Global Information Assurance Certification Paper

Securing Applications over Hostile Networks Using Citrix Metaframe

Warren O'Neill
9th July 2003

GSEC White Paper Version 1.4b

Contents

Abstract

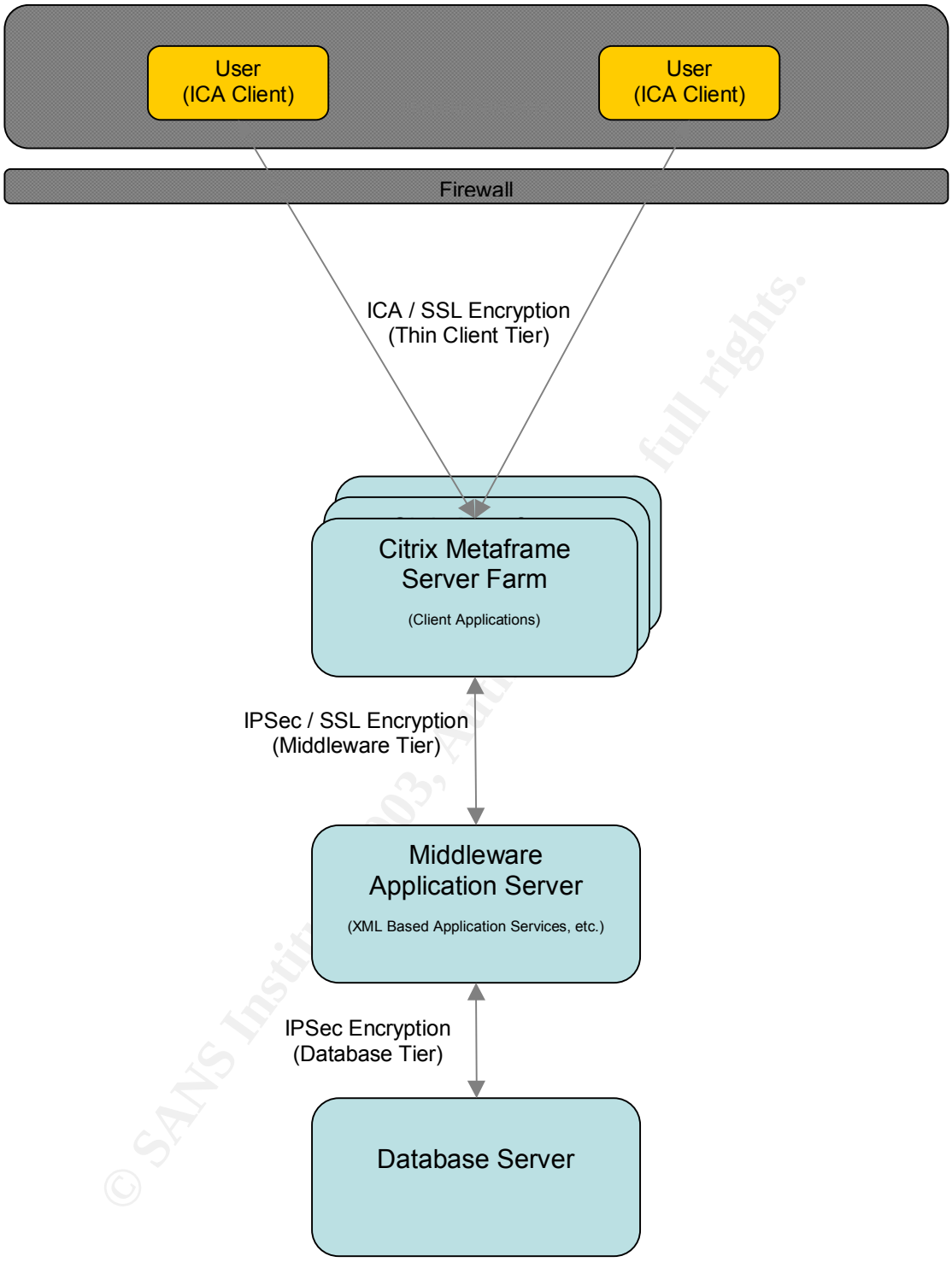
Many businesses today require the deployment of applications providing sensitive data to various departments within their organization. In many cases the recent trend of outsourcing complete business units means that this data must be transmitted to the authorised user (who is often at another site or company) over potentially hostile networks. The majority of these applications, such as payroll, HR, call handling systems and other database applications containing sensitive data use Windows client applications, however few of them address network security as a fundamental part of their design. In institutions such as universities it is often policy to allow virtually free access to the network, and have disparate datacentres, resulting in application servers spanning multiple buildings, whilst still having the requirement for securing of sensitive data. An increasing number of IT departments are choosing to deploy Citrix Metaframe as part of their infrastructure in order to address these issues (as well as cost saving issues – See Chris Johnson's paper *Understanding and Implementing Microsoft Terminal Services & Citrix Metaframe* in the SANS reading room for more information on this topic as well as a background introduction to the features of Citrix Metaframe and it's benefits over Microsoft Terminal Services). This paper focuses on the end-to-end securing of application data using Microsoft and Citrix technologies and industry standard protocols, in order that sensitive data can be securely transmitted in it's entirety from a central application or database to the authorised user. This paper mainly focuses on Windows applications, however whether the back-end database and application servers are Windows, UNIX, or some other OS, the principals here remain the same, although the implementation steps may change. Although focussing on Citrix Metaframe it must be noted that with many applications native data encryption is supported by the vendor, therefore the use of a thin client solution may not be appropriate and may add an additional undesirable overhead.

