



PERFECT STORM

THE BRAVE NEW WORLD OF SAP SECURITY

ABSTRACT

Access controls and other areas traditionally focused upon by security professionals are no longer the only major risks to SAP systems. Today, SAP is confronted with a growing landscape of threats that include injection attacks, cross site scripting, session hijacking and denial of service. For the most part, business owners and security professionals are unaware of profound vulnerabilities laying in the technical components of SAP.

These risks have arisen from the gradual shift towards open source languages, protocols, standards and Web-enabled services, as well as the increasing size and complexity of SAP. When combined with inherent weaknesses in existing network controls and the sophistication of attacks targeted at corporate applications and data, such a rare combination of circumstances should be viewed as a sinister perfect storm.

Vulnerabilities in critical SAP components and services could be exploited by external attackers to interrupt SAP services, implement malicious changes to programs and files, intercept and alter data in transit, and corrupt or modify data directly in databases. If left unattended, these vulnerabilities raise serious concerns over the ability of companies to comply with SOX, PCI and other standards.

This white paper discusses some of the methods used to attack and compromise SAP systems. It also addresses some lesser known risks to raise awareness within the community and improve the overall posture of SAP security.

Layer Seven Security
www.layersevensecurity.com
info@layersevensecurity.com
Tel. 1 888 995 0993

© Copyright Layer Seven Security 2011 - All rights reserved.

SAPSCAN is a patent pending trademark of Layer Seven Security.

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.

CONTENTS

Introduction	The Brave New World of SAP Security	4
Chapter One	The Problem with Remote Authentication	8
Chapter Two	Securing Remote Function Calls	10
Chapter Three	Evil Twin, Man in the Middle and other Attacks	12
Chapter Four	Controlling Default SAP Users	14
Chapter Five	SAP_NEW	15
Chapter Six	Breaking SAP Password Security	16
Chapter Seven	Exploring SAP Backdoors and Rootkits	18
Chapter Eight	Attacks against the SAP Java Engine	20
Chapter Nine	Managing Vulnerable SAP Web Services	24
Chapter Ten	Examining Vulnerabilities in SAP GUI and Web Clients	26
Chapter Eleven	The SOX and PCI Implications of SAP Vulnerabilities	28
Chapter Twelve	Software, Audit Programs and Web Resources	31
Endnotes		33

Introduction

Perfect Storm: The Brave New World of SAP Security

Until recently, SAP systems led a sheltered life, quietly supporting business processes on local networks behind a veil of corporate firewalls. This led to a widely held belief that security for such systems was largely a matter of protecting internal access, regulating changes to programs and controlling parameters known as configurables for passwords, pricing, payables and other areas.

The advent of SOX and other compliance requirements led security professionals to perform a systematic review of these areas. Since most SAP systems fell short of the expectations set by the requirements, many companies embarked on a long, arduous journey to harden their environments. In recent years, many companies have managed to narrow the compliance gap, helped in part by monitoring tools such as Approva and SAP GRC, designed to ease the burden of compliance.

The standard compliance baseline established for SAP security is outdated.

Today, the standard compliance baseline established for SAP security is largely out of date. SAP has evolved from a narrow, back-office system accessible exclusively through internal networks into a vast, complex association of a seemingly infinite number of applications.

Security efforts up to now have been largely focused upon the enterprise application known as SAP ERP. Today, ERP is only one of five interconnected enterprise applications in SAP's Business Suite. Many of these areas are uncharted territory for security professionals.

SAP has been Web-enabled since the 1990s

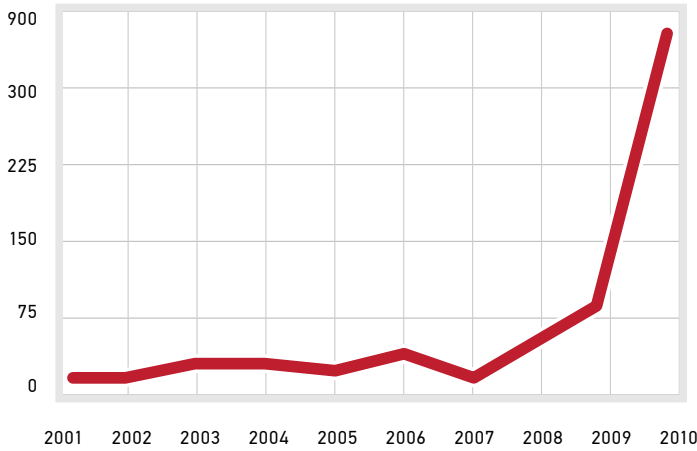
The widening of SAP's product offering has been accompanied by another more dangerous trend: Internet connectivity. SAP R/3 has been Web-enabled since the introduction of 3.1G in the 1990s. Today, companies leverage the Internet through SAP to

engage directly with customers, suppliers and other partners. As a result, business processes increasingly stretch across corporate boundaries and data is no longer concentrated within SAP but dispersed across a myriad of external systems connected through the Web.

To perform this feat, SAP embraced Java, a more common programming language than its own propriety code. It also adopted more open standards for communication and database interfacing such as HTTP, SOAP/ XML and Open SQL. Finally, it built an integration platform powerful enough to translate two languages and handle the intense loads placed by both SAP's own applications and external systems. This platform was called the SAP Web Application Server (Web AS), later renamed to the Netweaver Application Server. In essence, the Web AS was no different from any other web server. It served the same functions and crucially, suffered from the same vulnerabilities.

The introduction of the Web AS, as well as Java and open standards, enabled SAP to meet the demands for greater connectivity and accessibility. However, it has also exposed the inherent weaknesses in legacy SAP services that were never designed to withstand the threat of external attack and intensified the risks associated with other, more well-known vulnerabilities. SAP was swimming with sharks and what's more, it was badly ill-equipped.

Security is a fairly simple concern when you're dealing with internal-facing systems. Web applications, on the other hand, are an entirely different matter, especially when those tasked with managing the security of such applications are not trained to deal with external threats. In a rare interview during 2008, SAP's former Chief Security Officer publicly acknowledged the challenges faced by SAP. In his words, "The problem is that the classical, well understood Internet threats are often not understood by the ERP people. The people who are responsible for ERP understand the insider threat because they have dealt with it for years, but when there is a demand from the business to extend systems to the Internet, they don't think about threats like cross site scripting".¹



The number of Security Notes issues by SAP increased by a staggering four thousand percent in three years

The first sign that something was seriously wrong with the state of SAP security was the sudden, dramatic increase in the number of Security Notes. In 2007, SAP released approximately 20 Notes. By 2010, it had risen to almost 900. Today, SAP routinely issues more Security Notes in a single month than released in any year between 2000 and 2007.

Aside from the sheer quantity of Notes issued by SAP, other factors that raised concerns included the type of vulnerabilities patched by the Notes which often targeted communication components and protocols such as the Message Server, Internet Communication Manager (ICM) and Remote Function Calls (RFC), as well as areas such as database authentication, ABAP programming and password hashing. Tellingly, they also targeted attack vectors associated with web applications including SQL Injection, Cross-Site Scripting (XSS), Directory Traversal and Cross Site Request Forgery (XSRF).

Web applications are the preferred target for external hackers looking for remote, anonymous and accessible pathways to corporate applications and data stored in backend servers. They are made even more attractive by low barriers to entry, often requiring nothing more than a laptop and a browser. Attacks vectors are the methods deployed by hackers to compromise system resources.

The Open Web Application Security Project (OWASP) is a widely respected, not-for-profit body that is dedicated to supporting organisations develop and maintain secure web applications. OWASP periodically publishes an analysis of the Top Ten Most Critical Web Application Security Risks based on extensive industry research. This study is widely regarded as the most authoritative, independent analysis of the threat landscape for Internet applications. The most recent list is summarized in Figure 1.1.

A1	Injection
A2	Cross Site Scripting (XSS)
A3	Broken Authentitication and Session Management
A4	Insecure Direct Object References
A5	Cross Site Request Factory (CSRF)
A6	Security Misconfiguration (NEW)
A7	Failure to Restrict URL Access
A8	Invalid Redirects and Forwards (NEW)
A9	Insecure Cryptographic Storage
A10	Insufficient Transport Layer Protection

Figure 1.1: OWASP Top Ten Web Application Security Risks 2010

SAP is vulnerable to every attack vector in the OWASP Top Ten

If SAP systems are not patched, upgraded or otherwise secured, they are vulnerable to every single attack vector identified in the OWASP Top Ten. Furthermore, they are vulnerable to four of the eight other application security risks singled out by OWASP in the study.² This will be illustrated in the following chapters which will examine lesser known SAP vulnerabilities and provide practical advice on measures to counteract threats similar to those often targeted at web applications to devastating effect. The millions of customer and credit card records compromised during the well-publicized breaches at Heartland Payment

Systems, Sony and TJ Maxx are merely the tip of the iceberg. According to U.S Congress reports, such cybercrime costs the U.S economy an estimated \$8 billion a year and resulted in the theft of \$1 trillion worth of intellectual property from U.S businesses.³

Incidences of data breach are increasing and growing in sophistication. According to a recent study by the University of Toronto's Rotman School of Business, "present day threats routinely utilize sophisticated methods of invasion, evasion and propagation". The study also notes that threats are increasingly "designed for monetization, either through the theft of corporate secrets or through the acquisition and abuse of identities and credentials".⁴

The average cost of a data breach is \$7.2M

This echoed the findings of an earlier study performed by Verizon, the U.S Secret Service and the Dutch High Tech Crime Unit which concluded that attacks are generally perpetrated by well-trained external agents using automated tools, supported by organized, financially-motivated groups. It's worth noting that the average direct and indirect cost of a data breach calculated by the study was \$7.2M.⁵ This includes detection, investigation, notification, litigation and reputational harm, not to mention lost business.

