Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Case Study: Transforming a Traditional Windows Client/Server Application

Our software firm's financial application was developed on a traditional clientserver model. Individual user workstations run the application (on the Microsoft Windows Operating System) on a local area network. Our customer required that remote users from five locations across the country access the application over remote connectivity. They needed to provide an Application Service Provider (ASP) service with these sites accessing the application on central common hardware. It was critical that the individual locations...

Copyright SANS Institute
Author Retains Full Rights

AD

GIAC Certification GSEC Practical
David S. Strubbe
Version 1.4b Option 2 – Case Study in Information Security
June, 2003

Case Study: Transforming a Traditional Windows Client/Server Application Into a Secured ASP Offering

Abstract:

Our software firm's financial application was developed on a traditional client-server model. Individual user workstations run the application (on the Microsoft Windows Operating System) on a local area network against shared file, print, and database servers. Our customer required that remote users from five locations across the country access the application over remote connectivity. They needed to provide an Application Service Provider (ASP) service with these sites accessing the application on central common hardware. It was critical that the individual locations remain logically independent of each other.

Our financial application consists of millions of lines of code. It was not practical to rewrite it to operate effectively over a wide area network. Off the shelf technology, namely Citrix Metaframe and MS Terminal Server, was chosen to enable remote access to the application without major modification. Placing our application on Terminal Server and Citrix introduced new security concerns, as users no longer had dedicated workstations. Our application had resource requirements and security exposures that were a risk on shared hardware. We also had to consider the security of the network traffic to the remote users. This paper explores the process that we (the software vendor) and our client (the ASP provider) used to securely implement a solution.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Pre-Migration State

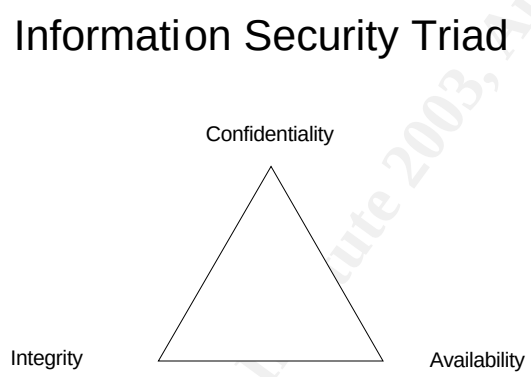
Overview of Application Needs

Our client wanted to securely provide our application to five distant offices using an Application Service Provider model. A fairly concise definition of Application Service Provider (ASP) is "a third-party software distribution and/or management service. Generally provides software via a wide area network from a centralized data center. [It] Allows companies to outsource and more efficiently upgrade software."[1]

Our client determined that it would be more cost effective to host our application centrally than to maintain a separate instance of the application at each office.

ASPs often service many different applications and offices on one platform. Although the users may share hardware and software, each client site's activities must be secured from the others. The information security triad of confidentiality, integrity, and availability of the application must be maintained across all users.

Figure 1. The CIA Triad[2]



Many ASP applications (including our application) are financial in nature. Serious financial losses could result from the release of private information, the inability to process transactions, or the malicious exploitation of the application into creating unintended transactions.

[1] ComputerUser.com. "High-Tech Dictionary." URL: http://www.computeruser.com/resources/dictionary/popup_definition.php?lookup=1580 (25 June 2003).
[2] Schwartau, Winn. "September 11, 2001 – Security Synergy." Information Security Magazine. November 2001. URL: http://www.infosecuritymag.com/articles/november01/industry_synergy.shtml (25 June 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Pre-Migration Application Overview

Our application is built on the traditional two-tier client server model. A 32-bit Windows client application runs on a dedicated end user workstation. A substantial portion of the processing takes place on this workstation. The client uses a database requester to communicate to a database server on a LAN, and there is a fair amount of network traffic travelling over the wire.

The application was never designed to work in wide area network. The amount of wire traffic precludes the ability to simply install the client on a remote client. The data stream from the remote client to the database server is often in plain text.

Our application also assumes that each user has substantial rights to a private workstation with their own unique environment (e.g. for temporary files, registry access, and file access).

There are millions of lines of Borland Delphi code to this application. As a result, re-coding the application for WAN access would require excessive resources, as well as extensive testing to confirm the proper port of the business logic.

ASP Model Application Threats to Consider

There are several threat vectors that we considered with the design of the ASP.

These included:

1. Authorized Application Users / External Clients attempting to cross into other client partitions.
2. Unauthorized Malicious Agents (External) attempting to access the system, inspect the data, or hijack a session.
3. Denial of service by authorized or unauthorized users (e.g. resource exhaustion, processor saturation).
4. Application Faults that cross to other application partitions (e.g. memory faults, buffer overruns).
5. Physical threats to the equipment (environmental, catastrophic).

Pre-Migration Application Technology Needs

The following table depicts the software used in our case-study financial application:

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 2. Current Application Technology

Item	Component	Platform
1	Client Operating System	Windows NT Workstation 4.0, Windows 2000 Professional, Windows XP Professional
2	Financial Application	32 bit Application consisting of 200+ executables located on a network share
3	Database requesters	Pervasive.SQL 2000i Client, MS SQL Server 2000 Client in addition with latest application release
4	Server Operating System	Windows 2000 Server (Pervasive supports Novell Netware, but this is not recommended due to a different authentication system from MS Windows – Novell NDS)
5	Database Management Systems	Pervasive.SQL 2000i, MS SQL Server 2000 in addition with latest application release
6	File Shares	Network shares must be used to share common files for a given instance of the application. Pervasive data files reside on shares.
7	Backup	Package backup solution (Backup Exec), Native MS SQL Server Backups