Architecture

The architecture of a basic environment consisting of clients, databases and middleware servers can be described as a model which is broken down into tiers corresponding to the type of traffic encryption which is available and appropriate. Many modern applications use this type of application model, particularly common three-tier applications consisting of a client, middleware or application server and a back end database, however for simple client-server applications requiring only client access to a database, the same model would apply, consolidating the Database and Middleware tiers (described later in this paper). Using Defence in Depth principals and industry best practices, these types of environment can be secured in several ways, depending upon your organisation's security requirements. The measures described in this document are intended to be applied in addition to the standard network and host based security practices (such as maintaining current OS patch levels, Limiting web permissions with IIS Lockdown, and so forth). The specific architectures examined revolve around using Citrix SecureICA and Citrix Secure Gateway, although both designs also include the use of IPSec for back-end server communication. The models presented in this paper are examples which should be easily mapped to other similar environments where more (or less) complex networks exist in terms of DMZ configuration and firewalling. The breakdown of this application model into tiers allows for the simple modular construction of specific application security frameworks, for example omitting the middleware – database tier IPSec encryption if the respective servers are co-located on the same subnet (or better still, the same physical switch) and protected by a common firewall.

Fig. 1 illustrates the basic overview of the conceptual network data transmission security model, outlining the tiers described below.

Fig. 1:



Database Tier

The Database Tier describes the action of connecting clients or related servers to a back-end database system. In several cases portions of this tier may be redundant, if middleware and database services reside on the same system or if a simple 2-tier application is in use. This database tier, which will be usually ODBC traffic over TCP/IP, can be most effectively secured using an infrastructure level solution, IPSec. IPSec can encrypt the entire contents of an IP datagram (above OSI layer 3), and can be implemented under Windows 2000, Windows 2003, and virtually all flavours of UNIX in order to encrypt traffic between certain hosts on the chosen specified TCP or UDP ports. Under Windows 2000 and above, IPSec policies can be configured to customize and control the behaviour of IPSec, such as to only encrypt data who's destination is outside of the specified network range (i.e. a hostile network).

