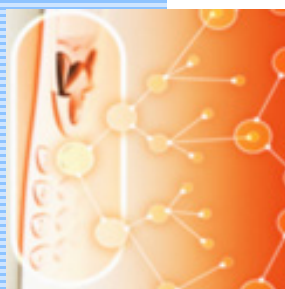



[Home](#)
[About Us](#)
[Products](#)
[Support](#)
[Evaluate](#)
[Search](#)

## Firewall and Proxy Primer

[Home](#) : [Support](#) : [Resources](#) : [Proxies and Firewalls](#)



### ***Client-side firewall and/or proxy configuration for CollabWorx RTC services***

This document describes required firewall and/or proxy settings that are necessary for corporate users separated from Internet by a firewall to use CollabWorx Real-time conferencing services. This memo focuses on the client side.

Network diagram highlighting necessary network elements is included [later in this section](#).

A frequently asked question in the context of on-line collaboration is: why cannot collaboration services use only HTTP/HTTPS protocols? The answer is the same as to the question: "why don't we travel to moon on a bike?": A bike is as good for space travel as HTTP protocol is suitable to implement high-performance, real-time collaboration. The vendors advertising their collaboration services as only using HTTP **are in reality admitting to mediocrity and inadequacy of their solutions**. It is not possible to encapsulate all protocols into HTTP/S and expect **rich functionality, stability, scalability, and performance** of a real-time communication application. The popular trend to block all connections on the firewall and constrain corporate connectivity to HTTP/S (AKA "port 80 dogma") is equivalent to invalidating the entire body of work by [IETF](#), [ITU](#), and [W3C](#) on Internet protocols.

CollabWorx RTC does not support HTTP-only communication and there are no plans to introduce such a solution. This article's main goal is to

help network administrators required to support real-time communication and collaboration services to correctly assess security impact of such services and to soothe their fears that on-line RTC may destabilize or expose their networks. We want to demonstrate that such fears are unfounded and that the problems can be elegantly and efficiently solved while maintaining desired security and control over the network.

### Firewall-friendly solutions from CollabWorx

Traditional conferencing solution based on H.323 and T.120 standards are very difficult to implement securely. These protocols have been designed when network security was an afterthought. As a result, they require multiple TCP ports and nearly a full range of dynamically assigned UDP ports. This is indeed undesirable.

In security context, UDP transport is more difficult to handle than TCP. The difference lies in how connections are initiated. With TCP a connection initiated from **within** firewall to an outside server (**outgoing connection**) supports bi-directional traffic. This means that it is not necessary to let outsiders to establish connection to corporate network (**incoming connections**) to be able to receive e.g. audio and video streams from them. With connectionless UDP transport the firewall must make a packet-by-packet decision to admit A/V streams without necessarily knowing if such streams have been actually requested by an internal user. For this reason UDP is treated with considerable mistrust by network security administrators. This is unfortunate since elimination of UDP implies elimination of multicast service, but it seems that, at present, security considerations are prevailing.

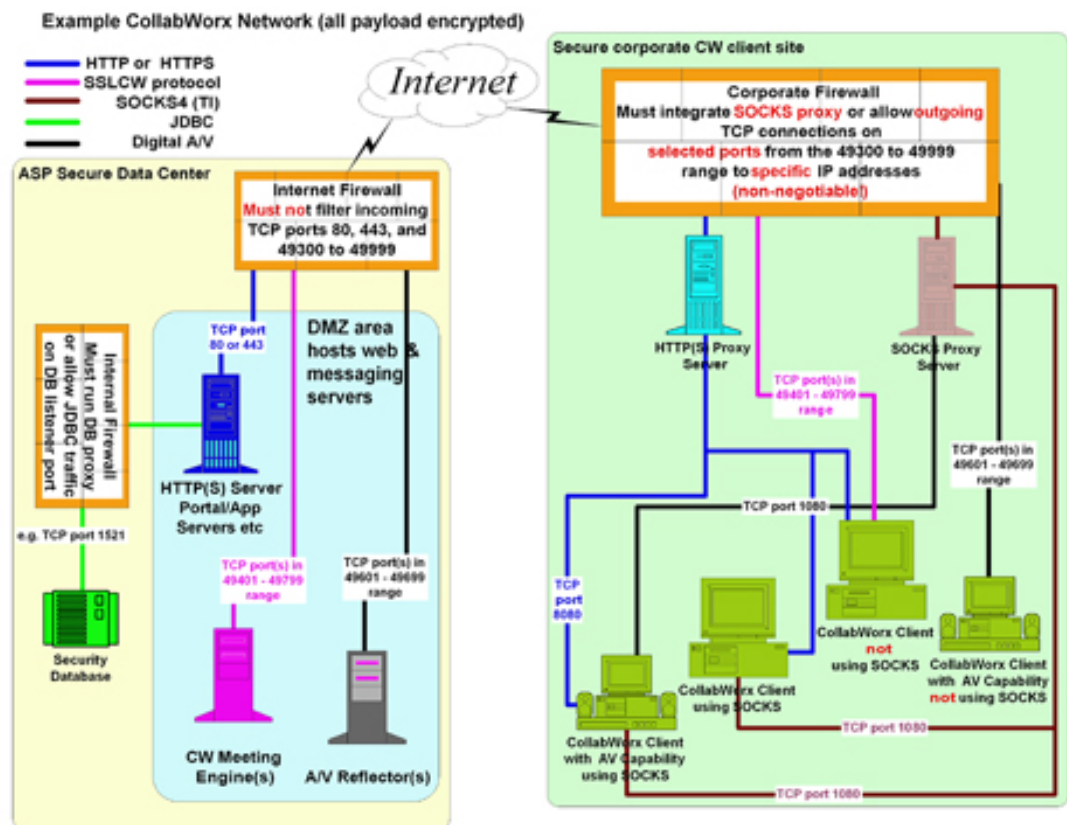
CollabWorx RTC products support three network topologies for data-intensive applications such as audio/video conferencing: multicast, UDP-based P2P unicast, and a central retransmitter ("reflector") topology using TCP transport. Given firewall interaction considerations, multicast and UDP P2P topology is applicable to Intranet situations. Most Internet-based deployments opt for the TCP-based reflector network topology.

Using TCP reflector solution is firewall friendly. As per the network diagram (see below), the CollabWorx real-time messaging servers should be collocated with corporate web servers in the so-called Demilitarized Zone (DMZ). This segment of the network must allow connections across Internet firewall to all services hosted in DMZ. The

port numbers of the CollabWorx services are shown in the diagram. Please, note that any particular implementation may be using as few as two or three of the TCP ports in the indicated range, although massive deployments may use nearly all of the ports. It is important to understand that the port range used by CollabWorx has been carefully and deliberately selected. First, it conforms to the [IANA recommendation](#) of port selection for private applications. Second, there are no “well-known” Trojan horses in this range. Please, consult [this document](#) for detailed explanation of the port selection range.

With reflector topology for data intensive applications CollabWorx clients communicate via a set of messaging engines residing in the data center of the service provider (in-house or an ASP). CollabWorx client software only needs to make outgoing TCP connections via corporate firewall to establish service access. No incoming connections are necessary. CollabWorx software works with both [NATed](#) and Internet IP addresses on corporate Intranets with no special steps to be taken.

## Firewalls and Proxies



[Click for a large image](#)

In general, two solutions are available to enable connectivity to

## messaging servers:

1. Firewall administrators can open necessary outgoing TCP connections on the corporate firewall. It is recommended that the rules controlling this connectivity are limited only to necessary (actually used) ports and to the known IP addresses of machines running CollabWorx messaging servers. A little more relaxed approach is to allow a range of ports and a domain name to which messaging servers belong. Neither of these methods introduces any significant security risk for the corporate network, or opens access to undesirable services that the corporation would rather eliminate from the Internet menu available to its employees.

This solution is illustrated on the [diagram](#) by connectivity pattern of client machines labeled as "CollabWorx client **not** using SOCKS".

2. Proxy servers are incorporated into firewall infrastructure. Corporations commonly use HTTP/S proxy servers to control and monitor access to the Web by their employees. [SOCKS proxies](#) are available to support similar functionality for arbitrary protocol. SOCKS proxy default port is TCP 1080. A firewall with integrated SOCKS proxy should block all outgoing connection except connection to arbitrary TCP ports originating from the machine running SOCKS proxy server. More often than not SOCKS proxy is a part of the HTTP/S proxy.

This solution is illustrated on the [diagram](#) by connectivity pattern of client machines labeled as "CollabWorx client using SOCKS".

To use HTTP/S proxy, the browsers used by all company employees must be instructed to do so. The configuration GUI used to set up HTTP proxy usually also allows users to configure SOCKS proxy. The application that intent to use SOCKS protocol need to be SOCKS aware. Most of the CollabWorx applications are; those that yet have not been SOCKS-enabled may use a small utility known as [SOCKSCap](#). Installation of this utility allows the user to select any of the network applications and direct it to use SOCKSCap wrapper.

Which of the above solutions is better? It depends. Both solutions (if properly implemented) help to maintain very high network security. Solution 1 is better for overall system performance and helps keeping audio and video conferencing delays at minimum and media quality at

maximum. It is also cheaper as proxy server software needs to be purchased and maintained. For organizations placing high value on traffic monitoring and management proxy servers offer these capabilities and are hence recommended.

### Security impact

How is implementation of the firewall solutions described above impacting network security?

In both cases the impact is minimal. [Solution 1](#) is, in principle, slightly less secure. General impact is low since no ports for incoming connections are being open, i.e., no additional access initiated outside of the corporate network to assets inside the firewall is created. Blockage of outgoing ports serves two general purposes (1) it limits number of ways corporate users have to transfer out privileged documents in uncontrolled fashion; (2) it prevents corporate users from accessing undesirable services, such as entertainment sites, especially those serving multimedia contents. This practice is aimed at limiting both employees' distraction and unwanted network traffic. Note that none of these is strictly security related, although access to certain consumer-grade IM systems opens a way for users to transfer files bypassing the rules a corporation might have on permissible e-mail attachments.

In the context of CollabWorx software this loss of security is mitigated as follows:

1. CollabWorx Secure Instant Messenger (SIM) does not support, per design, file transfer capability. SIM has been designed from ground up as a business tool and it eliminates all the factors that make consumer grade instant messengers a security threat and a distractive factor. For detailed discussion of these issues please consult [SIM whitepaper](#).
2. Port numbers used by CollabWorx software are not used and, according to [IANA recommendations](#), should not be by any other applications. Hence, by opening the ports used by our software the corporation is very unlikely to enable access to any unwanted applications that might distract the employees or increase unauthorized network traffic. This cannot be entirely ruled out however. If this is an important concern we recommend implementation of proxy servers ([solution 2](#)).
3. CollabWorx messaging servers do not serve any persistent data -

they are merely route and distribute messages.

