Global Information Assurance Certification Paper

Bryan S. Brandt
GIAC Securing Windows (GCNT) LevelTwo Practical Assignment v. 2.1b
SANS Lone Star II; Dallas, TX
31 May – 3 June 2001

Remotely Administering Windows 2000

Introduction

With the release of the Windows 2000 Server Family, Microsoft has attempted to position itself as the definitive market leader in the traditionally competitive server Operating System (OS) arena. According to a survey conducted by market research firm IDC, Windows NT held a 30.3% share of the server OS market prior to the release of Windows 2000. This left it trailing the fragmented Unix market share that includes Solaris, HP-UX, AIX, and Irix by 22.4%. OSes such as Linux or MacOS held the remaining 17%. (Hellman 12) Microsoft fully intended to extend its market share with the release of Windows 2000, and more recent estimates reveal that is has done so with a fair degree of success.

While operating systems such as Linux have risen to the challenge, it is undeniable that Microsoft continues to advance and pervade the server OS market. In January of 2001, Microsoft began touting the "five nines" reliability of the Windows 2000 Server Family. This performance metric indexes uptime as a percentage of the 24 x 7 x 365 ideal, where "five nines" indicates 99.999 percent uptime. "The Aberdeen Group found that Windows 2000 Servers delivered 99.95 percent uptime right out of the box, before the servers were fully optimized for the environment, and before the IT staff had gotten up to speed using the new operating system." (Microsoft, *Reliability* n. pag.) Microsoft has developed the Datacenter Server Edition of Windows 2000 specifically to meet the 99.999 percent uptime goal (which corresponds to just over five minutes of downtime per year) on qualified systems, and has delivered systems that it claims are capable of this standard to several companies already. While little factual (*i.e.*, non-promotional "real-world" performance metrics) evidence exists to support these claims, the prospect of only five minutes of downtime per year is indeed very promising.

As a result of the recent improvements in reliability, Microsoft Windows 2000-based servers have seen more widespread use in multi-site and collocation environments than their NT-based counterparts. This more distributed approach to enterprise computing architecture necessitates a stable and reliable means of remote administration, an area in which the graphical MS Windows environment traditionally has not excelled.

Unix and Linux are configurable almost exclusively in a command-line environment, although the capacity to forward X11 sessions from one host to another does exist. As a result, remote administration over reliable, secure (*i.e.*, encrypted) channels has been in existence for several years. Without the complex graphical infrastructure, early advancements gave *NIX hosts a distinct edge in distributed computing environments.

Microsoft made its first foray into the realm of centralized resources with Windows NT 4.0 Terminal Server Edition. Terminal Server was Microsoft's approach to the "Network Computing" paradigm, heralded as the imminent technological revolution by nearly every industry publication in the mid- to late- 1990s. The theory was that powerful computing resources could be made available to much less technologically advanced "thin clients." This package allowed less-capable desktops (running Windows 3.x on what even then was considered minimal hardware) to harness the resources of centralized NT-based servers over a network.

Terminal Server has since evolved (devolved?) from a specialized NT-based operating environment into a network service offered with every Windows 2000 Server platform available today. This service makes provisions not only for the centralized computing environment supported by Terminal Server, but also for a small scale, low penalty means of remotely and graphically administering servers *for free*.

With the inclusion of Terminal Services in every Windows 2000 Server package, remote, graphical administration is now easier and more accessible than ever; but is it the best option for every situation?

Microsoft Windows 2000 Terminal Services

Overview

Microsoft's Terminal Services, previously available only in the Windows NT 4.0 Terminal Server Edition, was originally intended to enable server-side application processing from thin clients. Client packages were available for Windows 3.x, 9x, NT, and (later) even CE that allowed less capable machines to function as viable desktop PCs. Older or smaller-footprint hardware is thus able to perform the same tasks as more expensive units without the added expense of modern hardware. As a secondary advantage, this model allows a single server software upgrade to affect all users without the need to reconfigure an organization's workstations. Terminal Services is now offered for use free of charge in Remote Administration mode, thus allowing for remote graphical administration of servers.

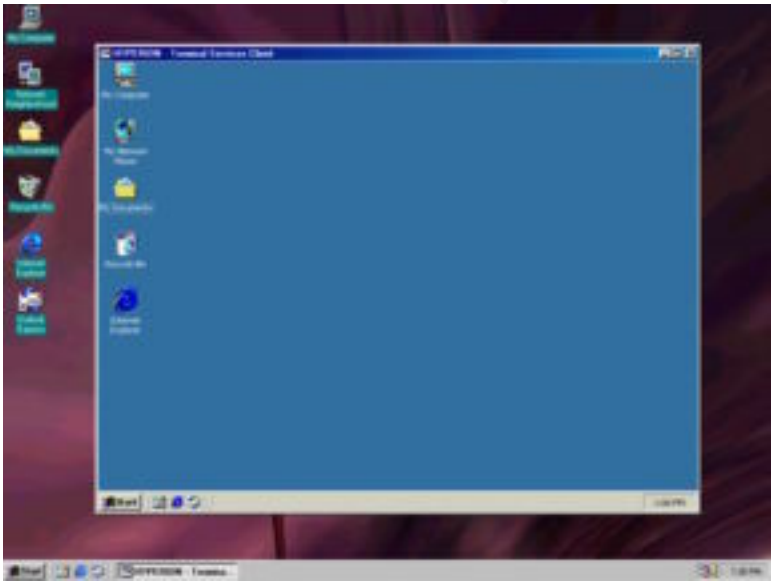
Terminal Services employs the Remote Desktop Protocol (RDP), a variant of the International Telecommunications Union (ITU) T.120 protocol suite. According to the ITU, "[t]he T.120-Series of Recommendations collectively define a multipoint data communication service for use in multimedia conferencing environments." (ITU, *Summary* n. pag.) Incidentally, Microsoft employs a variant of the same technology in its NetMeeting conferencing package. It should be noted that Terminal Services is also capable of supporting the similarly capable Independent Computing Architecture (ICA) protocol employed by packages such as Citrix Metaframe for interoperation with a wider range of client packages.

