International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Securing Sensitive Data in SAP: A Comprehensive Guide to Security Testing

Sireesha Perabathini

Independent Researcher
Illinois, USA
perabathinisireesha@gmail.com

Abstract

SAP (Systems, Applications, and Products) is the most used enterprise resource planning (ERP) solution companies worldwide to control key business functions. Keeping data secure within SAP environments is essential, as cyber threats, unauthorized access, and data breaches are inherent risks. This paper will guide you through the process of proactively testing SAP security, exploring tools and best practices, examining case studies, and understanding the process of conducting tests within SAP systems. This paper is designed to equip organizations with the knowledge to protect their SAP infrastructure before potential threats arise.

Keywords: SAP, Security Testing, Penetration Testing, Data Security, SAP Security Audit, Risk Analysis, Vulnerability Testing, and Vulnerability Analysis

I. Introduction

SAP (Systems, Applications, and Products) is the biggest enterprise software in the world, and it's used extensively for handling business processes such as finance, Human Resources, Supply Chain, and Customer relations. With SAP adoption, finding high-value and confidential organizational information in the SAP environment is common.

Sensitive data like client's personal information, bank accounts, and patents have key business value. If compromised, this information could lead to significant financial losses, harm to reputation, and potential legal consequences. SAP Security testing is required to maintain integrity and keep sensitive data safe in SAP systems [6].

This paper will help you to achieve this by covering the process of securing sensitive data in SAP by highlighting security testing techniques, tools, best practices, and real-life examples. It's geared towards helping companies identify risks early on and implement risk mitigation measures.

II. SAP Data Security challenges

SAP data is enormous, from customer details to accounting and business processes. This data can be exposed or misused in unethical ways. The risks associated with SAP systems are diverse and significant, such as Cyberattacks against SAP vulnerability, Permitted Access employees or contractors who abuse their privileges, Poor authentication, which results in unauthorized access, and Lack of encryption/masking for confidential information.

Attackers may exploit compromised user access to sensitive information, making SAP security a critical concern for organizations.

III. Key Security Principles of SAP

A. *Secrecy, Transparency, and Provision (CIA Triad)*

Data security starts with the CIA trifecta:

i. Data security: Only authorized personnel can see sensitive information.

ii. Integrity: Data is reliable and unmodified.

iii. Accessibility: Allows only those users to whom the data pertains to access it in times of need.

B. *Authentication and Authorization*

Organizations need to implement correct authentication (MFA, Multi-factor Authentication) and authorization (Role-Based Access Control) to avoid unauthorized access.

C. *Data Encryption and Masking*

Secure data both during transmission and while stored by encryption and masking data so that organizations can hide sensitive data but still allow businesses to conduct legitimate business.

D. *Auditing and Monitoring*

Regular audits and monitoring of user activity in SAP systems are very important to avoid a potential breach and to follow security policies.

IV. Overview of SAP Security Testing

A. *Common SAP Security Vulnerabilities and Their Mitigation*

TABLE I. SAP Security Vulnerabilities Vs Mitigation

Vulnerability	***Description***	***Mitigation Method***
Weak Authentication	Users can access SAP systems without strong validation mechanisms.	Implement Multi-Factor Authentication (MFA).
Excessive User Permissions	Users have more permissions than required for their roles.	Apply Role-Based Access Control (RBAC).
Data Leakage	Confidential data is exposed due to a lack of encryption.	Implement encryption for both in-transit and at-rest data.
Poor Patching and Updates	Systems are vulnerable to known exploits due to outdated patches.	Apply security patches regularly.

B. *Types of Security Testing*

SAP RBAC Testing SAP Role-Based Access Control (RBAC) Test: The SAP Role-Based Access Control (RBAC) test is performed so that users have the right roles and are allowed only to use the correct transactions and data in the SAP system for their role. This consists of verifying the roles and privileges of users to see if they have been allowed only the access they require. They follow the Principle of Least Privilege (POLP) which makes users limited to the tasks related to their role. The end product of this testing is the detection of instances where the misuse or overuse of roles and permissions results in an inappropriate increase of privileges for the user at the expense of security.

