RealSystem 5.0
Security Features Whitepaper

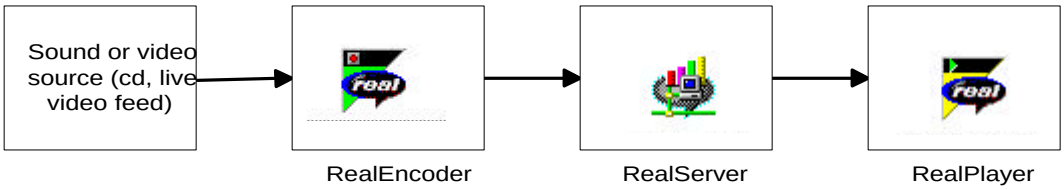
Who Should Read This?

This whitepaper is aimed at anyone:

- considering a streaming media server and unsure of what to look for
- comparing the security features of the RealSystem Commerce Solution with another streaming media server
- wanting to know more about security issues surrounding streaming media

What is RealSystem?

RealSystem 5.0 is a complete client-server streaming media solution for making sound and video available to thousands of users over the Internet or an intranet. RealSystem includes three components: RealEncoder™, which converts audio and video files to digital media clips, RealServer™, which streams the clips in real-time, and RealPlayer™, which plays the media clips. Use RealSystem with your Web server to make files available over an intranet or the Internet.



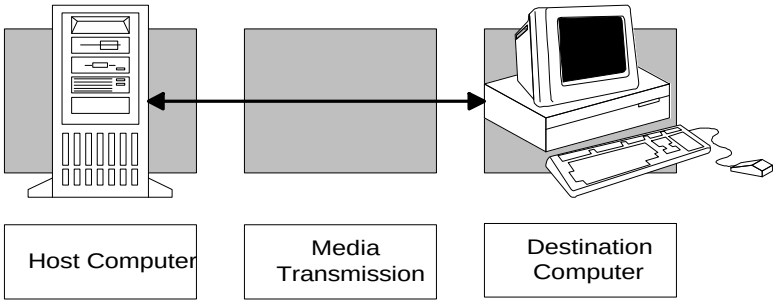
My Web Server is Secure—Isn't That Enough?

Until now, streaming media servers have relied on the security features of Web servers to protect their media streams. But the differences between these types of servers can mean that traditional Web server security methods will not provide true security to streaming media servers.

These differences can mean that although Web pages are safe from unauthorized users, the actual streaming media files can be intercepted by a third party. A pay-per-view movie could be watched by someone who hadn't paid for it, and a confidential corporate briefing could be viewed by a competitor.

In this paper, we describe security threats to media servers and some typical solutions, then we discuss the methods our streaming media server uses to repel those threats.

This whitepaper assumes that appropriate physical and virtual measures such as firewall computers, hardened kernels, and physical access restrictions have been taken by qualified systems managers to protect the fundamental elements of the network, hardware, and operating system software of the host computer.



Security Problems Facing Web and Streaming Media Servers.

There are three arenas where security can be compromised: file security on the host machine, transfer of files across the Internet or intranet once they've left the host machine, and storage of those files after they reach their destination.

File Security

After physical restrictions are in place, the first place security is needed is in controlling access to files. There are two parts to this: restricting access to specific files or directories, and verifying the identity of users who try to view those files.

Restricting Access to Specific Files

The top three methods for protecting files hosted on Web servers are Security Through Obscurity, File System Security, and Application Level Security.

Security Through Obscurity

Security Through Obscurity relies on a simple principle: if you don't know where a file is, and you have no way of locating it, you can't view or copy it.

In its simplest implementation, URLs to "secure" Web pages or media files are given only to select users, and no other Web pages link to the secure URLs. More complex implementations may rely on a CGI script and dynamic URL generation: when a user clicks a link on a Web page, the link activates a program which assigns the requested file a complicated URL address based on a temporary relative address, and passes the new URL to the browser. Thus the actual file locations are difficult to find.

