



LAYER SEVEN SECURITY

PROTECTING SAP® SYSTEMS FROM CYBER ATTACK

A SECURITY FRAMEWORK FOR
ADVANCED THREATS

WHITE PAPER

Version 4.0
February 2019

© Copyright Layer Seven Security 2019 - All rights reserved.

No portion of this document may be reproduced in whole or in part without the prior written permission of Layer Seven Security.

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.

This publication contains references to the products of SAP AG. SAP, R/3, xApps, xApp, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius and other Business Objects products and services mentioned herein are trademarks or registered trademarks of Business Objects in the United States and/or other countries.

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.



PROTECTING SAP SYSTEMS FROM CYBER ATTACK

A SECURITY FRAMEWORK FOR ADVANCED THREATS

CONTENTS

INTRODUCTION	SAP UNDER ATTACK	2
SECTION 1	SECURE THE NETWORK	5
SECTION 2	PROTECT REMOTE FUNCTION CALLS	13
SECTION 3	CONTROL ACCESS TO BASIS FUNCTIONS	18
SECTION 4	MAINTAIN LOG INFORMATION	22
SECTION 5	MANAGE THE CONFIGURATION	27
CONCLUSION	TRANSFORMING THE SECURITY BASELINE	33
APPENDIX	A ROADMAP FOR CYBER SECURITY IN SAP SYSTEMS	34

SAP systems are in the cross hair of cyber attackers. The first signs of the wave of attacks targeting SAP systems appeared in 2013 with the discovery of a variant of a widespread Trojan that had been modified to search for SAP clients. The reconnaissance performed by the Carperp Trojan was widely regarded by security experts as a preliminary phase of a planned attack against SAP systems. Carperp was capable of logging keystrokes and capturing screenshots which could lead to the theft of user credentials and other sensitive information related to SAP systems. It was also capable of attacking SAP servers through instructions received from remote command and control servers.¹

A year after the discovery of Carperp, a forensic review of the devastating data breach suffered by US Investigation Services (USIS), the largest provider of background checks for the U.S Federal government, confirmed that the initial breach was mostly likely caused by a vulnerability in an SAP system connected to the organization's internal systems. USIS lost \$3 billion in government contracts, laid off 2500 employees and filed for bankruptcy as a direct result of the breach.

Along with similar incidents experienced by the Greek Ministry of Finance, Nvidia, and Sony, the breach at USIS served to illustrate the devastating impact to organizations when SAP systems are not securely configured and monitored to protect against cyber attacks.

The security breaches experienced by such organizations were not isolated incidents. This was confirmed by a study performed by the Ponemon Institute in 2016 which revealed 65 percent of SAP customers had suffered security breaches in their SAP systems between 2015-16. According to the study, the average cost of an SAP breach was \$4.5M. 92 percent of organizations rated the impact of a breach within their SAP systems as serious or catastrophic.²

The United States Computer Emergency Readiness Team (US-CERT) issued a critical alert for a vulnerability in SAP systems following the release of the Ponemon study. The alert related to the invoker servlet vulnerability in NetWeaver AS Java systems. According to US-CERT, there was evidence that at least 36 organizations worldwide were effected by the vulnerability which enabled attackers to bypass authentication and assume complete control of SAP systems. Since the invoker servlet vulnerability was originally patched by SAP in 2010, the alert emphasized the importance of effectively patching SAP systems.³ US-CERT issued another warning in 2018 based on a report citing evidence of malicious cyber activity targeting ERP applications.

Enterprise applications developed by SAP are deployed by over 85 percent of Forbes 500 companies and often lay at the heart of information technology eco-systems, powering mission-critical processes and managing large volumes of sensitive data. SAP applications are therefore a prized target for cyber attackers.

Securing SAP systems against advanced cyber threats requires preventative and monitoring countermeasures across a broad range of areas. This paper presents a control

¹ Carperp-Based Trojan Attacking SAP, Microsoft Malware Protection Center, 2013

² Uncovering the Risks of SAP Cyber Breaches, Ponemon Institute, 2016

³ Alert TA16-132A – Exploitation of SAP Business Applications, US-CERT, 2016

framework to safeguard SAP components from known attack vectors that could be employed by malicious groups to perpetrate fraud, espionage and sabotage against SAP systems. The framework advocates twenty specific controls grouped into five control objectives (Figure 1).

CONTROL OBJECTIVE	CONTROL
Secure the Network	Configure Network Zones
	Filter Network Access
	Encrypt Network Communications
	Reduce the Attack Surface
Protect Remote Function Calls	Secure the Gateway Server
	Manage RFC Destinations
Control Access to Basis Functions	Manage Standard Users and Profiles
	Restrict Access to Authorization, Role and User Administration
	Restrict Access to System Administration
	Restrict Access to Transport Management
Maintain Log Information	Log Network Activity
	Log System Events
	Log System Changes
	Log Table Changes
	Log Document Changes
	Log User Actions
Manage the Configuration	Manage Authentication Parameters
	Monitor the System Configuration
	Apply Software Patches

Figure 1: Cybersecurity Framework for SAP Systems

The control objectives and corresponding controls are presented in detail within the white paper. Section 1 provides directions for implementing network-level controls to securely architecture SAP landscapes, filter access, encrypt communications and reduce attack surfaces. Section 2 outlines measures to protect the gateway server and configure RFC destinations to secure the most common communication protocol in SAP systems. Section 3 identifies the standard users, roles and privileges that could be abused to perform unauthorized administrative commands. Section 4 defines the multiple logs that should be enabled to support monitoring programs and forensic investigations. Finally, section 5 provides detailed recommendations for securing mechanisms used to authenticate users and front end clients. The section also provides recommendations for monitoring security settings and effectively patching SAP systems.

The framework is focused exclusively upon the SAP layer. Therefore, it excludes database, OS and endpoint controls required to secure SAP landscapes through defense in depth. Customers should ensure that such components are secured in accordance with vendor recommendations and generally-accepted security benchmarks. Please refer to the earlier white paper from Layer Seven Security *Defense in Depth: An Integrated Strategy for SAP Security*.

The framework also excludes measures related to development procedures for custom ABAP and Java programs. Organizations 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 using tools such as the SAP Code Vulnerability Analyzer.

Finally, the framework does not provide any specific guidelines for SAP HANA® platforms. For detailed guidance on securing HANA systems, please refer to the white paper *Security in SAP HANA* from Layer Seven Security.

The recommendations in this paper represent the best possible safeguards available in standard SAP components and do not require the licensing of additional SAP or third party software tools. Taken together, the controls embodied in the framework support the integrity, confidentiality and availability of information in SAP systems and greatly lower the risk of a successful data breach. Furthermore, the advanced forensic and monitoring capabilities recommended by the framework will greatly reduce the damage potential of attacks that exploit misconfigurations to access system resources.

The twenty controls across the five control objectives in the framework are implemented through eighty distinct actions. The required actions are reviewed in each section and listed in the appendix. Estimates are provided for the implementation of each action in typical SAP installations covering complexity, resource requirements and duration to support remediation efforts.

Layer Seven Security
New York, 2019

CONFIGURE NETWORK ZONES

A secure network architecture and configuration can prevent potential intruders from accessing and exploiting vulnerabilities within SAP systems. With the exception of certain implementation scenarios of the SAP HANA® platform, SAP systems adhere to a three tier client-server framework. The application, database and presentation components within the framework should be located in distinct network zones. Zone-based security should also be applied to segment and protect critical internal-facing SAP resources that store or process sensitive information from direct access.

The components of an SAP landscape should be categorized and grouped by function, purpose, access type or other factors. A separate Local Area Network (LAN) or Virtual LAN (VLAN) segment should be allocated for each group to create isolated and self-contained network zones. Each zone should be assigned a specific security classification based on the class of resources located within the zone. Finally, appropriate rules and policies should be configured using firewall systems to control communications between network zones based on security classifications. Default settings for Access Control Lists (ACLs) in most firewalls block traffic from lower to higher security zones. For example, incoming requests from an untrusted demilitarized zone (DMZ) to a trusted LAN are denied, whereas outgoing requests from the LAN to the DMZ are permitted. Firewall systems will vary the level of scrutiny applied to communication traffic based on the security classification of network zones. In virtualized environments, services can be located on the same physical hosts but separated as individual guest systems.

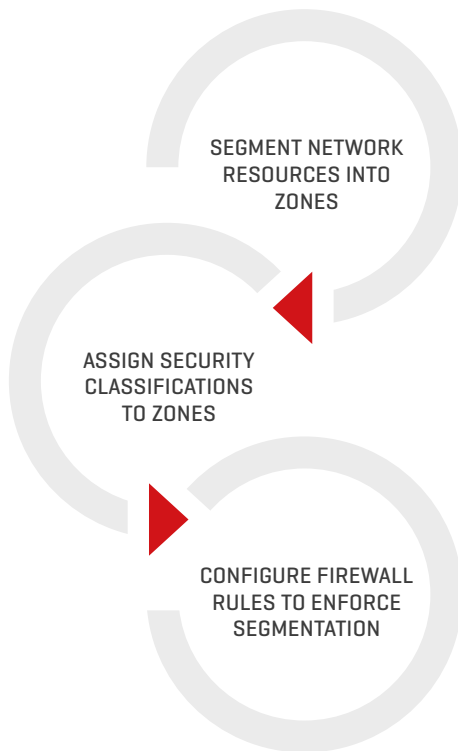


Figure 1.1: Implementing Zone-Based Security in SAP Landscapes

Network segmentation cannot be effectively applied in landscapes containing multi-purpose servers. Therefore, servers must be single-purpose and competing functions such as application and database hosting should not be performed within the same physical or virtual server. This approach will support a layered defence strategy for network resources.

The network topology for each SAP landscape including required zones should be tied to the specific systems operating within the network and the methods available to end users to connect to the components within the network. Figure 1.2 outlines a topology for a scenario that includes an external-facing Enterprise Portal and Mobile server. Therefore, it includes multiple network segments separated by internal and external firewalls. The Enterprise Portal and Mobile server are located in an inner-DMZ with no direct access to the LAN containing the back-end application and database systems.