Attacks are not only targeted at systems housing large volumes of sensitive personal, credit or banking information. Almost all companies are vulnerable, regardless of size and industry. Attackers routinely sell stolen credentials or backdoor access to corporate systems to any interested buyer in growing on-line markets. Like any other market, the buying and selling of privileged access to hacked systems follows the law of supply and demand.

Application and data centric attacks bypass network controls

Motivated by the large illicit profits from such sales, attackers are focusing their efforts at the application and data level through commonly used ports such as 80 (HTTP), 443 (HTTPS) and 22 (SSH). This bypasses port-level firewalls, intrusion detection or prevention systems and other security appliances designed to monitor and control traffic between trusted and untrusted networks. Network-level devices are not tuned to deal

with contemporary application and data centric attacks. In fact, according to research performed by the SANS Technology Institute, attacks such as those attributed to the Anonymous and Lulz hacking groups could not be repelled by traditional firewalls. In 2009, Gartner had warned that "the stateful protocol filtering and limited application awareness offered by first generation firewalls are not effective in dealing with current and emerging threats".⁶

SAP systems are attractive and vulnerable targets

Weak perimeter controls provide a direct route to information-rich systems for financially-motivated attackers. There are very few targets more prized than SAP, the crown jewels of corporate systems. The components, ports, protocols and services of SAP systems are widely known and easily accessible. Once a company is compromised, attackers can readily apply the same techniques to other companies suspected to be using similar configurations. SAP clients are not hard to find. In fact, many are publicized by SAP itself through customer testimonials.

Taken together, the increasing prevalence of targeted application and data level attacks that evade conventional controls, the lucrative market for stolen data and credentials, and the relative openness of SAP systems has created a rare combination of circumstances that could be referred to as a perfect storm. The outcome of such a convergence of events is likely to be catastrophic for companies caught in the midst of the storm.

SEC urges public companies to disclose cyber risks and events

Since corporations are not obligated to disclose major data breaches resulting in the loss of financial assets, intellectual property or sensitive information other than that belonging to customers, it is impossible to know whether SAP systems have already been compromised. Understandably, most organizations are unlikely to publicize such an event, presuming they are even aware of any breach (given the sophistication of today's attacks, many companies may be completely unaware of data breaches affecting their systems). However, recent guidance issued by the U.S Securities and Exchange Commission (SEC) should be viewed as a harbinger of a very different future. Acknowledging that

successful cyber attacks could lead to lost revenue, litigation and other financial costs, the SEC published clear guidelines in 2011 that urge public companies to disclose accurate, comprehensive and timely information about cyber risks and events that could impact the decisions of investors.⁷

Organizations relying upon SAP applications should review their security profile in light of these requirements. Protective measures that up to now have been focused on internal risks to access, change control and configuration management should be widened to tackle the entire landscape of threats faced by SAP systems.

Many of these threats are discussed in Chapters 1-10 of this Paper. Advice on countermeasures is provided wherever possible and is strongly recommended.

Chapter 11 maps the vulnerabilities to specific SOX and PCI requirements. This is presented to enable organisations understand the impact of vulnerabilities in terms of regulatory and industry compliance, as well as facilitate changes to audit and compliance programs.

SAPSCAN analyses over 300 vulnerabilities using SAP-certified software

Chapter 12 discusses open source programs and other useful resources for reviewing SAP security. It presents a compelling argument for the use of commercial tools and services that examine hundreds of complex and highly vulnerable technical settings in SAP. This includes SAPSCAN, a vulnerability assessment performed by Layer Seven Security that provides business and technology owners with a clear and comprehensive analysis of over 300 SAP vulnerability areas. SAPSCAN is performed by highly trained security professionals through patent-pending commercial software certified by SAP.

Chapter One

The Problem with Remote Authentication

Although SAP can be configured to work with almost any database including its own offerings, Hana, MaxDB and Sybase, most customers choose to power their systems with Oracle databases. This is not without good reason: Oracle is widely considered to have an edge over competitors in areas such as performance, durability and support.

Access to Oracle is usually handled through database authentication which, as the name suggests, means that that you need to provide a username and password that matches credentials stored in the database. All default Oracle accounts use database authentication. However, Oracle can be configured to accept operating system authentication based on trust relationships. As a general rule, trust and security go together like water and wine. Before we discuss any further, let's go over some background.

Oracle can accept operating system authentication based on trust relationships

In the eyes of the database, the SAP system is a single user, either SAPR3 or SAP, whose password is stored in the SAPUSER table (incidentally, the default password for SAPR3 is sap). In order to access the database and retrieve the password for the SAPR3/ SAP user, SAP uses something called the OPSS\$ mechanism. In a nutshell, SAP first logons to Oracle using an OPSS\$ user ID and then logs back on with the retrieved credentials for the SAPR3/ SAP user. This applies to both UNIX and Windows environments. In the latter case, the user would be OPSS\$<domain>\<sapsid>adm.

As well as setting up the connection to the R/3 database, OPSS\$ is also used to access BR TOOLS such as BRBACKUP, BRCONNECT and BRRECOVER. These are SAP tools used to manage data in Oracle. One of the most interesting is BRCONNECT which can be used to change passwords for SAP database users and stop/ start Oracle.

As stated earlier, Oracle can allow certain users to authenticate against the operating system rather than the database. In other

words, Oracle can be configured to trust the OS to validate a user's logon credentials and allow access to the database without supplying a password, providing the OS user is a valid database user. In this scenario, the database will look up an Oracle ID that matches the name of the OS user and make a connection if there is a match. Note that in most cases the Oracle ID is the same as the SAP System ID (SSID).

An attacker can exploit remote OS authentication to shut down, corrupt or raid the database

Operating system authentication can be either Local or Remote. Local OS authentication uses the database server's OS to authenticate users. Remote OS authentication is far trickier. This allows any client that can make a network connection to the database to authenticate users. Anybody with administrative access to their machine can create a user on their local box with the same name as an Oracle user and connect to the database. They would merely need (1) host name, (2) SSID and then bank on (3) the Oracle ID being the same as (2) which is a very safe bet. This would provide an attacker with privileged remote access to the database without authenticating directly against the database using a password.

Keep in mind that client separation doesn't exist at the database level. In other words, data can be viewed and changed across all clients in the database. It gets worse: once in the database server, a hacker can jump from one server to another if remote shell (rsh) is enabled.

The risk is obvious and restricted only by the imagination of the attacker. It can range from shutting down the database to complete data theft or corruption. In fact, the risk is so great Oracle deprecated remote authentication in version 11g. This is computer jargon for warning against its use with possible phasing out in later versions.

You can check whether remote authentication is enabled in your Oracle instance by reviewing the REMOTE_OS_AUTHENT

parameter. Oracle will trust connections from remote systems if the parameter is set to TRUE. In all cases, the parameter should read TRUE. This is because SAP requires remote authentication to function properly with Oracle databases. Therefore, you should only allow trusted servers to authenticate remotely with your database server. This can be achieved by configuring database access in the protocol.ora file or, if you're using Oracle 9i or later, the sqlnet.ora file.⁸

Only allow trusted servers to authenticate remotely with your database server

The content of the file should read as follows:

```
tcp.validnode_checking=yes  
tcp.invited_nodes=(address1, address2,)  
tcp.excluded_nodes=(address1, address2,)
```

Addresses can be host names or numeric IP addresses. You cannot specify IP ranges or network masks. You can choose whether to use invited nodes as part of a whitelist approach or list excluded nodes if you use decide to opt for a blacklist.

Chapter Two

Securing Remote Function Calls

RFC is the acronym for Remote Function Call, the interface driving communications within SAP and between SAP and external systems. Communication can be synchronous or asynchronous, transactional or queued, but in all cases is performed through TCP/IP or CPI-C connections.

ABAP function modules need to be remote-enabled to support RFC. Destinations are stored in table RFCDES, accessible through transaction SM59. All communications flow through the Gateway Server which includes a Reader to receive and process RFC requests, a Work Process to handle communication with IBM mainframes and a Monitor to analyze and administer the Gateway (the latter is very important but we'll get to that in the next chapter).

RFC calls between SAP systems can be untrusted or trusted. Untrusted calls are authenticated and authorized through the S_ICF object in the client system and S_RFC in the server system. No authentication is required for trusted calls since the server trusts the client.

Business data, usernames and passwords are transmitted in clear text

The most glaring problem that strikes you when you examine the default settings for RFC is that communication is in clear text. Therefore, techniques such as network sniffing can be used by attackers to seize credit card, bank, payroll and other confidential data, as well as the credentials required to logon to an SAP system: client, username and password. Passwords are obfuscated with a XOR algorithm using a fixed key. XOR encryption is simple to implement and equally simple to break, requiring very little expertise. XOR is not recommended for the encryption of sensitive data. In fact, the XOR key to decrypt SAP passwords transmitted through RFC is widely available on the Web.