Due to the differences in the implementation of TCP/IP and platform security between Windows and UNIX, the authentication method used by IPSec will need to be identified on a "per case" basis, as Windows platforms can happily perform IKE authentication for IPSec using Kerberos tickets issued by an Active Directory Domain. For pure Windows environments this is the simplest option, as all Kerberos tickets are issued automatically by an integrated ticketing authority in the AD, therefore there is no additional management overhead associated with maintaining certificate authorities, certificates or shared secrets. If the environment consists of a combination of UNIX and Windows systems, another approach may need to be adopted (depending on the UNIX in question), where either pre-shared key or certificate based authentication is used for key negotiation between systems, using certificates issued from a trusted Certificate Authority within a Public Key Infrastructure (PKI).

For all tiers only the traffic identified as requiring encryption by the business, legal or security department usually needs to be encrypted. This can easily be controlled by a system administrator (using IPSec policies under Windows 2000 and higher), based on system interconnectivity between certain IP addresses over given TCP ports. For example, in order to secure the conversation between a Citrix server (CTX01) connecting with ODBC over TCP 1433 to a Microsoft SQL Server instance running on server DB01, it is simply a case of configuring the IPSec policies on DB01 to require IPSec encryption for all traffic from CTX01 to it's self with destination port 1433, and assigning an authentication key (either shared secret, Kerberos or a certificate).

The additional overhead of encrypting data before transmission can cause a significant additional load to a system's CPU, therefore the use of hardware based IPSec offload cards is strongly recommended.

Middleware Tier

The Middleware Tier describes the connectivity between clients (including Citrix or Terminal Servers) and middleware application servers, which may support their own native encryption. If an application supports native encryption (and the algorithm it uses is approved for use by your company) then it should be transmitted over IP, and not further encrypted with IPSec, as this would create an additional unnecessary overhead on both systems in the conversation. An example instance of this is using SSL to encrypt HTML and XML traffic.

All web servers providing sensitive data should be configured to use HTTPS rather than HTTP, using an SSL encryption certificate allocated by a trusted Certificate Authority. All XML based applications which are to be used by third parties should also use HTTPS as the protocol to transmit data over the network, in exactly the same manner as native web applications. Application servers which are able to secure their web traffic using SSL do not require any further encryption with IPSec unless they will also host other insecure applications or data which transmit or receive sensitive data.

If SSL (or other upper-layer) encryption is not an option due to the application communicating via some other protocol (such as COM+ or native TCP Winsock communications) or the application it's self not supporting encryption, then IPSec should be adopted as the most effective method of securing this data conversation. Exactly the same "per case" considerations will apply which were highlighted for the Database Tier, depending on OS compatibility, IPSec authentication capabilities and specific data encryption requirements (for example encryption may not be required between Citrix and middleware servers residing in the same datacentre). In the case of most corporate networks, it is likely that a Windows Active Directory based Kerberos ticketing authority would be used for IPSec authentication to the client, as the middleware and database components may also both run under Windows. If any of the servers involved reside on a UNIX platform, then it may become necessary to use pre-shared key or certificate based authentication unless a platform-specific implementation of Kerberos key authentication is available.

Thin Client Tier

The role of the Thin Client Tier is to provide authorised users with a secure method of accessing sensitive data by using a thin client solution instead of direct client access to the application. For the case of Windows based client applications, Citrix Metaframe XP is recommended for this purpose, as it is a proven technology in terms of providing this connectivity, and is easily managed as only a single set of rules are required on any firewalls in front of each Citrix farm member. The main alternative to Metaframe would be Windows Terminal Services, however this does not provide any native data encryption and lacks the feature-rich environment provided by Metaframe XP.

Deploying a Citrix Metaframe thin client solution avoids the problems of users running non-Windows desktop operating systems, older Windows operating systems being unable to use IPSec, or client accessibility to a common certification authority.

