Remotely Owning 'Secure' Parking Systems

José Antonio Guasch

May 2015

WEB

www.securitybydefault.com

Index

Table of Contents

Introduction

While doing some research on parking management systems and associated technologies, I came across a specific manufacturer offering it's customers the possibility of complete remote management of their parking systems with the ability to manage parking rates on-the-fly, view connected security cameras and even control barriers.

Next, a world map showing the location of parking management systems connected to the Internet and therefore accessible by any user:

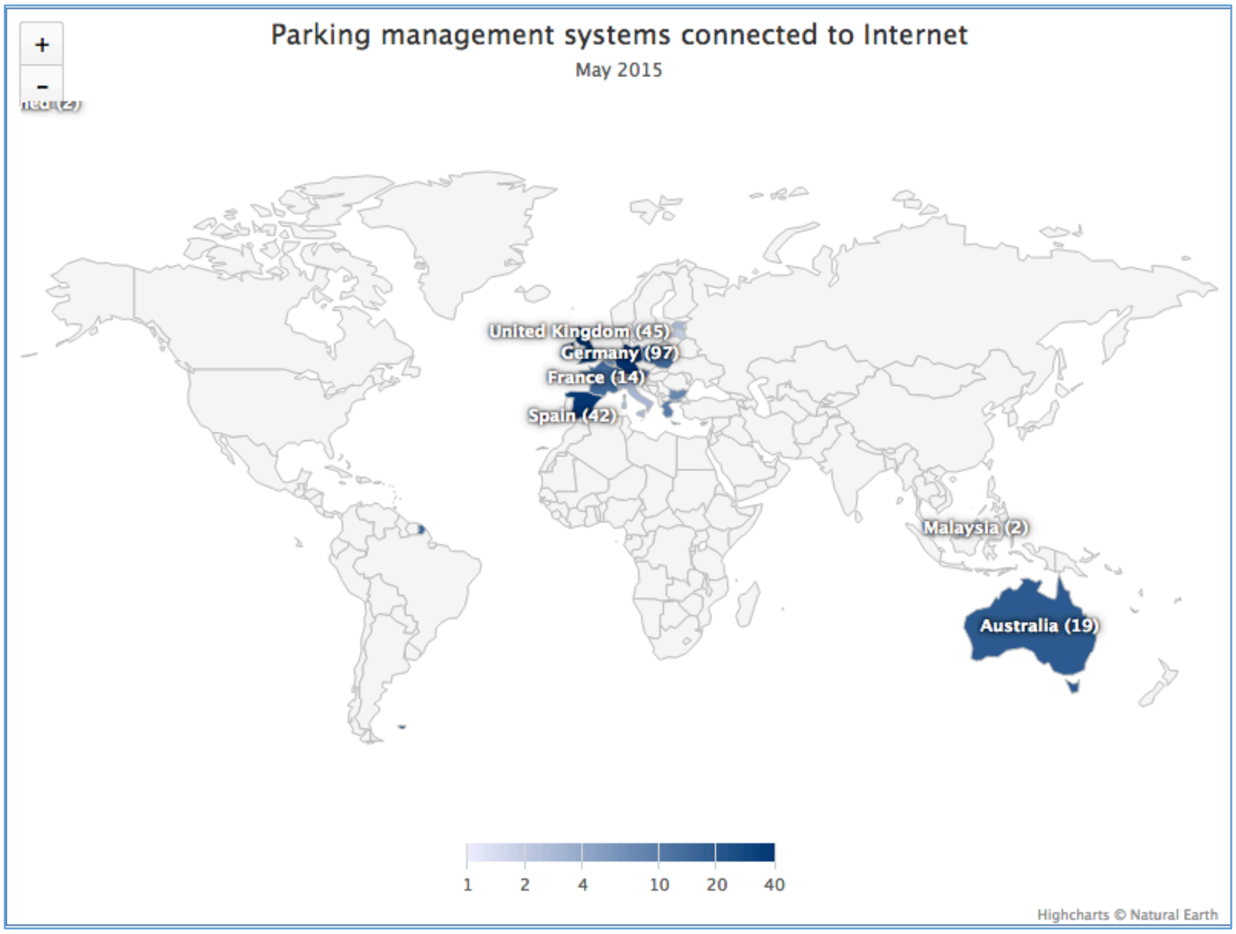


Figure 1 Worldwide distribution of vulnerable parking management systems

Executive Summary

By default, these management systems are insecure. Because the implementation does not follow typical security settings, all facilities contain certain vulnerabilities, which once discovered, would compromise any of these systems.

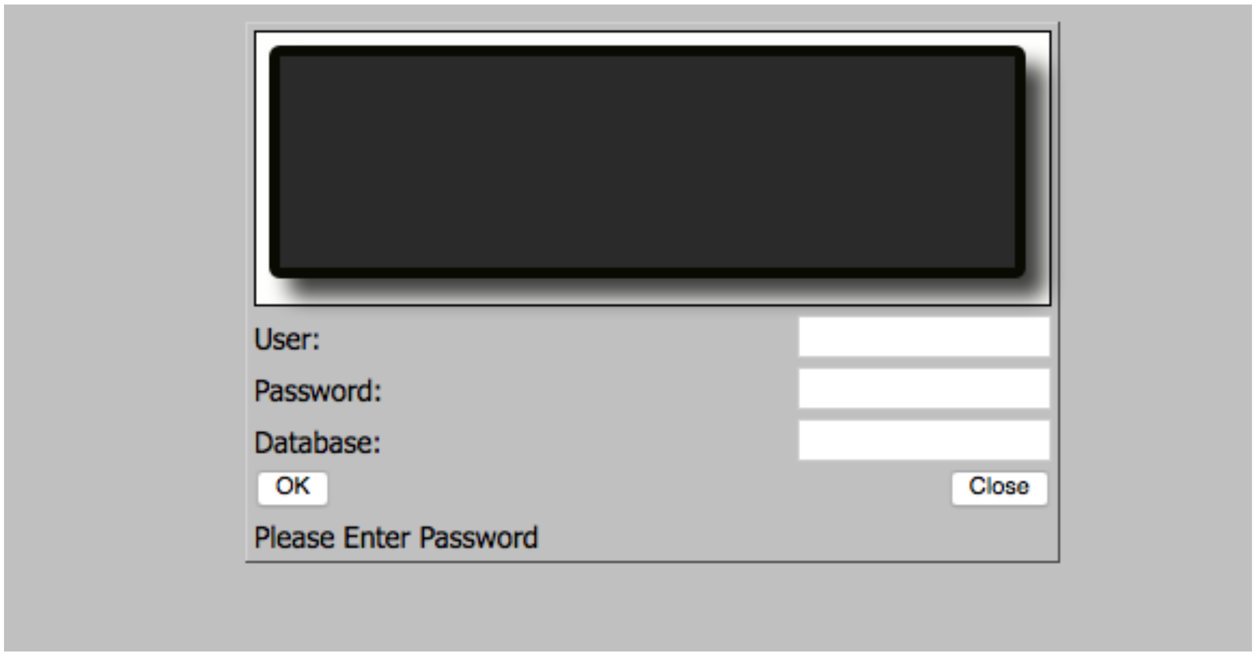
It was possible to locate certain keywords which, using different search engines, allow us to collect the main servers of parking management systems that are connected to the Internet.