4. CollabWorx [secure messaging and authentication servers](#) are independent and usually not collocated. Enabling access to the ports allowing communication with messaging servers is not sufficient to gain access. The users must also obtain credentials from the authentication server. CollabWorx authentication server (SafePerimeter) is a Web service accessible via HTTPS on standard TCP port 443. This means that in spite of opening necessary outgoing ports only the authorized users will be able to connect to RTC services.
5. One of the popular methods of attacking network security is to spoof the servers. With respect to CollabWorx there are following obstacles to do so: (1) there is no public domain software with equivalent functionality. Spoofing party would have to obtain original software and corresponding license (CW servers are IP-locked); (2) The servers communicate with authentication server to complete authorization process. All accesses to authentication servers are logged so that all rogue requests for confirmation will be immediately detected and denied. CW clients will not complete authentication cycle without receiving server confirmation in a proprietary format and including information that can only be received from authentication server. (3) Attacks via substituting a reverse-engineered messaging server will fail without the server also obtaining a digital certificate attesting to its legitimacy. Such certificates are unforgeable and would have to be stolen from CollabWorx.
6. Limiting access via firewall to specific IP addresses known to be associated with a legitimate CollabWorx service reduces the risk to astronomically small.

Hence, a scenario enabling hijacking of the CollabWorx services would necessarily involve (1) hijacking of the DNS server pointing to CollabWorx messaging servers; (2) theft and installation of the server software as well as certificates and software licenses, or (3) reverse engineering of proprietary aspects of CollabWorx meeting setup protocols; (4) theft of the data used to authenticate users of CW services and independent theft or hijacking of the CollabWorx authentication and authorization server. If the CW service setup does not involve DNS names of the servers but rather uses IP addresses, no attack scenario other than a physical seizure of the data center is conceivable. This, of course, remains to be a Hollywood-class event so far.

The drawback of the [solution 1](#) is that the corporation might want to monitor and manage access even to a legitimate service. Installation of proxy servers provides this functionality. All security mechanisms of solution 1 are still in force as proxy servers may be set up to cooperate with firewall as described above. Proxy servers permit tightening up of the firewall rules by only permitting outgoing connections from the machines running proxy server software. Proxy servers can also be used to log all connections, selectively disable user access, or implement time restrictions.

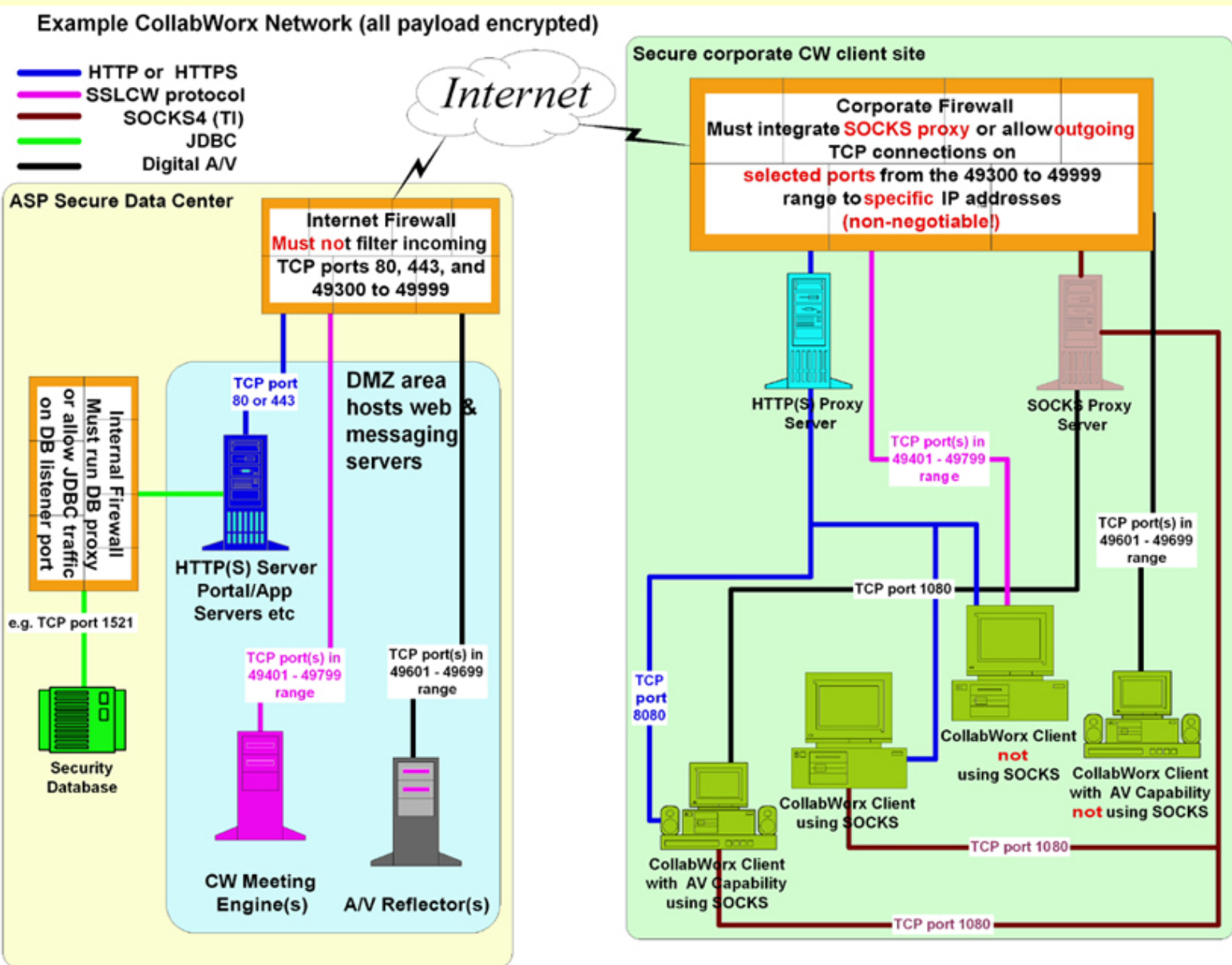
### **Need help?**

CollabWorx provides [professional and consulting services](#) for all your network configuration and security needs related to all types of collaborative applications.

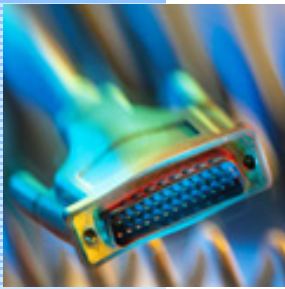
[\[Home\]](#) [\[About Us\]](#) [\[Products\]](#) [\[Downloads\]](#) [\[Search\]](#)

Copyright © 2000 CollabWorx, Inc. All Rights Reserved  
[Privacy Policy](#) | [Contact CollabWorx](#)

[Print this image](#) (landscape page orientation is recommended)



[Print this image](#) (landscape page orientation is recommended)



**Table 1: List of the ports used by CollabWorx servers and collaborative application modules.**

Server	TCP ports	UDP ports
Static Meeting Engine	49401 - 49499	-
Secure Instant Messenger	49501 - 49599	-
BuenaVista retransmitter	49601 - 49699	-
BuenaVista in P2P mode (also for multicast version)		49601 - 49699
CWLite (portlets) and DirectTouch	49701 - 49799	-
Dynamically Scheduled Meeting Engines	49801 - 49899	-

Notes:

- Port assignment follows the [IANA guidelines](#)
- There are no well-known [trojan horses](#) using this port range
- In actual configurations, the servers will usually use just one port number from the specified range
- In multicast mode, BuenaVista uses class D addresses 224.3.10.1+

**Table 2: TI application numbers:**

Application	TCP ports	UDP ports
Screen sharing	49399, 49398	-
NetMeeting based application sharing	1503 and 1720	- (NM uses UDP ports for audio and video conferencing, but CollabWorx product don't use this functionality of NM)

**Table 3: Ports used internally on machines running CollabWorx client software:**

<b>Function</b>	<b>TCP ports</b>	<b>UDP ports</b>
SIM - CW communication	49397	-
CW plugin	49381 - 49396	-

Firewall requirements for client software:

1. Open **outgoing** connections to the ports listed in Table 1, **OR**
2. [SOCKS proxy server](#) and [SOCKSCap](#) installed on each client workstation.

There is no requirement for any incoming connections unless(optional) NetMeeting-based application sharing is used.

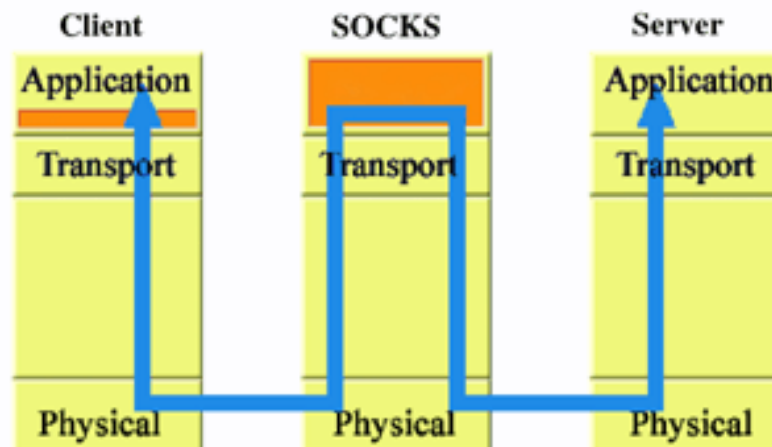
[\[Home\]](#) [\[About Us\]](#) [\[Products\]](#) [\[Downloads\]](#) [\[Search\]](#)

Copyright © 2000 CollabWorx, Inc. All Rights Reserved  
[Privacy Policy](#) | [Contact CollabWorx](#)

**Permeo Products**[e-Border](#)**Technical Resources**[IETF/AFT and other Mailing Lists](#)[Socks Protocol Documents](#)[Developer Documents](#)[Reference Software](#)**FAQs**[Socks General](#)[e-Border FAQ](#)**ABOUT SOCKS****SOCKS Overview**[Why SOCKS? ▶](#)

SOCKSv5 is an IETF (Internet Engineering Task Force) approved standard (RFC 1928) generic, proxy protocol for TCP/IP-based networking applications. The SOCKS protocol provides a flexible framework for developing secure communications by easily integrating other security technologies.

SOCKS includes two components, the SOCKS server and the SOCKS client. The SOCKS server is implemented at the application layer, while the SOCKS client is implemented between the application and transport layers. The basic purpose of the protocol is to enable hosts on one side of a SOCKS server to gain access to hosts on the other side of a SOCKS Server, without requiring direct IP-reachability.

**Place in OSI layer****What is a SOCKS Proxy Server?**

When an application client needs to connect to an application server, the client connects to a SOCKS proxy server. The proxy server connects to the application server on behalf of the client, and relays data between the client and the application server. For the application server, the proxy server is the client.

**SOCKS Model**

There are two versions of the SOCKS protocol, version 4 and version 5, [SOCKSv4](#) and [SOCKSv5](#), respectively.

The SOCKSv4 protocol performs three functions: makes connection requests, sets up proxy circuits, and relays application data. The SOCKSv5 protocol adds authentication. For more information on SOCKS versions 4 and 5, see:

[SOCKSv4](#)

[SOCKSv5](#)

### Control flow of SOCKS

Figure 1 shows the SOCKSv5 control flow model. The portion within the dashed-line represents SOCKSv4 functionality. Note that SOCKSv5 adds authentication.

