

Improving File Sharing Security: A Standards Based Approach

**A Xythos Software White Paper
January 2, 2003**

Abstract

Increasing threats to enterprise networks coupled with an ever-growing dependence upon them has elevated network security to the top of almost every IT manager's priority list. When the Internet is used as part of the enterprise's communications network, improving system security typically becomes a standard operating procedure. Internet file management (IFM) provides a framework for significantly improving the security of all file information exchanged within and between organizations, utilizing current security standards and protocols. This paper explores the primary security components of IFM including authorization, authentication, access control and data integrity, as they are employed by the Xythos WebFile Server. Although this paper is not designed to be a "how to guide", it is intended to demonstrate that when these primary security components are used in combination, they can often substantially improve the overall security of all enterprise file information.

HTTP/WebDAV Security Characteristics

The Xythos WebFile Server interacts with Web browsers and WebDAV clients using the HyperText Transfer Protocol (HTTP). Web browsers use the basic HTTP 1.1 requests and receive data formatted for presentation using HTML. WebDAV clients use WebDAV requests and receive XML data which is formatted on the client (For a more detailed discussion of WebDAV, please refer to [WebDAV Essentials.htm](#)). Because WebDAV is an extension of HTTP, all the same security mechanisms apply:

- HTTP connections can use transport-layer security (SSL or its successor, TLS) to provide data integrity
- HTTP implementations must support both basic and digest authentication, two standard mechanisms for authenticating users via passwords
- Many HTTP implementations support advanced authentication mechanisms

As requests to the WFS are HTTP-based, security precautions for HTTP access can also be used. Such precautions include firewalls, reverse proxies, and other advanced Web security techniques and software solutions. Therefore, as improved security precautions become available, the WFS allows administrators to integrate or "plug into" these latest security enhancements.

Because WFS uses HTTP, every WFS file and folder has a unique HTTP URL. This URL is used to access the file through either the Xythos WebUI or via WebDAV. The WFS also makes sharing files easy because users can share those files by sending other users a file's URL via email or even Instant Messaging (IM). Sharing files in this manner already improves information security exchange because users only send the resource's address. Typically, when users share files by sending the actual file, they risk a security breach because email and IM protocols are not usually encrypted. Even when email and

IM protocols are protected, often one or more extra copies of the file are stored on user's systems, multiplying the chances of that file becoming public. Sending the URL via email or IM not only improves messaging system performance, but it also ensures that the recipient can only access the file when they have been granted specific permission to do so, eliminating the opportunity for files to be accessed by unintended participants.

The most common HTTP security accessory is a proxy server. Proxies are used to determine which traffic is permitted to pass through an internal network from the public Internet. It is necessary to allow WFS HTTP traffic through a proxy in order to access a WFS machine located inside a corporate firewall. However, HTTP access can be restricted such that it must pass through the proxy server. The proxy can ensure that traffic can only go to the WFS machine. As HTTP access is indirect and limited, it is difficult to attack such a configuration. It is also possible to host the WFS outside of the firewall and secure the machine(s) with other technologies. It is important to note that the WFS can be secured in the same manner as any other Web applications already operating within the enterprise.

Data Integrity and Server Authentication

The most common way to provide data integrity with HTTP is to use the Secure Socket Layer (SSL), or its successor Transport Layer Security (TLS). (TLS is the successor to SSL version 2 and addresses transport security with additional security features. However, currently TLS is not as widely supported by common client software.) Both SSL and TLS can secure any TCP connection. These two versions do interact securely because clients and servers can automatically negotiate the most secure shared version.

SSL and TLS are almost always used for two purposes:

- To encrypt all traffic over the TCP connection in both directions. This secures all data that is transferred against integrity attacks and protects the privacy of all data
- To authenticate servers in order to certify that client systems are sending passwords and data only to the correct server.

SSL or TLS support for HTTP is provided by the Web or application server that the Xythos WebFile Server uses. For example, Apache 2.0 can use `mod_ssl`, which can support TLS as well as SSL and which also selects the appropriate version to communicate securely with each client connection.