A fairly in-depth technical overview is available in Microsoft's *Windows 2000 Security, Technical Reference*, excerpted here:

> RDP is a multichannel-capable protocol, which allows for separate virtual channels for carrying serial device communication and presentation data from the server, as well as encrypted data from the client's mouse and keyboard. RDP also permits real-time data distribution from an application to multiple users. Data to be transmitted from an application or service is passed down through the protocol stacks, sectioned, directed to a Multipoint Communications Service (MCS) channel, encrypted, wrapped, framed, packaged onto the network protocol, and finally addressed and sent over the wire to the client. The return data works the same way but in reverse: the packet is stripped of its address, then unwrapped, decrypted, and so on, until the data is presented to the application for use. (Microsoft, *Windows 2000 Security* 427-8)

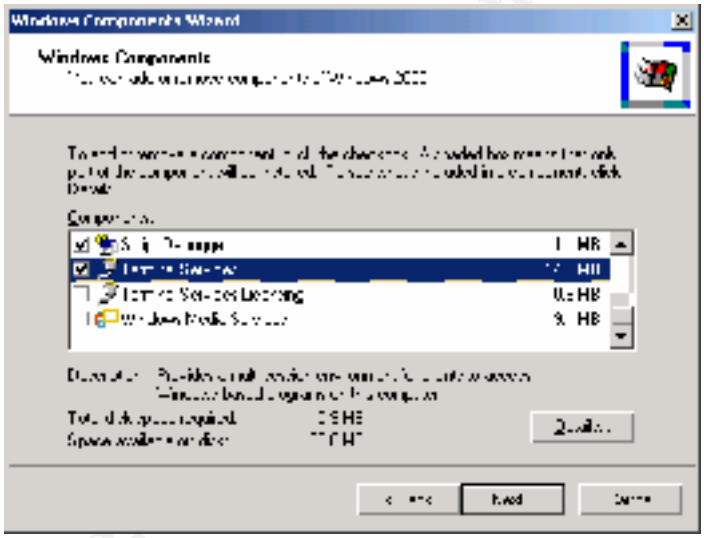
Client packages access the Terminal Server via TCP Port 3389 using the encryption level requested by the server. Encryption is available at three levels: Low, Medium, and High. Medium and High security RC4 encrypts traffic bi-directionally with 40/56- and 128-bit keys, respectively. (NOTE: 56-bit keys are reserved for Windows 2000 clients only. Older versions of Windows are limited to 40-bit encryption.) Low encryption uses the same RC4 encryption algorithm and key length as Medium, but only data transmitted from the client to the server is encrypted. (Mathers 41) While Low encryption does protect information such as usernames, password hashes, and keystrokes, it is possible to compromise other critical data as it passes from server to client through a simple traffic analysis.

Once Terminal Services have been properly installed, its usage is fairly seamless in Remote Administration Mode. The user is presented with a graphical session on the remote machine within their local session, as shown. For all intensive purposes, the remote user has an active session as if he or she had logged in from a local console. It should be noted that authentication is limited to passwords only, as this logical access does not allow for the use of tokens or smart cards. Audit logging and group policy apply exactly as they would in a local session. User-level preferences such as window style, color scheme, and desktop wallpaper also apply. For increased usability, Terminal Services allows port redirection as well as local resource addressing. This allows system beeps to be heard locally, printing to occur to local LPT or COM ports, and local drives to be accessed by server-side applications. Sessions may remain active even if the user closes the local window, and they may be resumed from any other host with network



access. Multiple administrators may even share a common session to collaboratively debug network or system issues. The server saves all state information, and sessions may be monitored, closed, or joined from the local console.

Integration

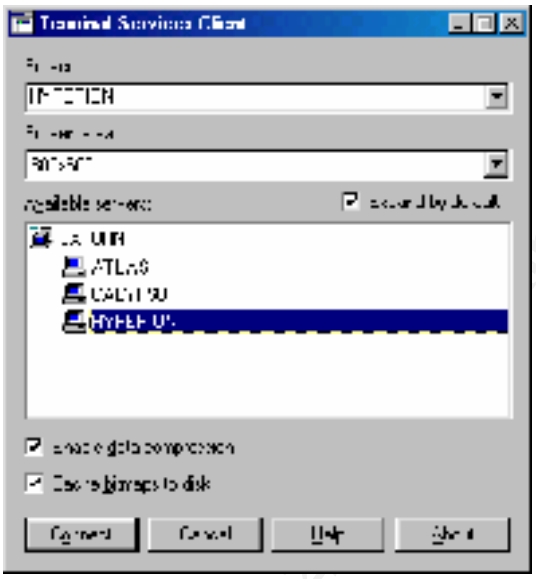


All members of the Windows 2000 Server Family include Terminal Services as one of the services packaged with operating system. As shown in the figure at right, installing this service is as simple as making a selection either during installation or from the Windows Components Wizard (found in Control Panel ⇨ Add/Remove Programs ⇨ Add/Remove Windows Components) at any time thereafter. The 'Terminal Services Licensing' option need not be installed for use in Remote Administration Mode. A dialog box appears immediately following installation requesting that the user select either Remote Administration or Application Server mode. Selecting Remote Administration does not require the purchase of any additional licensing, and connections are automatically tuned for minimum impact on the server's resources.