Overview of Proposed Technology Solution

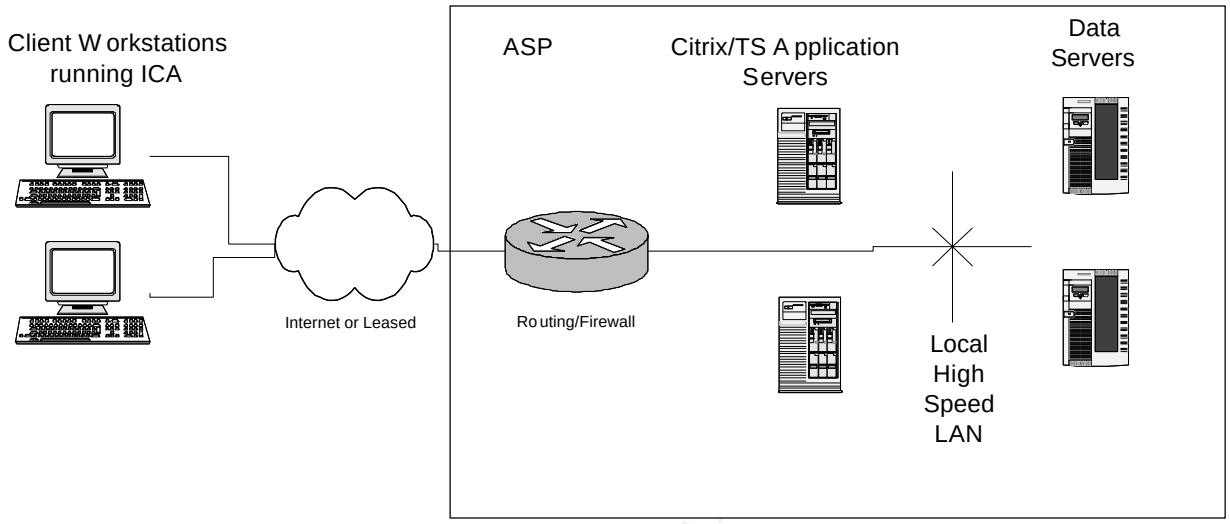
We recommended that the client use Citrix Metaframe as the foundation for their ASP. Citrix Metaframe[3], in conjunction with Microsoft Windows 2000 Server Terminal Services[4], can be used to provide thin client access to an application in a secure manner. Citrix also offers load-balancing services that allow for redundancy and improved application response.

All of the major processing would take place on the centralized platform, and only the presentation (screen input and output, mouse navigation, and printing) would need to travel across a wide area network or the Internet.

[3] Citrix, Inc. "Metaframe Access Suite." URL: http://www.citrix.com/site/PS/products/feature.asp?familyID=19&productID=186&featureID=3363 (25 June 2003).
[4] Microsoft, Inc. "Windows 2000 Terminal Services." URL: http://www.microsoft.com/windows2000/technologies/terminal/default.asp (25 June 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 3. Thin Client Solution



The installations of the software and respective databases needed to be logically partitioned to insure that users could only access their data without impacting other users.

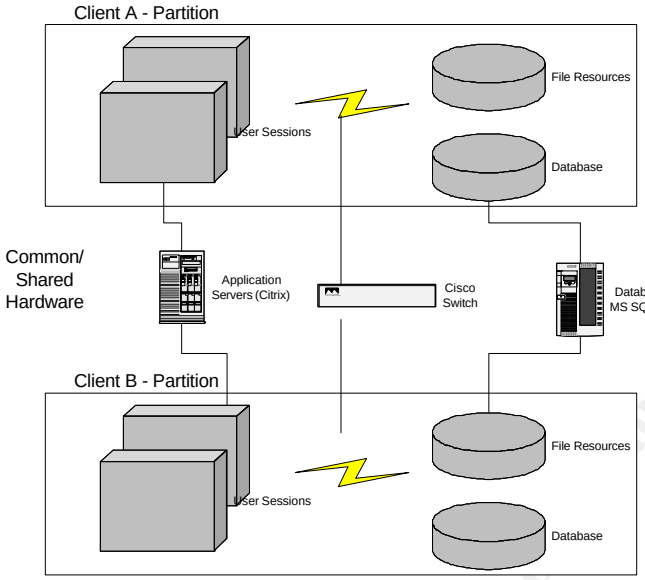
This “partition” concept is critical to a successful ASP. With this common hardware and software, there has to be an additional security layer between the overall platform and individual clients (sets of users).

Workstation and network access rights must be tuned according to the Principle of Least Privilege (PLP)[5] to prevent access to unauthorized data and denial of service to other users.

[5] Skoudis, Ed. Counter Hack. Upper Saddle River: Prentice Hall, 2002. 117-118.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 4. Logical Partitioning between Client Sites



The Citrix ICA Client was the thin client chosen to access the central ASP platform. MS Windows Security (i.e. domain accounts and associated rights) was used to apply access control to the file and database resources.

Connectivity to Citrix can be provided in a variety of modes. ASP Wide Area Networks generally implement TCP/IP connectivity using one of three options[6]

- Private – dedicated (fractional) T1 or higher
- Semi-Private – frame relay
- Wide Open – Internet (possibly in conjunction with a Virtual Private Network).

For additional security, Citrix traffic can be tunneled through a secure connection (e.g. a VPN), or Metaframe itself provides various modes of native support for encryption. We will examine the Citrix encryption options later in this case study.

Our client decided to utilize leased fractional T-1 connectivity with TCP/IP as the primary protocol.

[6] Anderson, Christa. "Pushing Applications to the Masses." Windows 2000 Magazine. August 2000, Vol. 6. p. 54.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

A Secure Migration Process – Step by Step

This implementation consisted of over two hundred users in five different locations with the need for five distinct instances of the application. We used our experience with similar configurations of a smaller scale to recommend a structured process to securely implementing this ASP model.

Figure 5. The Overall Process – Step by Step

1. Identify the customers and their specialized security needs.
2. Inventory the applications to be published by the ASP.
3. Analyze the application and modify it if required.
4. Provision hardware, software, and facility.
5. Provision secure connectivity.
6. Install the operating system environment.
7. Install the application.
8. Harden the configuration (application, rights, and authentication).
9. Test the application.
10. Deploy the application.
11. Maintain the ASP application (audit and update).

Step by Step – The Process in Detail

1. Identify the customers and their specialized security needs.

We had to evaluate the customer to determine if there where any specialized security needs. For example, health care related applications might need to comply with HIPAA. Financial firms may have record retention policies that they must adhere to.

For our application, it is essential that users have access to a minimum of two years worth of transactions, and service standards dictate that end of month full backup tapes must be retained indefinitely.

It must be determined whether the remote client, the ASP administrator, or both will be granted system administrator rights. Our client decided to retain all administrative functions at the central site.

We also had to consider the application rights. Our financial application has its own operator and rights database that is in addition to the operating system

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

operator database. We had to assist with creating five distinct and separate user databases for the five separate user sites.

2. Inventory the applications to be published by the ASP.

The client may identify a core application (such as our financial application) as the primary application. Often additional applications may be required in addition to the core application. These applications are easy to forget, and can be lost in the planning process.

For example, our financial application also requires Seagate Crystal Reports and Microsoft Access.

Some of these applications could potentially expose data if rights are not properly secured. For example, Seagate Crystal Reports makes a great tool to use to inspect unauthorized databases (if the database access rights are not properly configured).

3. Analyze the application and modify it if required.

To develop an ASP, you must examine the application for resource requirements. These requirements include client software, registry rights, file rights, and application authentication needs. This information is critical to tuning the rights on the multi-user terminal servers. For example, you cannot simply assume that each user has their own temporary directory (e.g. C:\winnt\temp) on their own PC.

