



Andreas Wiegenstein

- Founder of Virtual Forge (Heidelberg), responsible for Research & Development
- SAP Security Researcher, active since 2003
 - Received Credits from SAP for more than 20 reported 0-day Vulnerabilities
- Frequent Speaker at international Conferences
 - **SAP TechEd** 2004 (USA & Europa) / 2005 (USA) / 2006 (USA), DSAG 2009
 - **BlackHat** 2011 (Europe), **Hack in the Box** 2011 (Europe)
 - **Troopers** 2011, **RSA** 2012 (USA)
- Co-Author of „Sichere ABAP Programmierung" (SAP Press)
- Training Class WDESA3 @ SAP University



MEMBER LOGIN

Log in

[Forgot your password?](#)

[Not a member?](#)

- Getting Started
- Application Lifecycle Management
- Business Intelligence
- Data Warehousing
- Enterprise Information Management
- Business Process Management and Composition
- Service-Oriented Architecture
- SOA Middleware
- User Productivity
- Custom Development
- Security and Identity Management
- Technology Innovation
- In-Memory / SAP HANA
- Mobile

ACKNOWLEDGMENTS TO SECURITY RESEARCHERS

The SAP Product Security Response Team thanks all researchers and security IT professionals that helped with discovering and solving security vulnerabilities. Their findings have helped SAP to maintain the security and safety of its customers' and partners' SAP systems.

Our acknowledgements page lists those professionals we have worked with successfully in the past. The acknowledgements are published on a monthly basis and mention all security researchers who helped to improve the security and integrity of our customers' IT systems by respecting our disclosure guidelines. We thank all security researchers for their excellent work and hope to continue the fruitful relationship between security professionals and SAP.

ARCHIVE

[Here](#) you can find elder entries.

FEBRUARY 2012

- [Virtual Forge](#), Sebastian Schinzel & Frederik Weidemann, SAP Security Note [1586410](#)
- [Virtual Forge](#), Andreas Wiegenstein & Frederik Weidemann, SAP Security Note [1584930](#)
- [Virtual Forge](#), Erich Prosche & Sandra Möckel, SAP Security Note [1607529](#)
- [Virtual Forge](#), Andreas Wiegenstein & Sven Neuz, SAP Security Note [1597597](#)
- [Virtual Forge](#), Andreas Wiegenstein, SAP Security Note [1661349](#)
- [ERPSecurity](#), Joris van de Vis, SAP Security Note [1641329](#)
- [ERPSecurity](#), Joris van de Vis, SAP Security Note [1644746](#)
- [Zero Day Initiative](#), SAP Security Note [1649838](#)
- [Zero Day Initiative](#), SAP Security Note [1649840](#)
- [ESNC](#), Ertunga Aرسال, SAP Security Note [1667805](#)
- [akquinet AG](#), Ralf Kempf, SAP Security Note [1644043](#)





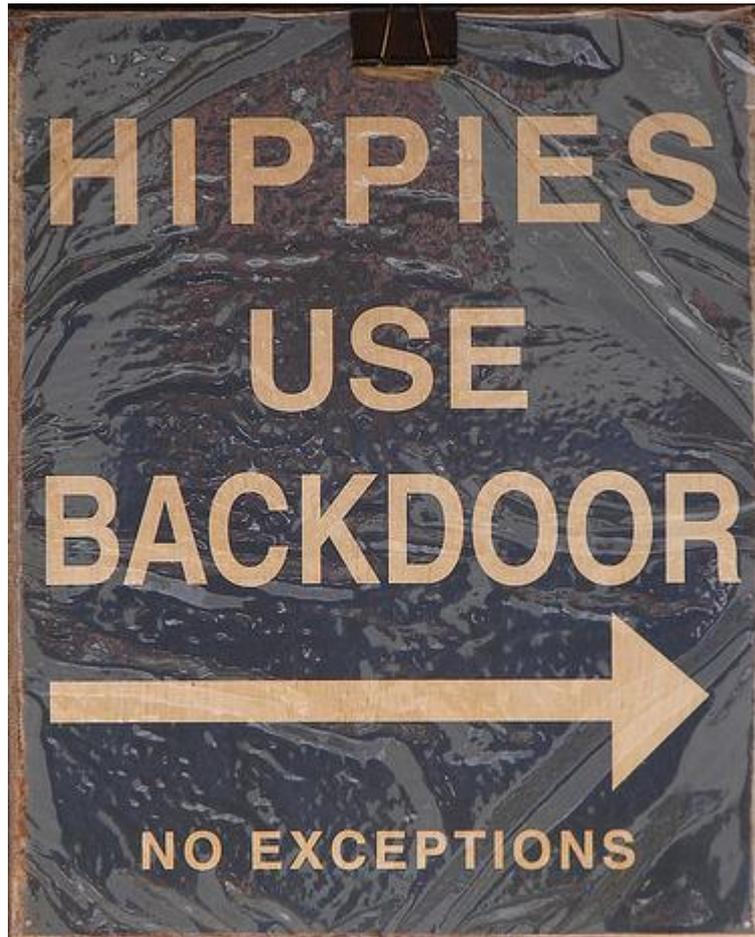
- 1. What is a Backdoor?**
- 2. SAP Technology / Security Basics**
- 3. SAP Backdoors**
- 4. How do you prevent Backdoors?**
- 5. Summary**

