onapsis
Securing Business Essentials

# SAP BACKDOORS

## *A ghost at the heart of your business*

**by Mariano Nuñez Di Croce**
***mnunez@onapsis.com***

**Abstract**

Backdoors have existed since the origins of information systems. However, there is still scarce public information on how this threat affects the SAP business platform.

Due to the nature of the information managed through these systems and the business processes implemented in them, the existence of a backdoor in an Organization's SAP platform can have devastating impacts.

This publication analyses some of the different techniques that can be used by malicious parties to install a backdoor in an SAP system, abusing of specific functionalities and bypassing security protections. More importantly, after reviewing the existing threats, different mitigation and protection measures are presented so that SAP customers can increase the security level of their critical business platforms.

Finally, this publication also introduces a new free security tool – Onapsis SAP Security Snapshot – which will help companies world-wide to detect unauthorized modifications of critical system components.

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

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

# TABLE OF CONTENTS

**Note:**

The latest version of this white-paper can be found at http://www.onapsis.com/resources/get.php?resid=ssid03.

# 1. INTRODUCTION

The term *backdoor* is commonly referred to special methods that are implemented in an information system after an unauthorized compromise, with the purpose of securing future access to the system while attempting to remain undetected.

The security industry has been researching on *backdoors* for several decades, working on different attack vectors and in the defense measures that can be applied in order to prevent systems from being *backdoored*.

However, the Onapsis Research Labs has detected that there is scarce public information on how these threats affect the SAP business platform, and this provides a false sense of security for *many Organizations which are not aware of, and therefore not managing, this kind of risk*.

Having an unauthorized party installed a *high-privileged backdoor* in an SAP system, he may be able to (silently) manipulate all the information and business processes implemented in the Organization's SAP platform, such as procurement-to-pay, order-to-cash, stocking and shipping, financial planning, human resources management, etc. Furthermore, if the attacker is skilled and no proper countermeasures were implemented, the incident *would be only detected several months after the initial compromise*.

Under this circumstances, *the corporative world is running behind malicious individuals/organizations*, due to the lack of precise information on how to assess their system for the existence of backdoors and how to implement proactive measures to prevent them. **The objective and responsible analysis of this problematic is the only sustainable choice the industry faces**.

In this publication, some of the different techniques that can be used by malicious parties to install backdoors in SAP systems are analyzed, as well as the possible countermeasures that exist in order to prevent and detect them.

Onapsis expects this information will help SAP customers manage risk in a holistic way, allowing them to **decrease fraud risks** and **enforce compliance requirements**.

# 2. BASIC CONCEPTS

In order to help the general community to understand the present publication, it is highly important to familiarize with some basic concepts about the SAP platform. These concepts are presented below:

## The SAP Change and Transport System

The Change and Transport System (CTS) is a tool that allows SAP customers to organize and monitor changes in their SAP platform.

Under this architecture, changes only take place in one system: the Development (DEV) system. After changes or new developments are ready, they are *transported* into the Quality Assurance (QAS) system. In this system, the QA team test for the fulfilment of the business requirements and their quality. Finally, after the changes/add-ons have been approved, they are *transported* into de Production (PRD) system, where end-users use them to perform different business processes.

It is expected that, following this procedure, the q*uality and availability of the SAP production systems is maximized.*

The Production system (the one where the *real* business information is managed) is secured by configuring it to be "*Not modifiable*" through transaction *SE03*.

This way, the system enforces that changes can *only* be introduced through the Change and Transport System and no modifications can be done directly in the production system.

## Custom Developments

Organizations running SAP business solutions many times face the need to implement special customizations to their platform, in order to adapt the system to their specific business requirements.

These adaptations can be done through the configuration of special parameters of the system and/or developing customized ABAP programs. It is possible to identify custom developments as they belong to specific namespaces, which start with the letters "Y" or "Z".

As it has been described in the previous section, all the changes and new developments have to go through the SAP Change and Management System. In this process, all modifications are registered and logged by the system for monitoring and auditing purposes.

## Modification of the SAP Standard (BC)

Finally, there is another possibility to adapt the SAP system to the customer's needs: the modification of the *SAP Standard*. The Standard is the set of ABAP programs and Dictionary Objects that is shipped in the base installation of the SAP system.

The modification of the Standard is only generally used as the last resort, as SAP does not guarantee that the modified programs will not be overwritten in future upgrades.