The Microsoft Windows 2000 Administrator's Companion states that the typical terminal services connection requires a minimum (*i.e.*, for the session alone, excluding any applications run from within it) of twenty megabytes of server memory and an average of between two and six kilobits per second of available network bandwidth to support the connection. (Crawford 901) Performance tuning in Remote Administration Mode will draw more conservatively on system resources should the system be experiencing heavy loads at the time a connection is made.

Once the service has been installed, the Terminal Services Client Creator application must be run to generate the required software for the hosts from which administration will take place. Win32 clients will require two 3.5-inch, 1.44-megabyte floppy disks, while Win16 clients will require four. The client installation process is fairly painless, offering to set default options for display resolution, caching, compression, and automatic logon (NOTE: This is **NOT** recommended for the purposes of remote administration. Should your laptop be lost or stolen, or should your workstation be physically compromised, stored passwords could grant an intruder full administrative privileges on your server!). The client software may override defaults at any time as desired.

The client package, shown below, displays a list of Terminal Servers available in your domain. Once a selection has been made, simply clicking 'Connect' and properly authenticating will grant access to the server as discussed in the Overview. Port 3389 may be forwarded at the perimeter firewall/router to allow for remote (*i.e.*, Internet accessible) access.



Cost Issues

Terminal Services, in its more traditional Application Server Mode, is licensed per seat in a Client Access License (CAL) format similar to that used by Windows 2000 Server. One CAL must be purchased for each client, regardless of the client package being used; a five pack of Terminal Services CALs has a published price of $749.00. (Microsoft *Pricing* n. pag.)

However, in Remote Administration Mode, Terminal Services is provided for use free of charge. Citrix *client* packages are also available for free download, although the use of ICA does require the purchase of a Metaframe server from Citrix.

Advantages of Terminal Services

The primary advantage of Terminal Services is its tight integration with the Windows 2000 Server platform. Employing the native RDP protocol to service Win32 clients provides perhaps the most seamless interface of any available remote administration option. Redraws in Terminal Services are the smoothest of any package, as it is the only one permitted the use of internal Windows system calls.

Terminal Services makes use of all local system and group policies as well as system auditing defaults. As a result, a sound overall system configuration may not require much additional effort to adequately monitor and control remote administration aside from granting Terminal Services login privileges to the Administrators group.

Login attempts may be limited in accordance either with default system policy or a specific Terminal Services policy. This reduces the effectiveness of a brute force password attack where a valid username is already known. Violations of either policy are recorded in the system logs, and the account is not permitted to logon for a specified period of time. Administrators should be cautious regarding the use of this feature, however, as the possibility exists that this safeguard itself may be used to denial-of-service an administrator attempting to log in to perform legitimate activities.

Disadvantages of Terminal Services

While Terminal Services does provide an encryption service, it defaults to the Low setting on installation. As such, it is possible to inadvertently have unencrypted data passing from server to client. Additionally, 40-, 56-, and even 128-bit encryption will soon be considered insufficient to safeguard sensitive data such as administrator-level passwords. To date, there are no options other than the Low, Medium, and High encryption levels already in place.

Terminal Services, in typical Microsoft fashion, does not address remote administration in a multi-platform environment. In a mixed computing environment, Citrix distributes Independent Computing Architecture (ICA) client packages that are derived from the same ITU protocol family as RDP. Client packages are currently available for: Win32, Win16, WinCE, PocketPC, EPOC, DOS32, DOS16, Linux, Solaris, Tru64, HP-UX, AIX, Irix, SCO, Macintosh, and OS/2. (Citrix *ICA* n. pag.) These packages require the Citrix Metaframe server, although Terminal Services will employ the ICA if present.

Several exploits have been released for this service since Microsoft NT 4.0 Terminal Server Edition became commercially available. If Terminal Services is being deployed to grant access across the Internet, it is potentially vulnerable to such attacks. Should an account be compromised, it would provide access on an administrator level; although, the majority of the exploits have focused on preventing legitimate access. Terminal Services connections should ideally be allowed only from authorized IP ranges and closely monitored (*e.g.*, Intrusion Detection, firewall rules that log SYN packets to TCP Port 3389, etc.) for signs that an attack is being leveraged.

AT&T Virtual Network Computing (VNC)

Overview

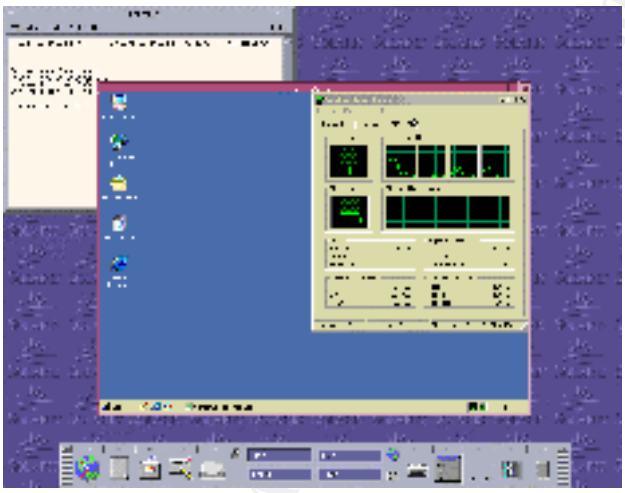
Virtual Network Computing, or VNC, is a software package originally developed by the Olivetti Research Laboratory (ORL) in Cambridge, England. ORL has since been acquired by AT&T, and is now the AT&T Laboratories Cambridge.