RFC functions can be abused by attackers to detect targets and disclose system information

Traffic sniffing on unencrypted data flows is an example of a passive attack vector. However, RFC is also vulnerable to active vectors which often exploit functions available in every external RFC server through the SRFC special function group. These functions can be called remotely and anonymously since authority checks are not configured by default and rarely enabled by administrators. They include:

1. **RFC_PING.** According to the Introduction to RFC Server Programs published by SAP and available on the company's Help Portal, this function "does nothing by itself" and is only used to "test the (RFC) connection". In fact, RFC_PING can be used to detect the availability of RFC interfaces between internal and external systems, which is usually the first step taken by an attacker attempting to exploit SAP vulnerabilities.
2. **RFC_SYSTEM_INFO.** With characteristic understatement, SAP states that "This function returns some information about the library and its environment". SAP fails to mention that this information can include the SAP kernel version, host name, time zone, database engine, database host, SAP system ID and operating system. This information can be used by a hacker to launch a targeted attack against an SAP system.
3. **RFC_TRUSTED_SYSTEM_SECURITY.** This function was developed by SAP for internal use only and inexplicably found its way into customer systems. It can be used by savvy hackers to examine Windows domains, groups and user accounts in external servers.
4. **RFC_SET_REG_SERVER_PROPERTY.** If called with the appropriate parameters, this function can provide an attacker with exclusive use of an RFC server, leading to denial of service.

Some RFC functions are vulnerable to buffer overflows which can enable attackers to compromise SAP servers

5. RFC_START_GUI

6. SYSTEM_CREATE_INSTANCE

7. RFC_START_PROGRAM

These functions are vulnerable to buffer overflows which could enable attackers to execute remote commands over SAP servers.

Given the danger associated with these functions, SAP released a number of Notes to patch the RFC library.⁹ Needless to say, you should immediately apply these patches to your environment if you haven't already done so.

SAP also developed Secure Network Communications (SNC) to encrypt network traffic using SAP protocols including RFC. This is strongly recommended, although you cannot apply SNC to the communication path between your application servers and database.

You should also restrict access to transaction SM59 and table RFCDES, as well as enable the use of authorization object S_RFCACL to improve the security of trusted RFC calls.

Our final recommendation is to take a long hard look at Notes 43417, 618516, and 1140031. These Notes are designed to address vulnerabilities in the RFC Software Development Kit (SDK), especially RFCEXEC, which poses a major security risk since it enables the remote execution of operating system commands. RFCEXEC includes functions such as REMOTE_PIPE, REMOTE_FILE, and REMOTE_EXEC. These functions are as dangerous as they sound.

Chapter Three

Evil Twin, Man-In-The-Middle and Other Attacks

The flow of communications traffic within SAP systems and between SAP and external systems is managed by the Gateway Server. Remote access to the Gateway Monitor, a vital component of the Server, is enabled by default in SAP Kernels 6.20 and below. The specific configuration parameter to watch out for is `gw/monitor = 2` Remote access enabled. Administrators don't always change these and other default settings, which allow the remote registration of any external server using the same ID as any other registered server. This exposes the Gateway to so-called Evil Twin attacks.

Evil Twin attacks can evade network firewalls and lead to denial of service or the leakage of sensitive data

Such attacks can unfold in several ways but an easy and simple variation involves an attacker blocking the serving of RFC requests by a legitimate registered server, registering a server with the Gateway using the same program ID as the server that was blocked a few moments ago and then waiting for the Gateway to route RFC calls intended for the legitimate server to the illegitimate server, registered with the same name. This is easier than it sounds and requires nothing more than a personal laptop, an SSH client such as Putty and some basic credentials for target systems which an attacker can sniff from unencrypted traffic flows or obtain directly from the Gateway.

An Evil Twin attack can lead to denial of service or the leakage of sensitive data including logon credentials contained in open communications traffic. These risks are present even if external servers are located in secure network segments behind a barrage of powerful, hardened firewalls.

MITM attacks can be used to modify data in communication streams without detection

The Gateway is also vulnerable to Man-In-The-Middle (MITM) attacks that use some of the same methods as Evil Twin attacks but are far stealthier and therefore, more difficult to detect. This

usually involves a hacker adjusting RFC calls intended for a legitimate external server before returning the results to the requesting client through the Gateway. Such an attack could be used to modify RFC requests and/ or the data returned to SAP and other clients. It's important to remember that data in the context of RFC communication can be almost anything: financial results, bank accounts, credit card numbers, customer records, salaries, etc.

Call Backs can enable attackers to take control of servers with SAP_ALL privileges

Often a server cannot complete an RFC request without obtaining more information from the requesting client through a call-back. When doing so, the server bypasses authentication in the client and assumes the same privileges as the user that was used by the client to initiate the call. Often, the client is an SAP Application Server and the user has SAP_ALL authorizations. You can guess the rest.

This illustrates how merely enabling the registration of external RFC servers on the Gateway can be exploited by an attacker to gain complete control over an SAP system.

You can disable remote access to the Gateway Monitor with the following parameter: `gw/monitor = 1` Local access only. You can also restrict the ability to register servers with the Gateway using a whitelist approach through the `gw/sec_info` and `gw/reg_info` files found in the data dictionary of your SAP instance. These files should be periodically reviewed for illicit changes. For more information, refer to SAP Notes 110612 and 64016.

In addition, you should ensure that RFC users are setup as system rather than dialog users to limit their ability to interact with SAP, although this may not be possible in all cases. You should also ensure they have the minimal level of rights in target systems to perform their functions. While you're at it, it may be a good idea to review the list of RFC users since users are often replicated in multiple systems without any business need during

cross-system synchronization.

Finally, we recommend locking down access to RFC and Gateway trace files located in the WORK directory of the application server. These files often contain important security and runtime information such as usernames and passwords. To learn how to perform this, refer to SAP Note 532918.

Attackers can target the Message Server to redirect business traffic in clear-text to their machines

The Message Server is primarily a load balancer for application servers. It can be found on the central instance, usually configured at port 3600.

Unlike the Gateway Monitor, remote access to the message server is disabled by default. You can verify this hasn't changed by checking the ms/monitor parameter which should be set to 0 rather than 1. However, the security seems to end there since, just like the Gateway, the Message Server allows anyone to register an application server from any location. The ACL for the Server should be an extensive whitelist specifying all host names, domains, IP addresses and/or subnetwork masks from which application servers are allowed to log on to the message server. Contrast this to the following default in ms/acl_info:

```
HOST=*
```

You don't have to be an über hacker to recognize the opportunities presented by this scenario. Before long, most would have guessed the following attack vector:

1. The attacker registers a rogue computer with the Message Server.
2. The rogue computer sends load information to the Server, making sure to let the Server know that it has plenty of capacity.
3. Since the Message Server is configured to distribute requests to servers with the least loads, the attacker waits for the Server to redirect a stream of business traffic to the rogue computer, most likely in plain text.

This highlights the importance of properly configuring the Message Server ACL to restrict connections to the service. You can augment a strong ACL with the SAProuter which works as an application-level gateway or reverse proxy, controlling connections with SAP systems.

The SAProuter route permission table should be used to define source and target IP addresses, SNC encryption, protocols and password authentication. Remember to allow only external connections using the SAP protocol and round off the list with an appropriate deny-all rule.

Chapter Four

Controlling Default SAP Users

SAP ships with a number of privileged standard user IDs. Examples include SAP*, DDIC, EARLYWATCH, SAPCPIC and TMSADM. These IDs are used for, among other things, management of the data dictionary, ABAP repository, program interfaces and transports, as well as system monitoring and troubleshooting.

SAP recommends changing the default passwords of standard users and even goes as far as urging customers to lock SAP*, EARLYWATCH and SAPCPIC and enforce regular password changes for most standard users. The argument for locking down such users is plain to see but there are a number of risks that should be taken into account when securing standard users. For example, password changes for TMSADM can be problematic since it can have an adverse effect on transports.

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

Figure 4.1: Default SAP users

Another risk associated with poorly designed lock-down strategies concerns our dear friend, SAP*.

The root privileges available to the SAP* user often sends alarm bells ringing in audit reports which in turn can lead to panicked, knee jerk reactions from administrators. Indeed, some may respond by deleting SAP* altogether from the user records. This can have dire consequences.

SAP* is unlike any other user. It is hard-coded in the SAP kernel. Since the user is required in emergency situations such as assisting administrators to get back into locked-out systems, it can never be completely removed even when deleted from the user records.

Deletion leads SAP to recreate the SAP user with unrestricted access and a widely known default password*

SAP* behaves like a normal user as long as you maintain its master record. Therefore, it's subject to authorization checks and password changes. Deleting the master record will remove SAP* from the user tables and give the impression that it's no longer available for use since it won't be visible in the tables or report RSUSR003. However, since SAP* is programmed in the system, it doesn't require a user master record to be active and accessible. Deletion from the user tables merely removes the checks performed on SAP* which provides the user with even greater authorizations. It also changes the password to PASS and provides no avenue to change the password. In short, deletion leads SAP to recreate the SAP* user with unrestricted access and a widely known default password.

A far better response is to deactivate SAP* and replace it with another super user. SAP's recommendations are provided below.

- Create a user master record for SAP* in all new clients and in client 066.
- Assign a new password to SAP* in clients 000 and 001.
- Delete all profiles from the SAP* profile list so that it has no authorizations.
- Ensure that SAP* is assigned to the user group SUPER to prevent accidental deletion or modification of the user master record.
- Set the system profile parameter `login/no_automatic_user_sapstar` to a value greater than 0.

Chapter Five

SAP_NEW

There is a common misconception that the authorization checks packaged in a SAP_NEW profile with each release are designed to enable users to gain access to newly introduced functions.

Although nothing could be further from the truth, this misconception is understandable since SAP artfully packages the authorization checks in upgrades and releases which creates an automatic association in the minds of many people between new functionality and the newly introduced authorization checks. Of course, it doesn't help that SAP places these checks in a profile labeled 'New'.

SAP_NEW introduces authorization checks in areas where previously no such checks were performed

The reality is that the authorization checks introduced with every upgrade are targeted not at enabling the use of new features but closing gaps related to the execution of existing functions. In other words, they are designed to apply authorization checks in areas where previously no such checks were performed.

