DIA 2, Taller: Autenticación web con DNIe

Isaac Amezaga i Saumell

Common Criteria Evaluator, Applus+

22 de noviembre de 2007

1º ENCUENTRO NACIONAL DE LA INDUSTRIA DE SEGURIDAD

Índice

- **Autenticación SSL con certificados cliente**
- **Configuración servidores**
- **Compatibilidad con navegadores**

Logo empresa,
inamovible y
de este tamaño

Autenticación HTTPs con DNIe

Lectores de tarjetas

Existen una gran diversidad de lectores de tarjetas en el mercado, con diferentes tipos de conexión: USB, RS-232... La mayoría de lectores USB son conformes al estándar CCID, la mayor parte de los lectores RS-232 usan estándares de comunicación propietarios.

En cambio, a nivel de sistema operativo estas diferencias quedan bastante reducidas (una vez disponemos de los *drivers* apropiados) ya que las diferencias entre lectores quedan escondidos por la capa PC/SC. La capa PC/SC es básicamente un API que permite el acceso por parte de los programas de usuario al subsistema de tarjeta inteligente. La capa PC/SC está implementado cómo el servicio de *smartcard* en Windows y cómo el demonio PCSCd en unixes.

Autenticación HTTPs con DNIe

Los *drivers* de DNIe:

En el caso de DNIe los *drivers* Existen una gran diversidad de lectores de tarjetas en el mercado, con diferentes tipos de conexión: USB, RS-232... La mayoría de lectores USB son conformes al estándar CCID, la mayor parte de los lectores RS-232 usan estándares de comunicación propietarios.

En cambio, a nivel de sistema operativo estas diferencias quedan bastante reducidas (una vez disponemos de los *drivers* apropiados) ya que las diferencias entre lectores quedan escondidos por la capa PC/SC. La capa PC/SC es básicamente un API que permite el acceso por parte de los programas de usuario al subsistema de tarjeta inteligente. La capa PC/SC está implementado cómo el servicio de *smartcard* en Windows y cómo el demonio PCSCd en unixes.

Trazabilidad requisitos-implementación

	Verificación del certificado	CRL	OCSP	Verificación de usuario
Configuración Apache	X	X		
Código PHP			X	X

Configuración Apache

Veamos la configuración:

```
<VirtualHost *:443>
ServerAdmin webmaster@localhost
ServerName webdniauth
SSLEngine on
SSLCipherSuite HIGH:MEDIUM:-SSLv2
# Lista de CAs revocadas
SSLCARevocationFile "/var/www/webdniauth/ssl/certlogin/cacrl.crl“
# Certificado del servidor y clave p’ublica correspondiente
SSLCertificateFile "/etc/apache2/ssl/webdniauth.crt"
SSLCertificateKeyFile "/etc/apache2/ssl/webdniauth.key“
# Control de acceso a nivel de directorio
</VirtualHost>
```

Configuración Apache

Control de acceso a un directorio, obligatorio el uso de DNIe:

```
<Directory /var/www/webdniauth/ssl/certlogin>
# Certificado raiz DNIe
SSLCACertificateFile "/var/www/webdniauth/ssl/certlogin/ca.crt“
# El cliente debe autenticarse obligatoriamente con el certificado
SSLVerifyClient require
# Nivel máximo de profundidad (según infraestructura actual, 2)
SSLVerifyDepth  2
# Exportar contenido de certificados y certificados (para poder
    usarlos con PHP)
SSLOptions +StdEnvVars +ExportCertData
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>
```

Código PHP

Extracción del DNI con el que se ha *loggeado* el usuario:

```
function getDNI(){
    $dn_piece=split('/',$_SERVER['SSL_CLIENT_S_DN']);
    foreach($dn_piece as $x => $tkv){
            list($key,$value)=split('=',$tkv);
            if($key=="serialNumber")
            return $value;
    }
    return null;
}
```

Código PHP

Comprobación del estado del certificado del DNIe mediante OCSP:

