



## **Controlling & Configuring Web Access with Symphony Identity Management Solutions**

### **Executive Summary**

In virtually every industry, companies are faced with the challenge of making Web resources available to external users in a secure fashion. External users may include a company's own employees working from offsite locations, employees of partner or customer organizations, and individual end-user customers. This task is an even greater challenge for businesses with a large number of applications that reside on many different platforms and domains — particularly when the applications follow autonomous security models.

Symphony Identity Management Solutions (IMS) are a set of comprehensive, standards-based Web access and management solutions that help companies solve identity and access management challenges.

At the core of Symphony IMS, is the Symphony Web Access Manager (WAM). Symphony Web Access Manager solves the complex needs of companies, their partners and end-users on the web to provide a seamless experience.

- Manage centrally disparate internal and external users, as well as their individual access privileges to resources that reside across heterogeneous IT environments
- Provide users with distinct levels of access within a specific company resource
- Shorten development cycles that were expanded to accommodate implementation of security on an application-by-application basis
- Comply with regulatory requirements for protecting privacy and ensuring the security of company information
- Reduce or eliminate complexities introduced by solutions that implement proprietary and redundant components

This white paper provides technical detail on the Symphony Web Access Manager architecture, with particular focus on Symphony Enforcement Agent, the component of the solution that controls access to Web resources.

### **The Symphony Identity and Access Management Approach**

Symphony has taken a unique approach to solving the problem of managing user identities and their access permissions. The Symphony solution neither conflicts with, nor duplicates, existing identity infrastructure. The result is a solution that is less complex, easier to install, and more cost-effective to maintain than alternate technologies.

This unique use of directory technology, and the directory-centric design of Symphony Identity Management Solutions, enables all solution components to directly access the directory for enforcement decisions (authentication and authorization) and all management decisions. By design, Symphony Identity Management Solutions leverages and relies on directory services for the same reasons that enterprises around the world have chosen to deploy them: standards-

based architectures, search performance capabilities, scalability, reliability, redundancy, and failover.

Symphony Identity Management Solutions does not require a middle tier of additional servers. This approach sets Symphony Identity Management Solutions apart from competitive products, which must first pull the directory information into their own servers to make enforcement decisions against their proprietary servers and technology. Their method depends on this proprietary middle tier for performance capabilities, scalability, reliability, redundancy, and failover. By eliminating the middle tier, Symphony Identity Management Solutions reduces the need for extra hardware and services, as well as the need for separate, complex local configuration and parameter files (e.g., .ini files) that introduce additional support burdens and security risks.

The Symphony solution gives you the ability to simplify identity and access management, while enabling you to take advantage of existing investments in infrastructure components, including your current directory assets.

### **Symphony Enforcement Agent Deployment Flexibility**

Symphony Enforcement Agents are lightweight software plug-ins that reside on either a proxy server or directly on a Web server hosting the protected application(s). Symphony Enforcement Agents intercept and validate every HTTP and HTTP over SSL request for a Web-enabled application. This configuration provides the most effective and cost-efficient way to centralize security across technologies, domains, and large, disparate user groups.

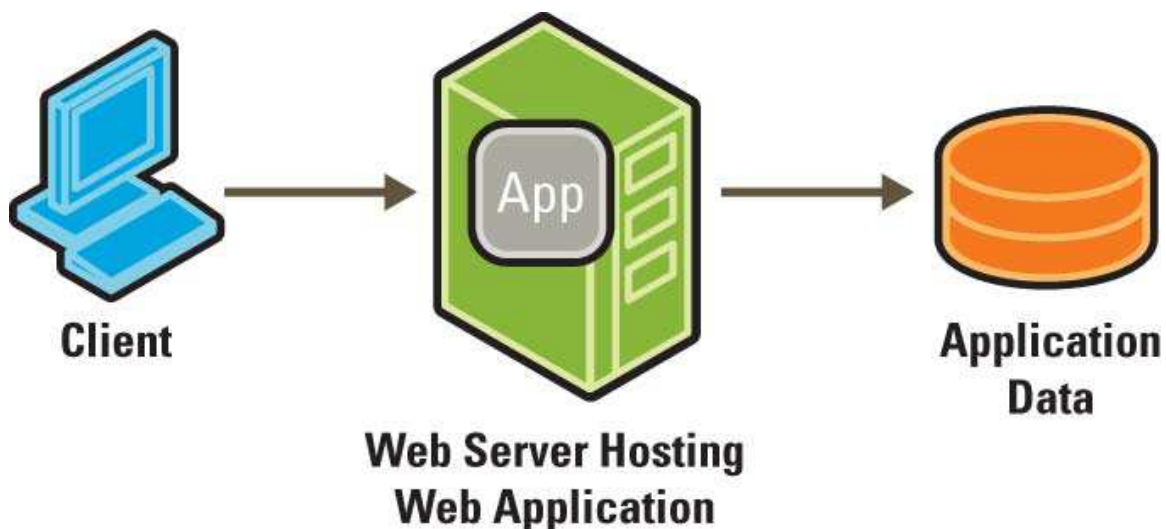


Figure 1. Common intranet architecture without identity and access management

Figure 1 shows a common architecture for internal Web applications that are not protected by an identity and access management solution.