Given the volume of new authorization checks introduced with each upgrade and the incredible task of assigning each check to the relevant roles and profiles, SAP recommends temporarily assigning SAP_NEW to all users after an upgrade. This shortcut ensures that users are able to continue to use functions that are now protected with authorization checks and provides administrators with some breathing room to analyze and distribute the authorizations to specific roles or profiles before deleting the SAP_NEW profile. This is great in theory since the recommendations minimize the window of opportunity during which end users could exploit any excessive privilege or potential conflict in the segregation of duties while ensuring there is no disruption to the availability of SAP functions.

Users assigned SAP_NEW are often provided with authorizations that go beyond their role requirements on a permanent basis

The reality is that very few administrators follow the recommendations, especially those related to the eventual deletion of the SAP_NEW profile. The end result is that many users are effectively provided with authorizations that go beyond their role requirements on a permanent basis. The situation is even worse if administrators fail to delete the individual profiles associated with each release from the composite profile. This is the difference between being assigned new authorizations in a single upgrade and being assigned authorizations in all upgrades performed in an SAP instance. Clearly, the risk is far greater with the latter, although it may go undetected if you're reviewing access on a transaction, rather than object, level.

To determine whether this scenario exists in your environment, take a fresh look at the composite SAP_NEW profile. If you see a long list of single profiles from multiple releases or upgrades, it's time to revise your authorization concept. You should also consider using the update procedure in SU25 (report SAPLPRGN) to adjust roles and profiles following an upgrade.

Chapter Six

Breaking SAP Password Security

Password security is a standard feature of most SAP security audits. However, the focus has traditionally been upon on parameter settings such as complexity requirements, password length, expiration time, lock-outs, etc. These parameters are well known and can be checked through report RSPARAM via transaction code SA38. In this Chapter, we will discuss some of the lesser known facts about SAP passwords. In particular, password hashing and downwards compatibility. Vulnerabilities in these areas can be exploited by attackers to crack user passwords and logon to SAP applications with stolen credentials. According to some assessments, more than 90 percent of SAP systems are vulnerable to such exploits.

Let's start with some good news: SAP does not store passwords in clear text. Passwords are stored as hashes in the tables USR02 and USH02. The hashing algorithms used by SAP have evolved over time to become progressively more secure in response to vulnerabilities in prior algorithms. Each algorithm is identified by a code version using the acronym CODVN. CODVN A is the least secure, whereas CODVN I, the latest algorithm, is the most secure.

CODVN B and F are of particular interest. The first is based on a MD5 hashing scheme, which supports case insensitive passwords up to eight characters long.

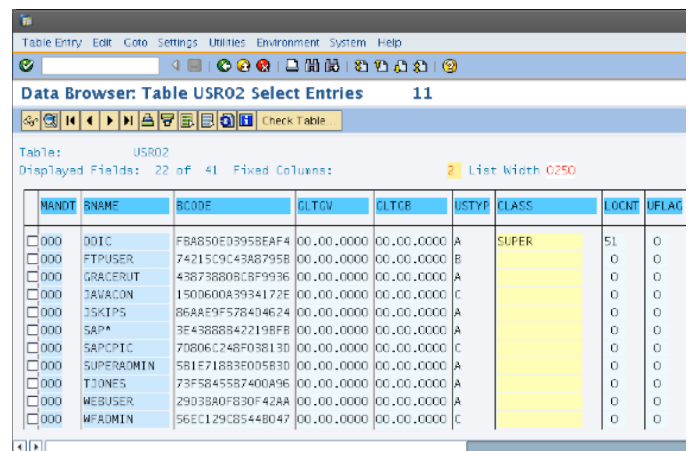
The second uses a SHA-1 hashing scheme and allows case-sensitive passwords up to 40 characters in length. When introduced in Netweaver 7.0, CODVN F represented a quantum leap in password security since it used fixed salt values.

Some SAP algorithms can be decrypted using password cracking tools

There is a common belief that password cracking is a high-tech crime. In fact, it's one of the easiest methods attackers can use to obtain user credentials. It's even more reliable and effective than social engineering. There are numerous software tools that attackers can use to decrypt hashes. One of the most popular is John the Ripper which can run on almost any platform including

UNIX, Windows and OpenVMS. John the Ripper auto-detects hashing mechanisms and can perform both dictionary and brute force attacks on DES, MD5 and other hashes. Since 2005, the tool has been able to crack CODVN B and CODVN F passwords in SAP. The patch that enables hackers to break such passwords was publicly released in 2008 and can now be downloaded from sites such as openwall.com for free.

There is a caveat to cracking SAP passwords: an attacker would need access to SE16 to extract the USR02 table (Figure 6.1) which would then be downloaded to a local file system in order to be processed by a cracking tool. However, a hacker could use one of the attack vectors outlined in other chapters of this Paper to obtain administrative privileges in SAP or alternatively, direct access to the database. Unencrypted Data backups can also be used to extract USR02. SOAP functions such as RFC_READ_TABLE provide yet another avenue since they allow any table to be read remotely.



MANDT	BNAME	BCODE	GLTGV	GLTCR	USTYP	CLASS	LOCNT	UFLAG
000	DDIC	FB850E03958EAF4	00,00,0000	00,00,0000	A	SUPER	51	0
000	FTPUSER	74215C9C43AB795B	00,00,0000	00,00,0000	B		0	0
000	GRACERUT	438738808C8F9936	00,00,0000	00,00,0000	A		0	0
000	TAWACON	1506600A3934172F	00,00,0000	00,00,0000	C		0	0
000	SKITPS	86AAE9F578404624	00,00,0000	00,00,0000	A		0	0
000	SAP*	3E438888422198FB	00,00,0000	00,00,0000	A		0	0
000	SAPCPIC	70806C248F03813D	00,00,0000	00,00,0000	C		0	0
000	SUPERADMIN	581E71883E00583D	00,00,0000	00,00,0000	A		0	0
000	T3ONES	73F5845587400A96	00,00,0000	00,00,0000	A		0	0
000	WEBUSER	29D3840F830F42AA	00,00,0000	00,00,0000	A		0	0
000	WADMIN	56EC129C85448047	00,00,0000	00,00,0000	C		0	0

Figure 6.1: Table USR02

Once the table has been extracted and processed, John the Ripper will display cracked passwords against the relevant user IDs. The attacker would then simply use the stolen credentials of dialog users to logon into SAP and execute fraudulent or malicious actions. Needless to say, the attacker is likely to target administrative and other privileged users.

Attackers can crack passwords even if stronger algorithms have been enabled

Password hashes are stored in specific fields of the USR02 table. CODVN B hashes are stored in the BCODE field, CODVN F hashes are stored in the PASSCODE field and so on. As you can see, hashes for each code version are stored in separate fields. Generally, there should only be one field in every USR02 table for password hashes. The name of the specific field used to store password hashes in the table will be determined by the code version in use.

There are some exceptions to this rule: table USR02 will contain fields for both CODVN B and F hashes if code version G is in use. This is because CODVN G isn't really an algorithm. It's a mechanism that allows the use of two different code versions at the same time. This scenario could create vulnerabilities if downwards compatibility is allowed through the login/ password downwards compatibility parameter. The default value set by SAP is 1 which means that the table will store passwords in the BCODE field using the weak CODVN B algorithm alongside the PASSCODE field which will contain the stronger CODVN F hashes. BCODE truncates the password to eight characters and converts it into uppercase. PASSCODE, on the other hand, will store the full password using upper and lower case. This is unlikely to deter hackers since most SAP passwords are no longer than eight characters and case sensitivity can be by-passed once a password is available in clear text.

In summary, the stronger security offered by later code versions through more complex algorithms can be defeated if downwards compatibility is enabled and hackers are able to target weaker hashes that are still retained in the database. This is especially a problem if a weak password policy is enforced requiring eight characters or less with minimal complexity requirements.

The most recent code version, CODVN I, also suffers from downwards compatibility by storing CODVN B and F hashes, together with the strong hashes for CODVN H which contain random salts.

As a countermeasure, SAP recommends a robust password policy, as well as restricting access to table USR02, together with other tables that store password hashes such as USH02 and USRPWDHISTORY.

Access to transactions SE11, SE16 and SE17 should also be tightly controlled.

SAP urges customers to activate the latest password mechanisms available for their release, disable downwards compatibility and delete redundant password hashes from the relevant tables.

However, downwards compatibility may be required in cases where a Central User Administration supports access to multiple systems. SAP provides a number of options for the password_downwards_compatibility parameter to deal with such scenarios. Note that if value 4 is selected for the parameter, SAP will allow successful logons against the downward compatible hashes without an entry in the system log. In the next chapter, we will reveal how this configuration can be used by an attacker to create a backdoor into SAP.

Chapter Seven

Exploring SAP Backdoors and Rootkits

Like most systems, SAP is vulnerable to backdoors and rootkits. These terms are often used interchangeably since they refer to similar threats. As the name suggests, backdoors are designed to bypass normal authentication and authorization mechanisms in systems. They can take a variety of forms from hardcoded users and passwords in programs to sophisticated malware that provides remote, privileged access to target systems while hiding the activities of attackers. The latter provides hackers with a method to compromise order-to-cash, process-to-pay, financial reporting and other processes without detection.

Attackers looking to install a backdoor in SAP can simply create a fictitious user with the SAP_ALL or similar privileged profile. However, very few are likely to choose this route. Most attackers are smart enough to know that creating a user with the SAP_ALL profile will draw attention from administrators and auditors and is likely to be quickly removed and investigated. They know that many companies monitor for such events through dashboards and other monitoring tools.

Therefore, attackers are likely to choose a more creative path to create a backdoor. Fortunately for them, they won't have to strategize for too long.