In order to modify a standard program, a special procedure called SSCR (SAP Software Change Registration) must be followed. This procedure basically consists in requesting a special *Access Key* from SAP, that must be entered in the customer SAP system before being able to modify the target program.

All the modifications to the standard are logged in the system, so that SAP can count on this information when support is requested.

# 3. BACKDOORS IN THE USER MASTER

The User Master contains all the user-related information in the SAP system, such as personal details, password, roles, authorizations, etc.

From an attacker's point of view, the most traditional (and simplest) way to install a backdoor in a system is to create a new user with high privileges. In the SAP world, this translates to the creation of a new user, with the *SAP_ALL* profile.

In the future, the attacker will access the system using this new user account and perform unauthorized activities with high privileges.

However, as the installation process is simple, so is the detection. Regular system audits or continuous control monitoring solutions will easily spot the existence of a new user with high privileges, which will be then removed from the system.

In the SAP world the situation can get more complex, though.

User passwords are stored in the *USR02* database table. This table contains different versions of the password hash, in order to provide compatibility with older systems where newer hashing schemes are not supported. Before continuing reading, it is strongly recommended to review the Onapsis “SAP Security In-Depth Vol. 1” [1] edition, where this subject is analyzed in detail.

The generation of the different hashes, and the selec*tion of the hash to compare against in the logon procedure,* is controlled by the value of the *login/password_downwards_compatibility* profile parameter. This parameter can be configured to a numeric value between 0 and 5, being 1 the default value in modern SAP releases.

Besides the risks described in [1], there are other values of this parameter that imply high risks for the security of the SAP platform:

| Parameter Value | Effect |
|---|---|
| 2 | If the logon attempt using the backwards-incompatible procedure fails, the downwards-compatible hashes are also evaluated to check whether they would have been accepted. This is registered in the system log and the logon fails. This is used to detect backward incompatibility problems. |
| 3 | The same as with 2, but the logon is considered as successful. This is registered in the system log. |

| | |
|---|---|
| 4 | The same as with 3, but no system log entry is written. |

By looking at the previous table, the risk become pretty clear: *If this parameter is set to 3 or 4, the SAP system also checks the received password against the downwards-compatible hash, if the check for the "strong" hash fails.*

This provides room for an attacker to install a backdoor by **modifying only the downwards-incompatible hash of a privileged user account.**

As the legitimate user will always succeed in authenticating with his "strong-hashed" password, he will never notice any problem and *the backdoor will remain undetected*. The attacker will specify his password, which will fail the first check but pass when checked against the "weak" hash value.

### Protection / Countermeasures

The most effective way to protect against this threat is to monitor the value of the *login/password_downwards_compatibility* profile parameter.

The detection of non-secure values, such as the described in this section, may indicate that some unauthorized activity was carried over the system and should be investigated accordingly.

It is also strongly recommended to implement a dedicated authorization group for the U* tables, such as USR02, in order to prevent access and modification of the password hashes from the SAP system.

Other complementary protection measures can include the regular cracking of user password hashes to detect inconsistencies and the configuration of auditing settings for critical tables such as *USR02* at the database layer.

# 4. HARDCODED USERNAMES IN STANDARD CODE

It has been detected that some of the Standard ABAP code is shipped with hard-coded checks for specific *usernames.* These checks were probably used in the development and testing phases of the product, and were inadvertently left there.

Following, some examples of this issue are provided:

Report **/SAPAPO/LCDPS_G1CFZM** [2]

```
...
IF sy-uname = 'C5007879'.
   nowd = sy-datum.
   nowt = sy-uzeit.
   g_time_zone = sy-zonlo.
...
```

Report **/SAPAPO/ADVC_MB_PERMISSION_CHK** [3]

```
...
e_flg_no_permission = true.

* #TR# (2) Perform permission check
* #TR# (2a) The user SAP_MACROS is allowed to do everything

  if sy-uname = gc_sap_macros. e_flg_no_permission = false. exit.
endif.
...
```

As it can be noticed in the last example, some of these hard-coded checks can be abused to bypass security controls. In this case, if an attacker creates a user account "SAP_MACROS" and executes this report, he would bypass a security validation in the logic of the affected program.

**Protection / Countermeasures**

SAP regularly checks its codebase to detect this kind of errors and produces the respective Security Notes in order to provide a solid protection for its customers.