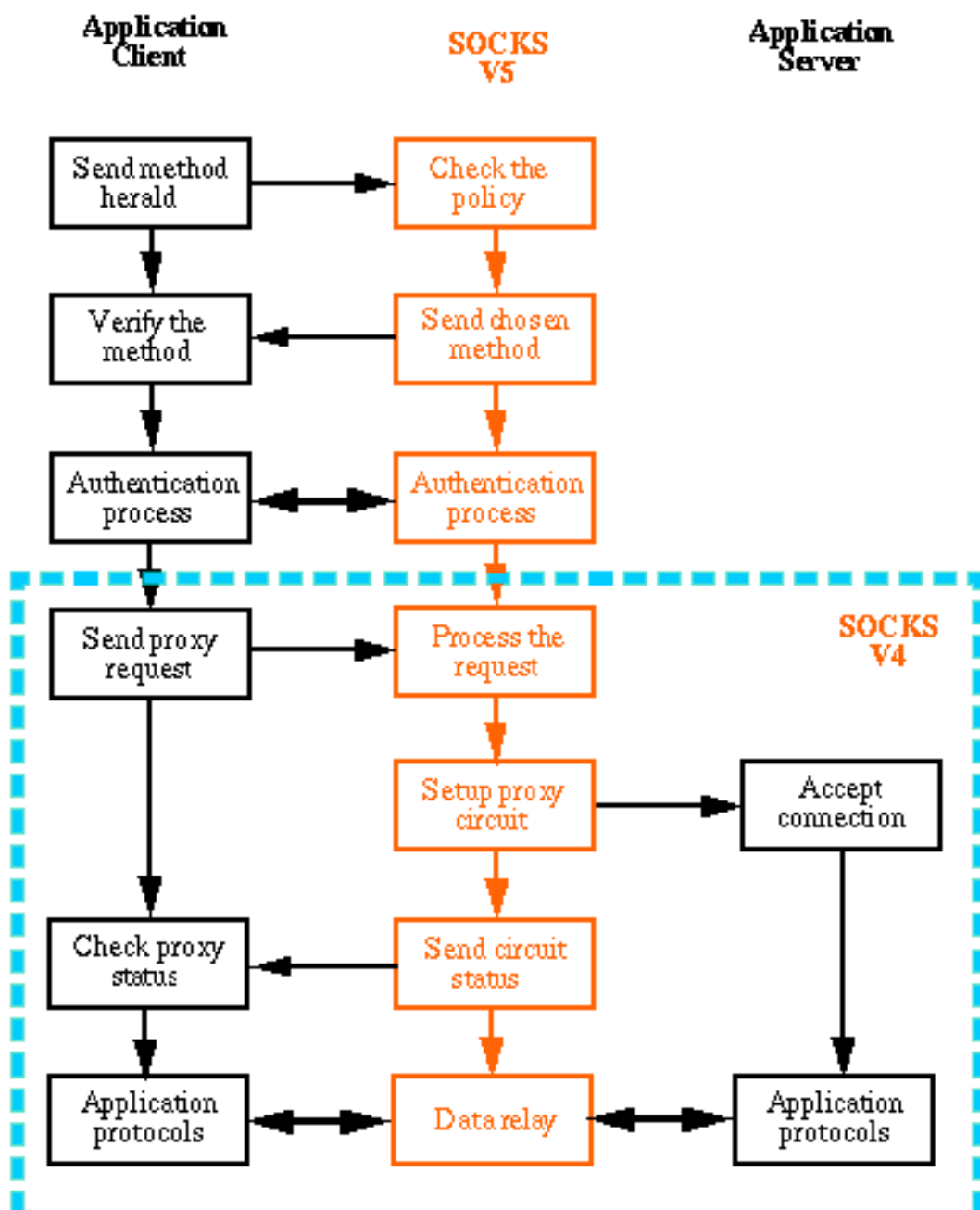




Figure 1

## [Why SOCKS?](#)

---

[SOCKS Home Page](#) | [About SOCKS](#) | [Site Index](#) | [Search](#)  
[e-Border](#) | [e-Border FAQ](#) | [Mailing Lists](#) | [Socks General FAQ](#)

Send questions/comments about Web content to [webmaster@socks5.com](mailto:webmaster@socks5.com)

[Copyright © 1996-2002, Permeo Technologies, Inc.](#) All Rights Reserved.

Permeo Technologies, Inc. - A subsidiary of [NEC USA, Inc.](#)

## Permeo Products

[e-Border](#)

## Technical Resources

[IETF/AFT and  
other Mailing  
Lists](#)

[Socks Protocol  
Documents](#)

[Developer  
Documents](#)

[Reference  
Software](#)

## FAQs

[Socks General](#)  
[e-Border FAQ](#)

All the software packages downloadable from this page are developed for SOCKS Version 5 protocol compliance testing and verification. They are for **non-commercial** (academic, research, and personal) use only. Without any further notice, the software may be changed or removed as the protocol and time evolve.

For commercial and supported SOCKS-compliant products, visit Permeo's [e-Border](#) Web site.

---

## Download Instructions

1. Select a package.
2. Click the Download button.
3. A new page with License terms displays. If you agree to the terms and conditions, select the Download link at the bottom of the page.
4. Each package includes a Copyright file. If you do not agree to the terms of the Copyright file after downloading the package, remove the package from your system.

Please select a package to download:

---

[SOCKS Home Page](#) | [About SOCKS](#) | [Site Index](#) | [Search](#)  
[e-Border](#) | [e-Border FAQ](#) | [Mailing Lists](#) | [Socks General FAQ](#)

Send questions/comments about Web content to [webmaster@socks5.com](mailto:webmaster@socks5.com)

[Copyright © 1996-2002, Permeo Technologies, Inc.](#) All Rights Reserved.

Permeo Technologies, Inc. - A subsidiary of [NEC USA, Inc.](#)



Home About Us Products Support Evaluate Search

## NAT vs. Collaboration

Home : Support : Resources : NAT-related issues



### Collaborative Services vs. Network Address Translation

What do you need to know if you intend to provide collaborative services?

CollabWorx collaborative framework is by far the most tolerant collaboration system when it comes to dealing with firewalls and private networks. Still, there are certain minimal requirements that site networking infrastructure must fulfill to be able to provide collaboration services.

The biggest obstacle in providing collaboration services is use of the so-called "NATted networks". What is a NAT gateway? NAT stands for Network Address Translation. In a nutshell, NAT is a concept that allows re-use of IP addresses.

Before we go into technical discussion of NAT, please, note the following:

1. It is a common misconception that NAT is an Internet standard. **It is not.** NAT is described in IETF RFC 1631. The 1<sup>st</sup> paragraph of this document states:

*"This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind."*

2. NAT has a number of very serious drawbacks. Among others, it severely limits modern security solutions such as VPNs. It breaks many network application, including conferencing tools. We again quote from the original document:

*"Conclusions: NAT may be a good short term solution to the address depletion and scaling problems. This is because*

Resources

FAQ

Glossary

Web Call Center

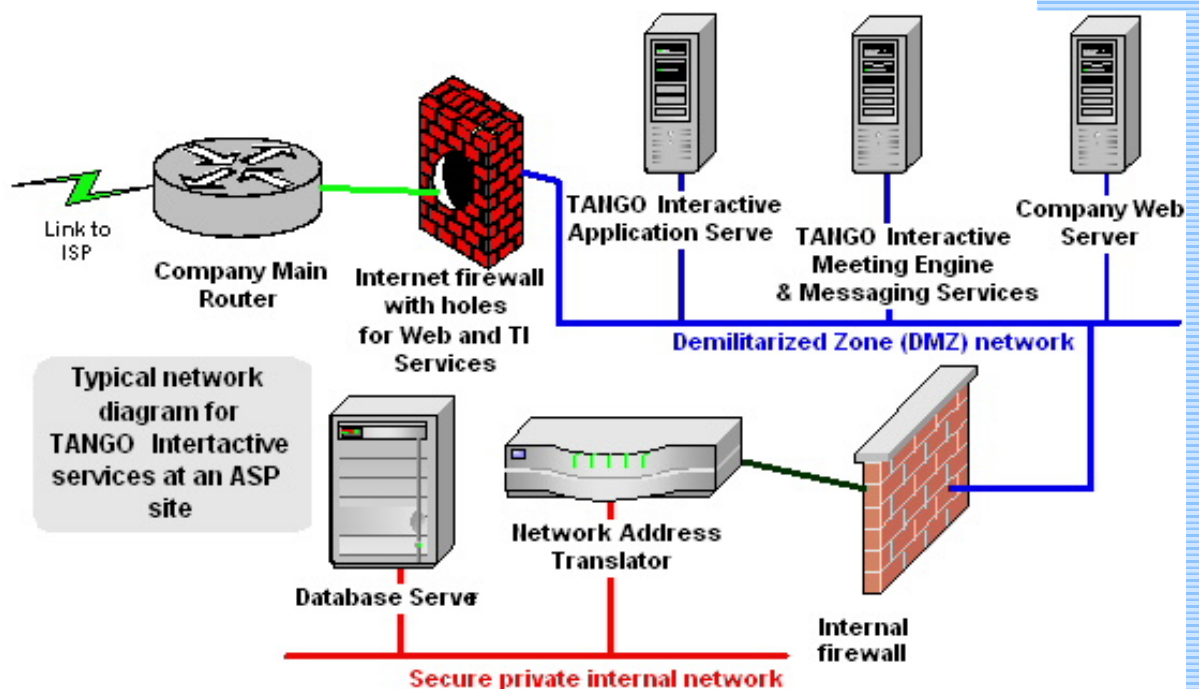
Live Sessions

*it requires very few changes and can be installed incrementally. NAT has several negative characteristics that make it inappropriate as a long-term solution, and may make it inappropriate even as a short-term solution. Only implementation and experimentation will determine its appropriateness.*

*The negative characteristics are:*

- *It increases the probability of misaddressing.*
  - *It breaks certain applications (or at least makes them more difficult to run).*
  - *It hides the identity of hosts. While this has the benefit of privacy, it is generally a negative effect.*
  - *Problems with SNMP, DNS, ... you name it."*
3. As we see from the above, NAT was never intended to become widespread industry-strength solution. Yet, the ISPs have started using it to increase their customer base and revenue, trading quality of service for quantity. As explained below, all ISP services entirely based on NAT are sub-standard. In particular, if your ISP service is entirely NAT based (see below for definition), you won't be able to support any collaborative activities served from your location.

Network setup for a provider of collaborative services



The figure above illustrates the minimal requirements for your network setup. These requirements are not related to bandwidth but rather to the logical architecture of the network. The setup depicted above is almost a “canonical” blueprint for most of the well-designed corporate sites.

The critical elements of the setup are as follows:

1. There is a main router connected to ISP link. The link can be anything – a cable, a DSL, or a T1. Either the router itself has appropriate interface card, or there is a link-specific modem or DSU in front of the router. The router uses public Internet addresses (see below) for all of its interfaces.
2. There is firewall behind the router. This is an optional element, and it is also possible that the main router itself implements firewall functionality.
3. Behind the firewall, there is a semi-secure network segment known as “Demilitarized Zone” (DMZ). This segment also uses public IP addresses. Typically, companies place just a few machines on the network, sometimes just one – the company public Web server.

