LAYER SEVEN SECURITY

DEFENSE IN DEPTH

AN INTEGRATED STRATEGY FOR SAP SECURITY

WHITE PAPER

Layer Seven Security offers no specific guarantee regarding the accuracy or completeness of the information presented, but the professional staff of Layer Seven Security makes every reasonable effort to present the most reliable information available to it and to meet or exceed any applicable industry standards.

SAP AG is neither the author nor the publisher of this publication and is not responsible for its content, and SAP Group shall not be liable for errors or omissions with respect to the materials.

DEFENSE IN DEPTH

AN INTEGRATED STRATEGY FOR SAP SECURITY

CONTENTS

INTRODUCTION

The protection of SAP systems against unauthorized access and changes requires security measures at multiple levels. According to SAP recommendations, this should include measures covering landscape architectures, operating systems and databases, as well as SAP technologies, applications and authorizations. [1]

This white paper outlines an integrated strategy for securing SAP systems based on the principles of Defense in Depth. The strategy is designed to protect the confidentiality, integrity and availability of SAP programs and data through countermeasures applied within each interconnected layer in SAP environments.

Defense in Depth is the only practical strategy for information assurance in highly integrated SAP landscapes susceptible to a variety of attacks through numerous access points. The strategy requires the implementation of multiple obstacles between adversaries and their targets. This is designed to lower the risk of a successful attack, contain the impact of a network intrusion and improve the likelihood of detection.

The deployment of nested firewalls coupled with intrusion prevention represents the first line of defense and is an important component of network-level security. However, organizations should not rely exclusively upon such technologies. Both firewalls and intrusion prevention systems can be bypassed by skilled attackers, evidenced by recent well-publicized data breaches. Firewalls are especially vulnerable. The most common form of network firewalls, stateful packet filters, do not analyze application payloads. Consequently, they are ineffective against SAP attacks. Application gateways provide a greater level of protection and are therefore recommended for high-integrity environments.

Network controls should be balanced with appropriate policies and procedures, physical controls and monitoring. The latter can be performed through Security Information and Event Management (SIEM) solutions. They should also be supported by technical measures such as encrypted communications, hardened servers, robust programs and effective access controls, designed to protect information resources even if a network is breached.

These measures are applied across four distinct areas in SAP systems: Application, Platform, Program and Endpoint. The secure configuration and management of these areas lowers the risk of system intrusion, protects the confidentiality of business information and ensures the authenticity of users. Each area is reviewed in detail in the white paper.

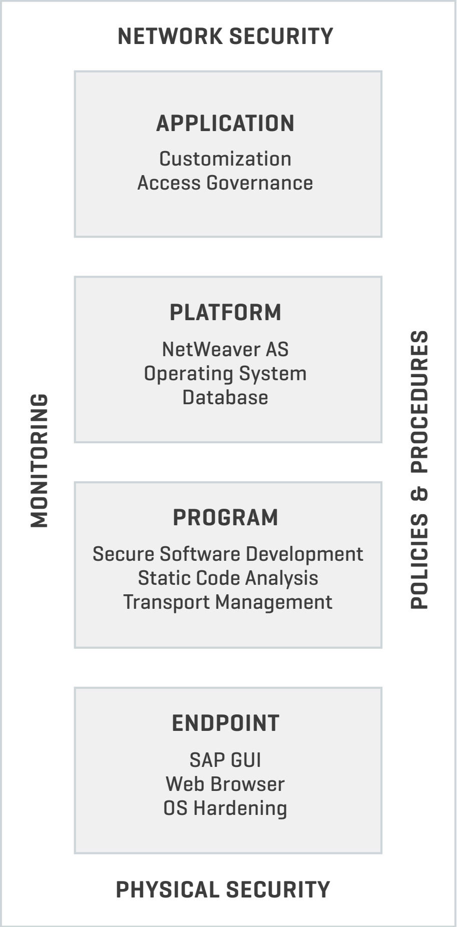

Figure 1: Defense in Depth for SAP Systems

[1] Secure Configuration of SAP NetWeaver Application Server Using ABAP, SAP AG, 2012

APPLICATION SECURITY

The starting point of an integrated SAP security strategy is application-level controls in the areas of Customization and Access Governance.

