Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Infrastructure Design Considerations When Using Client Certificates

This paper will investigate some of the considerations that should be evaluated when looking to bring a new technology into the design of an application. The security technology that will be used as an example is client-based certificates. It is easy to see that there are increasing requirements for web-based applications to use the Internet for conducting private business. This will sometimes require two-way authentication between the client and the server in addition to the more frequently addressed issues of integri...

Copyright SANS Institute

Author Retains Full Rights

AD

Infrastructure Design Considerations When Using Client Certificates

Tim Hollingshead
Practical Version 1.3

Introduction

Network and computer security is playing an increasingly more important role in today's software deployments. One of the reasons for this lies in the fact that software applications that have traditionally been deployed only on internal networks are now being made available across the Internet. Because of this, application and infrastructure design engineers must build security into every component of the framework while still maintaining the traditional design requirements of the system. In addition, they must continually re-evaluate their approach to security so that the latest technologies can be applied to protect the assets. Security must be planned from the beginning of the design process and integrated into the complete architecture or security itself may become the weakest link of the blueprint.

This paper will investigate some of the considerations that should be evaluated when looking to bring a new technology into the design of an application. The security technology that will be used as an example is client-based certificates. It is easy to see that there are increasing requirements for web-based applications to use the Internet for conducting private business. This will sometimes require two-way authentication between the client and the server in addition to the more frequently addressed issues of integrity and privacy that certificate use has provided. As with any design, there are several ways to accomplish a given task, with each one providing unique advantages and disadvantages that must be weighed against the criteria of the implementation goals. These points will be discussed and summarized to assist the reader in understanding the trade-offs associated with each approach. Security has become far too broad a subject to cover all aspects that should be raised within a single document. The considerations brought forth here will be from the viewpoint of the infrastructure designer and not that of the internals of the application. The paper will be general in nature and will try to emphasize some of the challenges that the security and infrastructure architect may expect to encounter.

Infrastructure Design Goals

The infrastructure can be thought of as the framework environment that all computer applications must have in order to run. This includes physical devices such as the computer hardware; operating system, network components and wiring that are all critical for the programs to run in and on. The design of computer applications has always required the architect to consider the relationship between the infrastructure and the application. Further, the inter-relationship between the individual infrastructure elements is crucial to insure an

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

environment that will allow for the best performance, availability and reliability of the software programs. The importance of understanding this inter-relationship has increased over time to the point that today's applications cannot be successful if all points are not taken into consideration.

The technical design goals of the infrastructure use characteristics that are familiar to everyone and are summarized as follows: Readiness, Redundancy, Scalability, Reliability, Performance, Security and Management. Each of these characteristics should be considered, not only as they relate to each individual system component, but in the overall effect to the computer-based solution.

Many designers understand how to construct an infrastructure that meets most of the above characteristics, but many still view security as an "add-in" and don't give proper consideration as to how it integrates with the other traits. If this approach is taken, the infrastructure may be weak and not scale over time to continually meet the goals of the design. Security must be viewed as the sum of many parts, all inter-related and functioning only when all of the single pieces compliment each other. This includes all components of the infrastructure, the application internals and the site policies. If these are constructed properly from the beginning they will provide a framework that will endure.

When a new technology such as client-based certificates is used, the effect of the inter-relating parts must be re-evaluated and understood from the global perspective. It may mean a re-write of part of the software, or an upgrade to the network infrastructure to leverage the capability of the new technology, but it will be worth the time and effort if the overall effect is positive.

This paper will use the example of introducing client-based certificates, or "two-way" certification to investigate infrastructure design issues. Before looking into the use of client certificates and their integration with other components however, a brief explanation of their structure is provided.

Secure Communications

The need for secure communications was understood as soon as connections between computers were first established. This was easy to control in the beginning as each system was usually under the control of a single organization and was accessed only by employees, partners or clients. The Internet has increased the need for secure communications as a large public client-base is now accessing web sites all across the world. The form of communication most often used for privacy on the Internet is the Secure Socket Layer (SSL) protocol. SSL encrypts the data and provides a mechanism that uniquely identifies a web-server but does not usually address client authentication. This is because it is desirable to leave the service open to the general public, as they are the target audience.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

As the Internet has matured companies are beginning to leverage the Internet more for internal purposes, offering either Virtual Private Network services so that employees can access the internal network or are placing applications on a web site so that employees can use them while at home or traveling. Using the Internet for this purpose can reduce, or sometimes even eliminate, costly internally managed wide-area networks. This offers a great financial benefit, but some form of user authentication must be implemented so that the general public cannot access the service. This is often done using a password, or token-based appliance that can uniquely identify the employee.