Network firewalls typically only function between the physical and transport layers and filter traffic based on destination IPs and ports. Therefore, firewalls should be augmented with application-level gateways capable of enforcing more granular security checks within host layers. Unlike network firewalls, application gateways perform deep inspection of data packets to filter traffic based on user, source network zone, source address and other criteria. They also provide advanced logging functions to record and monitor network traffic. Client-server and server-to-server communica-

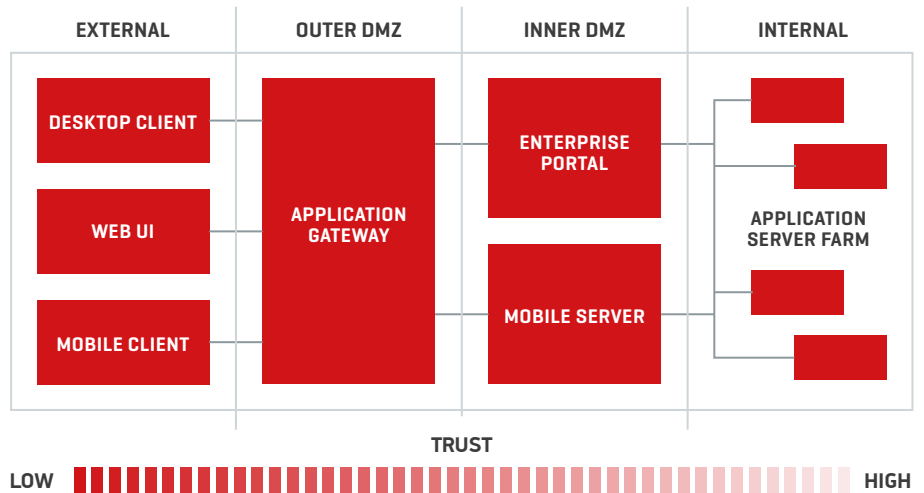


Figure 1.2: Network Topology for SAP Landscapes

tion for SAP systems should be filtered using the SAProuter and Web Dispatcher application gateways.

The SAProuter should be installed on the firewall host to filter traffic based on the SAP Protocol using the NI interface. The default listening port for the gateway is 3299. Therefore, firewalls should route all incoming SAP traffic to port 3299. The SAProuter will reroute the requests to SAP application servers based on information provided in route strings. Route strings contain multiple sub-strings for each predecessor and successor in connection threads. Permitted hosts and port numbers should be defined in the route permission table of each SAProuter. Connection strings in the table should not support insecure or non-SAP protocols. The KS and KT prefixes are therefore recommended for connection strings since this will only allow SNC connections using the NI protocol. The use of the P (Permit) prefix should be avoided since this can support native (non-SAP) connections. The D (Deny) prefix is redundant in most scenarios since the SAProuter will reject any connection that is not defined in the table.

Authorized connections including details of source hosts, destination hosts and destination ports are defined as follows for each entry:

```
KS / KT <source host> <dest host> <dest serv> <password>
```

Port ranges can be used for the <dest serv> field to minimize the number of required entries. Although the last field is optional, the use of passwords to secure SAProuter connections is recommended.

The insertion of a comment line starting with the # prefix at the end of an entry is recommended to record the rationale for the connection. Wildcards (*) for the target host and port fields should not be used with P and S entries.

The SAProuter performs a pivotal role in SAP landscapes by controlling connections to backend systems from untrusted networks. Therefore, it is often targeted by malicious attackers. Misconfigurations in the component may enable attackers to discover SAP systems through the use of administrative commands. Options `-I` and `-L` for example, can be used to display route information from the permission table including connected clients and IP addresses. The command `-H` will display route information to remote hosts. Other commands such as `-n` and `-X` can be abused to update permission tables and control the SAProuter from external hosts. Such attacks can be mitigated by avoiding rules that inadvertently enable connections from unauthorized destinations and regularly updating the SAProuter with the latest release. Changing the well-known default port using option `-S` can also defend against targeted attacks against the SAProuter.

Network filtering for HTTP traffic should be performed through the SAP Web Dispatcher. In common with the SAProuter, the Web Dispatcher is a software-based application-level gateway. However, it is installed on the host server connected directly to the Internet and can operate with both single-stack ABAP and dual-stack ABAP/ Java systems. The Web Dispatcher forwards incoming HTTP requests to an Internet Communication Manager (ICM) within back-end systems which are then forwarded to work processes within an application server. Responses follow the reverse path using the same network connection. However, HTTP connections initiated by an application server do not route through the Web Dispatcher. Therefore, the Web Dispatcher is akin to a reverse proxy rather than a proxy server.

URL filters can be configured in the Web Dispatcher to prevent external users executing specific programs. This is configured in a permission table referenced by the parameter `wdisp/permission_table = <ptabfile>`. However, SAP recommends an alternative approach that leverages the authentication handler function of the Web Dispatcher.⁴ A range of handlers are used by the Web Dispatcher to process HTTP requests. This includes an authentication subhandler to perform authorization checks for requested pages. The parameter `icm/HTTP/auth_<xx>` can be enabled to set up access restrictions in the Web Dispatcher. This activates filtering of HTTP requests using a wide set of criteria before the request is forwarded by the authentication sub-handler to downstream handlers. Filtering criteria includes not only URLs, but client and server IPs and user names/ groups. The parameter is enabled through the following syntax:

```
icm/HTTP/auth_<xx> = PREFIX=<URL-Präfix> ,PERMFILE=<permission file>,  
AUTHFILE=<authentication file>, FILTER=<name>
```

AUTHFILE contains the usernames, password hashes and user groups of authorized users. The file can be maintained through the command line programs `icmon` and `wdispmon`. Filters are defined in the PERMFILE using the syntax `P/D/S <URI pattern> <USER> <GROUP> <CLIENT-IP> <SERVER-IP>`. For illustration, the following filter permits HTTP connections to the specified web service at the server located at 10.0.30.91 from any user belonging to any group from the client at 10.0.21.60:

```
P /sap/bc/srt/IDoc * * 10.0.21.60 10.0.30.91
```

⁴ https://help.sap.com/saphelp_nw70/helpdata/en/7a/f2883c18be411ae10000000a114084/content.htm

Error messages are returned to clients for blocked requests if URLs are filtered by black or white lists configured in the Web Dispatcher. The parameter `is/HTTP/show_detailed_errors` should be set to `FALSE` to prevent the disclosure of sensitive information in default messages. Alternatively, custom static or dynamic error pages should be created and mapped using the parameter `icm/HTTP/error_tmpl_path =`.

Access to the Web Admin Interface used to manage and monitor the Web Dispatcher should be restricted to secure protocols and internal hosts and clients. This is performed through the appropriate configuration of the `PORT` and `CLIENTHOST` options of the parameter `icm/HTTP/admin_<xx>`.

Reverse invoke can be used to shield internal servers from external access and is recommended for high integrity environments. Since connections are initiated by ICMs in back-end servers rather than systems in the DMZ, firewalls are able to block all external access from a DMZ to a secure LAN. This requires the configuration of a registration port on application gateways to enable internal servers to respond to external connections requests. Reverse invoke can be enabled for the SAP Protocol through client and server-side parameters of the SAProuter configuration file. For Web-based connections, a range of parameters must be configured in the ICM and Web Dispatcher including `wdisp/reverse_invoke`, `wdisp/ri/client_serv` and `wdisp/ri/client_host`.

Specific parameters must also be configured in the message server, a system component responsible for managing communication and load balancing between application servers. The message server assumes a pivotal role in network communications and must be protected against spoofing and other attacks. Therefore, the `ms/monitor` profile parameter should be set to the value `0` to limit the privileges of external programs such as `msmon` used to change the internal memory of the message server and perform monitoring functions. Also, internal and external services should be configured on separate ports. External access to the internal port used by the message server should be denied. The port is specified by the parameter `rdisp/msserv_internal`. If external access to the internal message server port is required for business needs, access control lists can be configured to support access for specific hosts. The ACL file is referenced by the parameter `ms/acl_info`.

ENCRYPT NETWORK COMMUNICATIONS

Client-server and server-server communication in SAP systems is for the most part unauthenticated and in clear-text. Therefore, SAP data traffic is vulnerable to network sniffing, man-in-the-middle and other attacks capable of eavesdropping upon and tampering with the content of data packets containing business information transmitted between endpoints. There are several plugins available for open source network analysis tools such as Wireshark to decompress and dissect protocols supported by SAP systems. Although they cannot read SAP passwords which are obfuscated by default, attackers can use a variety of widely-available programs to break the algorithms used to encode such passwords.

PARAMETER	VALUE
snc/enable	1
snc/mode	1
snc/data_protection/min	3
snc/data_protection/max	3
snc/data_protection/use	3
snc/accept_insecure_gui	0 or U
snc/accept_insecure_rfc	0 or U
snc/accept_insecure_cplic	0 or U
snc/r3int_rfc_secure	1
snc/r3intrfc_qop	3,8 or 9
snc/permit_insecure_start	0

Figure 1.3: SNC Profile Parameters

As a result, SAP network connections should be protected through the application of transport layer security in the form of Secure Network Communications (SNC) for the SAP Protocol and TLS (HTTPS) for Web-based communications. This will support data confidentiality and integrity through message encryption and mutual authentication.

SNC should be enabled to secure dialog, CPIC and RFC connections used by AS ABAP, SAPGUI, SAProuter and SAPIpd. The latter is a transfer program used to forward print requests from SAP servers to host spools in Microsoft Windows systems. SNC operates below the application layer by connecting to an SAP or external security product through the GSS-API V2 interface for message encryption and decryption. The SAP Cryptographic Library (SAP Cryptolib) can be used to support encryption for server-to-server communication. Client-Server communication can be encrypted using the SNC Client Encryption add-on for SAP GUI and the Secure Login Library. Both programs are available at the SAP Software Download Center (refer to Note 1643878).

SNC offers three levels of protection: Authentication only, Integrity protection, and Privacy protection. The first option provides the least amount of protection since it only performs mutual authentication. The second enables the detection of data-level changes during transmission between endpoints. The third is the most preferred option since it enables both message encryption and mutual authentication. Consequently, the SNC parameter SNC_QOP used to configure protection levels should be set to 9.

Other SNC parameters should also be appropriately configured. The SNC_MODE parameter must be set to 1 to enable SNC. The initiators and acceptors of SNC connections should be specified in the SNC_MYNAME and SNC_PARTNERNAME parameters, respectively. Finally, the SNC_LIB parameter should specify the path and filename of the security product's library.

At the profile level, the recommended parameter settings are specified in Figure 1.3.

In order to minimize the performance impact of SNC, SAP does not recommend encrypting internal communications between application servers.¹⁰ According to SAP, the recommended parameter setting for snc/r3int_rfc_secure is therefore 0. However, performance considerations should be weighed against the security implications of unprotected traffic between application servers. Organisations that do not apply SNC in such scenarios rely exclusively on network zoning and are therefore more vulnerable in the event of a network breach.

SNC should be used to secure communications between SAProuters that traverse untrusted networks. This requires establishing the SNC environment in each SAProuter through the SNC_LIB variable and the -K <snc_name> option and configuring KT, KP, KD and KS entries in route permission tables.