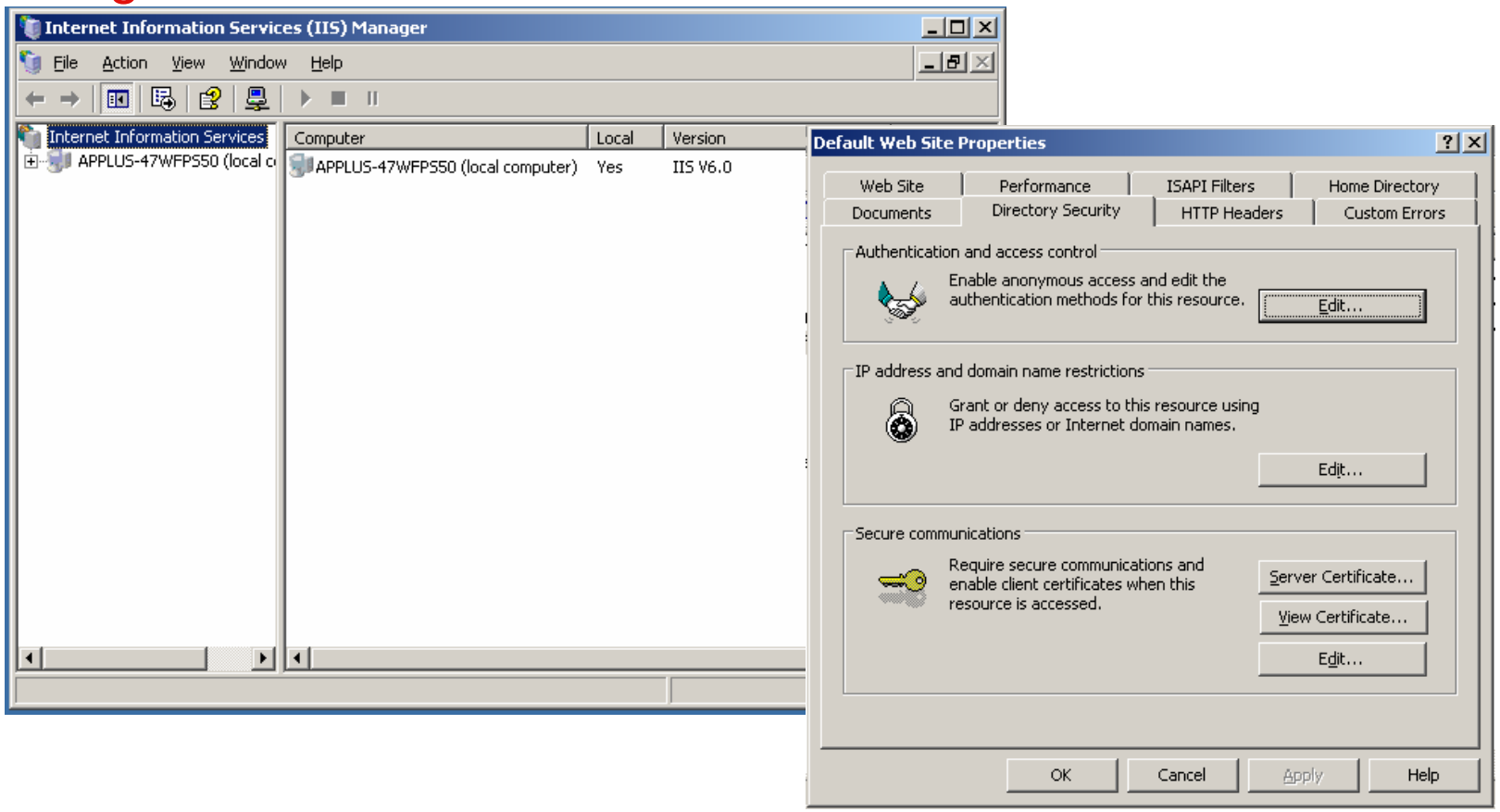
```
function OCSPCheck(){
    //write issuer certificate
    $issuercertfilename = tempnam("/tmp", "issuer");
    $issuercerthandler = fopen($issuercertfilename, "w");
    fwrite($issuercerthandler,$_SERVER["SSL_CLIENT_CERT_CHAIN_0"]);
    fclose($issuercerthandler);
    //write subject certificate
    $subjectcertfilename= tempnam("/tmp", "subject");
    $subjectcerthandler = fopen($subjectcertfilename, "w");
    fwrite($subjectcerthandler,$_SERVER["SSL_CLIENT_CERT"]);
    fclose($subjectcerthandler);
    //perform OCSP query
    $ocsphandler=popen("openssl ocsp -issuer ".$issuercertfilename." -cert
    ".$subjectcertfilename." -url http://ocsp.dnielectronico.es","r");
    list($filename,$result)=fscanf($ocsphandler,"%s %s");
    fclose($ocsphandler);
    //remove auxiliary files
    unlink($issuercertfilename);
    unlink($subjectcertfilename);
    return ($result=="good")?True:False;
}
```

