



PAPI-1.5

PAPI bajo Apache 2.0

Daniel García
daniel.garcia@rediris.es

- 1.** Qué hay de nuevo en PAPI-1.5
- 2.** Por qué portar PAPI para Apache 2.0
- 3.** mod_perl 1.0 vs mod-perl 2.0
- 4.** PAPI para Apache 1.0 & Apache 2.0
- 5.** Portando PAPI hacia Apache 2.0
- 6.** Porte del AuthServer
- 7.** Porte del Point of Access
- 8.** Nuevos módulos de criptografía
- 9.** Nuevo proceso de instalación
- 10.** Nuevo procedimiento de configuración
- 11.** Conclusiones

- 1. Qué hay de nuevo en PAPI-1.5**
- 2.** Por qué portar PAPI para Apache 2.0
- 3.** mod_perl 1.0 vs mod-perl 2.0
- 4.** PAPI para Apache 1.0 & Apache 2.0
- 5.** Portando PAPI hacia Apache 2.0
- 6.** Porte del AuthServer
- 7.** Porte del Point of Access
- 8.** Nuevos módulos de criptografía
- 9.** Nuevo proceso de instalación
- 10.** Nuevo procedimiento de configuración
- 11.** Conclusiones

- **Basado en mod_perl 2.0**
 - Por los tanto se ejecuta bajo Apache 2.0
- **Sistema de instalación**
 - Las dependencias se descargan desde CPAN
 - Las directas y las indirectas
- **Proceso de configuración**
 - Basado en XML
 - Configuración independiente
 - Directivas PAPI en XML
 - Contenedores de Apache en httpd.conf
 - Configuración mixta
 - Directivas PAPI y Apache en XML

1. Qué hay de nuevo en PAPI-1.5
2. **Por qué portar PAPI para Apache 2.0**
3. mod_perl 1.0 vs mod-perl 2.0
4. PAPI para Apache 1.0 & Apache 2.0
5. Portando PAPI hacia Apache 2.0
6. Porte del AuthServer
7. Porte del Point of Access
8. Nuevos módulos de criptografía
9. Nuevo proceso de instalación
10. Nuevo procedimiento de configuración
11. Conclusiones

- Pocas implantaciones nuevas de Apache 1.0
- Tendencia a migrar hacia Apache 2.0
- PAPI debe evolucionar con el ‘mercado’

1. Qué hay de nuevo en PAPI-1.5
2. Por qué portar PAPI para Apache 2.0
3. **mod_perl 1.0 vs mod-perl 2.0**
4. PAPI para Apache 1.0 & Apache 2.0
5. Portando PAPI hacia Apache 2.0
6. Porte del AuthServer
7. Porte del Point of Access
8. Nuevos módulos de criptografía
9. Nuevo proceso de instalación
10. Nuevo procedimiento de configuración
11. Conclusiones

- **mod_perl 1.0 API para Apache 1.0**
 - Apache 1.0 sólo soporta código mod_perl 1.0
 - No se **debe** ejecutar código mod_perl 1.0 en Apache 2.0
- **mod_perl 2.0 API para Apache 2.0**
 - Sólo compatible con Apache 2.0
 - Permite compatibilidad hacia atrás
 - Ejecutar código mod_perl 1.0 en Apache 2.0
 - Perdida importante de rendimiento
 - Mejora el rendimiento de mod_perl 1.0
 - Perdiendo compatibilidad hacia atrás

1. Qué hay de nuevo en PAPI-1.5
2. Por qué portar PAPI para Apache 2.0
3. mod_perl 1.0 vs mod-perl 2.0
4. **PAPI para Apache 1.0 & Apache 2.0**
5. Portando PAPI hacia Apache 2.0
6. Porte del AuthServer
7. Porte del Point of Access
8. Nuevos módulos de criptografía
9. Nuevo proceso de instalación
10. Nuevo procedimiento de configuración
11. Conclusiones