SAP Authorization Testing: Focuses on the specific permissions granted to users verifying that access is properly configured to allow or restrict specific actions or data. This process involves evaluating the roles and privileges of users by testing access to specific transactions and data, ensuring that permissions are correctly configured to prevent unauthorized actions. The Principle of Least Privilege (POLP) is applied so that users can only do tasks relevant to their job. This testing type identifies misconfigured authorizations that may result in unnecessary access.

SAP Security Configuration Testing: This test aims to protect the SAP system from unauthorized access and attacks [6]. This testing will review configurations for security settings, patches, network access, and authentication (e.g., SSO or Kerberos). It helps identify mistakes such as weak passwords, open ports, or services that don’t need to run and recommends fixes to strengthen the SAP environment’s security.

SAP SOD(Segregation of Duties) Testing: The purpose of this testing is to ensure that no user has conflicting roles which permit them to commit fraud by gaining access and control over contradictory business processes. We can achieve this objective by evaluating user roles to see if users can’t perform both sides of key transactions (like initiating and approving a financial transaction or purchasing and paying). The result of this evaluation is the enforcement of SoD policies which help to reduce fraud, mistakes, and phishing.

Testing of SAP System Log and Audit Trail: The objective of this testing is to ensure that the system activity is correctly recorded and that there is a robust audit trail for identifying and investigating suspicious activity. This is achieved by implementing the SAP logging and monitoring functions to log and review all events (role changes, login failures, data access). The result is the assurance of validity and completeness of logs and audit trails for incident management and forensic investigations.

SAP Vulnerability Analysis and Penetration Testing: The goal of this process is to exploit potential security gaps in SAP. This is achieved using automated vulnerability scanners and manual penetration testers who search for vulnerabilities such as old patches, weak setups, and common vulnerabilities like XSS, SQL injection, and buffer overflow. The result is an executive summary of identified flaws and suggestions on patching, configuring, or mitigating.

Security Test of SAP Fiori and UI Security: The purpose of this process is to test the security of SAP Fiori applications and user interfaces (UIs), which can be accessed by other users and possibly attacked. This is done by evaluating SAP Fiori apps for vulnerabilities such as Insecure direct object references (IDOR), XSS, and improper input validation. This results in the identification of security vulnerabilities at the front end of the Fiori app that could be used by a rogue user to manipulate data or perform unauthorized actions.

SAP Gateway & Web Services Security Test (Sansa gateway/web services): The focus of this process is to ensure the SAP Gateway and the web services (SAP NetWeaver, SOAP, OData) are protected from unauthorized use and data leaks. This is achieved by implementing strong security measures to safeguard these services from unauthorized use and data leaks. The result is the maintenance of securely configured web services and APIs exposed to external systems against unauthorized use and data capture.

SAP Data Protection and Encryption Test: Aims to ensure that the sensitive data in SAP systems is encrypted both during storage and during transport. This is accomplished by enabling encryption settings on sensitive data (personal or financial data) and have data encrypted using SSL/TLS. This process assures that proper encryption is done, eliminating any data breaches and theft risks.

SAP Patching and Update Testing: The purpose of this process is to verify that SAP systems are protected by the most current security patches and updates for security gaps. This is done by verifying that SAP uses the latest software and that patches are updated on all components. This will result in the validity and completeness of logs and audit trails for incident management and forensic investigations.

SAP Secure Transport Layer (SSL/TLS) Testing: This process aims to enable secure communication between SAP systems and external systems (e.g., SAP and web browser, mobile application, third-party system). Ensures that the SSL/TLS settings are tested thoroughly and SAP apps communicating with users/systems outside SAP apps are encrypted for potential interception. This enables data security on the go while reducing the chances of interception or tampering.