Customization refers to the process of configuring SAP systems to meet the specific needs of each customer. SAP software is highly configurable by design to support diverse requirements and reduce timeframes for deployment. In most cases, standard software is extensively modified to agree with business requirements defined for an implementation. Such customization is performed at multiple levels through the Reference or Enterprise Implementation Guide (IMG). This includes defining organizational structures, mapping system landscapes and maintaining global settings for areas such as currencies, reporting periods and time zones. It also includes adjusting application-specific parameters or variables. Parameters have an important bearing on system security. Therefore, parameter settings should be determined in accordance with standards issued by SAP to avoid security issues arising from misconfigurations. The standards are contained in detailed Security Guides covering all applications, components and industry solutions, available at the SAP Service Marketplace.

The second component of application-level controls in SAP environments is Access Governance. Access control structures must meet requirements for limiting access to sensitive data and the separation of incompatible duties. These requirements are designed to safeguard assets and minimize the risk of error. They are intended to mitigate strategic risks through the management of action-orientated risks. For example, limiting access to journal entry posting and implementing dual control for entries lessens the risk of inaccuracies in financial statements. This in turn reduces the risk of regulatory sanctions and losses in shareholder value.

In SAP systems, access to functions, programs and resources is controlled through the authorization concept. Actions such as creating a new vendor, changing an employee record or entering an invoice cannot be performed by a user without the relevant authorizations that allow such tasks. Authorizations therefore control the actions users are able to perform in SAP after logon and authentication. Often, combinations of several authorizations are required to perform a specific task.

Authorizations are grouped into authorization objects which are assigned to different object classes. Authorization objects must be combined with the appropriate fields and values to enable an action in SAP. Fields can include values for activities such as create (01), change (02), display (03) and delete (06), and restrictions on company codes or other organizational levels in which the activities are permitted. Authorizations are clustered into profiles and roles which are then assigned directly to user master records.

Program execution in SAP is performed predominately through transactions. These are grouped by application and module and can be called through the SAP menu path or through transaction codes. S_TCODE is the first authorization object checked by SAP at the start of any transaction. This is usually followed by a number of subsequent checks defined in the source code of the respective program through inspection strings known as authority-check statements. The checks are performed by the system against assigned authorizations and field values stored in the user buffer during each session.

The effective management of SAP authorizations presents a formidable challenge to organizations. SAP ECC 6.0 alone has over 370,000 programs, 70,000 transactions and 1000 authorizations objects. This does not include custom-developed objects which, in some cases, increases the number of programs, transactions and authorizations by 30 percent. The standards outlined below are designed to simplify the complexities of this challenge and enable organizations to design, implement and maintain robust access governance procedures for SAP systems.

SINGLE AND UNIQUE USER ACCOUNTS

Users should not be assigned multiple accounts in the identical system. This can lead to the accumulation of excessive or conflicting authorizations without detection. Also, users should be assigned unique, personal accounts for system access. Shared or generic accounts increase the risk of unauthorized access through password sharing. Actions performed by generic users are also difficult to trace to specific individuals which impacts logging and monitoring.

LEAST PRIVILEGE

Users should be granted the minimal authorizations required to perform their tasks. Furthermore, authorizations should only be granted for the required organizational groups. Broad business roles require greater authorizations and therefore increase the risk that users may be assigned conflicting privileges. This can be addressed through approval, monitoring and other compensating controls when business roles cannot be redesigned.

ROLES BASED ACCESS CONTROL (RBAC)

Authorizations should not be assigned directly to users. Rather, permissions should be incorporated into roles which are then provided to users. This creates transparent access control models and enables efficient user provisioning. SAP roles should closely align to the tasks performed by role members in organizations. Indirect role assignment through Organizational Management (OM) in HCM can be used to align SAP roles to actual roles and responsibilities and automate user provisioning.