- **Barajamos la posibilidad**
 - Ventajas
 - Mantener una rama única de PAPI
 - Comodidad para el usuario final
 - Inconvenientes
 - Disminuye la legibilidad del código
 - Requiere Perl 5.6.0 ó superior
 - Duplica partes de código
 - Perdida de rendimiento
 - Hace muy complicado el proceso de desarrollo
- **Decidimos portar sólo para Apache 2.0**

1. Qué hay de nuevo en PAPI-1.5
2. Por qué portar PAPI para Apache 2.0
3. mod_perl 1.0 vs mod-perl 2.0
4. PAPI para Apache 1.0 & Apache 2.0
5. **Portando PAPI hacia Apache 2.0**
6. Porte del AuthServer
7. Porte del Point of Access
8. Nuevos módulos de criptografía
9. Nuevo proceso de instalación
10. Nuevo procedimiento de configuración
11. Conclusiones

- **Lo primero conocer mod_perl 2.0**
 - Lo descargamos
 - Lo compilamos
 - Lo usamos y... ¡no funciona!
- **Comprobamos la documentación de mod_perl 2.0**
- **La primera en la frente**
 - Métodos movidos a paquetes diferentes
 - Métodos que se invocan de forma diferente
 - Cambio de nombre
 - Cambio en el prototipo
 - Métodos que se han eliminado
 - Su funcionalidad se proporciona de otra forma
 - Ya sea para mejorar el rendimiento
 - Ya sea para mejora la seguridad

- **mod_perl 2.0 ofrece varias soluciones**
- **Capa de compatibilidad hacia atrás**
 - **Apache2::compat** se encarga de
 - Cargar todos los módulos **Apache2**
 - Ajusta las llamadas a metodos cuyos prototipos han cambiado
 - Proporcionar una implementación para los métodos eliminados
 - A cambio de
 - Perdida **importante** de rendimiento
 - Perdida de control sobre el código
 - Es **Apache2::compat** quien nos controla
 - Obviamente no la hemos utilizado
- **Paquetes que facilitan el proceso de porte**
 - concretamente **ModPerl::MethodLookup**

- Una vez intuimos las dificultades del porte
- Separamos el proceso en dos partes
 - Porte de **AS** (Authentication Server)
 - Porte del **PoA** (Point of Access)

1. Qué hay de nuevo en PAPI-1.5
2. Por qué portar PAPI para Apache 2.0
3. mod_perl 1.0 vs mod-perl 2.0
4. PAPI para Apache 1.0 & Apache 2.0
5. Portando PAPI hacia Apache 2.0
6. **Portando el AuthServer**
7. Porte del Point of Access
8. Nuevos módulos de criptografía
9. Nuevo proceso de instalación
10. Nuevo procedimiento de configuración
11. Conclusiones

- El AuthServer es un script Perl grande
- No hace uso de funcionalidad de mod_perl
- Sólo se han modificado las funciones criptográficas
 - Ya que se han cambiado los módulos criptograficos basados en C por
 - Crypt::OpenSSL::RSA
 - Crypt::CBC
 - Crypt::Rijndael

1. Qué hay de nuevo en PAPI-1.5
2. Por qué portar PAPI para Apache 2.0
3. mod_perl 1.0 vs mod-perl 2.0
4. PAPI para Apache 1.0 & Apache 2.0
5. Portando PAPI hacia Apache 2.0
6. Porte del AuthServer
7. **Porte del Point of Access**
8. Nuevos módulos de criptografía
9. Nuevo proceso de instalación
10. Nuevo procedimiento de configuración
11. Conclusiones

- Portar directivas de configuración

- PerlHandler → PerlResponseHandler

- PerlScript → PerlRequire

- PerlSendHeader
- PerlSetupEnv

 → PerlOptions [+SetupEnv | +ParseHeader]
 [-SetupEnv | -ParseHeader]