An alternative method that can be used to identify the user is client-based certificates. These have been around as long as the Public Key Infrastructure (PKI) architecture has been in existence but is used less often than the server certificates. This technology offers a good solution for some environments and should be considered when web-enabling internal applications. Before presenting a detailed explanation of the client-based certificate, a summary of PKI is in order.

PKI is the architectural outline that defines the use of the hardware, software and policies used in authenticating the identity of a subject. There are four components that comprise the makeup of PKI: (1) the certificates that represent the authentication token, (2) the Certificate Authority (CA) that holds the decision on subject authentication, (3) the Registration Authority (RA) that accepts and processes certificate signing requests and (4) Lightweight Directory Access Protocol (LDAP) directories that hold publicly available certificate information. [1]

The certificates used in PKI are based upon the X.509 (v3) standard (described in RFC 2459) and represent data structures that bind public key values to subjects. This standard describes the method by which a public and private (asymmetric) key pair can be used to encrypt data in a secure manner. Using this technology, the users of the public key can be sure that a particular subject owns the corresponding private key. [2]

The Certificate Authority (CA) is an entity that is trusted by both parties that desire to exchange data via PKI and can vouch for the authenticity of one or both parties. The CA itself makes use of a certificate and either guarantees its own identity or relies on an additional CA to verify its identity. The CA systems form a hierarchy that is based on an eventual root server that vouches for it.

[1] Hontannon, [2] Housley

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

The Registration Authority server provides a system where users can apply for a certificate. The request can be automatically generated based on some pre-existing policy or can be reviewed by an administrator before being manually granted. The LDAP directories are databases of certificates that can be accessed via the Lightweight Directory Access Protocol. This cache of public keys is available to all clients so that they can be used to verify the authenticity of a subject.

The protocol associated with PKI is SSL, which is generic in its construction and can use several different mechanisms and algorithms to provide the key exchange and data encryption. SSL is not dependant on TCP/IP but is often layered in between the TCP/IP stack and the service that is using it, such as Hyper Text Transfer Protocol (HTTP). This combination renders the secure implementation of HTTPS and is commonly used to access web servers in a secure manner. The SSL protocol is divided into two separate layers. First, the Record Protocol encapsulates higher-level protocols, and second, the Hand-shake Protocol negotiates the characteristics of the communication session, authenticates the parties, and exchanges keys. Once the session has been established, the higher-layer protocol can send data securely over the channel. [3] An encryption algorithm is used to encrypt the data as it is transmitted and received by both the client and server. To decrypt this data, both the client and server use a "key", which allows them to quickly decode the message. Any other parties that might be listening in on the session can see the encrypted data, but cannot decipher it without the key.

There are two styles of keys that are used to encrypt the data during transmission, the symmetric and the asymmetric keys. When using symmetric keys the client and server have identical keys, both of which must be kept secret at all times. If anyone else were to acquire the key, the data could be understood and the session would no longer be secure. Symmetric keys, or "private-private" keys are very efficient and can be very useful in a small implementation, but have not been found to scale well. Asymmetric keys, or "public-private" keys, use different keys for the client and server of an encrypted session. With this style of keys, only one of the keys must be kept hidden and the other can be seen by anyone without fear of the data being compromised. In fact, the public key can be "advertised" to assist with identification in a widely used public network. When encrypted with a public key, the data can be sufficiently difficult to decrypt, yet very easy to decrypt if the user has the corresponding private key. The PKI architecture actually uses a combination of these two key styles to provide a good blend of security and efficiency. The asymmetric keys are used to start the session, but symmetric keys are then dynamically generated for the bulk of the transfer. The following paragraphs describe the handshake process in further detail.