SEPARATION OF DUTIES

SAP landscapes contain an array of diverse and integrated applications. In most cases, user management is performed centrally within such landscapes and users with broad authorizations are able to perform wide-ranging, cross-application functions that in combination may present a risk to organizations. For example, users granted authorizations for both invoice entry and asset disposal through transactions such as FB60, FB65 and F-92 are able to process fraudulent invoices for fictitious asset acquisitions and eliminate the risk of detection through asset sales or retirement.

Risks can also occur within an application area. For instance, creating and maintaining employee records using authorizations such as PA30 and PA40 can lead to a risk when combined with the permission to run payroll processing programs through the authorization object P_ABAP.

Hence, the identification and segregation of conflicting authorizations is a vital component of access management in SAP systems. The separation of duties implements preventative internal controls that significantly reduce the opportunity for fraud and error. It should be applied at both the profile and cross-profile level and include custom-developed authorization objects, profiles, roles and transactions.

CONTROL BENCHMARKS

The assignment of sensitive authorizations and the separation of duties should be conducted in accordance with generally-accepted benchmarks for SAP systems. Deviations from the standards in such benchmarks should be documented and supported with adequate compensating controls. Reliable sources for control standards include SAP Best Practices and rule-sets embedded in SAP GRC and leading third party authorization management solutions. Matrices available in the SAP Developer Network may be used as a reference but should not be considered a benchmark for control standards.

DOCUMENTATION

The SAP authorization structure should be maintained in documented form as a reference for internal and external stakeholders. This should include information related to profiles, single and composite roles, user groups, administrators, authorization and transaction assignments, segregation of duties and procedures for profile changes and access provisioning.

APPROVAL

An effective SAP access management framework should include requirements for approvals at multiple levels. The content of authorization profiles in SAP roles should be approved and controlled by designated role owners representing business and technical areas. The responsibility for managing access assignments and segregation of duties across roles should be assigned to overall system owners since role owners generally only have visibility to risks in their specific domain. Also, every assignment of authorizations to users must be pre-approved. In most cases, approval should be explicit. However, implicit approval is acceptable when provisioning access through indirect assignment using OM.

MAINTENANCE

Security Notes and Upgrades are issued by SAP to patch missing or incomplete authorization checks in standard programs. Notes should be applied within 30 days of the release date. Upgrades should be performed through procedures in transaction SU25. This provides a comparison of existing check indicators against SAP defaults. The alternative procedure involves assigning the SAP_NEW profile included in each upgrade to all users. This approach carries significant risks and could potentially violate the principles of least privilege and the separation of duties. The upgrade procedure is considerably easier and less time consuming in environments where SAP template roles are leveraged to provision user access, rather than custom-developed roles. Templates cover approximately 80 percent of the roles required for standard functions and processes.

LOGGING

SAP systems should be configured to provide an audit trail of significant user actions. This includes the creation, change or deletion of documents and other objects. It also includes user provisioning and actions related to the administration of authorization profiles and roles. Actions should be traceable to specific users and include date and time stamps. Logs should be retained for a sufficient period, usually 12 months. Regular archiving of logs will minimize the size of log files and any impact on system performance.

MONITORING

SAP roles, profiles and authorizations should be periodically reviewed and validated by business owners. Any errors in permissions and assignments should be removed within a reasonable period. Inactive and locked users should be identified and investigated. Monitoring should also include a regular review of relevant application-level settings in the IMG and should be augmented with independent assessments performed by internal or external auditors or consultants.

PLATFORM SECURITY

The technical components of SAP environments are NetWeaver Application Servers (AS) and underlying database and operating systems which together provide the platform for SAP applications. Vulnerabilities at the platform level can enable internal and external attackers to bypass application-level controls. Therefore, approaches to security that focus primarily upon applications may provide a false sense of security if architectural and configuration flaws are not addressed at the platform level.