If you plan to provide services based on TI, DMZ is the location for all TI services such as meeting engine, instant messaging server, and optional audio/video re-transmitter. **None of these services can run on a machine using private IP address!!!** However, it is possible to install these services on the hardware running company web server.

4. Connected to the far end of the DMZ segment is another firewall. This is optional, but recommended. Behind internal firewall there is a NAT gateway. This is optional as well, but shortage of the IP addresses makes it often unavoidable. The 2<sup>nd</sup> interface of NAT gateway uses a private IP address (see below), and so do all the machines connected to the network segment behind NAT gateway.

#### Private vs. public IP addresses

In principle, every machine connected to Internet should have a unique IP address. However, due to the rather wasteful way of assigning IP addresses to organizations (at least in the early stages of Internet), there is acute shortage of addresses. Facing this problem, IETF designated three blocks of addresses that can only be used on private networks. The address blocks are:

10.0.0.0 - 10.255.255.255  
172.16.0.0 - 172.31.255.255  
192.168.0.0 - 192.168.0.0

If your machine uses an address in this range, it uses a private IP address. This means that even if you seem to have access to Internet, your machine is not visible to any of the Internet routers. There may be any number of machines using the same IP address in other companies. You should also understand that even if an Internet router learns about your machine in some way, it would not propagate this information to any other router, as it would do with a public address.

If your private machine seems to have access to Internet browsing, or you can receive e-mail, this is probably because your company has either a proxy server, or a NAT gateway. With either of these solutions, you must understand and remember one important fact: **if one particular Internet service works over NAT, it does not give you any assurances whatsoever that any other service will work as well!** This is because the proxies are usually applications-specific, and the NAT gateways tend to disrupt many network services even if they are carefully configured. Hence, you may be able to browse Internet but you will probably not have ICQ access, etc.

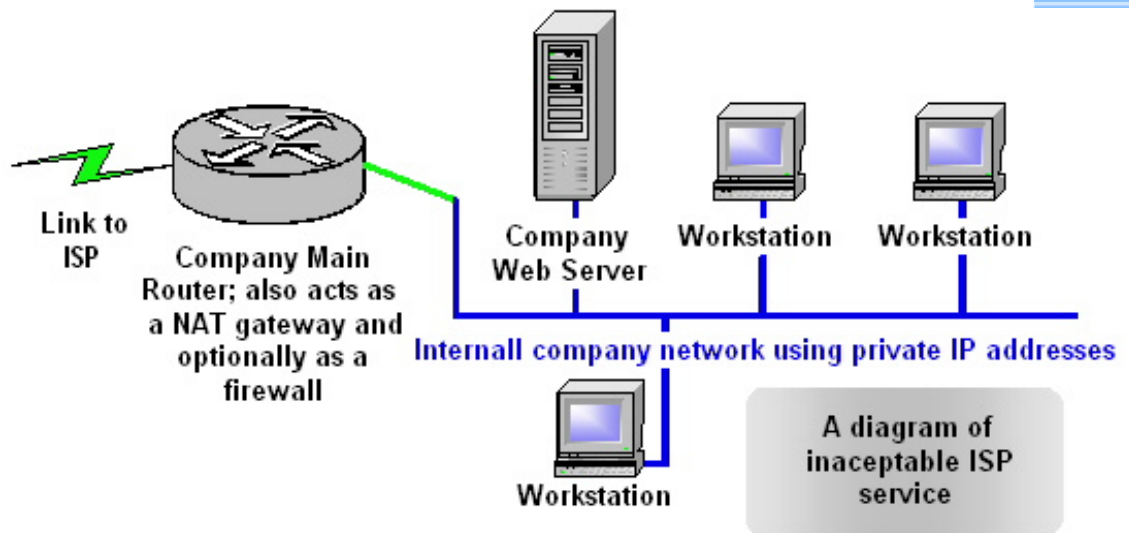
#### NAT gateways vs. firewalls

NAT gateways provide a certain level of security by hiding machines with private IP addresses. This is essentially “security by obscurity”. NAT gateways are not real firewalls and were never designed to be.

From an end-user perspective, there is an additional difference regarding use of networked applications. Some firewalls (i.e. Checkpoint) take a benevolent attitude of “anything that is not explicitly forbidden is allowed”. NAT gateway default attitude is “anything that is not explicitly allowed is forbidden”. In other words, to enable specific applications, a NAT gateway has to explicitly install special port mappings.

As a rule of thumb, you should assume that if you are behind a NAT gateway, you will only have access to a very limited selection of networked applications. Collaboration services are among those, which are disrupted first.

## An example of unacceptable Internet service



The figure above is an example an unacceptable ISP service. In this example, the provider assigns just one public IP address to the customer. This address is used by the Internet-side interface of the company router. The router acts as a NAT gateway and (possibly) as a firewall. All company machines are using private network addresses.

Such a service only supports the most rudimentary applications, such as Web browsing and, with some effort, a Web server. This is achieved by concurrent customized configuration of the router, DNS, and NAT service. Addition of any collaborative services requires elaborate modifications to the router, DNS, and NAT setup.

## NAT impact on CollabWorx clients.

Unlike many others conferencing systems, TI framework allows clients to be located behind a NAT gateway. However, certain limitations created by NAT gateway may cause problems with two applications: audio/video and screen sharing. The nature of the NAT-related limitations is as follows:

- For application modules where multiple application instances used the same port to receive information, only **one such instance** can be located behind the NAT gateway. The reason for this is the NAT gateway algorithm, not TI software.
- For screen sharing, only one participating machine can be located behind NAT gateway. It is not possible to use screen-sharing applications between two NATted machines if each machine is behind different NAT gateway.

### **Networking requirements for service receiver**

Networking requirements for customer receiving collaboration services are less stringent. The only limitation is that if the customer also uses a NAT gateway, only one workstation on the customer site will be able to receive audio and video streams if UDP-based transport is being used. This restriction does not apply to the reflector-based audio/video sessions.

[\[Home\]](#) [\[About Us\]](#) [\[Products\]](#) [\[Downloads\]](#) [\[Search\]](#)

Copyright © 2000 CollabWorx, Inc. All Rights Reserved  
[Privacy Policy](#) | [Contact CollabWorx](#)

[Home](#)[About Us](#)[Products](#)[Support](#)[Evaluate](#)[Search](#)

## IM for Enterprise

[Home](#) : [Support](#) : [Resources](#) : **IM for Enterprise**

[CollabWorx SIM](#)

### Enterprise Grade Instant Messaging



Instant Messaging (IM) is one of the "killer apps" in the consumer market. Slowly but surely IM gains popularity and it might one day challenge e-mail as the most important Internet consumer application.

Picking up habits from their home computers, Internet users started a trend of moving IM from their home machines to corporate desktops. During last year corporate use of Instant Messaging became a significant factor. Interestingly, this trend did not start in corporate IT departments. Rather, IT professionals have been largely caught off-guard. Instead of planning IM infrastructure they have found themselves trying to regulate user access to outside IM services. The natural reaction, especially after September 11, was, in many cases, to disable access to public consumer-grade IM services from corporate desktops.

Is this decision a correct one?

The answer is: yes, it is. Consumer grade instant messaging has no business on corporate desktops. This white paper explains why by examining various negative aspects of consumer grade IM and defining features of Enterprise-grade IM systems.

### External service vs. in-house installation

All popular IM systems run on top of infrastructure provided by large providers. AOL, Yahoo, and MSN dominate the market. This brings us to the 1st critical question: are corporations to outsource IM services to these providers? It is interesting to compare IM with e-mail. How many large corporations outsource their e-mail services to providers that also handle public, insecure mail systems? We all know the answer - they don't!. Why should IM infrastructure be different?

At this time, lack of a proven, secure IM infrastructure software makes corporations scramble for solutions and consider outside service as an alternative. We believe that this is a short-term trend only. All corporate IM infrastructure created in next few years will be installed in-house, as e-mail infrastructure is today. IM will become as business critical as e-mail is today, and IM infrastructure will be handled accordingly.

Hence, if you are looking today for a corporate IM solution, you should look for a IM infrastructure software vendor, not for a public service provider rooted in consumer service culture.

### Global vs. corporation-wide reach

Consumer grade IM systems have a global reach - in their separate domains. Major IM systems don't interoperate. Is this a concern for corporate users? Shall corporations wait with deployment of IM infrastructure until one dominant vendor or service provider emerges?

It seems that the answer is clearly "No". IM wars are not like browser wars - there will be no one winner. A browser solution, operating in a strictly standard environment, could have been hijacked. IM client software is a small part of a much larger infrastructure, which is proprietary. These infrastructures are not going to disappear or be conquered any time soon. The most likely course of action is that the [IETF standard for instant messaging currently under development](#) will provide interoperability between different systems. One day we will have a global IM infrastructure, and all viable IM systems will be able to participate. In short term, corporations should look for a solution that offers a set of features best suited to their needs.

## Dangers of using consumer-grade IMs in business

**Business tool or a fringe benefit?** IM is being introduced in corporate world to improve bottom line by increasing efficiency. Efficiency is not increased if employees can receive at any time messages from their friends and engage in private conversations. Interoperability with consumer grade IMs may actually be detrimental.

**What is all this stuff in my IM?** Consumer-grade IMs are designed according to "kitchen sink" principles: the more features designers can pack in, the better. IMs try to be a universal communication application: multiple chat windows clutter the desktop, audio is being added in rather haphazard fashion, file transfer became a standard feature, unmanaged "buddy lists" allow communication with arbitrary groups of users. Letting such service operate in corporate environment is equivalent to disabling corporate firewall and relinquishing control over information transfer in and out of corporation.

**Built to leak:** Corporations are, and should be, very particular about what information leaves corporate networks. File transfer functionality *does not make any sense* as a part of Instant Messaging. It does not provide any conceivable advantage over e-mail attachments, and e-mail systems are already well equipped to provide audit and filtering functions for such content. Also, large files have no business being transferred using the IM infrastructure consisting of messaging servers tuned for short messages. Introducing this type of traffic makes it much more difficult to provision networks supporting IM. We hence suggest that file transfer should be banned from IM functionality unless it is handled by automated hand-off to corporate e-mail system.

**Anybody eavesdropping?** Unencrypted data and no support for audit are another detrimental factor. Some of the vendors move IM traffic to VPNs. This is entirely incorrect solutions. VPNs are not designed

to support random connectivity. Data security in instant messengers should assume insecure network and be handled on application level, with embedded, maintenance-free PKI provided within IM infrastructure. Message logging should be supported to enable audit.

**"We encrypt messages hence we are secure"** - this is what AOL and Yahoo wants you to believe. What good is message encryption if you don't really know with whom you are exchanging messages? To be secure, a product MUST support AT LEAST user authentication, user authorization, and data integrity. Any of these elements is missing, and the product does not qualify as secure. Encryption alone does not solve anything. You may want to consider the fact that of all the security breaches we have seen since Internet went commercial not even one involved actually breaking data encryption!

**Am I your buddy, Susan?** "Buddy lists" seem to be a prevalent mechanism used to manage user groups. While working fine for private use, buddy list is a useless concept in corporations.