SAP Cloud Security Testing: The purpose of this process is to make sure that SAP systems running in the cloud are secured and follow cloud security best practices. This involves evaluating SAP in the cloud configuration, cloud storage security, network configuration, and data protection. The result is a comprehensive assessment of the risks associated with multi-tenant cloud environments and ensure the SAP system follows security policies for the cloud.

SAP API Security Testing: The focus of this process is to maintain the APIs inside the SAP landscape (e.g., SAP PI/PO, SAP API Management) safe and free from attack. Make sure to scan the APIs for common problems, like weak authentication, leaks, or faulty authorization. The outcome is a secures communication between SAP and third-party systems to prevent data loss.

SAP Authentication and SSO Testing (SAP-SSO): The goal is to ensure secure authentication in SAP operations, through SAP (Single Sign On) by scanning for authentication flaws, including weak password policy, session management, or non-existence of MFA.Te result is to keep secure authentication practices in place, and unauthorized access at bay.

V. SAP SECURITY TESTING APPROACHES AND PROCEDURES

A. *Static Application Security Testing (SAST) – What is it?*

SAST includes reading SAP applications' source code to discover bugs without running the program. It finds the problem at a very early stage of development.

B. *Dynamic Application Security Testing (DAST)*

DAST validates SAP applications at runtime — imitating real attacks — to find out whether an application could be compromised during execution.

C. *Penetration Testing for SAP*

Penetration testing — Simulation of SAP attack to find the flaws that may lead to data breaches or intrusion.

D. *Vulnerability Scanning and Remediation*

Automatic vulnerability scanners scan SAP systems for vulnerabilities. The vulnerability needs to be remedied when detected.

E. *Threat Modelling and Risk Analysis.*

Threat modeling determines which security threats to look out for, risk assessments determine the probability and severity of those threats and determine security priority.

Risk=Threat Likelihood × Impact Severity

Where:

- Threat Likelihood is a value representing the probability of a security breach (e.g., 1 to 5 scale).
- Impact Severity is a value representing the consequence of a breach (e.g., 1 to 5 scale).

VI. HOW TO PROTECT SENSITIVE DATA IN SAP

Data classification and sensitivity analysis involve defining data by severity, which helps organizations implement the appropriate security measures to protect sensitive information. Secure Development Lifecycle (SDLC) practices focus on writing safe code and conducting frequent code reviews to identify and correct vulnerabilities before deployment. User access controls, including Role-Based Access Control (RBAC), limit access to data and features required for users' jobs to minimize access. Logging and audit trails provide detailed level logs and auditing of user activities to prevent unauthorized access and serve as forensic evidence in the event of security breaches. Lastly, SAP Security notes and patches to be published frequently to address vulnerabilities and staying up to date with these patches is crucial for maintaining system security.

VII. SECURITY TESTING TOOLS FOR SAP

TABLE II. COMPARISION OF SECURITY TESTING TOOLS

Tool Name	***Type of Test***	***Features***
SAP NetWeaver Security Tool	Configuration and vulnerability tests	Built-in vulnerability scanners, risk management
OWASP ZAP	Web application security testing	Open source, checks for XSS, SQL Injection
Burp Suite	Penetration testing	Automated and manual tests, vulnerability scanner
Fortify	Static and dynamic security testing	Scan source code for vulnerabilities integrates with CI/CD

A. *SAP Security Audit Log:*

The SAP Security Audit Log is an effective system that records every security-related activity within the SAP environment, including logins, role changes, and data reads. This log serves as the foundation for monitoring and auditing SAP systems. SAP administrators can analyze the Security Audit Log in real-time to detect suspicious activity, allowing them to quickly halt unauthorized access and prevent configuration changes from being made.

B. *SAP NetWeaver Security Testing Tool / Test Kit:*

SAP NetWeaver has pre-built instruments to validate the security settings of SAP systems. The vulnerability scanner and risk management features it includes are used to monitor the overall SAP security status of the environment. This tool is perfect for spotting the misconfigurations of the system, weak access controls, missing patches, old security settings, etc.