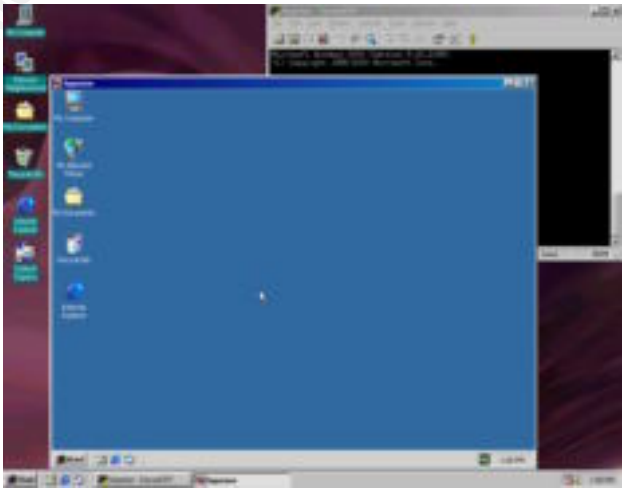
VNC is a freeware package, downloadable from http://www.uk.research.att.com/vnc/, designed specifically for replicating the operating environment on one computing device to another. The premise of VNC is fairly simple: "VNC is an ultra-thin client system

based on a simple display protocol that is platform-independent." (Hopper 33) A server exports any graphical interface at the framebuffer level, transmitting over any reliable networking technology. The original intent of the package was to provide backend computing ability for thin client devices; however, it is particularly well suited for remote administration tasks.

Processing is performed almost exclusively server-side, where the graphical interface is parsed into a common format, returned data streams are reassembled, and commands are executed. The client, or "viewer," need only reassemble simple graphical elements and transmit input (keyboard, mouse, stylus, etc.) data back to the server for processing. AT&T distributes binaries for Windows, X-Windows, and Macintosh; however, various contributors have expanded upon the concept to produce client and server packages for literally hundreds of different environments and technologies.

Once VNC has been installed, it provides an interface very similar to that of Terminal Services to the viewer. The remote desktop is replicated in an application window on the local desktop. Alternately, VNC offers the option of displaying the console in any Java-enabled browser directed at TCP Port 5800 of the server. As shown below, the same Windows 2000 Server used in the Terminal Services example is now being easily replicated using VNC port forwarded over Secure Shell in both the Solaris 8 Sparc Platform Edition and Windows 98 Second Edition environments.





As commands are entered on any of the available clients, the local console reflects each change in real time. Clients are stateless, and a session may be resumed from any other client without any interruption. Multiple instances of the same session may be opened concurrently, and several administrators may share one common desktop while network or server issues are being resolved. Unlike Terminal Services, VNC provides only *one* direct interface with the server. There are no provisions for mapped drives or other resources; all commands and processing are executed in the context of the remote server. This allows the protocol to remain more simplistic, and this in turn contributes to its unparalleled versatility.

Integration

While VNC is not packaged with Windows 2000 Server, its installation is fairly trivial. WinVNC must be downloaded from http://www.uk.research.att.com/vnc/download.html; version 3.3.3r9, released on 19 March 2001, is the most recent available to date. It should be noted that VNC is licensed under the GPL General Public License. The source code is freely available, and there are a wealth of contributed packages available as well. As a result, other compatible versions do exist (*e.g.*, TridiaVNC), and the user is not limited only to products of AT&T.

NOTE: There is no built-in safeguard against brute-forcing VNC session passwords. It is recommended that access to TCP Ports 5800 and 5900 be restricted during installation and until such time as adequate countermeasures are employed.

Installation consists simply of running the installation package and revising the default registry settings if desired. VNC is designed with versatility in mind, and as such several very useful options exist. These options offer the user the ability to control access to the server, specify whether the user account is to be logged off when a session is closed, or allow local loopback connections, for example. Unlike the Terminal Services environment, however, options must be set server-side, as the client package has no capacity to alter them real-time.

As VNC replicates the server *exactly* as it exists at the physical console, it may be helpful to reduce the server's screen resolution to a setting lower than that employed by the machines from which the viewer will run. For instance, an 800x600 server replicates fairly well on a 1024x768 desktop with no distortion and no additional processing overhead for software scaling. While a scaling option is available, it is often faster and more reliable (scaling may inadvertently distort images) to simply adjust the screen resolution.

Traditionally, VNC has been tunneled over Secure Shell (SSH) to enhance security. Secure Shell, in its most basic form, provides terminal access to a server over an encrypted tunnel. It is possible to "port forward," or have a port on the local host appear for all intensive purposes as a port on the server (or any other server accessible by the SSH server) such that all communication with that port is sent over the encrypted tunnel.

The RFB (Remote FrameBuffer) protocol on which VNC is based does not inherently encrypt data; therefore, this service is an excellent compliment to the VNC service. Encryption is not a direct requirement for remote administration. However, it is strongly recommended that these additional steps be implemented in order to preserve the integrity and security of your network.

SSH Communications Security distributes a Secure Shell server for the Windows platform that suits this application well; this server is the one employed for demonstrating the concepts in this paper. As an alternative, Van Dyke Technologies,

Inc. distributes VShell, another well-implemented Windows SSH2 server. A variety of viewer packages exist for virtually every platform.