Downwards compatibility can be exploited by attackers to compromise SAP accounts without detection

In the Chapter 6, we discussed the problem of downwards compatibility for password hashes. We noted that SAP will allow successful logons against backward compatible hashes without an entry in the system log if value 4 is specified for the password_downwards_compatibility parameter (in fact, value 3 will also permit logons against old hashes but will register the event in the system log). An attacker can exploit this vulnerability and create a backdoor into SAP by modifying the downwards compatible hash of a user account. This will enable the attacker to use the compromised user account to logon to SAP. When doing so, the system will first check the password provided by

the attacker against the stronger hash. This check will fail. However, since SAP is configured to accept downwards-compatible hashes, it will perform a second check against the weaker hash. The password provided by the attacker will sail through the second check. Note that stronger hashes are untouched by attackers. Since legitimate users will continue to authenticate with SAP through passwords stored using strong hashes, they will be completely unaware that their accounts have been compromised.

SAP programs and RFC connections may contain hardcoded usernames and passwords

If attackers don't wish to create their own backdoors, they can simply use the ones provided by SAP. Some standard programs contain hard-coded bypasses for specific usernames. These backdoors were created by SAP developers, reminiscent of the backdoor in the WOPR program in War Games (1983).

Credentials are also often hardcoded into RFC connections which, if left unencrypted, can be intercepted and used by attackers. Frequently, the credentials are for users with SAP_ALL privileges.

Attackers can bypass SAP change controls and modify programs directly in the database

All standard and custom programs installed in SAP are stored in the REPOSRC database table. This includes program source code which can be found in the DATA field in compressed form. An attacker can gain access to the database by exploiting vulnerabilities in the application, operating system or database layer, such as those outlined in this Paper.

With such access, an attacker can inject a rootkit directly into an ABAP program in the REPOSRC table using SQL queries. In doing so, the attacker would bypass SAP change controls that lockdown direct changes in production environments, control

access to development tools such as the Workbench, and restrict the ability to change the dictionary or programs by requiring a developer key issued by SAP.

After injecting the rootkit, the attacker will ensure that SAP regenerates and updates the program by removing the relevant record from the REPOLOAD table. Again, this is performed through a SQL query with a simple command line. The regeneration will activate the rootkit.

Rootkits can be used to capture and change financial, customer and supplier information including credit card, bank and payroll details

Once activated, the rootkit may, for example, relay every new customer record created in SAP to a remote server controlled by the attacker, change the bank details for new vendors, modify operational and financial results, etc.

The source code for some programs is protected at both the application and database level. Within SAP, this can be verified through the ABAP Editor (SE38) which will generate an error message if a user attempts to display or change the source code of certain programs. At the database level, such programs can be identified through the SQLX field. An attacker will not be able to modify the source code for these programs even with direct access to the database using the method outlined above.

Rootkits can also forward usernames and passwords to attackers through email

One of the most critical programs in SAP is SAPMSYST. This program receives authentication data from dialog users through SAP GUI and other clients. Understandably, access to the SAPMSYST source code is tightly controlled. However, even programs such as SAPMSYST can be modified directly in the database using SQL queries that target the program name or pivot. Backdoors in SAPMSYST can be used to forward authentication data (client, user and password) to an attacker through email and other methods.

There are very few ways to detect backdoors or rootkits in SAP other than through a full review of the source code of all pro-

grams. This is an impossible task given that SAP ERP alone has over 2M standard programs, not to mention custom programs. Some of these programs have more than 50,000 lines of code. Therefore, security efforts should focus upon preventative measures.

SAP often issues Notes to patch errors in the code base that were not detected and reversed during the original round of QA. Customers should monitor and implement such patches as soon as they are released.

Also, the SAP Code Inspector (transaction SCI) should be used to check the security of new or critical ABAP code. Alternatively, there are third party tools that can be used to periodically review and detect changes to ABAP programs. These are discussed in Chapter 12.

Chapter Eight

Attacks against the SAP Java Engine

Today, many SAP functions are accessible as Web services through the Netweaver Application Server (AS). This is achieved through the ability of the Netweaver AS to support Java code that meets the J2EE (Enterprise Edition) standard, as well as SAP's own language for business applications, ABAP.

Named after the copious amounts of coffee drunk by the team at Sun during its development in the early '90s, Java is a programming language and platform that derives much of its syntax from C and C++. A defining characteristic is its portability: Java is based on the principle of 'write once, run anywhere'. As a result, programs written in Java can run on almost any platform. This is achieved through bytecode which acts as an intermediary between Java language and virtual machines known as Java Runtime Environments (JRE) installed on hosts.

Java is installed on an estimated 3 billion devices

Portability is the key to Java's success. It's so popular, it can be found on an estimated 3 billion devices worldwide including everything from Blu-ray players to parking stations.

Java is also popular with developers due to its relatively simple instruction set. It generally has far fewer instructions than native code. It's also very accessible: most colleges and universities provide courses in Java programming. Furthermore, Oracle does a great job supporting Java programmers through its online Technology Network. In contrast, ABAP is an obscure language that is expensive to master.

There is a downside to Java's success and, as is often the case, security pays the price. The accessibility and popularity of Java makes it an attractive target for attackers. The situation is worsened by inherent vulnerabilities in the platform.

Java is susceptible to reverse engineering. Simply put, application-based virtual machines such as Java are easier to reverse engineer than native applications due to their open source code. Java source code compiled into byte code is stored in .class

files. These files can be viewed and modified by anyone with some technical background in Java programming which, as we just learned, is fairly widespread. Reverse engineering can be simplified through byte code disassemblers such as IDA and Eclipse.

Moreover, since a JRE is a virtual machine that is hardware independent, there are fewer hurdles presented to an attacker attempting to gain full control over a Java program.

Java security has improved markedly over recent years. However, much of the legacy code remains for backwards compatibility. Also, while security techniques such as encryption and obfuscation can be used to mitigate attacks, there are still points of vulnerability and security is not well understood or consistently applied by programmers. Therefore, Java can be vulnerable to exploits such as XSS, session hijacking and SQL injection. In fact, there are over 1500 records for Java vulnerabilities in the National Vulnerability Database.¹⁰ Clearly, not all of these vulnerabilities are specific to Java Enterprise Edition (EE) used in SAP. Nonetheless, the results are concerning to say the least.

Unlike Java Standard Edition (SE), Java EE is designed specifically for servers and mainframes and contains libraries for developing components such as Servlets, JavaBeans and JavaServer Pages. These are used by developers to create portable and scalable applications that integrate with multiple systems.

The SAP J2EE Engine (renamed to AS Java in version 7.1) consists of three hierarchical, logical layers: Enterprise Runtime that provides the core functions of the system, Components such as interfaces that provide various runtime services and APIs, and Applications deployed on the J2EE Engine.

The Engine drives many of the newer components of the SAP landscape designed to integrate business systems. Applications such as SAP Portal, SAP Mobile, Exchange Infrastructure (XI), Process Integration (PI) and Solution Manager are used to connect or control systems processing and storing sensitive data. As such, they present a lucrative target for both internal and external attackers.

INTERNAL ATTACKS TO THE SAP J2EE ENGINE

A typical internal attack against SAP applications through the J2EE Engine is likely to begin with a basic port scan. This is likely to reveal a dozen or so open ports including the following:

- 5NN00 – Web Server
- 5NN04 – Visual Administrator (replaced by the Netweaver Administrator in the v7.2)
- 5NN08 – J2EE Telnet

These services are used to manage everything from users to system configuration, often through default users such as Administrator and J2EE_ADMIN. Although there are no default passwords for such users, the Web Server and J2EE Telnet transmit authentication data in clear text. Therefore, an attacker can simply sniff the traffic flow to obtain the credentials required to compromise these accounts and control the J2EE Engine.

The Visual Administrator (VA) will present the attacker with a greater challenge since it transmits passwords in encrypted form. However, the password algorithm has been reversed by security researchers and appears to be a variant of base64 encoding. VA can be used to manage interfaces, libraries, modules and services in web-enabled SAP applications.

This attack vector can be prevented by disabling unnecessary services, restricting access to open ports and enabling SSL encryption between server connections. In our experience, such advice is rarely followed, especially SSL encryption since it's often associated with a drag in performance.

EXTERNAL ATTACKS TO THE SAP J2EE ENGINE

Although internal hackers are usually presented with a wide number of entry points to SAP applications, they are prone to detection.

External hackers enjoy the advantage of anonymity. Traditionally, the downside for external attackers was the difficulty in reaching internal SAP systems. To access the VA, for example, an attacker needs to use the P4 protocol over port 5NN04. This can only be reached internally.

However, accessibility is less of a problem today. Contrary to popular belief, many SAP systems are now connected to the Internet and discoverable through search engines such as Google or specialized engines such as Shodan. Figures 8.1 and 8.2 provides a sample of Shodan search results and Google hacking strings used by attackers to locate SAP targets in the J2EE Engine.