In order to be protected against these vulnerabilities, it is strongly recommended to stay updated with these Security Notes.

Furthermore, it is possible to use transaction *SCI* or report *RS_ABAP_SOURCE_SCAN* to detect this kind of dangerous ABAP code snippets in your system.

## 5. BACKDOORS IN ABAP BUSINESS PROGRAMS

As it has been described in the Introduction section, modifications to ABAP programs is tightly controlled through the SAP Change and Transport System.

Analyzing this process from a lower-level perspective, it is possible to observe that all the ABAP programs installed in the SAP system are *stored in the database* table *REPOSRC*.

This table contains information regarding each ABAP program in the system, as well as its (compressed) source-code in the *DATA* field. There is no integrity checking on this information at the SAP layer.

As the Onapsis Research Labs has presented and discussed before [4], it is highly common to discover that the **access to the SAP Database is not properly secured** in many Organizations.

Therefore, it is possible for an attacker to create an ABAP backdoor outside the target SAP system and inject it by modifying the appropriate fields of the *REPOSRC* table in the target SAP system's database using simple SQL queries, like the following:

```
UPDATE SAPSR3.REPOSRC
        SET DATA = <BACKDOORED_CODE>
        WHERE PROGNAME = <PROGRAM_NAME>
```

After the modification of the compressed source code of the target ABAP program at the database level, he would need to invalidate the ABAP load of that program, so that the program is re-generated by the system the next time it is executed. This can be done by removing the appropriate record from the *REPOLOAD* database table:

```
DELETE FROM SAPSR3.REPOLOAD
        WHERE PROGNAME = <PROGRAM_NAME>
```

**It is important to notice that this attack not only allows the modification of custom (Z*) programs, but also the modification of SAP Standard code.**

An attacker **would be able to modify any business-related ABAP program,** such as the ones involved in critical transactions like FK01 (Create Vendor), ME21 (Create Purchase Order), PA30 (Maintain HR Master Data), FI12 (Change House Banks/Bank Accounts), etc. which could result in severe business frauds to the Organization.

Through this type of attack, the malicious party is able to **bypass the change controls systems** implemented in the SAP platforms, and **remotely modify programs directly in the Production system**.

## Protection / Countermeasures ⛔

Once the attacker has compromised the security of the system with maximum privileges, either through the application or database layer, preventing him from installing a backdoor is a difficult task.

Therefore, efforts should be concentrated in avoiding the initial compromise. This can be enforced through regular specialized Vulnerability Assessments, Penetration Tests and/or Security Audits of the entire SAP platform.

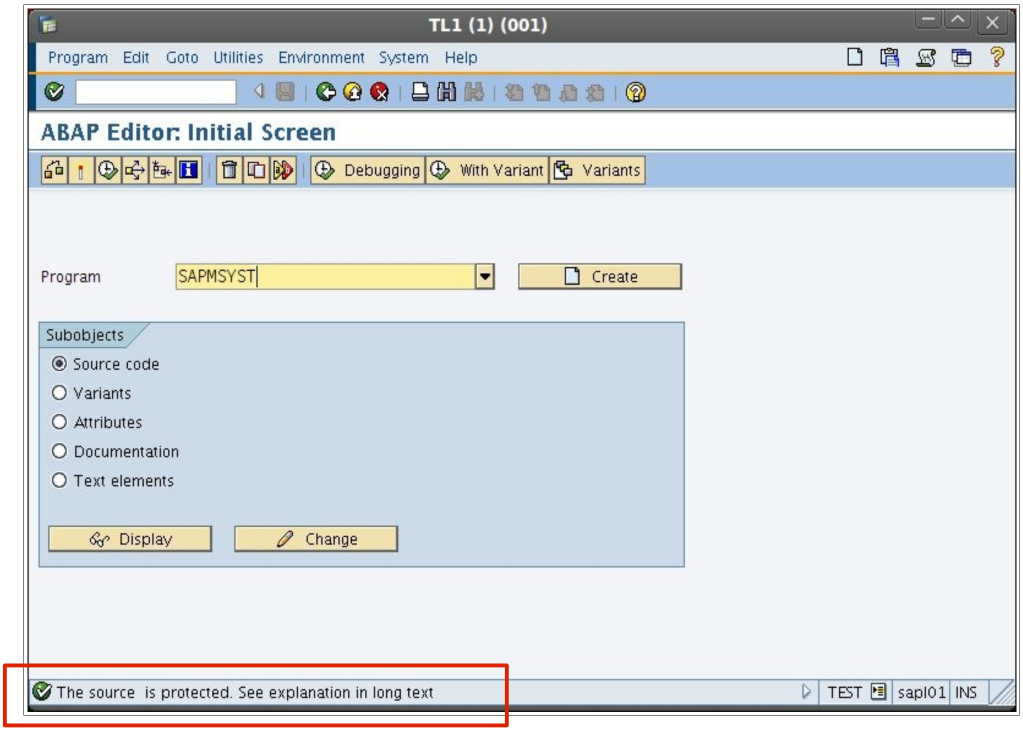
However, in order to provide a comprehensive solution, the Onapsis Research Labs has developed the *Onapsis Integrity Analyzer for SAP.* Please check section 7 for more information.

