Implementación de una infraestructura de clave pública con herramientas de software libre

Miguel Solinas[1], Ricardo Justo Castello[2], Leandro Tula[3], Cesar Gallo[4], Javier Jorge[5], Daniel Bollo[6]

[1,3,4,5] Lab.de Arquitectura de Computadoras, FCEFyN, UNC,
Av.Velez Sarsfield 1611, 5000 Córdoba, Argentina.
msolinas@efn.uncor.edu; leandrotula@gmail.com; cesarhgp@gmail.com; javierbrk@gmail.com

[2,6] FCE, UNC,
Av. Valparaiso s/n - Ciudad Universitaria, 5000 Córdoba, Argentina.
castello@eco.unc.edu.ar; dbollo@eco.unc.edu.ar

Este trabajo se enmarca dentro del Proyecto de Investigación "Firma Digital en la UNC" aprobado y financiado por la Secretaría de Ciencia y Tecnología de la UNC para el ciclo 2012-2013 y cuyo grupo de trabajo está integrado por Castello, Ricardo; Solinas Miguel; Gallo Cesar Hugo; Tula Leandro Adolfo; Morales Hector Ruben; Jorge Javier; Rocha Vargas Marcelo Emilio; Montes Alfredo Miguel; Bollo Daniel; Brunello, Miguel Fernando; y Gauna Eduardo Jesus.

Abstract.

En el año 2012, en oportunidad de participar en las auditorías de las Autoridades de Certificación de la República Argentina, surgió la propuesta de pensar un laboratorio de firma digital para la Universidad Nacional de Córdoba (UNC). En la Facultad de Ciencias Exactas Físicas y Naturales (FCEFyN), hay una carrera: Ingeniería en Computación, entre cuyos descriptores está contemplada el área de conocimiento de la Seguridad Informática. Desde el año 2004 cuenta con una materia, Criptografía y Seguridad en Redes, donde se aborda el desarrollo de este conocimiento y se realizan experiencias prácticas. Conjuntamente, la Facultad de Ciencias Económicas (FCE) tiene materias como Tecnología de Información I, Auditoría de Sistemas Computarizados y Comercio Electrónico, donde se desarrollan contenidos, referidos a firma digital, pensando en su utilización por parte del usuario final. Un laboratorio de firma digital impactaría directamente con sus servicios a más de mil quinientos alumnos de dos unidades académicas. Esto no descartaba la posibilidad, con la colaboración de la Prosecretaría de Informática (PSI) de la misma Universidad, de pensar en recoger las experiencias del uso de la firma digital en el ámbito académico, para trasladarlas al ámbito de la gestión administrativa de la Universidad. Definitivamente un escenario que potenciaba el valor de un espacio dedicado a brindar servicios de firma digital.

En ese contexto, se evaluó la posibilidad de construir una infraestructura de clave pública o "Public Key Infrastructure" (PKI), utilizando software libre, dentro de la Universidad Nacional de Córdoba y aplicar los resultados de su experiencia a los procesos administrativos de la UNC. Para llevar adelante esta propuesta, se la enmarcó en el proyecto de investigación conjunto entre la FCE y la FCEFyN denominado "Firma Digital en la UNC". La primera Unidad Académica evaluaría la mejor manera y los procesos para introducir la tecnología y la segunda construiría un espacio tecnológico para experimentar con su aplicación.

En este trabajo se relata el proceso de construcción de una infraestructura de clave pública, utilizando software libre, para una fase desarrollo y experimentación en el ámbito académico de la UNC. En título 1 presentamos una introducción a los conceptos básicos de PKI, en el título 2 relatamos los pasos previos a la búsqueda de una solución de software; en el 3 describimos las herramientas de software libre por las que optamos; en 4 describimos las experiencias realizadas y por último en el título 5 presentamos algunas conclusiones.

Keywords: Seguridad Informática, PKI, Software Libre.