Doing business is not about "buddies" - dynamic group structure is driven by tasks, projects, and relationships that must be centrally managed. Failing to do so results in broken communicating patterns - "You are on my buddy list but I'm not on yours - don't you like me?" With no concept of centrally managed communities, consumer grade IMs don't have tools to build and support workgroups.

**Don't talk to me, boss!** Since consumer-grade IMs are already being attacked by spammers and/or by viruses (courtesy [integrated file transfer](#)), designers were quick to provide filtering capability. In corporate setting, filtering should only be accessible for IT personnel. After all, users want to be certain that their messages are received!

**Can I manage my users, please?** Having no way to build communities, consumer grade IMs of course

lack tools for user authentication, authorization, and for setting access rights to workgroups. Implementation of secure group-wide communication is very problematic in such situation.

**Instant messaging vs. collaboration** Kitchen-sink design of consumer grade IMs demonstrates ignorance of the consumer-grade IM designers in the matter of structured communication and collaboration process. By packing "collaboration" features such as audio, chat, and whiteboards, IM vendors try to get into collaboration tools market. Yet, these ad-hoc solutions are inferior and, frankly, quite naive. *IMs cannot and should not replace high-quality enterprise collaboration tools.*

An IM client should provide three basic functionalities: *community access, short messages support and presence manager*. These features are all necessary but also sufficient to provide users with the *awareness and feel of connectivity*. Once these are established, the IM should be able to jump-start collaborative sessions using arbitrary collaboration software by providing a "*gateway menu*". This gives IT managers freedom to pick up best IM and, independently, best collaboration software. IM should support a "single sign-on" capability, i.e., if a user has already identified him/herself to instant messenger, her/his credentials should be securely transferred to the collaboration toolset.

**How many chat windows I need to stop me from doing my work?** The issue of HOW instant messages are delivered is very often ignored. Yet, it is critical if a corporation hopes to reap measurable benefits from deploying an IM system.

The consumer grade IMs adopt a multi-chat paradigm - if a user communicates with several other users, multiple mini-chat windows pop up on the desktop. In our experience, this is a terrible design. Managing these windows is a chore and a distraction. A much more efficient design is to deliver a message in a pop-up window, let user

respond, and hide the messenger interface - a la SMS. This approach lets users do whatever they are doing while being able to interact with minimal distraction.

If a situation calls for an extended one-to-one or group meeting, the above mentioned [gateway to a collaboration](#) systems brings up more advanced tools for an *instant* collaboration session. Conversely, notifications about scheduled virtual meetings can be delivered via IM. This is an important part of what we call *Secure Unified Collaboration* approach.

**CollabWorx SIM - Instant Messaging for Enterprise:**

We offer an IM solution that has been designed for corporate use from ground up. Please refer to the [detailed product description](#) for more information.

[\[Home\]](#) [\[About Us\]](#) [\[Products\]](#) [\[Downloads\]](#) [\[Search\]](#)

Copyright © 2000 CollabWorx, Inc. All Rights Reserved  
[Privacy Policy](#) | [Contact CollabWorx](#)


[Home](#)
[About Us](#)
[Products](#)
[Support](#)
[Evaluate](#)
[Search](#)

## Collaboration Security

[Home](#) : [Support](#) : [Resources](#) : **Security in Collaboration Systems**

### Security Issues in Web-based Collaborative Systems

*Lukasz Beca, Marek Podgorny  
CollabWorx and Syracuse University*

**Abstract:** Web-based collaborative services are becoming increasingly popular. Examples include chat rooms, virtual meetings, and distance learning environments. In this paper we analyze functionality and architecture of two types of real-time collaborative systems: a system for supporting online meetings and a system for supporting distance learning sessions. We discuss security issues that arise during design, deployment, and use of these systems, such as authentication, authorization, communication security, auditing, and integration with existing security frameworks. We identify currently available security solutions and evaluate their applicability in addressing identified issues. We also propose new solutions where currently used technologies do not seem adequate.

*Index Terms:* security, World Wide Web, collaborative systems

#### I. INTRODUCTION

The diversity of the services available on the World Wide Web is growing. The users, apart from accessing the most popular applications such as browsing documents and performing purchases, can also participate in online auctions, chat rooms, virtual meetings, distance learning sessions and other activities. The World Wide Web is becoming increasingly a virtual place where people not only operate in isolation but also interact with other Web users. Still, even though the existing infrastructure supports relatively secure Web browsing and purchasing transactions, there is no generic framework for building secure systems for on-line real-time collaboration.

Some of the already available solutions can be adopted (for example SSL [18],

[Technical Info](#)
[FAQ](#)
[Glossary](#)
[Web Call Center](#)
[Live Sessions](#)

digital certificates), but not all of them are sufficient or appropriate because of requirements imposed on real-time collaborative systems, which are different from the requirements imposed on standard Web applications. For example, the HyperText Transfer Protocol (HTTP) [7] does not provide ability to notify a Web client about changes in the content managed by Web servers. A notification mechanism is one of the several services that might be used to implement collaborative applications, which must receive updates about changes in the shared application state. In absence of the notification mechanism, new communication protocols must be used by such applications. This approach in turn causes a number of problems related to integration of the system communication infrastructure with already existing security solutions—for example deployed firewalls.

In this paper we address general issues of security in Web-based collaborative systems. The paper is organized as follows: In Section II we discuss how various aspects of the Web environment can be of benefit to collaborative systems. In Section III we discuss two typical collaborative applications and, based on a case study, we define generic system architecture for collaborative systems. In Sections IV to VIII we discuss various security aspects and available solutions. Section IX provides an overview of the integration issues related to the existing Internet security infrastructure. Conclusions are provided in Section X .

## **II. Web-based Real-Time Collaborative Systems**

Development of the World Wide Web is a phenomenon with many different facets. Socially, WWW transformed perception of computers from an obscure and arcane technology used by technical and business people to a home appliance and a subject of casual party conversations. In this aspect, computers and information access capabilities became a part of everyday social life. Technologically, the broad demand for these new, hitherto exotic services, led to an unprecedented explosion in creativity of system architects. In a span of just several years, the Internet evolved from a simple collection of few dozens of web servers into an advanced distributed computing system.

Computer Supported Collaborative Work (CSCW) premises and concepts predate the Web but it is one of the fields that is being profoundly affected by the Web: CSCW is traditionally concerned with human-computer interfaces, and its focus is computer support of human collaborative activities. With the social change of attitude towards every-day use of computers, CSCW is poised to leave its laboratory confines and move towards mainstream computing.

The Web environment offers significant benefits to the developers of collaborative systems. The most important features are: global scope, high availability, rich content, multi-platform clients, and ability to host executable content. All these properties make the Web a powerful platform for building collaborative applications. In fact, many of the new industrial groupware systems (for example WebEx [19], Lotus SameTime [14], PlaceWare [16]) are Web based.

This trend is accelerated by the nature of Web technology. Its social and technological impact on groupware makes Web –based collaborative systems increasingly sophisticated, and, slowly but surely, improves their acceptance. As the Web-based collaborative systems gain industrial acceptance, they will have to undergo the evolution from laboratory tools to strategic IT tools. This, among other things, implies making them secure and integrating them with the

existing Internet security infrastructure.

### **III. Case Study**

Initially, most of the groupware applications developed on the Web enabled only asynchronous collaboration (for example, BSCW [9]). These applications were built using Web technologies such as HTML pages, CGI scripts, or Java servlets. As a result, security problems encountered in such applications could be addressed using standard Web solutions (for example form based authentication and HTTPS protocol). Some of these technologies are also useful for development of real-time (synchronous) collaborative systems. However, inherently different requirements imposed on real-time collaborative systems force developers to combine Web technologies with non-Web solutions. Such hybrid systems are not entirely covered by the security models currently used on the Web.

In this section we describe our efforts to create two secure Web-based real-time collaborative services: a web conferencing system (called virtual meeting) and a distance learning system (called virtual classroom), based on TANGO Interactive framework [3] developed jointly by Syracuse University and by CollabWorx. We wanted to build the systems in such a way that:

§ After deployment at a Web site they do not decrease security of already present solutions.

§ The potential users of the environment can trust other participants of the interaction; the content of the interaction is protected from capture or malicious modifications.

§ Deployed security solutions do not impose computation or communication overhead that would be unacceptable to the service users or service hosts.

#### **A. Functionality**

Virtual meeting system provides functionality of scheduling and holding online meetings. The meetings can be scheduled and accessed only by authorized users. Usually one of the participants prepares presentation to be displayed for other users during the meeting through shared web browser. Communication with other participants is provided through audio/video links, and chat application. Whiteboard tool can be used to clarify presentation issues. Moreover, participants have an option to share their screens or selected applications. All users have access to the list of meeting participants.

Virtual classroom system enables delivering lectures to the geographically students in real-time. It supports structured patterns of interactions appropriate for teaching. The instructor has active role; he or she coordinates activities in the classroom to ensure efficient learning process. The students receive lecture content, they can ask questions and they can take control over certain tools while monitored by the instructor. The tools for signaling questions or continuous evaluation of the lecture presentation enable better interaction between students and the instructor. Our experience (the system was used to deliver fully accredited, semester-long graduate courses from Syracuse University to geographically remote colleges) indicates that in some cases, when the number of students is large (i.e., above 15), it is useful to introduce another type of a session participant – an assistant. Such person is responsible

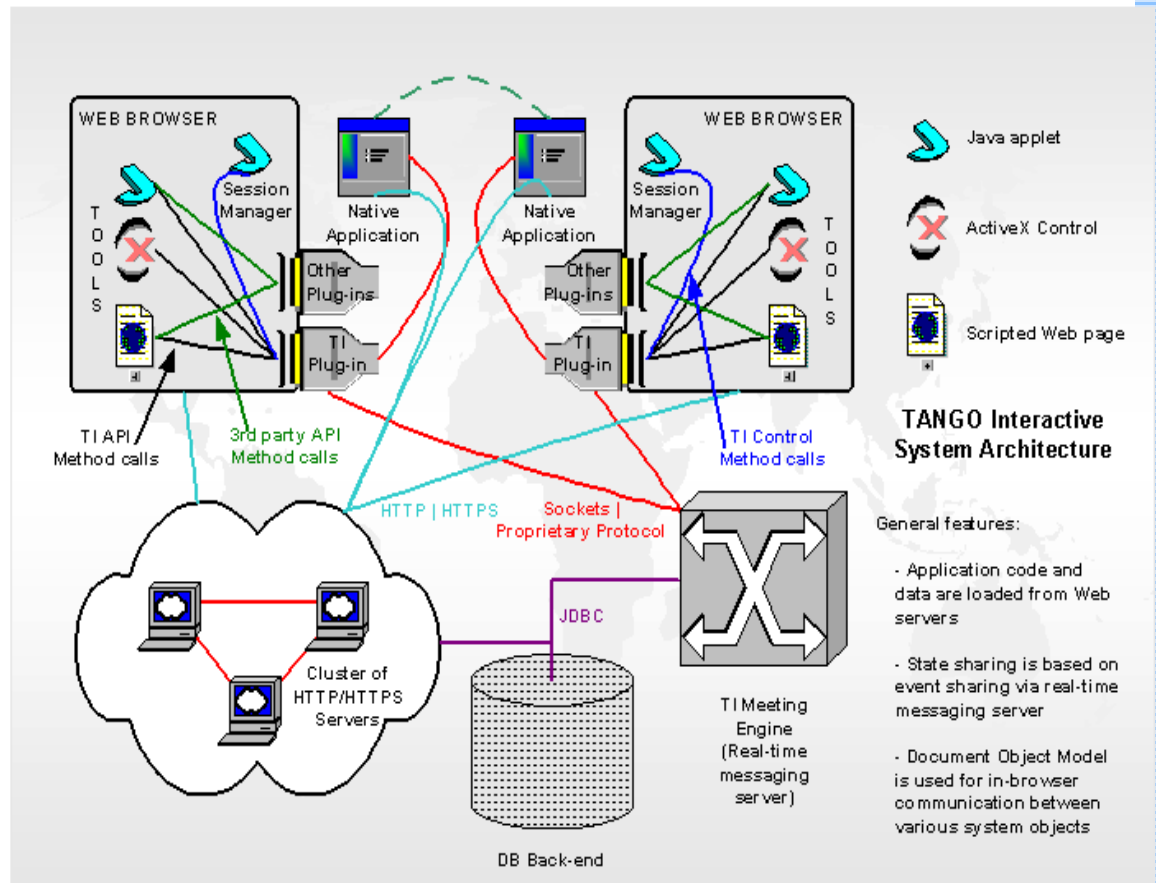
for monitoring activities of the students and responding to the questions related to technical problems that may develop during a virtual lecture. Many problems during the class sessions stem from the fluctuations in the available bandwidth of the Internet connections and this factor influences quality of the service delivered to the students.

## **B. Architecture vs. security**

The applications described in the previous section have similar architectures since they are built using the same collaborative framework—TANGO Interactive (hereafter referred to as TI). Figure 1. presents the architecture of the TI system.

TI system is built according to several basic principles. The most important are:

- The entire system is browser-based. It is started from a Web link, and most of the client side code executes in a browser.
- Collaborative tools are implemented as Java applets, scripted Web pages, or as ActiveX components. However, the framework takes a pragmatic approach and allows native collaborative tools. Native tools can be implemented in C/C++, as Java applications, or in other programming language. Functionally identical APIs are provided for each supported language.
- The system allows tools to be downloaded at runtime or installed on local machine.
- If the tools are downloaded by the system, they can originate from an arbitrary Web server. The tools can, in turn, request additional data. TI framework treats Web servers as a primary repository for both the tools code and data. This holds even for the native applications.
- Collaborative tools share the state using event-sharing mechanism. The events are shared via a special middleware layer, denoted as TI plug-in Figure 1., and via a real-time messaging server, called TI Meeting Engine (TIME). TIME does not have to be collocated with any of the Web code and data servers. It is implemented as a Java application.
- Collaborative tools, either in-browser or native, can communicate with TIME only via the middleware (plug-in). Importance of this arrangement is discussed below.
- The bi-directional communication for sending and retrieving messages between the collaborative tools and the middleware is provided. Browser supported document object model is extensively used.



*Figure 1: Architecture of the TANGO Interactive web-based collaboration framework*

Two critical components of the system are the plug-in [1] (middleware), and the Meeting Engine. Middleware has been introduced to ensure high system reliability. It also plays important role in system security. The crucial idea behind middleware is that collaborative tools cannot communicate directly with the messaging server. Instead, they use middleware method calls for such communication. The middleware acts as a message filter and multiplexer, preventing applications from sending ill-formed or illegal messages. An ill-behaved tool might not work but it would not disturb the overall system operation.

Use of the plug-in as data multiplexer simplifies implementation of the messaging server. Each system client connects to the engine just once. A proprietary protocol provides all services necessary to identify message sources and addresses. In absence of the multiplexing plug-in, each application would have to open a separate connection to the messaging server. Such a solution practically precludes implementation of strong and coherent security mechanisms. As described in further sections, in TI framework secure communication is implemented between the plug-in and the messaging server.

Since the messaging server is a central element of the system, it may become a target of attacks executed by hackers that want to take control over the environment. The server has been designed with security in mind. It is primarily responsible for message routing. In contrast to data servers such as HTTP or FTP, TIME does not perform any local file operations. For its authentication and authorization functionality, TIME communicates through encrypted connection with the database placed behind the firewall. In addition, digital certificates may be used for authentication during database access to eliminate possibility of TIME spoofing. The database is accessed by TIME in read-only mode. The

reason for this approach to TIME security design is that the messaging server cannot be placed in a highly secure environment without seriously impairing its basic function of message routing (see Section IX.B ). In the case that TIME is broken into, the server does not provide any means to further compromise OS or data source security.

As shown in Figure 1., TI framework has one special collaboration tool, the Session Manager. The tool is implemented as a Java applet, and, depending on system configuration, it may present system user with a graphical interface, or it may be hidden, in which case its operation is defined by startup scripts. Session Manager (SM) uses special protocol to communicate with the plug-in. This protocol supports complex and potentially unsafe operations such as loading the application code and data and starting application execution on remote machines. SM tool participates in user authentication and authorization, and it also provides unified mechanism for meeting floor control. Under these circumstances, code and communication protection of SM is critical. TI solves this problem by digitally signing both the SM and middleware code and by verification of digital signatures before communication can be established.

The final comment about the architecture is related to the peer-to-peer communication channel between the instances of a collaborative tool. In Figure 1, a dashed line connects the two instances of "Native Application". This schematically denotes a peer-to-peer communication model. Since we have stated above that TI framework uses messaging server to establish message routing between collaborative tools, the addition of peer-to-peer communication pattern may be confusing. Let us hence explain this issue in some detail.

TI framework achieves scalability and high performance by making a careful distinction between the *data* and *event* communication channels. Let us consider a shared browser tool. In the popular shared display (a.k.a. shared screen) mode of collaboration, a master browser would download the data from a Web server, and then the system such as NetMeeting would replicate display of the Internet Explorer on the workstations of all shared browser session participants. This clearly is not an optimal solution. TI framework works differently: under the control of Session Manager, a shared browser tool is started on multiple workstations. The session chair, or master, selects a URL to load into the browser. The browser sends the URL to TIME, which, in turn, distributes it to other instances. The "slave" instances download the URL from the same Web server or, if the system URL mapping facility is used, from a replicated server. With properly configured topology of the system it is possible to have sharing of HTML documents while incurring only minimal network traffic. The same principle can be now used to capture browser events such as page scrolling, DHTML layer control etc. Shared browser perfectly demonstrates all advantages of event sharing.

Clearly, the same principle can be used to implement a shared whiteboard, or a shared world described by a VRML scene graph. Event-sharing methodology of state sharing works best if the amount of information necessary for state synchronization is small compared to the amount of "static" data. In our solution, dynamic data is distributed via messaging server; static data always comes from Web infrastructure. Similar approach is applied in case of a chat application. We can distribute chat messages via TIME, since chat messages are short and, being generated by humans, infrequent.

The paradigm breaks with applications such as audio-video conferencing. The data streams become "fat" and they are also time-sensitive. In such case, TI

framework uses TIME only to establish collaborative sessions of the specialized tools, such as an audio or video conferencing, and it allows the tools to use their private communication channels. This is a mechanism very similar to the H.323 [11] approach, where Q.931 and T.245 protocols are used to establish the “calls” and then RTP protocol is used to distribute data streams in the peer-to-peer mode, optionally using IP multicast to optimize network bandwidth. TI framework provides security only for the session setup phase. Security of the media stream distribution is not supported. The media conferencing tools have to implement this functionality on their own. TI framework makes this process a little simpler since it can be used to, for instance, exchange the secret encryption keys and security policy parameters between instances of the tool.

#### **IV. Identified Security Issues**

Collaborative environments have a number of the requirements related to the security of these systems. In general, they can be broadly divided in two parts. One set of problems is related to traditional security measures such as authentication, authorization, and communication security. Each of these issues has new aspects in the context of collaboration systems. We will discuss these new aspects in detail. The second group of problems is related to integration of collaborative systems with the existing Internet security infrastructures.

Authentication of the users is necessary so that meeting participants can determine who they are interacting with and possibly have access to the information about other group member. Such approach increases trust among meeting participants and leads to more efficient interactions. Authorization mechanisms are necessary to determine access rights to the collaborative application resources. They can be used to limit the group of users that can participate in the online meetings or virtual classroom sessions. They can also be used to impose structure on the interaction by applying restrictions in access to the system resources. Another important issue is security of communication channels used for exchange of control and application data among collaborative clients and servers. Without protection the protocol messages can be intercepted and used to gain access to the system. Moreover, potential attacker can get access to the application data exchanged between collaborative client and server. Finally, there are numerous security solutions already deployed on the Web (for example, security models implemented by Web browsers, firewalls on corporate networks) and collaborative environments that operate in this environment should be able to integrate their security solutions with already present infrastructure.

#### **V. Authentication**

Activities performed during regular real-life meetings (for example, presentations, discussions, decision making) require that participants know and trust each other. The same necessity applies to the online meetings. One of the important features of the collaborative system that supports this requirement is assurance that the users that participate in online interactions are really the individuals who they claim to be. In this way, the participants of the meeting know that the specific identifier denotes certain real person that they probably met in real life or exchanged a number of e-mail messages and can expect specific type of behavior. The issue of authentication is also essential in other types of distributed applications, for example, distributed operating systems [4].

A number of technologies are available that support authentication. Kerberos

[13] was designed to provide authentication services for large distributed systems. Currently, it is included in several operating systems. However, it is not widely used in the Web environment. SSL protocol provides mechanisms for authentication, which allow Web clients to authenticate Web servers and vice versa. The authentication is based on the X.509 certificates [1] installed in the Web browsers and servers. Simpler solutions use form-based authentication supported by Web browsers. It is also possible to embed authentication solutions in the protocol used for communication between collaboration server and client.

In our solution we use combination of two mechanisms: X.509 certificate authentication mechanism provided by the HTTPS protocol and authentication messages embedded in the protocol used for communication between collaboration client and server (that correspond to plug-in and TIME as described in section III.B ). The first mechanism is used to gain the access to the Web site where the users can enter meetings. Another mechanism is used when the user enters specific virtual meeting or virtual classroom session. At that point the participant must submit user name and password. The information about users with corresponding passwords is stored in the database and it is accessible to the collaboration server (TIME), which can check validity of user login operation. We use SHA [13] algorithm for scrambling passwords. Since the password is sent without encryption from the client to the server, the communication channel must be protected (see section VII for more details). The described authentication mechanism provides base for enforcing authorization polices defined for specific types of environments (for example virtual classroom).

## **VI. Authorization**

Authentication enables verification of the user's identity and this information can be used to define user's access to various resources and services provided by the system. In the first approach we can define all-or-nothing policy. The user that was recognized by the system as authorized is allowed to participate in collaborative sessions and use the system without any restrictions. This approach may be appropriate for on-line virtual meetings where participants are subject to social norms that in natural way impose limits on the interactions. However, in certain situation more structured patterns of interactions are required; ability of defining such structure may be achieved by introducing refined access control mechanisms. A type of session management policy [5], access to specific collaborative tools, and modes of floor control [8] – all these parameters can be used to shape collaborative policy targeted for particular application.

For example, in the virtual classroom environment, an instructor needs to maintain control over collaborative activities at all times in order to deliver the content of the lecture efficiently. On the other hand, the students should have access only to limited functionality that would not allow them to disturb other receivers of the lecture. In our solution we addressed this problem by the introduction of the simple role based access control (RBAC) mechanism (for extensive description of this topic see [6] and [12]). Each participant in virtual classroom session has a role assigned. Example roles include a professor, a student, and an assistant. The roles in turn are used to describe access permissions to collaborative sessions, tools, or control tokens. Depending on the role, the system may decide to allow the participant to enter the meeting, start specific tool, or use already started tool with limited or full functionality. As a result, it is possible to control collaborative functionality that is accessible to specific user precisely and relatively easily.

## VII. Communication Security

One of the assumptions that must be made when building distributed application operating on the Internet is that all messages exchanged among application components over the network can be intercepted, fabricated, spoofed, or stored and retransmitted later by malicious users. The commercial Web sites that perform financial transactions use HTTPS (HTTP over SSL) to provide secure access to their services. Real-time collaborative systems are also vulnerable to this type of threat. The messages exchanged between collaboration server and clients (TIME and plug-ins as described in section III.B ) contain protocol messages (for example login message with user name and password) and application data that can be potentially used to harm collaborative session participants (for example, if the confidential business information is presented during a virtual meeting). Also, a malicious user can try to send messages that simulate actions of one of the meeting participants. Therefore two properties must be satisfied in order to provide secure communication among collaborative application components: confidentiality and integrity.

In order to provide these features in implementation of TI framework we used SSL as an underlying layer of our communication protocol (we evaluated BSAFE—SSL Java libraries developed by RSA [17]). SSL has a number of properties that make it feasible as a mechanism providing communication security for collaborative environments. It provides mechanisms for client and server authentication, channel encryption, and message authentication. Moreover, it allows selection of various cipher suites from the strongest (but slowest) to the weakest (but fastest) so that requirements of various types of user groups can be accommodated. The users that can accept lower performance for the benefit of improved security can use the strongest cipher suite. On the other hand when the security requirements are more relaxed, faster algorithms can be employed.

Obviously, additional protocol layer adds computational overhead to the message processing. This overhead is almost unnoticeable for applications with low bandwidth requirements such as chat or whiteboard that exchange data in form of small state updates. However, the overhead can be significant for applications that require high bandwidth, for example screen sharing applications. Also, applications that transmit large volumes of delay-sensitive data (audio/video conferencing tools) using connectionless protocols (UDP) cannot use SSL to provide communication security. In such situation, phone conference call can be used to provide secure audio link.

In our implementation of virtual meeting and virtual classroom systems, SSL (in combination with X.509 certificates) is also used to authenticate collaboration server (TIME) to the clients. Such approach enables control over deployed engines and makes difficult establishing unauthorized installations. However it requires creating infrastructure for issuing and management of keys and certificates used by the system (PKI framework can be used for that purpose [1]).

## VIII. Auditing

If the intrusion into the system already happened the mechanisms should be in place to allow discovery of the break-in. Auditing (recording of important events related to accessing the system functionality) can be used to enable such

detection. In our solution the collaboration server (TIME) maintains a log of the most important events in the system. Each entry in the log contains following data:

- Operation type: system login and logout, session join and leave, session creation and termination.
- Target: objects affected by the operation.
- User identity: the system assigns temporary identifiers to the users upon successful completion of login procedure, the identifier is later used to tag actions executed by the particular user in the log
- Status: provides information about whether operation was executed successfully or not. In case of failure, the reason is provided
- Timestamp of the operation

Analysis of the information contained in the log can give insight into activities of the users and can help in detection of the potential intrusion.

## **IX. Integration with Existing Security Frameworks**

Designers of the Web-based applications must take into account security technologies already deployed in the Internet environment. In this section we concentrate on the security model implemented by Web browsers for executable objects and on methods of implementing communication across firewalls.

### **A. Web browsers and executable objects**

The systems that use objects executable in Web browser environment (for example Java Applets or ActiveX controls) must take into account security models implemented by the Web browsers (Netscape Navigator and Internet Explorer). Java applet by default is not allowed to access any resources placed on local machine. Also, it is forbidden to open socket connections to the machine other than the one where the Web server that hosts the applet is placed. The latter limitation is particularly significant if we want to deploy a collaboration server on the machine other than the one from which the client was downloaded. Fortunately, it is possible to overcome mentioned restrictions by using appropriate mechanisms [10] provided by browser platforms. These mechanisms are based on capabilities model in which *principals* (digitally signed Java applets) can get access to certain *targets* (protected system resources). Authorization of access is represented by the *privileges* associated with the principal and related to particular target. As a result, in order to access a particular resource managed by the Web browser, the principal must explicitly ask for the privilege to use the resource (the code of the applet must contain an appropriate method call). After the privilege is granted, the resource can be used without restrictions.

In practice, before letting an applet to execute a potentially dangerous operation (for example, opening socket connection to arbitrary host on the Internet) the browser displays a window with information about the feature that is supposed to be accessed, information about object signer, and controls that enable user to make decision to grant or deny appropriate privilege.

Our system requires access to several restricted resources such as: socket connections to arbitrary hosts (necessary to connect to TIME) or ability to execute applications on client's machine. The access to this functionality is

acquired through the capabilities mechanism described above.

## **B. Firewalls**

For collaboration systems, whether Web-based or not, firewalls create a set of major problems. These problems are not widely known, and they have not been extensively discussed in the literature. This is probably due to the fact that academic campuses, where most of the CSCW research work is done, usually do not use firewalls, relying instead on other network protection mechanisms, such as tight traffic monitoring and policy enforcement via administrative methods. Academic institutions also enjoy a privileged access to the IP addressing space, eliminating the need for network address translation. Yet, the problem of traversing firewalls is critical for successful deployment of collaboration systems. Our experience with such deployment on corporate and DoD networks provided us with a body of evidence on how existing security infrastructure can break deployment plans for collaborative applications.

The most frequent reasons for the problems we have encountered in practice are:

- Network administrators responsible for security do not want to compromise firewalls by “punching holes” for additional network connection necessary for collaboration systems to function. For example, the popular NetMeeting application from Microsoft [15] requires opening of a number of TCP ports and all UDP ports with numbers above 1024! Deployment of applications that use UDP protocol is particularly difficult to negotiate, as many secure corporate networks block all UDP traffic. Similarly, H.323-compliant conferencing products require a block of both TCP and UDP ports to traverse the firewall.
- An alternative to opening ports in firewalls is installation of proxy servers integrated with firewall. The problem with this solution is that, with exception of SOCKS [20] proxy servers, all proxy servers are application protocol-specific. This means that every application requires a separate proxy to communicate through firewall. Installation of proxy servers incurs additional software license and maintenance cost. Further, proxy servers are only available for popular protocols such as HTTP and FTP. Certain application vendors provide proprietary proxy servers (a good example is Oracle Connection Manager, which provides proxying of the entire Oracle Net8 protocol suite), but for many other application a network-layer SOCKS proxy is the only available solution, which slowly is becoming a standard. Still, configuration of a SOCKS proxy server for real-time groupware is relatively complex.

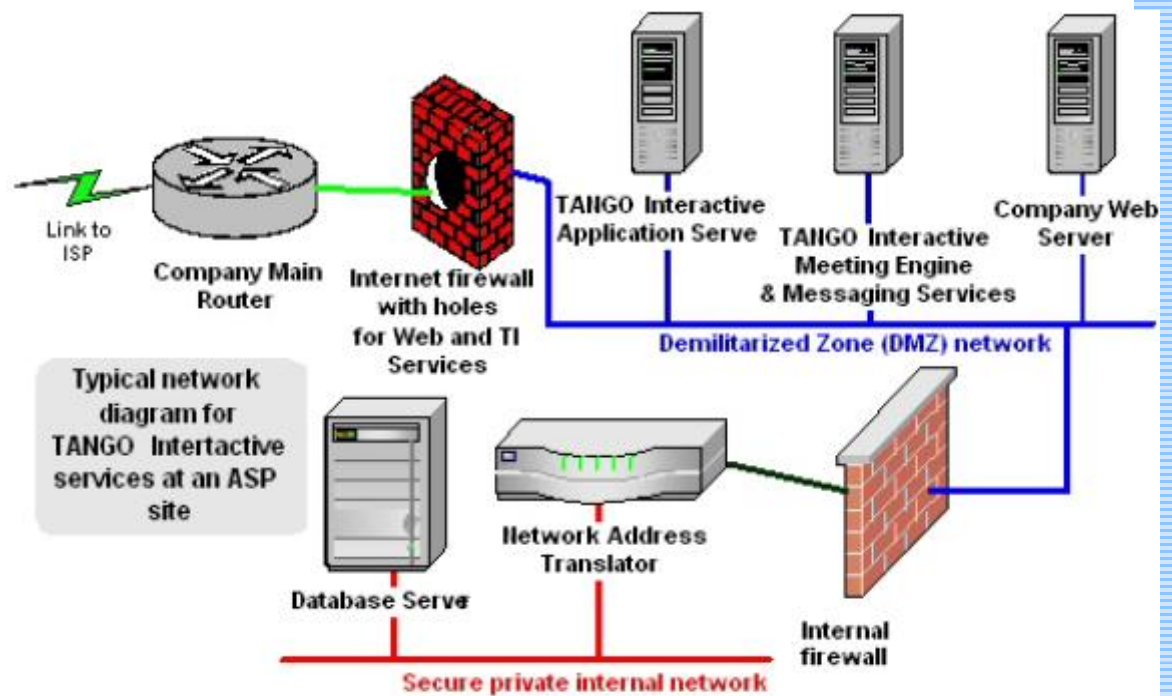


Figure 2: Secure deployment of a web-based collaboration service

Security integration issues for groupware systems for the service receivers are generally less complex than these for the service providers. In our solution, if the end-users of a collaborative service are connected to a network that uses generic firewalls without proxy servers the only available solution is to open ports in the firewall to enable access from the client-side middleware to the collaboration server (TIME). In this context the importance of the message multiplexing function of the middleware becomes apparent. It is sufficient to install a firewall channel for one TCP port to enable communication with the conference engine. For the firewalls with integrated proxy servers, a standard HTTP proxy server takes care about all HTTP traffic described in Section III.B . To provide communication with the TIME a SOCKS proxy should be used. To avoid complex setup procedures on the server side it is recommended that all end-user workstations install a SOCKS wrapper (a good example of such component is SOCKSCap [2]). The wrapper enhances TCP/IP protocol stack on the end user workstation and routes all outgoing connections to the SOCKS proxy server. This solution has been tested on large secure corporate networks with TANGO Interactive framework. A combination of the standard HTTP and SOCKS proxy servers enables the entire functionality of the collaborative application.

On the provider side the requirements are a little more complex. Figure 2 presents a recommended configuration of the site providing collaborative services.

