Global Information Assurance Certification Paper

Securing access to external networks with Authentication, Authorization and Accountability.

GIAC Certification

GIAC Security Essentials Certification (GSEC)

Laurent GELU

Practical 1.4b Option 1.

April 7th, 2004

Abstract

There are several solutions to interconnect trusted and un-trusted networks, such as Extranet or other company's networks. The purpose of this document is to introduce and discuss the different ways to provide secure access solutions to an external network.

The two major points we will outline are:

- The Individual authentication, authorization and accountability.
- The network routing facilities

We will start with a basic Firewall authentication solution with NAT, and continue with the introduction of socks server as a proxy based solution to facilitate routing, and will finish with the possibility to use a stepping stone server like Citrix Metaframe, when bandwidth is limited.

We will not talk about the VPN solution. VPN is a transport solution over a public un-trusted network to interconnect remote computers to the corporate network. Those remote computers are then logically treated as local computers.

My objective will be to describe all those solutions, provide pros and cons, and to introduce a typical implementation example.

As a conclusion, depending on your routing requirements and your budget, you will see that many different combinations are possible.

1° Firewall authentication with NAT

An Introduction to Network Address Translation (NAT)

Network address translation (NAT) is the process of changing the source or destination IP address of packets. Originally NAT was proposed as a short-term solution to the problem of limited amount of Public IP addresses (due to the IPv4 Limitation). NAT is described in RFC 1631 [1].

NAT concept is based on the fact that only a small part of the hosts in a private network is communicating outside that network. If each internal host have gets a Public IP address only when it needs to communicate to the external side, you only require a small number of Public IP.

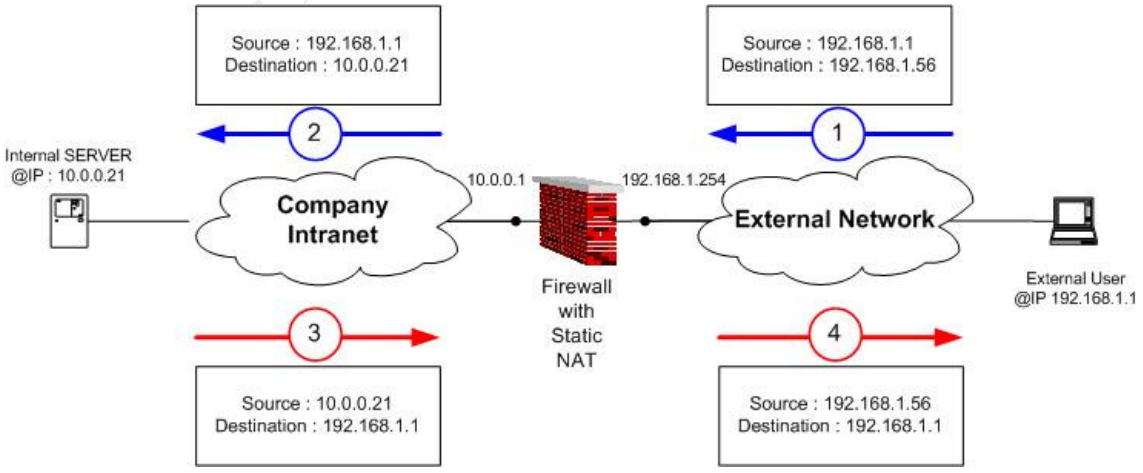
STATIC MODE / Network Address Translation

Static NAT provides a one-to-one assignment between the registered IP address and the internal IP address. The translated (valid) addresses must be published, on the external network so that replies will be routed back to the gateway. It allows clients with IP addresses that are not routable to access an external network using a routable address. The NAT gateway substitutes a client IP address with a routable address before sending the packet out.

Static mode is implemented when an organization needs to publish IP addresses for public servers, but does not wish to route or expose the private internal IP addresses.

FIGURE 1: Static NAT Example.

In the scheme below, the Internal Server is translated in the External Network with the @IP 192.168.1.56. Proxy ARP must be configured for the translated IP on the Firewall in order to reply to the ARP request for 192.168.1.56 with its own MAC address.

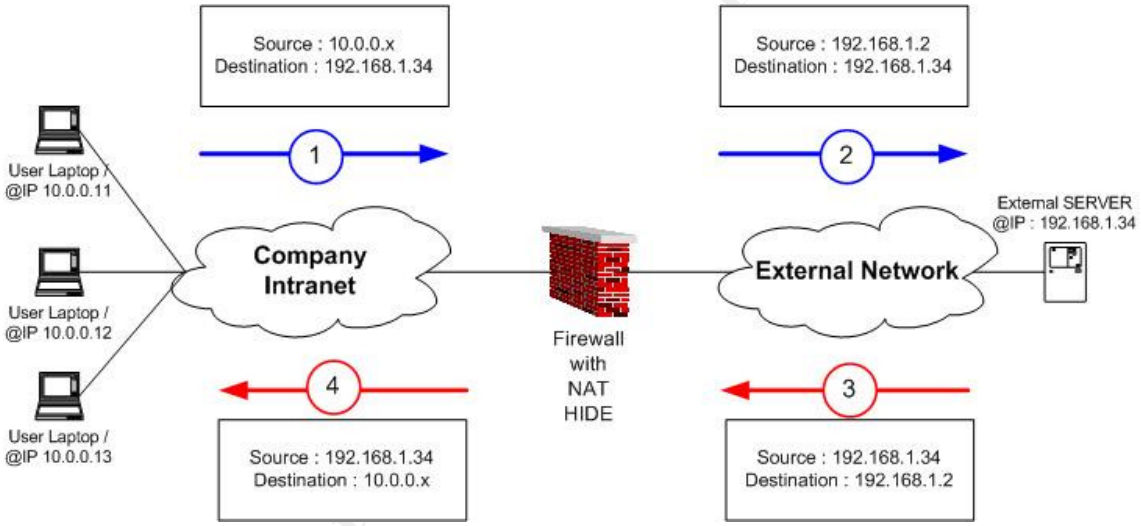


HIDE MODE / IP Masquerading (also known as PAT / Port Address Translation):

Hide NAT allows a single device, such as a Firewall, to act as a secure gateway between an external network and a private network. This means that a single, unique IP address is required to represent an entire group of computers on the external side. Hiding internal network's real IP addresses is beneficial for security reasons. It helps preventing IP spoofing attacks.
NAT HIDE is a solution for networks that have private IP address ranges or non registered addresses that need to communicate with hosts in an external network having registered public addresses, and which do not require communication coming from the outside.

FIGURE 2: NAT HIDE Example

In the scheme below, all internal devices are translated with dynamic address translation when they attempt to reach the external network.



Different Firewall authentication methods

In response to network authentication security requirements, a logical place to enforce the authentication functionality is on the firewall. It has a functionality that verifies the user identity by requesting a user-id and password. This enables secure access to external resources based on user identity.

The purpose of network authentication is to validate the right to access the external infrastructure by the use of a personal user-id and password. User will be authenticated at the firewall level as he goes from the intranet to the external network. With the use of external authentication servers, you can also extend this to two-factors authentication.

A firewall authentication service provides secure access to applications and other resources without requiring any modification of the underlying applications, servers or clients.

Firewall can provide three authentication methods:

- Proxy based User Authentication.
- Non-Transparent Network Authentication by using a separate authentication server (a manual action is required).
- Transparent / Agent based Network Authentication.

All these methods can be also combined to standard Firewall filtering based on the source or destination address. As an example, it is allowing administrator to grant access privileges to a specific user at a specific IP address. Most commercial firewall solutions also support encryption to protect the credential over the network, but also third party users' directory to put in place two-factor authentication.

Interesting additional resource on Firewall authentication can be found:

[2] For Cisco PIX, a paper from Scott Jensen:
http://www.giac.org/practical/scott_jensen_gsec.doc

[3] For Checkpoint Firewall1, the Authentication Phone Boy FAQ:
http://www.phoneboy.com/bin/view.pl/FAQs/AuthenticationFAQs

A) Proxy based User Authentication

User Authentication is provided for non-encrypted protocols where gateway or proxies are available. This is the case for protocols like TELNET, FTP, and HTTP. When a firewall rule specifies User Authentication, the Firewall will intercept the user's attempt to start an authenticated session on the external network and will redirect the connection to the appropriate internal Proxy. After the user is authenticated, the Firewall proxy server opens a second connection to the destination host.

The advantage of this method is that it is clientless; it does not require logging on any additional authentication server. But it is strictly limited to supported proxies on the Firewall, which are most of the time only available for unsecured protocols. In addition, new firewall authentication is required for each new connection.

B) Non-Transparent Network Authentication by using an authentication server [4]

This method is also called "client Authentication" on Checkpoint Firewall-1, or virtual telnet server on PIX. In contrast with the first authentication method, this one is not restricted to specific services, but can be used to authenticate any outbound connection. Client Authentication is not transparent, but it does not require any additional software or modifications on either the client or server.

The principle of this authentication method is to login first through Web browser or a telnet session to type their user-ID and password. Then, the Firewall will authorize establishment of outbound flows based on the user authorization. By the way, the credential is time limited: after the timeout, the user will have to re-authenticate.

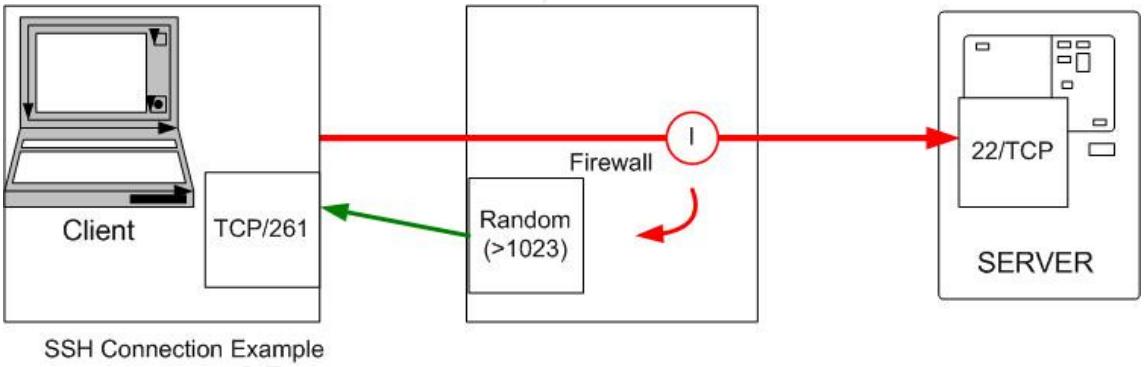
The limitation to this authentication is due to the way it works. During the valid authorization period of time, the firewall will trust the source IP address, and will not check if the workstation has changed.
In addition, if a user authenticates through HTTP, and has a proxy within its browser configuration, the authenticated IP will be the proxy IP. In that case, connection from the proxy server to your external network will be authorized to pass the firewall, and, at the same time, all workstations using this proxy! If you choose to implement this authentication method, you must deny all proxy and socks servers to authenticate on the Firewall with some specific filters, which will drop all flows from those servers. This is the major weakness of the solution.

C) Agent based Network Authentication. [4]

This method is available on Checkpoint Firewall-1. It is also named as “Transparent Session Authentication” or “session authentication”. It is used to authenticate users to any application / flow on a per-session “basis”.

As described on the figure 3, after the user initiates a connection directly to the server (red flow), the Firewall intercepts (I in figure 3) the connection, recognizes that it requires authentication, and initiates a connection with a Session Authentication Agent (in green). The Agent performs the required authentication by prompting the user with a pop-up. If the authentication is successful, then the Firewall allows the connection to pass through the gateway and reach the target server.

FIGURE 3: SSH Connection with Agent based authentication.



SSH Connection Example

This method is really secure, because the agent will authenticate for each new session, there will be no spoofing possibility and no credential timeout. From a security choice, it is the best one, but it has two limitations:

-The session agent must be installed on each workstation.
- It requires a connection from the firewall to the client.
- It does not work from multi-user / shared systems client like Unix or Citrix.

Example of Firewall authentication with NAT Implementation

Consider an internal network that is based on the private IP address space, where a user wants to use an application located in the external network from the internal network. The external network is routed on the Intranet. Within all examples, Firewall will be considered as statefull. TCP/ACK return connection will be automatically addressed.

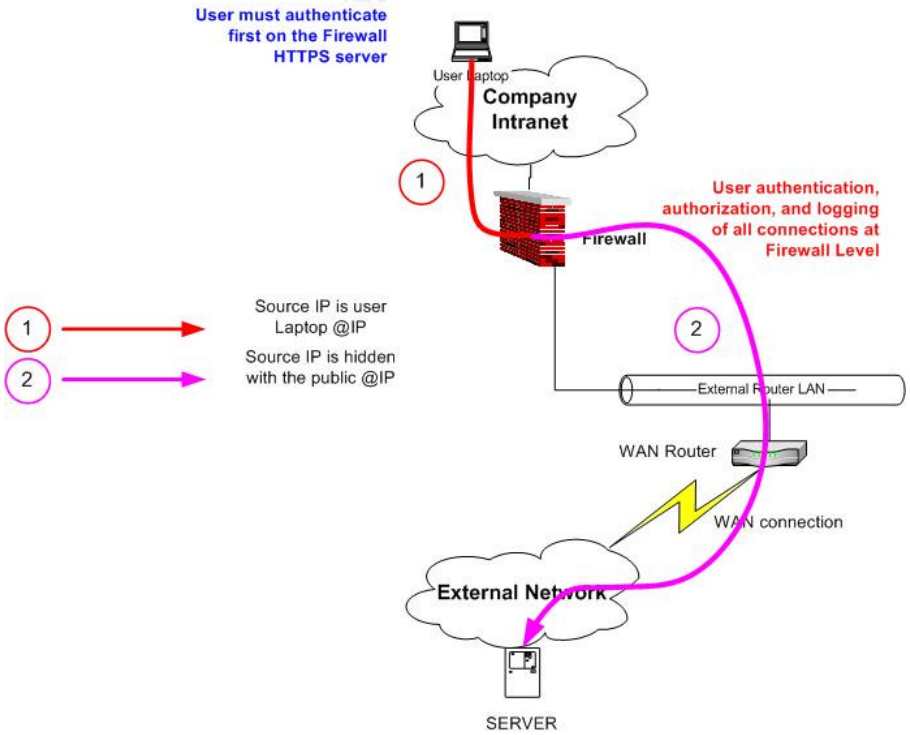
As a first security step, you should hide your internal network address space to the un-trusted network. In addition it will avoid any routing issue on the external side, as you won't need to route your internal network outside.

By the way, the firewalls will keep track of a session state, deny unauthorised session establishment and prevent spoofing, if configured correctly. Non-transparent authentication is used to authenticate and grant the access to the external network. For this purpose, the Checkpoint firewall includes a HTTPS service on port TCP/900. A connection to this service for authentication purpose is required to grant access to the external network.

A filter on the firewall will be defined to perform authorization based on the username and password. If DHCP is used on your internal network, you should only define firewall rules based on the users as the source and the DHCP subnet.

As a protection against internal IP spoofing (user taking the IP of an authenticated one), you should configure a short authentication timeout because, after authentication, the firewall grants the access during a short amount of time. If you have different kinds of user, you should specify user group to define some limited access per user role. In addition, do not forget to reject connections from your internal proxies, as already mentioned.

FIGURE 4: Example of NAT implementation:



Required Firewall Rules:

Source	Destination	Dst Port	Action
Internal Network	Firewall	900	Accept
User Laptop@internal Network	SERVER	Ex : FTP	Client Auth
Any	Any	Any	Drop

- The first rule allows users on the internal to authenticate to the firewall by reaching the port TCP/900.
- The second rule authorizes the traffic from the authenticated user to the Server.
- The last rule denies all other flows.

Required Address Translation (HIDE):

Source	Destination	Dst Port	Translated Source	Translated Destination	Translated Port
Internal Network	External Network	Any	Public IP	External Network	Equal

The Internal Network is hidden behind on the destination port.

Advantages and Drawbacks

Most of the benefits of NAT with Firewall authentication are network related. NAT consumes less system resources than a proxy based solution, so it provides faster performance.

In addition, it does not require any specific configuration on the user side, for both routing and authentication by the use of a non-transparent model. This solution should be selected if you have many workstations, and if you cannot manage each configuration.

On the opposite, NAT doesn't support protocols that use or transport IP addresses in the applications layer. An example is SNMP for Network monitoring.

Routing remains required if you do not perform "double NAT" to translate both source and destination addresses. If both networks are really large, a lot of NAT rules will be required. Within the next chapter, we will see the possibility to route only one proxy server on the internal side with the use of a SOCKS V5 server.

2° Adding a SOCKS server

An Introduction to SOCKS

A SOCKS server is similar to a proxy server and has the same goals: to break the session transparently at an intermediate barrier isolating the internal addressing scheme from the external world.

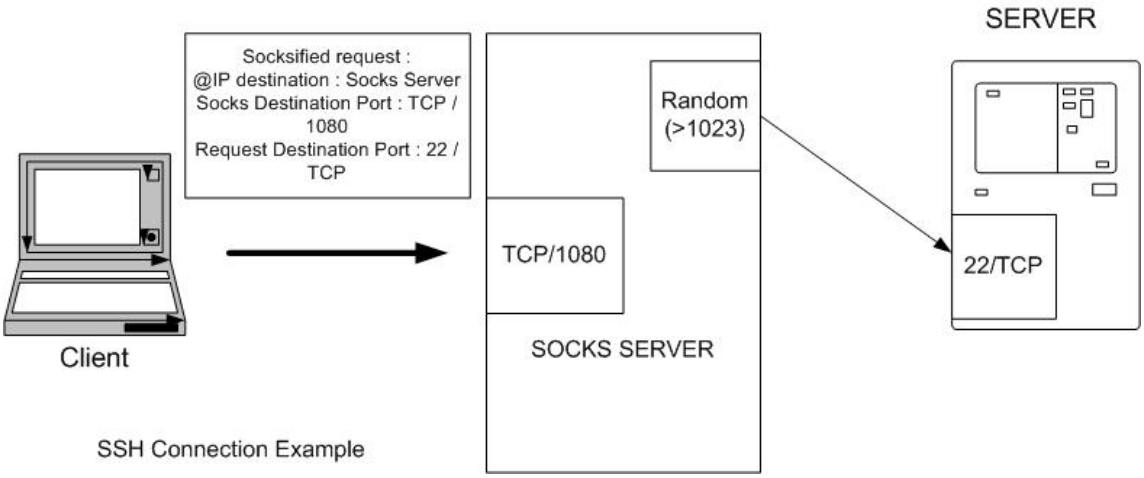
Whereas a standard proxy server is dedicated to some specific applications (most of the time HTTP) and can perform some caching operations, the SOCKS server just authenticates the user and redirects the traffic to the destination.

As the SOCKS service intercepts and redirects IP packets from client applications. They have to be "SOCKS enabled" (example: Internet Explorer) or the client workstation can also be "socksified" in order to have all TCP/IP applications working with socks. Only "socksified" clients are able to communicate with a SOCKS server via the SOCKS protocol. The SOCKS mechanism is transparent to the user, but a configuration file must be set on each client in order to know which SOCKS server it needs to connect to. (See Figure 5)

From a Network point of view, the main advantage is that we don't need to route the external network on the internal network, neither route the internal network outside external network as the socks client uses a configuration file which indicates the SOCKS server to use per destination subnet.

This kind of solution is pretty useful if you have really large corporate networks, and you do not want to add new routes within the Network.

Figure 5: Example of a SSH over Socks Connection



DANTE is one of the most known socks server implementation:
http://www.inet.no/dante/

Permeo provides a non-commercial socks client:
http://www.socks.permeo.com/Download/SocksCapDownload/index.asp

Security features of a Socks Server

There are two current versions of SOCKS protocol: SOCKS V4 and SOCKS V5. The two most important differences are:

- SOCKS V4 does not provide user / password authentication, while SOCKS V5 does. [5]
- SOCKS V5 supports UDP protocols (User Datagram Protocols).

From a security point of view, we will only keep the SOCKS V5 Protocol, as it provides network authentication. Even if you use V4 protocols, you can perform authentication on the Firewall level.

The SOCKS V5 protocol authenticates the client and may allow it whether to connect to an external service or not. This function is based on the IP addresses or/and the user name and password. With Socks V4, you will not be able to define rules on the socks per user, but only have possibility to define some shared rules based on the source IP. A SOCKS server can also log all established connections, just like a proxy does, in order to keep accountability evidences. [6]

Example of Socks V5 rule (socks5.conf):

Permit u – 10.0.1.1/255.255.255.255 192.168.0.0/255.255.0.0 - - Karen

The above rule allows the user-ID Karen, from her laptop with IP address 10.0.1.1, to connect to the 192.168.0.0/16 network.

Additional information on the socks5.conf syntax is available within:
http://www.socks.permeo.com/TechnicalResources/SOCKSFAQ/SOCKSvFFAQ/SOCKSvFManPage.asp

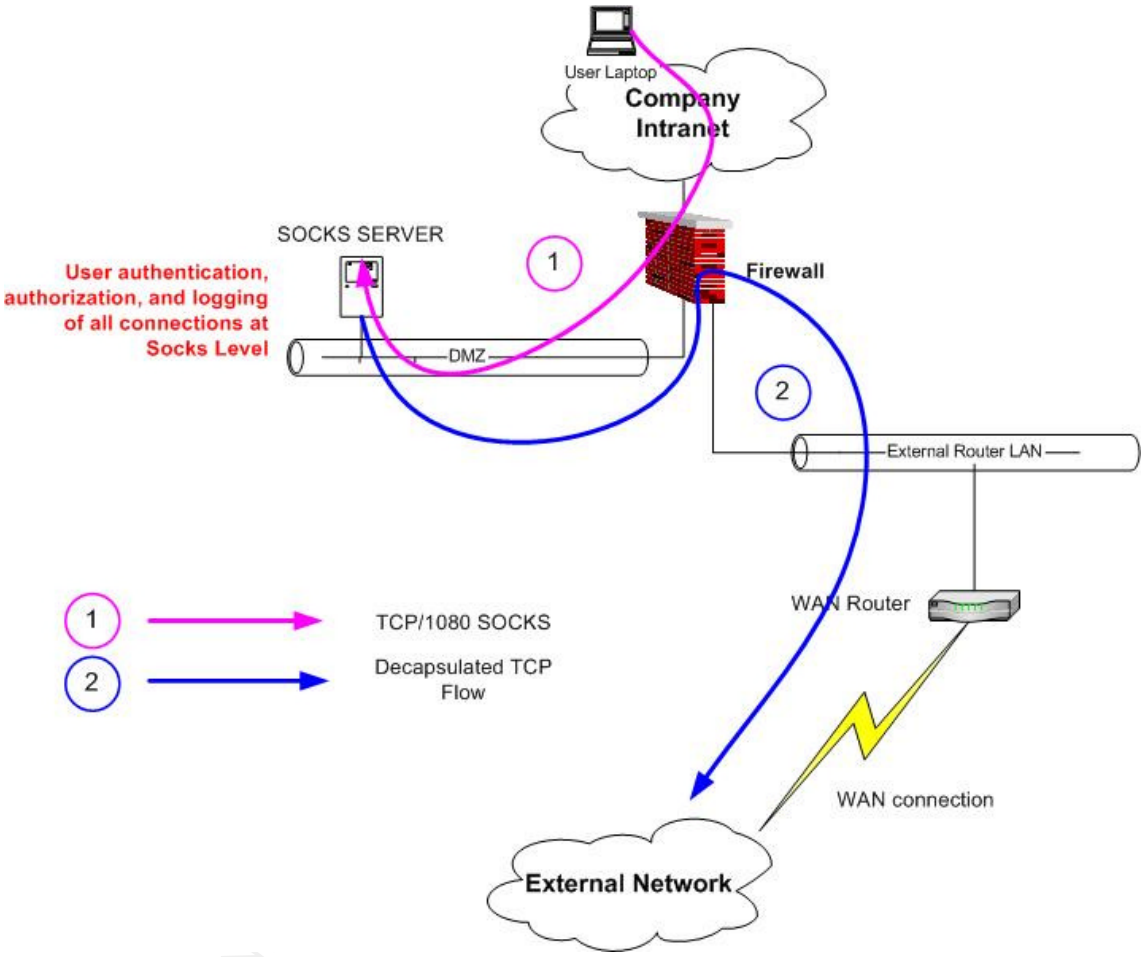
Some specific implementations of the SOCKS V5 also include:

> **Encryption support:** It is the case of the Aventail implementation (http://www.aventail.com), which provides SSL encryption. For free standard socks, such as DANTE http://www.inet.no/dante/, the user id and password remain in clear text in the packet,and can be easily sniffed. This problem can be solved by tunnelling Socks connection over SSH. [7]
>
> **External User Directory support:** One important point is to centralize UserID management. Standard SOCKS implementation was initially using flat files containing User-ID and password. New implementation now supports external directories such as LDAP. (See: http://www.inet.no/dante/FAQ/node29.html) External directory support can also help for implementing a two-factor authentication, such as Token based authentication.

Example of Socks Implementation

One standard implementation of socks server within a corporate network is when you have heavy network routing in place. In that case, you may want to grant access to an external network, without any deep modification. In that case, the socks will act as a proxy and will be the only server to know both internal and external networks.

Figure 6: SOCKS Model of Implementation



Consider an internal network that is based on the private IP address space. Users want to use an application protocol on the external network. Socks need to act as the application gateway to reach the external Network, because the external network is not routed nor translated internally.

The Socks server is located on a DMZ (**de**militarised **z**one) network. This DMZ is routed on both the external and internal Networks. User authentication we be performed on the socks to authenticate, authorize and log access to the external network with individual accountability. (Figure 6)

The Firewall is used to isolate the Internal Network, the DMZ, and the External Network. Firewall rules will be defined to limit access to the socks server on TCP/1080 port. Only IP addresses or users will be allowed to communicate to the

external network. (Flow in purple) In addition, the Firewall must control and limit access to the Socks server access by using some outbound rules. (Flow in blue). On a cost effective and less secure configuration, Firewall and socks software can run on the same device. In that case, we will have at least two network interfaces on this device.

Required Firewall Rules:

Source	Destination	Dst Port	Action
Internal Network	Socks Server	1080	Accept
Socks Server	External Network	Ex : Telnet	Accept
Any	Any	Any	Drop

- The first rule authorizes the Internal Network to access to the socks Server.
- The second rule allows the socks server to reach the external Network on port 23 (our example is Telnet).
- The last rule denies all other flows.

Required Socks Rules:

With the rule below, Karen cab establish telnet sessions towards the external network with authentication

Permit u – Internal Network/Netmask External Net/NetMask eq 23 Karen

Advantages and Drawbacks

The major security advantage of the SOCKS server is its access control mechanism which centralizes all access controls. In addition, from a network point of view, the use of socks server allows to reduce the amount of registered or routed IP addresses necessary when communicating with the outside world to only one. Therefore, Socks server facilitates more network routing than the NAT does.

There are still two major limitations: The need to have a client installed and configured on each workstation, and the lack of encryption support on non-commercial version to protect the user credential.

3° The "Stepping Stone" Server solution.

As indicated in its name, a stepping stone is a mandatory server where you need to logon first. It is used to establish connections to a destination which is not within the same network when a direct connection is not possible or not allowed.

This method is not transparent, since two steps are required for the user to connect:
1) The source establishes a secure connection to the appropriate stepping stone over the corresponding communication protocol. This includes an authenticated login.
2) Then, a second connection is established to the destination. If necessary an additional stepping stone can be involved.

There are two kinds of stepping stones used by the security community: the Windows Terminal server (WTS) and its Citrix Metaframe extension, and a UNIX box with SSH installed. You can also imagine a stepping stone based on VNC, Telnet, but they are not considered as secured protocols.

Citrix Metaframe

There are two alternatives for implementing a Terminal Server, use the Microsoft Windows terminal server or Citrix Metaframe.
Citrix Metaframe (http://www.citrix.com) is installed on a Windows Terminal Server to allow clients, to connect to the Citrix. Citrix Metaframe Client allows connecting with the use of several protocols, such as ICA (TCP/1494), HTTPS. The major difference is the possibility to "publish application" with Citrix ICA. All differences between RDP (Remote Desktop Protocol) 5.1 used on Windows 2000 / 2003 and ICA (Independent Computing Architecture) from Citrix are highlighted in this document: http://msdn.microsoft.com/embedded/devplat/thin/rdp_ica/default.aspx.yep

These Citrix Metaframe servers will be called "Metaframe step stones". ICA clients for the Metaframe servers are available on most operating systems.

The key advantages of the Citrix ICA solution are:

1) Published Application Mode:

The Citrix client can allow access to only specific Windows applications. This is known as "published applications". They are executed in a "seamless" mode which means they are integrated in the local windows manager on the user desktop. This is totally transparent. The end-user will see application windows on his laptop with the same classical look and feel, but it's running on the Metaframe Server (just like a remote X11 connection on UNIX). Also, it provides you also the possibility to use your local printer and clipboard.

The main advantage of published application mode is that the end-user has no access right at the Metaframe step stone desktop.

Access can be granted at user level for each published applications (By handling a list of authorized users: Explicit access). This feature offers the best way to secure

applications at user level. In addition to the ICA TCP port 1494, it requires to have HTTP or HTTPS allowed to browse and see the list of available applications.

More technical information's and tips [10] can be found on http://www.thinclient.net/.

2) Bandwidth Consideration:

The Citrix Metaframe can act as a "compression" gateway on the external network. It is pretty useful when only low bandwidth WAN links are available. Since most of the WAN traffic will be done through ICA protocol instead of standard graphical protocols, it allows reducing bandwidth usage.

For example, Citrix ICA communication typically requires 20 kb/s, when graphical protocols such as X11 require 80 kb/s or more. [8] These graphical protocols will be used for local traffic only. Therefore, the ICA servers will have to be located on the external network to optimize bandwidth.

Several Citrix Documentations can be found on the CCAHeaven.com [9] website, especially one from Citrix Consulting Services, related to Network bandwidth usage.

Bandwidth usage is also decreased by the Metaframe server acting as a staging area. When a software package is distributed to several servers on the external network, It is uploaded only one time on the WAN before being locally distributed from the Citrix server.

The drawback: One Major inconvenient of the Citrix solutions remains the cost needed for Server hardware and Software Licenses compared to an open source solution such as Dante Socks running on Linux.

Accountability of each user on the Citrix is maintained through the Windows event log by implementing the process tracking. It will be more limited than the other solutions as correlation between firewall logs and Windows Event logs will be required to know "who has done what". The standard example is when you have more than one user connected to the Citrix: you will need to identify who have started the application by using the Windows event log and correlate with the Firewall logs to obtain a correct accountability level.

I also strongly recommend to implement limited NTFS access (Read /Execute) to the user and everyone group on most of the systems resources, and to activate auditing of each NTFS ACL, in order to monitor activity (especially failure events) on systems files by general user. The Chapter 15, Metaframe XP Security Design, of the Brian Maden books will help you implementing required security controls. [11]

Example of Citrix Implementation

Let's take the Firewall authentication example.
We will just change address translation requirements, as explained on Figure 7 (see next page):

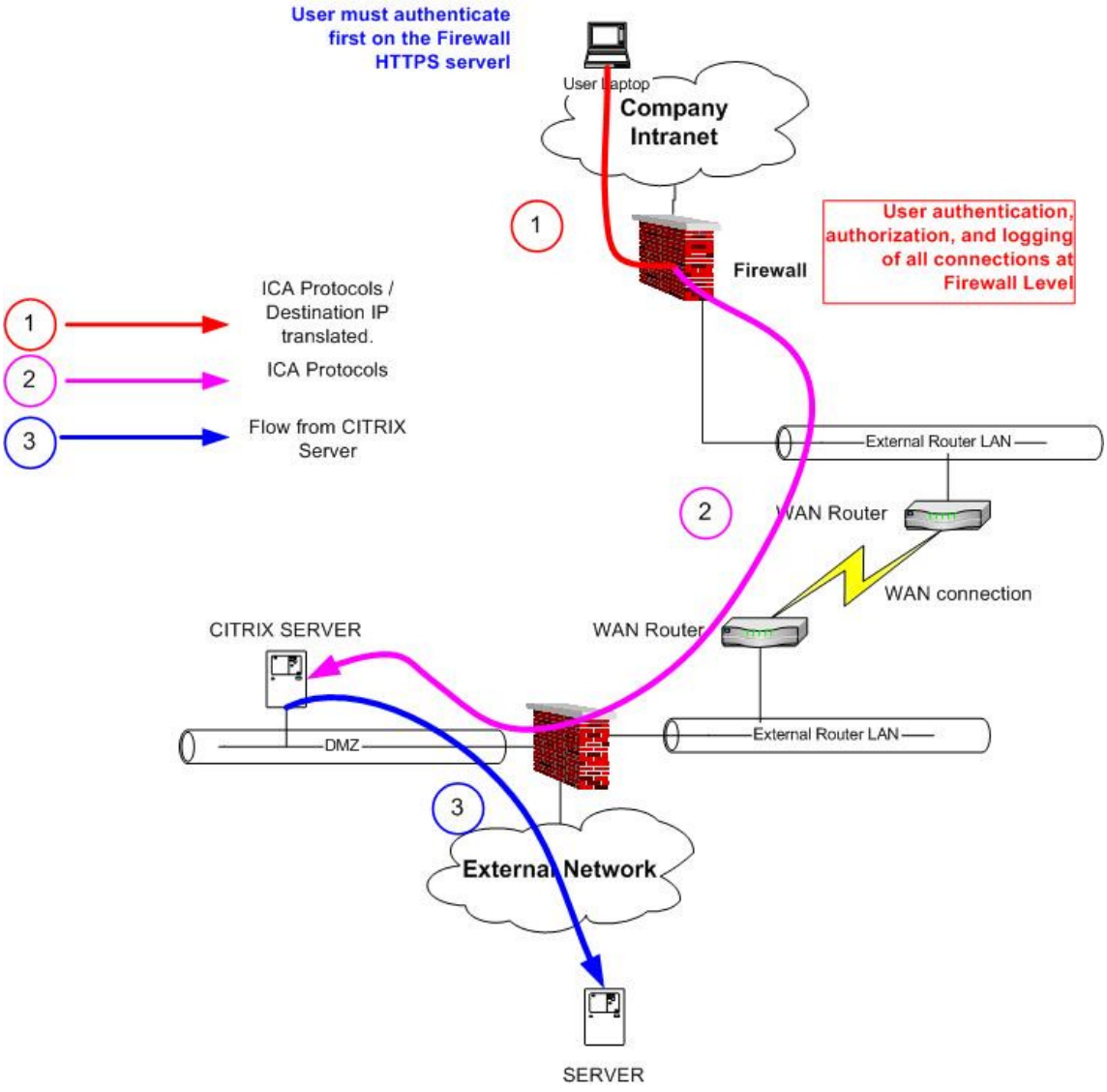
1. We do not need anymore source address masquerading: The Internal Network will be routed on the two firewalls.
2. We add a static translation of the Citrix server address in order to avoid any routing of the external DMZ over the internal network.

In addition, we will only have small amount of bandwidth available on the WAN link (Lower than 128kb/s), even if we use heavy bandwidth applications.

To solve this problem, we can implement a new DMZ on the external network, and place a Citrix Metaframe on it. All these applications will be installed on the Citrix Server and will be published. User will access the Citrix, and will get the required published applications, with an acceptable response time.

As already discussed, accountability on the Citrix Metaframe will be obtained by the use of the Windows Event logs, and the process tracking in correlation with the external firewall log.

Figure 7: Citrix Stepping Stone: Example of Implementation



Required Firewall Rules:

Firewall n°1 connected on the Internal Network:

Source	Destination	Dst Port	Action
Internal Network	Firewall	900	Accept
User Laptop@Internal network	Citrix Server (translated address)	80/TCP & 1494 TCP	Client Auth
Any	Any	Any	Drop

- The first rule authorizes the user on the internal network to reach the authentication server on the Firewall.
- The second rule allows authenticated user to reach the Citrix Server located on the external DMZ after authentication. Port 1494 is the ICA protocols. The HTTP is required to perform application browsing. The authorization is performed on the Citrix translated address.
- The last rule denies all other flows.

Firewall n°2 connected on the external Network:

Source	Destination	Dst Port	Action
Internal Network	Citrix Server (DMZ address)	80/TCP & 1494 TCP	Accept
Citrix Server	External Network	Ex : TELNET / FTP	Accept
Any	Any	Any	Drop

- The first rule authorizes the User Laptop to connect to the Citrix server. This time, it is performed on the correct Citrix address
- The second rule allows the Citrix Server located on the external DMZ to connect to the external Network on port 23 and 20/21.
- It will allow users connected on the Citrix to perform their File transfer and Telnet login to the external server.

Required Address Translation on Firewall n°1:

Source	Destination	Dst Port	Translated Source	Translated Destination	Translated Port
Internal Network	Citrix Server (Routed address on Internal Network)	Any	Internal Network	Citrix Server (DMZ Address)	Equal

The Citrix Server is translated on the internal Network with a routed internal IP address.

Published Application Requirement:

The Citrix server will have a Telnet client and a FTP client published on the Citrix. Others tools will be installed and published in order to manipulate the large files directly on the Citrix server and without sending it back over the low bandwidth WAN link.

The SSH Step Stones.

SSH, abbreviation of Secure SHell is a program used to login to another computer over a network, and to execute commands on this remote computer. It provides strong authentication and secure communications over insecure channels, such as the Internet.

Initially a replacement for the extremely insecure Berkeley remote (r*) tools (rsh, rcp, rlogin, etc), SSH can also be used as a telnet substitute. While originally developed for UNIX platforms, it has been ported to a number of popular operating systems, including Windows.

To the user, SSH is a computer program that looks like telnet. The SSH client for Windows/NT provides essentially the same look and feel as the Microsoft NT version of telnet. But, unlike telnet, however, SSH provides strong authentication and secure (encrypted) communications between computers. [12]

SSH is composed of several commands and configuration files that provide the following functions:

Authentication & Encryption: On top of the classical UserID / Password, SSH supports public and private key pairs for authentication and encryption of channels to ensure secure network connections. To provide a stronger authentication mechanism, the private key is used to decrypt the connection data, which is sent by the remote peer, and encrypted with the local peer's public key. The public key mechanism is a good solution to replace the unsecured .rhost file or to automate some network scripts securely. [13]

Port forwarding: (tunnelling): This is one of the major reasons why we can use SSH as a stepping stone. SSH provides tools for forwarding TCP/IP communications within a secure tunnel. This procedure is commonly referred to as port forwarding or tunnelling. When services are tunnelled, they are secure in the sense that outside the tunnel it is impossible to determine what information a tunnelled service is transporting, as it is encrypted like a VPN. [14]

But, tunnelling usage can be also dangerous, especially for the inbound tunnel, as there is no way to control what takes place under the TCP/22 flow, so it can provide ability to bypass all firewalls policy. When establishing tunnels, you need to make sure that client setting such as "allow remotes hosts to connect" is not be used. If your tunnels are accessible for others machines, you may create some big holes across your Corporate Network. So, be careful with the usage of SSH across Firewall and different security zone.

Therefore, one recommendation will be to prohibit tunnels or encapsulated protocols to pass directly from the Internal Network to the External Network. I do not personally recommend using SSH as a stepping stone for a large amount of users, but only accept SSH connections across the same Network, or at least limit incoming flows to the SSH server.
On the other hand, if the amount of people is small and can be fully trusted, then SSH is a true Swiss knife.

Conclusion

As you have seen, there are many different ways to perform authentication across the network. If your firewall provides the ability to perform authentication, it should be the first step.

You should introduce a proxy based solution, such as the socks server, if your internal LAN is really large, and the population accessing the external network is small. It will avoid most of the network routing problems.
At least, if there is only a small WAN bandwidth amount available, or you need to manipulate heavy data on the external side, Step stone solution, like Citrix or SSH, must be considered as the appropriate approach.

In all cases, these solutions have the ability to use an external user directory, centralize user management, or support stronger authentication mechanisms. It is vital to manage your user directory correctly, and perform periodic controls, such as business and employment revalidation of user accounts and access rights. This is a mandatory procedure to put in place around those technical authentication solutions.

REFERENCES

1. Kjeld Borch Egevang & Paul Francis, Network Address Translation RFC 1631, May 1994,
http://www.faqs.org/rfcs/rfc1631.html

2. Scott Jensen, Cisco PIX Authentication and Cisco Secure ACS, April 6th 2002
http://www.giac.org/practical/scott_jensen_gsec.doc

3. Check Point™ VPN-1/FireWall-1® Administration Guide:
http://www.checkpoint.com/support/technical/documents/docs-5.0/firewall_ng_sp0.pdf

4. The Authentication Phone Boy FAQ
http://www.phoneboy.com/bin/view.pl/FAQs/AuthenticationFAQs

5. Marcus Leech, Username/Password Authentication for SOCKS V5 RFC1926, March 1996,
http://www.ietf.org/rfc/rfc1929.txt

6. DANTE Socks server documentations and FAQ
http://www.inet.no/dante/,

7. Jason Boxman, Configuring a SOCKS Proxy and Tunneling, September 2003
http://www.trekweb.com/~jasonb/articles/dante_tunnel.shtml

8. CCAHeaven.com,
http://ccaheaven.com/metaframe2.htm

9. Citrix Consulting Services, Maximizing ICA Sessions with limited Network Bandwidth,
http://ccaheaven.com/wps/Maximizing_ICA_Sessions_with_limited_Network_Bandwidth.pdf

10. ThinClient.net Website,
http://www.thinclient.net/

11. Brian Madden, Citrix Technical Design Books,
http://brianmadden.com/books/bookdetails.asp?isbn=0971151032

12. Boran Consulting Publications, All about SSH Part 1 & 2, February 2000,
http://www.boran.com/security/sp/ssh-part1.html
http://www.boran.com/security/sp/ssh-part2.html

13. WindowSecurity.com, SSH, July 19th 2002,
http://www.windowsecurity.com/articles/SSH.html

14. University of Leuven, SSH as a VPN, October 2002,
http://www.cs.kuleuven.ac.be/system/security/ssh/vpn.shtml