It is recommended that an n+1 model be adopted when calculating the quantity of Citrix servers required to host any applications, based upon application profiling for CPU, memory and network requirements during a typical user session, and calculating the number of servers required, plus one additional system for redundancy and to allow maintenance without service outages. It is recommended that the Citrix Servers be built using Windows 2000 or 2003 (to the current patch level) and Metaframe XP 1.0 Feature Release 3 (again, patched to the current release level). Windows policies should be used to grant users the minimum rights required, and NTFS permissions should be revoked for users on certain executables in order to prevent users from "breaking out" of their published application session (for example TASKMGR.EXE or IEXPLORE.EXE can both be used to easily launch other applications which should not be available to the user). It is worth noting that with today's software, it is often possible to spawn another application process from within a published application, therefore each application to be published should be reviewed by your security department in order to mitigate the risks inherent with publishing server based applications to end users. All applications to be published should be reviewed for native security holes and patched accordingly, or the risk analyzed and accepted if deemed to be viable.

Native network security is available under Citrix Metaframe XP, which encrypts the contents of the Citrix ICA data stream using 40, 56 or 128-bit RC5 encryption with Diffie-Hellman key negotiation (SecureICA). This SecureICA encryption can be easily enabled per published application by simply checking a box in the Citrix Management Console administrative tool for Metaframe XP, where administrators can also select between 40, 56 and 128 bit RC5 encryption for either the entire session or only the password authentication phase of the conversation.

One alternative to using SecureICA would be to transmit standard ICA traffic over IPSec, however this would require that all ICA clients are running Windows 2000 or XP, and they would all need access to either the Active Directory or a trusted CA in order to gain IPSec authentication, therefore the Citrix model is perhaps more appropriate for connecting external partners or clients due to the management overhead related to supporting an IPSec infrastructure.

Another alternative solution provided by Citrix, is to implement Citrix Secure Gateway (CSG), which is a product running on dedicated hardware acting as an SSL encrypted ICA proxy between the clients and the Metaframe servers, hence removing the data encryption overhead from the Metaframe servers themselves.

When using a Citrix Secure Gateway solution a user's ICA client connects via SSL to a CSG server instead of the Metaframe server hosting their application, and the Secure Gateway server forwards the ICA data on to the respective Metaframe server and handles the SSL encryption / decryption of the traffic between the client and it's self. The ICA conversation between the Metaframe servers and the Secure Gateway can still be encrypted using IPSec so secure that portion of the conversation. For practical use of a Citrix Secure Gateway, users should connect to the Metaframe farm via an NFuse application portal, which provides a list of available published applications in a web browser over an SSL encrypted session, and allows secure ticketing between the NFuse server and the CSG (See *Best Practices for Securing a Citrix Secure Gateway Deployment* by Citrix Systems Inc. for further information on CSG deployment and configuration).

There is not a great deal of documentation available on implementing Secure Gateway "farms", with load balancing handled by a network device such as a Cisco CSS, however backup CSGs can be configured in each NFuse server's configuration file in order to allow failover of user sessions to a secondary CSG should the primary fail. The hardware load balancing option should be fully explored if a Secure Gateway solution is to be deployed, as it avoids building in any single points of failure to the application infrastructure without the need for deploying redundant hardware, as all members of a load balanced farm will participate in carrying the user connection load. Sticky sessions will be required on the load balancer in order for any hardware based load balancing of CSGs to function successfully.

NFuse is the recommended method of providing the ICA client application and list of available published applications to the users' desktops, as it can be centrally controlled and is easily maintained in terms of required firewall rules, as it utilises industry standard HTML and XML requests over HTTP(S) in order to retrieve the executable client code and the available applications list through a standard web browser. When using NFuse, more than one server should be deployed using some kind of session enabled load balancing, such as a Cisco CSS box, or Microsoft's Network Load Balancing Services. NFuse can run under both Windows (IIS) and UNIX (Apache). For further information on securing NFuse and CSG servers, see the Citrix Consulting document *Best Practices for Securing a Citrix Secure Gateway Deployment*.