[3] Freier

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

When a client accesses a web server using the SSL protocol, the characteristics of the session must be negotiated using the handshake portion of the protocol. This will permit the client and server to authenticate each other, determine if both the client and the server will exchange certificates and determine the specific algorithms and protocols used in the encryption process.

The client first sends the server its own preferences for cipher settings, the SSL version number and a randomly generated data value that will be used later in the handshake. The server responds by sending the same type of data along with its own certificate. The information on the server certificate will include a version number, the issuing authority, a signature, a valid date range and the public key of the server. If the server is configured to ask the client for a certificate it will be requested at this point.

Upon receiving the certificate data from the server, the client will verify that the date on the certificate is valid, that the Certificate Authority is trusted, that the public key of the CA validates the digital signature of the server and that the domain name on the certificate matches the domain name of the server. The client then generates a "pre-master" secret and sends it to the server encrypted with the server's public key. If the server requested client credentials, they are sent as well. Some of the data sent is encrypted with the client's private key, so that the server can use the corresponding public key to authenticate the client.

The server will verify the authenticity of the client and use its private key to decrypt the pre-master key sent by the client. The server will then generate a "master secret" which will be used by both the client and the server to generate "session keys". The session keys are symmetric keys that will be used only for the duration of the session. Remember, the symmetric keys are desirable because they are more efficient to use in the encryption/decryption process than the asymmetric keys.

This simplified description of the handshake process summarizes the steps that every SSL session must complete before further communications are established. This is probably done millions of times per day on the Internet, but is transparent to the user, unless the client finds a discrepancy in the credentials of the server. The optional step of the client supplying its own credentials can be just as transparent, or it can be configured to prompt the user for a "pass-phrase" as a precautionary measure before use. This feature provides a higher level of security so that a client-based certificate cannot be used by just anyone that is in control of the computer.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Design Considerations

An important point to understand is that if the designer does not know how to integrate security into the design of the application and infrastructure it will probably end up being the component that suffers the most. There will typically be more pressure from management and the users toward performance, scalability and redundancy than security. It is often viewed as an inconvenience anyway!

Once the designer has decided to use a particular technology such as client certificates, there are a number of questions that should be investigated to help answer the above concern. A few of them are:

> Where can certificate authentication be placed in the infrastructure and what are the trade-offs associated with this placement?"
>
> How can the design of the CA integrate with the infrastructure design goals of the overall application?
>
> How will security be affected with each potential configuration?
>
> How will the long-term goals of scalability, reliability, management, readiness, redundancy and performance be affected by the security structure?

Part of the answer to the first question listed above may be found in determining where the overhead associated with SSL calculations can be incurred without drastically affecting the other design characteristics and which solution is most cost-effective as the infrastructure grows. Another consideration may be in the ease of integration between the application and the CA and/or the authentication server. To assist in the investigation of these questions, several example configurations will be considered with discussion points highlighting the advantages and disadvantages of each.

Example 1 - PKI Integrated With Application

The first example to consider is a system that uses the application platform itself as the location for handling certificate responsibility. This type of configuration would generally require a more robust server as it would be responsible for all PKI services, SSL encryption/decryption and web services besides running the application and any associated database access functions.

Some application providers like to take the approach of making their software responsible for as many of the functions as possible. This has historically not been true however as it relates to PKI functions. This is beginning to change as can be seen in the Java™ specification. Here the application, or specification in

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

this case, is an example of how more of the security functions related to PKI are being handled by the product itself. Consider the following abbreviated list of services that are included in the J2EE (Java 2 Enterprise Edition) Version1.4 specification.

- An implementation of the Digital Signature Algorithm (DSA), described in NIST FIPS 186.
- An implementation of the MD5 (RFC 1321) and SHA-1 (NIST FIPS 180-1) message digest algorithms.
- A DSA key pair generator for generating a pair of public and private keys suitable for the DSA algorithm.
- A DSA algorithm parameter generator.
- A DSA algorithm parameter manager.
- A DSA key factory providing bi-directional conversions between (opaque) DSA private and public key objects and their underlying key material.
- An implementation of the proprietary "SHA1PRNG" pseudo-random number generation algorithm, following the recommendations in the IEEE P1363 standard (Appendix G.7).
- A certificate path builder and validator for PKIX, as defined in the Internet X.509 Public Key Infrastructure Certificate and CRL Profile.
- A certificate store implementation for retrieving certificates and CRLs from Collection and LDAP directories, using the PKIX LDAP V2 Schema (RFC 2587).
- A certificate factory for X.509 certificates and Certificate Revocation Lists (CRLs).
- A keystore implementation for the proprietary keystore type named JKS. [4]