In particular, you must examine the application for functionality that can be exploited to compromise security. This process is often very manual. One large ASP provider admits, “humans do all the work – Push hasn’t found any automated tools that work as well as an engineer.”[7]

Regmon and Filemon are two useful tools that we used to analyze our applications for an ASP model.[8] They show what resources are accessed, as well as the nature of the access (read versus write). They allow systems analysts to review necessary application file and registry activity in an effort to minimize resource rights.

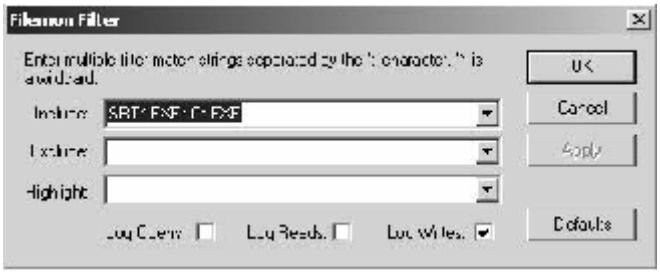
For example, when we hardened the rights to the Terminal Server \WINNT\TEMP directory, the reporting functions of our financial application would often fail. Filemon allowed us to see that the Borland Database Engine (BDE)

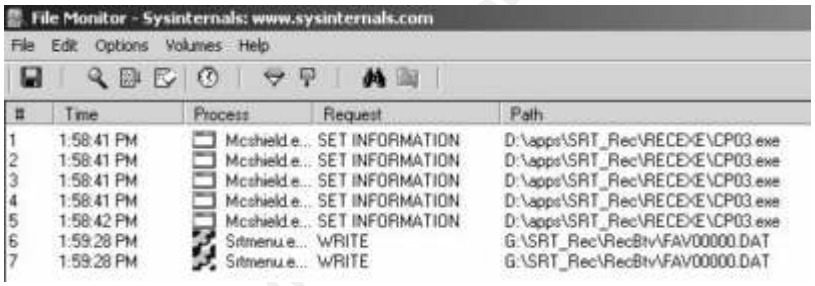
[7] Anderson, p. 57.
[8] "Utilities for Windows NT/2K" URL: http://www.sysinternals.com/ntw2k/utilities.shtml (14 May 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

was attempting to write out a temporary file to this directory. After expanding the user rights, the reporting function worked properly. The temporary files did not have any data that would place confidential information at risk.

Figure 6. FILEMON and REGMON Filters, and a network share write caught from our application



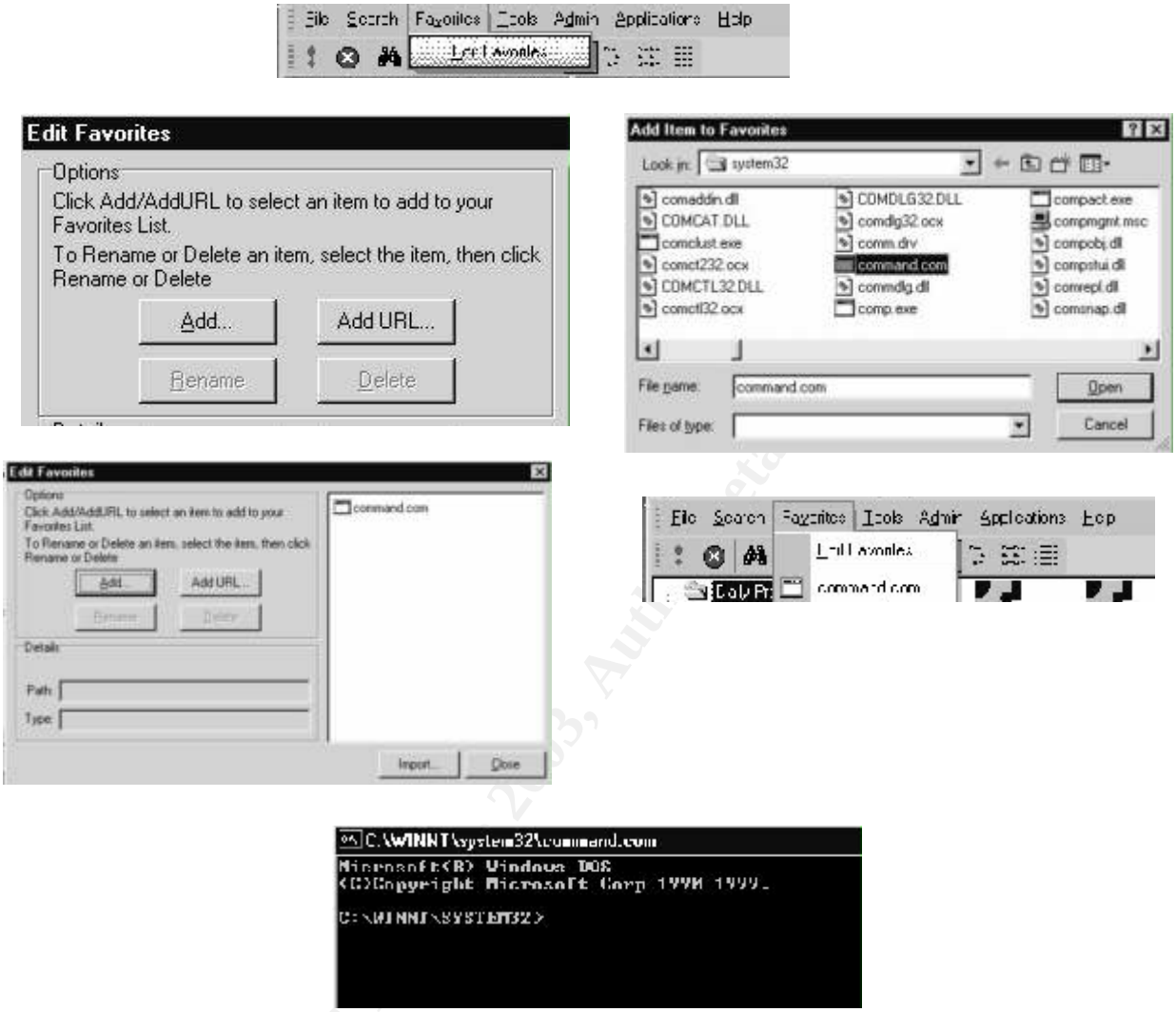


Applications may contain functionality that poses a risk to security. For example, an application may be used to launch additional executables. An application may also have resource connection dialogues that can be used to probe for additional databases or accounts. The application may need to have these functions removed or disabled.

Our application has a “favorites” functionality that can be exploited. Although the ASP may remove desktop shortcuts to applications, this function could still allow a user to find and run an application. We advised the client of this risk, and we plan to allow our customers to disable this feature in a future release.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 7. Demonstration of a unauthorized command shell from our financial application



Of particular concern should be any user interface that allows the adjustment of database connection resources or authentication strings. A malicious user could use these dialogues to attempt to change databases or authenticate to unauthorized resources. Our application has an initial login screen that allows the user to adjust database connection strings.