1 Introducción: Infraestructura de Clave Pública o PKI

La criptografía es el arte y ciencia de mantener los mensajes seguros; el criptoanálisis es el arte y ciencia de romper los textos cifrados. Ambas se engloban en la rama de las matemáticas llamada criptología [1]. Hoy se desarrollan sistemas de seguridad que aplican la criptografía para mantener seguros los datos sensibles de una comunicación. Entre ellos, por ejemplo, los que corresponden a transacciones comerciales sobre internet.

En la criptografía moderna se pueden diferenciar dos claras vertientes, la criptografía simétrica y la criptografía asimétrica o de clave pública. La primera se basa en algoritmos simétricos que usan una misma clave para encriptar y desencriptar mensajes, y la segunda se basa en algoritmos de clave pública que usan claves distintas para la encriptación y desencriptación.

Una PKI es el conjunto de hardware, software, personas, políticas y procedimientos que se necesitan para crear, manejar, almacenar, distribuir y revocar certificados digitales basados en criptografía asimétrica. Con ella es posible llevar adelante servicios de seguridad complejos para una comunicación distribuida geográficamente para una población numerosa de usuarios.

Los certificados digitales son una parte fundamental de la tecnología PKI. Son los contenedores para la distribución de una de las claves: la pública. Los esfuerzos por desarrollar una arquitectura basada en certificados en Internet llevaron a adoptar el modelo de arquitectura basado en los certificados X.509 desarrollado por el grupo de trabajo PKIX (Public-Key Infrastructure X.509) del IETF (Internet Engineering Task Force). Un paréntesis: la misión del ITEF es básicamente hacer que internet funcione mejor a través del uso de estándares abiertos. Actualmente el término PKIX hace referencia a la infraestructura de clave pública basada en certificados X.509 y el término certificado PKIX usualmente hace referencia a los perfiles de certificado y de listas de revocación basados en el estándar de certificados X.509 v3. Cabe mencionar que el grupo de trabajo PKIX ha producido una serie de estándares para satisfacer las necesidades de una PKIX, como el RFC 3280 (Certificate and CRL Profile), el RFC 2560 (Online Certificate Status Profile), el RFC 3161 (Time Stamp Protocol), entre otros.

1.1 Componentes del modelo de una PKIX

En el modelo de una PKIX se pueden distinguir los siguientes componentes con sus respectivas funciones: Entidad Final, Autoridad de Certificación, Autoridad de Registro, Publicador de listas de certificados revocados o “Certificate Revocation List” CRL y Repositorio. A continuación se presenta una breve descripción de cada uno de ellos.

Entidad final.

Es el nombre genérico que reciben no sólo los usuarios de una PKI sino también los dispositivos que forman parte de ella como ruteadores o servidores. Pueden ser identificados en el campo que define al propietario del certificado X.509. Las entidades usuarios normalmente utilizan los servicios que ofrece la PKI y los dispositivos normalmente los soportan.

Autoridad de Certificación (AC).

Es la entidad encargada de emitir los certificados digitales X.509 y usualmente también las CRL, aunque a veces delega la función a un elemento Emisor de CRL. También puede desempeñar funciones administrativas como las de registro de entidades finales o publicación de certificados pero normalmente estas funciones son desempeñadas por las autoridades de registro. Existen como mínimo dos tipos de AC – Ver Figura 1 – en una jerarquía de certificación: la AC Raíz (RootCA) y la AC Subordinada (SubCA). La primera es la que emite certificados a otras AC subordinadas. La segunda es la que emite certificados de entidad final y cuyo certificado ha sido firmado digitalmente por la AC raíz. En una jerarquía puede haber una o varias AC subordinadas.