The security concern with Security Through Obscurity is that users *can* discover the actual URL address of "hidden" media files, and once they do, nothing prevents them from freely accessing the file and from sharing that address with others.

File System Security

File System Security relies on controls built into the operating system of the host computer that block access to users with unknown IP addresses or restricted network permissions. The administrator must enter the IP address or permission of each authorized user into a database, which the serving computer examines before granting access to a particular file or directory.

While offering significantly greater protection for files than Security Through Obscurity, this method is only suitable for tightly controlled intranets or other situations in which each person only uses one computer. An executive who developed a media presentation at their desktop could not necessarily give that presentation from the boardroom computer.

Because file system security allows access to computers based only on IP addresses or permissions, it can be cumbersome to add authorized users. Another problem is that IP addresses can be spoofed (altered or masked) by intruders.

File System Security is not practical for medium-to-large intranets or the Internet because the administrator must list each authorized computer's IP address and every user must use only the one computer to which he or she has been granted access if the system is to be considered secure.

Application Level Security

In Application Level Security, the server does not relinquish files until the requesting user provides sufficient identification, usually in the form of a name and password that have either been entered into a database by an administrator, or sent to the database via a secure Web connection. The HTTP/1.0 Protocol is an example of this method.

The advantage of Application Level Security is that file access is tied to user name and password. Even if users know the exact URL of a secured file, they can be denied access until they can satisfactorily prove their identity.

How RealSystem Restricts Access to Specific Files.

RealSystem 5.0 uses Application Level Security modeled on the HTTP/1.0 RFC 2069 standard, in which the username and password are sent as encrypted text. With this method of security, users must provide satisfactory identification before they are given access to files. Furthermore, they must supply their names and passwords with each visit to the secured site.

In addition, RealSystem offers three different types of access to files:

- *Event-based access*: The visitor registers or is registered for unlimited access to one or more specific media clips.
- *Duration-based access*: A visitor is granted the right to access a clip for a specific length of time. (For example, five total hours of viewing, spent over any period).
- *Calendar-based access*: A visitor is granted the right to access a clip until a certain date (for example, unlimited viewing of any or all of some number of specified videos during the next week).

In all models, if the date and time of expiration arrives while the visitor plays a clip, transmission of that clip to RealPlayer is stopped, and an appropriate message is displayed.

A single RealServer can simultaneously deliver multiple types of access, up to one thousand concurrent viewers.

Verifying User Identity.

Popular methods of verifying the identity of viewers who attempt to view restricted files include the use of cookies, certificates, and username/password authentication.

Cookies

Cookies are character strings delivered to a storage file on the user's computer by the host server. Cookies have the advantage of being changeable by the host server, so that they can carry variable information, and they are user-transparent.

However, cookies are also easily changeable by the receiver or useable by any other person who can intercept the cookie during transmission, so **cookies do not provide a high degree of security**, although they are useful for lower-security applications.

Certificates

Like cookies, certificates are character strings delivered to the user's computer. Unlike cookies, certificates contain encrypted information about the sender that can only be unencrypted by a trusted certificate authority.

Certificates can be generated by a third party that both the server (the sender) and the browser (the receiver) have agreed to trust. Certificates can be used as proof that a user's computer is not being spoofed – that the data is in fact coming from the browser to which a unique certificate was assigned.

Certificates have the advantage of being highly secure – they prove that the computer, browser, player, or user to which they are attached is the original item to which the certificate was issued, not an illegal or forged copy. They are also highly user-transparent, like cookies.

Like cookies, **certificates do not necessarily identify the person sending a request; they may only identify the sending computer or software program.** In addition, unlike cookies, the procedure for the sender to get a certificate and the receiver to set up a verification process can be lengthy and expensive.

Username/Password Authentication

In this form of security, users are required to provide a password before they are given access to files. The HTTP/1.0 Protocol, as described previously, is an example of this method. Before users are given access to any files on the host computer, they must supply a pre-authorized name and password.

This method has the advantage of being portable (files can be accessed from anywhere on the Internet), while also offering true security. The system doesn't depend on hiding the files, but directly blocks access to files and media.

How RealSystem Verifies User Identity.

RealServer can use either of two methods of verifying the authenticity of a user name: Player-based Authentication, in which the user's RealPlayer provides the host server with a RealPlayer identification number (a process that is transparent to the user), and User-based Authentication, which requires the user to enter his or her name and password. Player-based Authentication uses an identification number stored on the user's hard-drive by the RealPlayer, and User-based Authentication uses an implementation modeled on HTTP/1.0 security, RFC 2069, to authorize the name and password pairs.

Player-Based Authentication

Player-based Authentication uses an identification number stored on the user's hard-drive – a method similar to but much more secure than a cookie, in that the ID number is less easily changeable. When the RealSystem is configured in this mode, Player-based Authentication permits user-transparent access to secured media while reducing the likelihood that the RealPlayer ID will be shared with other users.

User-Based Authentication

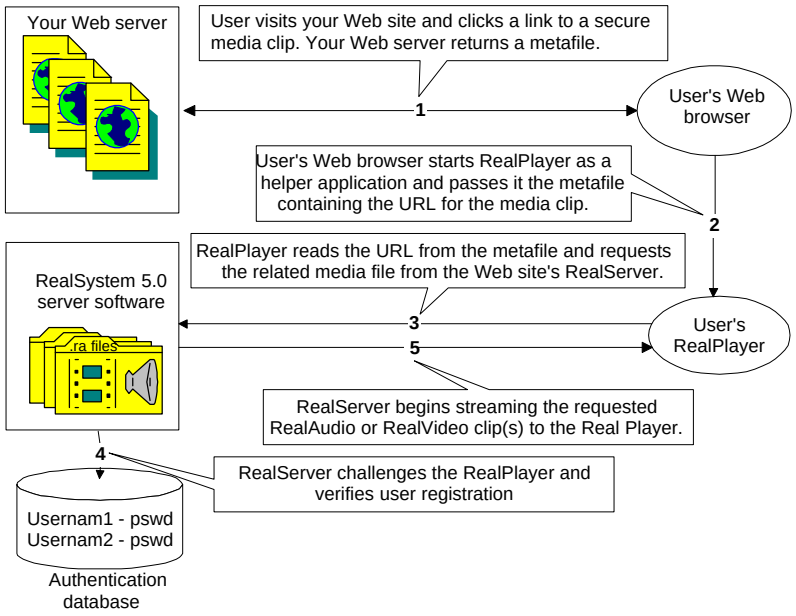
In its high-security mode, RealSystem 5.0 challenges the browser to provide a name and password. RealSystem uses protocol modeled on that specified by HTTP/1.0 RFC 2069. The password is sent as a time-stamped 128-bit MD5 hashed checksum (number), which is not useable by an unauthorized user even if it's intercepted.

	Player-based Authentication	User-based Authentication
Registration Method	1. Web page prompts visitor to create a user name or password. 2. Visitor clicks **Submit**. 3. Server activates RealPlayer and returns a confirmation Web page. 4. Visitor confirms the submitted information. User name and password are sent via a secure Web page.	1. Web page prompts visitor to create a user name and password. 2. Visitor clicks **Submit** and is confirmed in a single step. User name and password are sent via a secure Web page.
View Secured Content	No special action required from visitor.	When user tries to access secured clip or directory, RealPlayer prompts user to enter name and password.
Privileges Established by Administrator	Administrator sets visitor's privileges at the same time as RealPlayer installation or afterward.	Administrator sets visitor's privileges at any time. Administrator can distribute user names and passwords.
Sample Uses	■ Collecting demographics from Web site visitors ■ Fan clubs ■ Corporate training	■ Pay-per-view movies ■ Executive briefings for other executives ■ Sharing information with business partners ■ Premium content subscription

Player-based Authentication involves less viewer interaction, but each RealPlayer must be registered individually by the viewer or central administrator. User-based Authentication access privileges can be set by the central administrator prior to registration.

Player-based and User-based Authentication can be run simultaneously on the same Web pages and applied to different clips, but this requires two RealServers, one for each type of Authentication. In addition, you can customize Authentication to create different types of payment and registration procedures.

The diagram below outlines the interaction between a visitor to a Web site and the RealServer Authentication process.



RealSystem and Certificates.

RealSystem 5.0 was designed for minimal user expense and transmission overhead, and so does not require the use of certificates to operate.

However, you can use certificates with RealSystem; RealNetworks provides front-end templates and an open API, and users can create their own custom code or call on RealNetworks consulting to add certificate use to components of the system. RealNetworks has not based RealSystem entirely on certificates because RealNetworks doesn't want to force the heavy overhead of certificate management on our customers. Instead, RealSystem allows one-step registration, transparent access and demographic tracking, and the MD5 hash many certificates use.

Transferring Files Across the Internet.

Giving select users access to information is only the first part of the information distribution process; users must then be able to transfer that information (in the form of a Web page or media stream) across the Internet back to their computers without any unauthorized person seeing what's in the stream.

Most methods of protecting information as it travels involve encrypting and/or digitally signing the information, both of which can be cumbersome for streamed media.

SSL Encryption

Secure Sockets Layer (SSL) is the popular encryption technology used to establish a secure Web connection. When you place a credit card order via a Web page, your browser and the server are probably using SSL to keep the transaction unintelligible to prying eyes.

SSL involves an initial negotiation between the user and host computers to establish an agreement on method of encryption. Subsequently, all information transferred between the computers is encrypted with the agreed-upon method. Even if an unauthorized party was able to intercept the data, the party would not be able to decode it.

The problem with SSL is that although Web pages can be sent through it, most streamed media cannot. Web pages are sent over TCP, but most streamed media files are sent over the UDP protocol. UDP is outside the protection of SSL.

Digitally Signed Encryption

For even further security, it is possible to digitally sign as well as encrypt data.

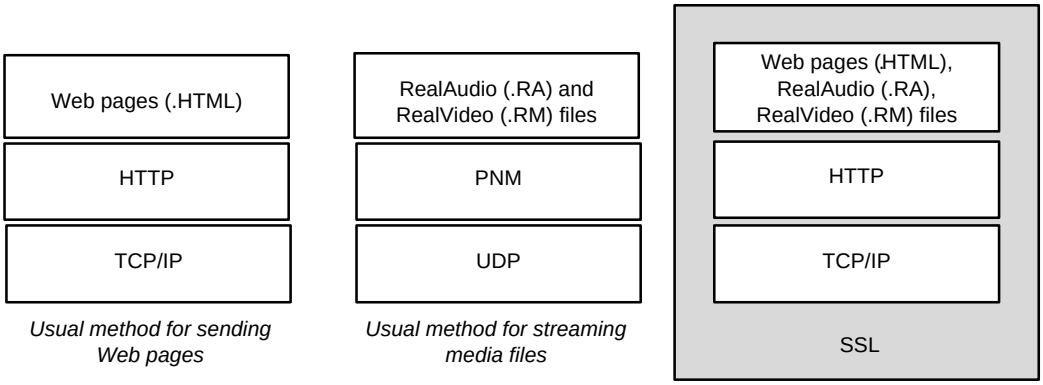
Digitally signing of data generally refers to the process of dual-encryption—sealing a message already encrypted with yet another encryption method, so that the computer receiving the information can verify the source of the information. RSA and Diffie-Hellman are two common algorithms for encrypting and signing data.

The challenge in using encryption and digital signatures with media streams is that unlike Web pages, which are downloaded in large blocks, a media stream is an ongoing flow of smaller packets. The additional load added to each packet by the digital signature will significantly slow transfer and processing. The recipient of such a secured stream would need a dedicated high-bandwidth Internet connection and at several hundred MHz CPU to be able to experience the media without significant degradation in quality. Most such users are on a corporate LAN or WAN already, and therefore don't need the high security in the first place.

How RealSystem Transfers Files Across the Internet.

Like other streaming media systems, RealSystem 5.0 does not currently encrypt or digitally sign its data stream. If the non-hashed data packets were intercepted, it would take an engineer who was very familiar with the RealNetworks protocols a significant amount of time to decode any useable information.

Moreover, **unlike many other media systems, RealSystem 5.0 can enclose the media stream in the SSL tunnel by sending it via TCP**. Administrators should be aware that such security has a cost; like any real-time media sent over TCP, SSL encrypted media streams will experience some quality reduction due to the increased ratio of overhead encryption data. (The computers spend so much time and encrypting, transmitting additional encryption data, and decrypting that real-time transmission of media becomes extremely difficult except over a very high-bandwidth connection with extremely fast computer processors at each end).



Usual method for sending Web pages

Usual method for streaming media files

Streaming media files within SSL

Storing Files at the Destination.

After the information has traveled to the user's computer, the information must remain where the host computer expected it to remain, and not proliferate.

On the user side, maintaining security after file transfer implies the host server has a mechanism for preventing the retransmission of data. Because Web page transmission involves downloading an entire copy of the page, this type of security is difficult for standard Web servers. However, because media files are streamed, it is possible to play each bit of data as it is transmitted, and then destroy that data.

On the host side of the transaction, maintaining security after file transfer implies encrypting passwords and password-protecting internal files, so that even if the host computer's security is penetrated, the server cannot be tampered with and users' passwords cannot be stolen and reused.

How RealSystem Provides Security After Files are Sent.

The RealSystem encoding tools allow you to specify on a file-by-file basis whether the user can save a media stream. In this way, you can control whether your media files are copied and distributed without your knowledge. This option is particularly useful if you are broadcasting copyrighted material that you don't want anyone else to copy without your express permission.

In addition, RealSystem can detect attempted identity theft. If someone tries to access content using the name and password of another person who's already viewing content, RealSystem either sends a warning message or prevents access.

Conclusions.

RealSystem 5.0 provides a complete set of tools to create media clips and protect them on their route from the host computer, across the Internet or intranet, and at their destination.

The RealSystem Commerce Solution is the only major streaming media system to directly prevent unauthorized access to media clips through embedded server technology. Other major streaming media systems rely on "hiding" or rotating the media clip URLs, which both increases reliance on a Web server and its limited security, and does not prevent discovery of the URL of the media clip.

The Commerce Solution improves security of media streams by embedding the username-password challenge into the streaming RealServer, so that authorized users can access content from any computer, and unauthorized users are prevented from doing so.

Traditionally, streaming media systems have favored ease over security. The Commerce Solution not only maintains and improves on ease of use and cost, but also provides more embedded security functionality.

The RealSystem 5.0 Commerce Solution is the ideal product for applications such as pay-per-view movies, detailed demographic tracking, and executive briefings.

For More Information.

To learn more about the RealSystem 5.0 complete Commerce Solution and how it can solve your needs for a more secure streaming media system, visit our RealNetworks Web site at www.real.com, or contact us directly at (800) 444-8011.

RealSystem and the Commerce Solution are easy to set up on your own, but if you want to work with the experts, our RealNetworks Consulting Group can show you the ropes. You'll find the Consulting Group Web site at www.real.com/consulting.

We look forward to hearing from you.

RealNetworks

1111 Third Avenue, Suite 2900
Seattle, Washington 98101
www.real.com

Phone: (206) 674.2700
Fax: (206) 674.2699