TLS is the de-facto standard for securing Internet-based communications and operates between the Hypertext Transfer Protocol (HTTP) and Transport Control Protocol (TCP) within layers 5 (session) and 6 (presentation) of the OSI model. It commonly leverages a public and private key encryption system developed by RSA. This includes the use of digital certificates for authentication. TLS replaces all versions of SSL which have been deprecated due to security weaknesses.

¹⁰ https://help.sap.com/saphelp_nwpi71/helpdata/en/23/3a91f8d1724bc6b9e693eb735bcf2f/content.htm

TLS should be used to secure the communication path between AS Java and server components, as well as intermediary servers such as Web proxies. Since the Java Connector (JCo) uses RFC to communicate with AS ABAP, this channel must be secured using SNC. This will secure connections from iViews and the user management engine (UME) to ABAP systems. However, connections between the UME and LDAP directories require TLS. Client and server connections should use TLS version 1.2 or higher due to known security weaknesses with earlier versions. This can be enforced using the `ssl/ciphersuites` and `ssl/client_ciphersuites` profile parameters. The parameter `sec/rsakeylengthdefault` should be used to define a secure byte length for PSE keys.

In common with SNC, the SAP Cryptolib can also be used to provide cryptographic functions such as digital signatures for server-to-server connections using TLS. Therefore, key pairs can be exported from ABAP servers to the J2EE Engine in dual-stack systems providing the server components use the same host name. Key pairs are required to establish TLS connections in AS Java. The public key must be certified by a recognised Certification Authority (CA) and distributed using a X.509 certificate. The use of self-signed certificates is not recommended.

Aside from the installation of the SAP Cryptolib and the generation of key pairs, the successful configuration of TLS in AS Java also requires maintenance of specific ICM parameters. This includes `icm/server_port_<xx>` which should specify the protocol and port information. A non-standard port for HTTPS can be configured for enhanced security. The parameter `icm/HTTPS/verify_client` should be set to 2 to ensure that the ICM demands client certificates to establish a connection. The default value (1) enables clients to logon through other methods if they are unable to provide a valid certificate. Note that the setting in `icm/HTTPS/verify_client` can be overridden by the option `VCLIENT` in the `icm/server_port_<xx>` parameter. Therefore, the value in `VCLIENT` should match the value in `icm/HTTPS/verify_client`.

For Single Sign-On (SSO), the profile parameter `login/ticket_only_by_https` should be set to 1 to ensure that logon tickets are not transmitted in clear text.

The SAP Web Dispatcher should be configured to support TLS termination to optimize load balancing and support the filtering of connection requests. However, connections should be re-encrypted before they are forwarded to application servers. Therefore, the value of the parameter `wdisp/ssl_encrypt` should be 1 for HTTPS requests and 2 for HTTP, rather than 0 (termination without re-encryption). The SAP Cryptolib and root certificates should be installed on the server hosting the Web Dispatcher to support encryption and decryption functions. For end-to-end TLS, the value of the protocol (`PROT`) option in the `icm/server_port_<xx>` parameter must be set to `ROUTER`.

Metadata sent from the Message Server to the Web Dispatcher including information related to application servers, logon groups and URL prefixes should also be secured using TLS. This data is required by the Web Dispatcher for load distribution. The `ms/server_port_<xx>` parameter should specify the HTTPS protocol and the port used for the protocol.

REDUCE THE ATTACK SURFACE

The demand for rapid deployment, accessibility and interoperability has transformed the architecture of SAP systems. SAP's commitment to Service Orientated Architecture (SOA) has led to the development of a SAP NetWeaver® platform that supports a broad range of open languages, protocols and standards. As a result, SAP systems can present a wide attack surface to potential attackers if the functions available to remote users are not closely aligned to actual business needs. This increases both the risk of a successful compromise and the resources required to manage the array of interfaces, ports, and services available to external attackers. The risk can be mitigated by reducing the number of entry points into SAP systems through minimizing open network ports, removing unnecessary services and proactively managing custom code.

The network ports required by SAP systems are determined by the specific applications and components installed on each host and connectors required for databases and programs such as SAProuter. Therefore, customers should refer to the relevant SAP security guides and the port table in the SAP paper TCP IP Ports used by SAP Applications for precise instructions. There are several services that are widely regarded as insecure due to known vulnerabilities that should be disabled or, at the very least, accessible only to internal users through appropriate firewall rules. This includes FTP, NFS, NNTP, Telnet, NetBIOS, RPC and, in Unix systems, r* services such as rsh, rlogin and rexec.

External clients can access services in SAP systems directly through the Remote Function Call (RFC) interface or indirectly through the Internet Communication Framework (ICF). Portal iViews also provide an access path to back-end SAP functions. However, this particular route is secured through dual level authorizations within the Portal and AS ABAP or AS Java.

The RFC interface supports the calling of remote-enabled function modules (RFM) in external SAP systems. This is performed through the CALL FUNCTION statement with a DESTINATION parameter that specifies the target system. Destinations are maintained in table RFCDES using transaction SM59. Function modules are ABAP routines that are grouped in logical function groups. SAP delivers a large number of default function modules with standard installations, Examples include BAPI_USER_CREATE1, DEST_SET_PASSWORD, and TH_CHANGE_PARAMETER used for provisioning users, configuring RFC destinations, and changing system configurations. The volume increases further with custom function modules developed through the ABAP Workbench Function Builder.

RFC calls from an external system should trigger the target system to check authorization object S_RFC to ensure the user initiating the call has the appropriate permissions for the function group containing the relevant function module. This should be specified in the field RFC_NAME of the object. However, the check is only performed if the profile parameter auth/rfc_authority_check is set to 1. Furthermore, authorization checks are not consistently configured for all function groups. For example, checks are rarely performed for the SRFC function group which includes functions such as RFC_GET_LOCAL_DESTINATIONS, RFC_GET_LOCAL_SERVERS, RFC_SYSTEM_INFO and SYSTEM_INVISIBLE_GUI. Functions within such groups can be called remotely and anonymously by external attackers to perform reconnaissance against SAP systems prior to launching a targeted attack. Therefore, it is critical

to disable remote access to function modules when such access does not serve business requirements. This can be performed by logging RFC calls to function modules using the Security Audit Log (transactions SM19 and SM20) which would enable identification of the function modules accessed by external systems and users.

Given the size and complexity of RFC logs in high volume environments, the Security Audit Log is not always a workable solution. The Unified Connectivity (UCON) framework, available in NetWeaver AS 7.4 SP02 and higher, provides a quicker and simpler mechanism for controlling external access to RFMs. UCON enables RFMs to be assigned to a Communication Assembly (CA) accessible to external systems through RFC. This blocks external access to all other function modules that do not need to be remote-enabled. The framework introduces virtual hosts to divide a single RFC communication port into multiple virtual channels. Configuring UCON is a three-step process that includes phases for logging, evaluating and activating runtime checks for RFMs.

The current version of UCON supports only the RFC scenario. However, future releases will also support HTTP services in the Internet Communication Framework (ICF). The ICF is the second method that can be used by external systems and users to access SAP functions. HTTP, HTTPS and SMTP requests for ABAP environments are forwarded by the Internet Communication Manager (ICM) to the ICF. Therefore, the ICF enables external parties to execute ABAP processes in SAP systems from any location via the Internet.

ABAP functions are presented to external users as ICF services. Each service has a specific URL path mapped in nodes and subnodes within a service tree managed through transaction SICF. In common with remote function calls, ICF services should be secured through the authorization concept. This includes the authorization object S_ICF and authorizations defined in the SAP Authorization field under Service Data for each specific service. However, as with RFC, authorization checks are not consistently applied and, in some cases, not required to execute certain default ICF services and services with stored logon data. This includes services in the sap/public node.

```
/sap/bc/soap/*  
/sap/bc/echo  
/sap/bc/FormToRfc  
/sap/bc/report  
/sap/bc/xrfc  
/sap/bc/xrfc_test  
/sap/bc/error  
/sap/bc/webRFC  
/sap/bc/bsp/sap/certreq  
/sap/bc/bsp/sap/certmap  
/sap/bc/gui/sap/its/CERTREQ  
/sap/bc/gui/sap/its/CERTMAP  
/sap/bc/bsp/sap/bsp_veri  
/sap/bc/IDoc_XML  
/sap/bc/srt/IDoc
```

Figure 1.4: Vulnerable ICF Services

There are several critical vulnerabilities associated with the standard ICF services listed in Figure 1.4 when exposed to external systems and users. The service /sap/bc/soap/rfc, for example, can be exploited to acquire shell access to SAP systems through the ability to call RFMs using HTTP requests. Therefore, these services should be deactivated using SICF. The attack surface in the ICF can be further reduced by deactivating services that are not called by external systems and users. These services can be identified through review of the HTTP Log in the ICM accessible through transaction SMICM. Alternatively, refer to the instructions in Note 1498575 for HTTP tracing and mass deactivation of ICF services.

Aside from minimizing the number of entry points, surface area reduction should also involve reducing the quantity of software code operating within an environment. This can be achieved by enabling only the required standard components delivered by SAP. For custom code, idle code that is not used for productive purposes should be identified using Coverage Analyzer in Solution Manager (transaction SCOV). Coverage Analyzer can be used to track the usage of custom programs, function groups and classes in connected systems. Once enabled, SCOV records relevant statistics in tables that include COVRES and COVREF. These tables are read by functions such as Global Display and Detail Display. The specific branches and statements within custom objects that were not executed during a recording period can be detected using source code display in the Branch and Statement Coverage Screen. Lines items highlighted in red may be redundant and therefore should be removed to lower the attack surface.

SECURE THE GATEWAY SERVER

The gateway is a kernel-level component that is distinct from the SAP NetWeaver Gateway. The former is responsible for performing RFC communications between SAP and non-SAP systems, whereas the latter is a middleware technology that provides connectivity to SAP applications through REST services and protocols such as Open Data. Since RFC is the most common protocol used by SAP systems, securing access to the gateway is a central component of the control framework presented in this paper.

The gateway is installed in each instance of ABAP-only and dual-stack systems. In Java systems, a single gateway is used to support RFC connections to other systems through the JCo. It is started by the ABAP Dispatcher and enables RFC communication between work processes in different instances in the same system or between work processes and external programs and systems. Connection requests are received and processed by the gateway read process (gwr.exe). Connections are monitored and administered through the gateway monitor, accessible through transaction SMGW or, at the operating system level, via the program gwmon.exe. The gateway performs load balancing for registered programs. The load balancing method defined for the gateway is determined by the parameter gw/reg_lb_level.