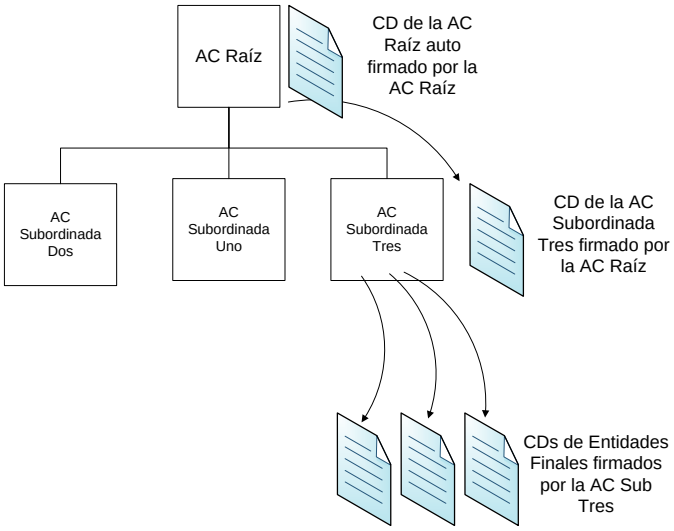

Figura 1: Jerarquía de certificación

Autoridad de Registro (AR).

Es un elemento opcional de la arquitectura PKIX que puede asumir algunas funciones administrativas de la AR, tal vez la más común sea el proceso de registro de las entidades finales, pero también puede realizar otras funciones como el proceso de revocación de certificados y el manejo de los datos de la entidad final. Dentro de una jerarquía de certificación pueden existir una o varias autoridades de registro.

Emisor de CRL

Es una entidad opcional de la arquitectura PKIX a la que la AC puede delegar la función de emitir la CRL. Algunas veces se encuentra integrada en la AC a modo de servicio.

Repositorio.

Es el término que hace referencia a cualquier método existente para almacenar certificados y CRL, y así poder ser obtenidos por las entidades finales. Uno de estos métodos es el protocolo "Lightweight Directory Access Protocol" (LDAP).

Autoridad de Validación (AV).

En ciertas jerarquías de certificación puede existir un sexto elemento que se encarga de dar información sobre la vigencia de los certificados digitales. Este elemento se llama Autoridad de Validación. Es elemento encargado de recoger la información de los certificados que han sido revocados y publicados en la CRL. Esta autoridad puede utilizar el protocolo "Online Certificate Status Protocol" (OCSP) para prestar los servicios de validación y no está incluida en la arquitectura PKIX porque es común que estos servicios los preste la misma AC. Pero en caso que se requieran aislar los datos de la comprobación de la vigencia de un certificado de los datos de identidad de su titular se recomienda usar una autoridad de validación distinta de la de certificación.

1.2 Funciones de gestión de una PKIX

Para la gestión de una PKIX se define un Manual de Procedimientos Operativas que debe contemplar las siguientes tareas:

Registro.

Es el proceso por el cual una entidad final registra sus datos directamente en una AC o por medio de una AR. También incluye procedimientos especiales de mutua autenticación, para los cuales se suelen generar claves privadas compartidas.

Inicialización.
Es el proceso por el cual el cliente se inicializa, de forma segura, con su clave pública y con otra información relevante de la AC en la que se ha registrado. La información de esta AC es la que se utilizará en el camino de validación de certificados. Esto es necesario para que el sistema del cliente pueda operar correctamente.

Certificación.
Es el proceso por el cual la AC crea un certificado que certifica que una determinada clave pública pertenece a una entidad final, y se lo retorna al sistema del cliente o lo almacena en un repositorio.

Recuperación del par de claves.
Este proceso permite a las entidades finales volver a obtener su par de claves, para esto las piden a una entidad autorizada de resguardo de claves. Usualmente es la misma AC que emitió el certificado la que hace el papel de esta autoridad.

Actualización del par de claves.
Es el proceso por el cual luego de un tiempo determinado se actualizan las claves de una entidad final y se le vuelve a emitir el respectivo certificado. Normalmente sucede cuando se cumple el tiempo de validez de un certificado o cuando el certificado ha sido revocado, y se realiza volviendo a crear un nuevo par de claves.

Pedido de revocación.
Ocurre cuando una persona autorizada avisa a la AC que ha ocurrido un suceso anormal que compromete la información contenida en un CD y pide la revocación del certificado de una entidad final. Los motivos del pedido de revocación pueden ser o el compromiso de la clave privada o el cambio de nombre de la entidad, entre otros.