- Through a web-accessible file, it is possible to obtain detailed information of the operating system environment, like the hostname and the internal path to the document web root.
- The application manages the backups, saving them to a specific path in the file system. This path is predictable and can be crafted knowing the hostname. It is possible to download backups from both the operative system and the parking management application database, directly for one model and taking advantage of vulnerability in other one.
- There is a path traversal vulnerability that allows downloading every readable file in the system, including application source code, logs, configuration files and scripts.
- The password policy implementation for both the application and system is weak, helping attackers to conduct brute force or password guessing attacks on user accounts to obtain valid credentials.
- Because the main server has access to other devices in the local network, it is possible to pivot from there (which is the only system connected to the Internet) to the rest of management systems (cameras, payment stations, cashier computers, etc)

Scope of research

Penetration test (web focused) against architecture behind the following login forms from two parking models:

Parking Model A (used before 2008)



[redacted]

User:

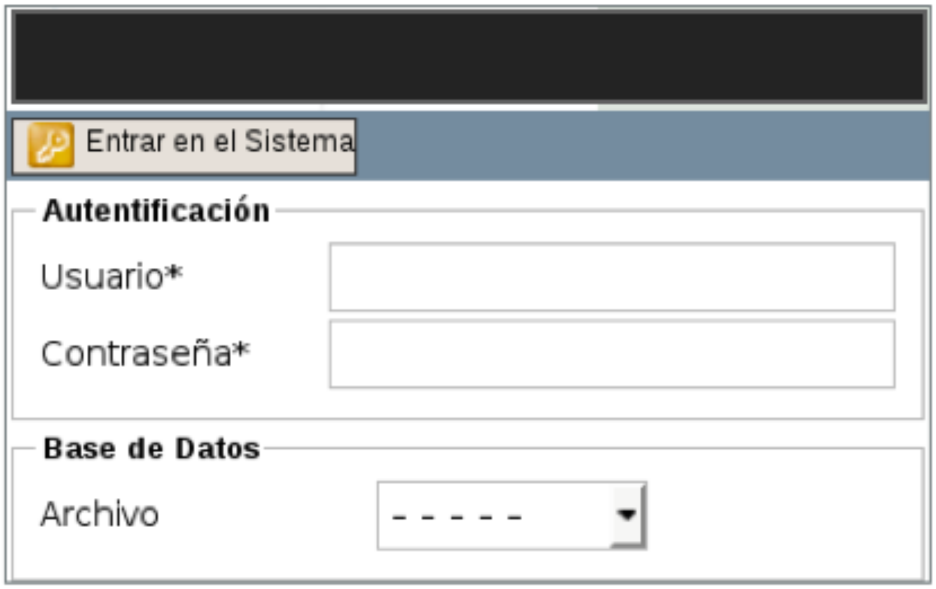
Password:

Database:

Please Enter Password

Figure 2 - Login form for parking model A

Parking Model B (newest product line)



[redacted]

Entrar en el Sistema

Autentificación

Usuario*

Contraseña*

Base de Datos

Archivo - - - - -

Figure 3 - Login form for parking model B

Parking Architecture

Even the research includes some different parking management systems models as the scope; it is possible to summarize the typical architecture of these kinds of systems with the following diagram, thanks to brochures and whitepapers from the vendor itself:

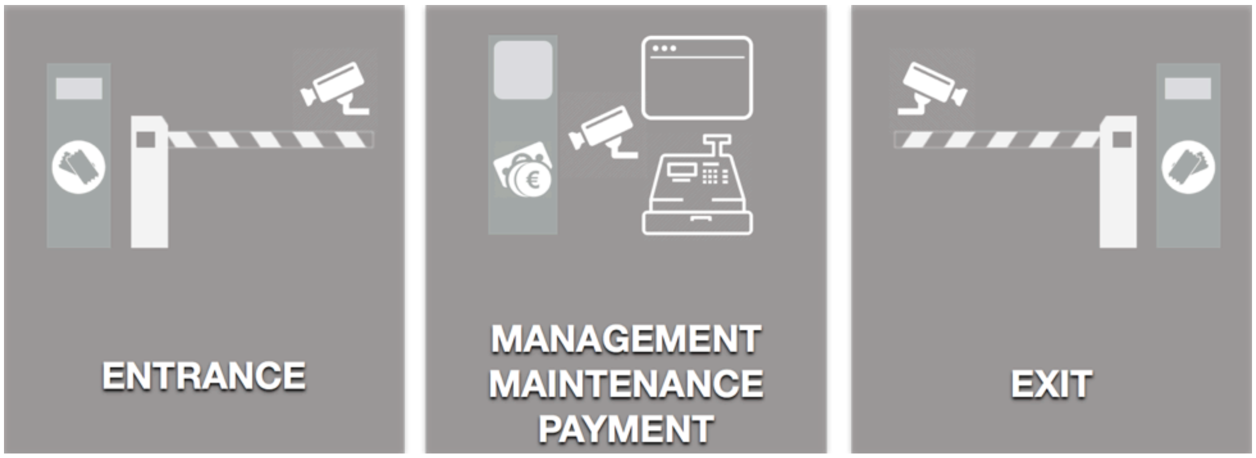


Figure 4 - General architecture for parking management systems

Entry/Exit elements

Both entry and exit environments have similar devices connected to the parking network. Most important of them are the following:

- **Lane Controller:** ticket dispensing, ticket verifying and season cards reading.
- **Barrier:** device which control the barrier of the parking, accepting several modes for its operation (manual, semi and full automatic)
- **License Plate Recognition:** element that is in charge of reading license plates and sends them to the management. These license plates identify cars and are used to link tickets and season cards.
- **Cameras (optional):** cameras placed at lane controllers or at entries and exits, which gives the possibility to visually identify drivers and cars from other angles. Mainly for security purposes and assistance.

Management elements

Following are the main components used at the parking system for its management, maintenance and payments

- **Management Application:** Main web application, which can manage the whole parking, its parameters and configurations, as well as financial, and presence reports. It has the ability to control the rest of the parking devices connected to the network. It is the key component for the whole infrastructure.
- **Payment station:** Station used for clients to pay tickets and receive receipts. Credit cards and coins accepted as payment methods.
- **Cashier:** System used by parking workers to manage tickets and payments for clients as an alternative of using payment stations.

Vulnerabilities in parking management systems

Vulnerabilities discovered are valid for every parking management system installation connected to the Internet from the vendor. While normally some of these vulnerabilities are often categorized as low in any security audit, used together can cause a full system compromise.