The gateway can be started from remote hosts using remote shell (rsh) or secure shell (ssh). Rsh does not encrypt information transmitted through the protocol including passwords. Therefore, the default setting for the parameter gw/rem_start should be changed to SSH_SHELL. The gateway should only accept commands from local gateway monitors through the parameter and value gw/monitor = 1. Furthermore, since trace files may disclose sensitive information, receiving systems should be configured to reject trace levels set by a remote gateway. This can be performed by setting the parameters gw/accept_remote_trace_level and rdisp/accept_remote_trace_level to the value 0.

There are several external server programs installed on SAP application server hosts that can be launched by specific client-side RFC requests. Once triggered, these programs could enable remote users to perform unrestricted operating system commands on an SAP server without authentication. Such commands could be combined with SQL statements to create SAP users directly within databases with SAP_ALL privileges or extract usernames and password hashes from SAP tables. Such scenarios could lead to the complete compromise of an SAP system by abusing the implicit trust relationship between an operating system and database. Therefore, it is critical to restrict external access to RFC server programs such as RFCEXEC and SAPXPG through operating system security files. In the case of RFCEXEC, this can be performed through logon handlers that check filters configured in the file rfcexec.sec. However, the preferred method is to use access control lists (ACL) through sec_info. This file can be used to prevent unauthorized access to all external programs, not just RFCEXEC.

Sec_info is created for each application server. However, it is possible to share a single file for all servers within an SAP system through a common working directory. The file path is set through the parameter gw/sec_info. The file can be maintained either through SAP using transaction SMGW or directly within the operating system. There are two methods that can be used for the file syntax. The most common method (version #1) follows the format below.

```
TP=<tp>, USER=<user>, HOST=<host>, [USER-HOST=<user_host>]
```

Each line is used to permit access to programs <tp> or hosts <host> for authorized users <user> connecting from specific hosts (<user_host>). The user-host parameter is optional. This format follows a whitelist approach. Therefore, connections that are not recorded in the file are automatically denied. The second method for the file syntax (version #2) can be used to configure access blacklists.

An ACL should also be used to control the registration of external programs including RFC servers in the gateway. This should be performed through the reg_info file. The file path is set through the parameter gw/reg_info. The file can also be maintained either through SMGW or the OS.

The required entries for the sec_info and reg_info files can be obtained from gateway log or trace files before enabling filtering. Alternatively, a restrictive policy can be applied from the outset and fine-tuned by identifying rejected connection attempts after filtering is enabled. Such a policy should only support access from local and internal hosts using the suggested entries in Figure 2.1. Note that the bit mask in parameter gw/reg_no_conn_info must be configured correctly to prevent a known bypass of the gateway security files (refer to Notes 1444282, 1633982, 1697971 and 1298433)

```
Sec_info  
TP=*USER=* HOST=local USER-HOST=local  
TP=*USER=* HOST=internal USER-HOST=internal  
  
Reg_info  
TP=*, HOST=local  
TP=*, HOST=internal
```

Figure 2.1: Recommended settings in gateway security files

There are several important profile parameters that must be maintained if the gateway security files are not configured. The parameter gw/acl_file can be used to reference an ACL file containing permitted connections to the gateway, while gw/acl_mode can restrict registration and starting of external servers to application servers within the same system when set to the value 1.

MANAGE RFC DESTINATIONS

RFCS enable clients to invoke remote-enabled function modules that run on external servers. They are used for a variety of functions including data replication and user management. RFCS are synchronous when both the client and server are available when the call is made. This type of call is known as sRFC. Asynchronous or transactional RFCS (tRFC) do not require server systems to be available when calls are performed. tRFC calls are stored in SAP databases with a unique transaction ID and processed when the server is active or as part of a batch process. This method can be used to manage the performance impact of multiple concurrent RFC requests on servers. Queued RFCS (qRFC) are an extension of tRFC and are used to process calls in a specific sequence.

Destinations are used to specify the parameters of each RFC connection including the connection type, target system, client, and for untrusted connections, username and password. This is performed through transaction SM59. Destinations are stored in table RFCDES and are required for sRFC but are not mandatory for tRFC. The target function modules are invoked in the calling system when no destination is specified using tRFC.

The risks associated with cross-system communication using the RFC interface cannot be addressed by network-level security including zoning, firewalls and application gateways since RFC traffic cannot be adequately filtered. Therefore, RFC communication must be secured through the SAP authorization concept. This can be achieved by selecting the appropriate user types for RFC destinations, controlling RFC user authorizations and carefully configuring trusted RFC connections between suitable systems.

RFC destinations should leverage system users rather than communication, service and especially, dialog users. Communication users are able to change their passwords. Dialog users are able to logon and interact with SAP servers through clients such as SAPGUI. There are no such drawbacks with system users. The profile parameter `rfc/reject_expired_password` will enforce the use of system users for RFC connections when set to the value 1 since passwords for system users do not expire.

A unique user should be configured for each RFC destination in accordance with the principle of cardinality. This will allow authorisations to be tailored for each destination and limit the damage resulting from a compromise of an RFC user account. Attackers would not be able to exploit the credentials used for a connection to perform unrelated functions or access other systems.

RFC authorisations should be provisioned based on the principle of least privilege. RFC calls can be logged and analyzed in the Security Audit Log for an extended period of time to identify the authorisations required by RFC users based on the actual usage of function groups. However, this approach has several drawbacks: many function modules do not belong to a function group and the Security Audit Log does not record the actual function modules called by RFC users. Therefore, the preferred approach is to enable traces using transactions STAUTHTRACE, STRFCTRACE or STUSOBTRACE. Once enabled, STAUTHTRACE will display RFC function modules called by users and the authorization checks that were performed during the execution procedure. Furthermore, trace results are accessible in transaction SU24 for automatic role maintenance.

An authorisation check for the object S_ICF is performed in the calling system before a user is able to call a destination in an external SAP system. The use of wildcards (*) in the ICF_VALUE field of this object should be avoided since this would enable the user to access any destination in external systems. Rather, the field should state the name of the specific destination required by the user. Note that ICF_FIELD must be set to DEST in S_ICF.

On the server side, the authorisation object S_RFC is checked before the RFM requested by the system user configured in the destination is invoked. Full authorisations through the use of a wildcard in the field RFC_NAME would enable the user to call any RFM in the system. Therefore, this field should identify the relevant function group containing the RFM required by the destination. Since authorisation checks are performed at the function group level, RFMs should be categorized by function area or other logical grouping.

SAP provides switchable authorization checks to secure remote access to specific high-risk function modules that can be used to access or modify sensitive data. The checks are delivered through SAP Notes, support packages or enhancement packages. They are inactive by default but should be activated using the switchable authorization check framework (transaction SACF). Users that require the additional authorizations for the effected function modules can be identified through review of the DUO and DUQ event logs in the Security Audit Log. The information in the event logs should be used to edit roles before activating the checks to avoid any potential business disruption.

The DUI and DUJ event logs should also be reviewed in the Security Audit Log. The logs record successful and unsuccessful RFC callbacks executed in systems. RFC callbacks enable servers to open RFC connections in clients during synchronous calls using the privileges of the RFC user in the client system. This can pose a serious security risk, especially when destinations are configured with privileged users and the callback establishes a connection from a system with a lower trust level. The information logged in the DUI and DUJ event logs should be used to build whitelists for permitted callbacks. The profile parameter rfc_callback_security_method should be set to 3 to activate whitelists maintained in transaction SM59 after an appropriate logging phase.

Trusted RFC connections between systems can provide greater security than untrusted connections by supporting the mapping of user authorisations between systems and removing the need to store logon credentials. However, trusted connections should only be configured between systems with the equivalent security classification or from higher order to lower order systems. For example, production-to-production or production-to-QA. Connections from lower order to higher order systems such as development-to-production can be exploited to perform privilege escalation and RFC hopping between a system landscape, as well as bypass controls in the SAP Transport Management System.

The authorisation S_RFACL is used to manage the risks of trusted RFC relationships. The object is used in conjunction with S_ICF and S_RFC. It performs a server-side check of the user logged in the client that is attempting to establish a trusted connection with a server system. The user transferred in the request from the client is checked against the user records within the server. The request is only

approved by the trusting system if the user has the required authorisations in the target server. There are several important fields in S_RFCACL that must be carefully maintained to support trusted connections. RFC_SYSID should be used to specify the system that is able to use the connection and RFC_CLIENT should state the client of the calling system. These fields should not contain full authorizations (i.e. the wildcard *). If specific users cannot be registered in the RFC_USER field due to the large volume of users that require access to the destination, the RFC_EQUUSER field should be set to Y to enforce a match between client and server users. However, if a single system user is assigned for the destination, RFC_USER can be used to specify the ID of the calling user. RFC_EQUUSER can be set to N to support user switching between client and server systems. The use of identical user IDs in both systems is recommended.

MANAGE STANDARD USERS AND PROFILES

Access to transactions, programs and services in SAP systems is controlled by the authorization concept. This ensures that users cannot perform any action without express permission assigned through authorization objects containing the required field values. There is a range of possible values for authorizations including 01 (create), 02 (change), 03 (display) and 06 (delete). Users are assigned authorizations with the required values through roles that contain multiple authorization profiles.

Authorisations provided to users are loaded into user buffers at the start of each session and checked before transactions, programs, RFCs or tables are returned in response to user requests. To ensure that checks are performed for all authorization objects, the profile parameter `auth/object_disabling_active` should be set to the value N. If the parameter is set to Y, the list of objects excluded from authorization checks should be closely reviewed through transaction SU25 or AUTH_SWITCH_OBJECTS.

USER ID	CLIENTS	PASSWORDS
SAP*	000, 001, 066	06071992
SAP*	New Clients	PASS
DDIC	000, 001	19920706
SAPCPIC	000, 001	ADMIN
EARLYWATCH	066	SUPPORT
TMSADM	000	PASSWORD

Figure 3.1: Standard SAP Users

The Basis authorizations outlined in this section provide users with powerful administrative permissions. They should be allocated cautiously and selectively to users based on business need. Some of these authorizations are included in roles assigned to standard users. Therefore, the ABAP users delivered by SAP in Figure 3.1 should be locked in all clients and default passwords should be changed. Default passwords should also be changed for users J2EE_ADMIN, J2EE_ADMIN_<SID>, J2EE_GUEST and J2EE_GUEST_<SID> in Java systems.

In accordance with SAP recommendations, all authorization profiles for the user SAP* should be deleted.⁵ The user should be assigned to an authorization group for super users and the parameter `login/no_automatic_user_sapstar` should be set to a value greater than 0 to prevent logons by SAP*. Authorisation groups should also be used to secure access to tables containing password hashes including USH02 and USRPWDHISTORY and custom tables storing sensitive data. Transaction SE54 can be used to create the groups and SE11 to assign tables to the newly created groups.