Taking this approach and building as many of the required functions as possible into a single platform may be desirable, especially from the viewpoint of the manufacturer. Caution should be used however on the part of a company implementing the product to avoid the situation of “putting all of your eggs into one basket”. Several questions must be asked regarding this type of design before deciding on this structure. The first is “does the design benefit the overall security of the application and/or site?” In answering this question, it is assumed that a PKI infrastructure is required for one or more purposes in the application itself. Many follow up questions must be addressed concerning all aspects of infrastructure design.

[4] J2EE Specification

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

A “self-contained” PKI implementation is illustrated in Figure 1 and may make sense from a security standpoint if the user base is separate from any other function in the organization. In short, it becomes a standalone system from an authentication viewpoint. For example, the application is for client use only (not employee use) and does not need to integrate with any internal network authentication method, domains or tools. On the upside, it may offer a less complex design, especially if the manufacturer is involved in the support of the system, allowing the administrative staff to view the PKI function as “turn-key” and internal to the application. In addition, if there is no separate Certificate Authority system and hierarchy to support, it may be less expensive to implement and maintain.

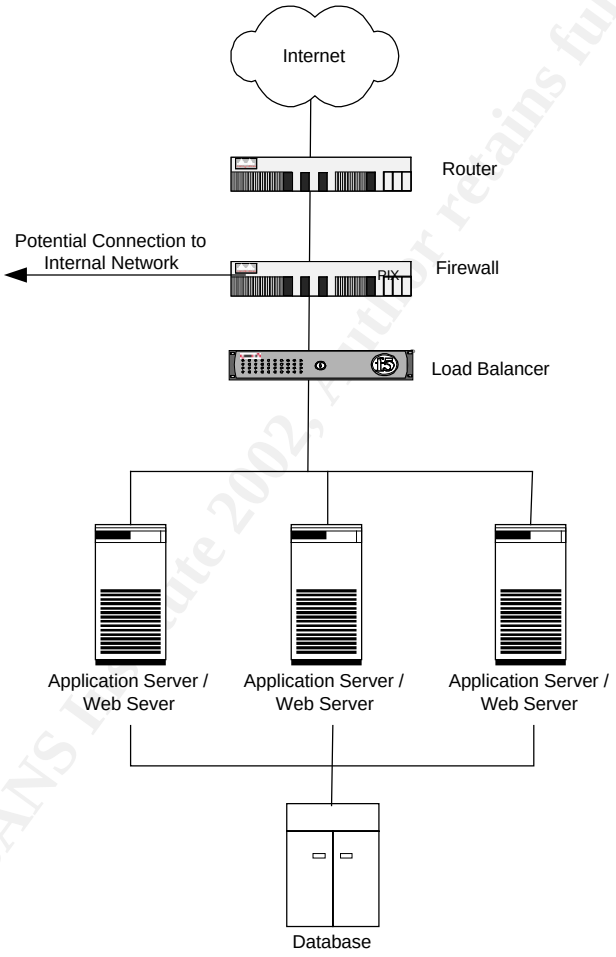

Figure 1

A concern that will be addressed in more detail later in this paper is how to pass the certificate information to the application for authentication and authorization. This particular configuration will make it easier to pass the user credentials to the application since it is all being done on the same system. This is not to be overlooked, as this can be a significant undertaking for a system that uses some form of external authentication.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Some thought will have to be given to how the system will scale and if the PKI architecture supports that scaling. Most software vendors today offer some form of application level clustering that at a minimum, is (1) aware of what systems are available to participate and (2) share some form of session data replication for the users. This will allow nodes to be added as the demand on the cluster increases, but careful consideration and testing should be done to insure that security is maintained. For example, if a user loses their connection due to a system crash, will the user be required to repeat the authentication process or will this data be preserved and prove that the new connection is indeed the same user as before?