As an example, the client machine makes HTTP requests to the Web server. The HTTP requests resolve to a Web application, and the Web application reads and writes its data to a backend data store. When a company needs to repurpose this Web application to enable access by external users, the architecture may initially resemble the one shown in Figure 2.

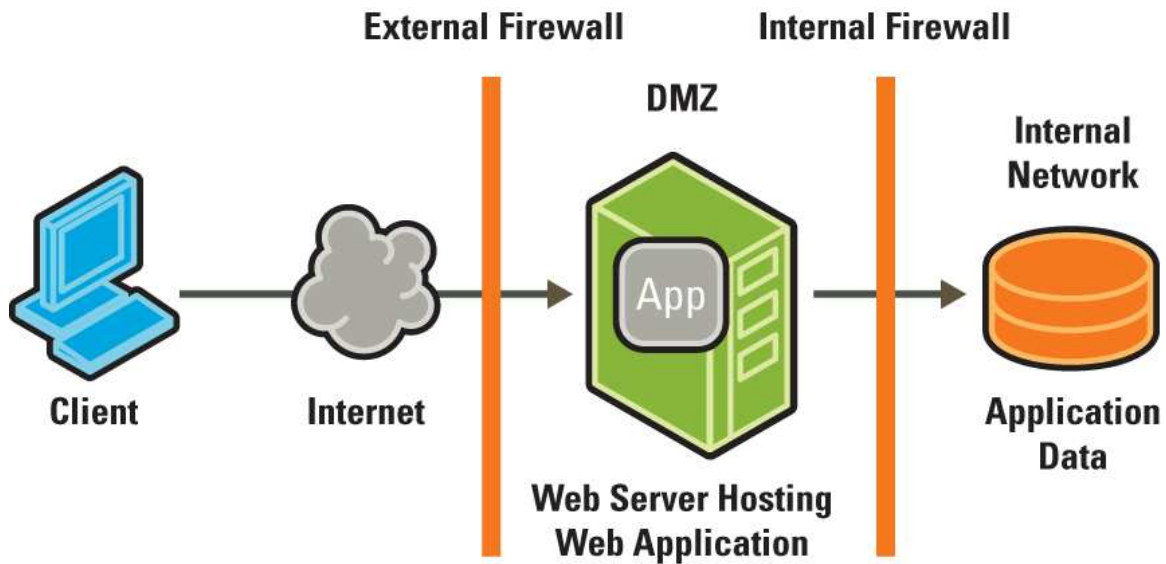


Figure 2. Common extranet architecture without identity and access management

In this arrangement, the client makes HTTP requests to the Web server from outside the enterprise's private network.

Putting the Web server in the Demilitarized Zone, or DMZ, allows limited access from the Internet at large. The DMZ is distinct from the internal network, which is completely inaccessible from the Internet. The Web server residing in the DMZ handles the HTTP requests by invoking the Web application, which is allowed to access the application data store resident in the internal network.

Identity and access management solutions, such as Symphony Web Access Manager, allow applications in the above scenario to focus on their intrinsic business value by offloading authentication, coarse-grained authorization, and audit functions to a centralized security solution.

There are two main architectures for identity and access management solutions. While both require a policy enforcement point (PEP) to be installed on the protected Web server or proxy server, a key difference between Web Access Manager and competing solutions is the location of the policy decision point (PDP) functionality. The first architecture, often referred to as a three-tier architecture, involves a separate policy server for PDP functions, as shown in Figure 3.