What is a backdoor?



**THIS IS
NOT
THE TOPIC
OF THIS TALK.**

NO WAY.



ALSO
NOT
THE TOPIC
OF THIS TALK.





“A backdoor in a computer system [...] is a method of bypassing normal authentication, securing remote access to a computer, obtaining access to plaintext, and so on, while attempting to remain undetected.”

(March 2012)



“A backdoor in software is a hidden feature that was designed to bypass a security mechanism.”

(Troopers, March 2012)

Characteristics:

1. Covertness
2. Bypass
3. Intent

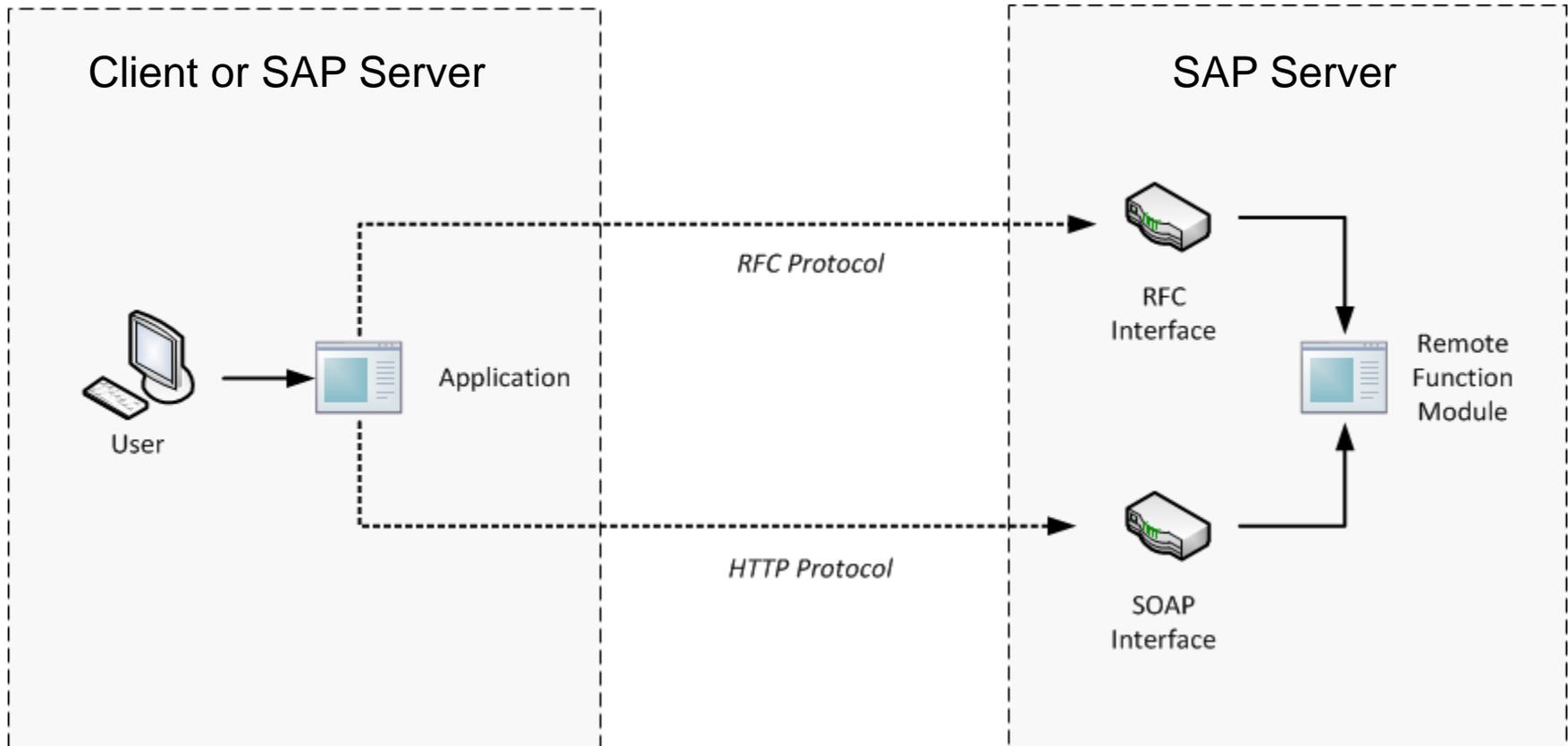
SAP Technology / Security Basics



Advanced Business Application Programming

- Proprietary language, exact specification not (freely) available
- Platform-independent code
- Built-in transport system and version control
- Various programming paradigms:
 - Programs & Forms, Reports, Function Modules, Dynpros
 - Classes & Methods, Business Server Pages, Web Dynpro ABAP
- Integrated platform-independent SQL Standard: Open SQL
- Built-in authentication, roles and authorization model