- Nos obliga a cambia la configuración de PAPI

- Aunque fácil de resolver... sustituir unas por otras

- **Detectar código incompatible con mod_perl 2.0**
 - **¿Qué módulos mod_perl 2.0 cargar?**
 - **ModPerl::MethodLookup**
 - **Métodos movidos a un nuevo paquete**
 - Indica a qué paquete hay que cargar
 - **Métodos que están en más de un paquete (tipo de datos)**
 - Indica a que paquetes pertenece
 - Pasando el objeto que lo invoca determina el paquete
 - Otra opción... pre-cargar **¡todos!** los módulos mod_perl 2.0
 - Ni nos la planteamos
 - **¿Y los módulos eliminados?**
 - Ningun paquete de mod-perl 2.0 proporciona ayuda
 - Solo nos queda la documentación de compatibilización hacia atrás
 - A Reference to mod_perl 1.0 to mod_perl 2.0 Migration
 - <http://perl.apache.org/docs/2.0/user/porting/compat.html>

- **Aun con la ayuda de ModPerl::MethodLookup**
 - Si una rama de código no se ejecuta **➔ no se detecta!**
- **El mejor método ha sido ‘prueba y error’**
 - Obliga a buscar en la documentación de compatibilidad
 - Ayuda a entender mejor los cambios de mod_perl 2.0
 - Pero sigue sin ser ‘completo’
 - Entendiendo por completo que detecte todo lo que ha de ser modificado
- **Por esto buscamos beta-testers**
 - Que prueben ‘cosas’ que no se nos han ocurrido

1. Qué hay de nuevo en PAPI-1.5
2. Por qué portar PAPI para Apache 2.0
3. mod_perl 1.0 vs mod-perl 2.0
4. PAPI para Apache 1.0 & Apache 2.0
5. Portando PAPI hacia Apache 2.0
6. Porte del AuthServer
7. Porte del Point of Access
8. **Nuevos módulos de criptografía**
9. Nuevo proceso de instalación
10. Nuevo procedimiento de configuración
11. Conclusiones

- **Se han sustituido los módulos criptográficos**
 - Implementados con C
 - Por módulos implementados completamente en Perl
- **Simplificamos así el proceso de instalación**
 - Al menos en principio debería ser más cómodo
 - Aunque sacrificamos algo de rendimiento
- **Módulos criptográficos usados**
 - **Crypt::OpenSSL::RSA** Encoding and decoding, using the openssl libraries
 - **Crypt::CBC** Encrypt Data with Cipher Block Chaining Mode
 - **Crypt::Rijndael** Crypt::CBC compliant Rijndael encryption module

1. Qué hay de nuevo en PAPI-1.5
2. Por qué portar PAPI para Apache 2.0
3. mod_perl 1.0 vs mod-perl 2.0
4. PAPI para Apache 1.0 & Apache 2.0
5. Portando PAPI hacia Apache 2.0
6. Porte del AuthServer
7. Porte del Point of Access
8. Nuevos módulos de criptografía
9. **Nuevo proceso de instalación**
10. Nuevo procedimiento de configuración
11. Conclusiones

- PAPI-1.5 incrementa el número de módulos de los que depende
- Se complica la tarea de resolver las dependencias
 - Tanto las directas que están controladas
 - como las indirectas, que no están tan controladas
- Por ello hemos modificado el proceso de configuración
 - Para que se intente descargar todos los módulos automáticamente
 - Desde el repositorio CPAN
- Esto se implementa en el Makefile.PL
 - <https://forja.rediris.es/websvn/wsvn/papi-perl/Makefile.PL?op=file&rev=0&sc=0>

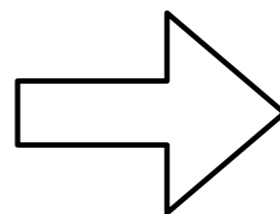
1. Qué hay de nuevo en PAPI-1.5
2. Por qué portar PAPI para Apache 2.0
3. mod_perl 1.0 vs mod-perl 2.0
4. PAPI para Apache 1.0 & Apache 2.0
5. Portando PAPI hacia Apache 2.0
6. Porte del AuthServer
7. Porte del Point of Access
8. Nuevos módulos de criptografía
9. Nuevo proceso de instalación
10. **Nuevo procedimiento de configuración**
11. Conclusiones

- La configuración se encuentra en un fichero XML
 - Tanto las directivas de PAPI como las de Apache
- Configuramos PAPI-1.5 fuera de Apache
 - También podemos definir los 'Locations' en Apache
 - Y las directivas PAPI en el fichero XML
- El proceso de configuración se inicia con Apache
 - Al iniciarse el servidor web
 - Mediante Perl Sections
 - Se configura Apache usando Perl
 - <http://perl.apache.org/docs/2.0/api/Apache/PerlSections.html>

- Con Perl Sections se evita crear directivas de configuración Apache que utilizará PAPI
 - Permite hacer cosas como

PERL CODE dentro de HTTPD.CONF

```
<Perl>
$Location{"/~doug/"} = {
  AuthUserFile => '/tmp/htpasswd',
  AuthType     => 'Basic',
  AuthName     => 'test',
  DirectoryIndex => [qw(index.html)],
};
</Perl>
```



Apache Configuration File

```
<Location /~doug/>
  AuthUserFile    /tmp/htpasswd
  AuthType        Basic
  AuthName        test
  DirectoryIndex  index.html
</Locations>
```

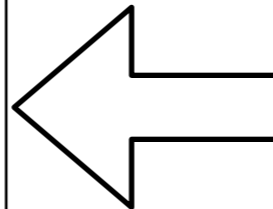
- **PAPI se configura en tiempo de inicialización**
 - Desde **httpd.conf** se incluye **Apache_PoA-1.5.conf**
 - Un fichero de configuración Apache pero con ciertas variables que definen el comportamiento del configurador de PAPI

Apache Configuration File

```
#Avoid real HOST goes away!!!  
<VirtualHost IP_Address>  
  ServerName hostname.fqdn  
  DocumentRoot /path/to/somewhere  
</VirtualHost>  
  
#Include configuration files  
Include Apache_PoA-1.5.conf
```

Apache_PoA-1.5.conf

```
PerlSetEnv PoA_CONFIG_FILE papi_conf.xml  
PerlSetEnv CONFIG_VERBOSE true  
PerlSetEnv CREATED_VIRTUALHOST true  
PerlSetEnv XML_CHECKER_ACTIVE false
```



- Estas variables son procesadas por **PAPI_Apache2_conf.pl**
- Este script llama al checker XML que genera la configuración de PAPI

- Qué hacen estas variables

- **PAPI_CONFIG_FILE** path/to/PAPI_XML_Configuration_File
 - Indica el fichero XML que debe procesar PAPI
 - El fichero XML puede ser chequeado contra su **DTD** o no
- **CEATED_VIRTUAL_HOST** [TRUE | FALSE]
 - Indica a PAPI si los contenedores/recursos a proteger ya están definidos en el httpd.conf
 - Es el usuario quien debe **establecer** esta variable
- **CONFIG_VERBOSE** [TRUE | FALSE]
- **XML_CHECKER_ACTIVE** [TRUE | FALSE]

- **El fichero de configuración XML**
 - Se usa el mismo principio que con PAPI-1.4
 - Se define una sección **Global** que afecta a todas las secciones o **PoAs**
 - Aunque se pueden redefinir parámetros para otras secciones
 - También se definen secciones de tipo **Server** que engloban
 - **Location**
 - **Directory**
 - **File**
 - Estas últimas también se pueden definir en la sección **Global**
 - Las secciones **Server** se convertirán en **VirtualHost**
 - Las directivas de configuración apenas han variado

- Para entender mejor el fichero XML...
- Veamos algunos ejemplos
 - Sección **<global>**

```
<Global>
  <Accept_File>/usr/local/apache/htdocs/image_ok.png</Accept_File>
  <Reject_File>/usr/local/apache/htdocs/image_nok.png</Reject_File>
  <Debug>1</Debug>
  <HKey>d41d8cd98f00b204e9800998ecf8427e</HKey>
  <LKey>22950f02a6d19ccff92447a48fdd985e</LKey>
  <Pubkeys_Path>/usr/local/apache/conf/PAPICConf/PoA</Pubkeys_Path>
  <Hcook_DB>/usr/local/apache/conf/PAPICConf/PoA/hcookdb</Hcook_DB>
  <Lcook_Timeout>18000</Lcook_Timeout>
  <CRC_Timeout>1800</CRC_Timeout>
  <Req_DB>/usr/local/apache/conf/PAPICConf/PoA/req_db.mldbm</Req_DB>
  <PAPI_AS id="PAPI-1.33" url="http://squirtle.rediris.es/cgi-bin/AuthServer">
    PAPI usando Apache-2.00
  </PAPI_AS>
</Global>
```

- Veamos algunos ejemplo II

- Seccion <Server>

```
<!-- Accessing locations at a remote server protected by form authentication -->
<Server name="squirtle2.rediris.es" port="8888" independent="0">
  <Apache_Tags>DocumentRoot "/usr/local/apache/htdocs"
  DirectoryIndex index.html index.php
  ServerAdmin daniel.garcia@rediris.es
  ErrorLog "logs/squirtle2_errors_log"
  CustomLog logs/squirtle2_acces_log common
</Apache_Tags>
  <Location path="/" id="remoteprot" filtered="true" independent="false">
    <GPoA_URL>wayf:built-in</GPoA_URL>
    <Remote_URL>http://localhost/~dgarcia/papiTestForm</Remote_URL>
    <Form_Processor url="http://localhost/~dgarcia/papiTestForm/testForm.php"
form_id="formAuthTest" buttonName="test">
      <Field name="usuario">daniel</Field>
      <Field name="passwd">daniel</Field>
    </Form_Processor>
  </Location>
</Server>
```


- Los ejemplos anteriores muestran la configuración mixta
 - Por qué mixta
 - Directivas PAPI → En el mismo fichero **papi_conf.xml**
 - Directivas Apache
- Existe la configuración independiente
 - Se definen los contenedores/secciones Apache en el fichero de configuración de éste
 - Las directivas PAPI se definen en el fichero XML de configuración

- Ejemplo de configuración independiente

httpd.conf

```
Include ./PAPIConf/PoA/Apache_PoA-1.5.conf
<VirtualHost *>
  ServerName squirtle.rediris.es
  <Location />
    PerlSetVar      ServiceID remote
    PerlOptions     +ParseHeaders
    SetHandler      perl-script
    PerlAccessHandler PAPI::Main
  </Location>
</VirtualHost>
```

papi_conf.xml

```
<Server name="squirtle.rediris.es" port="80" independent="0">
  <Location path="/" id="remote" filtered="true"
independent="false">
    <GPoA_URL>wayf:built-in</GPoA_URL>
    <Remote_URL>http://www.rediris.es</Remote_URL>
    <PAPI_Redirect src_regex="http://*.rediris.es"
      dst="http://squirtle.rediris.es:8888"/>
  </Location>
</Server>
```

- **PAPI-1.5 es aún una beta**
 - Por lo que seguimos trabajando y documentando PAPI
 - Os animo a probarlo y a enviarnos
 - BUGs
 - Sugerencias
 - Parches
 - etc...
- **PAPI-1.5 está alojado en la forja de RedIRIS**
 - <https://forja.rediris.es/projects/papi-perl/>
 - Aún no disponemos una versión Release Candidate
 - Pero si se puede hacer un **checkout** contra la forja
 - `svn checkout https://forja.rediris.es/svn/papi-perl`
- **Para cualquier otra cosa...**

daniel.garcia@rediris.es

- Algún
 - Comentario
 - Sugerencia
 - Duda
- Preguntas