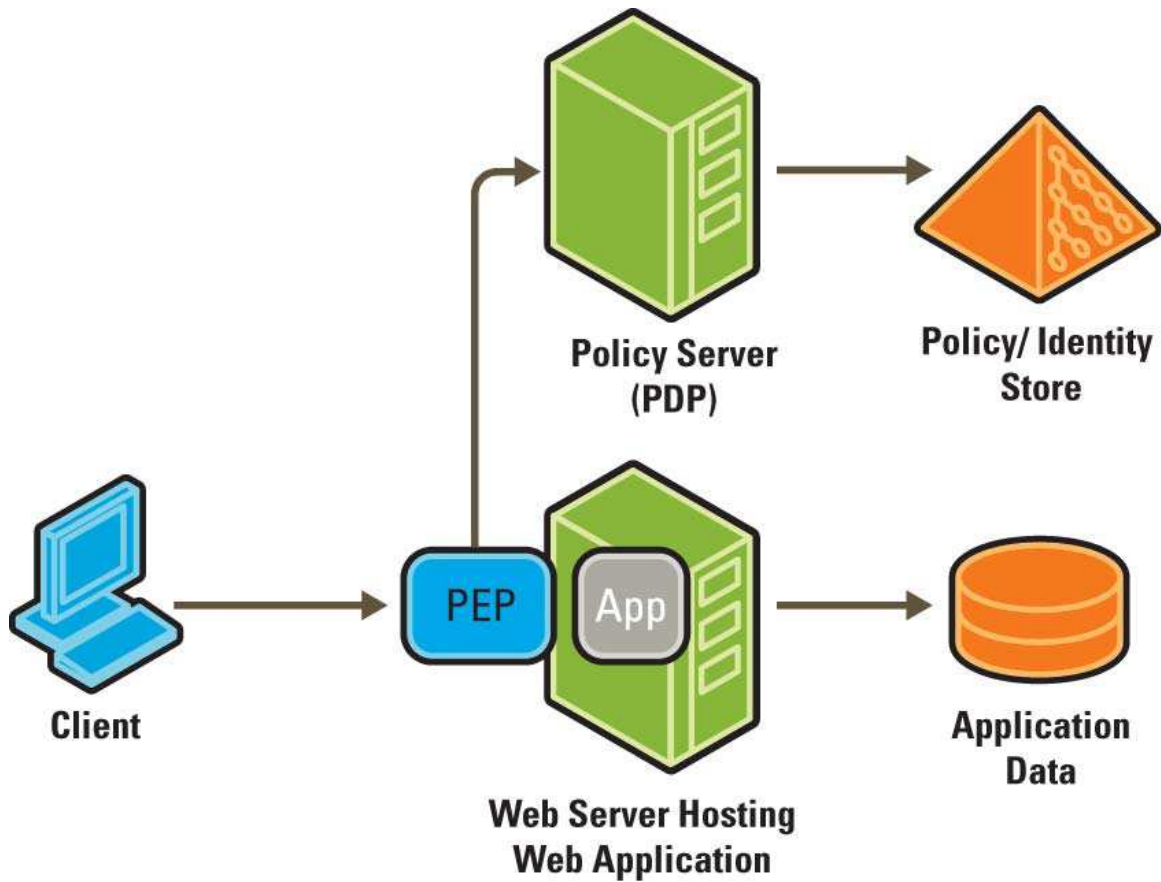


Figure 3. Common three-tier identity and access management architecture

The second architecture, often referred to as a two-tier architecture, does not involve a policy server, but rather combines PDP functionality with PEP functionality, as shown in Figure 4.

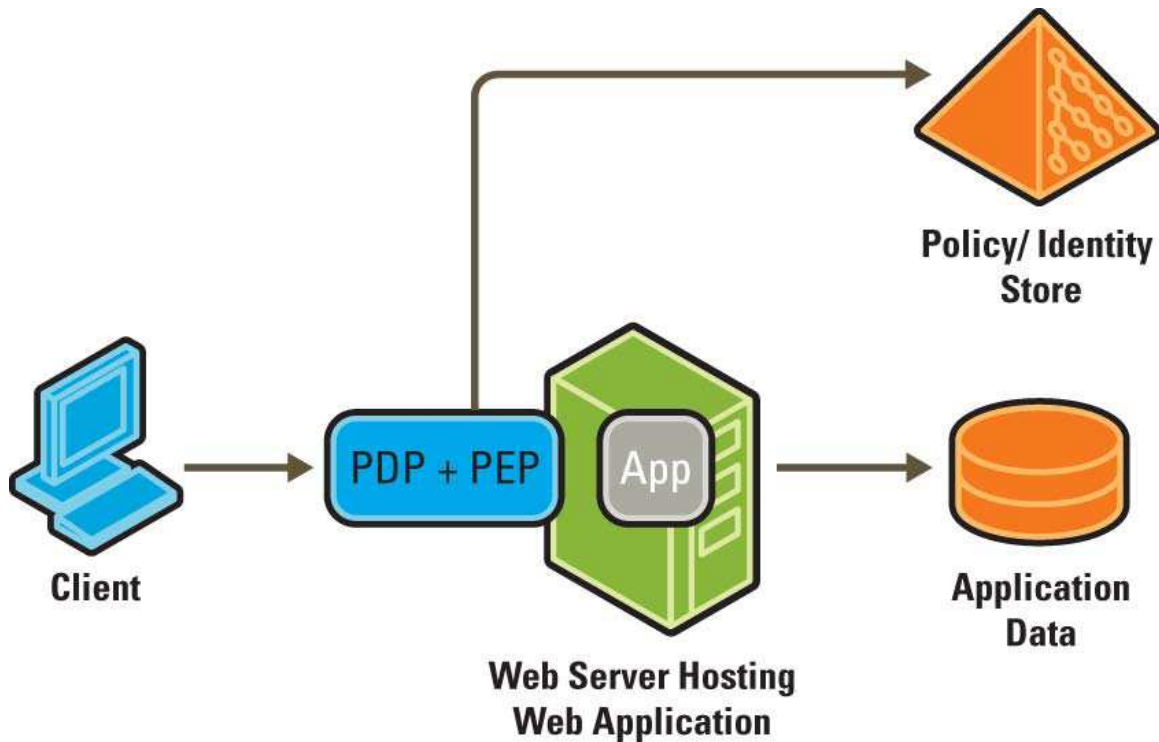


Figure 4. Two-tier identity and access management architecture

Symphony Identity Management Solutions uses the second architecture, with the PDP and PEP functionality combined in its Symphony Enforcement Agent software. This approach is simpler, involves less hardware and support, and scales naturally. In fact, independent tests prove that the Symphony architecture can outperform policy server-based architectures 2-to-1. Tests also showed that the Symphony architecture can scale linearly with additional hardware at a 1-to-1 ratio (of hardware added to increased throughput).