A typical security setup involves two firewalls: an Internet firewall and an internal firewall. The network segment behind the Internet firewall is often called "Demilitarized Zone" (DMZ). Typically, the public Web servers are placed on this network segment, which is not considered very secure. Many

installations refresh contents of the Web servers in regular intervals to prevent unauthorized changes by hackers.

The DMZ segment is the ideal location for both the collaboration server (TIME) and for the Web server acting as the code and data repository for the collaboration system. As discussed in previous sections, our collaboration server was designed to provide secure service. The code and data repository can be periodically refreshed in the same way as the other Web server content. The database back-end storing collaboration application user and policy data can be placed behind internal firewall.

The solution presented in Figure 2. suggests opening TCP ports in the firewall to enable access to both Web and collaboration servers. A more sophisticated and more secure solution can be implemented using an inverse HTTP proxy server for Web services and a custom-built inverse proxy server for the collaboration engine.

## X. Conclusions

Design, implementation and deployment of secure collaborative services on the Internet are complex tasks. We analyzed security issues that we encountered during our work on real-time virtual meetings and virtual classrooms. Some security aspects, such as authentication and channel security, can be addressed with relative ease using already available technologies (although not all aspects can be solved in this way; for example, there is no standard solution for encryption of audio and video streams). Other functions, such as authorization and auditing, must be designed and implemented specifically for the developed systems. Finally, the integration with already deployed security solutions on the Internet, especially firewalls, requires careful analysis of the installed infrastructure and often requires changes to the site configuration.

## References

[1] C. Adams, S. Kent, S. Lloyd, *Understanding the Public-Key Infrastructure*, New Riders Publishing, 1999.

[2] Beagle Software, SOCKSCap information:  
<http://www.beaglesoft.com/ProxySocksCap.htm>

[3] L. Beca, G. Cheng, G. C. Fox, T. Jurga, K. Olszewski, M. Podgorny, P. Sokolowski, and K. Walczak, "Java enabling collaborative education, health care, and computing", *Concurrency: Practice and Experience*, Vol. 9(6), June 1997, pp. 521-533.

[4] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, 3rd ed, Addison-Wesley Pub Co, August 2000

[5] W. K. Edwards, "Session management for collaborative applications", *Proceedings of the conference on Computer Supported Cooperative Work*, 1994, pp 323 – 330.

[6] D. F. Ferraiolo, J. F. Barkley, and D. R. Kuhn, "A role-based access control model and reference implementation within a corporate intranet", *ACM Transactions on Information and System Security* 2, 1 (Feb. 1999), pp 34-64.

- [7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol—HTTP/1.1*, RFC 2616, 1999.
- [8] S. Greenberg, “Personalizable groupware: Accommodating individual roles and group differences”, *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW’91)*, Kluwer Academic Press, 1991, pp. 17-32.
- [9] T. Horstmann, R. Bentley, “Distributed authoring on the Web with the BSCW shared workspace system”, *StandardView* 5, 1 (Mar. 1997), pp 9-16.
- [10] iPlanet, Object Signing Resources  
<http://devedge.netscape.com/docs/manuals/signedobj/>
- [11] ITU-T Recommendation H.323, *Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service*, November 1996.
- [12] J. B. D. Joshi, W. G. Aref, A. Ghafoor and E. H. Spafford, “Security models for web-based applications”, *Communications of the ACM* 44, 2 (Feb. 2001), pp. 38-44.
- [13] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*, Prentice Hall, 1995.
- [14] Lotus, SameTime, <http://www.lotus.com/home.nsf/welcome/sametime>
- [15] Microsoft Corp., NetMeeting,  
<http://www.microsoft.com/windows/netmeeting/>
- [16] PlaceWare, <http://www.placeware.com>
- [17] RSA Security, BSAFE Library  
<http://www.rsasecurity.com/products/bsafe/index.html>
- [18] S. A. Thomas, *SSL & TLS Essentials: Securing the Web*, John Wiley & Sons, 2000.
- [19] WebEx Communications, <http://www.webex.com>
- [20] E. D. Zwicky, S. Cooper, D. B. Chapman, *Building Internet Firewalls*, 2<sup>nd</sup> ed, O'Reilly & Associates, 2000.

---

L. B. Author is a web and collaboration technologies researcher at CollabWorx and Syracuse University, 111 College Pl, Syracuse NY 13210, USA (e-mail: beca@CollabWorx).

M. P. Author., was with Syracuse University, Syracuse NY13244 USA. He is now CEO of CollabWorx, 111 College Pl., Syracuse NY 13210, USA (e-mail: marek@CollabWorx).

[1] We use the notion of plug-in for historical reasons. In modern browsers the middleware component can be implemented using other mechanisms.

[\[Home\]](#) [\[About Us\]](#) [\[Products\]](#) [\[Downloads\]](#) [\[Search\]](#)

Copyright © 2000 [CollabWorx, Inc.](#) All Rights Reserved  
[Privacy Policy](#) | [Contact CollabWorx](#)





Welcome to the International Telecommunication Union

The ITU, headquartered in Geneva, Switzerland is an international organization within the United Nations System where governments and the private sector coordinate global telecom networks and services.



- [ITU Newsroom - ITU News magazine](#)
- [ITU Publications](#)  
The ITU is the leading publisher of telecommunication technology, regulatory and standards information. Many publications can be purchased through our [Electronic Bookshop](#) or the [ITU Publications Online](#) subscription service.
- [ITU Meetings and Conferences](#)
- [Job Vacancies](#)
- [Geneva Permanent Missions to the UN](#)
- [Plenipotentiary Conference, Marrakesh 2002](#)
- [ITU Telecom Asia 2002](#)
- [New Mobile Internet Report Released](#)  
**New !**
- [World Summit on the Information Society](#)
- [ITU Strategy and Policy Unit Activities](#)
- [ITU ENUM Activities](#)
- [Global Mobile Personal Communications by Satellite \(GMPCS\)](#)
- [International Mobile Telecommunications \(IMT\)](#)
- [E-Strategies](#)
- [ITU Gender Issues](#)



## *Leading the Web to its Full Potential...*

[Activities](#) | [Technical Reports](#) | [Site Index](#) | [New Visitors](#) | [About W3C](#) | [Contact Us](#)

The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding. On this page, you'll find [W3C news](#), links to [W3C technologies](#) and ways to [get involved](#). New visitors can find help in [Finding Your Way at W3C](#). We encourage you to learn [more about W3C](#).

### W3C A to Z

- [Accessibility](#)
- [Amaya](#)
- [Annotea](#)
- [CC/PP](#)
- [CSS](#)
- [CSS Validator](#)
- [Device Independence](#)
- [DOM](#)
- [HTML](#)
- [HTML Tidy](#)
- [HTML Validator](#)
- [HTTP](#)
- [Internationalization](#)
- [Jigsaw](#)
- [Libwww](#)
- [MathML](#)
- [Multimodal Interaction](#)
- [Patent Policy](#)
- [PICS](#)
- [PNG](#)
- [Privacy and P3P](#)

## ► XML Encryption, Decryption Become W3C Proposed Recommendations

*3 October 2002:* W3C is pleased to announce the advancement of [XML Encryption Syntax and Processing](#) and [Decryption Transform for XML Signature](#) to Proposed Recommendations. Encryption makes sensitive data confidential for storage or transmission. Comments are welcome through 31 October. Read about the [XML Encryption Activity](#). ([News archive](#))

## ► W3C Day 8 October in Sydney, Australia

*3 October 2002:* W3C Day is being held



Search w3.org

Search WWW

[Search W3C Mailing Lists](#)

### Mission

- [About W3C](#)
- [W3C in Seven Points](#)
- [Frequently Asked Questions](#)
- [Process Document](#)

### Contact Us

- [Quality Assurance \(QA\)](#)
- [RDF](#)
- [Semantic Web](#)
- [SMIL](#)
- [SOAP/XMLP](#)
- [Style](#)
- [SVG](#)
- [TAG](#)
- [URI/URL](#)
- [Voice](#)
- [WAI](#)
- [WebCGM](#)
- [Web Services](#)
- [Web Ontology](#)
- [XForms](#)
- [XHTML](#)
- [XLink](#)
- [XML](#)
- [XML Base](#)
- [XML Encryption](#)
- [XML Key Management](#)
- [XML Query](#)
- [XML Schema](#)
- [XML Signature](#)
- [XPath](#)
- [XPointer](#)
- [XSL and XSLT](#)
- [More topics ...](#)

on 8 October as part of the [Evolve 2002 Conference](#) in Sydney, Australia from 8-11 October 2002. Janet Daly, Hugo Haas, Dean Jackson, and Joseph Reagle of the W3C Team will be on hand, focusing on the W3C Privacy, Web Services, XML Signature, XML Encryption and XML Key Management Activities. Read the W3C Day [programme](#). ([News archive](#))

## ► XML Accessibility Guidelines Working Draft Published

*3 October 2002:* The WAI Protocols and Formats Working Group has released an updated Working Draft of [XML Accessibility Guidelines](#). The draft is a guide for tools designers and authors of XML formats. It explains how to design accessible XML applications that lower barriers to Web accessibility for people with disabilities. Comments are welcome. Read about the [Web Accessibility Initiative](#). ([News archive](#))

## ► W3C Launches Hungarian Office

*24 September 2002:* W3C is pleased to announce the launch of the [W3C Hungarian Office](#) (in Hungarian) based at the [Computer and Automation Research Institute](#) (SZTAKI) of the [Hungarian Academy of Sciences](#) (MTA) in Budapest, Hungary. Daniel Dardailler, Marie-Claire Forgeue, Max Froumentin, Ivan Herman, László Kovács, and

- [Contact W3C](#)

### Get Involved

- [Participate](#)
- [Mailing Lists](#)
- [Translations](#)
- [Open Source Software](#)
- [World Offices](#)
- [Employment](#)
- [Send comments about this page](#)
- [Subscribe to W3C Weekly News](#)

### Member Area

- [Member Home Page](#)
- [Current Members](#)
- [Join W3C](#)
- [Get Member Password](#)

### W3C Team

- [People](#)
- [Past Talks](#)
- [Upcoming](#)

### Past News

- [News Archive](#)
- [Press Releases](#)



*Dedicated to preserving the central coordinating functions of the global Internet for the public good.*

---

## **Application for User (Registered) Port Number**

---

The User (Registered) Ports are those from 1024 through 49151. (See <http://www.iana.org/assignments/port-numbers>.)

The IANA needs a technical description on your proposed use of a user port number. We require enough detail to understand how your application uses the network. Once we have the above information in hand, and understand it, we can assign a user port number.

Please note that a particular application or service should be able to operate only one registered user port number. For applications or services that offer multiple functions it is usually possible to use one port as a multiplexer or rendezvous service. That is, the client always initiates the use of a service by contacting the rendezvous port and indicating in its first message which function is needed. The rendezvous service then either (A) creates (forks, spawns) a process to perform that function and passes the connection to it; or (B) dynamically selects a (high-numbered) port and starts a process to perform the function listening on that port and sends a message back to the client telling it to call the new process on that port.

If you consider your information to be proprietary, we do have a non-disclosure agreement available. The non-disclosure agreement should be signed before you disclose any proprietary information to us.

Please see below for the application:

---

We need at least the following information, which is for our internal use only.