Building a server that has many of the required functions built into a single tier may not be advantageous from the perspective of the site. The hardware required to support the operating environment might have to be significantly larger and more powerful, thus more expensive. It also ties the purchaser to a single vendor and/or operating system that may produce undesirable results in the long term. There will also have to be some research and assurance regarding the performance capabilities of a system that performs so many functions.

If some of these questions can be answered and are found acceptable, the security benefit of keeping authentication on a single system has its advantages. As in any networked system, the operating system and application will have to be kept up to date and hardened to the full extent possible and all systems will need to be protected by a firewall. In addition, some applications are slow to support larger key lengths and/or new encryption algorithms. This is often done to make the product exportable, but some vendors do offer a domestic version that would offer higher security levels.

An alternative configuration could be to have local users in the application, but use an external certificate authority hierarchy that can be queried to verify certificates. This style of configuration will be investigated in the next example but will use a web server as the point of origin.

Example 2 - PKI Integrated With Web Server

A second approach that can be used for the PKI solution is implementation at the web server tier and probably represents the most common point at which web traffic and certificate based authentication are combined. Most major web server products already have the capability to use the SSL protocol and support the optional client authentication portion of the handshake. To enable the use of client based certificates, a property or rule is set on the web site to prompt the user for a certificate and then the credentials are automatically checked to be valid before the session can be initiated. Refer to Figure 2 for an illustration of this configuration.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

This type of configuration lends itself well to most of the design goals, with an emphasis on redundancy and scalability. Load balancers can distribute connection requests across web servers. If the load on the web servers gets too high, a new web server can be added and inserted into the load distribution pool. The same scaling capability holds true for the application servers as well, depending upon how the application and web server are implemented. Optionally, a second layer of load balancers could be inserted in between the web server and the application servers (again depending upon structure). This is a very common style of infrastructure design that can be found at many web sites.

For this example the design will use a certification authority external to the web and application server tiers. As connections are made to the web servers, the web site will prompt the client for a certificate. A locally managed certificate authority shown in this diagram on the internal network would probably have provided the client certificates. The web servers themselves may be issued a certificate from these same authorities, or may choose to use an external, Internet based certificate authority.

Some sites may consider using the web server as the Certificate Authority similar to the configuration used in Example 1 where most of the users are local to the application. As a general rule, the CA would be implemented externally on a separate hierarchy of servers. These CA servers could be on the same external network (in the DMZ) or they could be on the internal network. Assuming that most of the requests to the CA server are for local users (employees, clients and partners), it may make sense to use the existing internal systems as the CA. Consider the following diagram.

© SANS Institute 2002, Author retains full rights.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