Simply put, the process of scaling Symphony Web Access Manager is identical to the process of scaling a solution without an identity and access management product because there is no middleware server that has to be scaled separately.

Figure 5 illustrates one way Symphony Web Access Manager can be implemented in an externally facing Web application scenario.

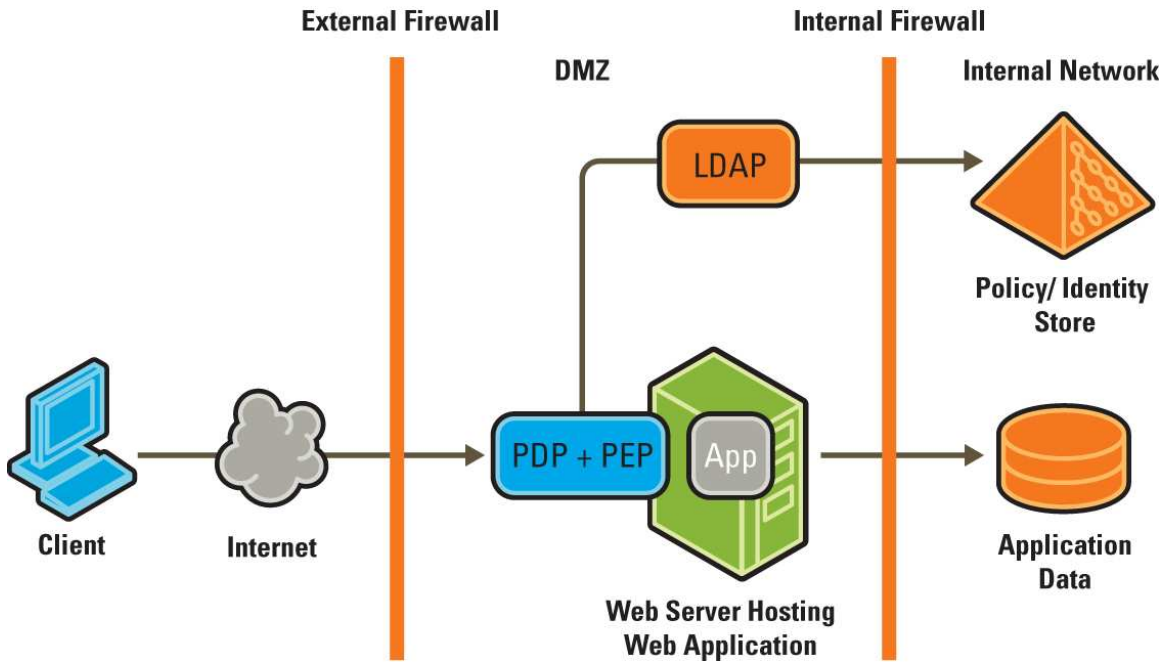


Figure 5. Symphony Identity Management in an extranet scenario

The Web application resides in the DMZ, co-resident with Symphony Enforcement Agent (PDP + PEP) on the Web server. Symphony Enforcement Agent accesses the policy and identity store in the internal network via standard LDAP. The external firewall can limit access to the Web server to only HTTP, and the internal firewall can limit access to just LDAP (Web server to policy/identity store) and the particular database protocol (Web server to application data store).

Figure 6 shows the typical environment when a three-tier architecture is applied to the same externally facing Web application scenario.