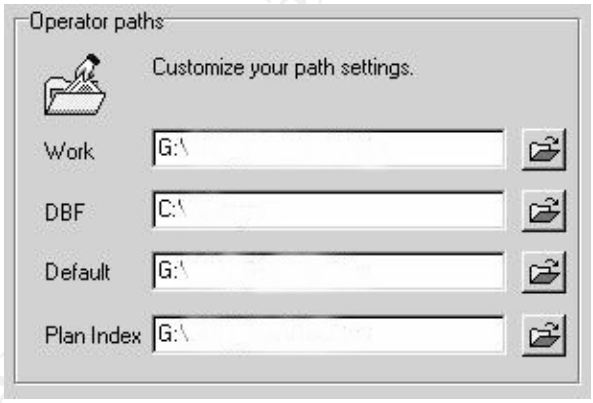
© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

To assist with such exposures, we added command line startup functionality, which allows these parameters to be hard coded into an application startup script. The user never sees any database connection parameters.

Figure 8. Example of a User Interface Configuration Exploit – Database Connection Parameters

While implementing our application on the ASP platform, we had several working directories that required careful management. If they had not been secured with unique drives and directories, users may have inspected each other's temporary data, or overwritten each others files causing a denial of service.

Figure 9. Work Directories Managed with Private Root Drive Mappings



The following table summarizes the primary application functionality issues that we surveyed with our application in an ASP environment:

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 10. Summary of Application Functionality Concerns with our Financial Application

Item	Threat	Description	Impact	Resolution
1	Cross Partition Temporary or Work File Access.	Temporary work files are visible across users, temporary files remain after logout, or users attempt to write to the same workspace.	Confidential data may leak across users.	Use scripting, application server rights, and drive mapping to insure private work areas.
2	Execution of arbitrary commands from functionality .	Application allows pointing to command shells, registry editors, or other system commands.	Compromise system integrity or data confidentiality, and possible denial of service.	Modify the application to remove such functionality. Use policies to restrict execution of dangerous applications like REGEDIT.
3	Conflict of user work space or registry settings.	Application may assume that workspace and registries are private. E.g. Current User versus Local Machine Registry Settings.	Inadvertent denial of service if users compromise other user environments.	Modify the application, and script essential settings at application startup.
4	Access to database, host connection, and login parameters.	Malicious end users may vary or attempt cracking into additional resources.	Data confidentiality and integrity could be compromised.	Script connection strings and settings into application startup. Remove these settings from end user interface.
5	Work Files of Excessive Size.	Temporary Files and Work Files exhaust application server resources.	Denial of Service.	Institute disk quotas.

4. Provision hardware, software, and facility.

The hardware must be sized to adequately support the quantity of users for the application. For our application, our experience has demonstrated that approximately fifteen users can be supported on a single dual processor server with two gigabytes of memory. Our application ASP client provisioned twelve separate servers for their 5 site / 200 user installation base.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The physical facility (power, environment, and access) should also be secured and redundant. Our client had an enterprise level data center with electronic access control, redundant power, and redundant cooling systems for their central ASP platform.

Citrix also offers software add-ons that can assist in securing the ASP environment.

Our client opted to purchase both the Citrix Load Balancing and Resource Management features. Metaframe XP Advanced includes Load Balancing.[9] This feature provides for the automatic failover to additional servers if a server should fail (e.g. a denial of service attack exhaust all processing power of a given server). Metaframe XP Enterprise includes Resource Management for monitoring use of storage, CPU and memory.[10]

Both of these features help the client to secure their ASP. If a malicious user should cause an application server to fail, the load balancing services will distribute additional users to the remaining functioning servers.

Resource monitoring allows our client to closely inspect the memory, process, and processor usage on the application servers. If a user were to introduce a rogue process, the resource monitoring would inventory this process, as well as any suspicious disk, memory, or processor usage.

In a highly secure environment, an ASP could even consider separate hardware for each of their clients as another layer of partitioning.

The following table summarizes the primary physical and hardware issues that we considered for our application as an ASP:

Figure 11. Hardware, Software, and Facility Considerations

Item	Threat	Description	Impact	Resolution
1	Physical Attack on the Server.	Malicious agents destroy or steal equipment.	Denial of service and possible data theft.	Secure facilities and recovery measures.
2	Rogue or runaway tasks.	Processor or Disk Resources are exhausted for unusually or unauthorized tasks.	Denial of Service.	Use Resource Management to watch for unusual or processor exhausting tasks; use load balancing to fail over to additional servers. Provision an adequate number of servers.

[9] Mathers, Todd. Windows NT/2000 Thin Client Solutions. Indianapolis: New Riders, 2000. 110-111.
[10] Mathers, pp. 592-594.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

5. Provision secure connectivity.

As clients of an ASP are often located vast distances from the data center, ASP providers must insure that the connectivity to their remote clients is available and secure.

Three connectivity aspects should be considered:

- Security of the activity transported between the ASP and the remote user
- Security of the ASP from the hostile Internet
- Security of the applications within the ASP itself.

For secure connectivity an ASP should consider a virtual private network (VPN) or native Citrix Encryption. VPNs, although more flexible, could pass various types of traffic, and increase the threat from end users to the ASP. Using native Citrix encryption reduces the breadth of the remote threat to Citrix traffic only.

Citrix SecureICA services can use the RSA RC5 algorithm with up to a 128 bit session key. You can also force remote ASP users to connect with a minimal length key.[11]

Citrix also offers the Citrix SSL Relay and Citrix Secure Gateway. These add-ons allow Citrix to use SSL 3.0 for connectivity.[12]