For example, a *NIX host would connect to the VNC server with the following syntax:

```
#ssh xxx.xxx.xxx.xxx +C –L 5900:xxx.xxx.xxx.xxx:5900 –l Administrator
```

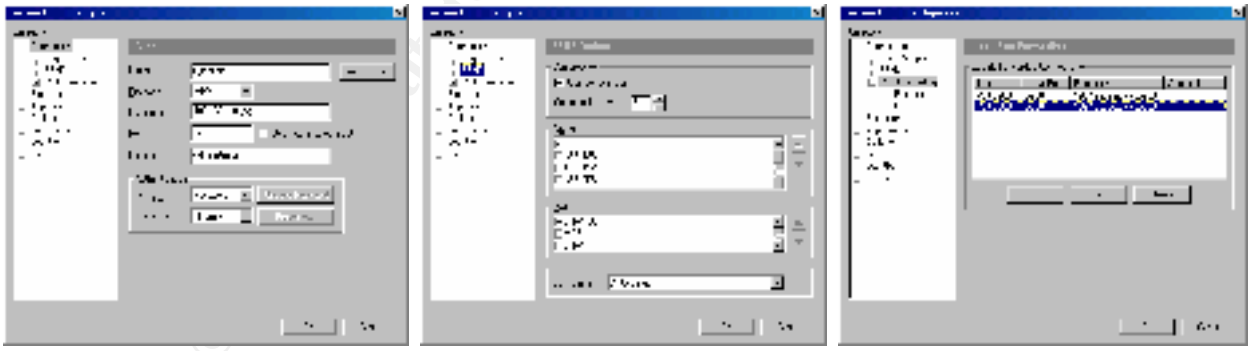
And subsequently, from a separate console window:

```
#vncviewer 127.0.0.1
```

This initiates a Secure Shell session to host xxx.xxx.xxx.xxx using the Administrator login and enabling data compression. Once this session is active, any connection made with port 5900 of the local host is actually made with port 5900 of xxx.xxx.xxx.xxx over the encrypted channel. The VNC viewer must be run from a separate console window, as the console from which SSH was run now represents a terminal on the server, and the viewer must be run as a local process.

It should be noted that text-based administration tasks, such as those employing command-line Resource Kit utilities, may be performed directly in the SSH environment. SSH provides a command shell window on the remote server, allowing for direct command execution without the graphical interface provided by VNC. Windows 2000 is graphically oriented, however, and it is thus assumed that the administrator would not rely on SSH alone as a remote administration solution.

Microsoft Windows-based clients can also be configured for port forwarding. Using SecureCRT 3.3.1 by VanDyke Technologies, Inc., the configuration would look something like the following:



Using third party encryption with VNC in this manner may, if properly configured, provide tighter security than that integrated with Terminal Services. While the current Windows release of the SSH Communications server does not support AES to date, the recent *NIX release of version 3.01 does and a Windows port is likely not far behind. Van Dyke's VShell already supports AES in the current release (1.1).

Once an SSH server has been installed for use with VNC, a DWORD registry entry named LoopbackOnly may be created and given an value of 1 in

HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\. This entry will prevent access from any network interface other than 127.0.0.1. As port forwarded connections appear to originate from 127.0.0.1, this limits access to the remote administration environment to SSH connections (and the local console, although this would be somewhat redundant).

If SSH is not employed, access control is made possible by creating a REG_SZ entry named AuthHosts, also in HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\. The value of this entry may be used to determine what subnets are allowed unencrypted access to the VNC service. In this manner, combinational variations of encrypted and unencrypted communications are made possible. For example, an internal file server that resides on a network with minimal insider threat may set the following values:

AllowLoopback	1
LoopbackOnly	0
AuthHosts	-:192.168.1

This configuration would allow unencrypted access from the local subnet (in this case, 192.168.1.x) as well as remote access via SSH when port 22 of the file server is made accessible using a port forward at the perimeter firewall or router. A Linux-based firewall, for example, would use the following syntax to implement this:

```
#ipmasqadm portfw –a –P tcp –L xxx.xxx.xxx.xxx 22 –R 192.168.1.yyy 22
```

Where xxx.xxx.xxx.xxx is one of the organization's assigned external IP addresses. The SSH Communications Security SSH server provides Windows audit logging of every connection made, including the time of access and the originating IP address.

Cost Issues

VNC is freely distributed under the terms of the GNU General Public License. There is no charge for its use.

Securing your VNC server through the use of SSH, however, will incur some cost. This is dependent, of course, on which Secure Shell server is employed. The SSH Communications Security server for Windows has a list price of $565. Van Dyke's VShell is available for as little as $249 for a limited license (two concurrent connections, one server). Said limited license is sufficient for the purpose of remote administration, provided that no more than two administrators would under any circumstances access the server simultaneously.

Disadvantages of VNC

VNC has one significant weakness: the RFB Protocol on which VNC is based has no inherent data encryption capability. In keeping with the lightweight ideal for its implementation, data being passed between client and server is sent in the clear.

Appendix I fully addresses this concern. The recommendation that SSH be used as a tunnel for VNC remedies this condition.

Several exploits have also been released for VNC, as in the case of Terminal Services, where the service is permitted access via the Internet. One widely publicized release included a buffer overflow for the VNC server itself; however, other methods exist for brute forcing passwords and denial-of-service. The recommendation that SSH be used as a tunnel for VNC remedies this condition as well.

The remotely replicated local console does present one other issue: replicating screen resolutions higher than that of your local console result in a window that must scroll in order to fit the entire desktop. The settings exactly reflect those of the server, and there is no client-side option to adjust this reliably. While there is a scaling option available, it very often produces output that is less than clear. As discussed previously, simply setting a lower resolution at the server will remedy this condition.

As the Microsoft Windows graphical interface is fairly complex, VNC as a third-party application cannot replicate it as well as an integrated package such as Terminal Services. There is a noticeable lag when slow (*e.g.*, dialup) connection is used for remote access, although much of this is due to the processing overhead incurred by the encryption being implemented by SSH. Even reasonably fast connections will need a manual refresh periodically.