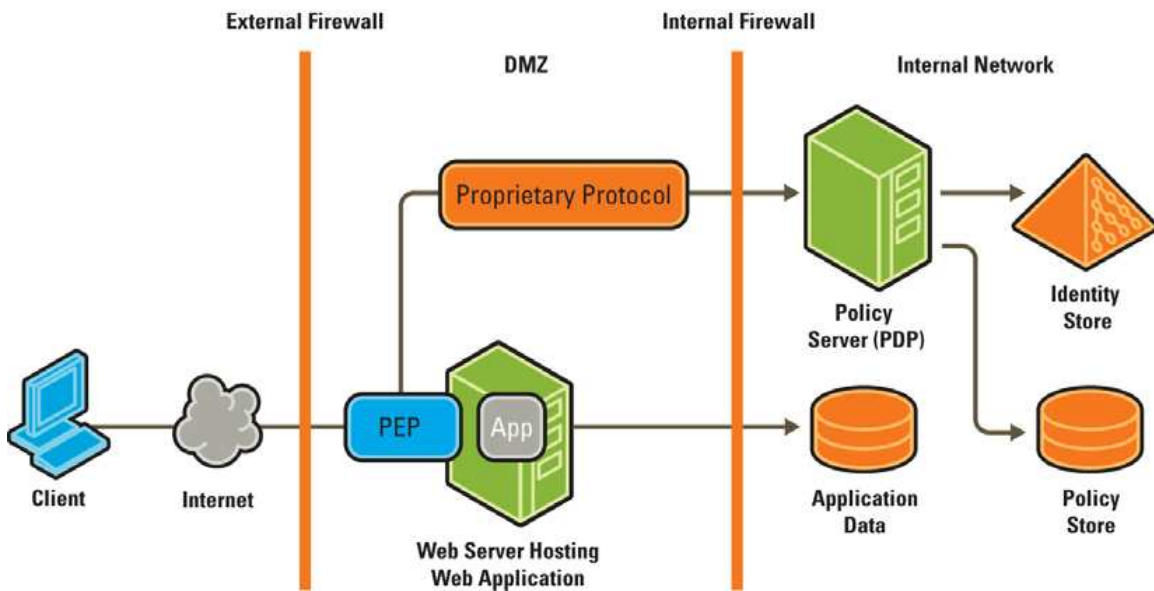


Figure 6. Common extranet three-tier architecture with identity and access management

There are two crucial differences implied by this architecture. First, it requires excess hardware and the associated support burden imposed by the addition of the policy server. Second, the internal firewall in this architecture must be configured to allow a proprietary, unmanageable protocol between the Web server and policy server (possibly on multiple ports) rather than the standard, widely-used LDAP protocol in the previous example.

While the performance, scalability, and support advantages of a two-tier architecture are widely acknowledged, security questions are sometimes raised. These usually boil down to two issues.

1. Are there additional risks imposed by opening access through the internal firewall to the identity store?

No. Policy servers routinely provide identity information to PEPs, but do so through some proprietary protocol. The net result of both two-tier and three-tier architectures is the same: making data from the identity store available to entities in the DMZ. The difference is that the three-tier architecture does this via a proprietary protocol, while the Symphony architecture uses the well-known LDAP protocol. Any additional protection that might be imagined by using a proprietary protocol amounts to nothing more than "security by obscurity," which adds no security at all.

Because the Symphony solution uses a standards-based protocol, the internal firewall and security monitoring systems are more easily configured to provide better security for the Symphony architecture than for competing architectures that use proprietary communication schemes.

2. Is performing policy decision functions in the DMZ more risky than performing them behind the internal firewall?

No. Vendors with three-tier architectures will often tell prospects that an escalation of privileges can result if the decision-making code (PDP) on the Web server in the DMZ is compromised. However, it follows that if the decision-making process can be attacked, then the enforcement process, which routinely caches previously received decisions, can just as easily be attacked. Therefore, moving the PDP out of the DMZ provides no additional protection.

For cases where companies are highly skeptical of the security in their DMZ, or where they are required to follow corporate policies that forbid either certain components in the DMZ or certain access through internal firewalls, Symphony supports a three-tier alternative that meets such requirements, yet still retains many benefits of the two-tier architecture. This is shown in Figure 7.

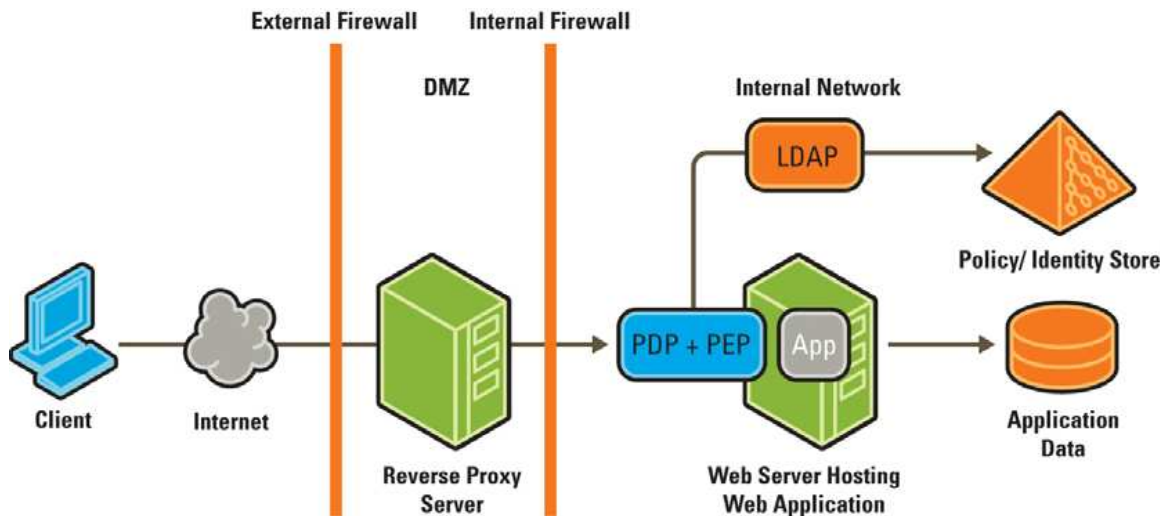


Figure 7. Symphony Web Access Manager in an extranet scenario using a reverse proxy server

The insertion of a reverse proxy allows all Web application assets to be moved out of the DMZ and into the internal network, including static content, dynamic content, policy decision-making, and policy enforcement. Furthermore, the internal firewall now only has to permit HTTP traffic between the proxy server and Web server, completely isolating the policy/ identity store and the application data store from the DMZ.

