 clsource / **guia-webpay**

Watch24

Star106

Fork39

<> Code

Issues0


Pull requests0









Pulse

Graphs

Branch: master ▾ **guia-webpay / webpay-kcc /**

Create new fileFind fileHistory

 clsource committed on GitHub agregado error de codigo de producción mal configurado Latest commit 5129dac 9 days ago

..		
 code	evidencia6.html	8 months ago
 examples	agregado nuevo proceso de certificación y documentos de ejemplo	8 months ago
 img/1	agregada imagen de error de producción	9 days ago
 webpay-openshift	Ordenado los directorios	2 years ago
 webpay-php	Agregado codigos de ejemplo disponibles	2 years ago
 webpay-woocommerce	Agregada instalación de Wordpress	2 years ago
 Proceso Certificacion Antiguo.md	agregado nuevo proceso de certificación y documentos de ejemplo	8 months ago
 README.md	agregado error de codigo de producción mal configurado	9 days ago

 README.md

Guía de instalación WebPay KCC en Linux

Una pequeña guía para instalar el sistema de pago de Transbank. La instalación de webpay que se detalla a continuación se ha probado exitosamente en un servidor compartido (shared hosting) con Linux.

Requisitos

- Servidor Linux
- KCC 6.0 Linux 32 Bit o 64 Bit
- Apache
- PHP 5.4
- Cliente FTP (Filezilla)

Paso 1 - Subida de Archivos de WebPay

Contenidos de KCC

Primero se debe descomprimir el archivo entregado por transbank, entre todos habrá un directorio llamado *cgi-bin*, éste será el que se debe subir (en modo BINARIO). La forma más efectiva es comprimir ésta carpeta, subirla comprimida y descomprimirla en el mismo servidor.

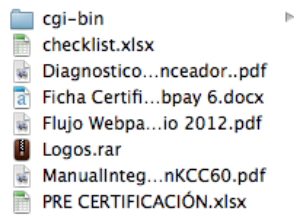


Figura 1: Contenidos del KCC Linux 32 Bit

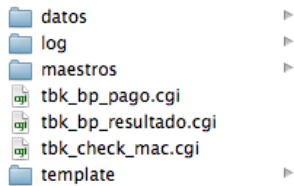


Figura 2: Contenidos de cgi-bin

Debemos crear un archivo llamado *webpay.zip* con los contenidos del directorio cgi-bin.

Archivo unzip.php

Éste script se encargará de descomprimir el archivo *webpay.zip* en el servidor.

```
<?php
/**
 * Descomprime el archivo webpay.zip
 * para asegurar de que los contenidos de cgi-bin
 * esten en modo BINARIO
 */

$zip = new ZipArchive();

$file = 'webpay.zip';

if ($zip->open($file))
{
    $path = pathinfo(realpath($file), PATHINFO_DIRNAME);

    $zip->extractTo($path);

    $zip->close();

    chmod('tbk_bp_pago.cgi', 0755);
    chmod('tbk_bp_resultado.cgi', 0755);
    chmod('tbk_check_mac.cgi', 0755);
    chmod('datos/', 0755);
    chmod('datos/tbk_param.txt', 0755);
    chmod('datos/tbk_config.dat', 0755);
    chmod('datos/tbk_trace.dat', 0755);
    chmod('maestros/', 0755);
    chmod('maestros/privada.pem', 0755);
    chmod('maestros/tbk_public_key.pem', 0755);
    chmod('maestros/tbk_public_key101prod.pem', 0755);
    chmod('log/', 0755);
}
```

```
    echo 'Descomprimido';

}

else
{

    echo 'Error al Descomprimir';
}

/** Fin del Archivo unzip.php */
```

Subiendo Archivos al Servidor

Ahora se debe subir los archivos al servidor, se recomienda tener un directorio con los archivos *webpay.zip* y *unzip.php* para mayor organización.



Figura 3: Contenidos del directorio webpay

Una vez que se suban se verán de la siguiente forma:

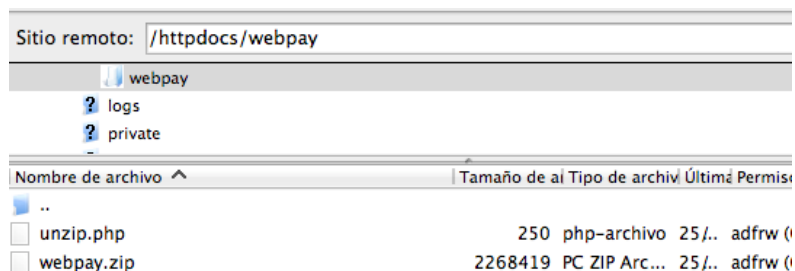


Figura 4: Archivos subidos en el servidor

Luego se debe ejecutar el script *unzip.php* en el servidor

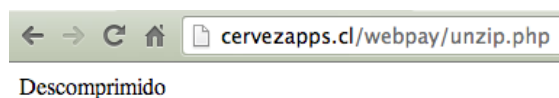


Figura 5: Ejecución del script unzip.php

El contenido del directorio webpay en el servidor quedaría de la siguiente forma:

datos	Directorio	25/..	flcdmpe (0755)
log	Directorio	25/..	flcdmpe (0755)
maestros	Directorio	25/..	flcdmpe (0755)
template	Directorio	25/..	flcdmpe (0755)
tbk_bp_pago.cgi	1601633	cgi-archivo	25/.. adfrw (0644)
tbk_bp_resultado.cgi	1597442	cgi-archivo	25/.. adfrw (0644)
tbk_check_mac.cgi	1597439	cgi-archivo	25/.. adfrw (0644)
unzip.php	405	php-archivo	25/.. adfrw (0644)
webpay.zip	2268419	PC ZIP Arc...	25/.. adfrw (0644)

Figura 6: Contenidos del directorio

webpay despues de descomprimir

Finalmente se deben mover los archivos al directorio cgi-bin del servidor y otorgar permisos 755.

Sitio remoto: /httpdocs/cgi-bin				
cgi-bin				
datos				
? log				
Nombre de archivo ^	Tamaño de al	Tipo de archiv	Últim	Permisos
..				
datos		Directorio	24/..	flcdmpe (0755)
log		Directorio	25/..	flcdmpe (0755)
maestros		Directorio	25/..	flcdmpe (0755)
template		Directorio	25/..	flcdmpe (0755)
test		Directorio	19/..	flcdmpe (0755)
tbk_bp_pago.cgi	1601633	cgi-archivo	25/..	adfrw (0755)
tbk_bp_resultado.cgi	1597442	cgi-archivo	25/..	adfrw (0755)
tbk_check_mac.cgi	1597439	cgi-archivo	25/..	adfrw (0755)

Figura 7: Contenidos del directorio cgi-bin del servidor

Una vez copiados los archivos se deben aplicar los permisos 755 a todo el directorio cgi-bin (asumiendo que solamente tenemos webpay en dicho directorio).

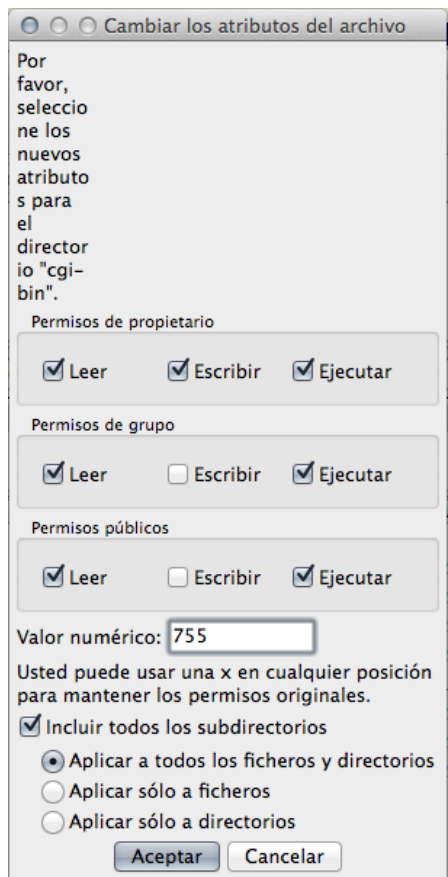


Figura 8: Permisos aplicados al directorio cgi-bin

Una vez hecho esto, solamente queda configurar webpay. Al visitar la url `tbk_bp_pago.cgi` debería aparecer algo como lo siguiente:

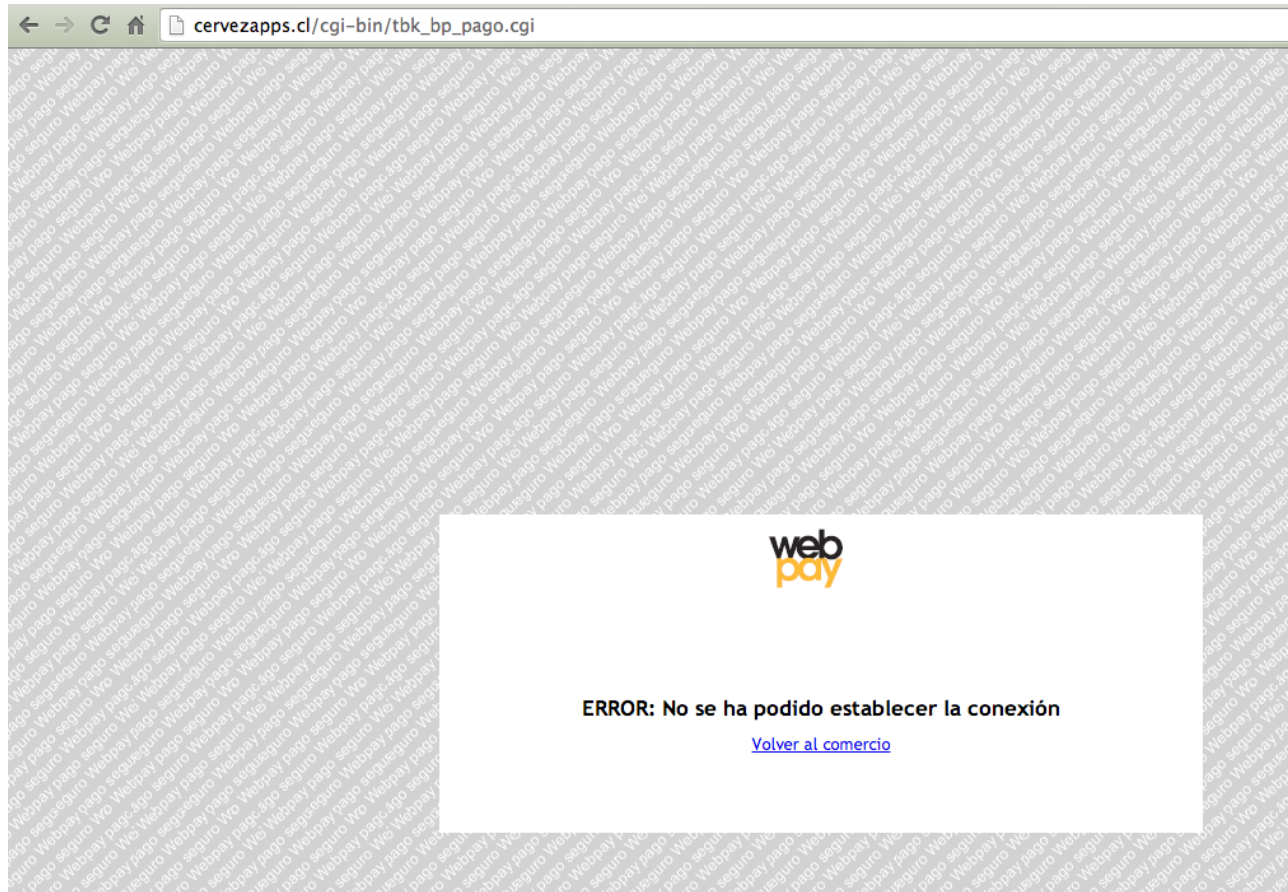


Figura 9: Ejecución de tbk_bp_pago.cgi sin configurar

Solución de errores

Si luego de realizar los pasos anteriores aparece un error 500 u otro donde debería aparecer la *figura 9* puede deberse a los permisos asociados al usuario. Todos los archivos suministrados por Transbank deben pertenecer al usuario Apache. Si el servidor tiene habilitado FastCGI, el usuario de Apache será el mismo que ha dado el hosting.

Apache debe ser propietario (owner) de todos estos archivos con permisos de lectura, escritura y ejecución (755):

```
tbk_bp_pago.cgi
tbk_bp_resultado.cgi
tbk_check_mac.cgi
datos/
datos/tbk_param.txt
datos/tbk_config.dat
datos/tbk_trace.dat
maestros/
maestros/privada.pem
maestros/tbk_public_key.pem
maestros/tbk_public_key101prod.pem
log/
```

Para cambiar los permisos se debe ejecutar el siguiente comando.

```
$ chown -R <usuario>:<grupo> .
```

donde *usuario* y *grupo* son los otorgados por el hosting, normalmente tienen el mismo valor.

Para comprobar el propietario y grupo se debe ejecutar el siguiente comando dentro del directorio de cgi-bin

```
$ ls -las
```

Otra causa posible para el error 500 es que estén corruptos los archivos de Transbank, por eso es recomendable subirlos en archivo zip y descomprimirlos en el servidor. Si se suben de forma individual puede que se corrompan. Se recomienda verificar que sean la última versión disponible para la arquitectura del servidor. Pueden ser descargados desde la [Página de Transbank](#).

Para saber la arquitectura del servidor se debe ejecutar el siguiente comando:

```
$ uname -m
```

Nota

Usar el KCC Linux 32 en Linux 64 causará error. Siempre usar el KCC correspondiente a la arquitectura del servidor.

Error 46

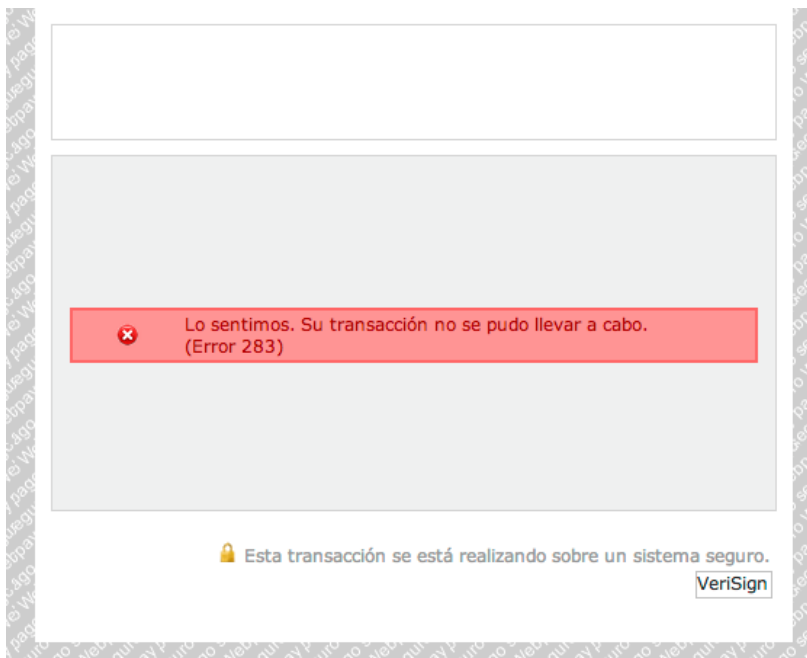
Este es un error que se produce por una mala configuración del kcc. Siempre mostrará la página de rechazo. Para solucionarlo simplemente se debe revisar que las rutas hacia el directorio cgi estén bien configuradas en el archivo `datos/tbk_config.dat`. También puede ocurrir cuando las rutas están mal configuradas en el plugin del ecommerce usado.

Error 283

En ocasiones puede ocurrir que aparezca el *Error 283*, Este se genera por que se modificó la llave *privada.pem* en el servidor del comercio. Puede también ocurrir si el contenido de los archivos pem esta corrupto. La solución mas simple es modificar el contenido de los archivos con un editor de texto plano y escribir el contenido correcto.

Nota

Los certificados incluidos en el comprimido disponible en la página oficial de Transbank están corruptos. Se deben reemplazar por certificados válidos.



Error 283

Se debe verificar que los archivos

```
maestros/privada.pem
maestros/tbk_public_key.pem
maestros/tbk_public_key101prod.pem
```

estén correctos, pertenezcan al usuario Apache y que los permisos que deben ser 755. Como último recurso variar el valor de TBK_KEY_ID entre 100 (1024 bits) o 101 (4096 (bits) según corresponda.

Los siguientes certificados fueron obtenidos desde el manual de instalación y son los que deben usarse en el ambiente de certificación.