Basic SecureICA / IPSec Infrastructure Architecture

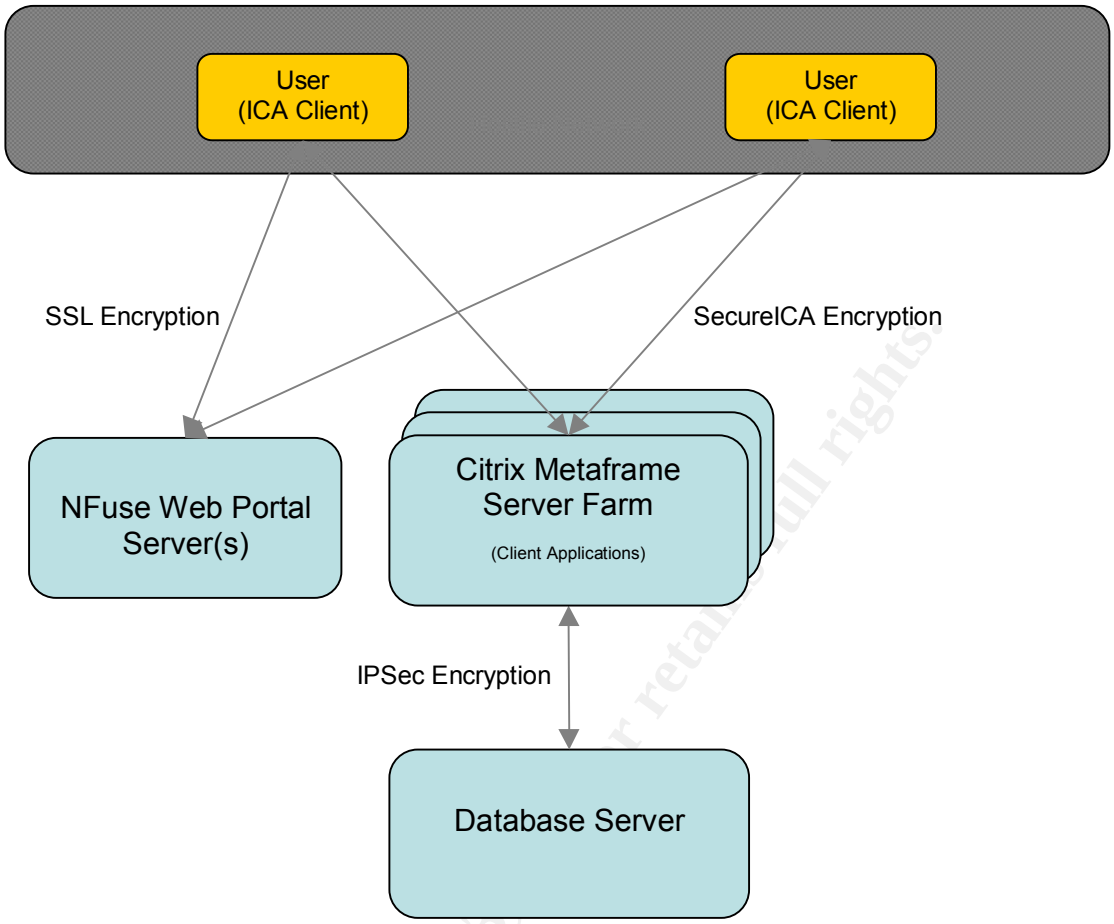
The most basic deployment of this model for application serving would consist of a single database / application server providing data directly to the client application (e.g. over an ODBC connection). For this scenario, the infrastructure could be configured using IPSec to encrypt the ODBC conversation between the database and the Citrix servers hosting the client application. The conversation between the ICA client and the Metaframe servers would most typically be secured using SecureICA, which is an integral component of Metaframe XP. Fig. 2 outlines this configuration.

