一种 NTP 协议隐蔽通道①

朱越凡[1,2], 马 迪[3], 王 伟[1,3], 毛 伟[1,3]

[1](中国科学院大学, 北京 100049)
[2](中国科学院 计算机网络信息中心, 北京 100190)
[3](北龙中网(北京)科技有限责任公司, 北京 100190)

摘 要: 网络隐蔽通道技术是一种被广泛应用的网络攻击技术. 掌握隐蔽通道的构建机制, 对制定相应网络防御策略具有指导意义. 利用互联网不可或缺的 NTP 时间同步协议, 提出了基于 NTP 协议的隐蔽通道构建机制. 通过分析 NTP 协议查询/应答机制的特点, 并研究可被用作载荷的 NTP 协议数据单元, 设计了下行通道和上行通道分离的 NTP 隐蔽通道, 它将隐藏信息伪装成普通 NTP 报文, 进行隐秘消息的传递. NTP 报文的普及性和不可替代性, 使得基于 NTP 的隐蔽通道具有穿透能力强、隐蔽性好的优点. 试验表明, 提出的 NTP 隐蔽通道可以携带较多的秘密信息, 穿透网络监测设备. 下一步的工作将围绕 NTP 隐蔽通道的认证、加密等安全机制进行研究.

关键词: 隐蔽通道; 网络安全; NTP 协议; 协议特点; 传输带宽

Covert Channel Based on NTP Protocol

ZHU Yue-Fan[1,2], MA Di[3], WANG Wei[1,3], MAO Wei[1,3]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)
[2](Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)
[3](Internet Domain Name System Beijing Engineering Research Center Ltd., Beijing 100190, China)

Abstract: The covert channel based on network protocol has been widely used for network attack. Mastering the mechanism of covert channel is important to the formulating of corresponding network defense strategy. Due to the widely use of network time protocol, a kind of covert channel technology based on NTP protocol is proposed in this paper. This paper analyzes features of query/response mechanisms in the NTP protocol, utilizes the message field as hidden payload and then designs separated downstream and upstream NTP covert channels, in which secret information is disguised and transmitted as normal NTP messages. The popularity and irreplaceability of NTP message make NTP covert channel endowed with the advantages of great ability of penetration and high quality of concealment. Test results show that the technology could carry relatively considerable information and easily penetrate the network monitoring device. The future work will focus on authentication, encryption and other security mechanisms in NTP covert channel.

Key words: covert channel; network security; NTP protocol; protocol features; transmission bandwidth

在信息传递过程中, 我们通常使用加密技术来对需要保密的数据进行处理. 加密技术使得信息对于第三方变得不可读. 但这并不能掩盖通信存在的事实, 并且很容易引起第三方的嗅探、窃听等被动攻击行为. 隐藏信道试图隐藏通信的存在. 隐蔽通道(Covert Channel, CC)是指允许进程以危害系统安全策略的方式传输信息的通信信道[1]. 最初它被提出是为了解决囚犯问题. 两个囚犯进行通信, 但他们之间的通信可以被狱警监听并决定是否允许. 因此囚犯必须设计一种方法来使它们的对话看起来无害却隐藏着秘密消息.

网络隐蔽通道技术是一种十分常见的网络攻击技术. 该技术将隐蔽消息伪装成普通消息在网络中进行通信. 防范隐蔽通道的最好办法就是掌握隐蔽通道建立的机制, 以便制定相应的网络安全防护策略, 来消除隐蔽通道给网络安全带来的隐患. 构造隐蔽通道的方法通常是通过修改数据包、利用协议机制的弱点或

① 收稿时间:2016-08-03;收到修改稿时间:2016-09-08 [doi:10.15888/j.cnki.csa.005715]

者利用协议的空闲字段来实现[2]. 例如, 张令通等利用TCP协议首部中的序列号和确认号字段建立了隐蔽通道[2]; 杨智丹等对 IP 报头指针字段选项的网络隐蔽通道技术进行了研究[3]; 罗成等利用 Windows 消息机制来控制拥有网络访问权限的应用程序, 进而构建网络隐蔽通道[4]; Iodine 利用 DNS 中 NULL 记录类型构造Raw UDP模式, 使得整个UDP载荷均为隐蔽通道[5]. 在 NTP 隐蔽信道的研究方面, Action Dan 利用 NTP 报文 Transmit Timestamp 字段作为隐蔽通道传递 forkbomb 指令[6], 实施隐蔽通道攻击; 但这个方法带宽有限, 不适用于传递消息.