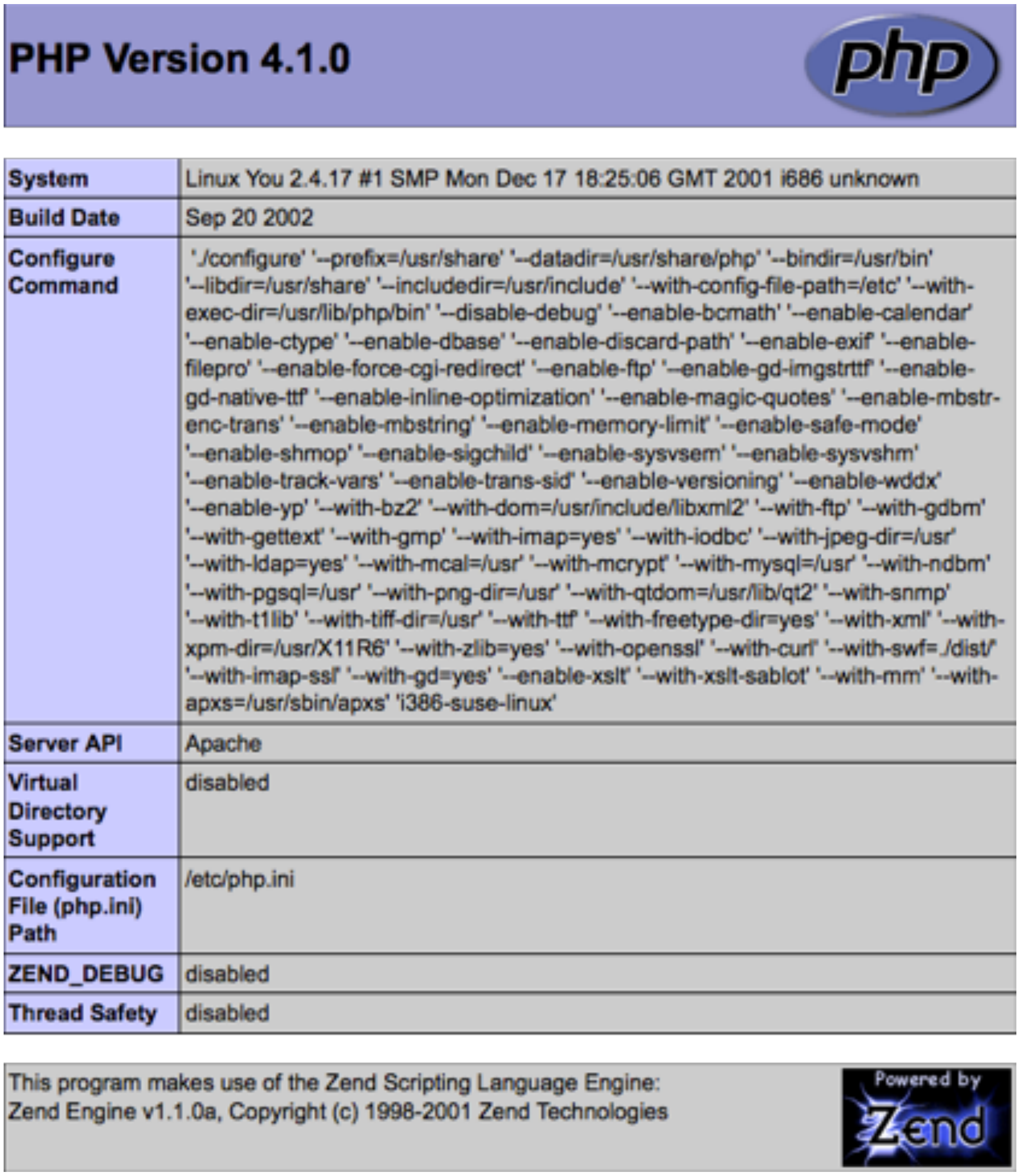
The aim of the research is to detect common insecure configurations and vulnerabilities that once exploited, could compromise any system of the same model.

VULNERABILITY	STATUS	
	MODEL A	MODEL B
Environment information disclosure	Vulnerable	Vulnerable
Directory Listing	Vulnerable	Not vulnerable
Access to backup files	Vulnerable	Vulnerable
Plain text passwords in database	Vulnerable	Vulnerable
Weak password policy	Vulnerable	Vulnerable
Path traversal in download page	Not vulnerable	Vulnerable
Predictable system users/passwords	Not vulnerable	Vulnerable
Predictable application users/passwords	Not vulnerable	Vulnerable

Environment information disclosure

There is a file available in the main folder for the management application called **info.php**, which includes the output of the PHP command phpinfo(). Includes information about PHP compilation options and extensions, PHP version, server information and environment (if compiled as a module), the PHP environment, OS version information, paths, etc. It is mainly used for debugging purposes and should not be available in production environments.

From this output, we are interested in information related to the hostname and full path where the document root is.