The Thin Client tier uses (up to) 128-bit RC5 encryption to secure the ICA session data over TCP 1494. The consolidated Middleware / Database layer uses IPSec to secure transport layer data transmitted between the Metaframe servers and the database server over the specified ports (e.g. TCP 1433). Traffic between the client and the NFuse server is encrypted using SSL over TCP 433. Fig. 2 outlines this configuration.

The conversation between the client and the back-end application (database) would be as follows:

1. Client machine browses to https://nfuseserver.company.net/ to access the NFuse web portal (TCP443, SSL)
2. User logs in to NFuse (TCP 443, SSL).
3. NFuse server contacts XML Service on Metaframe farm member (as defined in it's local browser list) to acquire list of available applications for the user (TCP 80, IPSec).
4. The available applications are displayed in the client's browser (TCP 443, SSL)
5. The user selects a published application to run by clicking it's icon in their browser. This action causes NFuse to contact the Citrix server's XML service (TCP 80, IPSec) to obtain the IP address of the least busy server running the requested application.
6. The NFuse server sends a dynamically generated ICA file containing the IP address of the Metaframe Server to the client (TCP 443, SSL)
7. The ICA file is automatically interpreted by the ICA Client software, and a encrypted session to the Metaframe server is established (TCP 1494, SecureICA).
8. Having established a session to a Metaframe server, the server runs the requested published application on behalf of the client, presenting it to the client over SecureICA.
9. The published application contacts the back-end database server (and associated middleware servers) over IPSec, encrypting whichever ports are required to secure the data transmission (e.g. TCP 1433)

Fig. 2:



Citrix Secure Gateway / IPSec Infrastructure Architecture

Although SecureICA provides data encryption, there are cases where it may not be fitting, as it adds an additional overhead to each Metaframe server in encrypting all it's client sessions. A more complex solution which would be required in the case of offloading this processing, or providing Metaframe published applications to users over the Internet could be achieved either using a VPN solution or a Citrix Secure Gateway tunnel. As VPN technology is fairly well understood and in widespread use, this paper focuses on the Citrix Secure Gateway (CSG) solution. Fig. 3 outlines this solution.

The Thin Client tier is secured by sending unencrypted ICA session data through an SSL encrypted tunnel, created between the CSG and the ICA clients. All client ICA traffic to any member of the Metaframe farm is channelled through the SSL tunnel, handled by the CSG server. For this reason, a secondary CSG should be configured as a standby, to which the clients can be redirected by the NFuse server if the primary is unavailable. Additionally, a secondary Secure Ticket Authority (STA) should also be configured to avoid the STA becoming a single point of failure. An alternative option would be to load balance the SSL client traffic between multiple CSG servers using a hardware load balancer such as a Cisco CSS using sticky sessions to maintain client session state, although this would require full testing prior to deployment.