Since authorisations are generally conferred to users through transaction codes packaged into roles, the wildcard value (*) should not be used in the TCD field of object S_TCODE. This will enable users to call any transaction in any SAP system. The use of the SAP_ALL profile presents an even greater danger since it enables users to perform almost any task. Therefore, this profile should not be assigned to users. Another dangerous profile that should be used sparingly is SAP_NEW. Although the profile is now obsolete and replaced by the SAP_NEW role, it should not be assigned to users long after a system upgrade since it may provide privileges that are not required by end users.

The use of service and reference users should be avoided. Service users provide anonymous access to SAP systems and reference users interfere with the auditing of user privileges in systems.

RESTRICT ACCESS TO AUTHORIZATION, ROLE AND USER ADMINISTRATION

The authorisation object S_USER_GRP with activity values 01, 02, 05 or 06 are required to maintain users, add or remove profiles, change passwords and lock/unlock users. It is checked within transactions such as SU01, SU10, SU12 and ST14.

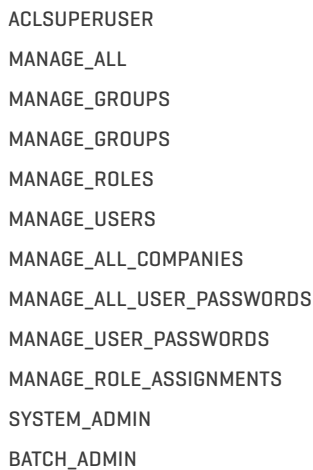
Other relevant authorisations include S_USER_AGR, S_USER_PRO and S_USER_ADM. These are used to protect roles, profiles and customizing. The authorisation S_USER_SYS is used to distribute users to child systems through Central User Administration (CUA). However, the activation of the authorisation S_USER_SAS replaces checks performed for this object, as well as S_USER_AGR, S_USER_PRO and S_USER_GRP.

The objects S_USER_AUT, S_USER_TCD and S_USER_VAL should be used to define the range of authorizations, transactions and field values administrators are permitted to maintain in order to segregate and protect sensitive permissions. It is also recommended to separate the creation and modification of roles from the assignment of roles to users. This can be performed by isolating the authorisation S_USER_AGR activity level 02 (change) from the authorisations S_USER_GROUP and S_USER_AGR activity level 22 (assign). As a prerequisite, the parameter ASSIGN_ROLE_AUTH must be set to ASSIGN rather than CHANGE. This parameter is located within the PRGN_CUST table containing customizing settings for the profile generator (Refer to Notes 312682 and 565108).

Authorization, role and user maintenance in AS Java are performed through the User Management Engine (UME). UME actions are the counterpart to ABAP authorisations. Default UME actions are listed in Figure 3.2. The permission AclSuperUser is Portal-specific and provides ownership privileges for all Portal content. It is included in the Super Administrator Portal role. The permission Manage_All provides access to all UME functions including group, role and user administration, user mapping with external stores such as LDAP and AS ABAP, importing and exporting of user data, and UME configuration. It is incorporated into both the Super Administrator and Administrator roles. Hence, such roles should be assigned restrictively to selective administrators.

RESTRICT ACCESS TO SYSTEM ADMINISTRATION

The object S_ADMI_FCD is used to authorise a wide array of important administrative tasks that could be abused to perform malicious actions in SAP systems. This includes starting and stopping work processes, administering the ICM, activating ICF services, viewing OS files and server caches, locking and unlocking transactions, and displaying, redirecting or exporting printer spool requests. It also used to manage Personal Security Environments (PSE) in SAP servers. The PSE is the store for public-key certificates, private address books and private keys required for SNC and TLS communication. S_ADMI_FCD should therefore be assigned selectively in combination with the relevant fields and values.



- ACLSUPERUSER
- MANAGE_ALL
- MANAGE_GROUPS
- MANAGE_GROUPS
- MANAGE_ROLES
- MANAGE_USERS
- MANAGE_ALL_COMPANIES
- MANAGE_ALL_USER_PASSWORDS
- MANAGE_USER_PASSWORDS
- MANAGE_ROLE_ASSIGNMENTS
- SYSTEM_ADMIN
- BATCH_ADMIN

Figure 3.2: Standard UME Actions

In common with S_ADMI_FCD, the object S_DATASET is also used to secure access to operating system files from ABAP programs containing commands such as OPEN DATASET, READ DATASET and TRANSFER. Full access to S_DATASET should be avoided. The authorisation should be qualified to enable access for only the required actions and files from specific programs. This is configured within the object fields.

Unauthorised access to batch operations including the release and deletion of scheduled jobs could have a significant impact on data integrity. Hence, the authorisations S_BDC_MONI, S_BTCH_ADM and S_BTCH_JOB should be closely guarded.

RFC destinations are maintained via transaction SM59 and requires authorization object S_RFC_ADM with values 01, 02, 03, 06 or 36. The latter value (36) enables users to perform extended maintenance including activating and displaying traces.

Trust relationships between SAP systems are configured using transaction SMT1. The field RFC_TT_TYP for authorisation S_RFC_TT is used to define whether users are able to maintain calling systems, called systems or both. The fields RFC_SYSID and RFC_INSTNR can be used to limit permissions for specific system IDs or installation numbers.

Other critical authorisations include S_LOG_COM used to execute OS commands, S_RZL_ADM, which enables users to register OS commands, maintain profiles and create/ delete clients, and S_PROGRAM, required to access SAP reports. Lesser-known but equally important are the objects S_TMS_ACT and S_NUMBER. These are used to control access to the TemSe store containing temporary data and maintain number ranges for business objects such as SAP documents.

RESTRICT ACCESS TO TABLE MAINTENANCE

Data tables are the repository of business information in SAP systems. The ability to browse or maintain data tables is one of the permissions most sought after by attackers since it can be abused to view or modify data elements stored within fields across multiple tables. This includes organisational, master, and transactional information, as well as SAP programs, function modules and transport data.

The authorisation object S_TABU_DIS is used to control access to data tables. Permitted values for the activity field of the object are 02 (change) and 03 (display). This object is checked during standard transactions such as SE16, SE17, SM30 and SM31. The authorisations S_TABU_CLI and S_TABU_LIN can be used together with S_TABU_DIS to restrict table access to specific clients and countries. The field DICBERCLS should be used to qualify access for specific authorization groups defined in table TDDAT. Full authorisations (*) within this field provides access to all data tables.

The majority of tables delivered by SAP are not assigned to any authorisation group and are categorized as unclassified (&NC&). Therefore, users with the S_TABU_DIS object could be able to view or modify any table that does not have an authorisation group assignment. In order to address the risk of unchecked table access, SAP provides the authorisation S_TABU_NAM to enable access control for individual tables. Required tables and activity levels should be specified in the TABNAME and ACTVT fields of the object for the relevant users and roles (refer to Notes 1481950 and 1500054).

VALUE	DESCRIPTION
CUST	Customizing requests
DTRA	Workbench requests
TASK	Tasks [repair or correction]
MOVE	Relocation transports [all three types]
TRAN	Transports of copies
PATC	Preliminary corrections and deliveries
PIEC	Piece list
CLCP	Client transports

Figure 3.2: S_TRANSPRT Request Types [TTYPE]

VALUE	DESCRIPTION
01	Add or create
02	Change
03	Display
05	Lock
06	Delete
23	Change in object list editor
43	Release
50	Change source client of a request
60	Import
65	Reorganize
75	Release other requests
78	Enter request in transport proposal
90	Change owner

Figure 3.3: S_TRANSPRT Activity Levels [ACTVT]

RFC destinations are stored in the RFCDES table which belongs to the SC authorization group. Therefore, it is important to control access to S_TABU_DIS with the value SC in field DICBERCLS. Relevant values include 02 (change) and 03 (display). This object is checked during standard table view and maintenance transactions such as SE16, SE16N, SE17, SM30 and SM31.

Metadata such as data types, definitions, attributes, domains, structures and relationships are maintained in the data dictionary. Since the dictionary is integrated in the ABAP Workbench, access to metadata is controlled by the development object S_DEVELOP (activity levels 01, 02 and 06).

RESTRICT ACCESS TO TRANSPORT MANAGEMENT

SAP landscapes consist of separate development, quality assurance and productive environments to regulate system changes that may impact the availability and integrity of systems. Changes are organized into and distributed through transports using the Transport Management System (TMS) or, more specifically, the transport program R3trans. The authorization object S_CTS_ADMI is required to move transports through the system landscape. The values IMPA, IMPS, INIT, QTEA and SYSC in the field CTS_ADMFCT enable users to import all or individual requests in import queues, configure the TMS, approve transports and set system change options.

Change requests are created and released through the Transport Organizer available through transactions SE01, SE09 and SE10. The Transport Organizer is most commonly accessed by developers and change managers and requires authorization object S_TRANSPRT. The fields and activity levels available within the object are specified in Tables 3.2 and 3.3. Users responsible for authorizing transports between systems should only require display access for S_TRANSPRT.

Transport management protocols should be underpinned by the removal of developer keys from the DEVACCESS table in production systems. Such systems should also be locked against changes using transaction SCC4. Furthermore, the use of the authorisation S_DEVELOP should be avoided in production, especially when combined with activity level 02 and the DEBUG value for the OBJTYPE field. This can be used to bypass authorization checks other than those performed at the kernel level. Finally, access to the objects B_LSMW* should be secured. Some of these authorisations enable users to call functions from external ABAP programs using the Legacy System Migration Workbench (LSMW).

Countering attacks against SAP systems requires proactive monitoring of comprehensive and up-to-date event logs covering network, system, table, document and user domains. This enables the detection of actions often associated with system intrusions and the blocking of attempted attacks in real-time. Effective logging and monitoring also acts as a deterrent against internal threats and enables organisations to quickly trace the origin of successful breaches, assess the impact and contain the damage resulting from attacks.

This section discusses the forensics capabilities of standard SAP components and provides specific recommendations for enabling logging across multiple areas to detect and respond to a variety of threats. In order to protect the integrity of log information, organisations should consider securing the transmission of log information, storing log files in an encrypted format and replicating files to separate time synchronized servers. Logs should also be periodically archived and maintained in accordance with relevant standards for data retention.

LOG NETWORK ACTIVITY

Network connections routed by the SAProuter are not logged in the default configuration and therefore should be enabled using the `-G` option. The path name of the log file must be specified with the option. Once enabled, SAProuter will write all actions to the log file with a timestamp. This includes permitted/ rejected requests, client/ server IPs and ports/ services. Logs should be reviewed to identify connection attempts using native protocols, info requests to the SAProuter and port-scanning attacks. The option `-J` should be used to set the size of the log file. A new file is automatically generated once the defined size is reached. The most recent log file is closed and renamed to `<logfile name>_a_<start date>_<start time>-<end date>_<end time>`.