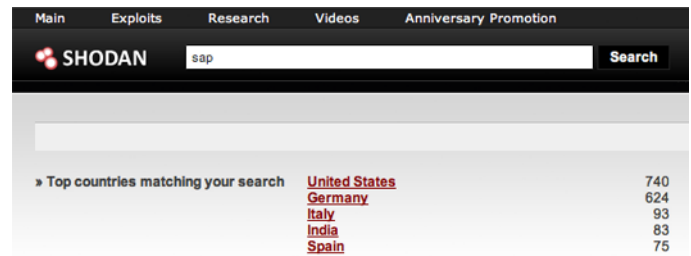


Figure 8.1: Shodan Search Results

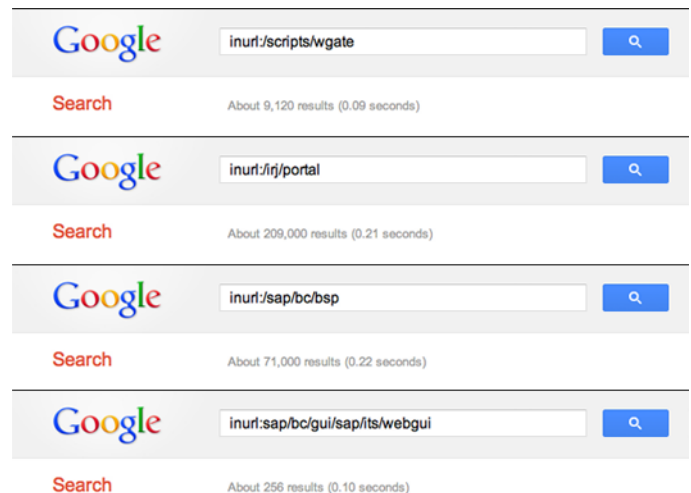


Figure 8.2: Google Search Results

Once discovered, an attacker can trigger error messages to fingerprint SAP components and analyze responses to gather information such as host names, SSIDs, system numbers, application versions and IP addresses

XSS can lead to session hijacking, data leakage and denial of service

The attacker can then attempt to launch a series of well-known cross-site scripting (XSS) attacks against components of the hundreds of default Java applications installed on the Netweaver AS. According to SAP, the effect of XSS attacks can include session hijacking, data leakage through the redirection of form input such as credit card and personal information, key logging and denial of service.¹²

MITM attacks can be used to obtain administrator credentials

The attacker can also execute an SMB Relay attack against Windows-based SAP servers. SMB Relay is a program used to execute Man-in-the-Middle (MITM) attacks through methods such as DNS poisoning and has been successfully tested against J2EE applications such as Meta Model Repository (MMR). It is often used by hackers to obtain administrator credentials which are then used to remote shell into target systems with privileged access.

Yet another avenue available to a hacker is the Invoker Servlet attack. A servlet is like a Java applet, except that it runs on a server instead of a browser. Servlet security is defined in the web.xml file located in the web.inf directory. The web.xml file specifies which resources can be served publically and which resources are private, accessible only by designated groups. The file also controls which servlets are available for end users since not all servlets are intended for direct client access. Many are designed for background processes.

Custom functions designed to process customer orders, update pricing, or transmit accounts payable and payroll information can be called directly without authentication

An attacker can bypass the authentication defined in the web.xml file through a HTTP request that directly calls a servlet by its servlet name or fully qualified servlet class name instead of using its URL mapping. This is known as the invoker servlet. Furthermore, the attacker can call servlets not declared in the web.xml file. The risks associated with this vulnerability should not be underestimated. It can provide attackers with unauthenticated access to critical functions in Enterprise Portals, Mobile, Process

Integration, Solution Manager and other SAP applications. This may include, for example, custom servlets designed to process customer orders, update pricing, or transmit accounts payable and payroll information.

The invoker servlet was developed by programmers at SAP to rapidly prototype and debug systems. SAP's standard code reviews failed to detect and remove the servlet before release.

The invoker servlet should be disabled immediately by changing the value of the EnableInvokerServletGlobally property of servlet_jsp on the server nodes to False. You should also update your security patch level.¹² There are patches that protect SAP against basic SMB Relay attacks. However, such attacks are difficult to guard against when combined with XSRF or XSS.¹³

For detailed guidance on countermeasures for session handling, XSRF, SQL injection, directory traversal, XSS and other vulnerabilities affecting SAP applications, you should follow security recommendations documented in SAP's White Paper Protecting Java and ABAP-Based SAP Applications Against Common Attacks. The recommendations are SAP's response to the rising tide of security threats faced by its product suite. However, they fail to adequately address one of the most critical vulnerabilities in the J2EE Engine related to Header Variable Authentication.

This vulnerability was first documented by Dr Jorg Wulftange in an article on the SAP Developer Network published in 2006.¹⁴ The article was intended to outline procedures for implementing third party Web Access Management (WAM) solutions such as RSA ClearTrust, CA Siteminder and Oracle Oblix but, in the process, stumbled upon a massive security flaw in the authentication model.

Attackers can bypass identity management systems

Header Variable Authentication delegates the verification of users, passwords and other factors that are part of an authentication scheme to an external WAM. When a user is successfully authenticated, the WAM directs the user to the J2EE Engine with a HTTP request that contains the logon name of the user in the header. There is no additional password authentication performed by the Engine. The flaw in the model arises from the fact that an attacker can send a HTML request directly to the Engine

listening on `http://j2eehost:5000/irj` without authenticating through the WAM. The Engine will then issue a SSO logon ticket in the form of a MYSAPSSO2 cookie directly to the attacker. This will provide access to applications supported by the Engine including the SAP Portal, Mobile, XI, PI, etc. The attacker may also get access to backend applications such as FI, CO, MM and HR that have trust relationships with the Engine, as well as install backdoors to secure future access in PAR files that contain Portal applications.

The vulnerability in the Header Variable Authentication model can be mitigated by firewall rules that control direct connections to the J2EE Engine. Such rules will verify the IP addresses of incoming HTTP requests against a list of trusted sources. Another option is configuring SSL which enables mutual authentication between clients and servers. However, SSL can be more difficult to configure and may impact system performance. It will also require ongoing maintenance since certificates have to be periodically renewed to retain their validity. ¹⁵

Chapter Nine

Managing Vulnerable SAP Web Services

In the previous Chapter, we discussed vulnerabilities arising from SAP's evolution from an internal system, developed exclusively through the propriety ABAP language to a more open system with many functions accessible to internal and external users through the Web. Driven by business needs, SAP adapted its software to improve user accessibility and information exchange between systems, referred to by system engineers as interoperability.

Java is an important component of SAP's strategy to develop more accessible and interoperable systems using open source standards. However, the J2EE platform is a work in progress. In the words of SAP "it will take some before the Java/J2EE platform offers the performance and reliability that the ABAP environment has already had for a long time".¹⁶ Business programs such as FI, CO and MM are still largely driven by SAP's ABAP platform.

Both the ABAP and J2EE platforms are components of the Netweaver AS which evolved from the technical Basis component to a powerful middleware engine at the core of SAP's technology. The Netweaver AS integrates users and systems through a variety of interfaces and protocols.

Web-based HTTP(S) communication requests to both the ABAP and Java stacks are processed by another component of the Netweaver AS: the Internet Communication Manager (ICM). Web-enabled services are defined in the Internet Communication Framework (ICF) accessible through transaction SICF.

Prior to the introduction of the ICM in version 6.10, HTTP requests flowed through the Internet Transaction Server (ITS) including a web filter known as the Wgate and a translator known as Agate. The ICM enabled SAP to process HTTP requests directly using URL handles without the use of middleware such as the ITS.

Attackers can analyze standard error pages to identify SAP targets

In common with the J2EE engine, the ICM generates standard error pages when a non-existing URL is requested or incorrect credentials are submitted. This includes HTTP 404 Not Found and 403 Forbidden which can disclose sensitive information to the requestor about the target system such as hostname, SSID and system number (refer to Figure 9.1). This can be fixed through customized error pages. The SAP Help Portal provides detailed instructions on how to create such pages.¹⁷

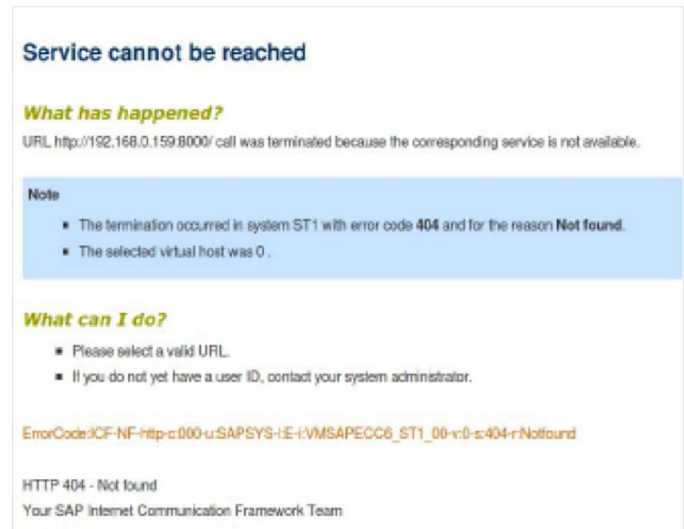


Figure 9.1: Default ICM 404 Error Message

Default ICF services can be exploited to access sensitive functions, often without authentication

The ICM draws upon services enabled in the ICF. Many of the services enabled by default have known security issues and may be exploited to allow unauthorized access to critical system functions. Therefore, services should be reviewed and disabled if they don't serve business needs or don't need to be accessible from the Web. This should include services in /sap/public which don't require any user authentication and services with hard-coded logon data. It should also include the following specific services: echo, FormtoRFC, xrfc, webrfc, IDoc and IDoc_XML.

For a complete list, refer to the SAP security recommendations in Secure Configuration, SAP Netweaver Application Server ABAP.

The services can be disabled using SICF (Figure 9.2). Disabled services are displayed in grey, whereas active services are blue. Remember to analyze and test services before any deactivation using the ICMAN server log. There are often dependencies between services and disabling some services could lead to errors in others.



