Perimeter Security - Protecting Your Network: SOCKS, Proxy Servers, and Firewalls

By Chris Mayers
Notice

The information in this publication is subject to change without notice.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. CITRIX SYSTEMS, INC. ("CITRIX"), SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR DIRECT, INCIDENTAL, CONSEQUENTIAL OR ANY OTHER DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS PUBLICATION, EVEN IF CITRIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES IN ADVANCE.

This publication contains information protected by copyright. Except for internal distribution, no part of this publication may be photocopied or reproduced in any form without prior written consent from Citrix.

The exclusive warranty for Citrix products, if any, is stated in the product documentation accompanying such products. Citrix does not warrant products other than its own.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Copyright © 2000 Citrix Systems, Inc., 6400 NW 6th Way, Ft. Lauderdale, Florida 33309 U.S.A. All rights reserved.

| Version History |
|-----------------|-----------------|------------------|
| January 1, 2000 | Chris Mayers    | Version 1.0      |
# TABLE OF CONTENTS

INTRODUCTION ................................................................................................................................. 1  

PERIMETER SECURITY ....................................................................................................................... 1  
  FIREWALL TRAVERSAL .................................................................................................................... 2  
  PROXY SERVERS .......................................................................................................................... 3  

ABOUT SOCKS .................................................................................................................................... 5  
  SOCKS AUTHENTICATION ............................................................................................................. 5  
  SOCKS CLIENT IMPLEMENTATIONS ............................................................................................. 5  
  SOCKS SERVER IMPLEMENTATIONS ........................................................................................... 7  
  EXAMPLE CONFIGURATION 1 - OUTBOUND ACCESS CONTROL WITH SOCKS .................................. 8  
  EXAMPLE CONFIGURATION 2 - INBOUND ACCESS CONTROL WITH SOCKS .................................. 9  
  EXAMPLE CONFIGURATION 3 - CHAINED SOCKS SERVERS FOR MULTI-SITE OPERATIONS (EXTENDED INTRANET) ................ 10  
  CONFIGURATIONS INAPPROPRIATE FOR SOCKS ..................................................................... 11  

SUMMARY ......................................................................................................................................... 11
Introduction

Security is not a single product or technology. Securing a Citrix product installation requires a combination of Citrix software, third-party software and hardware, together with best-practice operational policies and procedures.

This White Paper discusses perimeter security: the security measures applied at the boundary of a network which control traffic to and from that network.

Different organizations have a wide range of perimeter security policies, reflecting their different business needs. These security policies are implemented as a wide range of deployment scenarios. This White Paper illustrates some of these deployment scenarios, but does not aim to be a comprehensive guide to planning deployment. For detailed guidance, refer to Citrix product documentation, support information, and best practices documentation.

Perimeter Security

A major benefit of Citrix products is that they enable access to applications from anywhere. Such access must then be controlled depending where the service is being offered.

For example, an application could be provided by:

- an organization on its internal network (intranet)
- one part of an organization to another part, on a different site (extended intranet)
- an organization to another organization (extranet)
- an organization to many other organizations (internet)

A natural place to enforce controls is at the network perimeter, with a firewall. The general case (which covers all these examples) is of two organizations: a client-side organization with a firewall controlling outgoing access, and a server-side organization with a firewall controlling incoming access. A secure connection is needed between the two firewalls, and this could be wide-area public network such as the Internet.
Each organization enforces its own security policy within its own management scope. So the most general case is this:

**Perimeter Control**

![Perimeter Control Diagram](image)

Figure 1: Perimeter Control

**Firewall Traversal**

Firewall administrators control the traffic across the firewall according to their security access control policy. The most important factor in the policy is the service being offered: that is, the network protocol, for example ICA.

Enforcing the security policy is easiest if the traffic passes through a single gateway, sometimes known as a ‘choke point’. A firewall is often used in conjunction with a proxy server to form a gateway.
Proxy Servers

Even if the firewall is enforcing the access control policy, many network administrators do not wish to permit a direct connection via the firewall.

They require a second line of defense. The firewall ensures that only permitted protocols can pass across the network, but a second line of defense allows access to specific services to be controlled without reconfiguring the firewall.

One way is to place a proxy server between the client and the application server. Instead of the client connecting to the server, it connects to the proxy server; the proxy server then connects to the application server.