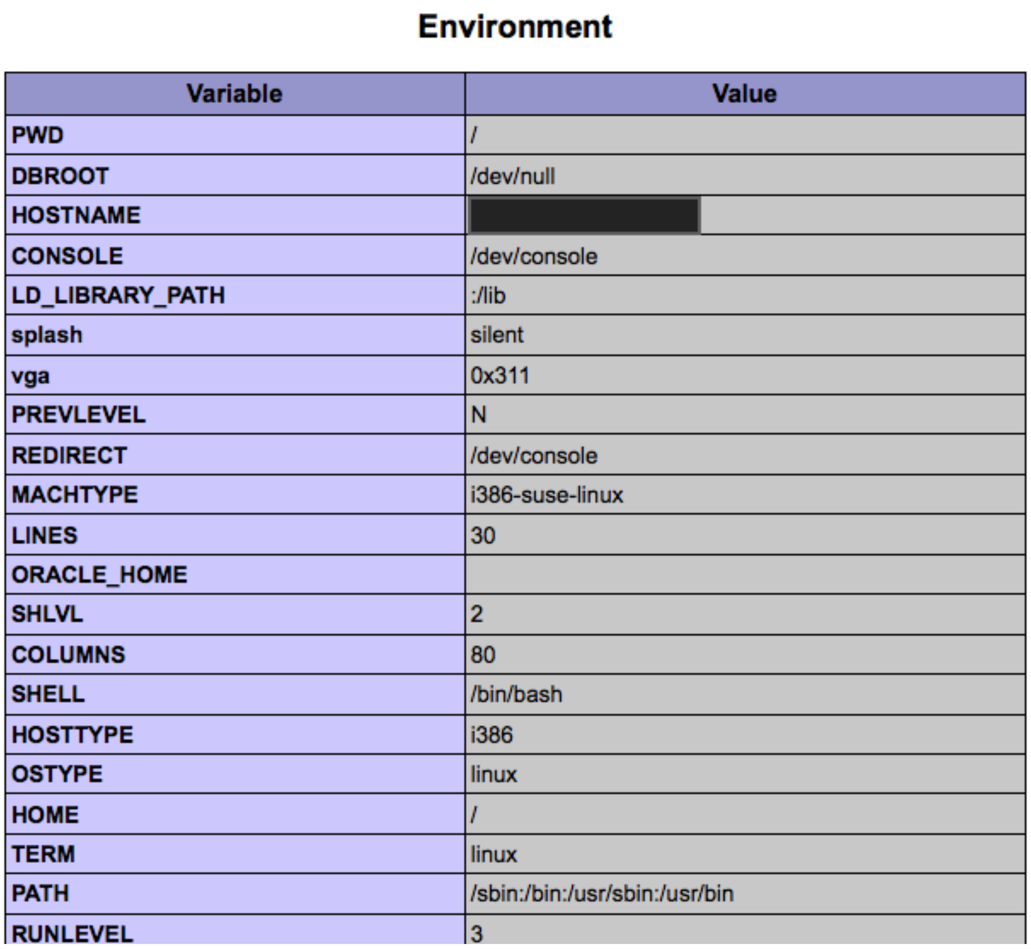


PHP Version 4.1.0

System	Linux You 2.4.17 #1 SMP Mon Dec 17 18:25:06 GMT 2001 i686 unknown
Build Date	Sep 20 2002
Configure Command	'./configure' '--prefix=/usr/share' '--datadir=/usr/share/php' '--bindir=/usr/bin' '--libdir=/usr/share' '--includedir=/usr/include' '--with-config-file-path=/etc' '--with-exec-dir=/usr/lib/php/bin' '--disable-debug' '--enable-bcmath' '--enable-calendar' '--enable-ctype' '--enable-dbase' '--enable-discard-path' '--enable-exif' '--enable-filepro' '--enable-force-cgi-redirect' '--enable-ftp' '--enable-gd-imgstrttf' '--enable-gd-native-ttf' '--enable-inline-optimization' '--enable-magic-quotes' '--enable-mbstr-enc-trans' '--enable-mbstring' '--enable-memory-limit' '--enable-safe-mode' '--enable-shmop' '--enable-sigchild' '--enable-sysvsem' '--enable-sysvshm' '--enable-track-vars' '--enable-trans-sid' '--enable-versioning' '--enable-wddx' '--enable-yp' '--with-bz2' '--with-dom=/usr/include/libxml2' '--with-ftp' '--with-gdbm' '--with-gettext' '--with-gmp' '--with-imap=yes' '--with-iodbc' '--with-jpeg-dir=/usr' '--with-ldap=yes' '--with-mcal=/usr' '--with-mcrypt' '--with-mysql=/usr' '--with-ndbm' '--with-pgsql=/usr' '--with-png-dir=/usr' '--with-qtdom=/usr/lib/qt2' '--with-snmp' '--with-t1lib' '--with-tiff-dir=/usr' '--with-ttf' '--with-freetype-dir=yes' '--with-xml' '--with-xpm-dir=/usr/X11R6' '--with-zlib=yes' '--with-openssl' '--with-curl' '--with-swf=./dist/' '--with-imap-ssl' '--with-gd=yes' '--enable-xslt' '--with-xslt-sablot' '--with-mm' '--with-apxs=/usr/sbin/apxs' 'i386-suse-linux'
Server API	Apache
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php.ini
ZEND_DEBUG	disabled
Thread Safety	disabled

This program makes use of the Zend Scripting Language Engine:
Zend Engine v1.1.0a, Copyright (c) 1998-2001 Zend Technologies

Figure 5 - PHP summary information



Environment

Variable	Value
PWD	/
DBROOT	/dev/null
HOSTNAME	[redacted]
CONSOLE	/dev/console
LD_LIBRARY_PATH	:/lib
splash	silent
vga	0x311
PREVLEVEL	N
REDIRECT	/dev/console
MACHTYPE	i386-suse-linux
LINES	30
ORACLE_HOME	
SHLVL	2
COLUMNS	80
SHELL	/bin/bash
HOSTTYPE	i386
OSTYPE	linux
HOME	/
TERM	linux
PATH	/sbin:/bin:/usr/sbin:/usr/bin
RUNLEVEL	3

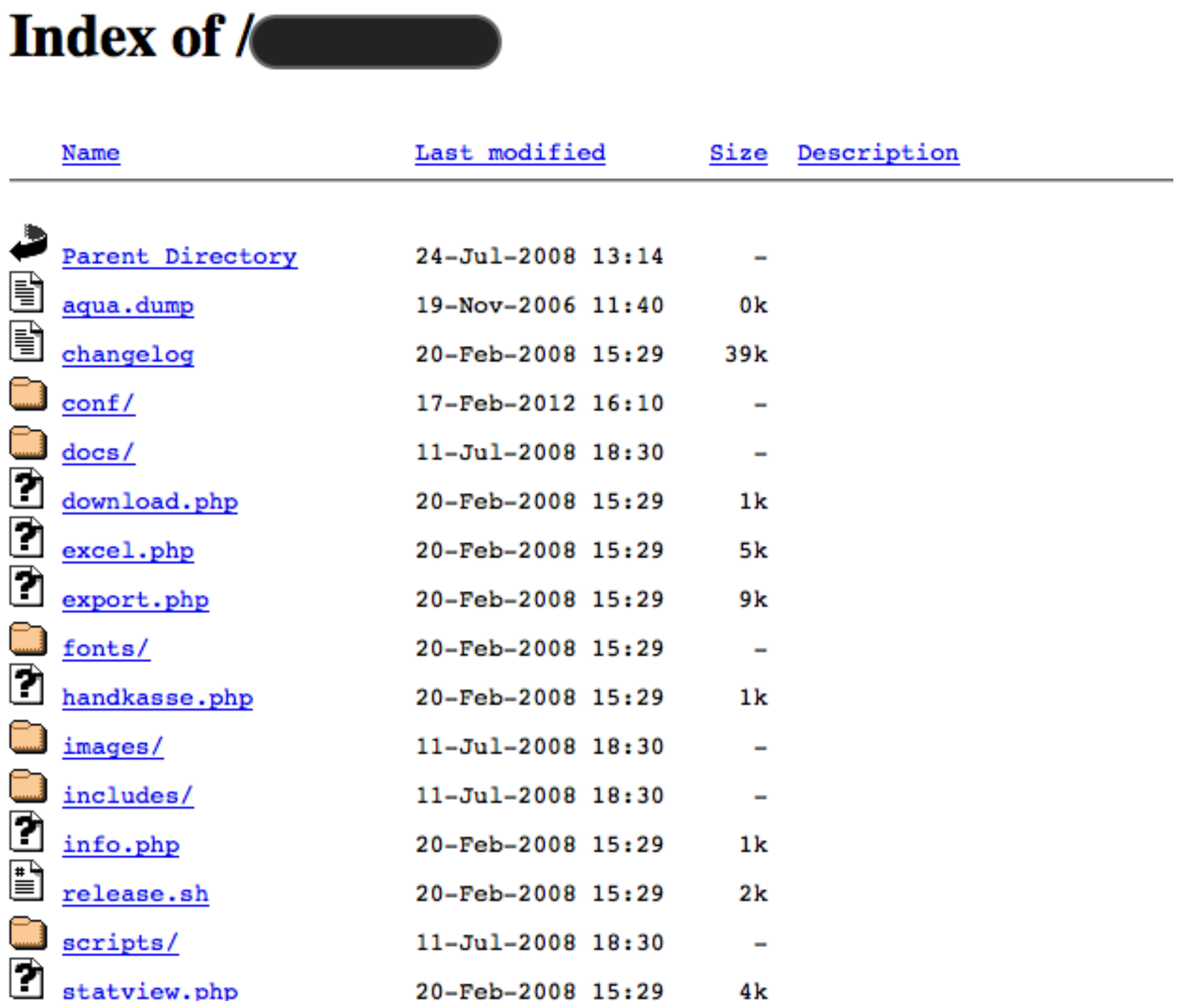
Figure 6 - Environment information

Directory listing

A directory listing exposes the complete index of all the resources located inside of the directory. The specific risks and consequences vary depending on which files are listed and accessible.

There are some predictable folders at the document root, which are browsable because directory listing is enabled at the web server.

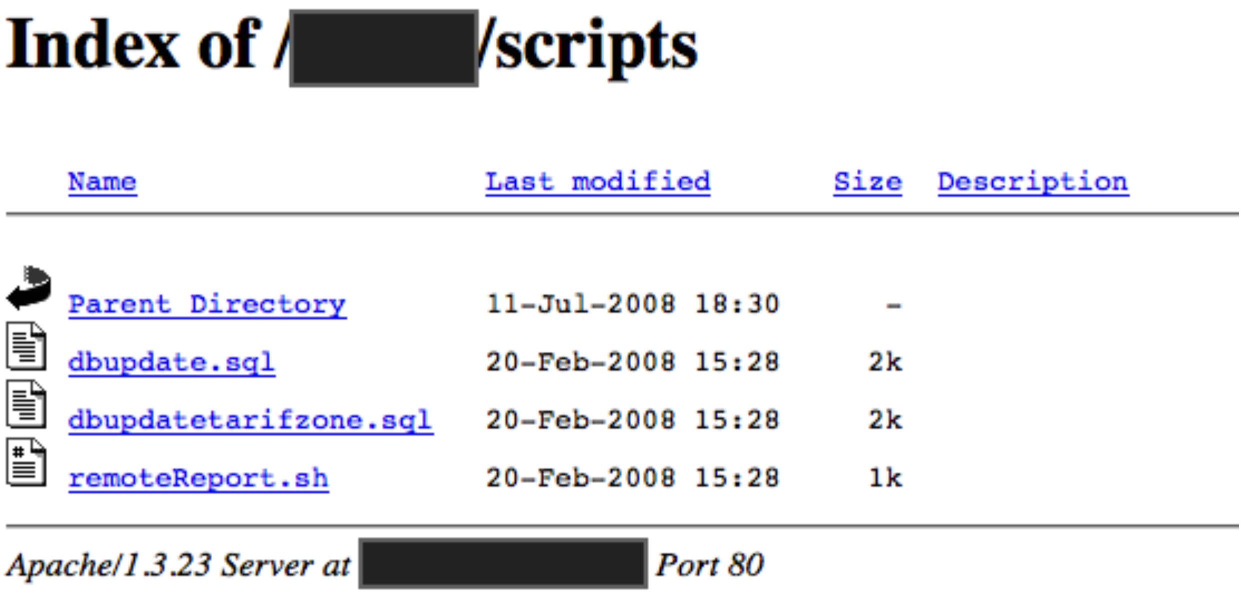
Typical folders that are browsable are **scripts**, **uploads**, **docs**, **conf** and also the **main folder** for the parking application (which is not at the first level of the document root):



Index of /[redacted]

```
     Name                    Last modified       Size  Description
-----------------------------------------------------------------------
     Parent Directory        24-Jul-2008 13:14      -
     aqua.dump               19-Nov-2006 11:40     0k
     changelog               20-Feb-2008 15:29    39k
     conf/                   17-Feb-2012 16:10      -
     docs/                   11-Jul-2008 18:30      -
     download.php            20-Feb-2008 15:29     1k
     excel.php               20-Feb-2008 15:29     5k
     export.php              20-Feb-2008 15:29     9k
     fonts/                  20-Feb-2008 15:29      -
     handkasse.php           20-Feb-2008 15:29     1k
     images/                 11-Jul-2008 18:30      -
     includes/               11-Jul-2008 18:30      -
     info.php                20-Feb-2008 15:29     1k
     release.sh              20-Feb-2008 15:29     2k
     scripts/                11-Jul-2008 18:30      -
     statview.php            20-Feb-2008 15:29     4k
```

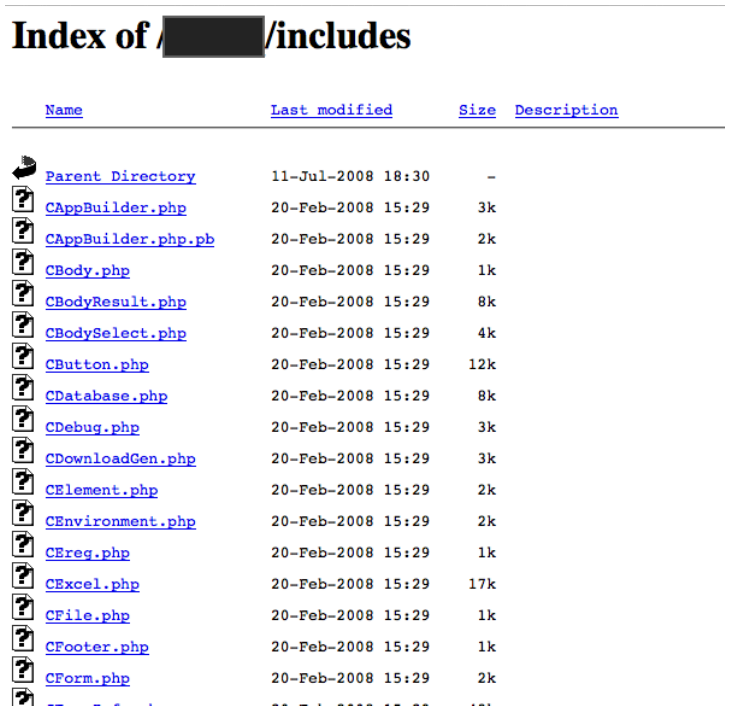
Figure 7 - Directory listing at main parking application folder



```
Index of /[redacted]/scripts

    Name                      Last modified      Size  Description
------------------------------------------------------------------
    Parent Directory          11-Jul-2008 18:30     -
    dbupdate.sql              20-Feb-2008 15:28    2k
    dbupdatetarifzone.sql     20-Feb-2008 15:28    2k
    remoteReport.sh           20-Feb-2008 15:28    1k
------------------------------------------------------------------
Apache/1.3.23 Server at [redacted] Port 80
```

Figure 8 - Content of scripts folder



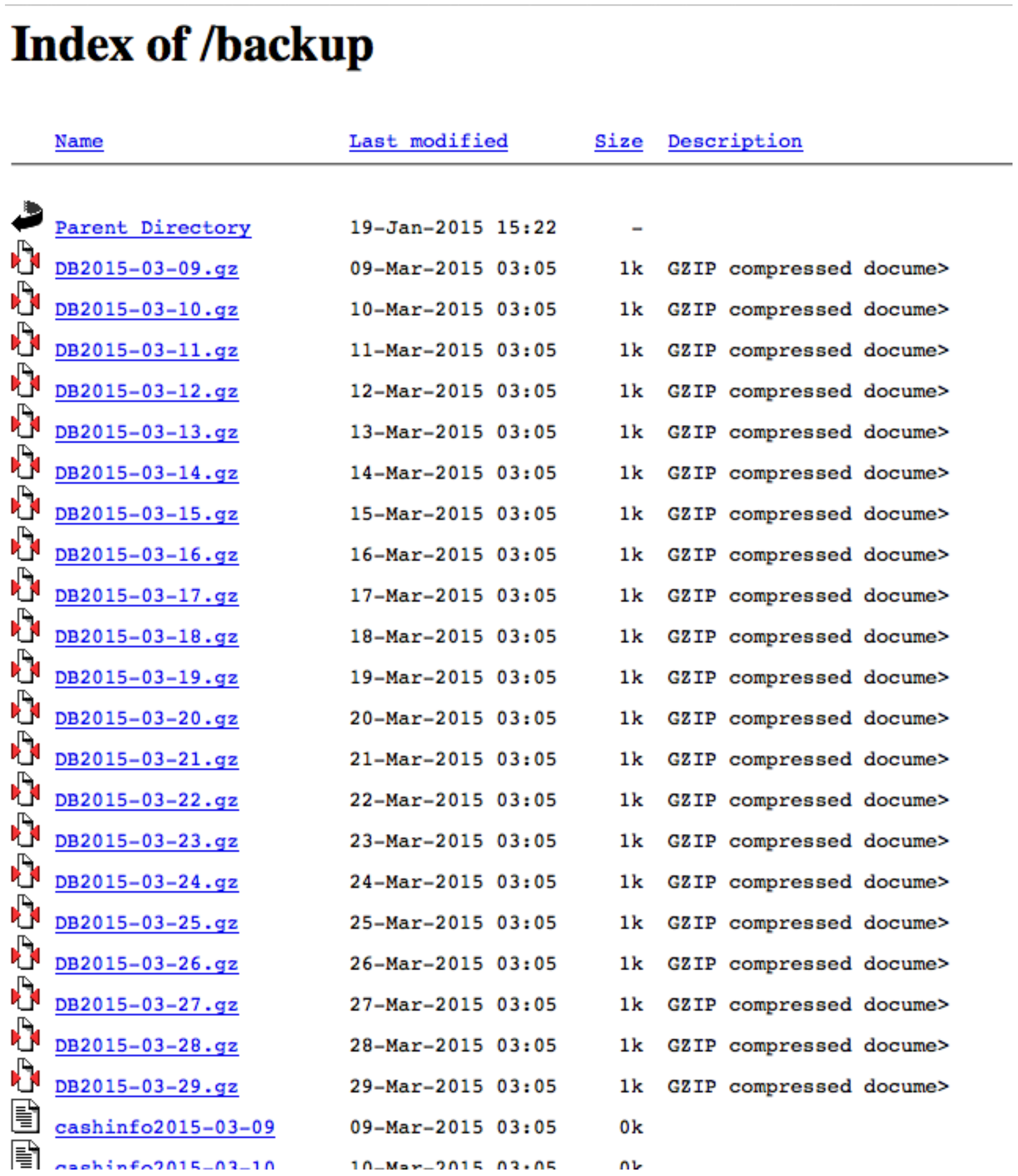
```
Index of /[redacted]/includes

    Name                      Last modified      Size  Description
------------------------------------------------------------------
    Parent Directory          11-Jul-2008 18:30     -
    CAppBuilder.php           20-Feb-2008 15:29    3k
    CAppBuilder.php.pb        20-Feb-2008 15:29    2k
    CBody.php                 20-Feb-2008 15:29    1k
    CBodyResult.php           20-Feb-2008 15:29    8k
    CBodySelect.php           20-Feb-2008 15:29    4k
    CButton.php               20-Feb-2008 15:29   12k
    CDatabase.php             20-Feb-2008 15:29    8k
    CDebug.php                20-Feb-2008 15:29    3k
    CDownloadGen.php          20-Feb-2008 15:29    3k
    CElement.php              20-Feb-2008 15:29    2k
    CEnvironment.php          20-Feb-2008 15:29    2k
    CEreg.php                 20-Feb-2008 15:29    1k
    CExcel.php                20-Feb-2008 15:29   17k
    CFile.php                 20-Feb-2008 15:29    1k
    CFooter.php               20-Feb-2008 15:29    1k
    CForm.php                 20-Feb-2008 15:29    2k
```

Figure 9 - Content of includes folder

Backup files available

While most of the files within a web server are directly handled by the server itself, it isn't uncommon to find unreferenced or forgotten files that can be used to obtain important information about the infrastructure or the credentials. There is a **backup** folder at the document web root of the application. The configurations of the web server also allows browsing its full content, as well as access and download the files listed.



Index of /backup

```
Name                      Last modified       Size  Description

Parent Directory          19-Jan-2015 15:22     -
DB2015-03-09.gz           09-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-10.gz           10-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-11.gz           11-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-12.gz           12-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-13.gz           13-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-14.gz           14-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-15.gz           15-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-16.gz           16-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-17.gz           17-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-18.gz           18-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-19.gz           19-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-20.gz           20-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-21.gz           21-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-22.gz           22-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-23.gz           23-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-24.gz           24-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-25.gz           25-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-26.gz           26-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-27.gz           27-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-28.gz           28-Mar-2015 03:05    1k   GZIP compressed docume>
DB2015-03-29.gz           29-Mar-2015 03:05    1k   GZIP compressed docume>
cashinfo2015-03-09        09-Mar-2015 03:05    0k
cashinfo2015-03-10        10-Mar-2015 03:05    0k
```

Figure 10 - Browsable backup folder

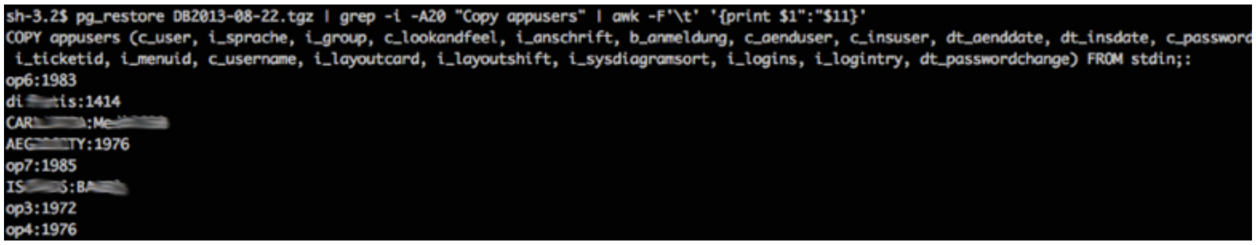
The folder contains compressed files of daily database snapshots of the current month, including tickets information and the whole configuration parameters for the parking itself. Everybody without restrictions can download every of these files.

Plain text passwords in database dump

After decompressing the backup compressed file, the dump is a typical PostgreSQL export. Analyzing the database dump and its content, one of the tables included is called **appusers**. It contains every management users details, as well as their passwords in plain text.

It is possible to quickly get a full list of users and passwords with a command like the following against the compressed backup file:

```
pg_restore X.tgz l grep -i -A20 "Copy appusers" l awk -F'\t' '{print $1":"$11}'
```



```
sh-3.2$ pg_restore DB2013-08-22.tgz | grep -i -A20 "Copy appusers" | awk -F'\t' '{print $1":"$11}'
COPY appusers (c_user, i_sprache, i_group, c_lookandfeel, i_anschrift, b_anmeldung, c_aenduser, c_insuser, dt_aenddate, dt_insdate, c_passwore
 i_ticketid, i_menuid, c_username, i_layoutcard, i_layoutshift, i_sysdiagramsort, i_logins, i_logintry, dt_passwordchange) FROM stdin;:
op6:1983
di[redacted]is:1414
CAR[redacted]:Me[redacted]
AEG[redacted]TY:1976
op7:1985
IS[redacted]S:B[redacted]
op3:1972
op4:1976
```

Figure 11 - Fragment of users and passwords list

One of the users corresponds with **admin**, which has full access for the main web application and to manage the whole parking system and its components. There are other users with management roles, like **manager**.

Weak password policy

After obtaining the list of users and passwords dump database, it is possible to conclude that there is no password policy. The system allows introducing weak passwords, with no minimum length, and even can match the username.

This problem allows an attacker to perform brute force or dictionary and have a high probability of success.

Path traversal in download page

As defined by OWASP methodology, a path traversal attack aims to access files and directories that are stored outside the web root folder. By browsing the application, the attacker looks for absolute links to files stored on the web server. By manipulating variables that reference files with “dot-dot-slash (../)” sequences and its variations, it may be possible to access arbitrary files and directories stored on file system

Parking model B contains a path traversal in “file” parameter for a download function. It is possible to obtain every file in the system.

Also, full path for backup files is predictable, so it is possible to download last database dump with this vulnerability.

Predictable application user/passwords

Default accounts for parking models in scope are:

- admin
- manager
- test

Both “**admin**” and “**manager**” have full privileges for managing the whole parking system, and “**test**” user only has access for downloading manuals and changes the account password.

Default passwords for these accounts are the same as the username. Even **admin** and **manager** password could be changed at installation; **test** account is not usually modified.

Accessing with **test** account is enough to exploit the path traversal vulnerability explained in last section.

Predictable system user/passwords

Taking advantage of directory traversal vulnerability is possible to obtain a list of valid system users for the parking management system main server. There are users with shell access (using SSH) and with access only to the FTP service.

Knowing the weak password policy, FTP users have the same password as the username.

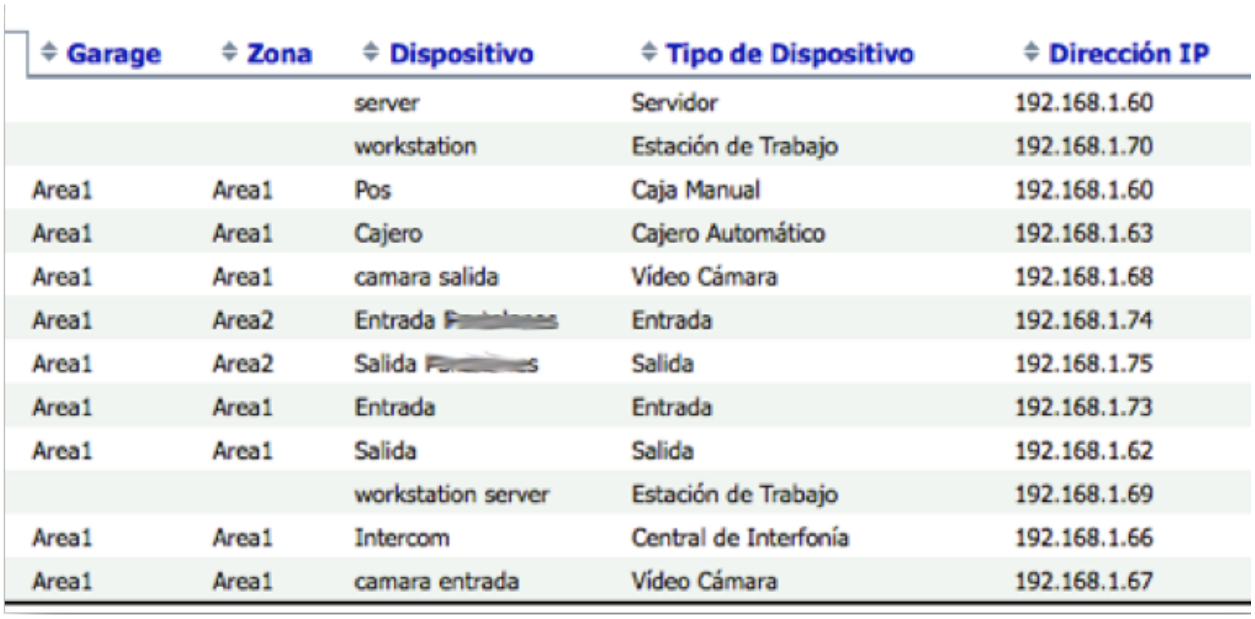
Accessing through FTP, it is possible to download backups for the system (as well as the parking database), like for example **home**, **tftpboot** and the whole **etc** folder (where the configuration fles are). Inside this backup, the **shadow** file is included, and is readable for everyone that can decompress the compressed file.

Cracking the *shadow* file, it is possible to obtain the common password for the rest of users with shell access, including **root**.

Vulnerabilities impact

Full access to any connected parking device

As explained in the Parking Architecture section, the parking management system manages a central server as well as any of the devices attached to the parking network, including workstations, cashiers, cameras…



Garage	Zona	Dispositivo	Tipo de Dispositivo	Dirección IP
		server	Servidor	192.168.1.60
		workstation	Estación de Trabajo	192.168.1.70
Area1	Area1	Pos	Caja Manual	192.168.1.60
Area1	Area1	Cajero	Cajero Automático	192.168.1.63
Area1	Area1	camara salida	Vídeo Cámara	192.168.1.68
Area1	Area2	Entrada [redacted]	Entrada	192.168.1.74
Area1	Area2	Salida [redacted]	Salida	192.168.1.75
Area1	Area1	Entrada	Entrada	192.168.1.73
Area1	Area1	Salida	Salida	192.168.1.62
		workstation server	Estación de Trabajo	192.168.1.69
Area1	Area1	Intercom	Central de Interfonía	192.168.1.66
Area1	Area1	camara entrada	Vídeo Cámara	192.168.1.67

Figure 12 - Parking devices

Because there is an inventory in a management menu, once access to the central server, you can also connect via SSH to other systems jumping from there.

Season cards control

Using the main application, it is possible to maintain and even generate new season cards associate it with a car, as well as compromising existing ones, managing the available credit, fetching sensitive information and create “free and unlimited” access cards.

Surveillance access

Abusing the parking management system, it is possible to use the parking for surveillance tasks. If the client has an account at the parking, and its personal information is registered, it is possible to know his presence, schedules, when enter or leave the parking…

If the parking has a system of cameras installed, you can access directly to live streaming or even past recordings.

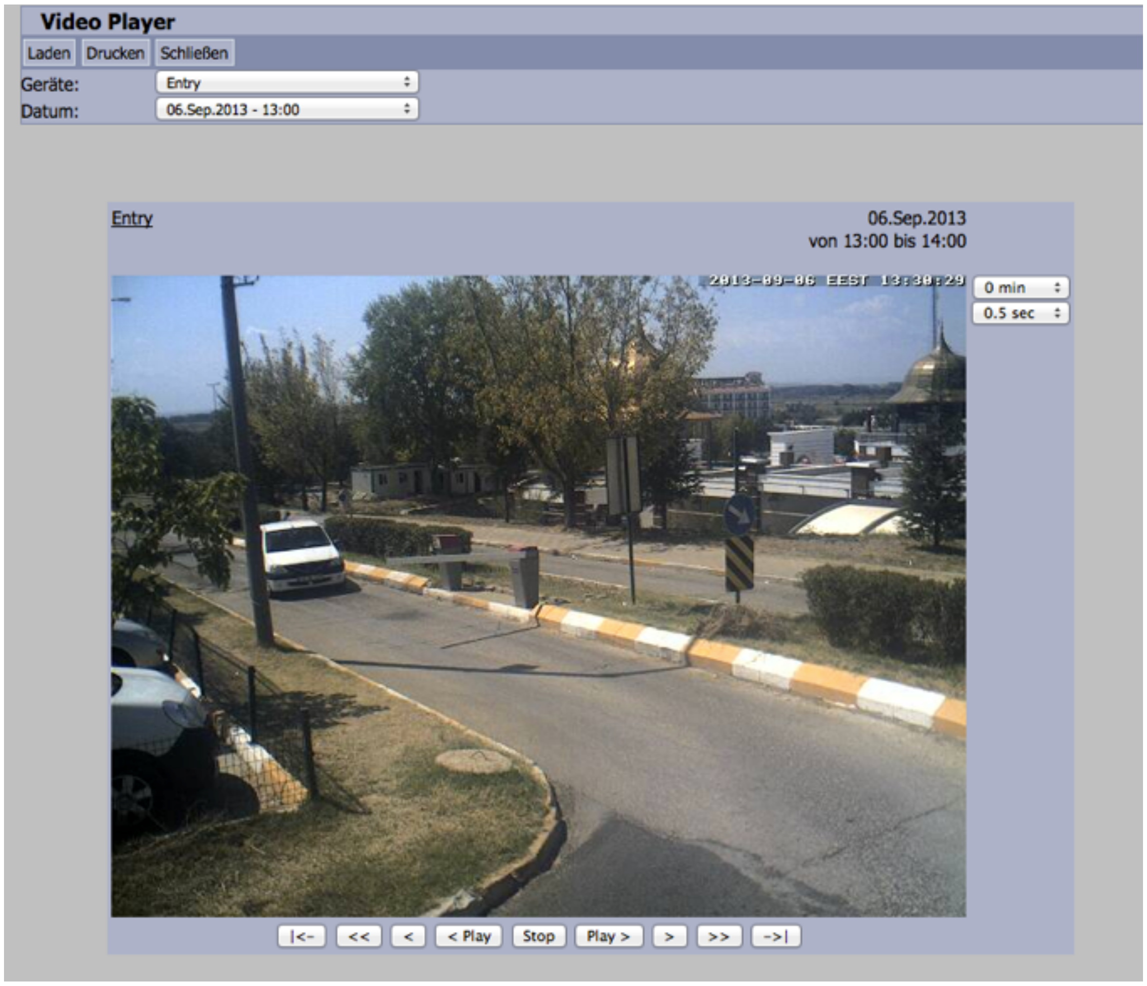


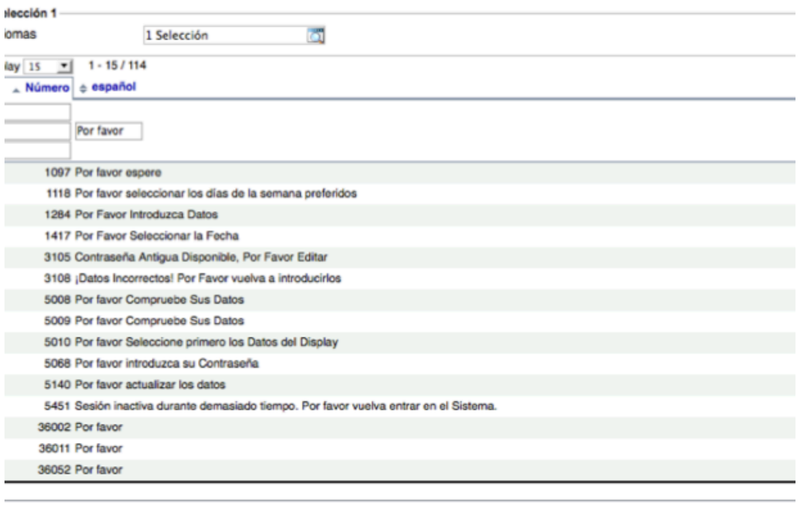
Figure 13. Access to cameras installed on the system

Credit card information

Because of full access of database dump, and credit cards information are stored in there (full number), it is possible to link information of people with its payment methods, such credit cards, establishing relation between tickets and associated payments.

Forging communications and messages

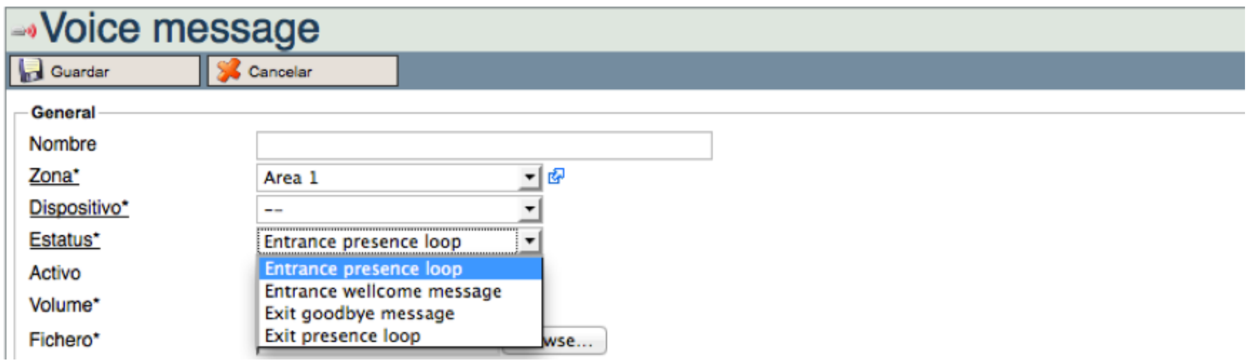
The management system allows high privileged users to access the intercom messages, as well as the tool to edit every string used in the parking devices. It is possible to upload new messages for the entry and exit systems, as well as forge every string of the cashier and payment stations.



lección 1
iomas 1 Selección
lay 15 1 - 15 / 114

Número	español
	Por favor
1097	Por favor espere
1118	Por favor seleccionar los días de la semana preferidos
1284	Por Favor Introduzca Datos
1417	Por Favor Seleccionar la Fecha
3105	Contraseña Antigua Disponible, Por Favor Editar
3108	¡Datos Incorrectos! Por Favor vuelva a introducirlos
5008	Por favor Compruebe Sus Datos
5009	Por favor Compruebe Sus Datos
5010	Por favor Seleccione primero los Datos del Display
5068	Por favor introduzca su Contraseña
5140	Por favor actualizar los datos
5451	Sesión inactiva durante demasiado tiempo. Por favor vuelva entrar en el Sistema.
36002	Por favor
36011	Por favor
36052	Por favor

Figure 14. Devices strings edition menu



Voice message

General

Nombre	
Zona*	Area 1
Dispositivo*	--
Estatus*	Entrance presence loop
Activo	
Volume*	
Fichero*	

Entrance presence loop
Entrance wellcome message
Exit goodbye message
Exit presence loop

Figure 15. Section to change voice messages for entrance and exit

Parking status and barriers

With full access to the management system it is possible to fake the state of parking, set it as full and deny access to other cars. Because the barrier system is controlled independently, it is possible to manage them manually.

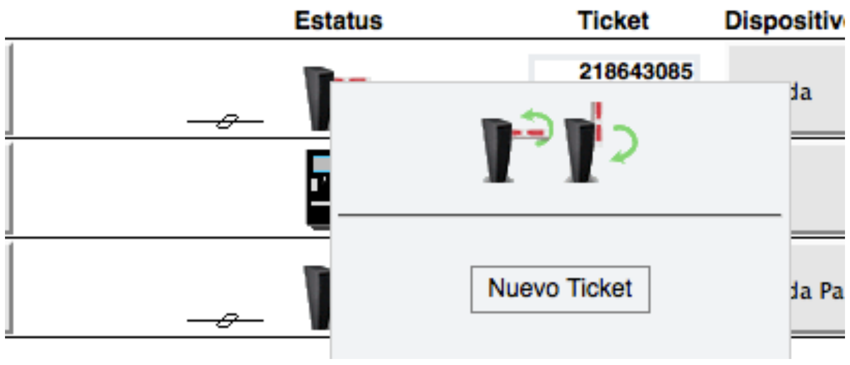


Figure 16. Barriers control

About the author

Jose Antonio Guasch is a Spanish security researcher who works as a technical coordinator of the Tiger Team in a Spanish security-consulting firm. With more than eight years of experience, his specialties are ethical hacking, pentesting, forensics, systems and webapp security, etc. Jose is one of the main editors of the awarded Spanish security blog SecurityByDefault.com, translator of several OWASP projects and one of the organizers of the Rooted CON security congress held in Madrid, where he gave a talk about the (in)security of several components used to manage user certificates through the browser, specifically the ones related with the Spanish Electronic ID.