Logging in the ICM and Web Dispatcher is configured using the parameter `icm/HTTP/logging_<xx>` for incoming requests and `icm/HTTP/logging_Client_<xx>` for outgoing requests. The former has the following syntax:

```
icm/HTTP/logging_<xx> = PREFIX=<URL prefix>, LOGFILE=<log file name> [,  
LOGFORMAT=<format>, FILTER=<filter>, MAXSIZEKB=<size in KBytes>,  
SWITCHTF=<options>, FILEWRAP=on]
```

LOGFILE is used to specify the location of the log and MAXSIZEKB is used to set to the maximum file size in kilobytes. The option FILEWRAP=on should not be included in the parameter. This will reset and overwrite the log file when the maximum size is reached. SWITCHTF can be used to open a new log file by the hour, day or month. There are several predefined formats available for ICM logs. The SAPSMD and SAPSMD2 formats can be used to track HTTP requests. Logging procedures will not record sensitive URL parameters, header fields, cookies and form fields. This includes `jessionid`, `sap-contextid` and `sap-password`. The User-Defined Format enables organisations to create custom log formats by selecting from a wide range of log properties

including HTTP request paths, source and destination hosts, IP address, and port names or services. Logging can be limited to specific header fields using the FILTER property of the icm/HTTP/logging_<xx> parameter. Note that log files do not contain entries in the buffer that have not been written to disk. The option Write to Buffer can be used to manually update files through the HTTP Log of the ICM Monitor.

A secondary source of logs for HTTP requests can be provided by the message server, a standard component used to provision information to application servers and balance loads for dialog and RFC requests. This is enabled through the parameter ms/http_logging=1. The log format is defined using ms/HTTP/logging_o. The message server uses the same syntax as the ICM. SAPMSG is the default format.

LOG SYSTEM EVENTS

The SAP system log provides a mechanism to track system-level changes and events that may be initiated by attackers. This includes errors, messages and warnings that can be read using transaction SM21. The system log generates both local and central logs in UNIX systems but only local logs on Windows and AS/400 platforms. Local logs are generally more up-to-date than central logs since logs are not synchronized in real-time. Local logs are also activated by default, whereas central logs must be initiated through system parameters beginning with the rslg prefix. Log paths are set through the parameter rslg/local/file for local logs and rslg/central/file for central logs. Since local logs are overwritten when they reach the maximum permissible length specified in the rslg/max_diskspace/local parameter, it is recommended to schedule a regular background job for the program RSLG0000. A scheduled task should also be created for the central logs program RSLG0001. Although the system log performs a log file switch for active files, the process overwrites the old log file referenced by the rslg/central/old_file parameter. Therefore, central log files are not automatically retained or archived by the system log.

The logging of logons and logoffs by application servers to message servers should be enabled by setting the ms/audit parameter to value 1 or 2, especially if external clients are able to register application servers and control the internal memory of a message server. This can be inadvertently enabled by misconfiguring the ms/monitor parameter, supporting unauthorized connections with missing or insecure ACLs, or failing to configure a separate port for internal communications.

The logging of network connections opened and closed by the SAP gateway, as well as other actions such as monitor and operating system commands, starting of external programs, RFC transmissions, server registration and changes to dynamic parameters, should be enabled through the gw/logging profile parameter. Timestamp variables should be assigned to file names to ensure that logs are not overwritten when file sizes are exceeded. This will ensure that logs are retained in the specified directory.

Trace functions can also be used to record detailed information for a range of system events. Transaction ST01 is used to create and view system traces for authorization checks, kernel functions, table buffers and other actions. Developer traces configured using SM50, SMGW, SMMS and SMICM register operations performed through the gateway, message server, RFC, SAPGUI and transport programs, among other areas. These traces can be displayed using transaction ST11 or through the operating system.

Trace files support the reconstruction of system events and are therefore a valuable resource during forensic investigations. However, extensive and prolonged tracing in high-volume environments can impact system performance. In order to manage the performance impact, parameters for the maximum size of trace files and the maximum number of trace files tend to be configured restrictively by many organisations. This reduces the forensic value of trace files since they often do not cover a sufficient period to support a complete analysis. The solution is to periodically export trace files to external secure stores. This can be performed for developer traces using the ABAP report RSMON000_DOWNLOAD_TRACES.

LOG SYSTEM CHANGES

Changes implemented using the Change and Transport System are captured in logs stored in the Common Transport Directory (CTD). The directory includes data files and cfiles that contain details of changes and transport activity logs. This includes the name and description of each transport, timestamps, the owner of the change request, the operating system user that called the tp transport tool to implement the change, and the name of the target system. The parameter transport/tp_logging should be set to ON and log files in the transport directory should be regularly archived to maintain a complete history of changes to SAP systems.

LOG TABLE CHANGES

Table change logging supports the recording of changes to values in sensitive tables such as the transport tables E070 and E071. According to SAP recommendations, logging should be enabled within a single client for customizing and productive systems using the profile parameter rec/client.⁶ Tables are flagged for logging using transaction SE13. Changes are logged in the table DBTABLOG and can be viewed using transaction SCU3. The report RSTBHIST lists the tables that are currently logged by SAP systems. Transaction SARA should be used to periodically archive table change logs. The relevant archiving object is BC_DBLOGS.

LOG DOCUMENT CHANGES

Auditing of changes to critical standard and custom SAP documents should be maintained at the client level. Documents are business objects used to support common transactions such as invoicing, journal entry posting, sales order processing and vendor payments. Auditing is enabled through change documents created with the development transaction SCDO for the relevant document types. Change documents register changes to values in specific fields selected from relevant database tables. Once enabled, changes to the selected fields are logged with a unique change document number in tables CDHDR and CDPOS. The former contains header information such as object class, user ID, timestamp and transaction code, whereas the latter contains detailed information such as table name, field name, and old and new values. These tables can be read using the standard report RSSCD200.

LOG USER ACTIONS

The Security Audit Log should be used to track user-related actions including dialog and RFC logon attempts and transaction starts, RFC calls to function modules, and changes to the audit configuration. Since changes to user master records are logged in non-transparent tables read by the User Information System (SUIM), it is not necessary to maintain logs for this specific class.

The Security Audit Log must be enabled by setting the profile parameter `rsau/enable` to 1. Once enabled, logs for the selected classes and filters are stored in the directory specified by the `rsau/local/file` parameter. Audit events generate alerts in the Computing Center Management System (CCMS). Events can also create alerts in external systems using BAPIs (Business Application Programming Interfaces). Audit policies are maintained through transaction SM19 and logs can be viewed through SM20. Dynamic filters are used to configure event logging without the need to restart application servers. However, they are deactivated after a system restart. Therefore, static filters should be used to permanently log events.

Logging can be restricted to specific systems, users and categories (critical, important or all). A narrow audit policy will consume fewer system resources but will offer less forensic value than a more broadly defined policy. Hence, it is important to balance performance issues with forensic requirements.

The Security Audit log can be configured to create a single or multiple log files for each day. Since logging is disabled when the maximum size for log files is reached, the values for the parameters `rsau/max_diskspace/local`, `rsau/max_diskspace/per_file` and `rsau/max_diskspace/per_day` should correspond to the maximum expected volume of data for the specific filters configured within each system. The parameter `rsau/log_peer_address` should be set 1 to log the peer address or last routed IP address rather than terminal IP address since the peer address cannot be modified by clients.

Security events in the J2EE Engine are logged in the UME security log. This includes user and group creation, edits and deletions, user mapping, successful and failed user logons and configuration changes. The log is located in the system directory of the J2EE cluster and can be viewed in the Log Viewer of the Visual Administrator. Log entries in the file follow the format below.

```
[TimeStamp] | [Severity] | [Actor] | [Event] | [ObjectType] = [ObjectID] |  
[ObjectName] | [Details]
```

J2EE Engine log files should be archived using the Log Configurator. The archiving process should automatically trigger when log files reach a predefined size. The process will convert log files into a ZIP file which is then transferred to a specified destination. The default location is `.log/archive`.

Read Access Logging (RAL) should be enabled to audit user access to sensitive data. RAL supports calls through RFC, Dynpro, Web Dynpro and Web service channels and is

available in NetWeaver AS ABAP 7.40 and above. Prior to enabling RAL, customers should follow several predefined configuration steps using the SAP_BC_RAL_CONFIGURATOR and SAP_BC_RAL_ADMIN_BIZ roles and associated authorizations delivered by SAP. The first involves defining logging purposes to create logical groupings of log events. The second involves creating log domains to group related fields. A domain for customer-specific information, for example, could be created to band together fields such as address, date-of-birth, SSN, etc.

Steps one and two establish the overarching structure for log information. The actual fields to be logged are identified during step three through recordings of sessions in supported user interfaces. Once identified, fields are assigned to log conditions and domains in step four. RAL is activated when the Enable Read Access Logging in Client parameter is selected in the Administration tab of the RAL Manager accessed via transaction SRALMANAGER. This represents the final step of the configuration process.

Logs can be accessed through transaction SRALMONITOR or the Monitor tab of SRALMANAGER. Log entries include attributes such as time of the entry, user name, channel, software component, read status, client IP address and details of the relevant application server. Extended views provide more detail of log events than default views. The log monitor supports complex searches of events and filtering by multiple parameters.

RAL configuration settings can be exported to other systems through an integrated transport manager accessed through transaction SRAL_TRANS. Furthermore, logs can be archived using standard Archive Administrative functions via transaction SARA.

Once configured, RAL will log all access to sensitive data including any changes performed by users. Specific users can be excluded from the analysis. In the example below, the highlighted log entries reveal that the user SAPADMIN successfully read the salary details of employee number 109815 during a SAPGUI session at 9.12AM on September 18 through the software component SAP_HRRXX.

Figure 4.1: Read Access Logging

Created At (Local Time)	User Name	Channel	Device	Logging Purpose	Read Status	Read Access Error	Software Component
18.09.2015 09:12:48.2950000	SAPADMIN	Dynpro	Output	PRIVACY	Success		SAP_HRRXX
18.09.2015 09:12:48.2950000	SAPADMIN	Dynpro	Output	PRIVACY	Success		SAP_HRRXX

Details			
Access Environment			
Type	Log Domain	Field	Field Value
Output	HUMAN_RESOURCES/SALARY	MP000000/2010/SUBSCREEN_EMPL/SAPMPS0A_CE(0110)MSPAR-PERNR (Log Context)	109815
Output	HUMAN_RESOURCES/SALARY	MP000000/2010/SUBSCREEN_TC0000.MP000000/0300/0001M00005-BETR0	0.00
Output	HUMAN_RESOURCES/SALARY	MP000000/2010/SUBSCREEN_TC0000.MP000000/0300/0001M00005-LGTXT	

MANAGE AUTHENTICATION PARAMETERS

Single Sign-On (SSO) can be used to not only support a seamless user experience but improve overall system security by enforcing a consistent authentication policy across network resources and applications. However, the use of SSO can also present a threat to organisations if improperly configured. SSO logon tickets used to authenticate users are vulnerable to interception. Therefore, attackers often target such tickets to access SAP systems using the identity of legitimate users. This can be countered by transmitting tickets through the TLS protocol. Once TLS has been configured, the parameter `login/ticket_only_by_https` should be set to 1 to prevent the transmission of tickets in clear-text. Furthermore, ticket expiration times and timeouts for HTTP sessions supported by SSO should be securely maintained through the parameters `login/ticket_expiration_time` and `http/security_session_timeout`. The parameter `login/ticket_only_to_host` should be used to ensure that tickets are sent only to servers that created the tickets rather than all servers within a domain. Direct logons by users can be disabled in SSO environments for SAP systems by setting the parameter `login/disable_password_logon` to the value 2. Specific user groups can be excluded from this rule using the parameter `login/password_logon_usergroup`.