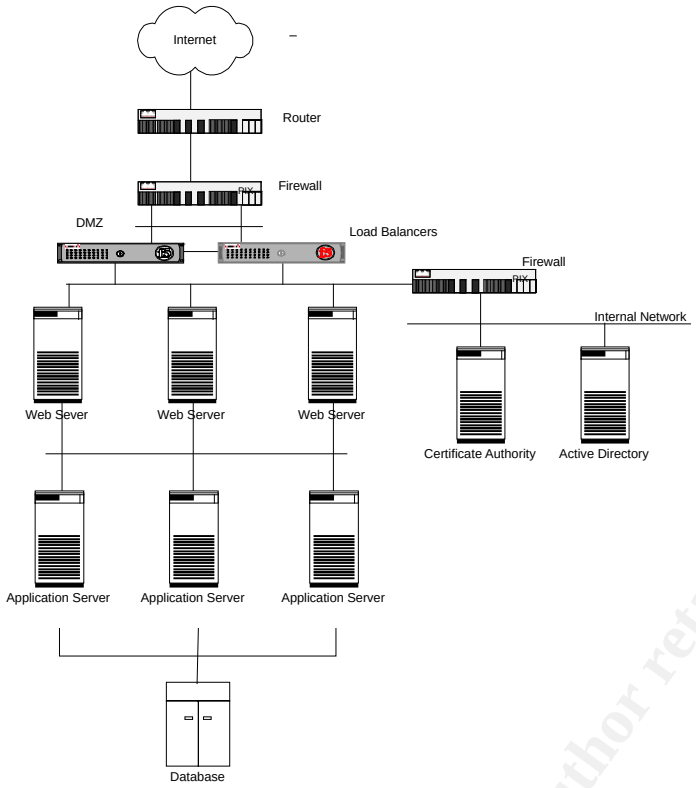

Figure 2

The Certificate Authority system (representing a hierarchy of CA systems) shown in the above diagram would most likely be associated with the system that authenticates internal users, and can be accessed by the Web Server during the certificate verification process. For example, many sites today are beginning to use Microsoft Windows 2000™ Active Directory as the product for providing network based services. While there are several options for designing the internal Certificate Authority, it is possible to map the certificates to individual Active Directory user accounts. Consideration should be given to the benefits of mapping internal users to their certificate as it offers a common point in the management of the two authentication tools.

Assuming the web servers in the diagram above were to use Microsoft Internet Information Services (IIS), the web page could be configured to prompt for a user certificate, and then authenticate the user against the Active Directory. When the certificate was supplied, IIS would verify the validity of the certificate through normal means, then take the “principal” credential from the certificate and supply it to the Active Directory as a part of the authentication process. The Active Directory system can use the principal to uniquely identify the user and map it to a user in the Active Directory. This capability is not limited to IIS but is very easy to construct if using a single vendor.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

The advantages of this type of configuration would be most evident when internal users will be accessing the web site. Remember that there is no actual password used in accessing the web page, only the certificate, but there can be a direct tie between the certificate and the user. It is always a good idea to make the user enter a pass phrase when their local certificate in invoked, but this is in no way tied to their domain password.

One specific use for tying the user certificate to the internal domain is for initial certificate requests. Continuing with the Microsoft product set, a web page can be made available to a client that provides a form to fill out and request a certificate. Depending upon the site policy, the user could be automatically provided a certificate immediately if proper authentication is made against the internal domain. If this is not desirable, the policy can be set to record the request and wait for administrator review and approve before the certificate is generated.

Security in this configuration is still good as long as the internal domain is properly protected at the firewall and if the internal policies are correctly set. For example, if the password policies are too lenient a certificate could be falsely issued to a hacker that guesses a password. As always, security is the sum of many parts, and must be provided in depth. The configuration can be made secure as long as the basic rules of network security are provided.

Example 3 - PKI Integrated With External Appliance

A third approach for PKI implementation is to keep it separated from the web site and application servers and would be constructed just as in Figure 2. The only difference is that the load-balancers (appliance) shown at the top of the picture would terminate SSL sessions thus offloading this responsibility from the web servers.

The trend has been growing over the last several years to keep SSL overhead away from the web and/or application servers. This is usually done with an appliance device that is installed between the firewall for the site and the web server combining the encryption/decryption duties with load-balancing functions and even some security responsibilities. An appliance device of this nature offers many benefits to the infrastructure design and should always be used if the budget and other design goals allow.

The appliance would not perform the actual certificate authority responsibilities, but it would handle all calculations associated with the session, and make requests to the CA if necessary. Refer back to the section of this paper that described the SSL handshake. All of these steps can be accomplished within the appliance itself, and assuming the device is robust enough, can handle all of the

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

key negotiation, and encryption/decryption overhead for potentially hundreds of simultaneous sessions.

This configuration adds maximum flexibility to the design of the infrastructure allowing for superior load balancing features and transparent scaling of systems and appliances. One negative associated with the use of an off-loading appliance is in how user credentials from the certificate could be passed from the device to the web server and/or application. This is a new area that vendors are just beginning to investigate and support. The method may vary from the appliance inserting client certificate fields into an HTTP header request to providing an API that can probe the appliance for the desired certificate data. This is an area that must be investigated carefully to insure that two-way certificate services can be supported in the manner that the site and application require.

The advantage of this type of configuration is in removing the SSL calculations away from the web tier, providing better scalability and performance. While the appliance itself will add cost it will allow for a smaller web server, or lower the number of web servers required. If the appliance is load-balancer and has complex load balancing algorithms, it can greatly benefit the performance and reliability of the site by keeping track of which systems are available and which are most/least heavily loaded. This concept can even become global in scope with site availability decisions affecting how client requests are routed. It is very desirable to use this type of configuration, but the security requirements of the application must be able to leverage the design. In this case, insure that the device can pass all clients certificate credentials to the application server.