The reverse proxy server requires comparatively minimal computing resources, as it does nothing more than pass HTTP requests to the internal Web servers housing the protected applications. In most cases, multiple Web applications can share common reverse proxy resources. And in some cases, the firewall solutions themselves can perform reverse proxy functions, eliminating the need for separate hardware.

## Highlights of the Symphony Web Access Manager Architecture

### Scalability

Symphony Web Access Manager enables businesses to more easily secure and manage millions of users and their access to company information. The solution provides lightweight Web server plug-ins that run inline with protected Web servers. As traffic dictates scaling of the Web server, Symphony Enforcement Agent, with its use of native LDAP calls for authentication and authorization, scales as well. By using an LDAP directory as the identity and policy store, Symphony Web Access Manager leverages standard LDAP deployment strategies to support millions of users.

### Application availability

The Symphony Web Access Manager architecture is modular, with each component independently configurable to provide high availability. Since Symphony Enforcement Agent runs inline as part of the Web server or proxy server it protects, it is available as long as the server is available.

### Directory availability

Directory servers generally provide replication functions so that access to directory data is not dependent on a single directory server. Symphony Enforcement Agent is designed with options to leverage this directory server capability, including:

- > The ability to specify multiple directory servers when establishing a connection
- > The ability to reconnect after a directory server error without establishing a new connection handle

Using these features, Symphony Enforcement Agent is able to connect to the first available directory server in a configured list of servers. In the event of a server error during a directory operation, it can also reconnect to the first available server in that same list.

### **Manageability**

Symphony Web Access Manager has a minimal network footprint requirement, resulting in limited support and maintenance requirements for the enterprise. Symphony Enforcement Agent can be configured from a central Web-based application (WAM Policy Manager) which enables administrators to deploy, manage, and upgrade it with minimal effort.

## **Symphony Enforcement Agent Functionality**

### **Login**

When a user requests access to a Web application protected by Symphony Web Access Manager, Symphony Enforcement Agent looks for an appropriate credential. A Symphony Web Access Manager credential is created upon the user's first successful authentication. If Symphony Enforcement Agent does not find a credential, the user is asked to log in. The only Web applications that do not require a credential are those with an authorization level of "all users."

After completing a login, the user is redirected to the originally requested URL or sent to a configurable URL as a starting point. The Web application defines the authentication method required for access to the application. The supported choices of authentication methods are ranked according to relative strength, so that at runtime, Symphony Enforcement Agent can verify that the user has authenticated by a method of suitable strength. If the user has not authenticated, the agent presents the required authentication process. Out-of-the-box authentication methods include Basic, X.509, Basic +X.509, and RSA SecurID. Other authentication methods are supported through custom login pages using the Symphony Application Programming Interfaces (APIs) in .NET and Java.

#### **BASIC AUTHENTICATION**

Basic authentication is the verification of the user ID and password against ID/password pairs stored in the directory.

#### **X.509 AUTHENTICATION**

Symphony Enforcement Agent can use X.509 client certificates presented in the SSL connection for authentication. This form of authentication may be used alone, in which case username and password data will not be collected; or in conjunction with basic authentication, where the submitted user identity will be verified to match the identity derived from the certificate. This is an extra measure of authentication above the acceptance of the client certificate by the Web server. Elements of the certificate's SubjectDN may be mapped to attributes in the directory, and selected elements can be used to uniquely identify a user in the directory.

#### **STRONG AUTHENTICATION (SUCH AS RSA SECURID)**

Symphony Enforcement Agent provides out-of-the-box integration with RSA SecurID technology to deliver strong two-factor authentication. It does this by redirecting the user's browser to the SecurID Web agent. Once the SecurID Web agent completes its authentication process, a Symphony Enforcement agent on that same server will construct runtime credentials and forward the user to the originally requested URL. Other strong authentication methods, such as Biometrics, Smart Card, OATH, and others, can be added via minor customization.

### **Validating runtime credentials**

The Symphony Web Access Manager credential is an encrypted, session-based HTTP cookie that is used for authorization. The following information in the user's Symphony Web Access Manager credential is verified each time access is requested:

- > The name of the user's directory entry
- > The user's ID
- > The IP address of the browser from which the user logged in
- > The user's last access time for timeout conditions

With this information, Symphony Enforcement Agent:

- > Ensures that the current date/time is within the start and end dates in the user's directory entry. This allows administrators to configure user accounts to be valid for only a specific date range.
- > Ensures that the account status attribute in the user's directory object is set to "active." This enables administrators to immediately disable a user account with a single click, using the Symphony Web Access Manager Web-based management console. (WAM Policy Manager)

Based on configuration, Symphony Enforcement Agent can also:

- > Ensure that the IP address of the current request matches the IP address in the user's credential.
- > Ensure that the last access time of the user is within the configured timeout limits.