The NetWeaver AS is the technical foundation of the entire SAP software stack. It provides the runtime environment for SAP applications and includes work processes for ABAP and Java programs, gateways and modules for managing RFC, Web-based and other forms of communication protocols, tools to manage user roles, profiles and authorizations, and utilities that control certain database and operating system functions. The secure configuration and management of the NetWeaver AS is therefore a vital component of an integrated SAP security strategy. Application servers must be secured against common threats and vulnerabilities that can lead to fraud, espionage and sabotage. This should include the measures outlined below.

NETWORK FILTERING

Unnecessary network ports and services should be disabled. In most cases, this means blocking all connections between end user networks and ABAP systems other than those required by the Dispatcher (port 32NN), Gateway (33NN), Message Server (36NN) and HTTPS (443NN). NN is a placeholder for SAP instance numbers. Administrative access should only be allowed through secure protocols such as SSH and restricted to dedicated subnets or workstations through properly configured firewall rules

SAP GUI

Installation of the latest version of SAP GUI, ideally 7.20 with active and properly configured security rules. Scripting and input history should be deactivated or otherwise controlled. The section related to client security contains further information on security measures for SAP GUI.

PASSWORD MANAGEMENT

Implementation of strong password policies, access controls for password hashes in tables and activation of the latest hashing algorithms. Default passwords should be changed for standard users and password hashing mechanisms should be upgraded to the most current applicable versions. Wherever possible, downward-compatible hashes should be removed from databases.

NETWORK ENCRYPTION

SAP client and server communication traffic is not cryptographically authenticated or encrypted. Therefore, data transmitted within SAP networks can be intercepted and modified through Man-In-The-Middle attacks. Secure Network Communication (SNC) should be used for mutual authentication and strong encryption. This can be performed natively if both servers and clients run on Windows. Non-SAP software is required to secure connections between heterogeneous environments such as AIX to Windows.

WEB SERVICES

SAP functions and programs can be Web-Enabled. This is managed through the Internet Communication Framework (ICF), accessible through transaction SICF. Many of the default services in ICF could enable unauthorized and malicious access to SAP systems and resources, often without authentication. Hence, services that are not required for business scenarios should be deactivated. This should include SAP/RFC, Echo, XRFC, WEBRFC, IDOC and IDOC_XML.

REMOTE FUNCTION CALLS (RFC)

RFC is the most widely used communication protocol in SAP landscapes and supports integration between SAP systems and environments. Trust relationships should not be established in RFC connections between systems with differing security classifications. Furthermore, hardcoded user credentials should be avoided in RFC destinations. Connections with excessive privileges are a particularly high risk. This includes connections configured with SAP_ALL privileges, regardless of whether the user type is set to communication or system rather than dialog.

SAP GATEWAY

The Gateway is used to manage RFC communications which support SAP interfaces such as BAPI, ALE and IDoc. Access Control Lists (ACL) should be created to prevent the registration of rogue or malicious RFC servers which can lead to the interruption of SAP services and compromise data during transit. Furthermore, Gateway logging should be enabled and remote access disabled.

MESSAGE SERVER

The Message Server is used primarily as a load balancer for SAP network communications. Similar to the Gateway, the Message Server has no default ACL. Therefore, it is susceptible to the identical vulnerabilities. Network access to the Message Server port should be filtered through a firewall and an ACL should be established for all required interfaces.

PATCH MANAGEMENT

SAP periodically releases patches for programming and other flaws through Security Notes, available at the Service Market Place. Standard reports such as RSECNOTE should be regularly reviewed to identify missing Security Notes. Alternatively, the SAP Solution Manager should be configured to manage Notes and support the change process for registered components. Notes with a severity rating of 1 require immediate attention. Notes with a severity rating of 2, 3 or 4 should be targeted for implementation within 30 days of release.