Logons through SAP GUI should be strictly governed to manage known vulnerabilities in the desktop client that can provide an unguarded corridor to SAP systems. Wherever applicable, SAP GUI should be upgraded to the latest available release. The SAP GUI scripting API should be disabled via the parameter `sapgui/user_scripting`. This can be abused to execute transactions and processes in the background and may enable attackers to access unencrypted logon information stored in local files. 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 a local Access database.

The SAP GUI security module available in later releases should be leveraged to protect the local environments of users. The module applies 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 configured for actions not defined in the rules should be 'Ask' or 'Deny' rather than 'Allow'.

Multiple dialog logons in the same client should be disabled through the parameter `login/disable_multi_gui_login`. The list of users excepted from this rule can be maintained using `login/multi_login_users`. The maximum number of simultaneous sessions is controlled by the `rdisp/max_alt_modes` parameter. Users can have up to 6 terminal sessions open at one time. Since each session uses shared memory within an application server, it is recommended to limit the number of concurrent sessions below the default value.

Automatic timeouts for SAP GUI sessions are not enabled by default. Therefore, the parameter `rdisp/gui_auto_logout` should be set to a recommended value of no more than 600 seconds (10 minutes).

User authentication for Java applications is performed through either the UME or the J2EE Engine's Web Container. Access to Java applications and components that do not authenticate through the UME should be defined for specific roles using the `<security-constraint>` tag in the `web.xml` file. The file is located within the `WEB-INF` directory of the J2EE. Any `<http-method>` elements should be removed from the file. This will protect against verb tampering attacks against Java Servlets and Java Server Pages.

Authentication controls provided by security constraints in the `web.xml` file may be bypassed if servlets are invoked directly and anonymously through HTTP requests that call a servlet by its servlet name or fully qualified servlet class. This is enabled by the invoker servlet implemented in the `InvokerServlet` class within the Web Container. The invoker servlet is disabled by default in some patch levels of version 7.2 and in all versions of 7.3 and above. For earlier versions, the `EnableInvokerServletGlobally` key in the `servlet_jsp` should be set to `false` to prevent the authentication bypass.

Servlets, JSPs and other Java specifications should be protected against Cross-Site Request Forgery (XSRF) attacks by the SAP XSRF Protection Framework. The Framework was introduced in 2010 to combat XSRF attacks that attempt to send requests through compromised browsers to application servers using the credentials of trusted users. Applications that rely exclusively on automatically submitted credentials such as session cookies, certificates or username/password combinations are vulnerable to this form of attack since they are unable to distinguish between legitimate and malicious requests. The Framework applies tokens as an additional parameter to protect against XSRF attacks. The token is generated after logon and bound to the user session. Implementation instructions for the Framework are attached to Note 1450166.

SAP passwords can be protected against brute force and other attacks by using the latest hashing algorithm. Currently, the algorithm iSSHA-1 (iterated Salted SHA-1) applied through code version H provides the highest level of protection. Organisations using iSSA-1 should increase the default saltsize and iterations in the `login/password_hash_algorithm` parameter.

Password security should also be supported by selecting the value 0 for the parameter login/password_downwards_compatibility. This will prevent systems from storing older and more vulnerable hashes that can be targeted by attackers but may require aligning releases between central and child systems in CUA landscapes.

Automatic locks should be enabled for users after three consecutive failed logon attempts. This is enforced using the parameter login/fails_to_user_lock. User locks should be valid for an unlimited period and should only be removed by security administrators. Therefore, the value of the parameter login/failed_user_auto_unlock should be 0.

Forbidden passwords are maintained in table USR40. Vulnerable passwords should be blacklisted within the table. A strict password policy should also be applied through the profile parameters listed in Figure 5.1. The default policy supports relatively short, non-complex passwords and does not mandate the use of alphanumeric, mixed case and special characters. Compliance to a strong password policy should be enforced through the value 1 for parameter login/password_compliance_to_current_policy. This will prompt users to change their passwords to conform to the policy during the next logon attempt.

PARAMETER	DEFAULT VALUE	RECOMMENDED VALUE
login/min_password_lng	6	8
login/min_password_digits	0	1
login/min_password_letters	0	6
login/min_password_lowercase	0	5
login/min_password_uppercase	0	1
login/min_password_specials	0	1
login/password_charset	1	2
login/password_max_idle_productive	0	30
login/password_max_idle_initial	0	5
login/password_max_new_valid	0	5
login/password_max_reset_valid	0	5
login/min_password_diff	1	3
login/password_expiration_time	0	90
login/password_change_for_SSO	1	1
login/password_history_size	5	12
login/password_change_waittime	1	1
login/disable_cplic	0	1
login/update_logon_timestamp	m	m or s

Figure 5.1: Password Parameters

Java logon and password properties in the UME Security Policy should be aligned to the parameters in the data source. This includes the properties in Figure 5.2. However, user account locks based on a predefined number of failed logon attempts should be deactivated in the UME if the feature is enabled in the data source.

PARAMETER	DEFAULT VALUE
ume.logon.security_policy.lock_after_invalid_attempts	6
ume.logon.security_policy.auto_unlock_time	60
ume.logon.security_policy.password_change_allowed	TRUE
ume.logon.security_policy.password_change_required	TRUE
ume.logon.security_policy.password_expire_days	99999
ume.logon.security_policy.password_min_length	0
ume.logon.security_policy.oldpass_in_newpass_allowed	TRUE
ume.logon.security_policy.password_alpha_numeric_required	0
ume.logon.security_policy.password_mix_case_required	0
ume.logon.security_policy.password_special_char_required	0
ume.logon.security_policy.userid_in_password_allowed	TRUE
ume.logon.security_policy.password_history	0

Figure 5.2: Logon and Password Properties in UME Security Policy

MONITOR THE SYSTEM CONFIGURATION

A hardened system configuration will present a formidable challenge to potential att-ackers and block most intrusion att-empts at the SAP layer. The implementation of the recommendations in this paper will fortify SAP systems against the malicious threats posed by many advanced and complex methods of attack. However, the safeguards provided by a secure system configuration can be reversed by unauthorised changes that may expose systems to vulnerabilities that could be exploited by att-ackers. For this reason, organisations should continually monitor the configuration of all systems within SAP landscapes and rapidly respond to changes that could lead to a security breach.

SAP provides customers with several mechanisms to monitor the security of SAP software. This includes the EarlyWatch Alert (EWA). However, the EWA is not focused exclusively on security. The majority of checks performed by the alert cover areas such as system performance, workload distribution, hardware capacity and database administration. There are also concerns related to the rating scale used by EWA to classify security risks.

In the whitepaper *Managing Security with SAP Solution Manager*, SAP recommends securing, patching and monitoring SAP systems using state-of-the-art applications in Solution Manager. This includes the areas outlined in the table below. The integrated applications enable SAP customers to perform comprehensive vulnerability management, patch management and threat detection for SAP systems without licensing solutions from independent software vendors. Security Alerts generated by Solution Manager can be integrated with Security Information and Event Management (SIEM) systems for cross-platform correlation. Solution Manager can also be integrated with SAP Code Vulnerability Analyzer (CVA) to monitor vulnerabilities in custom code using a centralized platform that combines security-related system, user, event and code-level results into a single solution for holistic ERP security. SAP Solution Manager is certified for Information Security Management against the ITIL framework by organizations such as SERVIEW. Solution Manager can also support compliance with the European Union (EU) General Data Protection Regulation (GDPR) by detecting and alerting for breaches of personal data in SAP systems.

APPLICATION	SCENARIO
Service Level Reports	Automated and scheduled vulnerability reporting
System Recommendations	Patch management and change impact analysis for security notes
Dashboard Framework	Monitoring of security-related key performance indicators (KPIs)
Interface Monitoring	Mapping of cross-system connections and detection of dangerous RFM executions and URL calls
Security Alerting	Event monitoring for SAP logs including email/SMS notifications for security breaches
Guided Procedures	Best practices for security alert handling

Figure 5.3: Security Monitoring using SAP Solution Manager

APPLY SOFTWARE PATCHES

Patches for externally-reported security vulnerabilities as well as programming errors discovered through SAP's internal quality assurance efforts are applied primarily through Security Notes. SAP does not disclose detailed information related to the underlying vulnerabilities addressed by Security Notes in order to prevent the exploitation of unpatched systems. Therefore, the only indicator of the urgency of each Note is the risk rating provided by SAP.

The four-tier priority model used by SAP to rate Security Notes was replaced with a simplified three-tier approach in December 2013. Security patches are now rated on a high-medium-low scale. Priority 3 and 4 patches released in the prior model are now only provided through Support Packages. The Common Vulnerability Scoring System (CVSS) is sometimes used by SAP to estimate the risk level of vulnerabilities addressed by Notes. However, the CVSS system cannot be relied upon to rank patches since SAP does not provide a base score for all Security Notes.

Security Notes are generally released on the second Tuesday of each month known as Patch Tuesday. This interval enables SAP to synchronize patch cycles with other vendors, most notably Microsoft.

Correction instructions can be automated or manual or a combination of both. Certain Notes are targeted at raising awareness of software changes and features or providing other forms of general information. Such Notes do not contain any corrections. Corrections often have dependencies upon other Notes. Therefore, organisations should ensure patch levels remain up-to-date.

The SAP Service Marketplace is the central resource for all Security Notes. Customer security contacts should be registered at the Marketplace and review the Notes released by SAP immediately after Patch Tuesday, as well as during regular intervals during each month to identify Notes released outside the standard patch cycle. Security Notes can be filtered for customer systems registered at the Marketplace. Customers can also request automatic notifications and newsletters from SAP Active Global Support to alert security contacts of relevant Notes.

Required Security Notes for each managed system should be identified using System Recommendations accessed through the Change Management Work Center in Solution Manager. System Recommendations can also be accessed through the Easy Access Menu via WDC_NOTE_CENTER.

System Recommendations is a Fiori application that connects to SAP Support to retrieve patches directly from SAP on a weekly basis. It also connects to individual systems in SAP landscapes to determine their patch level. The results can be filtered by system, component, or other factors. Automated patches can be downloaded from SAP Support by System Recommendations to target systems. The application integrates with areas such as Usage and Procedure Logging (UPL) and the ABAP Call Monitor (SCMON) to perform change impact analysis for notes before they are implemented. It also integrates with Solution Documentation to identify business processes impacted by notes.

As previously mentioned, SAP no longer delivers all corrections for vulnerabilities through Security Notes. Some corrections are provided through support packages. Therefore, SAP strongly recommends the implementation of support packages as soon as they are available at the SAP Service Marketplace.⁷

SAP systems are not secure by default. Few of the actions discussed in this paper across the multiple controls required to protect SAP systems from cyber attack are enabled in the standard SAP installation. Network filters are not preconfigured in application gateways. Network communications are not encrypted. Remote access to function modules and web services is not blocked. Security files for gateway servers are not delivered by SAP. Most logs are not enabled. And so on. As a result, standard SAP systems are exposed to a range of internal and external risks that could threaten the security of such systems and organisational processes that rely upon the availability and integrity of system resources.

The trend towards minimizing changes to standard configurations in order to accelerate implementations and lower maintenance costs may have dire consequences if applied to the security arena. Protecting SAP systems from cyber attack demands a proactive approach. The aim of hardening SAP systems to withstand advanced threats can only be realized by organisations that actively challenge and transform the basic security baseline.

This transformation does not call for software add-ons or third party solutions. The tools to effectively secure SAP systems are provisioned by SAP to all customers through standard license agreements. The SAProuter and Web Dispatcher are freely available at the SAP Service Marketplace. Digital certificates required to secure communications can be obtained from the SAP Trust Center. Authorizations can be analyzed through the User Information System. Unified Connectivity and Read Access Logging are packaged in the most recent versions of NetWeaver AS. Lastly, the powerful monitoring and patch management capabilities of Solution Manager are available to all SAP customers as part of standard and enterprise support agreements.

A comprehensive security approach based upon employing mechanisms available within SAP systems across the five domains of network security, RFC communications, user authorisations, logging and system configuration management will enable organisations to safeguard information assets from cyber attack and realize the potential of SAP solutions.

APPENDIX A ROADMAP FOR CYBER SECURITY IN SAP SYSTEMS

CONTROL OBJECTIVE	CONTROL	ACTION	REMIEDIATION EFFORT			
			COMPLEXITY	RESOURCES	DURATION	
Secure the Network	Configure Network Zones	Configure firewall rules to segment systems into relevant network zones	H	M	H	
		Configure single-purpose servers	H	H	H	
	Filter Network Access	Install and configure the SAProuter to filter SAP traffic	H	M	H	
		Install and configure the Web Dispatcher to filter Web-based traffic	H	M	H	
		Enable reverse invoke to block incoming connections to servers in the internal LAN	M	M	H	
		Protect message servers by limiting the privileges of external programs and configuring separate internal/ external ports or controlling external access through ACLs	H	M	M	
	Encrypt Network Communications	Enable SNC to encrypt SAP traffic	H	M	H	
		Enable TLS to encrypt Web-based traffic	H	M	H	
	Reduce the Attack Surface	Disable unnecessary ports and services	H	M	H	
		Limit external access to remote enabled function modules	H	H	H	
		Disable unnecessary ICF services	H	H	H	
		Disable redundant software components	H	M	H	
		Remove redundant custom code	M	H	H	
	Protect Remote Function Calls	Secure the Gateway Server	Reject commands from remote monitors	H	L	L
			Configure sec_info files to control access to gateway servers	H	M	H
Configure reg_info files to control the registration of external programs in gateway servers			L	M	H	
Manage RFC Destinations		Configure RFC destinations with system users	M	L	M	
		Configure RFC destinations with unique users	H	L	M	
		Enable checks for RFC authorizations and provision access based on least privilege	H	M	H	
		Remove trusted RFC connections from lower order to higher order systems [e.g. Development to Production]	H	M	H	
		Configure the authorisation object S_RFCACL to secure trusted RFC connections	L	M	H	
		Activate switchable authorization checks	M	L	M	
		Enable whitelists for RFC callbacks	M	L	M	
Control Access to Basis Functions	Manage Standard Users and Profiles	Enable global authorizations checks	L	L	L	
		Lock standard ABAP users and change default passwords	L	L	L	
		Change default passwords for standard Java users	M	L	L	
		Assign standard ABAP users to a super user authorisation group	M	L	L	
		Remove object S_TCODE with full authorizations for field TCD from users	M	L	M	
		Remove the profile SAP_ALL from users in all systems	H	M	M	
		Remove SAP_NEW from authorization profiles within a reasonable timeframe following a system upgrade	M	M	H	

CONTROL OBJECTIVE	CONTROL	ACTION	REMIEDIATION EFFORT			
			COMPLEXITY	RESOURCES	DURATION	
Restrict Access to Authorization, Role and User Administration	Restrict Access to Authorization, Role and User Administration	Control the assignment of the following authorizations with the relevant activity levels: S_USER_GRP, S_USER_PRO, S_USER_ADM, S_USER_SYS, S_USER_SAS, S_USER_AGR, S_USER_AUT	M	L	M	
		Segregate authorisations for role creation/ modification and role assignment between administrators	M	M	M	
		Control the assignment of UME administrator roles and permissions	M	M	M	
	Restrict Access to System Administration	Control the assignment of the S_ADMI_FCD authorisation object	H	M	H	
		Remove the object S_DATASET with full authorisations from users	M	M	M	
		Control the assignment of the batch operations authorisations S_BDC_MONI, S_BTCH_ADM and S_BTCH_JOB	M	M	M	
		Control the assignment of the RFC administration authorisations S_RFC_ADM and S_RFC_TT	M	M	M	
		Restrict permissions to maintain RFC trust relationships to specific systems	H	M	H	
		Control the assignment of the authorisations S_RZL_ADM, S_PROGRAM, S_TMS_ACT and S_NUMBER.	M	M	M	
	Restrict Access to Table Maintenance	Control access to critical tables using authorisation groups and/ or the object S_TABU_NAM	H	H	H	
		Remove full authorisations for table access from users	M	M	M	
		Control access to the data dictionary	M	M	M	
	Restrict Access to Transport Management	Control the assignment of the authorisation S_CTS_ADMI with the relevant values in the CTS_ADMFCT field	H	M	M	
		Control the assignment of the authorisation S_TRANSPRT for the relevant request types and activity levels	H	M	M	
		Enable change locks in production systems	M	M	M	
		Remove developer keys from production systems	H	M	M	
		Remove the authorisation S_DEVELOP from production users	M	L	M	
		Control access to the Legacy System Migration Workbench [LSMW]	M	M	M	
	Maintain Log Information	Log Network Activity	Enable SAProuter logging	M	M	M
			Enable ICM and Web Dispatcher logging	M	M	M
			Enable HTTP logging in the message server	M	M	M
Log System Events		Enable local and central system logs	M	M	M	
		Enable message server logging	M	M	M	
		Enable gateway server logging	M	M	M	
		Enable trace functions for critical system events	H	M	H	
Log System Changes		Log SAP transports	M	M	M	
Log Table Changes		Log changes to critical tables	H	H	H	
Log Document Changes		Log changes to critical documents	H	H	H	
Log User Actions		Enable and effectively configure filters in the Security Audit Log	H	H	H	
		Log UME security events	H	M	M	
	Enable Read Access Logging for sensitive data	H	H	H		

CONTROL OBJECTIVE	CONTROL	ACTION	REMIEDIATION EFFORT		
			COMPLEXITY	RESOURCES	DURATION
Manage the Configuration	Manage Authentication Parameters	Securely configure Single Sign-on	H	M	M
		Manage SAP GUI client settings	H	M	H
		Secure declarative authentication and remove <http-method> elements for Java applications within web.xml files	H	M	H
		Set the EnableInvokerServletGlobally key in the servlet_jsp to false	M	L	L
		Implement the SAP XSRF Protection Framework to safeguard Servlets, JSPs and other Java specifications against XSRF attacks	H	M	M
		Enable the latest algorithms for password hashes	M	M	M
		Increase the default saltsizes and iterations for the iSSA-1 password hashing algorithm	M	M	M
		Enable reasonable automatic locks for user accounts	L	L	L
		Define and maintain forbidden passwords	M	M	H
		Effectively configure password parameters	M	M	M
	Effectively configure Java logon and password policies in UME Security Policies	M	M	M	
	Monitor the System Configuration	Review and remediate issues identified by vulnerability reports using Configuration Validation and Service Level Reporting in SAP Solution Manager	L	L	L
		Identify and secure vulnerable system connections using Interface Monitoring in SAP Solution Manager	M	L	L
		Investigate alerts for critical security events using the Monitoring and Alerting Infrastructure [MAI] and Guided Procedures in SAP Solution Manager	H	H	H
		Leverage Fiori Dashboards and the Dashboard Builder in SAP Solution Manager to monitor key security metrics	M	M	H
	Apply Software Patches	Identify and retrieve Security Notes from SAP Global Support using System Recommendations immediately after the second Tuesday of every month	M	L	L
		Develop regression test plans incorporating the results from Business Process Change Analyzer [BPCA] or equivalent	H	M	M
		Leverage Solution Documentation in SAP Solution Manager to identify business processes impacted by Security Notes	M	M	M
		Identify application components impacted by Security Notes with Usage and Procedure Logging in SAP Solution Manager	H	H	H
		Generate required change requests through Change Request Management [ChaRM] or equivalent	M	L	L
Implement relevant support packages within a reasonable timeframe after release		H	H	H	

REMIEDIATION EFFORT SCALE

	LOW	MODERATE	HIGH
COMPLEXITY	Non-complex, requires minimal experience and expertise	Reasonably complex, requires modest experience and expertise	Highly complex, requires extensive experience and expertise
RESOURCES	Requires a single resource	Requires between two and three resources	Requires more than three resources
DURATION	One to two days	Three to five days	More than five days



Layer Seven Security secure, patch and monitor SAP systems against cyber threats using SAP Solution Manager. Layer Seven's innovative and patent-pending Cybersecurity Extension for Solution Manager extends the capabilities of Solution Manager for advanced vulnerability management, threat detection and incident response.

CONTACT US

99 Hudson Street
5th Floor
New York, NY 10013

www.layersevensecurity.com
info@layersevensecurity.com