Código PHP

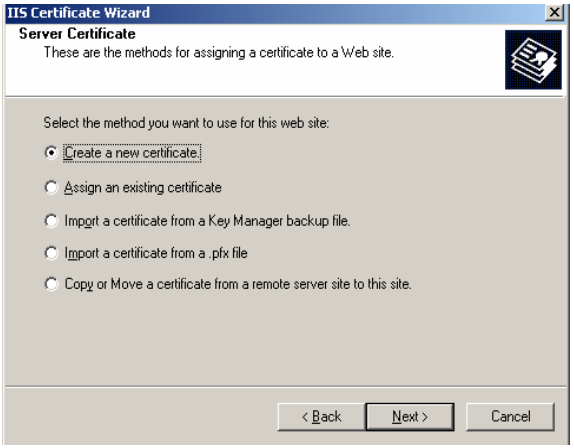
Función para comprobar el estado del certificado y proceder al login:

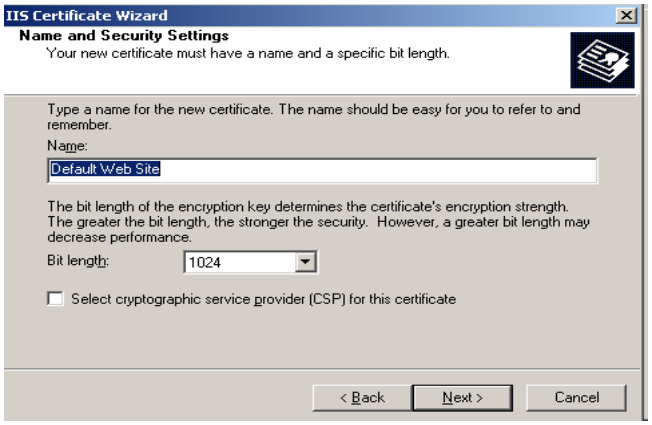
```
function certlogin(){
      global $passwd_db,$passwd_username,$passwd_password,$passwd_host;
      if(OCSPCheck()==True){
        $currentLoggedDNI=getDNI();
                    $link = mysql_connect($passwd_host, $passwd_username, $passwd_password);
                    if (!$link) { die('Could not connect: ' . mysql_error()); }
                    $db_selected = mysql_select_db($passwd_db, $link);
                    if (!$db_selected) { die ('Can\'t use passwd : ' . mysql_error()); }
                    $query = sprintf("SELECT username,role FROM passwd WHERE dni!='*' and
      dni='%s'", mysql_real_escape_string($currentLoggedDNI));
        $result = mysql_query($query);
        if (!$result) { die('Invalid query: ' . mysql_error()); }
        while ($row = mysql_fetch_assoc($result)) {
            $_SESSION['username']=$row['username'];
            $_SESSION['role']=$row['role'];
        }
      mysql_free_result($result);
      mysql_close($link);
      }
}
```

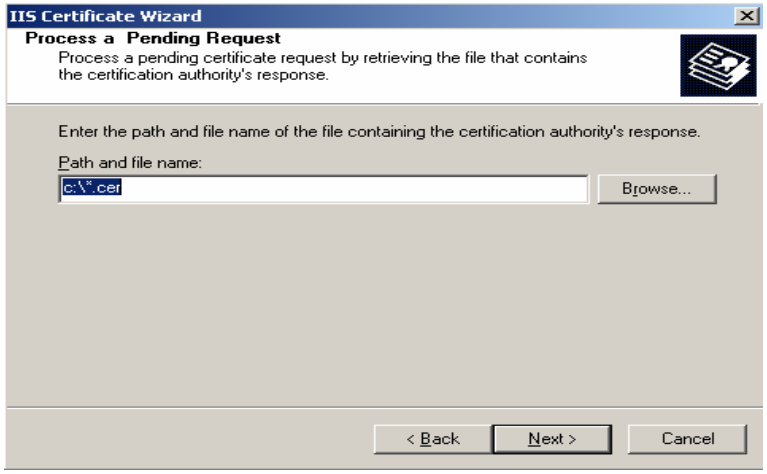
Configuración IIS

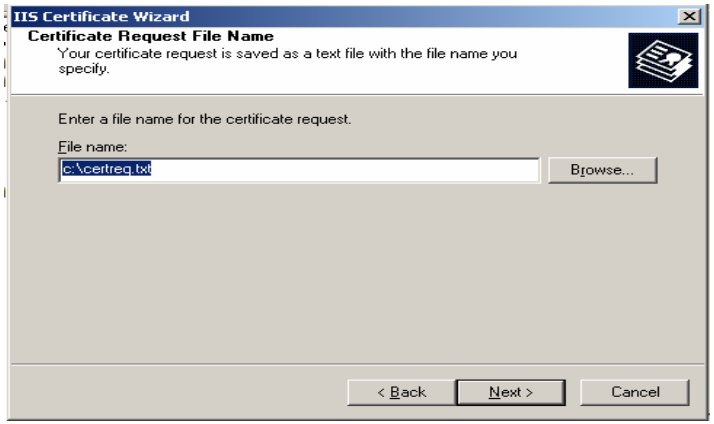


Configuración IIS, configurando el certificado del servidor

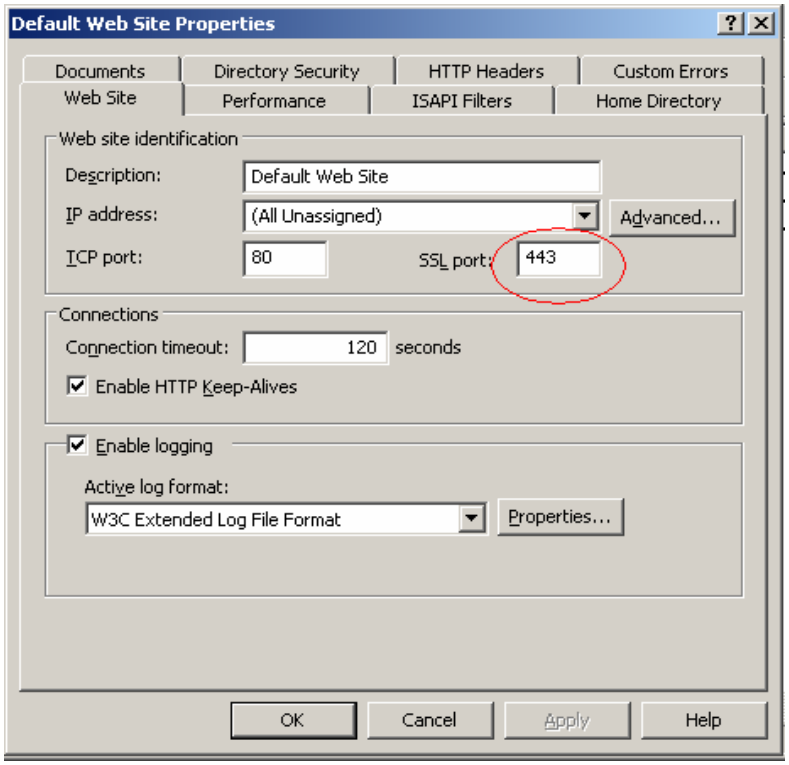


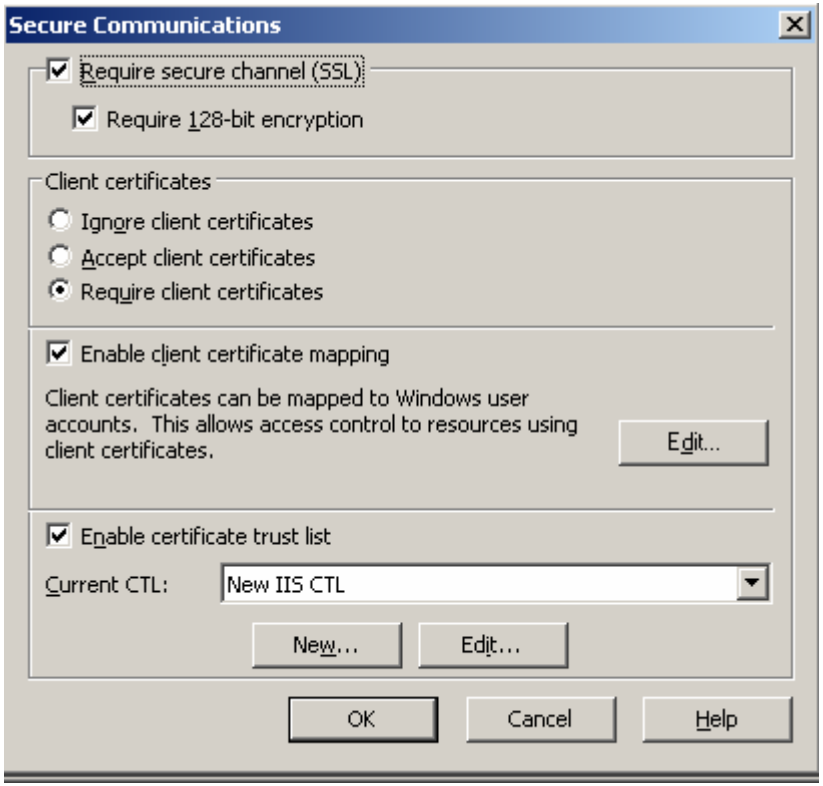




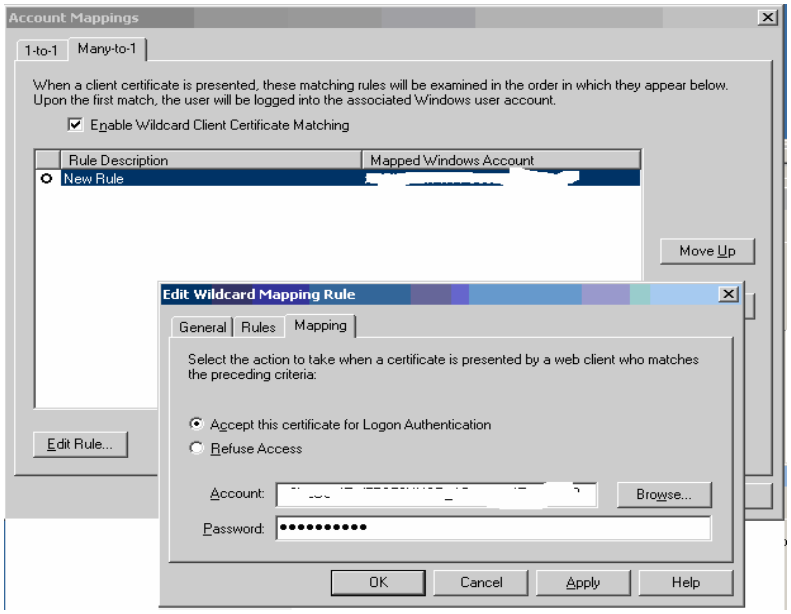


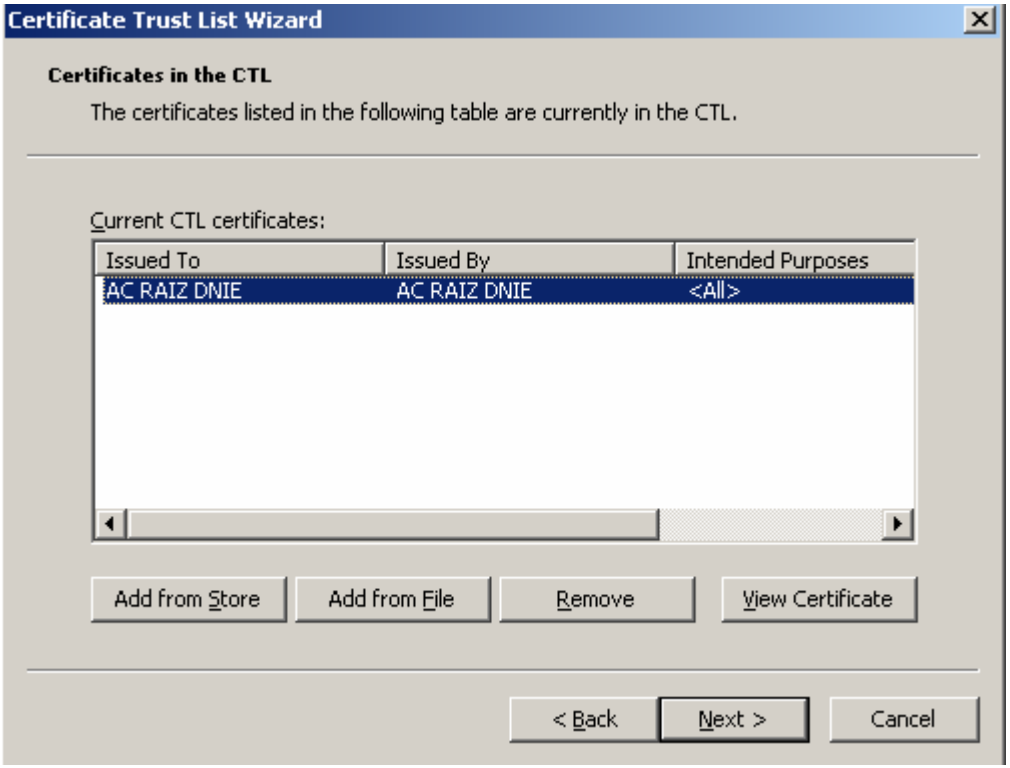
Configuración IIS, configurando autenticación mediante certificado (1)





Configuración IIS, configurando autenticación mediante certificado (2)





Compatibilidad con navegadores

Familia Mozilla y otros

Se carga una librería que proporciona la librería de soporte de DNIe para la librería PKCS#11. PKCS#11 es una API que estandariza el uso de mecanismos criptográficos. El uso de una librería PKCS#11 permite que otros navegadores (y de hecho otros programas) usen la librería para DNIe sin modificaciones y, cómo el estándar es accesible para todo el mundo facilita la interoperabilidad.

Internet Explorer

Al igual que con PKCS#11 es necesaria una librería que proporcione el acceso a bajo nivel al sistema de DNI electrónico. En este caso esta librería es llamada CSP (Criptographic Service Provider).

Desventajas: CryptoAPI sólo existe en sistemas Microsoft, aunque puede ser usada por cualquier programa de Windows que soporte CryptoAPI.