If checks for start/end date, account status, or IP address fail, Symphony Enforcement Agent will redirect the user to a configurable "Invalid Access" page. If the check for timeout fails, the user's credential will be revoked and the user will be forced to authenticate again.

Validation of runtime credentials enables support for single sign-on (SSO) using HTTP cookie values encrypted by Symphony Web Access Manager.

### **Authorization**

While authentication is the process of validating the user's identity and relationship with the company, authorization is the process of determining what resources they can access.

For authorization, Symphony Enforcement Agent uses three objects from a company's central LDAP compliant directory: users, roles, and resources. Access to requested resources is enforced at runtime based on the user's identity, his or her assigned roles (i.e., a job title), and the permissions associated with those roles.

#### **USERS**

The user represents a unique identity within the enterprise. Each user has attributes and credentials which uniquely identify him or her in the system.

#### **ROLES**

Symphony Web Access Manager is based on the National Institute of Standards and Technology (NIST) role-based access control (RBAC) model. Role objects represent logical containers of application access and administrative permissions. Standard static roles are assigned to users through the Symphony Web Access Manager Web-based identity and policy management interface. A single role can be assigned to multiple users, and each user can be assigned multiple roles. The cumulative permissions of the assigned roles define application access and administrative privileges for the user. This method enables privileges for large groups of users to be quickly altered simply by adjusting a commonly assigned role.

In addition to the standard roles described above, Symphony Web Access Manager enables the use of dynamic roles and adaptive roles. Dynamic roles are dynamically evaluated against a

user's profile at time of access. Any change to a user's attributes, such as their job title, automatically changes the evaluation result of the dynamic roles and, consequently, their permissions. Dynamic roles provide functionality similar to rules-based Identity Management solutions, but with the manageability enabled by a role-based access control model. Dynamic roles differ from all other roles supported by Symphony Web Access Manager because they are not statically assigned to users. Instead, they are dynamically associated with a user each time access is requested.

Adaptive roles are statically defined, but dynamically determine a user's rights based on his or her identity and specific organization. This allows businesses to create one management role that can be leveraged by various organizations across the enterprise.

Users that share a certain adaptive role have the same set of permissions; however, those permissions are only relative to the organization(s) in which they belong.

#### RESOURCES

Symphony Enforcement Agents control access to HTTP or HTTP over SSL requests by identifying each URL as part of a Web resource that is secured by Symphony Identity Management Solutions. The Web resource is a logical grouping defined by a protocol (for example, HTTP), host, port, and set of paths or URLs. Web resources are given authorization levels required for access. Possible authorization levels are "all user," "valid user," and "role-based." An authorization level of "all users" allows access without an authorization check. An authorization level of "valid users" permits access to any user with a valid Symphony Web Access Manager credential. And "role-based" authorization requires that the user have a valid Symphony Web Access Manager credential and a role object with access to the Web resource.

### **Symphony Enforcement Agent Configuration**

Symphony Enforcement Agent can be customized to meet the needs of evolving business environments. Administrators have broad flexibility, including configuration options for authentication, inactivity timeouts, and access to resources within a URL, and security for multi-domain deployments.

#### **Authentication**

Symphony Enforcement Agent can be configured to use either custom or internal logins. A custom login allows for a tailored login process and can be written in any language that produces a Web page. An internal login is a static Web page streamed back from Symphony Enforcement Agent. This increases performance in systems with high numbers of logins. An internal login configuration can also allow Symphony Enforcement Agent to check for password policy information, if desired.

#### **Enforcement of inactivity timeouts**

Inactivity timeout thresholds may be specified at the following levels:

- > User level
- > Web application level
- > Role level
- > Symphony Enforcement Agent default

At each level, an inactivity timeout value may or may not be present. When present, this value may specify a number of minutes for the threshold, or it can be set to reflect that there is no timeout.

Administrators can configure the method in which Symphony Enforcement Agent examines levels and derives a timeout threshold for a specific request. For example, Symphony Enforcement

Agent can be configured to choose the shortest or longest timeout available, or to search the levels in a particular order for a defined value and select the first one it finds.

### **Access to resources within a URL (special suffixes)**

Symphony Enforcement Agent can be configured to automatically grant access to requests where the end of the URL matches any in a list of suffixes. The list is completely customizable. For example, it can be configured to automatically grant access to any URL ending in “.gif” or “.jpg.” Static and dynamic HTML pages to which a user is granted access will often reference other elements, such as image files that become separate HTTP requests. Rather than taking the time and CPU cycles to determine the user’s access to these additional elements, this configuration setting improves performance by allowing them to pass without additional processing.