- **ABAP runs with very high Privileges**
- **ABAP uses an *explicit* Authorization Model**

Remote Function Call (RFC)



- S_RFC authorization required to call Function Modules remotely
- > 33.000 RFC-enabled Function Modules on ECC 6.0

RFC authorizations are complex to maintain



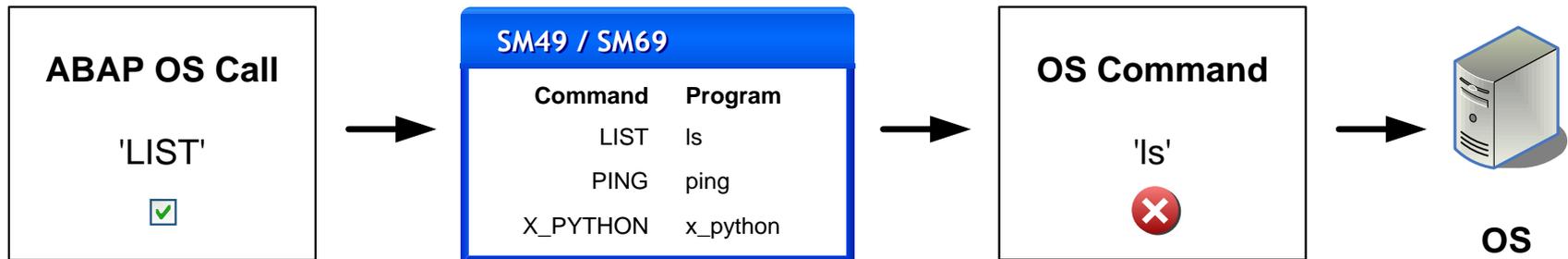
The screenshot shows the 'ABAP: Program Attributes ZFT Display' window. The title is 'We are Troopers'. The original language is 'EN' (English). It was created and last changed on '09.03.2012' by 'WIEGENSTEINA'. The status is 'Active'. Under the 'Attributes' section, the 'Type' is 'Executable program', 'Status' is empty, 'Application' is 'Cross-Application', and 'Authorization Group' is 'SES', which is circled in red. The 'Package' is 'SUP_RESEARCH' with a description 'Contains research material and 0-days'. Other attributes include 'Logical database', 'Selection screen', and several checkboxes: 'Editor lock' (unchecked), 'Unicode checks active' (checked), 'Fixed point arithmetic' (checked), and 'Start using variant' (unchecked).

- Reports can only be executed locally via restricted transactions
- ~ 220.000 ABAP reports on ECC 6.0 in the SAP standard
- ABAP command `SUBMIT` executes reports and checks authorizations
 - Authorization is checked only if Authorization Group is maintained

SAP Backdoors



Controlled Operating System (OS) Command Execution



© 2010 Virtual Forge GmbH. All rights reserved.

- OS Commands must be pre-defined by Admin (white list)
- OS Commands must be executed through special API (`SXPG_CALL_SYSTEM` / `SXPG_COMMAND_EXECUTE`)
- Execution requires special authorization (`S_LOG_COM`)

Case #1 – Code (1)



VIRTUALFORGE
we harden your software

```
FUNCTION oiuh_submit_unix_call.  
* "-----  
* "*"Local interface (simplified excerpt):  
* " IMPORTING  
* "     VALUE(SCRIPT_NAME) LIKE    RLGRAP-FILENAME  
* "     VALUE(LOGICAL_PATH) LIKE   FILENAME-FILEINTERN  
* " TABLES  
* "     RESULTS STRUCTURE  OIUH_SYS_CONSOLE  
* "     SCRIPT_DATA STRUCTURE  OIUH_SYS_CONSOLE  
* " EXCEPTIONS  
* "     CALL_FAILURE  
* "-----
```

Case #1 – Code (2)



VIRTUALFORGE
we harden your software

```
DELETE DATASET script_name.  
OPEN DATASET script_name FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.  
LOOP AT script_data.  
    TRANSFER script_data TO script_name.  
ENDLOOP.  
CLOSE DATASET script_name.  
  
* CHANGE THE FILE MODE TO EXECUTE.  
CONCATENATE 'chmod 777' script_name INTO unix_command SEPARATED BY space.  
...  
CALL 'SYSTEM' ID 'COMMAND' FIELD unix_command  
           ID 'TAB'      FIELD results-*sys*.  
...  
* Execute the actual command  
CALL 'SYSTEM' ID 'COMMAND' FIELD script_name  
           ID 'TAB'      FIELD results-*sys*.
```



Function Module `oiuh_submit_unix_call` is *designed to execute arbitrary OS commands, bypassing* the white list defined in SM49/69.

Characteristics:

1. Covertness
2. **Bypass**
3. **Intent**





Function Module `oiuh_submit_unix_call2` is an exact copy of `oiuh_submit_unix_call`.

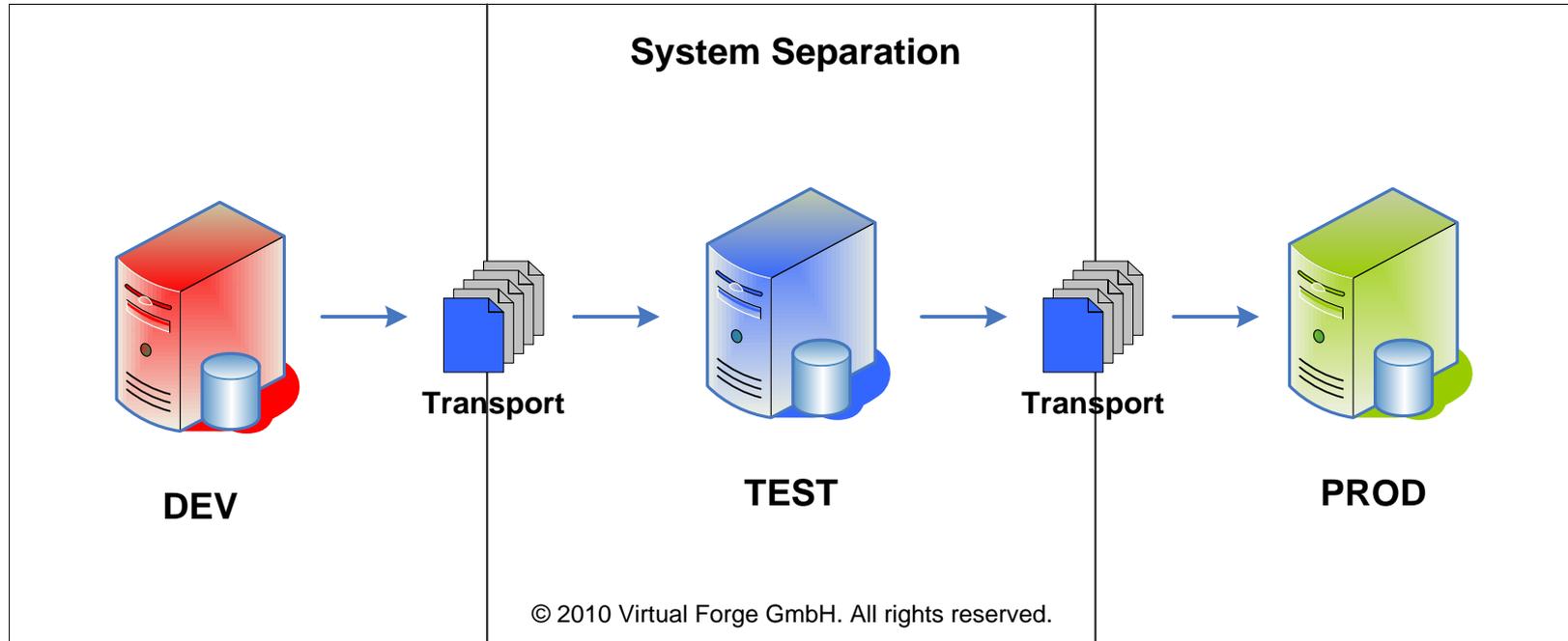
Both Function Modules also contain a Directory Traversal vulnerability.

VF Advisories: **SAP-BACK-01** and **SAP-BACK-02**

SAP Notes: 1560360 and 1558010

SAP CVSS Base Score: 6.0

SAP CVSS Base Vector: AV:N/AC:M/AU:S/C:P/I:P/A:P



Development process is well defined : DEV, TEST, PROD

- All ABAP code is tested before productive use
- No development possible on productive system

Case #2 – Code (1)



VIRTUALFORGE
we harden your software

```
FUNCTION rs_functionmodule_insert.  
* "-----  
* " * "Local Interface (simplified excerpt):  
* "   IMPORTING  
* "     VALUE(FUNCNAME) LIKE   RS38L-NAME  
* "     VALUE(FUNCTION_POOL) LIKE  RS38L-AREA  
* "     VALUE(REMOTE_CALL) LIKE   RS38L-REMOTE DEFAULT SPACE  
* "     VALUE(SHORT_TEXT) LIKE   TFTIT-STEXT  
* "     VALUE(SUPPRESS_CORR_CHECK) LIKE  RS38L-EXTERN DEFAULT 'X'  
* "     VALUE(SUPPRESS_LANGUAGE_CHECK) LIKE  RS38L-HEAD DEFAULT 'X'  
* "     VALUE(AUTHORITY_CHECK) LIKE  RS38L-HEAD DEFAULT 'X'  
* "     VALUE(SAVE_ACTIVE) LIKE   RS38L-HEAD DEFAULT 'X'  
* "   TABLES  
* "     SOURCE STRUCTURE  RSSOURCE OPTIONAL
```

Case #2 – Code (2)



VIRTUALFORGE
we harden your software

```
...  
CALL FUNCTION 'RS_ACCESS_PERMISSION'  
  EXPORTING  
    authority_check = authority_check  
...  
IF sy-subrc = 0.  
...  
  l_source = source[].  
  LOOP AT l_source INTO l_line.  
    INSERT l_line INTO code INDEX tabix.  
    tabix = tabix + 1.  
  ENDLOOP.  
  
  INSERT REPORT rs381-include FROM code.  
ENDIF.
```



```
FUNCTION rs_access_permission.
```

```
*"-----
```

```
*"*"Lokale Schnittstelle (simplified excerpt):
```

```
*"  IMPORTING
```

```
*"    VALUE(AUTHORITY_CHECK) DEFAULT 'X'
```

```
...
```

```
l_authority_check = authority_check.
```

Case #2 – Code (4)



VIRTUALFORGE
we harden your software

```
...  
CASE mode.  
  WHEN 'MODIFY'.  
    IF l_authority_check NE ' '.  
      PERFORM accp_authority  
          USING      modus  
                   object  
                   object_class  
                   auth_object  
                   s_develop  
          CHANGING  trdir_inf.  
    ENDIF.
```



Function Module `rs_functionmodule_insert` is *designed to* create arbitrary remote-executable ABAP Code, *bypassing* the TEST System.

Characteristics:

1. Covertness
2. Bypass
3. Intent



VF Advisory: **SAP-BACK-03**

SAP Note: 1589919

CVSS Base Score: 3.5

CVSS Base Vector: AV:N/AC:M/AU:S/C:N/I:P/A:N

SAP: 'BACKDOOR'



```
FUNCTION RKC_FUNCTION_INTERFACE_GEN.  
* "-----  
* "Lokale Schnittstelle (simplified excerpt):  
* "      EXPORTING  
* "          REPID LIKE SY-REPID  
* "      TABLES  
* "          REP_TAB STRUCTURE RFCLINE  
* "      EXCEPTIONS  
* "          NOT_INSERTED  
* "-----
```

Case #3 – Code (2)



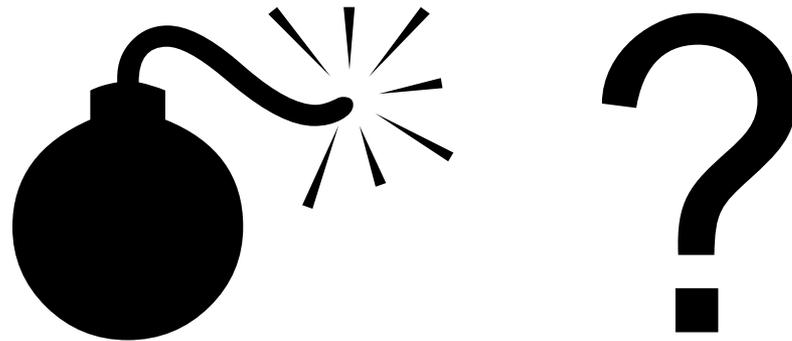
VIRTUALFORGE
we harden your software

```
DATA: BEGIN OF REP OCCURS 20.  
    INCLUDE STRUCTURE ABAPTEXT.  
DATA: END OF REP.  
  
REFRESH REP.  
LOOP AT REP_TAB.  
    REP = REP_TAB.  
    APPEND REP.  
ENDLOOP.  
REPID = 'RKCINTER'.  
INSERT REPORT REPID FROM REP.  
IF SY-SUBRC <> 0.  
    RAISE NOT_INSERTED.  
ENDIF.  
ENDFUNCTION.
```



Now we can create a report with arbitrary content.

But (how) can we execute it (remotely) ?



Case #3 – Code (3)



VIRTUALFORGE
we harden your software

```
FUNCTION HR99B_PARALLEL_REPORT_RUN.  
* "-----  
* "*"Local Interface (simplified excerpt):  
* "  IMPORTING  
* "    VALUE (REPID) TYPE  TRDIR-NAME  
* "  TABLES  
* "    VALUTAB STRUCTURE  RSPARAMS  
* "  CHANGING  
* "    VALUE (CV_TASK_NAME) TYPE  HR99B_TASK_NAME OPTIONAL  
* "-----  
  
SUBMIT (REPID) WITH SELECTION-TABLE VALUTAB AND RETURN. "#EC CI_SUBMIT  
  
ENDFUNCTION.
```



Function Module `RKC_FUNCTION_INTERFACE_GEN` is *designed to* create a Report that contains arbitrary ABAP Code, *bypassing* the TEST System.

Function Module `HR99B_PARALLEL_REPORT_RUN` is designed to execute reports *remotely*.

Characteristics:

1. **Covertness**
2. **Bypass**
3. **Intent**



VF Advisory: **SAP-BACK-06**

SAP Note: 1592312

CVSS Base Score: 3.5

CVSS Base Vector: AV:N/AC:M/Au:S/C:N/I:P/A:N

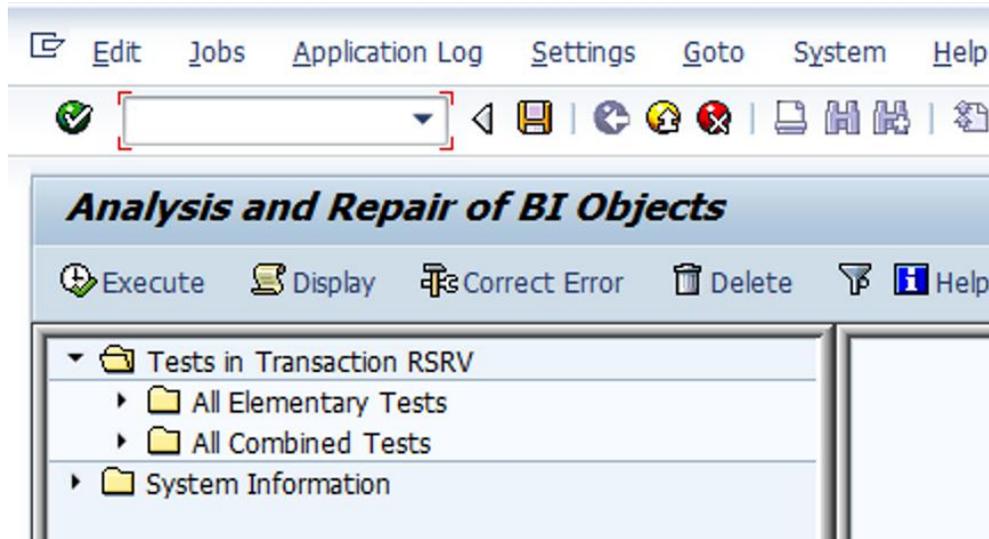
SAP: 'BACKDOOR'

VF Advisory: **SAP-BACK-04**

SAP Note: 1558284

CVSS Base Score: 8.2

CVSS Base Vector: AV:N/AC:M/AU:S/C:C/I:C/A:P



Characteristics:

1. **Coverttness**
2. Bypass
3. **Intent**



Security Audits of a BSP (Web) Application of an SAP customer

One of the pages appeared to be blank

In the source code, the page checked for the usernames of three external developers...

...and would allow them to read data from a table of their choice in the SAP database

- Financial data
- Production data
- HR data ...



Generic table reader in BSP page.

Characteristics:

1. **Covertness**
2. **Bypass**
3. **Intent**



A nice backdoor and 100% remote accessible

How do you prevent Backdoors?



- Perform peer reviews of *all* code
 - The backdoor can be everywhere
- Check for proprietary authorization logic / unusual options
- Check for (unexpected) modifications to the database
- Check for generic database access
- Prohibit certain coding practices by strict guidelines but don't rely on them



- Use static code analysis to detect suspicious code
- Check for command execution based on input
 - ABAP
 - Operating system
- Expect stealth techniques
 - Dynamic ABAP
 - Hidden OK Codes
 - #EC suppression
 - ...

Summary



- ABAP code can have backdoors
- Backdoors are difficult to spot
 - Designed to be covert
 - „Needle in the haystack“
- Check the background of your (external) developers
- Perform code audits before productive use
- Perform static code analysis as additional line of defense



Links

SAP Security Advisories researched by Virtual Forge
<http://www.codeprofilers.com/index.php/advisories.html>

Organizations



BIZEC – Business Security Initiative
<http://www.bizec.org>

Literature



Sichere ABAP-Programmierung
(SAP PRESS, 372 S., 2009)
*Andreas Wiegenslein, Markus Schumacher,
Sebastian Schinzel, Frederik Weidemann*



Questions?

McFly:

“Listen, you got a backdoor to this place?”

Bartender:

“Yeah, it's in the back.”

(Back to the Future III, 1990)



VIRTUALFORGE
we harden your software

Contact Information

VIRTUALFORGE GmbH

andreas.wiegenstein@virtualforge.de

Speyerer Straße 6
69115 Heidelberg
Deutschland

Telefon: + 49 (0) 6221 86 89 0 - 0

Fax: + 49 (0) 6221 86 89 0 - 101



SAP, R/3, ABAP, SAP GUI, SAP NetWeaver and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only.

The authors assume no responsibility for errors or omissions in this document. The authors do not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

The authors shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of this document. Hippies are not supposed to read this. No exceptions.

No part of this document may be reproduced without the prior written permission of Virtual Forge GmbH.

© 2012 Virtual Forge GmbH.