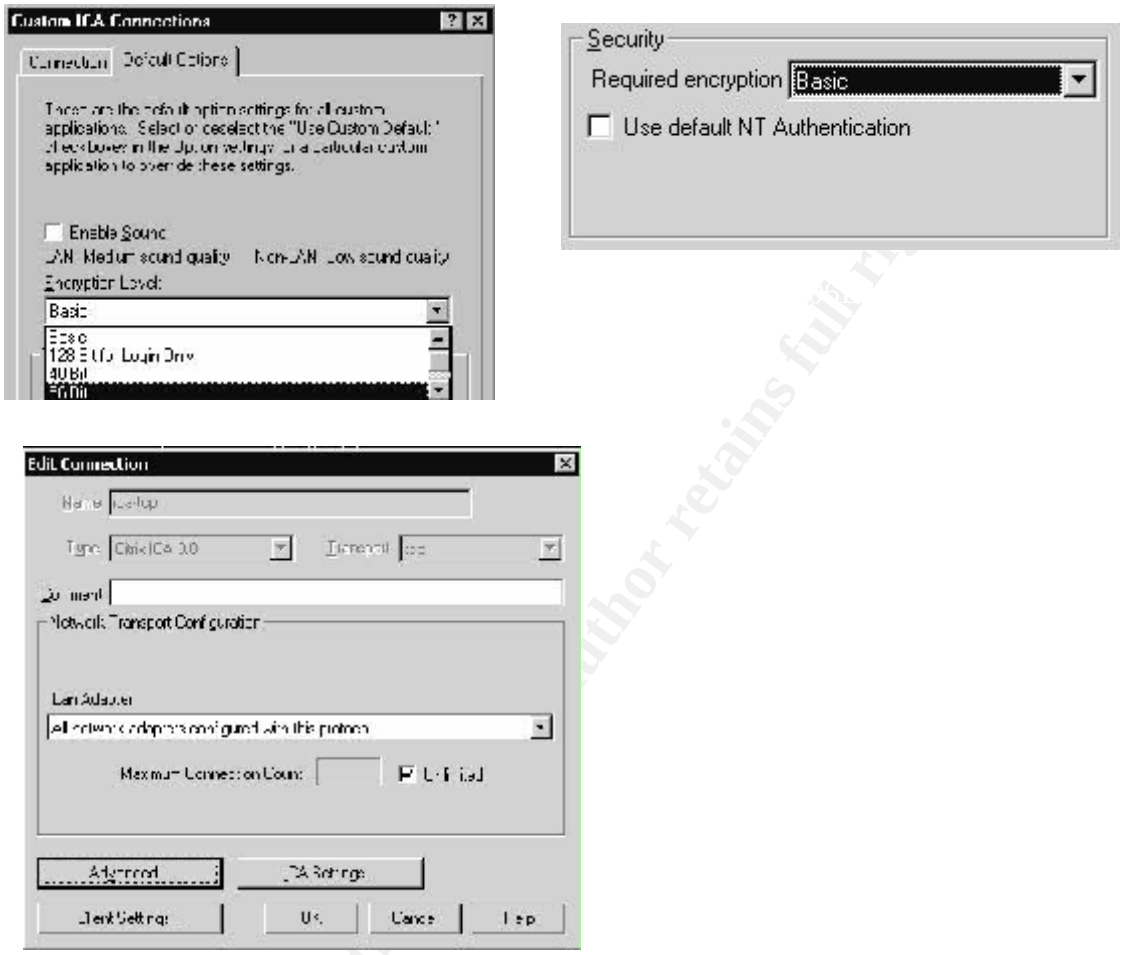
If possible, edge routers and firewalls should be adjusted to only pass Citrix traffic to client sites. This includes filtering based on ports and addresses.

The client in this case study opted to use dedicated leased connectivity from the five remote offices to the central office. As a result, they opted to simply use the basic encryption capability of Citrix Metaframe. We have had several clients utilize VPN technology with end users that telecommute from home.

[11]Citrix, Inc. "Configuring ICA Encryption." Metaframe Books Online. Citrix Metaframe 1.8 for Windows 2000.
[12] Citrix, Inc. "Citrix Secure Gateway 1.1." 27 May 2002. URL: http://download2.citrix.com/ctxlibrary/products/pdf/Citrix_Secure_Gateway_Datasheet.pdf (29 June 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 12. Native Citrix Encryption Capability Options in Citrix Configuration



We also recommend that our clients disable the Microsoft Remote Desktop Protocol (RDP) and leave only the Citrix ICA Protocol enabled on the terminal servers. This is consistent with the general security recommendation to disable unnecessary services.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 13. Summary of Connectivity Considerations

Item	Threat	Description	Impact	Resolution
1	Malicious Attackers on the Internet.	Malicious users probe and attack the ASP.	Confidentiality of the data may be compromised, or denial of service.	Utilize a DMZ Model, Implement the appropriate Access Control Lists on Routers, Utilize Intrusion Detection.
2	Attack on the ASP.	Physical or network based attack on the ISP used by the ASP.	Denial of Service.	Utilize two separate ISPs with Failover.
3	Traffic Snooping.	Malicious Agents sniff traffic for data or account credentials.	Confidentiality compromised.	Use a Virtual Private Network or Citrix Encryption.
4	Non-Citrix Traffic traversing the Internet.	The DMZ model may permit malicious traffic other than ICA traffic to or from the ASP.	The ASP may be attack, or serve as a host to attack others.	Secure the DMZ to permit traffic only to the ASP on TCP port 1494[13]. Limit source IP addresses to known business partners.

6. Install the operating system environment.

For our financial application, the required operating system components included the following software:

- Microsoft Windows 2000 Server Operating System
- Windows 2000 OS Patches (SP2 or SP3)
- Compaq/HP Insight Management
- Metaframe XPe for Windows 2000
- Network Associates Antivirus
- Backup Exec Backup Agents

We recommended that all disks be formatted with the NTFS file system. This allows the appropriate security and usage quotas to be applied to the system.

We also recommended that the support for non-essential OS services and applications (e.g. the Microsoft Internet Information Server) should be removed from both the application and data servers.

[13]Reece, Thomas. "Citrix ICA Perimeter Security Issues." URL: http://www.giac.org/practical/Thomas_Reece_GSEC.doc (29 June 2003): 2.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Note that the operating system environment not only includes the operating system, but applications that complement the security capabilities of the operating system (such as anti-virus software and backup software).

Some ASPs may want to implement a host based intrusion detection system (such as Tripwire[14]) to proactively detect unauthorized system changes.

7. Install the application.

Installing an application within an ASP environment is more complicated than installing on typical workstations.

An inventory of all installed components and locations had to be compiled. Any files shared between user partitions had to be read-only. This was to prevent any leaking of information or introduction of malicious software across user partitions.

We recommended to the client that our application startup be completely scripted. This can be accomplished by scripting tools such as Windows Scripting Host, Winbatch, KixTart, or a simple batch file. The script should guarantee that the software environment (search path, drive mappings, registry entries, and launched executables) are exactly what the application needs. If something is changed (e.g. maliciously or by accident), the scripting insures that the appropriate setup returns.

For this instance of our application, a simple batch file was used to start the application.

Figure 14. Example Script for the Case Study Application

```
rem this batch file is used to start up the application
rem 4/2002

rem optionally add regini.exe command here


rem map the drive
rem use q: \\server2\apps

rem set the current directory
c:
cd\app_rec\

rem start up sr.menu
start q:\app_rec\exe\menu

rem exit this script
exit
```

[14] Tripwire, Inc. “Tripwire for Servers.” URL: http://www.tripwire.com/products/servers/ (29 June 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

To present the application in Citrix, the application is “published” out to the authorized users. Generally, the ASP must publish the script that launches the application. In a load-balanced environment, the application must be installed on all application servers.