### **Security for multi-domain environments**

In many cases, a single Web-based system may span multiple DNS domains. Symphony Web Access Manager can provide SSO and manage runtime credentials across these multiple domains. In order to do this, one domain within the Symphony Web Access Manager installation is chosen as the primary domain and all other domains are considered satellite domains. Within each domain there may be any number of Web servers. Each Symphony Enforcement Agent must be configured with a primary or satellite designation.

When a user accesses a protected Web server in a satellite domain, the Symphony Enforcement Agent instances in that domain will communicate with a Symphony Enforcement Agent instance in the primary domain to obtain runtime credentials. If no credentials exist in the primary domain, the Symphony Enforcement Agent in the satellite domain presents the user with the authentication process which is used to create credentials in both the primary and satellite domains. The list of satellite domains that have been given runtime credentials are then tracked in the primary domain and is used during log out and timeout. At log out time, Symphony Enforcement Agents in the primary domain communicate with Symphony Enforcement Agents in the satellite domains to ensure credentials in all domains are cleared. When a timeout condition occurs in one domain, another domain may have credentials with a last access time that would satisfy the current request. Symphony Enforcement Agents in the satellite domains can exchange credentials with Symphony Enforcement Agents in the primary domain to avoid unnecessary re-authentication.

### **Caching and pooling**

To increase performance and decrease network traffic, Symphony Enforcement Agents can be configured to cache information needed to perform the authorization of a request. Symphony Enforcement Agents can also keep a set of live connections to the LDAP directory open for all directory operations. The connection pool increases performance by eliminating the time needed to negotiate a new connection, which can be significant when using LDAP over SSL.

### **Bookmark usage**

Users can bookmark Web application URLs and use them to attempt access at a later date. Because Symphony Enforcement Agent intercepts all access attempts to a given Web server, if a user no longer has permission to access a bookmarked resource, access is still governed by the policies in the directory.

### **Securing Access to XML Web Services**

Just like Web applications, XML-based Web Service requests can be protected to verify the identity of the calling application and its business relationship with your company.

Symphony Web Access Manager protects Web Services via transport-level security, using one of two mechanisms:

> SSL with client certificate authentication

## > HTTP Basic Authentication over SSL

SSL with client certificate authentication will cryptographically verify that the Web Services client — the requesting application — is recognized by the organization providing the Web Service. The Web Service does this by verifying that the client certificate was issued by a trusted certificate authority. Symphony Web Access Manager provides further authentication functionality by checking that the client certificate was issued with a name that matches a known party or organization in its central identity repository (an LDAP directory).

HTTP Basic Authentication over SSL is simpler to setup than SSL with client certificate authentication, but doesn't enjoy the same level of cryptographic assurance. Using this mechanism, a Web Services client sends a username and password in a well-known header of the HTTP message, but it is sent over SSL to prevent attackers from reading the username/password data. The Symphony Enforcement Agent authenticates the username and password just like it would for a human-submitted login form.

Regardless of the mechanism used, there must be a user identity created for the Web Service client in the directory used by Symphony Identity Management Solutions. This allows the Symphony Enforcement Agent to go beyond authentication and perform its characteristic authorization process to determine the Web Service client's entitlement to access the logical web resource definition that represents the physical Web Service.

### **Audit**

Symphony Web Access Manager delivers comprehensive auditing and reporting capabilities for security and administrative entitlements, enabling you to track potential security problems, help ensure user accountability, and analyze evidence in the event of a security breach. Auditing is increasingly important for compliance with regulations, including HIPAA, Sarbanes-Oxley, Basel II, and the Gramm-Leach-Bliley Act.

Symphony Web Access Manager creates extensive logs that can be easily integrated with leading intrusion detection products. The software records the details of each user request for a managed resource, and each change made to objects within the directory. The solution provides:

- > Access pattern logging
- > Failed access logging
- > Administration logging
- > Directory snapshot reporting

All changes made to the directory through Symphony Web Access Manager can be logged. Each log event identifies who made the change, the data content prior to change, and the end-result data. This includes any changes made to the user, role, organization, or application. These audit logs can also be digitally signed to prevent log tampering in the database.

### **Conclusion**

A growing need for interoperability and other Web initiatives are presenting IT administrators with increasingly complex challenges. These administrators are looking for effective answers to key questions:

- > How can we scale cost effectively and with minimal network impact?
- > How will we manage access for applications spread across varying locations, technologies and domains?

Symphony Web Access Manager provides the solution. Lightweight Symphony Enforcement Agent plug-ins install directly on Web or proxy servers to enforce access control. The Symphony Enforcement Agents access role, user, and application data stored in a centralized directory to

eliminate the need for cumbersome middleware components — such as policy servers — and enable deployment of a single security framework across the extended enterprise.

Symphony Web Access Manager is a modular, end-to-end identity management solution. It is easily deployed to solve tactical problems, while laying a foundation that can be leveraged and built upon to meet longer-term strategies. This phased approach will help you benefit from the advantages of identity management more rapidly, and drive the successful rollout of centralized identity management across the enterprise.