Conclusion

There are many aspects to consider when planning computer security. It is a concept that is illusive, ever changing and can never be fully attained. An acceptable level of security can be provided however, if proper planning, attention to detail and thoroughness is pursued. This document has tried to bring forth a single point, that security must be architected and built into the design of any implementation from the very beginning of a project or it may never have a change of succeeding.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

References

[1] Hontannon, Ramon J. "Keeping PKI Under Lock and Key"
October, 2000
Network Magazine
http://www.networkmagazine.com/article/printableArticle?doc_id=NMG20001004S0015

[2] Housley, R. , Ford, W., Polk, W., Solo, D.
"Internet X.509 Public Key Infrastructure (RFC 2459)"
January, 1999
ftp://ftp.isi.edu/in-notes/rfc2459.txt

[3] Freier, Alan O., Karlton, Philip, Kocher, Paul C.,
"Secure Socket Layer V3 Internet-Draft"
November, 1996

[4] "Java 2 Enterprise Edition Specification, Version 1.4"
"Java Cryptography Architecture API Specification and Reference Manual"
http://java.sun.com/j2se/1.4/docs/guide/security/CryptoSpec.html

Smith, Will. "Managing Infrastructure Deployment Projects"
July, 1997
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/winntas/deploy/projplan/net304.asp

Kalra, Vimal. "E-Commerce Technical Readiness – System Redundancy"
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/ecommerce/plan/sysredun.asp

Microsoft Corporation. "Step–By-Step Guide To Mapping Certificates To User Accounts"
February, 2000
http://www.microsoft.com/windows2000/techinfo/planning/security/mappingcerts.asp

CCTA - the Government Centre for Information Systems. (Corporate Author)
"An Introduction To IT Infrastructure Planning "
February, 1995

Java is a registered trademark of Sun Microsystems.
Microsoft, Microsoft Windows 2000, IIS and Active Directory are registered trademarks of Microsoft Corporation.
PIX is a registered trademark of Cisco Systems Inc.
F5 is a registered trademark of F5 Networks Inc.

© SANS Institute 2002, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 19th, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	**Cyberjaya, MY**	**Jun 03, 2013 - Jun 08, 2013**	**Live Event**
SANS Pen Test Berlin 2013	**Berlin, DE**	**Jun 03, 2013 - Jun 08, 2013**	**Live Event**
Industial Control Systems Security Training - Houston	**Houston, TXUS**	**Jun 10, 2013 - Jun 15, 2013**	**Live Event**
Security Impact of IPv6 Summit 2013	**Washington, DCUS**	**Jun 14, 2013 - Jun 16, 2013**	**Live Event**
SANSFIRE 2013	**Washington, DCUS**	**Jun 14, 2013 - Jun 22, 2013**	**Live Event**
SANS Canberra 2013	**Canberra, AU**	**Jul 01, 2013 - Jul 13, 2013**	**Live Event**
SANS London Summer 2013	**London, GB**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
Digital Forensics & Incident Response Summit 2013	**Austin, TXUS**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS Rocky Mountain 2013	**Denver, COUS**	**Jul 14, 2013 - Jul 20, 2013**	**Live Event**
SANS Mumbai 2013	**Mumbai, IN**	**Jul 22, 2013 - Jul 27, 2013**	**Live Event**
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	**Washington, DCUS**	**Jul 22, 2013 - Jul 26, 2013**	**Live Event**
SANS San Francisco 2013	**San Francisco, CAUS**	**Jul 29, 2013 - Aug 03, 2013**	**Live Event**
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	**Bangalore, IN**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
SANS Boston 2013	**Boston, MAUS**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
Critical Security Controls Summit	**Washington, DCUS**	**Aug 12, 2013 - Aug 18, 2013**	**Live Event**
Industrial Control Systems Security Training - DC	**Washington, DCUS**	**Aug 12, 2013 - Aug 16, 2013**	**Live Event**
Mobile Device Security Summit 2013	**OnlineCAUS**	**May 30, 2013 - Jun 06, 2013**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**