LOGGING AND MONITORING

The Security Audit Log should be enabled to record specific security events such as changes to user master records, logon attempts using SAP* and successful and unsuccessful logons. These events are recorded in local files stored on application servers. Static and dynamic filters should be configured for specific clients, users and classes to ensure that critical events are configured and logged.

1	Network Filtering
2	SAP GUI
3	Password Management
4	Network Encryption
5	Web Services
6	Remote Function Calls (RFC)
7	SAP Gateway
8	Message Server
9	Patch Management
10	Logging and Monitoring

Table 3.1: NetWeaver AS Security

SAP services such as EarlyWatch (EWA) and the Computing Center Management System (CCMS) should be used to monitor certain security events and parameters. However, they do not provide the same coverage as professional-grade vulnerability assessment solutions engineered specifically for SAP systems.

The other components of SAP platforms include database servers and operating systems. Such components should be configured in accordance with SAP recommendations covering areas such as establishing authentication schemes, protecting system and login users, changing default passwords, domain-level settings and managing access privileges for data tables, files, directories and other resources. However, SAP recommendations are not intended to be exhaustive. For example, SAP has no specific recommendation for database encryption. Therefore, databases and operating systems should also be secured in line with the more comprehensive recommendations issued by software vendors. Security guidance issued by organizations such as CIS, NIST or SANS may also be used as a benchmark. However, both vendor-specific recommendations and benchmarks should be applied carefully and selectively since they may conflict with SAP requirements.

PROGRAM SECURITY

The third component of an integrated SAP security strategy is the development of secure custom programs, free of code-level vulnerabilities.

SAP systems are designed to perform thousands of distinct functions ranging from adding a vendor to a list of approved suppliers, performing a transport to implement a change in a specific system, and encrypting/decrypting traffic between servers or clients. These functions are performed by programs stored in the database table known as REPOSRC that are called when requested by work processes in the NetWeaver AS.

SAP programs are developed using two distinct programming languages: Advanced Business Application Programming (ABAP) and Java. Both are vulnerable to coding errors that could expose SAP programs to exploits such as code, OS and SQL injection, cross-site scripting, cross-site request forgery, buffer overflow, directory traversal and denial of service. SAP programs are also susceptible to missing or broken authority-checks that could lead to the unauthorized execution of programs. Finally, programs can contain backdoors through hardcoded credentials that bypass regular authentication and authorization controls, as well as malware known as rootkits that provide attackers with remote, privileged access to system functions and resources. Table 4.1 lists the 25 Most Dangerous Software Errors identified by the Common Weaknesses Enumeration (CWE), co-sponsored by the Office of Cybersecurity and Communications at the U.S Department of Homeland Security. ABAP and Java programs are vulnerable to 70% of the vulnerabilities in the list and are highlighted in red.

SAP performs a rigorous code review for all standard or delivered programs prior to release and regularly issues Security Notes to patch vulnerabilities detected after release. Custom programs are rarely subject to the same level of scrutiny. Programs developed by in-house or off-shore developers to meet the needs of customers not met by standard SAP functionality are often laden with vulnerabilities that, when exploited, undermine the integrity of entire SAP landscapes. Such landscapes are only as strong as their weakest point. A robust application layer fortified with properly configured platforms can still be breached through vulnerabilities at the program level.

SAP does not assume responsibility or liability for losses arising from the exploitation of vulnerabilities in custom code. Customers are expected to develop and apply appropriate software development procedures to manage such risks. Procedures should include requirements for software integrity and security and should not focus exclusively on measures such as functionality and performance. Specific examples include the use of open rather than native SQL, avoiding arbitrary input for dynamic SQL statements, encoding user input before output, removing hardcoded users, secure construction of SELECT statements, and input validation through existence and length checks, canonicalization, type checks, range checks and white or black list filters.