Another consideration when selecting a remote administration option is that VNC provides access to a host *by remotely replicating the local console*. This is in contrast to the Terminal Services approach of opening a session on the remote system. A registry option does allow for multiple administrators to have the same session open; however, it is just that: *the same session*. In an environment where multiple administrators may be required to log in to the same system simultaneously to perform independent tasks, this may be an undesirable approach.

Advantages of VNC

VNC has several advantages over the more traditional Terminal Services approach to remote administration. Perhaps the most significant of these is its interoperability with a much wider range of platforms and operating environments. While it is possible for Terminal Services to provide remote desktops to *NIX machines, the reverse does not apply: there is no RDP-based server for *NIX hosts. VNC may be used to serve remote desktops from a variety of disparate architectures, making it ideally suited for a mixed computing environment.

In addition to the binaries provided by AT&T, contributors have offered perhaps the widest range of platforms addressable with a single protocol. Unix X sessions may be exported to a PC, PC sessions may be exported to a Macintosh, etc. Server and client packages exist for a significant number of platforms, and all interoperate seamlessly. It

is actually even possible to remotely *and securely* administer servers from handheld PalmOS-based devices (http://www.btinternet.com/~harakan/PalmVNC).

One client package can access any server, regardless of the platform on which it operates as they all employ a common protocol. The Windows version of the viewer occupies only 172k of disk space, and yet it is as powerful, if not more so, than the four-disk Terminal Services client. Additionally, the client binary requires no installation and may even be run from a floppy.

Conclusion

As is the case with virtually any enterprise-grade IT solution, careful planning and precise, security-conscious implementation is critical to the overall success of a remote administration effort. A significant design phase should precede any multi-site or collocated effort; indeed, the same holds for any effort in which your organization establishes an Internet presence. Data integrity and security is as important, if not more so, than its availability. Windows 2000 offers much greater flexibility than any Microsoft platform to date; however, its configuration and implementation should be fully understood before exposing your system to the world, as it is also the most complex Microsoft platform to date.

With regard to remote administration, both Terminal Services and VNC maintain their strengths.

In the case of an exclusively Microsoft Windows 2000 environment, Terminal Services is an excellent option for remote administration. The higher levels (56- and 128-bit) of encryption are available to Windows 2000 clients, and the screen refresh is unmatched. The resource penalty is fairly low, and all local policies and auditing apply.

If, however, an administrator must address a variety of infrastructures, VNC is a solid choice. It offers a remote graphical interface quite comparable to Terminal Services, however it will do so for a much wider range of devices. One low-end PC is sufficient to replicate hundreds of operating environments in as many physical locations. Performance is certainly acceptable, and the available flexibility allows for a much broader range of possibilities. As with any package so user-configurable, it does require a fairly knowledgeable administrator with a well-defined set of objectives to maximize its benefit.

Indeed the inclusion of Terminal Services free of charge is very insightful on Microsoft's part, as it encourages distributed computing and furthers their market share. AT&T likewise encourages the establishment of an open standard for cross-platform replication, a very laudable goal indeed.

In either case, careful implementation will yield a setting in which Microsoft Windows-based services may be offered on a much broader scale. Remote administration makes possible a more easily managed (and therefore less imposing) computing environment.

Appendix I

VNC Traffic Analysis

One weakness of VNC is its lack of encryption while data is in transit from one host to another. As a result, an analysis of the traffic passing between a VNC server and a client host may yield critical information such as usernames and passwords unless precautions are taken.

Assuming that an attacker has identified a VNC client and server, either through network reconnaissance or searching network traffic captures for VNC connections, a simple tcpdump filter can be written to isolate and capture data transmitted from the client to the server as follows:

```
src host sss.sss.sss.sss and dst host ddd.ddd.ddd.ddd and dst port 5900
```

Applying this filter yields exclusively data transmitted from the client to the server. The capture shown here was taken after host Y authenticated with VNC server Z and the Ctrl-Alt-Del keystroke was generated by the client. Said keystroke is easily identifiable, and as such it is not difficult to search captured traffic for its occurrence.

While an examination of the packet structure employed by VNC is beyond the scope of this document, it should be noted that keystrokes are passed from client to server in the clear and are located at the end of the packet data. A simplistic approach employed here to expose a username is an examination of the last four bytes of packet data for the characters 0000 00XX, where XX includes only values in: [30,39], [41,5A], and [61,7A]. This accounts for all alphanumeric values, thus assuming that no special characters are part of a username.