Figure 2: Proxy Server

Proxy servers are widely used within the Internet to cache frequently used web pages.
Proxy servers can be chained to other proxy servers: this is sometimes known as server-to-server proxying. This allows two organizations which each use a proxy server to connect them together:

**Figure 3: Proxy Server Chaining**

The proxy server is transparent to the application server. The application server cannot distinguish between a connection from a proxy server, and a connection from a client. However, proxy servers are not transparent to clients:

- client programs need to be modified to handle proxy servers
- client programs need to be configured with the address or name of the proxy server

Some network protocols have explicit support for proxy servers; HTTP (HyperText Transfer Protocol) does. Other protocols do not have explicit support for proxy servers; Citrix ICA does not.

Since Citrix ICA does not have explicit support for proxy servers, a general-purpose protocol for interacting with proxy servers is needed. This is the purpose of the SOCKS protocol.
About SOCKS

SOCKS is a protocol for authenticated firewall traversal. It allows a SOCKS-enabled client to connect via a SOCKS server to an application server. A SOCKS server is a particular kind of proxy server, sometimes called a circuit-level gateway.

The name "SOCKS" is a historical quirk; it was an abbreviation of "SOCKetS" and was intended to be superseded by a more meaningful name, but this never happened.

SOCKS uses TCP port 1080 by default. Many firewalls have this port preconfigured.

SOCKS is not a secure channel protocol such as SecureICA, SSL (Secure Sockets Layer), TLS (Transport Layer Security), or IPSEC (IP Security). Secure channel protocols are for encryption and integrity, not for firewall traversal. SOCKS can be used with secure channel protocols or independently.

SOCKS is a collection of Internet standards. The original version of SOCKS, version 4, was never made an Internet standard, but has been widely implemented. Version 4 does not support authentication. It supports TCP connections, but not UDP datagrams.

SOCKS version 5 adds support for authentication. It also adds support for UDP.

SOCKS version 6 is at a early stage of development as an Internet standard. It is now known as Secure Transport Proxy (STP). Implementations in the near future are not expected.

To exploit the features of SOCKS version 5 and above, both the client programs and the SOCKS server must support SOCKS version 5.

SOCKS authentication

SOCKS version 5 has flexible support for authentication. The SOCKS version 5 protocol allows the client and the SOCKS server to negotiate a mutually acceptable authentication method. Various authentication methods are being standardized, including username/password and challenge/response methods. There is also provision for private (proprietary) authentication methods. However, most of the proposals for SOCKS authentication methods have not completed the Internet standardization process.

SOCKS client implementations

Client programs generally need to implement explicit support for SOCKS. (This process is sometimes known as socksification.)

The memory overhead for SOCKS support in the client program is small: a few kilobytes. Bandwidth overhead is also small. Latency overhead will depend on the SOCKS server; any proxy server will add some latency since traffic must travel via the proxy.

Examples of client programs explicitly supporting SOCKS are:

- Citrix ICA Client
- Hummingbird SOCKS Client
- Microsoft Internet Explorer
- Netscape Navigator/Communicator

On installation, clients must then be configured with the location of the SOCKS server. For example, in Microsoft Internet Explorer 4, the SOCKS server is configured in View/Internet Options…/Connection/Proxy server/Advanced…

For client programs that do not explicitly support SOCKS, an alternative approach may be available on some platforms. Suppliers offer a SOCKS support client module which replaces a corresponding module of the platform network protocol stack. On some platforms, the SOCKS support client module may insert itself into the network protocol stack, rather than replacing part
of it. On Microsoft Windows platforms this is usually a replacement for the Windows Sockets (WinSock) library, using the LSP (Layered Service Provider) architecture.

Examples are:

- Aventail Connect (formerly AUTOSOCKS/VPN Client)
- Microsoft Winsock Proxy Client for Microsoft Proxy Server
- NEC SocksCap

The advantage of this approach is that it is transparent to all client applications. The disadvantage is that the SOCKS support client module is invasive: it may interact unexpectedly with the network protocol stack, and there may be restrictions on some application features. In any case, this approach can only be used on a few platforms.

Therefore Citrix decided to explicitly support SOCKS in ICA Clients. Customers who are already using a SOCKS support client module with an ICA Client can continue to do so, or can upgrade to the latest ICA Client with SOCKS support, as it becomes available. SecureICA clients will also support SOCKS.

Citrix ICA Clients support both SOCKS version 4 and SOCKS version 5, but do not support SOCKS version 5 authentication. Citrix is evaluating customer demand for SOCKS version 5 authentication, and is monitoring the outcome of the Internet standardization process for the various SOCKS authentication methods.

Figure 4: Citrix support for SOCKS
SOCKS server implementations

A SOCKS server can be implemented in several ways:

- as part of a firewall (for example, the IBM eNetwork Firewall)
- as part of a general-purpose proxy server (for example the Microsoft Proxy Server or Netscape Proxy Server)
- as an independent server (for example, Aventail ExtraNet Center)

Many customers who already use a general-purpose proxy server for caching access to the Internet, can use the same server for SOCKS. Not all general-purpose proxy servers support SOCKS.

Different SOCKS servers also support different versions of SOCKS; some support SOCKS version 4 only, some SOCKS version 5 only, some both. Those that support SOCKS version 5 may support only a subset of SOCKS version 5 features.

Citrix does not supply a SOCKS server.

Figure 5: Socks Requires a Firewall

Needed to prevent clients bypassing the SOCKS server
Example Configuration 1 - Outbound access control with SOCKS

An organization wishes to control internal user access to external Citrix servers. The organization already uses Microsoft Proxy Server 2.0 for caching access to the Internet. The firewall is configured to accept connections only from Microsoft Proxy Server.

Microsoft Proxy Server 2.0 supports SOCKS version 4. Since SOCKS version 4 does not support UDP, ICA Browsing functionality does not pass through SOCKS.

In this example, the Client platforms are Windows 95 (using the ICA Client for Win32) and Windows for Workgroups 3.11 (using the ICA Client for Win16).

The planned firewall configuration requires that outgoing traffic on TCP port 1494 (ICA) is permitted, but only from the Microsoft Proxy Server:

**Outbound Access Control with SOCKS**

![Diagram of Outbound Access Control with SOCKS](image)

*Figure 6: Outbound Access Control with SOCKS*
Example Configuration 2 - Inbound Access Control with SOCKS

An organization wishes to control external user access to internal Citrix servers. The organization already uses Netscape Proxy Server 3.5 to control incoming connections.

In this example, the Client platforms include web browsers using the Citrix ALE⁠¹ clients, Netscape Communicator (using the Citrix ICA Client for Java), and Microsoft Internet Explorer (using the Citrix ICA ActiveX Control). The organization already uses Netscape Enterprise Server as a secure web server using HTTPS; the Citrix ALE clients will be downloaded from there.

The planned firewall configuration requires that incoming traffic on TCP port 1080 is permitted, but only to the Netscape Proxy Server. It also requires that incoming traffic on TCP port 443 (HTTPS) is permitted, but only to the Netscape Enterprise Server:

---

¹ ALE -- Application Launching and Embedding. A feature of Citrix® MetaFrame® and WinFrame® products that enables full-function, Windows-based applications to be launched from or embedded into HTML pages without rewriting application code.
Example Configuration 3 - Chained SOCKS Servers for Multi-Site Operations (Extended Intranet)

An organization wishes to provide access from users at one site to a Citrix server at a central site. The organization already uses Aventail ExtraNet Center.

In this example, the Client platforms include Windows NT desktops, and WBTs (Windows Based Terminals).

An additional copy of Aventail ExtraNet Center is required on the client site.

On the client site, the planned firewall configuration requires that outgoing traffic on TCP port 1080 is permitted, but only to the other site. Similarly on the server site, the planned firewall configuration requires that outgoing traffic on TCP port 1080 is permitted but only from the other site.

The two copies of Aventail ExtraNet Center can be configured to encrypt the traffic, effectively providing a VPN (Virtual Private Network) for the ICA connection:

Figure 8: Extended Intranet Support with SOCKS
Configurations Inappropriate for SOCKS

Some organizations are unable to use SOCKS. This may be because:

- the firewall does not support SOCKS, and they do not wish to install a separate SOCKS server.
- they do not permit use of TCP port 1080 (SOCKS) via the firewall.
- they already enforce authentication at the firewall using a proprietary firewall authentication mechanism which the SOCKS protocol cannot exploit.

Citrix is aware of the need to support such configurations, and is evaluating supplementary and alternative solutions to SOCKS.

Summary

SOCKS is a flexible standards-based firewall traversal protocol for perimeter security. SOCKS support in ICA clients, in conjunction with third-party SOCKS server products, will support many deployment scenarios.