C. *Open-Source Security Testing Tools:*

1) OWASP ZAP: It’s an open-source web application security test tool that can check SAP web applications for popular vulnerabilities, like Cross-Site Scripting (XSS) and SQL injection [1].
2) Burp Suite: Another widely used open-source penetration test software, Burp Suite, is used to find gaps in SAP web apps by modeling attacks. It’s a must-have to detect any vulnerability in SAP web interfaces [2].

D. *Commercial Security Testing Tools*

1) Fortify: A powerful static and dynamic security testing tool that scans the SAP applications’ source code for flaws. It’s good for finding gaps in the ground early on [3].
2) Checkmarx: Like Fortify, Checkmarx is all about application security and can work wonders to find flaws in your SAP applications. It automatically flags coding problems like buffer overflows and wrong input checks [4].

3) SAP Solution Manager: SAP Solution Manager connects with SAP systems for security checks. It aids in monitoring the system for vulnerabilities and compliance breaches and generates automated dashboards for administrators.

E. *SAP System Integration Tools.*

1) SIEM Tools: SIEM (Security Information and Event Management) tools like Splunk can be deployed on SAP to monitor and analyze real-time logs. They are used to identify a security risk due to suspicious activity or pattern within SAP [5].
2) Continuous Scanning: Streamlined connectivity to SAP means security scans, alerts, and reports on ongoing security breaches, keeping organizations up to date with the latest risks [7].

VIII.How Security Testing In SAP Can Be Effective

For measuring the success of security testing in SAP, there are various techniques that companies can follow:

Security Testing KPIs (Key Performance Indicators) are used to evaluate the success of security testing. One key indicator is the reduction in vulnerabilities, which signifies the success of security testing when vulnerabilities decrease over time. A significant drop in vulnerabilities following patches and updates is a clear indicator of effective testing. Another important KPI is incident response time, which measures how quickly a company responds to discovered vulnerabilities or attacks. Good security testing needs to be fast and have vulnerabilities patched quickly.

Risk mitigation metrics are also crucial in measuring the effectiveness of security testing. Risk prevention is one such metric, as it helps quantify how much risk has been reduced after security testing. When gaps are patched, the risk profile of the organization improves. Compliance audits are another important metric, as they ensure adherence to industry standards (e.g., GDPR, SOX). A decrease in audit findings related to security gaps indicates the success of security testing efforts.

Reviewing user access is another critical aspect of security testing. Regular role-based access reviews make sure that only the right individuals have access to sensitive data, and proper security testing results in fewer misconfigured access permissions. Regular check-ins help keep user access up to date.

Post-testing penetration testing success rate is also an important metric. Once security testing is completed, it is also a must to do a further round of penetration testing. If fewer vulnerabilities are discovered in this second cycle of testing, it means that previous testing is good for finding and closing security gaps.

Finally, Audits and reviews of system logs can identify if prior security testing has led to less access by unauthorized users or other suspicious activity.

IX. SAP Security testing case studies

A. *Case Study 1: Securing SAP Systems in a Global Energy Provider*

*1) Background:*A multinational energy provider needed to secure its SAP ERP system after migrating to the cloud. The system manages financial, operational, and customer data globally.

2) *Testing Process*
   a) Role-Based Access Testing: During the security review, SAP role-based access was scrutinized to ensure users only had the necessary permissions. Privilege escalation was tested across key modules.
   b) Automated vulnerability scans identified outdated patches, while pen testing focused on exploiting role-based misconfigurations.
   c) Access control audits highlighted excess admin roles and inappropriate user authorizations.

3) *Results*
   a) Identified excessive admin roles and misconfigured access rights.
   b) Implemented least-privilege access and multi-factor authentication (MFA) and reduced unnecessary admin privileges.

4) *Lessons*
   a) Regular role-based access testing is crucial for mitigating internal risks.
   b) Enforce the principle of least privilege to minimize the impact of insider threats [7].

B. *Case Study 2: Securing SAP in a Utility Company's Operational Network*

1) *Background:* A utility company integrated SAP with IoT for grid management and needed to secure sensitive data and operational systems.