El contenido de *privada.pem* en certificación es:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAn3HzPC1ZBzC03edUCf/XJiWj3bzJpjjTi/zB090+DDzZCaMp
14aspxQryvJhv8644E19Q+NHfxtz1cxd2wnSYKvay1gJx30Z1TOAkzUj4QMimR16
vomL1Q3T2MA21znt/PVPVU7T/JOG9R+EbiHNVKa/hUjwJEFVXLQNM97nHoLjb3v
V5yV2aVhmox7b54n6F3UVPVhCsHkbJpXpE+vnLpVmdETbNpFVrDyGxYg+mnEvyiO
BLIwEY3XTMrGxvS069groLi5Gg8C5LdaYOWjE9084T4fiWGrHhn2781R1rykunTu
77wiWpuQHMS0+YC7mhnsk8Z/ilD+aWz/vhsGhWIDAQABoIBAQCMA+Nrt4cpNKQmn
+Ne8348CGRS9ACSp6WRg6OCQX04zM71RZAminVgZgSQXE6aJR+T9rIWMeG7GwydX
aJGzEEQJZ0jv0MkUr+7mk9qiTOGkGHmGlyHnRQU8jDU59vXe3UE1315+NmwHbQht
waf9F7XLmoLK/WoVJA6tICRpC11oQrpziqN+gJdmMpZ9i8I1sMFE7+Y7xf+7S2u7
c1MRPUWqgdS9yViQVh3vZi25m5CyKRvN0B0hpNuZ7nrJymtADYSwt9wV2W1fX+MX
UuoYfxyQqVwryHhGdedU7GGAnoEdblUcDkBuAaFmsm1P8K4HQZLWP4v6pY1W2JLa
Zoaerb3BAoGBANCRev10CLB0HBU7sCs0eN9fTkIEsh30VixPSBqDnKsynJrIWovK
cs37Vb6phzdQ03ADofJvR9ck8+v6Cv0KR8IOF19wfc4ZoxkKBBeq94ZLN+YhE2PW
KiRFybqcgCtZxK53MyWgpIcT9xFtHVjlorZ8Jk51fgLZbZgamtLhderVAoGBAM00
mIiiv412vXzu4tFfkpu/G0x/D9/vAic3X9F0ky0B9NCyUMXMQGI8e3wWSGEZgh1s
Vg9KDV5EPxAmpumcdPFK2IMACaH41ac7vys3ZD8kMK0INQkuDAG4YsxMatWEp00
p1i3zwwEwknw1yJkOyozz0EcIzS9NrZJEjnBHEjAoGAQ81XdeqzvHEyg/CQd6sq
NCtubGXMZYi1C4d2Yi5kKn2YRcK4Hdi23V+TwodK+0oNWToZIQKjbVUmn0Bv3rt
EvezbD1MFUx+SfCIng0VRJIFTQmpnQYNUxdg2gpwXC/ZWfa6CNxtQABMjFy1cqXM
PJild1IYseJurgBu3mkvBTUCgYBqA/T1X2woLuis2wPIBAv5juXdh3lkB6eU8uxX
CEe2I+3t2EM781B2wajrKadWkmjLuMhN9AGV5UZ8S1P0DStUYwUywdx1/8RnmZIP
qSwHAGXV9ji0zNr7G4Em0/leriwkRM26w6fHjLx8EyxDfsohSbkqBROptcwQoEUx
MOQ5HQKbgAS4sbdd0as2MapuhKU2surEb3Kz3RCIpta4bXgTQMt9wawcZSSpvnft
zs5sehYvBFszL3MV98UC50HXMf7gykRCmPRmB9S+f+kiVRvQDHfc9nRNg2XgcotU
KAE16PQM8GihQ0C+EcXhouyud5CRJGfyurokR1H/jY3BiRAG5c+6
-----END RSA PRIVATE KEY-----
```

El contenido de *tbk_public_key.pem* en certificación es:

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAACAg8AMIICCGKCAgEAtKe3HHWwRcizAfksS92V
fQr8cUb94TRjQPzNTqBduvvj65AD5J98Cn1htE3NzOz+PjPRcnfVe53V4f3+Y1Ib
6nnxyeuYLByiwoPkPmOPFBxNp04/Yh3dxN4xg0ANXA37rNbDe04WIEMG6zbdQMNJ
7RqQU1JSmui8gt3YxtqWBhBVW79qDCYVzxFrV3SH7pRuYer+cxDvzRy1xnJgr6ee
N7gmjoSMqF16f9aGdQ12obzV0A35BqpN6pRFoS/NvICbEedS9g5gyUHF54a+juB
OV2HH5VJ3CCgcb7I7Sio/xXTyP+QjIGJfpukKE8F+ohwRiChZ9jMXofPtuzYZ1FQ
/gX08s5Qdpaph65UINP7crYbzpVJdrT2J0etyMcZbEanEkoX8YakLEBpPhyY7mC
73fWd9sTuBEK6GkzCuG2JAyo6V8eyISn1KDEVd+/6G/Zpb5cUdBCERTYz5gvNoZN
zkUq4isiXh5MOLGs91H8ermuhdQe/1qvXF8Op/EYrAuxcdrZK0orI4LbPdUrC0Jc
F102qgXRrSpXo72an01Fc9P0b1D4CMevW2+1wvIPA0DaJPsTnwBW0Uqcfa7GAFH5
KGs3zciz5YTLdlInaps8koSssTVRi7LVT8HhIC5mjBklxmZjBv6ckgQeFWgp18kuU
ve5Elj5HSV7x2PCz8RKb4XcCAwEAAQ==
-----END PUBLIC KEY-----
```

El contenido de *tbk_public_key101prod.pem* en certificación es:

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAACAg8AMIICCGKCAgEAXKjroxE7X44TQovh9A9
ZpntP7LrdoyFsnjBDKj00Coiid92FydN5qemyQCeXhsc7QHUXwGdth22fB8xJr3a
MZBEUJ+BKFRl+W6yE5V+F5Bj0Uq31L0QMAIftGhLpgqw0ZMtU89kyd9Q4Rclq4r8
p2m/ZD7Pn5EmTOFSeyowTMZQD170EoCKh/cZH5NjdlW081CI+sGL0ghRmFkve4h
```



```
F9JCwKA7NYG7j3BWh390j2NIXEY/T01Y3Y2WfNv9nvTpr46SpFlyp0K0hSiqgvXX
DgeX1ebyqS82ch2DzOV9fjDAw7t71WXJBaev8Gd6HXwIXE/JP6AnLCa2Y+b6Wv8K
GWBCMIbXWL0m7WHeCaJ9Hx2yXZmHJh8FgeKffFKCwn3X90JiMocOSGs0E+Sfo85S
h/39Vc7vZS3i7kJDDoz9ab9/vFy30RuJf4p8Erh7kWtERVog6/EhR+j4N3mgIOBZ
SHFzDAoOnqP5l7t2RXYcEbRLVN6o+XgUta1X33EJxJR5Xoz9a6PxYlesIwPbKteD
BZ/xyJDwTc2gU2YzSH8G9anKrcvITBDULSAuxQUkY0iLbkb7vSKWDYKe0do6ib03
RY/KXI63Q7bGKYaI2aa/8GnqVJ2G1U2s59NpqX0aaWjn59gsA8trA0YK0ZP4xJIh
CvLM94G4V7lxe2IHKPqLscMCAwEAAQ==
-----END PUBLIC KEY-----
```

Al ejecutar el comando `file *` dentro de maestros debe arrojar todas como tipo ASCII Text.

```
$ file *
privada.pem:          ASCII text
tbk_public_key.pem:   ASCII text
tbk_public_key1.pem:  ASCII text
```

Nota

En algunos servidores puede decir

privada.pem: PEM RSA private key

Importante

Se debe verificar que todos los archivos existan y tengan el nombre correcto

```
maestros/privada.pem
maestros/tbk_public_key.pem
maestros/tbk_public_key101prod.pem
```

Puede que el archivo `tbk_public_key1.pem` exista y el archivo `tbk_public_key101prod.pem` no. El archivo `tbk_public_key101prod.pem` siempre debe existir con el contenido expuesto en la guía más arriba.

Otras consideraciones

Si se está en ambiente de producción y todo lo demás ha fallado, la solución es generar nuevamente las llaves asimétricas y enviar la llave `publica.pem` a sosporte@transbank.cl para que sea cargada en el servidor web de Transbank.

Mod Security

Nota

Configuración no probada, intentar bajo propio riesgo.

Finalmente revisar el parámetro `mod_security`, debe estar todas las reglas desactivadas para las siguientes IPs

```
Ambiente de Certificación
200.10.12.55
```

```
Ambiente de Producción
200.10.14.162
200.10.14.163
200.10.12.162
200.10.12.163
200.10.14.34
200.10.14.177
```

Ejemplo de Configuración

```
# Acceso a certificacion 200.10.12.55
SecRule REMOTE_ADDR "^200\.10\.12\.55$" phase:1,nolog,allow,ctl:ruleEngine=Off
```

```
SecRule REMOTE_ADDR "^200\.\10\.\14\.\162$" phase:1,nolog,allow,ctl:ruleEngine=Off
SecRule REMOTE_ADDR "^200\.\10\.\14\.\163$" phase:1,nolog,allow,ctl:ruleEngine=Off
SecRule REMOTE_ADDR "^200\.\10\.\14\.\164$" phase:1,nolog,allow,ctl:ruleEngine=Off
SecRule REMOTE_ADDR "^200\.\10\.\14\.\177$" phase:1,nolog,allow,ctl:ruleEngine=Off
```

Para permitir la página de cierre

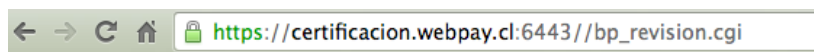
```
SecRule REQUEST_URI "index.php?route=payment/webpay_occl/callback" phase:1,nolog,allow,ctl:ruleEngine=Off
```

Nota

Si ninguna de las posibles soluciones ha funcionado se recomienda llamar a Transbank, por último revisar que todos los scripts de la tienda (PHP, Java) estén funcionando correctamente. Si nada de esto resulta probar copiar los archivos desde una instalación exitosa de Webpay o considerar cambiar de servidor.

Correcciones Previas a Configuración

El archivo *tbk_config.dat* proporcionado por Transbank contiene un carácter no visible (CTRL_M) que causa el siguiente error:



Not Found

The requested URL /bp_revision.cgi was not found on this server.

Figura 10: Error causado por carácter no visible

Para solucionarlo se deben ejecutar los siguientes comandos en una terminal:

```
$ cat tbk_config.dat | col -b > tbk_config.dat.ok
$ mv tbk_config.dat.ok tbk_config.dat
```

Adicionalmente se proporciona un archivo de configuración limpio en la carpeta examples de éste tutorial. También podría ser creado un nuevo archivo en blanco con fin de línea tipo UNIX.

Luego de ser corregido, el archivo *tbk_config.dat* debe ser subido al servidor en *cgi/datos/* reemplazando al anterior.

Instalación en Servidores sin Exec

La mayoría de los servidores tendrán habilitada la función exec de php, para el caso de los que no la tengan, existe la alternativa de realizar los pasos a continuación.

Es importante mencionar que cuando un servidor no tenga habilitada la función exec la página de cierre no será ejecutada al ser llamada por Transbank, ya que normalmente la configuración de seguridad bloqueará la ejecución del script.

Requisitos

- CGI Perl
- Acceso FTP
- No tener función exec en PHP.
- Curl (Si es que file_get_contents esta bloqueado)

Código

El siguiente código debe llamarse *chkmac.cgi* y ser almacenado en el directorio *cgi-bin* donde se encuentra el KCC. Se le deben dar permisos 755.

```
#!/usr/bin/perl
use strict;
use CGI;
my %conf;

#####
$conf{'passwd'} = 'definir una contraseña';
$conf{'log_path'} = 'log/';
#####

my $query = new CGI;
my $addr = $ENV{'REMOTE_ADDR'};
my $filename = $query->param('filename');

$filename =~ tr/[a-zA-Z0-9\.\_\-]//cd;
print "Content-type: text/html\n\n";

my $cmd = "./tbk_check_mac.cgi " . $conf{'log_path'} . $filename;

if($query->param('passwd') eq $conf{'passwd'}) {
    exec($cmd);
}
```

Para poder ejecutar el CGI dentro de PHP se debe llamar a una url del tipo local (localhost), dando los parámetros de *contraseña* y *filename* que serán utilizados.

Parámetros

passwd

Contraseña definida en *chkmac.cgi*

filename

Archivo log de la transacción. Solamente el nombre.

Ejemplo: log_20080118103954_0663580797.log

Uso

Si la función *file_get_contents* está bloqueada se debe utilizar cURL.

Algunos links de interés.

http://www.phpgang.com/replace-file_get_contentsfopen-with-curl_30.html <http://stackoverflow.com/questions/7893211/need-to-change-codes-file-get-contents-with-curl-code> http://phpcode.mypapit.net/how-to-solve-file_get_contents-and-file-php-errors/45/ http://webddr.net/tips-and-tricks/curl-replacement-for-file_get_contents/

Para ejecutar el script usando perl y *file_get_contents*

```
$result[0] = file_get_contents("http://localhost/web/cgi-bin/chkmac.cgi?passwd=contraseña&filename=$filename")
```

Éste código debe reemplazar a cualquier llamada a *exec*, como aparece en el archivo *xt_compra.php*

```
$cmdline = $docRoot . "/cgi-bin/tbk_check_mac.cgi $filename";
exec($cmdline,$result,$retint);
```

MD5

Los Hash MD5 de los archivos permiten asegurar que no haya problemas de corrupción. Se debe comprobar que sean los siguientes una vez subidos al servidor.

Se utilizo el siguiente comando en Unix dentro del directorio cgi-bin

```
find . | xargs md5
```

(En algunas versiones de Linux el programa se llama md5sum)

entregando los siguientes resultados para el KCC60 entregado por Trankbank

```
MD5 (./datos/tbk_config.dat) = 2f46a5618198cd685805bbbc498e68d
MD5 (./datos/tbk_param.txt) = 1d000f161fa0432fdea15396451f11a6
MD5 (./datos/tbk_trace.dat) = d5afbe74d8e72f80c9b6f0d1c93f3138

MD5 (./maestros/privada.pem) = aabf697f63597c9ce0e6f8d949e13af4
MD5 (./maestros/tbk_public_key.pem) = f8947e768098ecb0eca0607e2d6a2562
MD5 (./maestros/tbk_public_key101prod.pem) = 3d1fad60d7b984706f51e1f2fe2eac01

MD5 (./tbk_bp_pago.cgi) = 9c842e6d5fa09a788a4388de9f79987e
MD5 (./tbk_bp_resultado.cgi) = 397c5fe57e245a32d2ce154ddb767381
MD5 (./tbk_check_mac.cgi) = 06e14ad62efb823666ce014b8bdfac5c
```

Los archivos de los directorios de */datos* y */maestros* pueden variar su MD5 debido a las configuraciones que se deben hacer, se recomienda comprobar los contenidos de */maestros* en la etapa de instalación.

Se ha creado el siguiente script para facilitar la comparación. Se asume que los archivos estan instalados en *cgi-bin* y se debe ejecutar en el navegador. ejemplo

<http://cervezapps.cl/webpay/md5check.php>

Archivo md5check.php

```
<?php
/**
 * Calcula el MD5 y los compara
 * para asegurar de que la integridad
 * de los archivos subidos al servidor
 * sea correcta.
 */

// Hashes

$md5s['tbk_pago'] = '9c842e6d5fa09a788a4388de9f79987e';

$md5s['tbk_resultado'] = '397c5fe57e245a32d2ce154ddb767381';

$md5s['tbk_check_mac'] = '06e14ad62efb823666ce014b8bdfac5c';

$md5s['tbk_param'] = '1d000f161fa0432fdea15396451f11a6';

$md5s['tbk_config'] = '2f46a5618198cd685805bbbc498e68d';

$md5s['tbk_trace'] = 'd5afbe74d8e72f80c9b6f0d1c93f3138';

$md5s['privada'] = 'aabf697f63597c9ce0e6f8d949e13af4';

$md5s['tbk_public'] = 'f8947e768098ecb0eca0607e2d6a2562';

$md5s['tbk_public_101'] = '3d1fad60d7b984706f51e1f2fe2eac01';

// Archivos
```

```

$directory = pathinfo(realpath('.'), PATHINFO_DIRNAME) . '/cgi-bin/';

$files['tbk_pago'] = $directory . 'tbk_bp_pago.cgi';

$files['tbk_resultado'] = $directory . 'tbk_bp_resultado.cgi';

$files['tbk_check_mac'] = $directory . 'tbk_check_mac.cgi';

$files['tbk_param'] = $directory . 'datos/tbk_param.txt';

$files['tbk_config'] = $directory . 'datos/tbk_config.dat';

$files['tbk_trace'] = $directory . 'datos/tbk_trace.dat';

$files['privada'] = $directory . 'maestros/privada.pem';

$files['tbk_public'] = $directory . 'maestros/tbk_public_key.pem';

$files['tbk_public_101'] = $directory . 'maestros/tbk_public_key101prod.pem';

// Comparación

foreach($md5s as $file => $hash)
{
    if($hash === md5_file($files[$file]))
    {
        echo "<i>". $file . "</i> Hash Igual (OK)<br>";

    }
    else
    {
        echo "<i><b>". $file . "</b></i> Hash Diferente (?) <br>";

    }
}

}

/** Fin del Archivo md5check.php */

```

```

tbk_pago Hash Igual (OK)
tbk_resultado Hash Igual (OK)
tbk_check_mac Hash Igual (OK)
tbk_param Hash Igual (OK)
tbk_config Hash Diferente (?)
tbk_trace Hash Igual (OK)
privada Hash Igual (OK)
tbk_public Hash Igual (OK)
tbk_public_101 Hash Igual (OK)

```

Figura 11: Ejecución de Archivo md5check.php

Instalación Usando SSH

Si se tiene acceso a una terminal utilizando SSH se puede seguir éstos pasos. Para Windows usar [Putty](#).

Las siguientes instrucciones son para entornos Linux o Mac OS en una terminal.

Ingresa al servidor

```
$ ssh -p < puerto > < usuario >@< ip servidor >
```

Luego ir a cgi-bin

```
$ cd public_html/cgi-bin/
```

Descargar los archivos de Webpay, éstos deben estar en un servidor comprimidos en zip, con los contenidos del directorio cgi-bin del KCC brindado por Transbank.

```
$ wget http://cervezapps.cl/webpay/webpay.zip
```

Después se deben descomprimir

```
$ unzip webpay.zip
```

Nota: Los archivos oficiales de Transbank están comprimidos en archivos .rar para descomprimirlos se debe ejecutar lo siguiente:

```
$ unrar x kcc-6.0.2-linux-32.rar
```

El programa *unrar* normalmente no viene instalado de forma predeterminada, por lo que será necesario instalarlo con el sistema de gestión de paquetes del servidor, para lo cual es necesario tener permisos de administrador (root).

Luego la descompresión daría el siguiente resultado

```
Archive:  webpay.zip
creating:  datos/
inflating: datos/tbk_config.dat
inflating: datos/tbk_param.txt
inflating: datos/tbk_trace.dat
creating:  log/
creating:  maestros/
inflating: maestros/privada.pem
inflating: maestros/tbk_public_key.pem
inflating: maestros/tbk_public_key101prod.pem
inflating: tbk_bp_pago.cgi
inflating: tbk_bp_resultado.cgi
inflating: tbk_check_mac.cgi
creating:  template/
inflating: template/leeme.txt
inflating: template/reintento.html
inflating: template/transicion.html
```

Finalmente se le dan los permisos 755 con el siguiente comando

```
$ chmod 755 -R .
```

Instalación en Servidores Windows

WebPay puede funcionar en servidores Windows, sin embargo se necesita de privilegios administrativos para poder configurar correctamente la ejecución de CGI, los cuales no siempre están disponibles en servidores compartidos. Si no se tiene acceso a un servidor Windows VPS o Dedicado, se recomienda utilizar servidores Linux por mejor Costo - Oportunidad.

Comentarios Finales

Webpay permite el pago utilizando el peso chileno (CLP) o dólar estado unidense (USD). Si se necesitase la utilización de ambos se deben tener dos copias completas del kcc con distinta configuración en directorios separados.

```
cgi-bin
- clp
```

```
-- tbk_bp_pago.cgi
-- ... (demás archivos)
- usd
-- tbk_bp_pago.cgi
-- ... (demás archivos)
```

Paso 2 - Configuración de WebPay

Una vez instalados los archivos, se debe configurar webpay para su posterior utilización.

EZ Webpay Configurator

Ninjas.cl ha creado un programa para ayudar a la configuración de este archivo. pueden descargarlo gratuitamente acá <https://github.com/NinjasCL/EzWebpayConfigurator>

Archivo tbk_config.dat

Éste archivo contiene la configuración más importante para el correcto funcionamiento de WebPay, se debe abrir en un editor de texto plano como Vim o Notepad ++. Esta ubicado en el directorio *cgi-bin/datos/*.

Configuración de Fábrica

El archivo que viene con el kcc tiene la siguiente estructura predefinida.

```
IDCOMERCIO = 597026007976
MEDCOM = 1
TBK_KEY_ID = 101
PARAMVERIFCOM = 1
URLCGICOM =
SERVERCOM =
PORTCOM = 80
WHITELISTCOM = ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz 0123456789./:=&?_
HOST =
WPORT = 80
URLCGITRA = /filtroUnificado/bp_revision.cgi
URLCGIMEDTRA = /filtroUnificado/bp_validacion.cgi
SERVERTRA = https://certificacion.webpay.cl
PORTTRA = 6443
PREFIJO_CONF_TR = HTML_
HTML_TR_NORMAL =
```

Campo	Valor	Comentario
IDCOMERCIO	597026007976	Normal Pesos Ambiente certificación. Ver Anexo A de Manual Integración KCC60. Debe modificarse antes de iniciar el proceso de certificación.
MEDCOM	1 ó 2	1 = Usar SSL, 2 = Usar Redirección. Recomendado 2. El medio de conexión indica a KCC la forma en la que se comunicar con Transbank. Éste parámetro abre un canal SSL (canal de comunicación seguro) entre el comercio y Transbank, por el cual se envían los datos encriptados y en segundo plano generando un token identificador de la transacción que se le envía al tarjeta habiente para su redirección al formulario de pago Webpay.
TBK_KEY_ID	101	Llave pública asociada al comercio, solo modificable si Transbank lo solicita. Puede tener un valor de 100 o 101.

PARAMVERIFCOM	1 ó 0	1 = Habilitar filtro y revisión de parámetros, 0 = sin filtro y revisión de parámetros. Recomendado 1.
URLCGICOM	-	Ruta absoluta donde estará el archivo tbk_bp_resultado.cgi en el servidor del comercio. No puede ser https. Ejemplo http://www.ejemplo.com/cgi-bin/tbk_bp_resultado.cgi
SERVERCOM	-	Número IP del servidor del comercio. Para obtenerlo utilizar el comando ping. Ejemplo ping www.google.cl (173.194.42.223)
PORTCOM	80	Puerto donde esta el servidor web del comercio que aloja el archivo tbk_bp_resultado.cgi . Normalmente 80.
WHITELISTCOM	(ABC...?_)	Listado de caracteres permitidos al utilizar GET o POST
HOST	-	Número IP del servidor del comercio donde están alojadas las páginas de confirmación de las transacciones. Normalmente es el mismo valor que SERVERCOM.
WPORT	80	Puerto donde esta el servidor web del comercio que aloja las páginas de confirmación de las transacciones. Normalmente es el mismo valor que PORTCOM.
URLCGITRA	/filtroUnificado/bp_revision.cgi	Ruta Relativa a bp_revision.cgi alojado en Transbank. Para producción su valor debe ser /cgi/bp_revision.cgi
URLCGIMEDTRA	/filtroUnificado/bp_validacion.cgi	Ruta relativa a bp_validacion.cgi alojado en Transbank. Para producción su valor debe ser /cgi/bp_validacion.cgi
SERVERTRA	https://certificacion.webpay.cl	Dirección del servidor de Transbank. Incluir https. Dirección de produccion https://webpay.transbank.cl
PORTTRA	6443	Puerto del servidor de Transbank. Puerto de producción 443
PREFIJO_CONF_TR	HTML_	Prefijo que contiene el nombre del parámetro que indica cual es la página que se ejecuta en el comercio para confirmar la transacción. Valor recomendado HTML_
HTML_TR_NORMAL	-	Ruta absoluta donde esta alojada la página de confirmación de la transacción en el servidor del comercio. No puede tener https.

A continuación se muestra un ejemplo de configuración

```

IDCOMERCIO = 597026007976
MEDCOM = 2
TBK_KEY_ID = 101
PARAMVERIFCOM = 1
URLCGICOM = http://cervezapps.cl/cgi-bin/tbk_bp_resultado.cgi
SERVERCOM = 85.25.108.124
PORTCOM = 80
WHITELISTCOM = ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz 0123456789./:=&?_
HOST = 85.25.108.124
WPORT = 80
URLCGITRA = /filtroUnificado/bp_revision.cgi
URLCGIMEDTRA = /filtroUnificado/bp_validacion.cgi
SERVERTRA = https://certificacion.webpay.cl
PORTTRA = 6443
PREFIJO_CONF_TR = HTML_
HTML_TR_NORMAL = http://cervezapps.cl/webpay/cierre.php

```


Nota

WHITELISTCOM no contiene el caracter gui3n "-", se recomienda agregarlo si ocurren problemas de conexi3n.

Quedando

WHITELISTCOM = ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz 0123456789./:=&?_ -

Nota

El parámetro MEDCOM = 1, requiere de SSL para su funcionamiento, lo que implica que el comercio pueda establecer una comunicaci3n directa con las IPs de Transbank al puerto 443.

Una vez configurado y ejecutando los scripts del comercio se debera ver lo siguiente:



Figura 12: Inicio Sistema Webpay post configuraci3n

Nota: Existe una variable de configuraci3n llamada *TBK_REFERERER*, 3sta no debe ser incluida en *tbk_config.dat* o causará un error de conexi3n.

Pasos Siguientes

Ahora Webpay esta configurado para realizar las pruebas. 3stas son algunas de las cosas que quedarían pendientes:

- Crear script propios (opcional).
- Configurar sistema de carro de compra con WebPay.
- Preparar el ambiente para comenzar certificaci3n.

Lista de Servidores

A continuaci3n se mostrará un listado de servidores que se han probado para instalar el KCC. Se recomienda un VPS para tener control total del sistema, de preferencia hospedado en servidores extranjeros, ya que los proveedores de hosting nacionales tienden a tener poco ancho de banda internacional y un centro de ayuda al cliente bastante mejorable. Ninguno de estos servidores estan relacionados con el autor, simplemente son un listado donde se ha comprobado el correcto funcionamiento del KCC.

Actualizado al 30 de noviembre del 2014

La siguiente es un listado donde se ha instalado exitosamente el KCC. Se debe instalar el KCC correspondiente a la arquitectura del servidor.

Hosting	Características	Permite Exec	Comentario
Centroclic	Plan Básico Linux 64 Shared (CLP \$6.000)	No	Hosting usado para la creación de ésta guía.
HostGator	Hatchling Plan (USD \$3.96)	Sí	Plan Básico de pago mensual
HostGator	VPS Level 3 Linux 64 (USD \$49.95)	Sí	Plan de coste mensual, acceso total sin restricciones a la configuración del servidor.
SilverHost	Plan Básico (CLP \$18.000)	Sí	Plan Básico de pago anual

Nota

Existen proveedores gratuitos como [Heroku](#) y [OpenShift](#) que permitirían configurar una tienda con webpay en teoría. Pero no se han realizado pruebas para comprobar su validez.

Referencias

[Configurar Webpay en PHP](#)

[Tutorial WebPay](#)

[Mod Security y WebPay](#)

[Cómo configurar Webpay](#)

[WebPay en Tecnoera](#)

[KCC Transbank](#)

[WebPay Magento](#)

[Woocommerce Webpay](#)

[Página de Transbank](#)

[Transbankiano](#)

[Manual Integracion KCC60 - Transbank](#)

Proceso de Certificación

Una vez configurado el KCC y esté funcionando con las claves de prueba se debe comenzar el proceso de certificación. Para ésto se debe verificar que los datos detallados en el Manual de Integración KCC 60 Anexo C (p. 54-55) se cumplan a cabalidad. El comercio debe tener la siguiente información:

- Página que explique los términos y condiciones del servicio.
- Página que explique la política de devoluciones.
- Página que explique la forma de entrega.

Paso 1 - Documento de Pre - Certificación

Transbank ha complicado el proceso de certificación exigiendo a los comercios enviar capturas de pantallas de diversas

pruebas en un solo archivo pdf. Es necesario llenar un documento llamado *Evidencia Comercio.docx*, el cual está dentro del archivo descargable desde [Transbank Descargas](#). En éste caso dentro del archivo llamado *linux_64_6.0.2.zip*. Dentro del documento se solicitan diferentes datos y responder una serie de preguntas. Una vez llenado se debe exportar en formato *pdf* y ser enviado a sopORTE@transbank.cl con una copia al contacto comercial. Finalmente se deben adjuntar los archivos generados en el proceso de compra llamados *tbk_bitacora* y *tbk_evn* correspondientes a la fecha donde se solicita la certificación.

Paso 2 - Llenar Documento Evidencia Comercio

El contenido del documento solicita la información mostrada a continuación.

Información de Comercio

- Nombre Comercio
- Url Comercio
- Rubro Comercio
- Rut Comercio
- Código Comercio

Información Contacto Comercial del Comercio

- Nombre
- Mail
- Teléfono

Información de Contacto Técnico del Comercio

- Nombre
- Mail
- Teléfono

Información de Cuenta de Prueba

- Usuario
- Contraseña
- URL de Login

Adicionalmente se deben responder una serie de preguntas

- Flujo Pago Exitoso Sin Cuotas
- Flujo Pago Exitoso 3 Cuotas Sin Interés
- Flujo de Pago Exitoso 4 Cuotas (normales, con interés)
- Flujo de Pago Exitoso con Tarjeta de Débito RedCompra
- Flujo de Pago Fracaso con tarjeta de crédito MasterCard
- Validación de Orden de Compra
- Validación de Página de Cierre

Para facilitar la entrega de datos se ha creado una plantilla de ejemplo disponible en [examples/Evidencia Comercio](#) en formato pdf, docx y pages. La cual contiene más detalles sobre como llenar la información y responder las preguntas.

También se ha creado un ejemplo de formulario html para realizar la "Evidencia 6: Validación de Orden de Compra". Puede ser [descargado de este link](#).

Una vez que soporte pruebe los datos enviarán al sitio a certificación, proceso que tardará entre 24 a 48 horas.

Envió a Certificación

Una vez que soporte ha aprobado el envío a certificación, Luego de 24 a 48 horas certificación enviará un correo solicitando los archivos listados anteriormente. Se deben enviar de la misma forma a certificacion@transbank.cl.

Paso 3 - Paso a Producción

Certificación tardará unos días en realizar las pruebas. Una vez finalizadas enviará un correo con las instrucciones para generar las claves PEM necesarias para pasar a producción el sitio. Lamentablemente Transbank solo da instrucciones para generarlo utilizando Windows.

Nota

El paso a producción no debe tomar más de 10 días o certificación anulará el proceso y deberá comenzar desde cero.

Nota2

Para ahorrar tiempo es recomendable generar los archivos pem antes de enviar el zip a certificación. De esta forma podrán descargarlo fácilmente más tarde.

Generación de llaves

Windows

El generador de llaves se debe descargar desde <https://www.transbank.cl/public/documentacion/descarga-kits/generador-llaves.rar>

y tendrá los siguientes archivos:

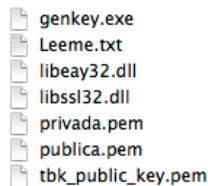


Figura 13: Contenidos de generador-llaves.rar

El archivo *Leeme.txt* contiene instrucciones sobre la generación de las claves y los correos donde se debe enviar posteriormente *publica.pem*. También contiene el archivo *tbk_public_key.pem* que debemos usarlo para pasar a producción el sitio.

Para generar las llaves debemos abrir el archivo *genkey.exe* e introducir el código de comercio asociado a la tienda.

Nota

El generador de llaves dice KCC 5.0 pero servirá de todas formas para el KCC 6.0

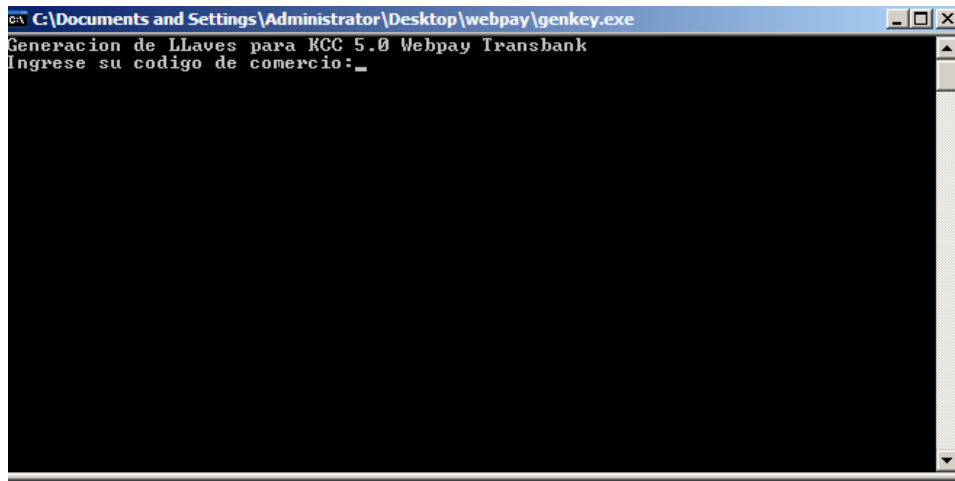


Figura 14. genkey.exe

Una vez ejecutado generará dos archivos

- privada.pem
- publica.pem

El archivo *privada.pem* contiene la llave privada asociada al comercio, ésta debe permanecer oculta al público. El archivo *publica.pem* debe ser entregado a certificación.

Una vez generadas las llaves se debe enviar el archivo *publica.pem* junto al código del comercio a los correos indicados en el archivo Leeme.txt o en su defecto certificacion@transbank.cl.

Linux

Según el manual de Transbank para generar las claves pem en Linux se deben utilizar los siguientes comandos:

Nota

No se ha comprobado su funcionamiento

```
# openssl genrsa -out privada.pem 4096
# openssl rsa -in privada.pem -pubout -out publica.pem
```

También se puede ejecutar el programa *genkey.exe* usando la aplicación *wine*.

```
$ sudo aptitude install wine
$ wine genkey.exe
```

La versión probada es la wine-1.4.1 bajo un sistema debian wheezy.

Mac

Para usar el programa *genkey.exe* en Mac. Hay distintas alternativas, desde una máquina virtual con Windows. usar Bootcamp o Parallels. Pero la opción más simple sería usando *wine*. La siguiente es una guía en inglés que explica cómo instalar *wine* en Mac. <http://www.davidbaumgold.com/tutorials/wine-mac/>.

Configuración Final

Finalmente se debe reconfigurar el sitio para utilizar las llaves generadas anteriormente.

Se debe reemplazar el archivo *privada.pem* por el generado con el programa. También se debe reemplazar el archivo *tbk_public_key.pem* por el que se encuentra dentro del comprimido generador-llaves.rar. Los archivos a reemplazar son:

- /cgi-bin/maestros/tbk_public_key.pem
- /cgi-bin/maestros/privada.pem

Nota

Todos los archivos deben ser subidos como ASCII. En linux verificar que los archivos terminen con el retorno de carro tipo UNIX.

Ahora se debe modificar el archivo /cgi-bin/datos/tbk_config.dat con los datos usados en producción, conservando los valores de los campos que no sean especificados en la siguiente tabla:

Campo	Valor	Comentario
IDCOMERCIO	5970xxxx	El Código de Comercio que Transbank dio.
SERVERTRA	https://webpay.transbank.cl	Servidor de Producción
PORTTRA	443	Puerto de Producción
URLCGITRA	/cgi-bin/bp_revision.cgi	URL del programa de revisión en Producción.
URLCGIMEDTRA	/cgi-bin/bp_validacion.cgi	URL del programa de validación en Producción

Quedando de la siguiente forma:

```
IDCOMERCIO = 597073392290
MEDCOM = 2
TBK_KEY_ID = 101
PARAMVERIFCOM = 1
URLCGICOM = http://cervezapps.cl/cgi-bin/tbk_bp_resultado.cgi
SERVERCOM = 85.25.108.124
PORTCOM = 80
WHITELISTCOM = ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz 0123456789./:=&?_
HOST = 85.25.108.124
WPORT = 80
URLCGITRA = /cgi-bin/bp_revision.cgi
URLCGIMEDTRA = /cgi-bin/bp_validacion.cgi
SERVERTRA = https://webpay.transbank.cl
PORTTRA = 443
PREFIXO_CONF_TR = HTML_
HTML_TR_NORMAL = http://cervezapps.cl/webpay/cierre.php
```

Nota

MEDCOM = 1 puede ser usado si el comercio cuenta con SSL.

Nota 2

Se debe anteponer el número 5970 como prefijo al número de comercio en IDCOMERCIO

Paso 4 - Prueba Final

Una vez configurado el sitio en producción se hará una prueba final para comprobar que todo este funcionando correctamente, una vez que Transbank informe que el comercio ha sido agregado se debe crear una compra.

La prueba consiste en crear un producto de valor \$10 CLP (10 pesos chilenos) y ejecutar una venta con un banco real.

Se debe enviar a transbank una prueba de que el comercio efectua las ventas correctamente usando este producto de prueba.

Paso 5 - Limpieza, Seguridad y Otros

Limpieza

Una vez completado el proceso de paso a producción se recomienda limpiar todos los registros de las compras realizadas en el proceso de pruebas. Usuarios, carros de compra, estadísticas, etc.

Seguridad

Se recomienda asegurar que solamente las siguientes IPs de Transbank puedan ejecutar los archivos en `/cgi-bin/`:

- 200.10.14.162
- 200.10.14.163
- 200.10.12.162
- 200.10.12.163
- 200.10.14.34
- 200.10.14.177

Logo de Comercio

Se puede enviar a Transbank un logotipo del comercio para mostrarse en el formulario de pago, ésto es totalmente opcional.

El logo debe tener las siguientes características:

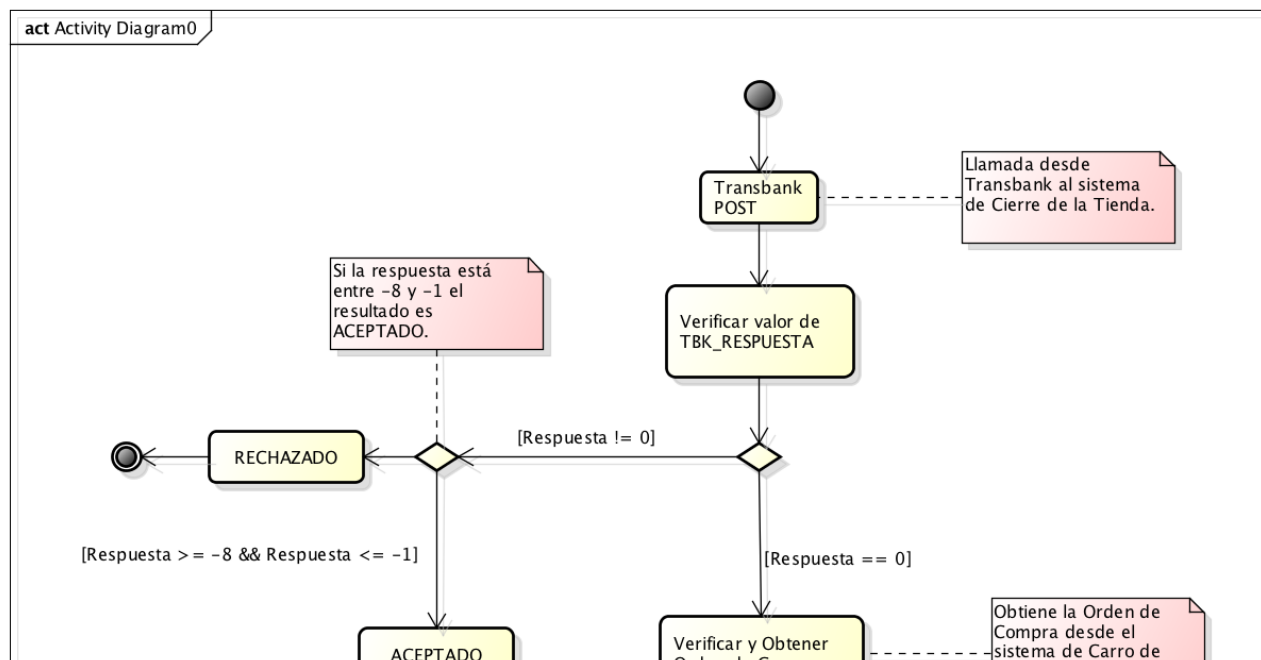
Formato: GIF **Dimensiones:** 130 x 59 px.

Anexos

Anexo A: Proceso de Compra

El proceso de compra está ejemplificado con el siguiente diagrama de actividades. Este proceso es iniciado luego de que la persona ha comprado un artículo e ingresado sus datos bancarios. Es cuando Transbank informa a la tienda si una orden de compra fue exitosamente pagada o no.

[Ver en Tamaño Completo](#)



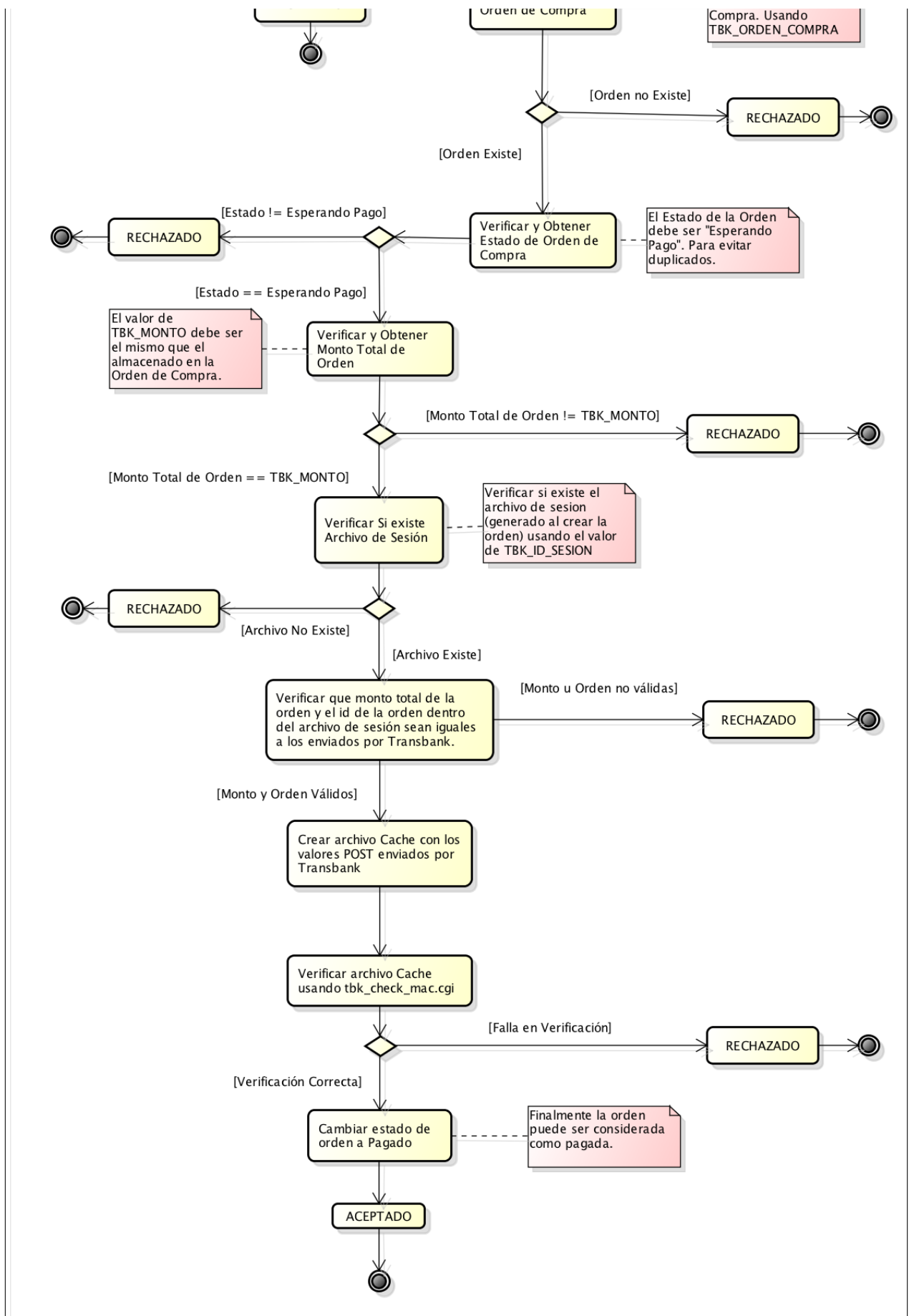


Figura 15: Diagrama de Actividades del Proceso de Compra. Fuente: Elaboración Propia.

Nota :

El paso de verificación de monto se debe agregar dos ceros (00) al monto original del carro. Ya que Transbank entrega el monto con dos ceros adicionales.

Ejemplo

```
$total_order_amount = $cart->getOrderTotalAmount();

$total_order_amount_formatted = $total_order_amount . '00';

if ($total_order_amount_formatted == $tbk_total_amount) {
    log("Amounts are Equal");
}
```

Anexo B: Orden de los parámetros de Validación de MAC en Transacción Tipo Mall

La transacción tipo Mall recibe parámetros con diferentes nombres que la transacción normal, para la validación del MAC de tipo Mall se requiere que el archivo temporal tenga escrito los parámetros en el siguiente orden:

- 1) TBK_TIPO_TRANSACCION
- 2) TBK_RESPUESTA
- 3) TBK_MONTO
- 4) TBK_ORDEN_COMPRA
- 6) TBK_FINAL_NUMERO_TARJETA
- 7) TBK_FECHA_CONTABLE
- 8) TBK_FECHA_TRANSACCION
- 9) TBK_HORA_TRANSACCION
- 10) TBK_ID_SESION
- 11) TBK_ID_TRANSACCION
- 12) TBK_COD_AUT_M001
- 13) TBK_TIPO_PAGO_M001
- 14) TBK_NUMERO_CUOTAS_M001
- 15) TBK_COD_RESP_M001
- 16) TBK_MONTO_TIENDA_M001
- 17) TBK_ORDEN_TIENDA_M001
- 18) TBK_VCI_M001
- 19) TBK_VCI
- 20) TBK_MAC

Ejemplo: TBK_TIPO_TRANSACCION=TR_MALL&TBK_RESPUESTA=0...

Anexo C: Cuentas para Ejecutar Pruebas

Al hacer pruebas siempre se deben utilizar alguna de las cuentas establecidas en el manual. A continuación se copian los datos.

Para autenticar, el usuario (RUT) a ingresar es **11.111.111-1** (nueve veces el número 1) y la clave es **123**. Una vez autenticados, dentro del emisor podrán aceptar ó rechazar la transacción.

Para pruebas con Tarjetas de Crédito

*	VISA	MASTERCARD
Número de Tarjeta	4051885600446623	5186059559590568

Año Expiración	Cualquiera	Cualquiera
Mes Expiración	Cualquiera	Cualquiera
CVV	123	123
Resultado	APROBADO	RECHAZADO

Para pruebas con RedCompra

Número de Tarjeta	Resultado
4051885600446623	APROBADO
5186059559590568	RECHAZADO

Anexo D: Amazon EC2

Webpay puede tener algunas consideraciones adicionales para funcionar bien en servidores hospedados en Amazon.

Permisos de la VPC

Primero hay que autorizar explícitamente las IPs de Transbank en la VPC. Con el fin de evitar cualquier bloqueo de las llamadas realizadas por Transbank.

Ambiente de Certificación
200.10.12.55

Ambiente de Producción
200.10.14.162
200.10.14.163
200.10.12.162
200.10.12.163
200.10.14.34
200.10.14.177

Configuración de la IP Pública

La IP pública del servidor, la cual es usada en el archivo *tbk_config.dat* debe poder ser usada para llamar directamente a los archivos del cgi-bin.

Ejemplo

La llamada a la dirección http://54.xxx.yyy.zzz/cgi-bin/tbk_bp_pago.cgi debe mostrar un logo de Webpay y no un error de servidor como 500 o 400.

Esto se debe a que los servidores de Transbank toman la IP Pública y la llaman directamente para realizar la validación con *tbk_bp_resultado.cgi*

Archivo *tbk_config.dat*

Si Los contenidos del archivo son los siguientes

```
(...)  
URLCGICOM = http://www.ejemplo.com/cgi-bin/tbk_bp_resultado.cgi  
SERVERCOM = 54.xxx.yyy.zzz  
(...)
```

Se debe verificar que se pueda ejecutar ambas direcciones bajo la url y la ip. http://www.ejemplo.com/cgi-bin/tbk_bp_resultado.cgi y http://54.xxx.yyy.zzz/cgi-bin/tbk_bp_resultado.cgi y ambas den un resultado correcto.

Anexo E: Solución a comportamiento errático

Cuando webpay lleva funcionando un tiempo considerable en el sitio (algunos años), puede que se presenten comportamientos erráticos en el sistema. Carros de compra no pagados, procesos no concluidos y otras eventualidades problemáticas. Si bien las causas pueden ser diversas y ajenas al KCC, una posibilidad es que el directorio *cgi-bin/log* este saturado. Cuando existen demasiados archivos dentro del directorio *log* el KCC puede presentar comportamientos no deseados. La solución es mover estos archivos a un directorio nuevo y dejar limpio el directorio *log* para los nuevos logs.

Para esto se deben mover los archivos a un nuevo directorio

```
$ cd cgi-bin/log
$ mkdir backup
$ mv *.* backup
```

Anexo D: Consideraciones al Pasar a Producción

Se debe asegurar de que el código de comercio sea el adecuado o se puede encontrar con problemas no deseados al momento de realizar compras. Como no hay forma de saber si el código de comercio está habilitado para webpay plus es mejor preguntar a Transbank directamente antes de enviar los certificados de producción.

Al configurar mal el código de comercio puede aparecer un error similar a

"No se puede acceder a la dirección <https://webpay2g.transbank.cl>"

