, Passive NAT Detection Using HTTP Access Logs, Information Forensics and Security (WIFS).

[2] L. Rui, Z. Hongliang, X. Yang, X. Yang, and W. Cong, “Remote nat detect algorithm based on support vector machine,” in Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on. IEEE, 2009, pp. 1–4

[3] S. M. Bellovin, "A technique for counting NATted hosts" in , New York, USA:, 2002.

[4] R. Beverly, "A robust classifier for passive TCP/IP fingerprinting", Passive and Active Measurement Workshop:5th international workshop, pp. 158-167, 2004.

[5] T. Kohno, A. Broido, K. C. Claffy, "Remote physical device fingerprinting", IEEE Transactions on Dependable and Secure Computing, vol. 2, pp. 2, 2005.

[6]Sophon Mongkolluksmaee,Kensuke Fukuda,Panita Pongpaibool.”Counting Nated Hosts by Observing TCP/IP Field Behaviors.

[7] 高骥翔.基于网络流量特征的NAT识别方法.电子科技大学硕士学位论文.2012