NOTE: It is not unusual for a duplicate character to appear in the stream. Duplicates have been removed for the purpose of this example; however, in practice these duplicates can often be removed based on little more than intuition. For example, it is unlikely that there would be an 'Admministrattor' account.

```
22:07:28.480839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 94:102(8) ack 947 win 63712 <nop,nop,timestamp 225450 2150374> (DF)
                     4500 003c 85bc 4000 4006 06d7 c0a8 166d
                     c0a8 166b 8001 170c f52f a113 1516 3a31
                     8018 f8e0 0e36 0000 0101 080a 0003 70aa
                     0020 cfe6 0400 ffbf 0000 0041

22:07:28.610839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 122:130(8) ack 1263 win 63712 <nop,nop,timestamp 225463 2150375> (DF)
                     4500 003c 85bf 4000 4006 06d4 c0a8 166d
                     c0a8 166b 8001 170c f52f a12f 1516 3b6d
                     8018 f8e0 0ccc 0000 0101 080a 0003 70b7
                     0020 cfe7 0401 ffbf 0000 0064

22:07:28.740839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 158:166(8) ack 1777 win 63712 <nop,nop,timestamp 225476 2150378> (DF)
                     4500 003c 85c3 4000 4006 06d0 c0a8 166d
                     c0a8 166b 8001 170c f52f a153 1516 3d6f
                     8018 f8e0 0a8d 0000 0101 080a 0003 70c4
                     0020 cfea 0401 ffbf 0000 006d
```

```
22:07:29.280839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 194:202(8) ack 2347 win 63712 <nop,nop,timestamp 225530 2150381> (DF)
                        4500 003c 85c7 4000 4006 06cc c0a8 166d
                        c0a8 166b 8001 170c f52f a177 1516 3fa9
                        8018 f8e0 07fa 0000 0101 080a 0003 70fa
                        0020 cfed 0401 ffbf 0000 0069

22:07:29.370839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 212:220(8) ack 2783 win 63712 <nop,nop,timestamp 225539 2150383> (DF)
                        4500 003c 85c9 4000 4006 06ca c0a8 166d
                        c0a8 166b 8001 170c f52f a189 1516 415d
                        8018 f8e0 0624 0000 0101 080a 0003 7103
                        0020 cfef 0401 ffbf 0000 006e

22:07:29.390839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 230:238(8) ack 3291 win 63712 <nop,nop,timestamp 225541 2150384> (DF)
                        4500 003c 85cb 4000 4006 06c8 c0a8 166d
                        c0a8 166b 8001 170c f52f a19b 1516 4359
                        8018 f8e0 0419 0000 0101 080a 0003 7105
                        0020 cff0 0400 ffbf 0000 0069

22:07:29.730839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 302:310(8) ack 4029 win 63712 <nop,nop,timestamp 225575 2150386> (DF)
                        4500 003c 85d3 4000 4006 06c0 c0a8 166d
                        c0a8 166b 8001 170c f52f a1e3 1516 463b
                        8018 f8e0 00c0 0000 0101 080a 0003 7127
                        0020 cff2 0401 ffbf 0000 0073

22:07:29.860839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 338:346(8) ack 4511 win 63712 <nop,nop,timestamp 225588 2150389> (DF)
                        4500 003c 85d7 4000 4006 06bc c0a8 166d
                        c0a8 166b 8001 170c f52f a207 1516 481d
                        8018 f8e0 fea8 0000 0101 080a 0003 7134
                        0020 cff5 0401 ffbf 0000 0074

22:07:30.030839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 374:382(8) ack 5025 win 63712 <nop,nop,timestamp 225605 2150390> (DF)
                        4500 003c 85db 4000 4006 06b8 c0a8 166d
                        c0a8 166b 8001 170c f52f a22b 1516 4a1f
                        8018 f8e0 fc72 0000 0101 080a 0003 7145
                        0020 cff6 0401 ffbf 0000 0072

22:07:30.440839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 410:418(8) ack 5571 win 63712 <nop,nop,timestamp 225646 2150393> (DF)
                        4500 003c 85df 4000 4006 06b4 c0a8 166d
                        c0a8 166b 8001 170c f52f a24f 1516 4c41
                        8018 f8e0 fa11 0000 0101 080a 0003 716e
                        0020 cff9 0401 ffbf 0000 0061

22:07:30.700839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 446:454(8) ack 6165 win 63712 <nop,nop,timestamp 225672 2150397> (DF)
                        4500 003c 85e3 4000 4006 06b0 c0a8 166d
                        c0a8 166b 8001 170c f52f a273 1516 4e93
                        8018 f8e0 f76a 0000 0101 080a 0003 7188
                        0020 cffd 0401 ffbf 0000 0074

22:07:30.800839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 482:490(8) ack 6795 win 63712 <nop,nop,timestamp 225682 2150397> (DF)
                        4500 003c 85e7 4000 4006 06ac c0a8 166d
                        c0a8 166b 8001 170c f52f a297 1516 5109
                        8018 f8e0 f4cb 0000 0101 080a 0003 7192
                        0020 cffd 0401 ffbf 0000 006f

22:07:30.970839 eth0 > 192.168.xxx.yyy.32769 > 192.168.xxx.zzz.5900:
   P 518:526(8) ack 7465 win 63712 <nop,nop,timestamp 225699 2150400> (DF)
                        4500 003c 85eb 4000 4006 06a8 c0a8 166d
                        c0a8 166b 8001 170c f52f a2bb 1516 53a7
                        8018 f8e0 f1f2 0000 0101 080a 0003 71a3
                        0020 d000 0401 ffbf 0000 0072
```

Once the packets have been identified, assembling their trailing byte (highlighted in red throughout this example) yields the following:

Hex	41	64	6d	69	6e	69	73	74	72	61	74	6f	72
ASCII	A	d	m	i	n	i	s	t	r	a	t	o	r

Following the stream only a few packets more would yield the unencrypted password just as easily. While the character set must now be expanded to include special characters, even the strongest of passwords requires little more than a cursory traffic analysis to "break."

Automated tools that are freely available on the Internet have built-in analysis functions for protocols such as VNC. Tools such as ettercap (http://ettercap.sourceforge.net) are capable of automatically extracting passwords from real-time network traffic with little or no interaction. The use of such a tool makes extracting usernames and passwords from VNC (and a variety of other services) fairly trivial.

Appendix II

Remote Administration Exploits (and How To Avoid Them)

Remote administration environments are often subject to attack for a variety of reasons. Perhaps the most self-evident of these is that they provide access, particularly superuser level access, to a host in a context from which much may be achieved. Simply reading the password out of a VNC session or more forcibly extracting the password from an SSH1 session using a man-in-the-middle attack presents an assailant with the means to access and reconfigure your server as he or she sees fit.