With Citrix, you can publish a complete desktop or a single application. For our application, we recommended publishing our financial application in a seamless window. As a result, the user only sees the application that they wish to run, and they cannot easily access any additional desktop functionality that could compromise the application server.

Figure 15. Desktop (showing programs) versus Seamless Published (with Notepad)

Installed software should be minimized to the essential components that are required to run the application. For our financial application, we recommended

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

that the Pervasive and Microsoft database tools should be removed or restricted. They present powerful tools to malicious users attempting to exploit the system.

Figure 16. Default Client Options for Pervasive, MS Platforms that should be Removed

The following table summarizes the critical items when installing our application in an ASP environment:

Figure 17. Summary of Application Installation Issues

Item	Threat	Description	Impact	Resolution
1	Environment is Changed.	Paths, environment variables, or registry entries are maliciously modified.	Possible denial of service or breach of confidentiality.	Script the startup of the application to insure the proper environment. Secure and protect the startup script from modification.
2	Unauthorized application access.	Malicious users may try to run administrative tools or other applications outside of the authorized ASP application.	Possible denial of service or breach of confidentiality.	Publish only the application that the users should see. Remove powerful and unnecessary software tools such as database utilities.

8. Harden the configuration (application, rights, and authentication).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

To properly secure our application within an ASP, the client must take full advantage of the Microsoft Windows security model. All operating system and Citrix security features should be used to enforce the Principle of Least Privilege to ASP users.

We recommended that the following rights should be minimized:

- Rights to the application servers running Citrix
- Rights to the database servers housing application data
- Rights to databases themselves
- Rights to any file shares used by the application

It is critical that the databases, temporary work space, and file shares used by different ASP customers are securely partitioned between each other. Under no circumstances should system rights allow customers to cross over and see or modify another customer's data.

Typical end users should not obtain rights beyond the Windows 2000 Users Group, which is more restrictive than the Windows NT 4.0 Users Group.

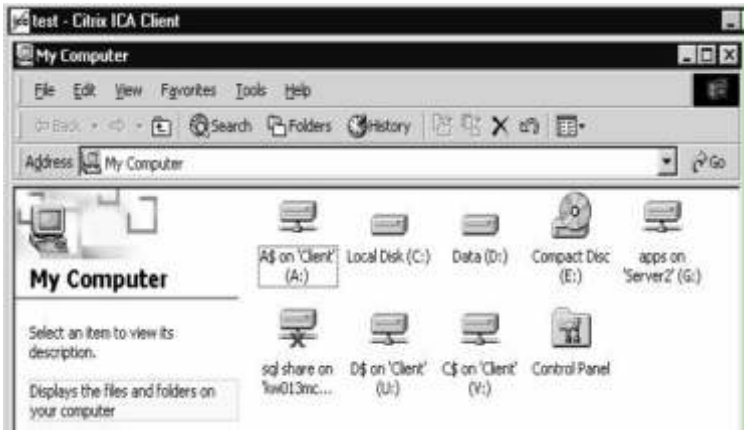
The Citrix integration with Windows Security (Domain or Active Directory) should be used to publish applications only to users that are authorized.

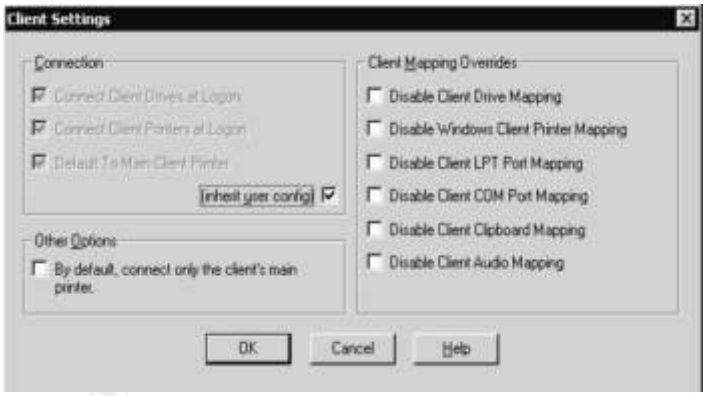
In the example application, each client has their own directory tree on the data server that is secured to the respective client via Domain Rights. Pervasive transactional files (Btrieve files) are simply secured by the operating system rights to the data files.

There are several additional Citrix and Windows 2000 settings that are essential to hardening an ASP application Server. Citrix has the ability to share back resources from the remote client to the application server. This can include the Windows clipboard, local disk systems, and printers. For our financial application, remote printing was enabled to provide the functionality back to the end users. We advised the client of the benefits and risks of clipboard and disk system sharing. For example, a sharing loop back to a remote workstation could increase the risk of accessing mal-ware located at the remote workstation.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 18. Sharing Back of Resources, Metaframe Configuration Option to Disable



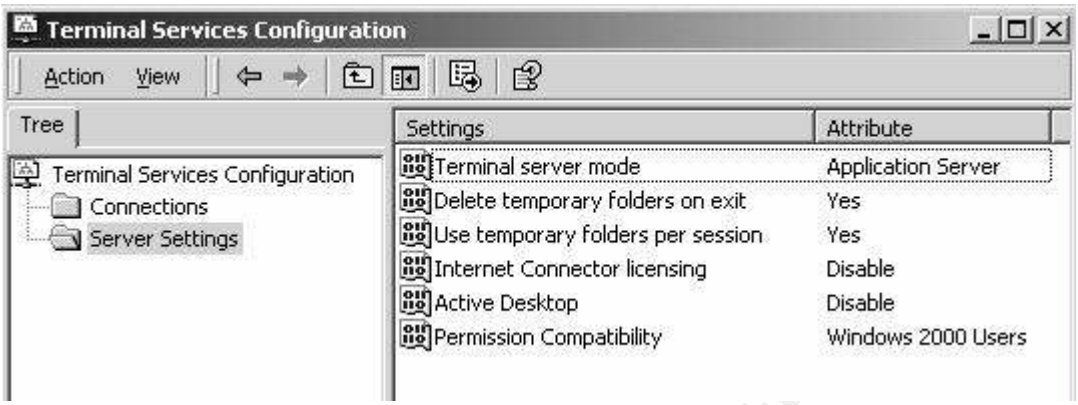


Users of a Windows system often have work files and directories stored in a temporary folder. Our financial application generates temporary work files from Crystal Reports and the Borland Database Drivers. These files and directories may contain sensitive information that could be compromised if later users attach to the same work directory. Terminal Server has an option, which forces the cleanup of these work directories at the end of a session. [15] We recommended enabling this option.