NTP 是网络时间协议(Network Time Protocol), 是为实现高精确度的时间同步, 而设计的网络时钟同步协议. NTP 使用层次式时间分布模型, 具有相当高的灵活性, 可以适应各种互联网环境, 是互联网上公认的时间同步工具[7]. 目前, 基于 NTP 的隐蔽通道并没有引起应有的重视. 然而, 网络中大量存在的开放式分布的 NTP 服务器, 高度开放的 NTP 协议 123 端口, 以及 NTP 报文基于不可靠的 UDP 进行传输的协议特性, 使得 NTP 协议非常容易被黑客利用, 实施隐蔽通道攻击. 因此, 研究基于 NTP 协议的隐秘通信方法对网络安全防护具有重要的指导作用. 本文利用 NTP 查询/应答机制的协议特性, 使用NTP协议报文中易被忽略的数据单元字段, 提出一种高效的上下行分离的隐蔽通道方法, 以达到穿透防火墙和躲避入侵检测系统的目的.

1　构建隐蔽通道的模型和方法

隐蔽通道早期的定义只限于操作系统内部, 重点研究操作系统内的安全. 随着网络技术的发展, 隐蔽通道逐渐被应用到网络技术中. 隐蔽通道由此有了更加广泛的定义: 任何利用非正常的通信手段在网络中传递信息, 突破网络安全机制的通道都可以称为隐蔽通道[8].

1.1 隐蔽通道类型

根据隐藏隐蔽通道又分为存储型和时序型[8]. 存储型隐蔽通道通常利用协议字段的冗余嵌入隐蔽信息. 时序型隐蔽通道常利用一系列协议数据包的某种排序或相邻数据包到达的时间差传递秘密信息.

根据嵌入的位置, 协议隐藏可分为结构型和非结构型[9]. 结构型隐蔽通道指在协议的固定格式部分(协议首部)嵌入信息. 非结构型隐蔽通道在协议携带的数据部分嵌入信息.

因为时序型隐蔽通道易受网络中不稳定因素干扰, 导致数据包延迟或丢失, 不利于隐藏信息的传递, 因此本文 NTP 隐蔽通道将使用存储型隐蔽信道. 在嵌入位置方面, 本文对结构型和非结构型 NTP 隐蔽通道的效率和带宽等也做了相关研究和分析.

1.2 隐蔽通道构建方法

构造隐蔽通道的关键有三点[8]:

1) 寻找合适的协议. 通常网络应用层中隐蔽通道都是以 HTTP、SMTP、DNS 为载体; 这主要是因为这些协议对现在的企业网络来说是必须的.

2) 寻找协议中的漏洞. 即选择协议头部中的合适字段作为隐蔽通道的数据域. 虽然这些数据域的有效载荷有限, 但是 Lampson[7]指出即使一个只能一天传输一位的隐蔽通道在特定的环境下也能够带来危害.

3) 伪造要发送的数据. 通常对要发送的数据进行加密. 这样可以躲避 IDS 检测, 让隐蔽通道发挥更大的作用.

2　NTP协议介绍

2.1NTP 协议的工作原理

NTP 以客户机/服务器模式进行通信: 客户机发送一个请求数据包, 服务器接收后回送一个应答数据包[9]. 两个数据包都带有发送和接收的时间戳, 根据这四个时间戳来确定客户机和服务器之间的时间偏差和网络时延.

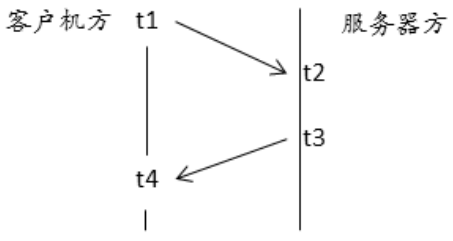

图 1　时间同步算法时序

如图 1 所示, t1 为客户机发送查询请求包的时刻, t2 为服务器收到查询请求包的时刻, t3 为服务器回复时间信息包的时刻, t4 为客户机收到时间信息包的时刻(t1、t4 以客户机的时间系统为参照, t2、t3 以服务器的时间系统为参照). 由此可得信息包在网络上的传输

时间为:

$$\Delta=(t4-t1)-(t3-t2)$$

当请求信息包和回复信息包在网上的传输时间相等时, 单程网络时延为:

$$\delta=((t2-t1)+(t4-t3))/2$$

时间偏差为:

$$\theta=((t2-t1)+(t4-t3))/2$$

可以看到, θ、δ 只与 t2 和 t1 的差值、t3 和 t4 的差值相关, 而与 t3 和 t2 的差值无关, 即最终的结果与服务器处理请求所需的时间无关. 据此, 客户机即可通过这 4 个时间戳计算出时间偏差 θ 和网络时延 δ 去调整本地时钟.