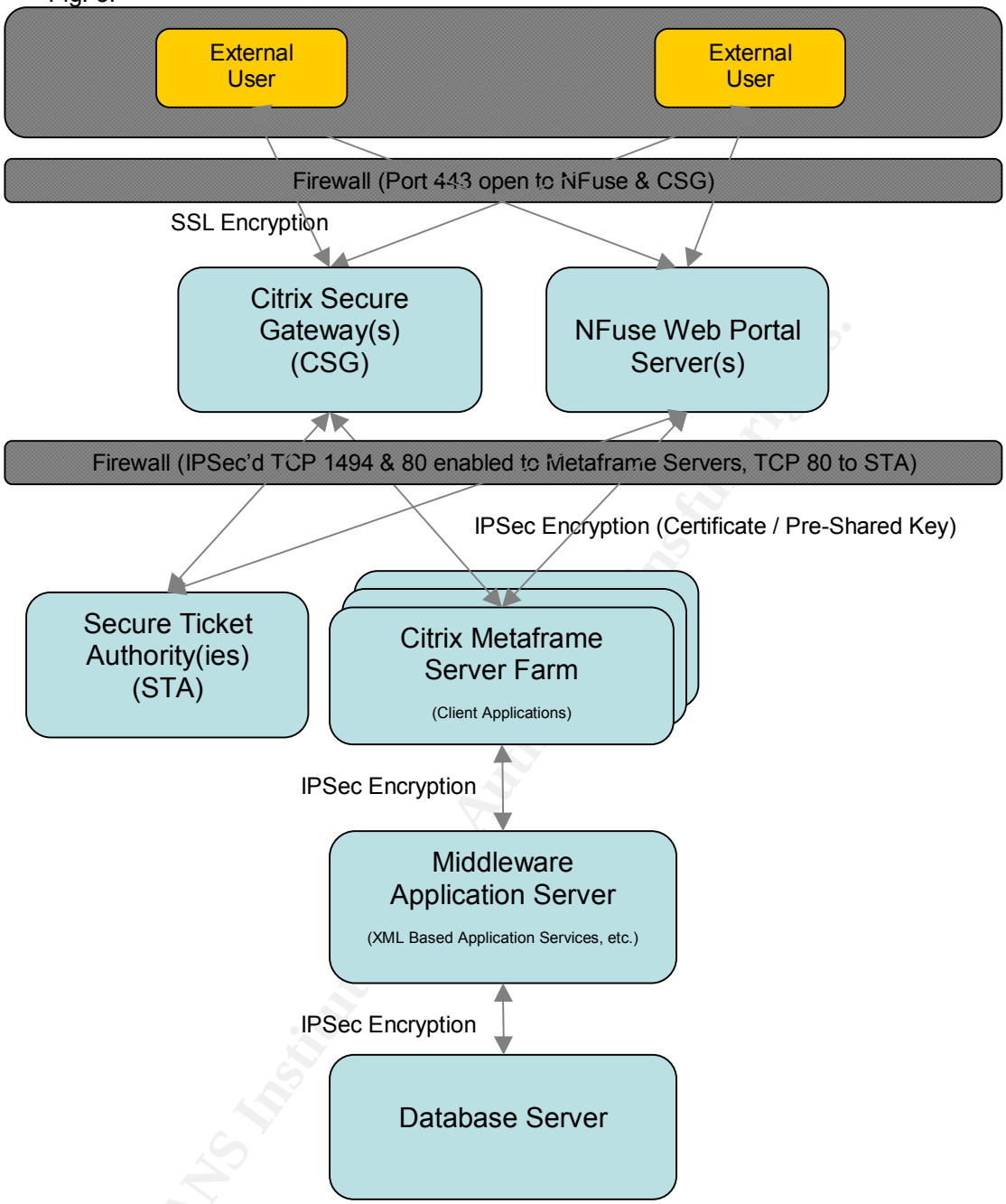
The Middleware / Database tier(s) are encrypted using IPSec. As the CSG and NFuse servers sit in a DMZ, no native Windows Kerberos ticketing should be available, therefore IPSec between the DMZ and internal network should be authenticated using Certificates or pre-shared keys.

The conversation between the client and the back-end application (database) would be as follows:

1. Client machine browses to https://nfuseserver.company.net/ to access the NFuse web portal (TCP443, SSL)
2. User logs in to NFuse (TCP 443, SSL).
3. NFuse server contacts XML Service on Metaframe farm member (as defined in it's local browser list) to acquire list of available applications for the user (TCP 80, IPSec).
4. The available applications are displayed in the client's browser (TCP 443, SSL)
5. The user selects a published application to run by clicking it's icon in their browser. This action causes NFuse to contact the Citrix server's XML service (TCP 80, IPSec) to obtain the IP address of the least busy server running the requested application. This IP address is sent (TCP 80, IPSec) to the STA, which returns an authentication ticket to NFuse for the user's session (the ticket has a short expiry period).
6. The NFuse server sends a dynamically generated ICA file containing the authorisation ticket and the IP address of the CSG to the client (TCP 443, SSL)