[15] Microsoft Support. "Windows 2000 Server Documentation." 28 February 2000. URL: http://www.microsoft.com/windows2000/en/server/help/default.asp?url=/windows2000/en/server/help/ts_con_ss_020.htm (29 June 2000).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 19. Citrix Configuration to Delete Temporary Folders

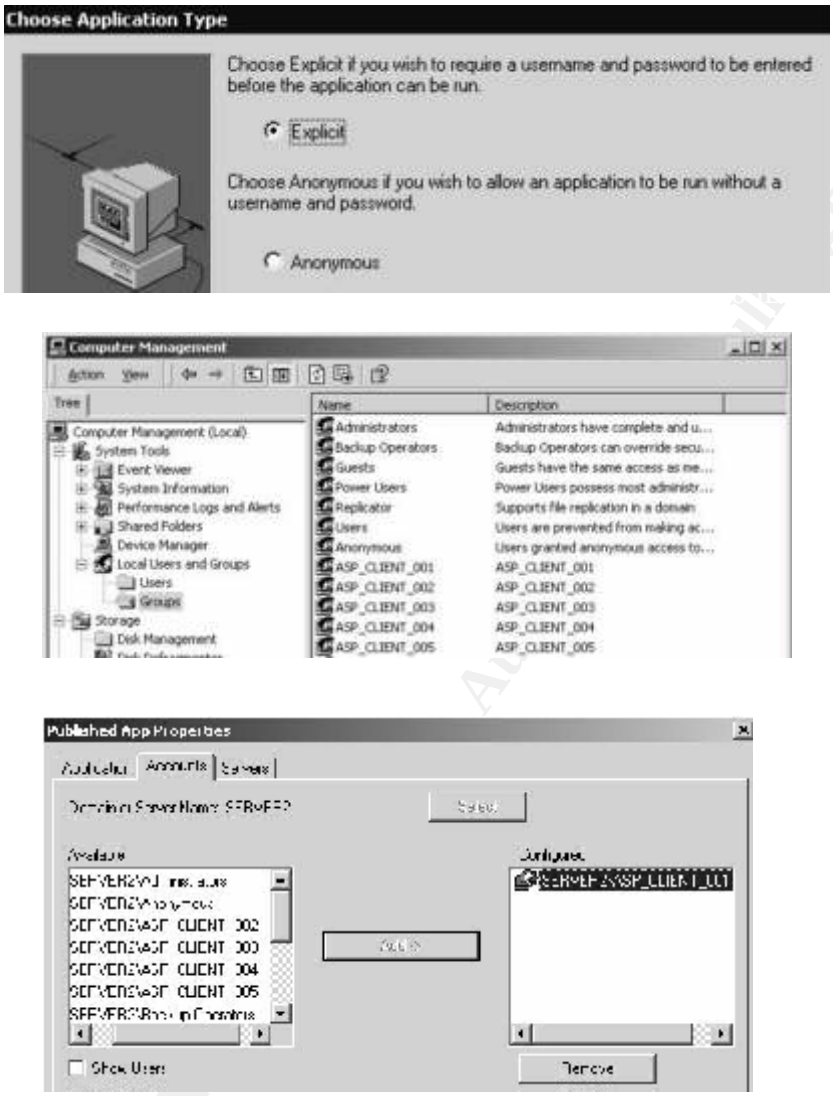


It is critical that Windows Security be used to partition different clients from each other. Citrix, Microsoft SQL Server 2000, and Pervasive / file sharing security all integrate to Windows. For our financial application, Windows groups were set up for each respective client. Discrete user accounts were then created for each individual user. Each individual user is assigned to one and only one client group. The resources for that client are then attached to that client group.

For this model to work, we recommended that the applications be published in Citrix Explicit Security mode (versus Anonymous). This requires the end user to enter Windows account credentials to access the application.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

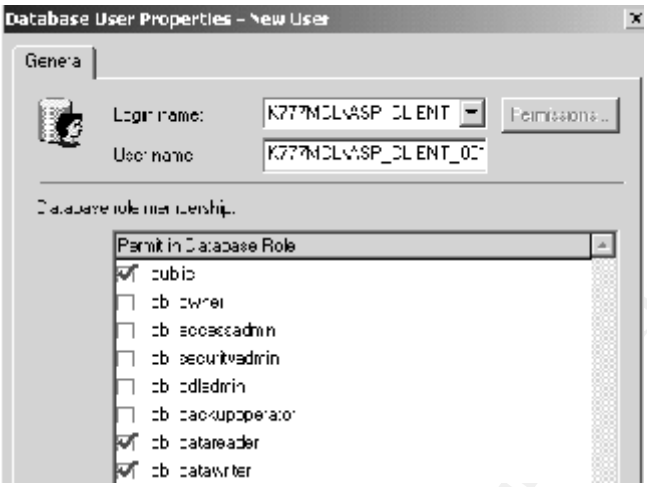
Figure 20. Setting Application Publishing Characteristics, Windows Groups, and Assignment of the Application Instance



After securely structuring the user groups, these Windows groups can be used to secure the Citrix publishing, application server rights, and file share rights. They also should be used to secure database access. Our Pervasive databases are secured via file share rights. Our current product revision now integrates with MS SQL Server 2000, and we can leverage MS SQL Server Windows integrated security to the respective client database.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 21. Example of Using Windows Groups to assign an ASP Client Group to a MS SQL Database with Minimal Database Rights



The following table highlights the critical steps that we considered while hardening the setup of our application for the ASP:

Figure 22. Summary of Hardening Steps

Item	Threat	Description	Impact	Resolution
1	Cross Partition Resource Access.	Users access files or databases of other customers.	Confidentiality.	Utilize Windows Security to restrict access to databases and published application. Model Windows groups after the customer base. Minimize user rights attached to these groups.
2	Introduction of Mal-ware.	Malicious Users introduce trojan or other harmful executables.	Confidentiality or Denial of Service.	In addition to virus scanning software, disable client disk redirection in Citrix.
3	Leak of Temporary Work.	Temporary files remain from previous sessions, and are viewable by other clients.	Confidentiality.	Enable Citrix to delete temporary directories on end of session.
4	Resource Exhaustion.	Users fill up drive volumes (maliciously or inadvertently).	Denial of Service.	On NTFS volumes on application servers, activate Disk Quotas.[16]

[16] World of Windows Networking. "Windows 2000 Disk Quota." 2002. URL: http://www.wown.com/j_helmig/w2kdiskq.htm (29 June 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

9. Test the application.

After hardening the application platform, it is critical to test it.

The application should function as expected. Hardening may break some of the functionality, and the rights or application may need to be adjusted.

While our client initially tested their ASP, some of the functionality did not work. We found that we had to open up the rights to certain areas of the registry, as well as certain temporary directories on the application servers. As stated earlier, Regmon and Filemon are invaluable for this process.

It is critical that an end user is not able to access the files, applications, or databases of other customers outside of their application partition. Engineers should test for any such exposures.

Vulnerability Scanners such as Nessus[17] may be used to verify system exploits visible from within the DMZ, from the outside customer sites, or the Internet. Since our client decided to use secure leased connectivity for the remote connectivity, they did not utilize any vulnerability scanners in their testing.

10. Deploy the application.

Once tested, the ASP may begin to deploy the application into the field. The ASP must assist their remote offices to install ICA Clients on the remote clients.

There are alternative models for deployment from Citrix, such as the Nfuse web based front end or Embedded Clients (e.g. Java based from a browser).

For this ASP installation, the full Citrix ICA Client was deployed to the remote workstations. This client has had a better track record with out application than the embedded browser clients or Nfuse. The remote users simply had to download the ICA client install, run the setup, and set the host connection properties.

11. Maintain the ASP application (audit and update).

Once it was deployed into the field, our client has had to actively manage their ASP platform to keep it secure. Critical activities they conduct include:

- Audit of the platform activity, including Citrix Servers and Database Servers.
- Regular Testing of the Backup and Recovery Solutions.

[17] Deraison, Renaud. "Nessus." 4 June 2003. URL: http://www.nessus.org/ (29 June 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

- Verification and application of Operating System, Citrix, and Database Server Patches.

Some additional steps that we suggest for a public connectivity ASP include:

- Regular Vulnerability Scans with an appropriate scanner.
- Audit of the IDS and Firewalls.

Tools such as the Microsoft Baseline Security Analyzer[18] and resources such as the BugTraq mailing list[19] are useful references for managing Terminal Server and Citrix exploits.

We recommend that an incident response team and written response plan should be assembled to address any potential system compromises that occur.

Appropriate off-site facilities (tape storage, and potentially hardware and connectivity) should be obtained to mitigate complete physical loss of the ASP site. For this case study, our client had a separate data center in another city to which they regularly shipped the backup tapes. There was a smaller hardware platform available for recovery if needed at the second site.

For our case study client, many of the details for recovery and support were agreed upon in a SLA (Service Level Agreement) between the ASP and the remote offices. This document was key to agreement on many of the important features of the ASP offering, including some of the security policies and procedures.

[18] Microsoft, Inc. "Microsoft Baseline Security Analyzer." 2003. URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/Tools/MBSAhome.asp (29 June 2003).
[19] SecurityFocus, Inc. "Vulnerabilities by Vendor." URL: http://www.securityfocus.com/bid/vendor/ (30 June 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Summary and Conclusion

Deploying an application as an ASP has special security considerations. The connectivity between the ASP and remote client must be secure, reliable, and monitored.

Co-locating applications for different customers on common hardware requires an additional level of security that is not typical for a traditional Windows application. Different clients must not see each other's data. There must be a secured logical partition for each client instance of an application.

Many features of Citrix Metaframe and MS Terminal Server assist with managing resources and connectivity security. Often Microsoft Windows' security can be effectively utilized to secure each customer's data properly.

Our application was never intended for an ASP on multi-user platform. By engaging us in a consultative role, we were able to help analyze and retrofit our application to securely function on multi-user application servers. We also helped advise our client on some of the security exposures, as well as some of the platform configurations that could be utilized to mitigate these risks.

Once deployed, our client's ASP platform has required active management to insure that the applications remain accessible and secure to the remote clients.

To date, we have had many other clients implement similar platforms with fairly good success. As outsourcing becomes more popular and IT expenditures decrease, we project that the ASP model of deploying our application will continue to become more popular.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

List of References

Anderson, Christa. "Pushing Applications to the Masses." Windows 2000 Magazine. August 2000, Vol. 6: 54, 57.

"Citrix Secure Gateway 1.1." Citrix.Com. 27 May 2002. Citrix, Inc. 29 June 2003. <http://download2.citrix.com/ctxlibrary/products/pdf/Citrix_Secure_Gateway_Datasheet.pdf>.

"Configuring ICA Encryption." Metaframe Books Online. CD-ROM. Fort Lauderdale: Citrix, 2000.

Deraison, Renaud. "Nessus." Nessus.org. 4 June 2003. 29 June 2003. <http://www.nessus.org/>.

"High-Tech Dictionary." ComputerUser.com ComputerUser.com, Inc. 25 June 2003. <http://www.computeruser.com/resources/dictionary/popup_definition.php?lookup=1580>.

Mathers, Todd. Windows NT/2000 Thin Client Solutions. Indianapolis: New Riders, 2000. 110-111, 592-594.

"Metaframe Access Suite." Citrix.com. Citrix, Inc. 25 June 2003. <http://www.citrix.com/site/PS/products/feature.asp?familyID=19&productID=186&featureID=3363>.

"Microsoft Baseline Security Analyzer." Microsoft.com. 2003. Microsoft, Inc. 29 June 2003 <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/Tools/MBSAhome.asp>.

"Windows 2000 Server Documentation." Microsoft.com. 28 February 2000. Microsoft, Inc. 29 June 2000. <http://www.microsoft.com/windows2000/en/server/help/default.asp?url=/windows2000/en/server/help/ts_con_ss_020.htm>.

"Windows 2000 Terminal Services." Microsoft.com. Microsoft, Inc. 25 June 2003 <http://www.microsoft.com/windows2000/technologies/terminal/default.asp>.

Reece, Thomas. "Citrix ICA Perimeter Security Issues." Sans.org. SANS Institute. 29 June 2003 <http://www.giac.org/practical/Thomas_Reece_GSEC.doc>.

Russinovich, Mark and Cogswell, Bryce. "Utilities for Windows NT/2K" Sysinternals.com. Wininternals, Inc. 14 May 2003. < http://www.sysinternals.com/ntw2k/utilities.shtml>.

Schwartau, Winn. "September 11, 2001 – Security Synergy." Information Security Magazine. November 2001. 25 June 2003. < http://www.infosecuritymag.com/articles/november01/industry_synergy.shtml>.

Skoudis, Ed. Counter Hack. Upper Saddle River: Prentice Hall, 2002. 117-118.

"Tripwire for Servers." Tripwire.com. Tripwire, Inc. 29 June 2003. <http://www.tripwire.com/products/servers/>.

"Windows 2000 Disk Quota." World of Windows Networking. 2002. J. Helmig. 29 June 2003. <http://www.wown.com/j_helmig/w2kdiskq.htm>.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

"Vulnerabilities by Vendor." Bugtraq. SecurityFocus.com. 30 June 2003. <http://www.securityfocus.com/bid/vendor/>.

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 23rd, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced