

# Mainframe-to-Web Security Overview

White Paper prepared by OpenConnect Systems, Inc.





## Web-Mainframe Security Overview

Is your mainframe safe?

It probably is, at least for the moment. The real question is: Will you know how to keep your mainframe safe when you Web-enable it?

This white paper discusses general Web-mainframe security issues. It also offers specific, practical advice for organizations that want to take advantage of the business and operational benefits associated with Web-enabling their mainframes while still protecting their systems.

The mainframe remains the most secure computing platform, even when connected to the Web. But don't relax too much. The mainframe is still new to the Web. The bad guys haven't yet figured out how to exploit the situation, but they're no doubt working on it. Whether you're thinking about Web-enabling your mainframe or already have, security should be high on your to-do list.

Start by choosing a robust, scalable and proven Web-to-host solution. There are plenty of choices. Due diligence is demanded here. Find a company that has been around for a long time, one that has overseen many deployments in a variety of verticals. And make sure that company understands not just Web-to-host, but security.

### WebConnect by OpenConnect

One example of a solution many companies are choosing: WebConnect by OpenConnect. WebConnect is client-server based software that provides secure browser-based emulation to mainframe, midrange and UNIX systems. The advantage of a solution like this is that it enables enterprise organizations to provide suppliers, partners and employees with access to vital applications and information while maintaining security.

Thus organizations reap all the benefits of Web-enabling their mainframe - increasing productivity and profits and unlocking valuable information in real time – while maintaining secure host connectivity.

But organizations considering any of these steps must do their homework.

Any solution under consideration should provide secure SSL encrypted information migration and access without requiring modification to the host – as does WebConnect. And only WebConnect's patented "persistent connectivity" technology can support tens of thousands of concurrent browser-based users.

WebConnect provides:

- Non-intrusive access
- Average install time of less than 15 minutes
- Protection at application, session, transport, and host points
- Information migration without modification at host
- Patented secure connection and authentication encryption

When exploring ways to secure your Web-enabled mainframe, the need to find an experienced partner cannot be overstated. OpenConnect Systems has been a leader in server-based solutions that integrate SNA with TCP/IP environments for over 16 years. WebConnect expands on that expertise in allowing secure browser-based access to IBM enterprise hosts. With built-in industry standard RSA encryption, support for standard mainframe-based security systems such as RACF (Resource Allocation Control Facility), ACF2 and Top Secret, plus SNA session level security, it is the most secure solution available.

Flexibility is another concern. WebConnect, for example, can be deployed directly on the host system. WebConnect for S/390 runs as a UNIX System service on IBM's zSeries IFL, and WebConnect for AS/400 uses the latest version of the OS/400 operating system along with an IPDS board to interface with WebConnect NT. WebConnect also has traditional, 3-tiered deployments to UNIX, Windows and LINUX environments.

Drilling down: WebConnect configuration

To illustrate the issues involved with configuring leading products, consider the specifics of WebConnect. WebConnect can be used in various combinations to enable varying levels of security.

The main areas of concern dealing with security in WebConnect are:

- Data privacy (encryption)
- Data integrity (message authentication)
- Firewalls and network topology
- Authentication of client to server
- Authentication of server to client
- Authentication of host to server
- Authentication of CGI access

It is important to establish a desired overall network topology and security requirement criteria before starting to configure WebConnect. For example, are firewalls to be used and do sessions on the internal network require encryption? WebConnect can be designed into a network topology with or without firewalls; it can also run encrypted and non-encrypted sessions simultaneously



Knowledge of the security and topology requirements allow for a simpler WebConnect installation and customization. In most cases security administration is done entirely at the server. No client administration is required unless X.509 certificates are used for client authentication. In this case, a certificate should be installed into each connecting browser and JRE.

If server authentication is a requirement, typically a concern when deploying over a public network, the Secure Socket Layer (SSL) option should be used. Server authentication in SSL is provided through X.509 certificates. SSL also provides message authentication to prevent message tampering.

Client authentication refers to verifying user's attempting to connect to the system. Users who cannot be authenticated are denied further access. WebConnect supports HTTP basic authentication (user ID and password), SSL client certificates, and integration into existing commercial Identify and Policy management systems such as RSA SecurID and ClearTrust.

In the case of integrating into the customer's existing authentication mechanism, client authentication is first established from the browser to a Web server. A token-passing scheme is then used to extend that authenticated state to the WebConnect applet running in the browser.

The following two message encryption types between the WebConnect server and the client are provided:

#### SSL

A choice of eight cipher suites (also includes the RC4 algorithm)  
Client/Server encryption algorithm negotiations

RC4 from RSA Data Security, Inc.

- 40-bit encryption key
- 128-bit encryption key

Note: SSL and RC4 cannot be used simultaneously.

WebConnect uses a communication protocol called JCP (Java Control Protocol) to transfer information between the Java applet and the mainframe. The SSL version of the protocol is referred to as JCPS.

## Firewalls and Network Topology

The multi-tier security approach in WebConnect is designed to complement existing Internet security, not replace firewalls and other security devices. Due to the many types of firewalls that can be deployed in various configurations, not all possibilities are discussed here.

In general, WebConnect works with any transparent or masquerading firewalls that operate below the application level - those that filter IP packets rather than content. Proxies, which do filter content, are supported in most but not all configurations. Generally client-side proxies that can be configured for SSL tunneling are supported. However, server-side reverse proxying is not normally supported since this type of proxy configuration typically rejects all non-HTTP traffic. (WebConnect's HTML Client is one solution that resolves this constraint.) To operate in a reverse proxy environment, a separate port for the WebConnect JCP, or JCPS service, or both, must be configured. The proxy is either bypassed entirely, or it is set up in a simple packet-filtering mode for the non-HTTP traffic to the server.

Although WebConnect acts as its own proxy server, certain network configurations require a separate proxy server. A companion product to WebConnect often used in these secure installations is the JCP Proxy Server. The JCP Proxy Server is a security component protecting the points of entry where the Internet meets the private network. This server resides between an external client application, such as the WebConnect Java emulation client, and the WebConnect server. It intercepts all JCP requests directed to WebConnect and validates each to be in accordance with the JCP Protocol specification. If the request does not conform to the specification, the request is rejected and the connection between the client application and the JCP Proxy Server is terminated. Otherwise, the request is forwarded on to the WebConnect Server. The JCP Proxy Server provides the following functions:

- Provides certificate-based authentication and encryption by supporting SSL on all connections between client applications and the WebConnect Server.

- Filters all requests and rejects any request that does not conform to the JCP/JCPS Protocol specification.

## Using SSL

It is recommended that you use SSL on the standard HTTPS port, 443, to connect to the WebConnect server, although any available port can be used

Besides the obvious data security issue, this helps prevent modifying existing rules configured in a firewall or proxy server that may restrict connections to use standard ports only. It may be necessary to modify a connection time-out parameter in the firewall or proxy server to prevent sessions from being disconnected, as web traffic is typically transient rather than persistent.

Client-side proxy servers are supported through the SSL tunneling protocol. Typically, the SSL tunneling protocol must be used even if SSL is not, as most proxies will not support a persistent connection any other way. If the use of client-side proxies is anticipated, for the reasons stated above, it is recommended to use SSL on the standard HTTPS port, 443.

Current browsers do not expose their proxy configurations to java applets, so the WebConnect applets must discover a client-side proxy another way. By default, an applet will attempt to connect directly back to the server that it was loaded from. If this fails it attempts to read the configuration files of the browser to try to find a proxy setting, and attempt a connection to the proxy. If this also fails it prompts the user for a proxy setting. However, if a given site is connecting through a proxy server, the administrator can configure a special session definition for that site that includes the proxy server address. This saves the applet from having to go through the proxy discovery process, and prevent the users from ever having to see a proxy prompt.

WebConnect provides several variations on the concept of using dynamically generated tokens to authenticate a connection. Of course, the initial connection must be authenticated by some other means; the token only authenticates subsequent connections. This concept is particularly useful when a third-party web server is used for the initial connection or an external authentication server is desired. An OpenConnect-provided CGI or plugin, which is installed on the web server, connects to the admin port of the WebConnect server to fetch the token. The browser must then present this token to gain access to the user port. In all variations of token authentication, it is important that the admin port be protected from general user access to prevent unauthorized dispensing of tokens. The best way to protect the admin port is to set up WebConnect on a dedicated machine without general user accounts and configure the admin port for local host access only. If a third party web server is to be used in conjunction with WebConnect it can be setup on the same machine to give it access to the admin port. If performance or the use of non-standard ports is an issue, the servers can be setup on separate machines connected by a private LAN. At a very minimum, the admin port should be blocked from direct access from the Internet by a firewall. Remote access to the SSL-protected admin port could still be safely allowed, but only if protected by client certificates.

An additional security choice within WebConnect allows the use of stand-alone token generating mechanisms (JAVA Bean or Windows COM object) which may be embedded in a secure environment such as a Web portal to generate tokens for WebConnect. In this scenario access to the admin port is unneeded for session generation and it can be completely blocked accepting whatever product administration needs exist.

### **Protecting Host Resources**

In WebConnect, access to host resources are through session definitions. For instance, host name, port, and LU are all components of a session definition. Access can be restricted to a given session by using one of the built-in user authentication features, in conjunction with the token authentication.

For example, WebConnect can authenticate the user and then present different choices depending on the user. Individual users or groups could be confined to use certain session definitions mapping to specific host resources.

SSL cipher suites are set by session, allowing the configuration of different cipher suites on different sessions. The SSL Required option requires the user to use SSL (connect through HTTPS) when accessing a session with this option set. If the SSL Required option is not set, SSL is optional for the session.

### **SSL Versus RC4**

Two types of encryption are available between the WebConnect Java client and the WebConnect server: SSL (by enabling the HTTPS (secure) port on the WebConnect server), and RC4.

SSL-enabled applets take longer to download. This is because SSL capabilities in today's browsers are not accessible to applets; therefore, SSL libraries must be included and downloaded with the applet. Once downloaded, a session using the SSL option sets up faster than a session using RC4 with Diffie-Hellman. For keys shorter than 1024 bits, SSL sessions also use significantly less server CPU during session startup.

SSL can have varying levels of performance depending on the cipher suite implemented. Cipher suites utilizing RC4 encryption and the MD5 hashing algorithm yield the highest performance.

The RC4 encryption option allows for a quicker applet download time since it uses a smaller applet; however, it takes longer to set up a session due to the Diffie-Hellman algorithm used for key generation.

Also, the server CPU load for session startup is generally higher. There is no specific performance data on RC4 versus SSL for large numbers of clients. It is a logical assumption, however, that RC4 runs faster than SSL since SSL adds padding and performs message authentication.

On the client side, SSL should not be perceptible. On the server, SSL does not degrade performance unless the server becomes CPU bound.

For additional technical information on SSL and RC4, see the following Web URL sites:

SSL— go to <http://www.netscape.com> and search for SSL.

RC4—go to <http://www.rsa.com/>

### **SSL in WebConnect**

WebConnect uses SSL to secure connections between a WebConnect client and the WebConnect server without requiring any special configuration on the client machine. This is achieved by leveraging the SSL provided in the browser. That is, security parameters are passed to the client over the secure browser-to-Web server connection. When the browser requests a WebConnect session from the server, applet parameters are delivered to the browser. These parameters include the cipher suite to be used for the session and a hash, or fingerprint, of the WebConnect server certificate. This fingerprint is later used to verify the certificate received from the server during SSL negotiations, thereby authenticating the WebConnect server.

WebConnect uses an alternate port for SSL connections so different security measures can be applied to the SSL and non-SSL ports. For instance, an installation of WebConnect could choose to hide the unsecured port behind the corporate firewall but expose the SSL port to Internet traffic.

A key pair must be generated for WebConnect. The private key is password protected and used only by WebConnect. An X.509 Certification Authority certifies the public key. The resulting certificate is used by clients to authenticate the server as part of the SSL protocol. Before enabling SSL in WebConnect, a private key and certificate for the server should have been installed earlier.

#### **Cipher Suites**

SSL defines a Handshake Protocol for negotiating a cipher suite and allowing the client and server to authenticate each other. The cipher suite specifies the algorithms for peer authentication, data encryption, and message authentication when normal session traffic begins. The algorithms defined by a cipher suite are independent of the SSL protocol.

WebConnect supports several popular encryption algorithms, such as DES, Triple DES, and RC4. The RSA public-key algorithm is used for both key exchange and peer authentication. The Secure Hash Algorithm (SHA-1) and MD5 are supported for message authentication.

A separate cipher suite can be selected for each configured session. Cipher suites are set from the security section during session configuration.

## **Key Pairs and X.509 Certificates**

SSL utilizes public-key cryptography for peer authentication and key exchange. WebConnect uses the RSA public-key algorithm for both of these functions. The server's public key is given to the client in a digital certificate (X.509 standard). The client generates a master secret to be used to derive a session key for data encryption. The client then encrypts the master secret with the server's public key and sends it back to the server. Now the server can decrypt the master secret and communicate with the client using the encryption algorithm specified in the negotiated cipher suite.

This all requires that a key pair be generated for the server. The private key must be kept secret, only to be used by the server. The public key is given to an X.509 Certification Authority (CA) for certification. The CA generates a certificate containing the server name and public key, the CA name, validity dates, and a serial number for the certificate. Finally, the CA signs the certificate with its own private key, so that its authenticity can be verified by anyone in possession of the CA's public key.

An SSL client authenticates an SSL server by verifying the signature in the server certificate with the public key of the CA specified in the certificate. For this to work, the client must have ready access to the CA's certificate. An SSL server can be configured to request a certificate from the client to the server, and also to authenticate the client. HTML extensions exist to trigger a browser to generate a key pair, request a certificate, and accept a certificate for installation. This allows a browser to operate with a Web-based CA. Netscape and Microsoft both support this type of browser configuration under user control

## **Third-Party or Private CA**

Whether you use a trusted third party for the CA or establish a private CA within a company or organization will depend on the targeted users of the system and the level of security required. If the users are typically anonymous or outside of administrative control, or if the security requirements are not stringent, you can use a certificate issued by a trusted third party. On the other hand, if the highest level of security possible is wanted, you should establish a private CA within the company or organization using a third-party certification tool such as iPlanet Certificate Server, Entrust Web CA, or XCERT Sentry CA. Then you can issue certificates for servers and clients and configure them to honor only specific certificates.

The WebConnect server must have a key pair and certificate before you can use the SSL feature. These are normally generated during the server installation process, but they can be generated later using the configuration utility. WebConnect can either generate its own certificate or generate only a PKCS #10 certificate request to be submitted to the CA. If the server issues its own certificate, it creates a temporary CA (by default) that is removed after the certificate is issued. By setting the environment variable `OCS_CA` before running SSL configuration, a permanent CA is created in the directory specified by `OCS_CA`. Setting the same `OCS_CA` during future upgrades would allow new versions of WebConnect to use the existing CA. If a CA is chosen to provide the certificate for the server, the certificate is manually installed.

The server certificate should be a base64-encoded, DER-formatted, X.509 certificate, stored in a file named `cert.txt` in the security subdirectory. This is usually a concatenation of the server's certificate, the issuer's certificate, plus any others in the hierarchy if the issuer is not the root CA. They should be ordered server certificate first, ROOT CA certificate last. Creating `cert.txt` might require cutting and pasting from two or more files.

## Supported Cipher Suites

The specifications for cipher suites, SSL protocol without encryption, and RC4 encryption are described in this section.

### Cipher Specifications

#### RSA\_WITH\_RC4\_128\_SHA

RSA algorithm for key exchange and peer authentication.

RC4 128-bit encryption.

SHA-1 (Secure Hash Algorithm) for message authentication.

#### RSA\_WITH\_RC4\_128\_MD5

RSA algorithm for key exchange and peer authentication.

RC4 128-bit encryption.

MD5 algorithm for message authentication.

#### RSA\_WITH\_3DES\_EDE\_CBC\_SHA

RSA algorithm for key exchange and peer authentication.

Triple DES encrypt-decrypt-encrypt (EDE) encryption, in cipher block chaining (CBC) mode.

SHA-1 (Secure Hash Algorithm) for message authentication.

#### RSA\_WITH\_DES\_CBC\_SHA

RSA algorithm for key exchange and peer authentication.

DES encryption in cipher block chaining (CBC) mode.

SHA-1 (Secure Hash Algorithm) for message authentication.

#### RSA\_EXPORT\_WITH\_RC4\_40\_MD5

RSA algorithm for key exchange and peer authentication.

RC4 40-bit encryption.

MD5 algorithm for message authentication.

#### RSA\_EXPORT\_WITH\_DES40\_CBC\_SHA

RSA algorithm for key exchange and peer authentication.

DES 40-bit encryption in cipher block chaining (CBC) mode.

SHA-1 (Secure Hash Algorithm) for message authentication.

### **SSL Protocol with No Encryption**

RSA\_WITH\_NULL\_MD5

RSA algorithm for peer authentication.

MD5 algorithm for message authentication.

### **RSA\_WITH\_NULL\_SHA**

RSA algorithm for peer authentication.

SHA-1 algorithm for message authentication.

### **RC4 Encryption Option**

The RC4 option uses the Diffie-Hellman algorithm for key generation at the time a connection is made. The RC4 encryption option and key length is set by session. Because RC4 is susceptible to man-in-the-middle attacks, this option is best for intranets.

The following limitations apply:

The Diffie-Hellman RC4 option cannot be used in conjunction with the SSL encryption option.

When RC4 is configured for a given session, it is always required for that session.

RC4 encryption begins after the session start command in the JCP protocol.

### **Client Authentication**

Most host systems or applications implement user ID/password-based authentication tied to an access control scheme which directs known users to a set of designated system resources. Such is the case with WebConnect which implements HTTP basic authentication as the default authentication mechanism when user port authentication is enabled.

When a user first attempts to connect to the WebConnect server, the browser (at the request of the server) prompts the user for an ID and password. The user is then authenticated against the server's encrypted password file. The authenticated ID is then used by the server to map designated resources and preferences, for each specific user. It can also be used to plug into WebConnect Single Sign-On to automate logon to host systems and applications.

Since a weak password can compromise the security of an installation, particularly if deployed over a public network such as the Internet, it is often desirable to have a stronger form of authentication than user ID and password. One such mechanism supported directly by WebConnect is the use of X.509 certificates, through SSL, for authenticating users. This can be used instead of, or in addition to, basic authentication.

If both forms of authentication are configured, the user logon process will be bypassed if the “common name” field in the certificate matches a configured user ID. Once the browser has been authenticated to the Web server component of WebConnect, a token authentication mechanism will be used to re-authenticate the WebConnect applet back to the server.

Client authentication using SSL/X.509 certificates is one of the most secure mechanisms available to guarantee that only valid users can access a given server. For the highest possible security with WebConnect, activate SSL in WebConnect and use client authentication accompanied by the token authentication feature.

In WebConnect, client authentication is activated by setting the SSL client certificates option in the security section of server configuration. This option can be set for the user port, the admin port, or both. It causes the WebConnect server to request a certificate from the connecting browser as part of the SSL protocol. If the browser cannot produce a valid certificate, either known by the server or signed by a Certification Authority (CA) known by the server, then the server denies access to the user by disconnecting the browser.

To take advantage of the client authentication feature, a CA should issue certificates for each user and browser. The CA can be created and operated by the organization using a third-party CA product such as XCert Sentry CA, Netscape Certificate Server, or Entrust Web CA; or use the services of a trusted third-party CA, such as Verisign. WebConnect supports X.509 V3 certificates using the RSA signing algorithm with MD5 or SHA-1. In general, certificate extensions are not supported.

To configure WebConnect to accept the browser certificates, install the CA certificates or the individual client certificates in the server’s client database. For the user port, this is a flat file named certsacc.txt for individual certificates, and a file named cacerts.txt for CA certificates. Both are installed in the security subdirectory of the WebConnect installation. To support client authentication over the admin port, install individual certificates into a file named admcerts.txt.

The certificates must be DER-formatted, base 64-encoded, and delimited by the following lines:

```
-----BEGIN CERTIFICATE-----  
-----END CERTIFICATE-----
```

If the certificate of the issuing CA is installed, any browser with a certificate issued by this CA will be connected. For this reason, an internal CA will typically give more security control.

If the acceptance of all certificates, except certain ones, issued by a configured CA is desired, create the following file:

Create a file named certsrej.txt that contains these certificates, and install it in the security subdirectory of the WebConnect installation directory. The format of this file is identical to the certsacc.txt file.

Alternatively, the Certificate Authority can publish the client certificates to an LDAP directory. WebConnect can be configured to validate received client certificates against the LDAP directory. This eliminates the need to manually maintain the client database file under WebConnect.

## **WebConnect Server Token Authentication**

### **WebConnect Token Generation**

When the token authentication feature is enabled on the server, it is enabled for all sessions on the server. In addition to the on/off switch, you can specify a time-to-live value in seconds. The default value is 90 seconds. The token authentication feature encrypts the session authentication parameters using triple-DES encryption. A token is generated each time a request for applet parameters is received over the admin port. This token is given a time stamp and marked with the name of the session. When the applet connects to request a session, it must present its token to the server.

If the server cannot validate a token, or if a session mismatch is detected, the client is disconnected and a descriptive error message is written to the system log, including the port and address of the offending client. For applets needing to establish additional sessions, for instance when you select New from the file menu, protocol exists between the client and the WebConnect server to allow an existing authenticated session to connect to a new session using the same, authenticated token.

#### Limitations

Note: This feature does not work with any other WebConnect implementation relying on static HTML pages for applet launching.

#### Dependencies

Note: For this feature to be effective, a secure authenticated link must be provided between the Web server and the client browser.

Note: SSL should be enabled between the applet and the WebConnect server to protect the token when a session is connecting.

Note: The admin port must be protected.

Note: A live connection must be made from the Web server to the WebConnect server to fetch the token. A cgi-bin and plugins are provided with the product for performing this function with a third-party Web server.

## **Standalone Token Generation**

Additionally, a standalone token generation scheme may be implemented using a supplied server-side Token Builder Java Bean or COM object included in the wchome/samples/token directory. WebConnect's Token Generator does not require any client-side software.

These utilities can be used to generate JCP tokens for WebConnect sessions and for Single Sign On.

## **CGI Authentication**

CGI authentication provides an added level of security for WebConnect servers. To prevent unauthorized access to the WebConnect server with the CGI protocol used by WebConnect plug-ins, a token authentication mechanism is used to ensure that only authorized plug-ins can gain access to the server.

CGI authentication is enabled/disabled on the server from the Server>Security administration page. When enabled, any access requesting CGI data must first provide a valid token. If a token is not provided or is invalid, the connection is immediately terminated. Each plug-in can be configured to generate a token that is valid only by knowing the server's Cookie Domain Name and CGI Authentication identified on the Server>Security admin page

## **TN Server Encryption and Authentication**

WebConnect includes host (TN server) encryption and authentication from the server. In order to use this feature, the host must support SSL. TN server encryption and authentication are both disabled by default; TN server encryption may be used with or without host authentication.

If the host authentication feature is used, a certificate from each host that has been enabled for TN server authentication is required. Copy the certificates into a file named tncerts.txt and place the file in the security subdirectory.

Finally, to enable SSL, select a cipher suite to be used by each defined session.



## **WebConnect Security Solution Overview**

WebConnect provides a secure software connectivity link that enables browser-based access to information residing on mainframe and other host computer systems. In addition to the numerous benefits provided by a secure, SSL encrypted web-based emulation solution, WebConnect's Single Sign-On feature provides additional security by facilitating the automation of host application sign-on. Users are authenticated once by dedicated security and policy managers such as RSA SecurID, and allowed access to host applications with user ids and passwords that are maintained within a secure data source. This automation eliminates security issues arising from desktop stored login information as well as monitor-attached post-it notes with logon information. It can be used to grant mainframe access to authenticated end users without ever revealing any direct mainframe or host access user id and passwords.

### **Main Components:**

- > SSL encryption for both applet and TN connections.
- > X.509 client and server certificate authentication via Secure Sockets Layer
- > Encrypted/authenticated Token integration with industry leading security, access and policy managers such as RSA ClearTrust, RSA SecurID, Netegrity SiteMinder, etc.
- > Maintains host based security mechanisms such as RACF, Top Secret, ACF2
- > Centralized management of authentication information through LDAP or via XML SOAP messaging into Web Services and/or custom repositories
- > Integration within common security network models via OpenConnect's JCP Proxy server that validates protocol integrity of the patented persistent Applet JCP connection

## **Security Questions**

### **Where can I get more information on SSL and RC4?**

Both Netscape and RSA have Web sites where more information is available. For more information on SSL, go to [www.netscape.com](http://www.netscape.com) and search for SSL.

For more information on RC4 from RSA Data Security Inc., go to [www.rsa.com](http://www.rsa.com).

What if I want to get my own certificate?

An alternative is to use a CA product. If you do not use the web server built into WebConnect, we recommend that the third-party CA or CA product be used to generate the certificate for the secure Web server only, and that WebConnect generate its own certificate. It is less trouble, and no less secure. WebConnect always generates its own keys and certificate request, so there is no increase in security by having a third party certify the request.

### **Conclusion**

Consider three levels of protection: at the internet networking level, via public-key infrastructure or encryption; the system software level; and the data level, usually handled by the specific application associated with the data. Whatever you do, you must address all three levels to be really secure.

With WebConnect you can:

:: Drive additional revenue through extended use of mainframe information and applications

:: Efficiently and securely push vital mainframe information and applications out to customers, partners and employees

:: Integrate mainframe information with other line of business applications for a more holistic view of the business

:: Deliver a cost-effective secure solution through concurrent user licensing

OpenConnect Systems, Inc.



2711 LBJ Freeway, Suite 700  
Dallas, TX 75234  
Phone: 972.484.5200 Fax: 972.484.6100 Web: [www.oc.com](http://www.oc.com)  
Email :: [info@oc.com](mailto:info@oc.com)

© 2006 OpenConnect Systems Incorporated. OpenConnect Systems is a registered service mark of OpenConnect Systems Incorporated. OpenConnect and all other OpenConnect product names are trademarks of OpenConnect Systems Incorporated. All other products or services mentioned herein are trademarks of their respective companies. The technology of WebConnect is covered under U.S. Patent number 5,754,830.