7. The ICA file is automatically interpreted by the ICA Client software, and if the associated ticket is validated successfully, an SSL encrypted session is established between the ICA client and the CSG (TCP 443, SSL).
8. The CSG contacts the STA (TCP 80, IPSec) to obtain the IP address of the Metaframe server hosting the application for the client, and establishes a session to that server (TCP 1494, IPSec) on behalf of the client. The client is now able to use the Metaframe Server's published applications seamlessly, with the CSG performing the data encryption and decryption.
9. Having established a session to a Metaframe server, the server runs the requested published application on behalf of the client, presenting it to the client via the CSG.
10. The published application contacts the back-end database server (and associated middleware servers) over IPSec, encrypting whichever ports are required to secure the data transmission (e.g. ODBC traffic over TCP 1433)

Fig. 3:



Allowing IPSec Traffic Through Firewalls

In order to implement the above architectures, it will be necessary in most cases for the back end IPSec traffic to traverse a firewall. The simplest method of implementing this is by implementing a Windows IPSec policy at each host in the conversation, requiring IPSec encryption for the desired ports to be allowed, with a common certificate or pre-shared key used. In addition to this, a "deny" type IPSec policy should be put in place on the internal systems in the conversation, in order that a simple "allow IPSec host – host" rule can be implemented on the firewall, whilst ensuring that if the DMZ server is compromised, additional IPSec ports cannot be opened by the attacker. The "deny" policy is created by adding a custom filter with the "block" action to the IPSec policy, and applying this to all ports and addresses other than those required for the application to function. This method, although adding a level of security, is not ideal as a great deal of management overhead is involved in maintaining the IPSec configuration, particularly for configuring authorised access to any additional ports which may become required.

A more desirable option for allowing IPSec traffic through the firewall for certain ports only would be to deploy a firewall capable of decrypting the IPSec Transport mode traffic, inspecting it's payload, then either allowing or blocking it based upon the rule set in place. This requires the placement of the certificate or pre-shared key on the firewall it's self, in order that decryption of the ESP payload can take place. Although this is theoretically possible (and draft papers have been published to the IETF on the subject for several years), finding a practical implementation of this method has proved more difficult. The technology exists today, and applications such as *tcpdump* can monitor IPSec traffic if issued with information on the correct encryption protocol and key in use by the traffic. Although there appears to be no (or at least very few) currently available implementations of this technology being deployed on firewalls in the field, this is an area which could benefit from future investigation by the information security community.

References

Best Practices for Securing a Citrix Secure Gateway Deployment. Citrix Systems Inc. http://support.citrix.com/

Citrix SecureICA Administrators Guide v1.1. Citrix Systems Inc. http://support.citrix.com/

Citrix SecureICA Administrators Guide v2.0. Citrix Systems Inc. http://support.citrix.com/

Windows 2000 Professional Resource Kit. Microsoft Press.

Johnson, Chris. Understanding Microsoft Terminal Services & Citrix Metaframe. December 10 2001. http://www.sans.org/rr/papers/59/319.pdf

Riley, Steve. Using IPSec to Lock Down a Server. January 15 2003. Microsoft. http://www.microsoft.com/serviceproviders/columns/using_ipsec.asp

Kent,S. IP Encapsulating Security Payload (ESP). IETF IPSec Working Group. http://www.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v3-05.txt

Huttunen, A; Swander, B; Stenberg, M; Volpe, V; DiBurro, L. UDP Encapsulation of IPsec Packets. IETF IPSec Working Group. http://www.ietf.org/internet-drafts/draft-ietf-ipsec-udp-encaps-06.txt

"Soft Associations" Between IPSec-Enabled and Non-IPSec-Enabled Computers. http://support.microsoft.com/default.aspx?scid=kb;en-us;234580