Client Authentication

Client authentication provides a method to identify the user accessing a system and is the first step in determining whether actions are authorized. All HTTP clients and servers supporting the 1.1 standard must support two authentication mechanisms: Basic and

Digest. Basic authentication sends the password to the server as clear text within the request. Basic authentication does not present a security problem if SSL/TLS is used to protect the privacy of the connection. However, if the connection is not protected with SSL/TLS, Digest authentication should be used instead of Basic. In Digest authentication, the password is encrypted together with a server challenge. Digest authentication makes it difficult to discover the original password even if an attacker can examine all the data transmitted in the connection.

Both Basic and Digest authentication are supported by the Xythos WFS. In addition, the WFS has an open API that allows proprietary or advanced authentication systems to be integrated. It is also possible to configure the SSL/TLS module to require advanced client authentication such as requiring certificates before allowing a connection. Support for these methods allows each customer to define their own requirements for authentication so that they comply with their overall security standards.

The pluggable authentication model for WFS also permits future changes to the authentication mechanism even after the WFS is deployed. Thus, as new security standards are developed, customers can implement the latest authentication technology such as biometrics (fingerprints or retinal scans)

User/Group Management

Standalone User/Group Management

Xythos WFS includes basic user and group account management for situations where the WFS must stand alone. Such a configuration can be useful during a pilot program or early deployment. It can also be useful in situations where the WFS implementation is used by a number of people from different organizations who do not otherwise share a list of user accounts, groups, and login credentials. In this mode, users and groups can be managed through the WFS Administration GUI. A bulk loader is also available for customers to create or modify users, groups, and directories in bulk (using a command line script and comma-delimited text files).

Directory Service Integration

A directory service provides a single place to manage user accounts, group membership, and logon credentials. Organizations are rapidly adopting directory services to improve security by reducing system management complexity for both administrators and users with centralized account management and by providing a single secure user credential for application services. Centralized account management and a single secure user credential eliminates the problem of requiring users to remember multiple passwords, as well as requiring IT managers to handle multiple system accounts and passwords for one set of users. Xythos WFS helps address this goal by integrating with a directory service, including Microsoft Active Directory, Novell NDS or eDirectory, Netscape Directory Server, and any LDAP server.

Combined Management

Often, a directory service is used to manage users and credentials and some groups. For example, a university's IT department might set up global groups for faculty and staff. However, the IT department may not want to be involved in defining and managing the many temporary student groups often needed for collaboration on class projects. The Xythos WFS provides the flexibility of storing groups on a directory service, in the WFS database, or in both places. Groups can consist of global groups which every user can see and grant permissions to or, private groups that are managed by an individual user.

Advanced Security Systems

The WFS also supports advanced security systems which can provide functions such as, access profiling combined with user authentication together like Netegrity SiteMinder. These systems can be integrated with the WFS security model using the WFS Java API.

Access Control Lists (ACLs)

Access Control Lists (ACLs) are generally the primary mechanism controlling access to and the use of shared files. The ACL for a resource (file or directory) determines which users and groups can read, write, delete, and administer that resource. Every WFS folder and file has its own independent ACL providing unlimited management flexibility and control to the system.

The resource owner is typically the primary user who manages access control for that resource. In addition, the resource owner can delegate "administer" permission for that resource, so that other users can help manage access control settings. The WFS System Administrator can change any ACL in the system, but typically their intervention is not needed because users are given the power to manage their own files.

WFS ACLs behave according to the following rules:

- Any user or group known to the system can be used in an ACL.
- If the user is granted a permission directly (e.g. Joe has "read" permission) or the user is granted permission indirectly through a group (e.g. Joe is a member of acct-group which has "read" permission) then, any request authenticated as that user's is allowed.
- If a permission is not granted through a matching row in the ACL, then access is denied.
- Every ACL has an entry for "Public." If a permission is granted to "Public", all users authenticated or not, may access the file with that permission. Giving "Public" read permission is equivalent to placing the file on a public Web server such that anyone with that URL will be able to download that file. This is a simple way to easily operate a public Web site because users can read files

without requiring authentication.

- Every ACL has an entry for “Authenticated Users”. “Authenticated Users” allow similar access as “Public”. However, users are required to be logged in to be allowed that access.
- A new file or folder placed in a directory will inherit its permissions from the parent directory. Thus, placing a file in a folder that is readable by “Public” automatically makes that file readable by “Public”. In this way it is easy to manage access control for a hierarchy of files with consistent permissions.
- “Inheritable” permissions (“Inherit Read”, “Inherit Write”, “Inherit Delete”, “Inherit Permission”) can be set on a folder. The set of inheritable permissions are used to initialize the access control for a new file or sub-folder.
- Directory listings display the contents of a folder and are defined by the “read” access control. In order to view a directory listing on a folder, the user must have “read” access to the folder. (Note that access to a folder can be granted on either an individual or group basis.) Additionally, the user must have “read” access to each of the items within a folder in order to view that item. Different users can see different listings from the same folder according to how “read” permissions have been granted on the child resources.

It is possible, and can also be very powerful, to extend the WFS ACL implementation. For instance, a customer could easily implement a system where users could not change files between midnight and 4:00 am. During the remaining hours of the day users would regain “write” permissions. This example could be employed to support a network backup schedule, for example.

Ticket-based Authorization

WFS users may also want to share files with others who do not have credentials on the system. These external users can be customers, suppliers or simply other collaborators. In this case, the WFS provides an additional file sharing feature known as “tickets.” A ticket is a temporary password for a file. The ticket has a lifetime, a maximum number of uses, and provides a choice of “read” or “read and write” permissions. As an example, a file owner can create a ticket for that file which lasts for one day, which can only be used once, and only allows the user to view or, “read” the file. A ticket can also be deleted by the file owner at any time, which immediately renders the ticket invalid and useless to others.

Tickets can be useful when sending email to a mailing list. The file owner may not wish to make the file readable by “public” but needs to guarantee that each member of the mailing list can properly access the file. Sending a URL to the mailing list with a “read” ticket grants all members the same access without requiring the sender to grant individual permissions to each user. When tickets are used together with the WFS’ file logging

feature the file owner can accurately monitor each recipient's access behavior, permitting them to observe exactly when a file has been accessed, how many times its been accessed, etc.

Tickets are an optional WFS feature that may be disabled by the system administrator.

Protection Against Denial of Service and Other Attacks

Denial of service attacks typically attempt to debilitate a server by overloading it with enough requests to stop it from being able to respond to legitimate requests within a reasonable time. For example, on any Web server, a denial of service attack could be a number of clients which all ask for the same large Web page at roughly the same time. Xythos WFS provides an added level of protection from these types of attacks by establishing bandwidth and quota controls.

Directory quotas determine how much information can be stored in a specific directory. Even if the underlying file system has enough disk space to allow a new file to be added, the server refuses to allow the new file to be written once the directory quota has been reached. This feature helps to minimize malicious file uploads to the system.

In addition to storage quotas, WFS also implements bandwidth quotas to prevent server overload. Because files can be shared with any number of users, a small file requested by numerous users can quickly create a substantial load on a server. To reduce the danger of such an incident, Xythos WFS allows administrators to place a bandwidth quota on a directory. Once that directory has used all of its available bandwidth for a specific time period, all future accesses to information contained in that directory will fail and the server will not allocate resources to respond to requests that are over quota.

Administrators can provide further protection from denial of service attacks by using firewall software that is capable of limiting requests of certain file types or traffic from certain addresses.

Summary

The Xythos WFS has been designed for maximum file security and management flexibility. Xythos WFS allows administrators to use best-of-breed security technologies that exist today and allows for easy migration to the security enhancements of the future. With Xythos WFS, there is no limit to the amount or type of security an enterprise can use. Equally important, the WFS provides a simple solution for users to begin sharing information more safely which encourages adoption and use. For organizations seeking to improve the overall security of information exchanged over their networks the combination of simple and flexible security methods that are designed to work together with existing application standards should become a standard selection requirement.