# 6. BACKDOORS IN THE AUTHENTICATION MODULE

Among the thousands of SAP Standard ABAP modules, there are some of them which involve a high level of criticality. One of them is the *SAPMSYST* program.

The *SAPMSYST* program is the ABAP code responsible of receiving the authentication information from users that want to log into the SAP system using SAPGUI, among other methods. **This procedure is absolutely critical, as any unauthorized modification/vulnerability in this code will introduce a severe threat to the overall security of the SAP system.**

Taking this into account, SAP has implemented several controls in order to prevent access to the source code of this program. Then, if a user tries to display the source code using, for example, transaction *SE38*, the following message is presented:


TL1 (1) (001)
Program Edit Goto Utilities Environment System Help
ABAP Editor: Initial Screen
Debugging
With Variant
Variants
Create
Display
Change
TEST
sapl01
INS

Program: SAPMSYST

Subobjects
● Source code
○ Variants
○ Attributes
○ Documentation
○ Text elements

The source is protected. See explanation in long text

Therefore, in order to assess the security provided by this mechanism, a research on how this protection is implemented was started:

Looking at the *REPOSRC* table, it is possible to observe a field named *SQLX*, which description is “Source code protection”. In the table record for *SAPMSYST*, this field is set to “X”.

While it seemed that this field could be responsible for restricting access to the protected source code, its modification leads to no noticeable results.

Our research indicates that the protection of the source code *is done in a different way*: If the ABAP source code contains the special string "*@#@@[SAP]", when the kernel processes the *READ REPORT* directive , it returns a special code (*sy-subrc=8)*. Then, display access to the source is *forbidden* by the system.

However, this protection is used for some important programs, but not for *SAPMSYST*. Analyzing the SAP kernel, it was possible to discover that **there is *a specific check at the kernel level, in order to restrict access to the SAPMSYST* program**. This is done by comparing the requested ABAP program's name to the "SAPMSYST" string.

This protection is as far as the SAP kernel can go, and it is not considered trivial to bypass from within the SAP system itself.

However, the platform is again susceptible of attacks at a lower level: the database. An attacker can perform a direct SQL query, with which he will copy the source code of the SAPMSYST program to a program with a different name *(eg. ZTEST)*, therefore bypassing the kernel protection.

Having done this, the attacker is able to edit the *ZTEST* program just as any other program, and inject malicious code in it. Then, a new SQL query will overwrite the original *SAPMSYST* source code, **effectively installing a backdoor in the logon procedure**.

The backdoor payload chosen by the attacker is only restricted by *his imagination*. **For example, a malicious party can install a backdoor that sends him an HTTP or email message containing the authentication information (client, user and password) of every user that logs into the SAP system.**

## Protection / Countermeasures ⛔

Once the attacker has compromised the security of the system with maximum privileges, either through the application or database layer, preventing him from installing a backdoor is a difficult task.

Therefore, efforts should be concentrated in avoiding the initial compromise. This can be enforced through regular specialized Vulnerability Assessments, Penetration Tests and/or Security Audits of the entire SAP platform.

However, in order to provide a comprehensive solution, the Onapsis Research Labs has developed the *Onapsis Integrity Analyzer for SAP.* Please check section 7 for more information

# 7. ONAPSIS INTEGRITY ANALYZER FOR SAP

The detection of unauthorized modifications in an SAP system is a difficult mission. To begin with, it is highly important to emphasize that **this task can not be performed from *within the SAP system itself***, because of the following reasons:

1) The operations performed by the attacker *may have manipulated the information available in control fields*, such as the "Last modified time" property of an SAP program. This turns this information into completely untrusted.

2) It is not possible to ensure that the programs used to analyze ABAP code or Audit the system from the SAP system itself have not been modified to hide the backdoor's presence.

Therefore, the only feasible possibility is to analyze the ABAP code through an external procedure, and from a different layer. In this aspect, the analysis at the database level is considered the best option. While the database kernel may have also been compromised to hide the attacker's activities, this type of modification requires a much higher level of skills.

In order to assist Organizations in this process, Onapsis has developed the Onapsis Integrity Analyzer for SAP, which can be downloaded free-of-charge from http://www.onapsis.com/ianalyzer.

Through this tool, the auditor connects remotely to the SAP database and automatically performs a security snapshot of the current state of the ABAP programs available in the SAP system, which is saved in his local computer.

Periodically, the auditor will perform new security snapshots of the target SAP system, and, through the comparison with older samples, the tool will notify if any of the programs was modified.

This tool was presented at the Black Hat Europe 2010 Briefings, and was designed as a *proof-of-concept* to provide basic protection capabilities to SAP customers. The Onapsis Research Labs Team is working in an advanced version of the solution, which will provide a higher level of defense against these threats.

# 8. CONCLUSIONS

Backdoors represent a high risk threat to any information system. However, in the case of SAP business solutions, the risk is even more critical. **Having a malicious party backdoored an SAP system, he may be able to perform sensitive business processes or obtain/modify confidential information, which can result in severe economic losses for your Organization.**

While there is scarce public information about the techniques used to install them, there is even less information on how to protect the system against this threat. Under these circumstances, it may be possible for underground and malicious organizations to exploit this security knowledge gap and perform unauthorized activities over a company's business information.

This publication has highlighted only a few possibilities in order to illustrate the problem and allow the general SAP customer community to understand the involved risks and available protections.

**Protecting against backdoors is difficult, but possible.** To begin with, if an attacker can not accomplish the initial compromise, he would never be able to install a backdoor. Therefore, the biggest effort should be applied in defense measures to protect the system against known vulnerabilities and weaknesses. **This can be enforced by performing *regular specialized Vulnerability Assessments, Penetration Tests* and Security Audits over the entire SAP platform.**

In this direction, **Onapsis' Bizploit** [5], the open-source ERP Penetration Testing Framework, can be useful to perform a basic security assessment of the Organization's SAP systems.

Adding an extra layer of security, the ***Onapsis Integrity Analyzer for SAP*** [6] allows you to keep control of the modifications to your ABAP programs and detect unauthorized or suspicious activity.

By following the recommendations presented in this publication it is possible to decrease the probability of attacks in this aspect, raising the overall security level of the platforms and reducing business fraud risks.

**For further information into this subject or to request specialized assistance, feel free to contact Onapsis at info@onapsis.com**

# 9. REFERENCES

[1] Onapsis SAP Security In-Depth Publication Vol. 1 – http://www.onapsis.com/resources/get.php?resid=ssid01

[2] SAP Note 1334396 - https://service.sap.com/sap/support/notes/1334396

[3] SAP Note 1335103 - https://service.sap.com/sap/support/notes/1335103

[4] Onapsis "Penetration Testing SAP Systems" Presentation- http://www.onapsis.com/research/slides/ONAPSIS-Penetration_Testing_SAP.pdf

[5] Onapsis Bizploit – http://www.onapsis.com/bizploit

[6] Onapsis Integrity Analyzer for SAP – http://www.onapsis.com/ianalyzer

# 10. ABOUT ONAPSIS

Onapsis is the leading provider of solutions for the security of business-critical systems and applications.

Through different innovative products and services, Onapsis helps its global customers to effectively increase the security level of their core business platforms, protecting their information and decreasing financial fraud risks.

Onapsis is built upon a team of world-renowned experts in the SAP security field, with several years of experience in the assessment and protection of critical platforms in world-wide customers, such as Fortune-500 companies and governmental entities.

Some of our featured services include SAP Penetration Testing, SAP Gateway & RFC security, SAP Enterprise Portal Security, Security Support for SAP Implementations and Upgrades, SAP System Hardening and SAP Technical Security Audits.

For further information about our solutions, please contact us at **info@onapsis.com.**