2) *Testing Process*
   a) Role-Based Access Testing: SAP user roles in grid management were thoroughly tested. Overly broad roles were found that allowed unauthorized access to critical systems.
   b) Penetration testing and vulnerability assessments uncovered weak authentication and communication between SAP and IoT systems.
   c) Risk assessments prioritized securing SAP IS-U (Utilities Industry Solution) data and configuring proper role access.

3) *Results*
   a) Found misconfigured roles leading to access issues in SAP PM (Plant Maintenance) and asset management.
   b) Applied stricter role-based access controls (RBAC) and enhanced authentication mechanisms.

4) *Lessons*
   a) Tighten role-based access to ensure users only have access to relevant data.
   b) Secure IoT integration with proper role segregation to prevent unauthorized data manipulation.

C. *Case Study 3: Protecting SAP Data in a Renewable Energy Firm*

1) *Background:* A renewable energy firm deployed SAP S/4HANA for financial, asset, and resource management across global operations and required robust data security.

2) *Testing Process*
   a) Role-Based Testing: The SAP role configuration was assessed, especially for access to sensitive financial and environmental data. Testing revealed overly permissive roles, especially in SAP Fiori and SuccessFactors.
   b) Automated scans and penetration tests confirmed the need for tighter role segregation and audit trails for access to sensitive modules.

3) *Results*
   a) Overprivileged roles in financial systems and weak session management were corrected.

b) Applied granular role-based permissions, enhanced encryption, and strengthened audit trails for user actions.

4) *Lessons*

a) Role-based testing is essential to prevent unauthorized access to critical data.

b) Regular audits of roles and user privileges are necessary to align with security standards.

X. Conclusion

Role-based access testing proved vital in identifying and mitigating security risks within SAP systems in the energy and utilities industry. By focusing on the principle of least privilege, organizations can ensure sensitive data and systems remain protected from both external and internal threats. Regular role audits, along with granular permission controls and strong authentication mechanisms, are key to securing critical business operations and maintaining compliance.

As cyber threats continue to evolve, regular security testing is essential to ensure that SAP systems remain secure over time. SAP data security for the future will also be driven by the increasing use of AI to detect threats, the adoption of zero-trust security concepts, and increased collaboration with cloud environments [8]. To stay ahead, businesses should take a proactive approach to security testing, implement best practices, and utilize advanced tools to protect their SAP environments.

XI. References

[1] OWASP, "OWASP Benchmark," *OWASP Foundation*, 2019. [Online]. Available: https://owasp.org/www-project-benchmark/.

[2] Burp Suite, "Burp Suite: Web Application Security Testing," PortSwigger, 2003[Online]. Available: https://portswigger.net/burp

[3] Micro Focus, "Fortify on Demand: Dynamic Application Security Testing," *Micro Focus*, 2023[Online]. Available: https://www.microfocus.com/media/brochure/fortify_on_demand_dynamic_application_security_testing_brochure.pdf

[4] B. Brucker, "SecEntis: Static Analysis in the Context of Security Testing," *Brucker.ch*, 2016. [Online]. Available: https://brucker.ch/bibliography/download/2016/talk-brucker-secentis-static-analsyis-2016.pdf

[5] Splunk Inc., "Splunk SIEM Solutions for SAP Security," Splunk, 2024. [Online]. Available: https://www.splunk.com/en_us/solutions/security-for-sap.html

[6] J. Hirao, *SAP Security Configuration and Deployment: The IT Administrator's Guide to Best Practices*, 1st ed. New York, NY, USA: McGraw-Hill, 2019.

[7] Tolk, A. and Bell, J., "Adopting Zero Trust Security for SAP Systems," International Journal of Cybersecurity and SAP Systems, vol. 5, no. 4, pp. 72-80, 2024.

[8] Richardson, A. and Mukherjee, S., "Case Studies in SAP Security Testing: Role-Based Access Control and Vulnerability Scanning," Journal of Enterprise Security, vol. 18, no. 2, pp. 45-59, 2023.