2.2 NTP 的报文格式

NTP 数据包的传输采用 UDP 协议. 在 UDP 消息头之后, 紧接着的数据格式如图 2 所示[10].

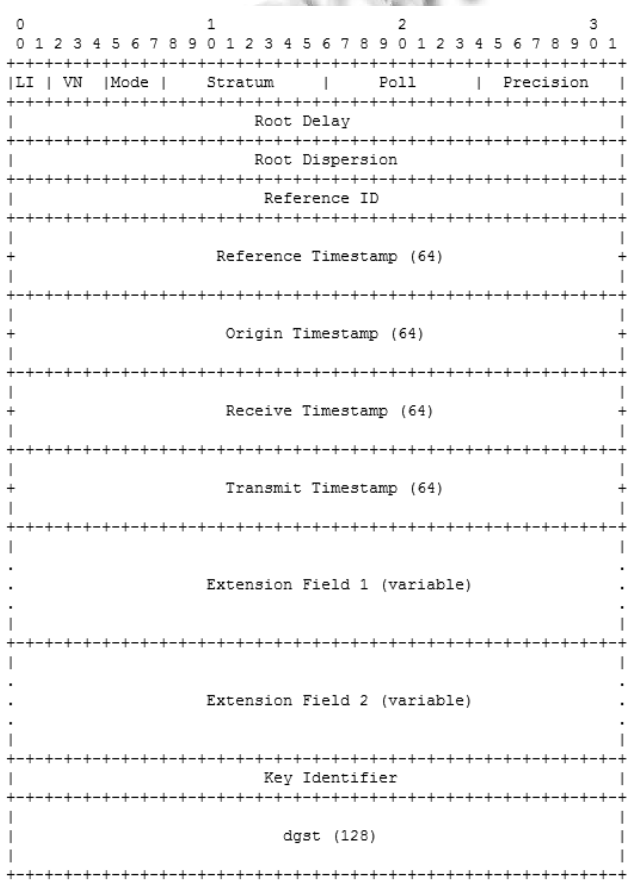

图 2 NTP 报文格式

主要字段的解释如下:

LI(Leap Indicator): 长度为 2 比特, 值为“11”时表示告警状态, 时钟未被同步. 为其他值时 NTP 本身不做处理.

VN(Version Number): 长度为 3 比特, 表示 NTP 的版本号, 目前的最新版本为 4.

Mode: 长度为 3 比特, 表示 NTP 的工作模式. 不同的值所表示的含义分别是: 0 未定义、1 表示主动对等体模式、2 表示被动对等体模式、3 表示客户模式、4 表示服务器模式、5 表示广播模式或组播模式、6 表示此报文为 NTP 控制报文、7 预留给内部使用.

Stratum: 系统时钟的层数, 取值范围为 1～16, 它定义了时钟的准确度. 层数为 1 的时钟准确度最高, 准确度从 1 到 16 依次递减, 层数为 16 的时钟处于未同步状态, 不能作为参考时钟.

Poll: 轮询时间, 即两个连续 NTP 报文之间的时间间隔.

Precision: 系统时钟的精度.

Root Delay: 本地到主参考时钟源的往返时间.

Root Dispersion: 系统时钟相对于主参考时钟的最大误差.

Reference Identifier: 参考时钟源的标识.

Reference Timestamp: 系统时钟最后一次被设定或更新的时间.

Originate Timestamp: NTP 请求报文离开发送端时发送端的本地时间.

Receive Timestamp: NTP 请求报文到达接收端时接收端的本地时间.

Transmit Timestamp: 应答报文离开应答者时应答者的本地时间.

Extension Field: 可选扩展字段. 最小填充长度为 16 个字节, 最大填充长度尚未被定义.

Key Identifier: 客户机和服务器双方协商的密钥(可选字段).

dgst: 用密钥计算得出的 NTP 头部和扩展字段的哈希值(可选字段).

3 基于NTP协议的高效隐蔽通道实现方法

3.1 NTP 隐蔽通道的基本原理

NTP 隐蔽通道技术的基本思想是利用 NTP 请求和应答建立隐蔽通道, 实现数据传输. 通信前双方商量好隐藏信息的字段和编码方式等规则, 发送方按照这个规则将隐秘信息伪装、编码、发送, 接收方按照约定接收、解码、提取隐秘信息. 简单的 NTP 隐蔽通道

的工作模式如图 3 所示.

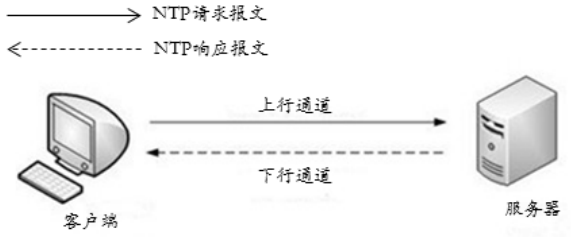

图 3　上行通道和下行通道分离的 NTP 隐蔽通道

普通的NTP隐蔽通道在通信过程中会产生大量来自同一主机的 NTP 请求包和返回该主机的 NTP 应答包, 流量分布特征十分显著, 很容易被入侵检测系统识别出异常[1]. 为了降低 NTP 隐蔽通道被检测的风险, 我们利用 NTP 报文基于面向无连接的 UDP 协议工作的特点, 如图 4 所示, 在客户端节点处伪造其他主机 IP 地址发送 NTP 时间同步请求, 最终服务器返回响应包到伪造 IP 主机处, 从而平衡流量分布.

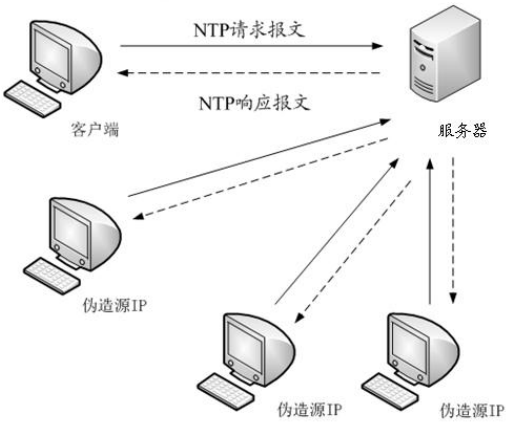

图 4　伪造源 IP 发送 NTP 同步请求报文

3.2 使用 NTP 协议构建隐蔽通道的详细方法

我们可以使用一种非常暴力的方法来构建NTP隐蔽通道: 将基于 UDP 的服务运行在 123 端口, 从客户端直接向服务器发起通信, 让整个 NTP 报文处于 Raw UDP 模式. 此时, 整个 UDP 载荷均为隐蔽通道, 通信效率显著提升. 然而, 这些报文不是有效的 NTP 消息, 流量分析工具解析这些报文时会出现格式错误, 从而引起怀疑. 因此, 我们采用了更加具有欺骗性的手段, 既能保证一定的通信带宽, 又使得携带隐蔽消息的报文符合基本的 NTP 协议包特征, 避免被网络上流量分析检测工具识别出来.

1) 基于报文数据单元构建上行通道

NTP 时间同步机制是由客户机直接发起时间同步请求, 因此在上行信道只需要考虑哪些字段可以隐藏数据, 而不用考虑信道的发起时间和通信频率. 根据网络防护设备的一般规则设置, 对于 NTP 协议包的校验主要是报文头部和 MAC 校验. 因此, 除去报文头部 4 个字节的属性特征分布(状态、版本、模式等)和末尾 MAC 字段常被提取和检查, 其余字段都易被检测设备忽略, 很适合作为隐藏消息字段. 结合 NTP 协议中的设计不严密的部分, 我们在上行通道使用下面几个字段来隐藏数据.

① 利用时间戳(Timestamp)最低有效位(LSB)

NTP 报文中有四个时间戳字段: 参考时间戳、原始时间戳、接受时间戳、传送时间戳. 时间戳字段用的是二进制补码编码方式. 此时我们将时间戳信息进行轻微的变换而使其仍符合时间信号特征. 我们可以利用时间戳的最低有效位来隐藏信息. 最低有效位指的是一个二进制数字中的第 0 位(即最低位). 但是使用这种方式无法携带较多的数据, 难以满足文件传送、远程桌面控制等大数据量密集通信应用的需求.

② 利用参考标识符(Reference Identifier)字段

KOD(Kiss of Death)包是一种当连接状态未定义或者无效时, 客户机和服务器之间传递状态报告和访问控制的数据包. 这种数据包的参考标识符字段没有格式要求且不认为具有有效数据. NTPV4 协议指定时规定[10]: 如果客户端或者服务器的一端使用 IPV6 地址, 另一端使用 IPV4 地址, 两方进行通讯时, 将无法完成时间循环(time looping)校验. 此时参考标识符字段也没有格式限制且不认为具有有效数据. 因此, 几乎所有的网络分析工具都不把参考标识符字段当成访问控制的判断依据. 这为基于该字段的信息隐藏提供了非常有利的条件.

③ 利用扩展域(Extension Field)填充字段

在 NTP 协议的最新版本 NTPV4 中, 可以在头部之后, MAC 校验字段之前添加一个或多个扩展域. RFC5905 中, 扩展域的内容没有被明确定义, 但被强制要求最少填充至 16 字节, 最大填充长度未作要求. 这样, 根据 UDP 协议包的最大长度 512 字节, 除去 UDP 报文首部 8 字节和 NTP 报文首部 48 字节, 我们可以让填充字段扩充到一个十分可观的长度(456 字节).

因此, 上行通道即把客户端的隐蔽信息传送给服务器端. 可携带数据的域: Timestamp、Reference ID、Extension Field.

客户端请求:

LI: 0 VN: 4 Mode: 3 Stratum: <client_stratum> Poll: <client_poll> Precision: <client_precision>
Root Delay: <client_delay>
Root Dispersion: <client_dispersion>
Reference ID: <client_refid_data>
Reference Timestamp: <client_reftime_data>
Origin Timestamp: <client_org_data>
Receive Timestamp: <client_rec_data>
Transmit Timestamp: <client_xmt_data>
Extension Field: <client_extention_data>

2)基于协议构建下行通道

上文阐述的基于报文的隐蔽信道的方法同样也适用于下行通道. 但如同 Action Dan 利用 NTP 隐蔽信道传输数据时, 遇到的服务器对客户端单向被动下行信道带宽不足的问题, 我们可以利用 NTP 协议中不严密的地方, 构建“合法”的通信信道.

① 利用 KOD 包

一个可控的隐蔽通道应该具备指令信号, 对上行和下行通道的流量开关和传输速率作出适当的控制. 此时, 我们可以使用KOD包中的参考标识符字段作为这种信号. 这个字段特定的 ASCII 字符串(DENY and RSTR、RATE 等)原本就是设计来实现客户端和服务器之间的访问控制. 这种信号可以被沿用作为隐蔽通道信息传递的控制信号, 作为下行信道中服务器端通知客户机开启和关闭隐蔽通道的指令信号.

② monlist 指令

NTP 服务包含一个 monlist 功能[11], 它被设计用于监控 NTP 服务器. NTP 服务器响应 monlist 指令后会返回与自己进行过时间同步的最后 600 个客户端的 IP. 这意味着, 一个很小的请求包, 就能获取到大量的活动 IP 地址组成的连续 UDP 包. 响应包按照每 6 个 IP 进行分割, 最多有 100 个响应包, 通常每个包为 480 字节. 利用 monlist 指令, 可以得到一个非常不错的下行带宽.

因此利用NTP服务的monlist功能, 可以将隐蔽数据伪装成合理的响应包在网络中进行传输. 在 monlist 触发指令下, NTP 服务器可以使隐藏信道具有不错的带宽, 实现“接受简短命令, 返回大量结果”的下行通道模式.

③ 利用 listpeers 指令

Listpeers 指令的使用和 monlist 指令类似. NTP 服务器响应 listpeers 指令后就会返回与 NTP 服务器进行过时间同步的所有对等机的 IP 地址. 因此 listpeers 指令也可以作为下行信道的触发指令来传递隐蔽信息.

因此, 下行通道即把服务器端的隐蔽信息传送给客户端. 可携带数据的域: Receive timestamp、Transmit timestamp、Reference ID、Extension Field.

服务器端请求:

LI: 0 VN: 4 Mode: 4 Stratum: <server_stratum> Poll: <server_poll> Precision: <server_precision>
Root Delay: <server_delay>
Root Dispersion: <server_dispersion>
Reference ID: <server_refid_data>
Reference Timestamp: <server_reftime_data>
Origin Timestamp: <server_org_data>
Receive Timestamp: <server_rec_data>
Transmit Timestamp: <server_xmt_data>
Extension Field: <server_extention_data>

由于 NTP 协议具有普遍性和不可替代性, 网络监控设备通常不对 NTP 数据包做检查. 但在网络安全措施较为严格的网络中, 网络管理员强制关闭 123 端口, 此时只允许从内部网络向外部网络发送 NTP 请求, 而不允许外部网络主动向内部网络发送 NTP 数据包. 这种情况有两种解决办法: 第一, 如果外部主机需要主动对内部主机发起隐蔽通道建立请求, 那么就需要把 NTP 报文 Mode 字段设置成应答模式, 伪装成服务器或对等主机的应答报文, 使其穿过防火墙; 第二, 内部主机首先发起 NTP 请求, 内部主机和外部主机的连接即可建立[12].

3.3 基于 NTP 协议隐蔽通道的实现

在发送端, 需要先构造 NTP 数据包: 需要加强隐秘性的情况下, 首先可以对隐秘信息进行加密, 把加密后的信息映射到合法的 NTP 地址段单元, 这样的 NTP 数据包是“合法”的. 在接收端通过提取相应字段的数据, 进行解密操作, 恢复隐秘信息. 整个过程如图 5 所示.

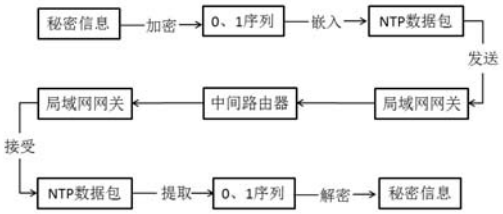

图 5　基于 NTP 协议的隐蔽通道的实现

通常情况网络设备并不检测 NTP 协议包的时间戳和扩展字段, 甚至在一些入侵检测系统(例如 Snort)的用户手册里还强调: “如果用户的服务器(比如 NTP、NPS 和 DNS 服务器)会妨碍端口扫描的探测, 可以通知相应模块忽略源自这些主机的 TCP SYN 和 UDP 端口扫描”. 由此看出, 这些网络安全设备一般都在过滤包时主动忽略 NTP 包. 因此从理论上说这种隐蔽通信是可行的; 本文通过 XCAP 方式进行 NTP 隐蔽通道通信试验, 穿过防火墙和局域网网关, 在接收端正确接收到含有隐秘消息的数据包, 从实践中证明了本文的方法.

```
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 42 6f   ........ ......Bo
0070  62 2d 69 73 2d 61 2d 73  70 79 2d 6b 65 65 70 2d   b-is-a-s py-keep-
0080  68 69 6d 2d 75 6e 64 65  72 2d 73 75 72 76 65 69   him-unde r-survei
0090  6c 6c 61 6e 63 65 00 00  00 00 00 00 00 00 00 00   llance.. ........
```

图 6　客户端向服务器传输隐蔽信息

在图 6 的实例中, 客户端发起 NTP 时间同步请求, 将隐蔽信息 Bob is a spy keep him under surveillance 嵌入到 NTP 报文中. NTP 报文最终送至 NTP 服务器, 服务器端解析 NTP 报文可以获得隐蔽信息.

4　安全性考虑

隐蔽通道建立时, 还必须考虑安全方面的因素, 以保护隐蔽通道资源. 本研究下一步工作将从以下几个方面展开[13]:

1) 认证: 客户端和服务器必须相互认证, 以免第三方的数据干扰, 防止隐蔽通道资源被恶意占用.

2) 数据流加密: 由于 NTP 协议报文的传送一般情况下采用明文形式传送, 所以许多网络嗅探器可以捕捉到网络中传输的 NTP 数据包, 并分析出 NTP 报文中的内容, 这样 NTP 隐蔽通道易被发现, 从而降低 NTP 隐蔽通道的安全性. 对于这种情况, 通常我们可以对 NTP 隐蔽通道中数据流进行加密.