Figure 9.2: SICF

soap/rfc can be used to obtain full-blown command over the operating system and database

The ICF service `/sap/bc/soap/rfc` deserves a special mention. In fact, this service is so dangerous, SAP devoted an entire note to it.¹⁸ A vulnerability in the TH_GREP function module could be exploited by an attacker to obtain administrative access to UNIX operating systems and therefore, full-blown command over the database. TH_GREP is part of the task handler suite of function modules and is designed to search for strings in SAP log files stored on the OS. An attacker can inject the following command into the search string used by TH_GREP and then export the DISPLAY to a terminal emulator such as xterm to acquire shell access to SAP (Figure 9.3):

```
`export DISPLAY=<IP address> @ @ xterm`
```

Terminal emulators provide remote access to applications and resources running on other machines. This attack can be performed through a SOAP request if the `/sap/bc/soap/rfc` service is activated. SOAP is an XML based protocol supported by SAP that is used for the exchange of structured information through the Web. To make matters worse, the TH_GREP function module can be executed by any user with access to SE37. This includes the EARLYWATCH user. As we discussed in Chapter 4, administrators often fail to change the default password for this user. TH_GREP is also accessible through SM51.

The remote OS command injection vulnerability was originally discovered in UNIX systems such as AIX. Researchers subsequently found the same vulnerability in the Windows platform.

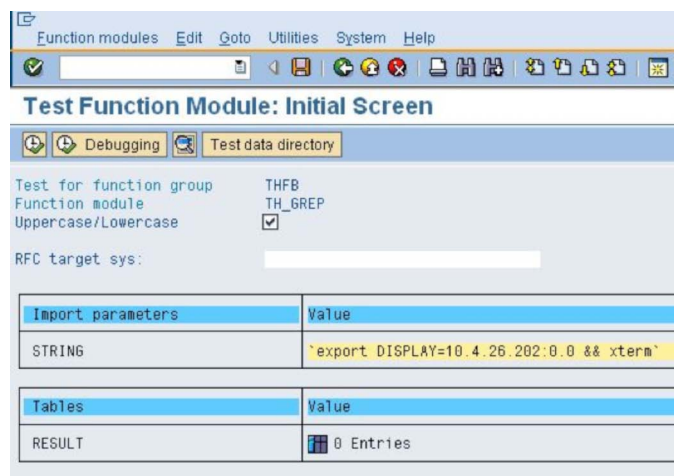


Figure 9.3: TH_GREP

Chapter Ten

Examining Vulnerabilities in SAP GUI and Web Clients

Despite the availability of Web interfaces, SAP GUI remains the most popular means of connecting to SAP servers. It's a standard application installed on workstations and a vital element of SAP's three tier client, server and database architecture.

Given the prevalence of SAP in today's businesses, SAP GUI has almost as many installations in corporations as Internet Explorer and Microsoft Office. However, there's a crucial difference between these applications: SAP doesn't patch and support its product nearly as well as Microsoft.

Furthermore, security professionals are far less informed about SAP GUI vulnerabilities than IE, Office and Windows issues.

Buffer overflows can lead to denial of service, enable attackers to take control of client workstations and compromise SAP usernames and passwords

Some of the earliest known vulnerabilities in SAP GUI were discovered in 2008 by researcher Luigi Auriemma who uncovered several buffer overflow flaws in the Windows SAPlpd daemon, a printer service available on port 515. ¹⁹ The vulnerabilities can be exploited by an attacker to provoke denial of service and obtain administrative access to user workstations. With such access, an attacker can install Trojans or sniff user credentials. An attacker can also obtain usernames and passwords directly from the sapshortcut.ini configuration file since SAP provides users with the ability to store passwords in their local directories for auto logon. ²⁰ This will provide direct access to SAP servers.

Buffer overflows were also found by researchers Mark Litchfield and Alexander Polyakov, effecting some of the thousands of ActiveX controls embedded in SAP GUI used to read and write files, execute programs, and connect remotely to SAP servers. Many of these flaws were rated highly critical by vulnerability assessors. ²¹

Client Scripting is vulnerable to espionage and the theft of user credentials

SAP GUI includes a scripting Application Programming Interface (API) that communicates and interacts with servers in the same way as an end user. Although the API is disabled by default, it is enabled by companies that find it useful for automating repetitive tasks, as well as server-side testing and client-side integration. Many security professionals don't see any risk associated with the use of the scripting API since it uses the same credentials provided to the end user and therefore, can only execute transactions granted to the underlying user. However, the script can be used to log end user interaction with SAP GUI. Consequently, it's vulnerable to corporate espionage especially if an attacker has disabled the security warnings automatically generated by SAP GUI when a script is executed in the background. This will lessen the risk of detection.

Also, since scripts can contain logon data and are stored in unencrypted form in local files, attackers are provided with another avenue to steal credentials from compromised client workstations.

Web-based access is growing

Web-based connections to applications such as the Enterprise Portal (EP), Supplier Relationship Management (SRM) and Customer Relationship Management (CRM) through Internet browsers are increasingly common in the SAP landscape. These connections are managed by the Netweaver AS and offer even less client-side security than SAP GUI.

There are a number of attack vectors for SAP Web clients. One of most dangerous effects a program called cFolders which is a Web-based platform that enables business partners to share information and collaborate on documents. The platform uses cookies to authenticate users and is fully integrated into SRM, Product Lifecycle Management (PLM) and Knowledge Management (KM).

Business partners can gain access to classified information and even ERP applications such as FI, CO and MM

cFolders is vulnerable to a form of Cross-Site Scripting known as HTML injection that can lead to the theft of administrator credentials. One of simplest ways to execute this attack is through SRM. This application allows any user to create and store HTML documents in specific folders. Note that given the nature of the application, a user is often a supplier or other partner. The partner can create a HTML document injected with a trojan designed to capture the cookie of any user that accesses the document in cFolders. Since user sessions are not tied to IP addresses, the partner can then use the cookie to access and modify the documents of other partners and administrative functions of the system.

Another variant of the attack can be used to obtain the credentials required to access the central ERP instance. If the document is opened by an employee of the target company, the script can attack the vulnerable SAP GUI ActiveX components in the user's workstation.

Web clients are susceptible to phishing attacks designed to capture usernames and passwords

Hackers can also target the standard Web interface used to access SAP with phishing attacks. Methods such as email spoofing can be used to lure users to phony logon screens that appear identical to the standard screen in Figure 10.1. According to some studies, up to 50 percent of users that receive a spoof email with malicious hyperlinks fall victim to phishing that enables attackers to capture their usernames and passwords.

Sophisticated, convincing websites can be created by attackers using MITM Phishing Kits that provide simple, easy-to-use interfaces.

Responses to phishing should include a combination of social and technical components. Along with user education, you should examine implementing SSL/TLS to enable users to identify legitimate sites, URL filtering to block malicious sites, and upgrading, hardening and regularly patching Internet browsers.

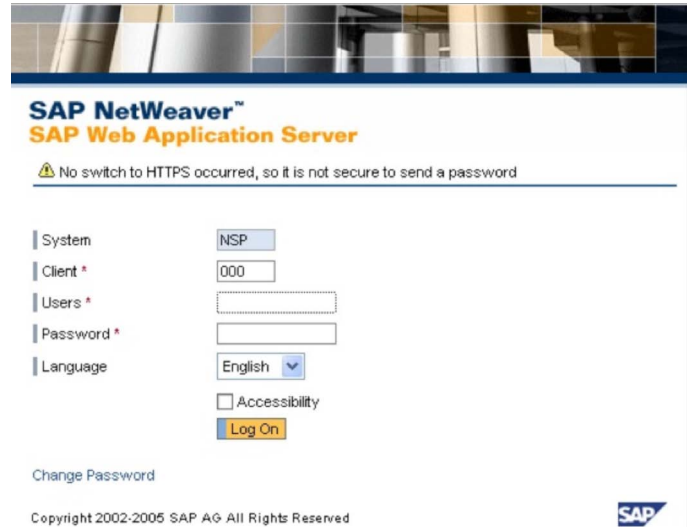


Figure 10.1: Web GUI

SAP GUI should also be patched or upgraded to address many of the buffer overflow vulnerabilities outlined in this chapter. The majority of files in SAP GUI 7.20 are digitally signed by SAP. Modification breaks the signature, enabling the identification of compromised files.

Another option is to host SAPGUI on servers rather than workstations using desktop virtualization. Citrix is generally regarded as the leading solution provider in this space.

Finally, we recommend disabling SAP GUI scripting if it has been enabled. At the very least, you should consider restricting the use of scripting APIs to designated groups using SAPAdmin. For more information, refer to SAP Note 480149 and the SAP GUI Scripting Security Guide issued by SAP.

Chapter Eleven

The SOX and PCI Implications of SAP Vulnerabilities



The central theme of this Paper is that standard approaches to audit and secure SAP do not address many of the vulnerabilities confronted by today's systems. The generally accepted methods used by professionals to assess SAP security should be enriched to manage newly identified and emerging risks, as well as respond to lesser known dangers, previously overlooked by the community.

A comprehensive vulnerability assessment of the technical layer incorporating the use of commercial or open source tools will

manage threats to the confidentiality, integrity and availability of information within SAP systems. It will also enable companies to comply with regulatory and industry standards for internal controls such as SOX and PCI.

In this section, we will discuss the compliance implications of the specific vulnerabilities presented in this Paper to illustrate the importance of such an assessment and enable organizations to map countermeasures to specific requirements.

COBIT	IT CONTROL OBJECTIVE FOR SOX	OS Authentication	Remote Function Calls	SAP Gateway and Message Server	Default SAP Users	SAP_NEW	Password Hashing	Backdoors and Rootkits	SAP Java Engine	Web Services	SAP Clients
AI2	Acquire and maintain application software				●			●			
AI4	Enable Operations		●	●		●		●	●	●	
AI6	Manage changes	●	●	●	●		●	●	●	●	
DS1	Define and manage service levels		●	●		●		●	●	●	●
DS5	Ensure system security	●	●	●	●	●	●	●	●	●	●
DS9	Manage the configuration	●	●	●	●	●	●	●	●	●	●
DS8	Manage problems and incidents	●	●	●	●	●	●	●	●	●	●
DS11	Manage data	●	●	●	●	●	●	●	●	●	●

Figure 11.1: IT Control Objectives for SOX

IT CONTROL OBJECTIVES FOR SOX

Developed by the IT Governance Institute in 2004 and revised in 2006, the IT Control Objectives for SOX set the benchmark for technology controls required by public companies to meet standards of financial reporting and corporate governance implemented in countries across the globe. It defines twelve wide-ranging standards, drawn primarily from the COBIT framework.

Figure 11.1 provides a high level mapping between the SAP vulnerability areas and most of the IT Control Objectives for SOX. The pervasive impact of such vulnerabilities is self-evident, especially in the areas of system security, configuration management, problem and incident management and data management. We can illustrate the impact by looking at one particular example: SAP backdoors and rootkits, which can bypass controls designed to:

- Ensure the integrity of application controls through SDLC standards (AI2);
- Operate applications and systems in accordance with policies and procedures (AI4);
- Control program changes (AI6);
- Maintain expected service standards (DS1);
- Control system access (DS5);
- Preserve configuration settings (DS9);
- Detect and resolve problems and incidents (DS8); and
- Protect stored or transmitted data from unauthorized access or modification (DS11).

IT CONTROL OBJECTIVE FOR SOX	OS Authentication	Remote Function Calls	SAP Gateway and Message Server	Default SAP Users	SAP_NEW	Password Hashing	Backdoors and Rootkits	SAP Java Engine	Web Services	SAP Clients
Install and maintain a firewall configuration to protect cardholder data			●					●	●	
Do not use vendor-supplied defaults for system passwords and other security parameters	●	●	●	●	●		●	●	●	●
Encrypt transmission of cardholder data across open, public networks		●	●					●	●	●
Use and regularly update anti-virus software or programs								●	●	●
Develop and maintain secure systems and applications	●	●	●	●	●	●	●	●	●	●
Restrict access to cardholder data by business need-to-know	●	●	●	●	●	●	●	●	●	●
Assign a unique ID to each person with computer access	●	●	●	●	●			●	●	
Track and monitor all access to network resources and cardholder data						●	●		●	●
Regularly test security systems and processes						●	●		●	●

Figure 11.2: PCI DSS

PAYMENT CARD INDUSTRY DATA SECURITY STANDARD (DSS)

The PCI DSS applies detailed prescriptive information security controls to companies that store, process or transmit credit card data.

There are twelve specific requirements in the Standard organized into six broad objectives related to network security, the protection of cardholder data, vulnerability management, access control, monitoring and governance. There are multiple controls within each requirement, making PCI DSS one of the most expansive frameworks in information security.

The intersections between the SAP vulnerability areas and nine of the twelve PCI requirements are illustrated in Figure 11.2.

Presuming SAP systems are not segregated from cardholder data environments through network segmentation, the impact is very high, particularly in areas such as the development and maintenance of secure systems and applications, restricting access to cardholder data and changing vendor-supplied defaults for passwords and security parameters.

These areas are heavily affected by vulnerabilities that evade authentication and authorization controls, exploit weak default settings in application and OS layers, and target data during transmission and in the database.

Chapter Twelve

The SAP Security Toolkit: Software, Audit Programs and Web Resources

SOFTWARE

There are a number of software tools including open source programs that companies can leverage to audit and secure various areas of SAP systems. Some are custom tools, designed expressly for SAP. Others are generic but can be used to detect specific flaws in SAP systems when combined with the appropriate plugins.

Open source programs should be handled with care. Since the source code is open to everyone, they can be infected with Trojans and other malware. Also, you should be aware that some of the tools listed below can potentially crash SAP systems or worse if they are used by inexperienced professionals unfamiliar with vulnerability assessment and penetration testing techniques using a command line interface. Few provide any form of support and user guides are, for the most part, non-existent.

Sapyto and Bizploit are capable, open source tools that can be used to perform pen tests for SAP using RFC connectors. Prerequisites for these tools include RFCDSK which must be downloaded and installed from the SAP Service Marketplace, Python v2.5 or higher with development libraries and GCC runtime and utilities.

WEBXML Checker is a tool that can test for certain vulnerabilities in SAP J2EE applications including verb tampering and the invoker servlet bypass.

Integrity Analyzer and Static Source Code Analyzer can perform SAP code comparisons and therefore can potentially detect backdoors and rootkits in ABAP programs.

Burpsuite and Nmap are generic web application assessment platforms which can be used for port scanning, ping sweeps and the detection of host services.

Most of these tools are available on the Web for both Linux and Windows.

PROGRAMS

Publicly-available audit programs for the technical components of SAP are limited and do not address many of the areas discussed in this Paper. ISACA has published several guides designed to test components of SAP Basis which is now part of the much broader Netweaver platform. Although imperfect, these and other programs serve some purpose by covering many of the critical administrative functions, configurables, and database and operating system services in SAP.

WEB

Security Notes are available at the SAP Service Marketplace. Be warned, Notes are often deliberately vague since SAP doesn't want to broadcast the technical details of many vulnerabilities which can be exploited by attackers before systems are properly patched. Nonetheless, you should periodically review the site for critical bulletins and patches.

The SAP Help Portal is a great resource for security guides and other SAP documentation.

The SANS Institute, OWASP and NIST provide valuable information on the latest security trends and vulnerabilities. However, for information and advisories related specifically to SAP security, you should follow the Layer Seven Security blog which provides an insight into critical SAP vulnerabilities based on cutting-edge research performed by professionals across the world.

SAPSCAN

The challenge of auditing and securing SAP systems to respond to threats such as those outlined in this Paper should not be underestimated. The technical complexity of many of the areas is not only high but constantly evolving. It is unrealistic to expect security administrators and auditors to effectively manage the hundreds of intricate vulnerabilities in today's expanding SAP environments without support.

SAPSCAN is an automated vulnerability assessment performed by Layer Seven Security that provides an efficient and cost-effective alternative. It enables companies to combat fraud, reduce audit costs and rapidly assess the security profile of their SAP systems against compliance requirements and SAP recommendations. Our team of highly experienced security professionals leverage patent-pending commercial software that examines over 300 technical security risks in both the ABAP and Java components of Netweaver. This includes:

- Insecure default configurations
- Dangerous active services
- Unauthorized remote command executions
- Information disclosure
- Use of unencrypted interfaces
- Improperly applied security filters
- Broad administrative user privileges
- Weak access credentials
- Missing SAP Security Notes and patches

The software draws upon the industry's largest knowledge base of SAP security threats and is constantly updated by a world-renowned research lab to counteract new vulnerabilities. It is certified by SAP for integration with Netweaver and is employed by many Fortune 500 companies and government organizations worldwide. SAPSCAN delivers a clear and concise report to stakeholders that effectively conveys the business impact of technical risks and provides detailed recommendations to remediate vulnerabilities.

To learn more or schedule a SAPSCAN, email sapscan@layersevensecurity.com or call 1 888 995 0993.

ABOUT LAYER SEVEN SECURITY

Layer Seven Security specialize in SAP security. We serve customers worldwide to protect information assets against internal and external threats and comply with industry and statutory reporting requirements. The company fuses technical expertise with business acumen to deliver unparalleled vulnerability assessment, audit and consulting solutions targeted at managing risks associated with contemporary SAP systems.

Our consultants have an average of ten years of experience in field of SAP security and proficiency in regulatory compliance including Basel II, GLBA, HIPAA, FISMA, PIPEDA, PCI DSS and SOX.

The company is privately owned and headquartered in Toronto, Canada.

www.layersevensecurity.com

ENDNOTES

1. The ERP Security Challenge, CSO Online, 2008
2. Denial of Service, Header Injection/ CRLF, Improper Error Handling, and Malicious File Execution
3. Office of the Senate Sergeant at Arms, 2010
4. Joint Study on Canadian Security Practices, Rotman School of Management, University of Toronto, 2010
5. 2011 Data Breach Investigations Report (DBIR)
6. Defining the Next Generation Firewall, Gartner RAS Core Research Note G00171540, 2009
7. CF Disclosure Guidance: Topic No. 2, Securities and Exchange Commission, October 2011
8. Refer to SAP Note 186119
9. Refer to Notes 1003908, 1003910, 1003910, 1004084 and 1005397
10. nvd.nist.gov
11. Protecting Java and ABAP Based Applications Against Common Attacks, SAP, December 2010
12. Refer to SAP Notes 1467771 and 1445998
13. Refer to SAP Note 1483888
14. Securing Header Authentication, SAP Developer Network, 2006
15. help.sap.com/saphelp_nw70ehp2/helpdata/en/d0/a3d940c2653126e10000000a1550b0/frameset.htm
16. sdn.sap.com/irj/scn/index?rid=/library/uuid/b161a590-0201-0010-5590-91fa5076a914#q-4-1
17. help.sap.com/saphelp_nw73/helpdata/en/48/45acaf43a64bb8e10000000a42189b/frameset.htm
18. SAP Note 1394100 – Access to RFC enabled modules via SOAP. Also refer to Notes 1433101 and 1580017
19. aluigi.altervista.org/adv/saplpdz-adv.txt
20. Refer to SAP Note 146173
21. computerweekly.com/news/2240081941/ActiveX-security-flaws-plague-SAP-GUI