Allowing VNC or Terminal Services access from external IP addresses also immediately subjects your server to a higher level of scrutiny than before. Any network service may serve as a potential point of entry for an attacker. Especially in the case of a publicly accessible host, every possible software weakness must be addressed prior to allowing network access.

As recently as 25 July 2001, a denial-of-service (DoS) attack for Microsoft Terminal Services was released. Microsoft Security Bulletin MS01-040 addresses a vulnerability discovered by Peter Grundl in which a memory leak in Terminal Services' handling of incoming data may be exploited to create a condition in which all system memory is consumed. The only remedy is a forced reboot, as *all* system functions are affected. Specifically malformed RDP packets sent in succession would create this condition, effectively shutting down the server and preventing remote access of any type. While there is no opportunity for access or the escalation of privileges, the server is still seriously affected.

The majority of available VNC and Terminal Services exploits focus on denial of service. DoS attacks allow an intruder to disrupt proper operation of a server, prevent legitimate administration, and in some cases even masquerade as the crippled host. Access to these services may be restricted using access control lists (ACLs) at the router level or by firewall rule sets. This technique may be used to restrict access only to branch offices or selected other IP addresses or ranges.

It may even be advantageous to permit port forwarding through the firewall and disallow external access to certain services altogether. This creates a situation where a port scan detects little more than Secure Shell available as a service. Secure Shell exploits are seldom discovered, and as such it presents a much more formidable external appearance than the variety of services potentially being offered.

For instance, a simple nmap (http://www.insecure.org/nmap) port scan of the external address of the test network used for the examples in this paper:

```
[root@portosan /root]# nmap xxx.xxx.xxx.xxx -P0

Starting nmap V. 2.54BETA25 ( www.insecure.org/nmap/ )
Interesting ports on xxx.xxx.xxx.xxx:
(The 1537 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh

Nmap run completed -- 1 IP address (1 host up) scanned in 1090 seconds
```

It is then possible to access VNC or Terminal Services servers behind the firewall using the following or its Windows client equivalent:

```
#ssh xxx.xxx.xxx.xxx +C –L 5900:yyy.yyy.yyy.yyy:5900 –l <username>
```

or

```
#ssh xxx.xxx.xxx.xxx +C –L 3389:zzz.zzz.zzz.zzz:3389 –l <username>
```

Connections are made directly with the firewall, and requests to access internally offered services are made from its internal interface. Thus, perimeter devices may block access to all services and servers such as VNC may be configured to allow access from the local subnet only.

Through careful configuration and implementation, remote administration services may be made available to authorized users while their very existence remains concealed externally.

List of References

Citrix Systems. *ICA Clients*.
http://www.citrix.com/products/clients/ica/clients.asp, 1 Jul 2001.

Crawford, Sharon and Charlie Russel. *Microsoft Windows 2000 Server Administrator's Companion*.
Redmond, WA: Microsoft Press, 2000.

Hellman, Reed. "Industry Trends: The Future of the OS for Internet Applications."
IEEE Computer May 2000 : 12-15.

Hopper, Andy, Tristan Richardson, Quentin Stafford-Fraser, and Kenneth R. Wood.
"Virtual Network Computing." *IEEE Computer* January/February 1998 : 33 – 38.

ITU. *Summary of ITU-T Recommendation T.120*.
http://www.itu.int/itudoc/itu-t/rec/t/s_t120.htm, 1 Jul 2001.

Mathers, Todd W. *Windows NT/2000 Thin Client Solutions*.
Indianapolis, IN: Macmillan Technical Publishing, 2000.

Microsoft Corporation. *Microsoft Security Bulletin MS01-040*.
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-040.asp,
26 Jul 2001.

Microsoft Corporation. *Windows 2000 Server Family: Delivering the Level of Reliability You Need.*
http://www.microsoft.com/windows2000/server/evaluation/business/overview/reliable/default.asp, 1 Jul 2001.

Microsoft Corporation. *Windows 2000 Pricing Information*.
http://www.microsoft.com/windows2000/server/howtobuy/pricing/pricingwindows.asp, 1 Jul 2001.

Microsoft Corporation. *Windows 2000 Security Technical Reference*.
Redmond, WA: Microsoft Press, 2000.

Richardson, Tristan, and Kenneth R. Wood. *The RFB Protocol*.
http://www.uk.research.att.com/vnc/rfbproto.pdf, 1 Jul 2001.

Software Used For Demonstrations

Ettercap, SourceForge, http://ettercap.sourceforge.net

Microsoft Terminal Services, Microsoft Corporation, http://www.microsoft.com

Nmap, Insecure.org, http://www.insecure.org/nmap

SecureCRT, Van Dyke Technologies, Inc., http://www.vandyke.com

SSH Windows Server, SSH Communications Security, http://www.ssh.com

Tcpdump, Lawrence Berkeley National Laboratory, http://www.tcpdump.org

Virtual Network Computing, AT&T, http://www.uk.research.att.com/vnc

Last Updated: July 25th, 2017

Upcoming Training

SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC505: Securing Windows and PowerShell Automation	SEC505 - 201709,	Sep 18, 2017 - Nov 13, 2017	vLive
Secure DevOps Summit & Training	Denver, CO	Oct 10, 2017 - Oct 17, 2017	Live Event
SANS San Francisco Winter 2017	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	Live Event
San Francisco Winter 2017 - SEC505: Securing Windows and PowerShell Automation	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	vLive
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CA	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced