Defense in Depth. Securing an application dependant on Microsoft Transaction Server.

Karri Kanerva

April 29, 2002

Assignment version 1.3

Executive Summary

Selling products on the Internet have been one of the most lucrative businesses recently. We have seen many of them come and go.
It doesn't require lot of effort to put up a few servers with a simple application and start doing business on the Internet.
However, Internet is a hostile environment. The majority of the people out there don't intend to hur t your systems. But a very large number of hackers or script-kiddies really do. These guys are not hard to stop, but it's basically up to you as a systems administrator to make it so.
This document is intended to help other people in the Security Community when securing a Microsoft Transaction Server dependant application accessible from the Internet. It is also intended to explain the importance of the "Defense in Depth" principle in order to gain as much security as possible by adding layer by layer of pr otection.
Throughout the document I will build a scenario where we add layer after layer of protection so the application can run in a secure way.

Scope

This document is mainly considered for Security System Administrators that use Microsoft Transaction S erver applications accessible from the Internet, and who are concerned about how to secure such a critical environment.

Structure

Application overview

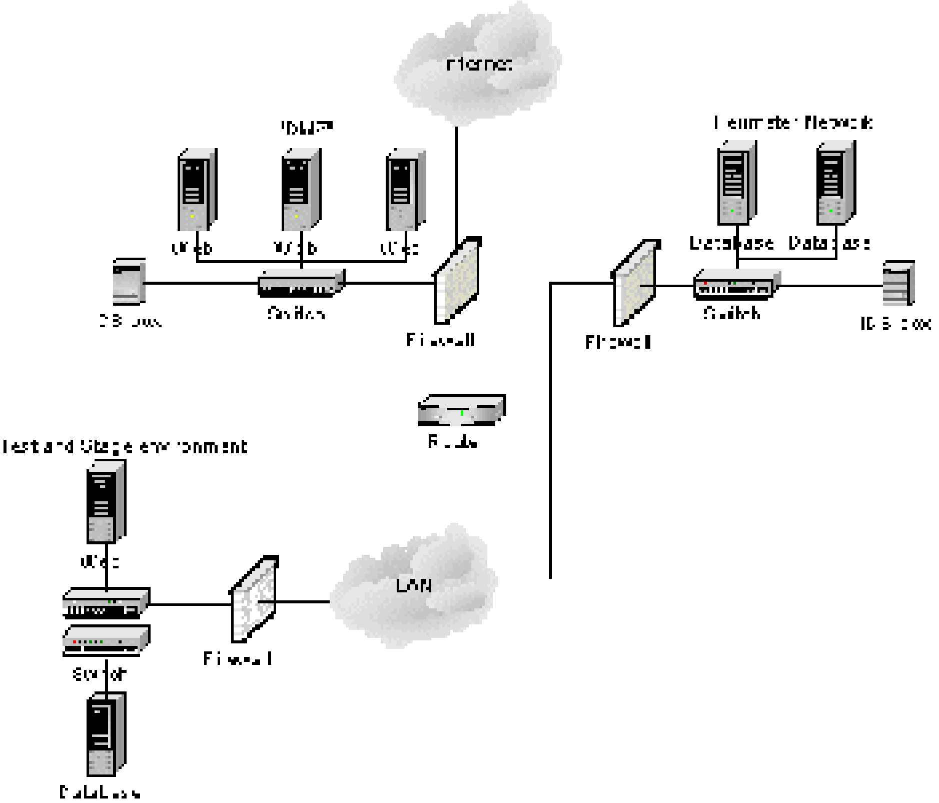
The application is hosted on two servers. One front -end server with ASP Web application communicating via DCOM objects to second server, which is a database server containi ng application data. Application workflow is simple.

1. Customer accesses the web servers ASP pages and fills in forms with the customers purchase.
2. The application uses DCOM objects, which utilizes Transaction Server passing the information to the database server, which will store the purchase for later handling.

The web server is installed with Microsoft Windows NT Server 4.0 with Internet Information Server 4.0. Bundled within IIS 4.0 is also Microsoft Transaction Server 2.0.
The database server is instal led with Microsoft Windows NT Server 4.0 with Microsoft SQL Server 7.0.
Both servers have Service pack 6a installed.

Network architecture

Picture1.



The network architecture consists of five major elements. Internet, DMZ (Demilitarized Zone), Perimeter Network, LAN and the Test and Stage environment.
The Internet really doesn't need any closer description. The DMZ protected with a firewall has three interfaces, one for connection to Internet, one for DMZ network and one for LAN connection.
Hosted inside the DMZ are the Web servers located on a VLAN switch. An IDS box is also located at this network to sniff on network traffic.
The Perimeter Network consists of a Firewall with two interfaces, one for connection to LAN and one for network behind the firewa II.

A VLAN switch is used for the network here as well. The second IDS box is installed in this network.
The LAN will resemble the normal corporate network environment. It can be just a few servers and PC's but could also be a huge network with hundreds or thousands of connected servers or PC's.
The Test and Stage environment is a replica of both DMZ and Perimeter Network environments. A Firewall protects the networks and is equipped with three interfaces; one for LAN connection, one for DMZ replica network and one for Perimeter replica network.
Both networks use VLAN switches.

Host security

One of the most important things to remember when implementing Defense in Depth is the host hardening. We painfully noticed how important this really is during the outb reaks of CodeRed and Nimda incidents. Both CodeRed and Nimda exploited known vulnerabilities in IIS 4.0 and IIS 5.0. It is of utmost importance to lock down the systems, both the operating system and the applications running on the systems. I will not describe in detail on how and what to do when tightening the security on systems. There are already many good guidelines available on the Internet. I will however point you to the ones I really think are good.

- The first comes from the US. National Security Ag ency and is a very comprehensive guideline covering the hardening of Windows NT. http://nsa1.www.conxion.com/winnt/guides/wnt -securityguides.zip
- Microsoft has also contributed with many guidelines that also are good to run through. This one covers the hardening of the Windows NT 4.0 Server: http://www.microsoft.com/technet/security/tools/chklis t/nt4svrcl.asp
- The next two covers the hardening of IIS 4.0 and using the Microsoft IIS Lockdown tool: http://www.microsoft.com/technet/security/tools/chklist/iis4cl.asp , http://www.microsoft.com/technet/security/tools/tools/locktool.asp

However, it doesn't do any good if you harden the systems if they have known vulnerabilities. Make sure to patch the systems with the latest patches and post SP6a security roll -up patches.
Make sure you have a backup strategy in place before you start the hardening process. Locking down a system to a point where it wont start is very likely to happen if you don't pay attention to exactly what you are doing. It is vital to apply the latest patches to these systems.

- Microsoft has a tool for keeping track on what patches are installed on system and should be used on a regular basis. http://www.microsoft.com/technet/security/tools/tools/hfnetchk.asp

The DMZ environment needs special attention. This is the most critical environment. This server is accessible from the Internet, whic h means that the only thing stopping intruders is the Firewall and its rules. The

web server here must have an absolute minimum of functionality available after the hardening process has been completed.
The IIS server also needs to be forced to use TCP protocol instead of NetBIOS, which is default in server installations. This is done by SQL Client Configuration utility, available at SQL Server 7.0 server CD. Install and configure the connection to SQL server and make sure TCP is used as default protocol.
Web server should have SSL (Secure Socket Layer) installed so that all traffic between the browser accessing the web server is encrypted. This way we don't send any clear text over the Internet and make it very hard for session hi-jacking.

Database security

It's not just the operating systems that are vulnerable and needs hardening. The database server, in this case Microsoft SQL Server 7.0 also needs to be hardened as well.
One site I find particularly good is: http://www.sqlsecurity.com/checklist.asp
It will provide you with a checklist of what important things you have to consider when securing a SQL server.
Patching the SQL server is just as important as patching an IIS server. Database server needs to be configured to use TCP as default network protocol. Change this with SQL Client Configuration Utility and make sure that web server connection is using TCP as default protocol.

DCOM specific settings

Now it's time to deal with the most complex problems with this Defense in Depth scenario. Problem consists of the very basic functionality within DCOM, the SCM Service Control Manager. Whenever a connection is to be established, SCM will use port 135 for handshaking with the database server. The rest of the transaction will use port 135, unless this is redirected to other ports. Port 135 is a big "no-no" port since it's a NetBIOS port. Firewall administrators don't like the idea of having this port opened in their firewalls at any time.
Redirection of DCOM is not very difficult, however the port 135 handshakes will always be required by DCOM. Add following keys in the registry to redirect DCOM traffic:

HKEY_LOCAL_MACHINE \Software\Microsoft\RPC\Internet
Name: Ports
Type: REG_MULTI_SZ
Value: 4000-5000, 135

Name: PortsInternetAvailable
Type: REG_SZ
Value: Y

Name: UseInternetPorts
Type: REG_SZ
Value: Y

After a reboot of server, DCOM will use the port 135 for handshake and then switch over to use port 4000 -5000 for the actual transactions. **Note!** Do not use the DCOMCNFG utility to make these changes. This utility has a habit of messing up the registry in some cases. Use RegEdt32.exe and as always when changing the registry, pay attention on what you are doing.
This article from Microsoft explains the process of using DCOM with Firewall environments: http://www.microsoft.com/com/wpaper/dcomfw.asp

VLAN

In order to gain more security in the network we should use VLAN switches. One of the main benefits besides the increased network performance is the option to make each port protected with a virtual boundary that prevents the server connected to that port to access any other ports on the VLAN. This way we can protect the other server on the network from attack if one of the servers has been compromised in some manner. Example. If an attacker is successful in compromising one of the servers, he will be unsuccessful in using other servers in the same network as attack vectors by exploiting some vulnerabi lity in them.

Firewall settings

All three firewalls in this scenario will use protocol -level firewalls. This way we will utilize such technologies as packet -filtering, dynamic filtering and stateful inspection. The rules on each firewall will be described according to the application need. Rules for normal business need will not be covered.

Internet/DMZ firewall
Since this firewall is the main firewall and is located in the frontier of the corporate network, the firewalls rules should be kept to a minimum .

- Port 21 open for FTP traffic from Test and Stage web server for transfer of changes to production web server.
- Port 135 open for DCOM handshake traffic between web server and database server.
- Port 443 open for inbound/outbound SSL traffic to/from web server and out on the Internet.
- Port 1433 open for SQL traffic between web server and database server.
- Port 4000-5000 open for redirected DCOM traffic between web server and database server

Perimeter Network firewall
This is the firewall protecting the datab ase servers. Even though this firewall is inside the corporate network it should be handled as the main firewall and rules set to a minimum.

- Port 21 open for FTP traffic from Test and Stage database server for transfer of changes to production database ser ver.
- Port 135 open for DCOM handshake traffic between database server and web server.

- Port 1433 open for SQL traffic between database server and web server.
- Port 4000-5000 open for redirected DCOM traffic between database server and web server.

Test and Stage firewall
This firewall must resemble the other two firewalls functionality in order to deliver a suitable test environment for the application. As such this firewall will be more open.

- Port 21 open for FTP traffic to production web server and production database server
- Port 135 open for DCOM handshake, port 1433 open for SQL traffic and ports 4000-5000 opened for redirected DCOM traffic between Test and Stage web server and Test and Stage database server.
- Port 443 open for inbound/outbound SSL traffic to/from internal networks.

IDS

"Protection is ideal, but detection is a must." These words are really worth thinking about. It doesn't really matter on how much security you install in your networks if you don't know what's happening in them. The Gartner Group has made a market analysis on IDS which describes thoroughly what IDS is all about and why it is so important.
http://www.gartner.com/DisplayTechOverview?id=320015
In this scenario we will put an IDS box in the DMZ network segment and one in the Perimeter Network segment. We will use SNORT from Open Source Network, http://www.snort.org/, main reason for this it's free and pretty easy to use. Sn ort is a network based intrusion detection system also known as NIDS. Both IDS boxes will run on Linux RedHat 7.2 and should be hardened as well as the web server and database server. Even Linux boxes have vulnerabilities that need to be patched.
Providing IDS in DMZ and Perimeter Network will let us know if someone is attacking our systems in production.

System Integrity checking

How will you know if something has happened to your systems if you don't know exactly how your systems is supposed to look like ?
This is very hard and very tedious work. You have to spend hours after hours browsing through log files in order to get an idea on what happened. This is where Integrity checking systems comes in handy.
In this scenario we will use Tripwire for servers f rom Tripwire.
http://www.tripwire.com/products/servers/
Tripwire will keep track on the files on the servers and make sure they stay intact from changes. If something happens to the files, the administrator is alerted by it and can take action.
This will of course not relieve our administrators from browsing through log files. That is something we all have to do just to make sure our systems work correctly.

Antivirus

Both the web server and databas e server must have an Anti -Virus system installed. Even though we have tightened the security on these servers, virus can infect them. The web server should use on access scanning or systems scan. This means scanning all accessed files or documents on serv er. If a virus is trying to infect some files, the Anti - Virus should prevent that if the Virus Definition File, more commonly known as DAT can detect the virus.

The database server don't really need system scan activated since the server is basically only working within the SQL server.

However, both servers should have regular file scans scheduled to clean up any infected files.

Change management

Keeping the production environment as clean and untouched as possible is very important if you intend to have a good availability on the systems. This is why we have a Test and Stage environment, which is a replica of both the DMZ and the Perimeter Network. The Firewall in this environment has the same rules as the production firewalls. The whole meaning with this environment is to make all new changes here first and make sure application is still functioning correctly before either sending the changes via FTP to the production servers or manually make the changes in the production.

Note! All changes must be tested before going in to production, and I mean all. Even the slightest change in an ASP page could have a great impact on the applications functionality if you do this in live production. Say this application is selling music CD's. An average day the net sell is around 1000 CD's a' 10 US$. This would mean 10000 US$ in sales per day. Split this with 24 hours per day and you get 420 US$ earnings per hour. Wouldn't it be a pity if you wasted some hours of availability by changing this ASP page in production instead of doing this in an environment, which has no impact to your business?

The web server and the database server must be installed identically as the production servers. The only thing that should differ between them is the IP -address and their NetBIOS name s.

Hardening process on these servers must be identical as the production servers.

Backup

Having a backup policy on all systems is critical for any business to be successful. Make sure the servers in production do a full backup every week and incremental backups every day.

Database server databases should have their transaction logs backed up at frequent intervals.

The Test and Stage environment should have some special attention. These systems needs to be able to be restored completely and using traditional LAN-backup solutions can be time consuming.

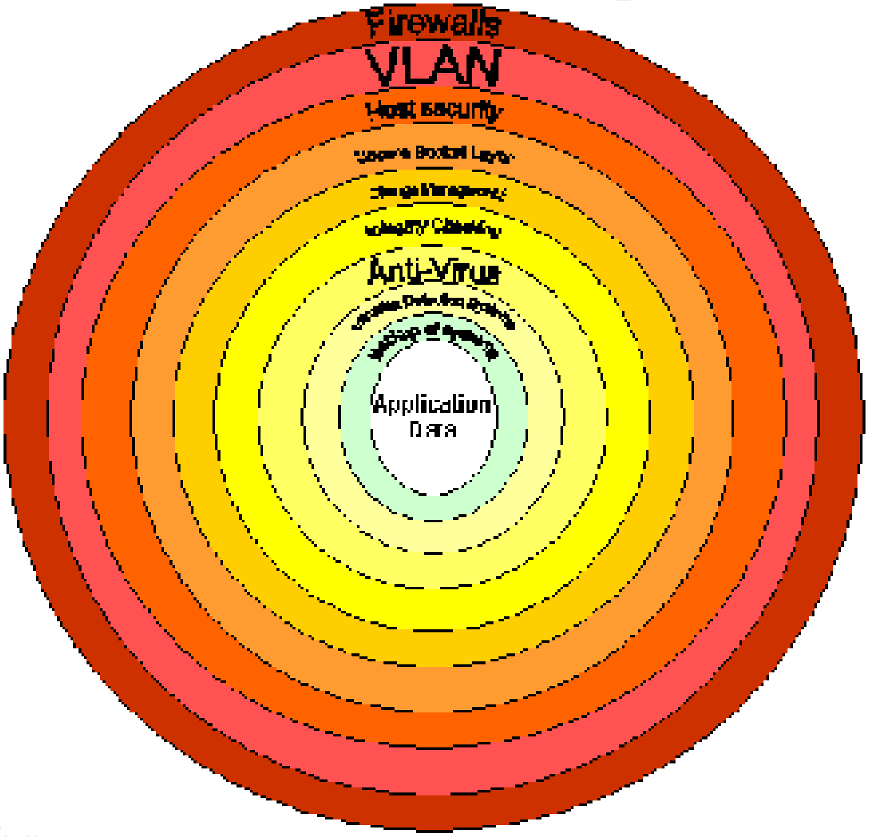
I would suggest using besides normal LAN -backup, another media such as Iomega's JAZ Zip drive, http://www.iomega.com/jaz/index.html . This device is capable of storing 2Gb on tape. Together with image packing software such as Symantec Ghost, http://enterprisesecurity.symantec.com/products/products.cfm?productID=3 or Power Quest Drive Image, http://www.powerquest.com/driveimage/ , the restore process can be as short as 15 minutes from time that server crashes.

Using this solution on the web server in production is also ideal. An ASP web server rarely changes and making an image of the production server could also make server crashes not to painful.

Summary

Picture 2.

"Defense in Depth illustrated"



By using all earlier mentioned ways to add security layer by layer, we have now deployed a Defense in Depth solution.

- The DMZ has been secured with Firewall rules protecting the DMZ and rest of the networks.
- Web server is tightly locked down and sends it's data to database server in a secure way. If compromised it can't be used to attack other server.
- The VLAN setting disallowing traffic to pass between ports in switch will stop any further attacks. Attack into the LAN is not

going to work either, since it is only allowed to access the database server on specific ports.

- The database server is protected from the Perimeter Network Firewall.
- Both operating system and SQL server has been hardened and a compromise on this system will be futile. Same things apply to database server as web server.
- The confidentiality is secur ed since the systems use encrypted traffic over the Internet.
- The availability on the systems is secured as well, since we have a test environment where every new change is tested before applied in the production environment.
- The integrity is kept due to t he Tripwire installations on the web server and the database server.
- Virus attacks by worms of file infections from PC's are protected as well since all servers have Anti -Virus software running on them.
- IDS will keep track on what's happening on the protec ted networks and hopefully keep us alerted of any network misbehavior.
- All systems are backed up frequently which adds to the availability of the application.

Hopefully all of these measures we have applied will make the Security System Administrator sle ep a bit better during nights.
These are only few steps to gain a more secure environment. The main idea is to address security and get people and management to start thinking in Defense in Depth manner.

References

1. US, National Security Agency, Securit y Recommendation Guides, Windows NT Guides
URL:http://nsa1.www.conxion.com/winnt/guides/wnt -securityguides.zip

2. Microsoft, Windows NT 4.0 Server Baseline Security Checklist
URL:http://www.microsoft.com/technet/security/tools/chklist/nt4svrcl.asp

3. Microsoft, Internet Information Server 4 Baseline Security Checklist
URL:http://www.microsoft.com/technet/security/tools/chklist/iis4cl.asp

4. Microsoft, IIS Lockdown Tool
URL:http://www.microsoft.com/technet/security/tools/tools/locktool.asp

5. Microsoft, HFNetChk
URL:http://www.microsoft.com/technet/security/tools/tools/hfnetchk.asp

6. SQLSecur ity.com, SQL Server Security Checklist
URL: http://www.sqlsecurity.com/checklist.asp

7. Microsoft, Using Distributed COM with Firewalls

URL:http://www.microsoft.com/com/wpaper/dcomfw.asp

8. Intel, Technical Brief VLANs
URL:http://www.intel.com/network/connectivity/resources/doc_library/tech_brief/virtual_lans.htm

9. Gartner Group, By Allan Ant, Intrusion Detection Systems (IDSs): Perspective
URL: http://www.gartner.com/DisplayTechOverview?id=320015 - h2

10. SourceFire, Snort The Open Source Network Intrusion Detection System
URL: http://www.snort.org/

11. Tripwire, Tripwire for servers
URL:http://www.tripwire.com/products/servers/

12. Iomega, Iomega JAZ
URL: http://www.iomega.com/jaz/index.html

13. Symantec, Symantec Ghost Corporate Edition 7.5
URL:http://enterprisesecurity.symantec.com/products/products.cfm?productID=3

14. Power Quest, Drive Image 2002
URL: http://www.powerquest.com/driveimage/