3) 数据完整性: 为了检查隐蔽通道在传输数据的过程中数据是否被篡改, 可以沿用 NTP 报文中的可选字段 dgst 字段存放校验哈希值来保证数据完整性.

4) 重放保护: 为避免攻击者利用以前的通信记录来访问客户机或者服务器, 客户机和服务器应该制定相应机制应对重放攻击.

5　基于NTP的隐蔽信道的优点

目前, 许多黑客正在利用基于 NTP 的隐蔽信道来进行攻击, 这主要是因为基于 NTP 协议的隐蔽通道有着它特有的优势:

1) 目前绝大多数网络都开放 NTP 服务. 在 Fyodor 组织 2014 年互联网主机端口开放状况报告中, NTP 所在的 123 端口排名第四位, 仅次于 NetBIOS 协议 137 端口、SNMP 协议 161 端口和 SQLServer 查询的 1434 端口, 而居然比 DNS 查询 53 端口开放程度还要高. NTP 协议数据包能顺利通过防火墙、NAT 设备而不被过滤或拦截. 这些为基于 NTP 协议构建隐蔽通道创造了有利条件.

2) 蒙骗式通信. UDP 协议是面向无连接的. NTP 隐蔽通道利用 NTP 报文基于 UDP 协议进行传输的特点, 在 NTP 隐蔽通道的内部网络源主机节点处随机伪造内部网络其他主机 IP 发送伪装的 NTP 查询请求, 最终服务器响应包会返回到其他主机 IP, 从而平衡流量分布特征.

3) 较高的带宽. NTP 协议报文有较多可以携带隐蔽信息的字段. 仅在 NTP 报文头部(48 字节), 就有若干个常被忽略而不被检测的协议数据单元. 此外, 我们还可以利用 NTP 报文的扩展字段作为隐藏信息的载体.

6　结语

本文利用互联网公认的时间同步工具 NTP 协议的通用性和透明性, 提出了基于 NTP 协议的隐蔽通道构建机制, 是非常具有发展前景的信息隐藏技术. 本文通过分析 NTP 协议数据单元, 最大限度地利用 NTP 报文载荷, 构建了上行隐蔽通道; 充分利用查询/应答机制特点, 设计了“接受简短命令, 返回大量结果”的下行隐蔽通道: 将隐藏信息伪装成普通的 NTP 报文, 进行隐秘消息的传递. 本文提出的基于 NTP 协议的隐蔽通道技术具有穿透力强, 隐蔽性好, 流量分布特征不明显等优点.

参考文献

1 王永杰,刘京菊.基于 DNS 协议的隐蔽通道原理及性能分析.计算机工程,2014,40(7):102–105.

2 张令通,罗森林.基于 TCP 协议首部的网络隐蔽通道技术研究.计算机工程与科学,2014,36(6):1072–1076.

3 杨智丹,刘克胜,王康,汪松鹤.基于 IP 报头选项的网络隐蔽通道技术.计算机工程,2009,35(13):125–127.

4 罗成.基于 Windows 消息机制的 HTTP 隐蔽通道的设计与实现[硕士学位论文].上海:上海交通大学,2008.

5 廖晓锋,邱桂华.一种基于 Web 访问模型的网络隐蔽通道.计算机系统应用,2013,22(2):10–14.

6 曹自刚,熊刚,赵咏,郭莉.隐蔽式网络攻击综述.集成技术,2014,(2):1–16.

7 曹自刚.隐蔽式网络攻击检测关键问题研究[硕士学位论文].北京:北京邮电大学,2015.

8 王相林,赵颜昌,李黎.一种基于源 IP 地址的信息隐藏技术.计算机应用与软件,2010,27(10):222–224.

9 王永吉,吴敬征,曾海涛,丁丽萍,廖晓锋.隐蔽信道研究.软件学报,2010,21(9):2262–2288.

10 沈燕芬.用于网络时间同步的 NTP 协议.现代计算机(专业版),2004,(4):54–56.

11 杨先杰.NTP 协议的研究与应用.电力信息化,2011,9(6):28–32.

12 师海燕,梁洪波.基于ICMP协议的网络隐蔽通道技术的分析.电脑知识与技术(学术交流),2007,2(10):963,989.

13 强亮,李斌,胡铭曾.基于 HTTP 协议的网络隐蔽通道研究.计算机工程,2005,31(15):224–225.