1	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
2	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
3	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
4	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
5	Missing Authentication for Critical Function
6	Missing Authorization
7	Use of Hard-coded Credentials
8	Missing Encryption of Sensitive Data
9	Unrestricted Upload of File with Dangerous Type
10	Reliance on Untrusted Inputs in a Security Decision
11	Execution with Unnecessary Privileges
12	Cross-Site Request Forgery (CSRF)
13	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
14	Download of Code Without Integrity Check
15	Incorrect Authorization
16	Inclusion of Functionality from Untrusted Control Sphere
17	Incorrect Permission Assignment for Critical Resource
18	Use of Potentially Dangerous Function
19	Use of a Broken or Risky Cryptographic Algorithm
20	Incorrect Calculation of Buffer Size
21	Improper Restriction of Excessive Authentication Attempts
22	URL Redirection to Untrusted Site ('Open Redirect')
23	Uncontrolled Format String
24	Integer Overflow or Wraparound
25	Use of a One-Way Hash without a Salt

Table 4.1:
CWE Top 25 Most Dangerous Software Errors
(Source: CWE/MITRE, 2012)

Organisations should adhere to the Secure Programming Guidelines issued by SAP to prevent common code-level vulnerabilities and implement static code reviews to detect and correct coding errors. Standard SAP tools such as Code Inspector can be used to perform static checks and tests for development objects. Code Inspector is accessed through the ABAP Workbench or directly through transaction SCI. The default check variant includes some checks for security risks. Errors, warnings and messages generated by the Code Inspector should be investigated and resolved before the release of transports.

Code Inspector does not match the performance of SAP add-ons designed to detect a wider array of vulnerabilities in SAP programs. The Security Scan Solution within the Extended Program Check (SLIN_SEC) should be used for both quality assurance of new programs and the existing custom code base. SLIN_SEC is a component of the ABAP Test Cockpit (ATC) which is integrated into the SAP Workbench. It is tuned to detect and auto-correct suspicious statements in programs. It also prevents the deployment of malicious code through the SAP Transport Management System.

ENDPOINT SECURITY

The fourth component of an integrated SAP security strategy is endpoint security. SAP clients are often a target for attack since they can provide an unguarded corridor to enterprise applications and platforms. There are two primary clients supporting a variety of user interfaces. The first is SAP GUI. The desktop variant of SAP GUI is a thick client installed on Windows, Apple or UNIX workstations. It is the most widely used method of accessing SAP application servers and is the rendering engine for the desktop version of the recently introduced NetWeaver Business Client (NWBC).

There are several precautions required for the secure use of SAP GUI. This includes disabling the scripting API which can be abused to execute transactions and processes in the background. Another reason for disabling scripting is that scripts often store unencrypted logon data in local files. Credentials can be read and used by attackers if a workstation is compromised. Automatic security warnings should be enabled for users that require the use of SAP GUI scripting. This will alert users when a script is executed.

Security rules should be configured to prevent the ability of attackers to exploit SAP shortcut commands. Similar to scripting, such commands can enable attackers to interact with SAP servers without the knowledge of the user.

Input history should be disabled. Although SAP GUI does not store data entered in password fields, it can be configured to store data keyed by users in other fields. This may include sensitive customer, financial or other information. The data is stored in local Access databases.

The communication path between SAP GUI and application servers is not encrypted by default. Therefore, data transfers between clients and servers are in clear text. SNC (Secure Network Communication) should be used to secure the path. This is a software component that applies symmetric encryption algorithms for DIAG and RFC protocols through the GSS-API V2 interface to external security products. SAP Note 1643878 provides instructions for enabling SNC in SAP GUI 7.20, Patch Level 7 and higher.

Earlier versions of SAP GUI are vulnerable to buffer overflow exploits that can lead to the injection of malicious code designed to corrupt SAP programs and processes. Solutions include either applying the relevant patches released by SAP or upgrading SAP GUI to the latest available version. Program files in SAP GUI 7.20 and higher are digitally signed by SAP. This protects sensitive files against unauthorized modification.

Later versions of SAP GUI also feature a security module to protect the local environments of users. The module leverages rules to control potentially dangerous or malicious actions triggered by back-end systems related to specific files, extensions, directories, registry keys and values, ActiveX controls and command lines. Rules should be configured and applied centrally but can vary by user or system groups. They can also be context-dependant. The rules are employed when the module is configured in 'Customized' mode and are applied in sequential order. Therefore, higher order rules take precedence over lower rules. The default response for actions not defined in the rules should be set to 'Ask' or 'Deny' rather than 'Allow'.

SAP NetWeaver
SAP GUI FOR WINDOWS

720 Final Release | Version 7200.3.10.1074

Copyright © SAP AG 1993-2013

SAP GUI for HTML provides a thin alternative to thick desktop clients. This provides a browser-based platform for SAP access. Browsers also support the presentation layer for NWBC for HTML, the CRM WebClient UI and Java applications such as the Enterprise Portal. Supported browsers include specific versions of IE, Firefox, Safari and Chrome. However, certain browsers are not supported by some SAP applications.

Since browsers provide varying levels of protection, the ability to access SAP resources should be restricted to specific types. This can be performed by deploying standardized desktop images that include only approved browsers, supported by group policy rules that restrict end user privileges to install executable programs. Controls can also be implemented at the SAP level. For example, the Enterprise Portal can support browser-checking to block connection attempts from unsupported browsers.

Microsoft Internet Explorer offers the greatest protection for SAP access. The ability to configure trusted zones provides a seamless user experience while safeguarding against malicious applets, scripts and downloadable content from untrusted sites. The use of Firefox should be avoided, wherever possible. Weaknesses in the existing architecture of the browser can enable vulnerabilities in add-ons to go undetected by anti-virus solutions.

Web browser security should be supported with Web content filtering, anti-virus and anti-spyware, two-factor authentication for remote access, as well as regular patching of browsers and operating systems. Personal firewalls can be enabled for added protection, including stateful inspection firewalls available in some Windows operating systems. However, firewall rules should be thoroughly tested to ensure they do not inadvertently block access to SAP and other business applications. Operating systems should be hardened in line with security recommendations issued by vendors or in accordance with generally-accepted configuration benchmarks. For Windows systems, hardening should include enabling file protection, strong password policies, account lockouts, roles-based access based on least privilege, and disabling services such as FTP, Messenger, Remote Desktop Sharing and Telnet.

Basic authentication should be avoided for HTTP connections since it does not sufficiently protect user credentials during transport between clients and SAP servers. Also, it is susceptible to phishing attacks since servers are not authenticated. Phishing involves the redirection of users to malicious servers with logon screens that appear identical to those of legitimate SAP servers. User credentials entered into malicious servers can be used to compromise SAP systems. As a result, SAP strongly recommends the use of SSL/ HTTPS to secure basic authentication. This encrypts client-server communication and authenticates SAP servers. SSL requires the configuration of digital certificates which can be obtained from SAP Trust Center Services

SSL also protects SAP logon tickets used for single sign-on. These are authentication tickets stored as non-persistent cookies in browsers. However, this does not include safeguards against cross-site scripting attacks that attempt to read cookies through the execution of client-side scripts. This requires alternative counter-measures including the configuration of cookies as HTTP-only. The parameter setting ume.logon.httponlycookie=true will prevent malicious attempts to read SAP logon tickets.

LAYER SEVEN SECURITY

Layer Seven Security empowers organisations to realize the potential of SAP systems. We serve customers worldwide to secure systems from cyber threats. We take an integrated approach to build layered controls for defense in depth.

CONTACT US

Westbury Corporate Centre
2275 Upper Middle Road East, Suite 101
Oakville, Ontario, L6H 0C3, Canada
Tel. (Toll Free): 1 888 995 0993
Tel. (Office): 905 491 6950
Fax.: 905 491 6801
E-mail: info@layersevensecurity.com
www.layersevensecurity.com

SAP Partner