Certificación cruzada.
Es el proceso en el cual dos AR intercambian la información necesaria para emitir un certificado cruzado. Este certificado es el que una AC raíz emite a una AC subordinada, siempre y cuando esta última AC cuente con una clave de firma autorizada para emitir certificados, normalmente, de entidad final.

2 Proceso de construcción de la PKI de la UNC

Para llevar adelante la construcción de la PKI de la UNC se tuvo que evaluar algunos otros requerimientos aparte del software. Brevemente mencionaremos en este título la resolución dada a los problemas de espacio físico, hardware y recursos humanos. En el título siguiente describiremos el proceso de selección de software.

La disponibilidad física se resolvió utilizando una sala de cómputo con la que cuenta el Departamento de Computación de la FCEFyN, en la que residen una Supercomputadora y el Cluster de la Secretaría de Ciencia y Tecnología (SeCyT) de la UNC. Esta sala fue construida con fondos provenientes de un Proyecto de Adecuación y/o Mejora de Infraestructura (PRAMIN) y dispone de 40 metros cuadrados de superficie adecuados para la instalación de equipos de computación de alto desempeño

Para resolver el problema del hardware, en el mismo Departamento de Computación de la FCEFyN se disponía de un servidor de la marca DELL, cuádruple core, ocho gigabytes de memoria RAM y discos redundantes de quinientos gigabytes. Adquirido con fondos PACENI para alojar desarrollos para las carreras de Ingeniería en Computación y Biología, este servidor aún disponía del espacio y capacidad de proceso suficientes para dar los primeros pasos en la construcción de la PKI.

Para cubrir la necesidad de recursos humanos se reunió un equipo profesional coordinado por el Titular de la cátedra de Criptografía y Seguridad en Redes, al que se sumó uno de los primeros Ingenieros en Computación y dos alumnos avanzados de la carrera, en condiciones de realizar su Proyecto Integrador a los cuales se les gestionaron becas para finalización de carrera TICs.

Esto nos permitió en un principio tener una capacidad de trabajo para investigar, diseñar, implementar y evaluar en el término de un par de años la construcción de una PKI.

El paso siguiente era evaluar la existencia de software libre para la construcción de una entidad que emitiera certificados digitales y permitiera definir las funciones básicas de una infraestructura de firma digital, como son la AC Raíz, la AR, una entidad encargada de publicación de CRL e implementaciones del protocolo Online Certificate Status Protocol (OCSP).

3 Herramientas para la construcción de la PKI

Al momento de tener que elegir una solución para la construcción de la infraestructura de firma digital, pensando en software libre, existían dos posibles candidatos. OpenCA, oficialmente conocida como OpenCA PKI Research Labs [2] y Enterprise Java Bean Certificate Authority (EJBCA) [4].

OpenCA es el resultado de un esfuerzo colaborativo para desarrollar una autoridad de certificación con todas sus funcionalidades manteniendo la filosofía del código abierto. Si bien tiene dentro de sus objetivos el desarrollo de los principales protocolos utilizados junto a una criptografía fuerte, se puede observar que no ha tenido la evolución esperada, ni en el desarrollo de sus componentes, ni en la consolidación de sus métodos para lograr la construcción de un sistema complejo. Creemos que los criterios en las decisiones, al momento de definir el proceso de construcción de software y las herramientas a utilizar, impactan en mayor medida que las propias personas que participan en su construcción. Por otro lado, esta herramienta cuenta con un limitado soporte para diferentes proveedores de servicios de certificados o “Certificate Services Provider” (CSP), “smartcards” y no es escalable.

Por otro lado se encuentra EJBCA. Se trata de un proyecto que ha tenido un desarrollo continuo desde hace más de una década, produciendo una solución que se ha ido realimentando en la solución de sus problemas hasta tener hoy una versión confiable, estable, robusta y de código abierto. Hay que destacar que esta solución esta soportada por PrimeKey [5], quien ofrece soluciones que integran el framework EJBCA en aplicaciones de seguridad complejas como las que brinda una infraestructura de firma digital.

La decisión por EJBCA se vio favorecida por estos factores:

- Continuidad del proyecto.
- Procesos de producción y actualización de componentes.
- Existencia de una soporte externo.
- Mayor difusión de su aplicación en soluciones en producción.
- Escalabilidad basada en Java 2 Enterprise Edition (J2EE) y bajo acoplamiento de módulos.

EJBCA se eligió también por la flexibilidad de configuración de sus componentes y por la facilidad con la que estos se pueden integrar dentro de su arquitectura de certificación.

Cuenta un módulo AR desplegado por lo que en un futuro se podrían plantear solo integrar el componente AC de la herramienta, esta integración podría realizarse utilizando la interfaz “web service” o el servicio “XML Key Management Specification” (XKMS) que ofrece EJBCA.

Por otro lado la herramienta es lo suficientemente robusta como para soportar infraestructuras de múltiples niveles de AC dentro de una sola instancia de ejecución y se despliega correctamente dentro de ambientes de “clustering” para el que ofrece servicios de monitorización. También está diseñada para soportar diferentes módulos “Hardware Security Module” (HSM).

Otro motivo importante es que la herramienta utiliza la licencia “Lesser General Public License” (LGPL). La diferencia de esta licencia con la “General Public License” (GPL) es que puede ser integrada

casi sin ninguna limitación con cualquier programa propietario. Algunos programas que también la usan son el navegador Mozilla y el proyecto OppenOffice.

La herramienta puede ser instalada en cualquier servidor de aplicaciones J2EE y aunque la instalación más sencilla y más probada es sobre JBoss [6] (licenciado bajo GPL), también existe documentación para su instalación sobre otros servidores como GlassFish [7], Weblogic [8], u Oracle Containers for J2EE (OC4J) [9].

EJBCA es independiente de la base de datos que se utilice ya que se encuentran en niveles distintos de la arquitectura, o sea se puede instalar tanto con MySQL [10], PostgreSQL [11], DB2 [12] y MS-SQL [13] como con cualquier otra.

Entre otras características favorables de EJBCA está el que no solo soporta claves de firma RSA sino también las "Elliptic Curve Digital Signature Algorithm" (ECDSA). Los algoritmos que se basan en la criptografía de curva elíptica son los que se perfilan como los algoritmos base de la criptografía de clave asimétrica del futuro; ECDSA es uno de ellos.

Algunas de las instalaciones de EJBCA más relevantes son las de la Autoritat de Certificació de la Comunitat Valenciana [14], con más de 200.000 certificados personales emitidos. El Ministerio de Defensa y Ministerio de Finanzas Francés [15], dos de los organismos públicos más grandes de Europa. El Cuerpo Nacional de Policía Sueco [15], que tiene emitidos ya más de 25.000 certificados de empleado y ofrece soluciones de firma para el pasaporte electrónico sueco.

En Argentina, existen experiencias exitosas de utilización tanto de OpenCa como de EJBCA. La primera es utilizada por la Universidad Nacional de La Plata para su PKIGrid [3]. Mientras que EJBCA está siendo utilizada como componente principal de una de las Autoridades de Certificación homologadas por la Oficina Nacional de Tecnologías de la Información (ONTI): AFIP.

3.1 El proyecto EJBCA

Enterprise Java Bean Certificate Authority (EJBCA), en su sitio web oficial [4], se define como una autoridad certificadora multifuncional, basada en tecnología Java 2 Enterprise Edition (J2EE). Constituye un AC robusta, escalable, de alto rendimiento y basada en componentes. Es multifuncional ya que puede realizar todas las funciones propias de una AC, y alguna más, sin delegarlas a otras herramientas. Se dice que está basada en componentes porque cada una de las funciones las realiza un componente de la misma herramienta.

Como autoridad de certificación su principal función es la emisión de certificados. Estos se pueden emitir para distintos propósitos como autenticación robusta de usuarios y dispositivos, comunicación segura con clientes y servidores SSL, firma y encriptación de correo electrónico, firma de documentos, acceso a sistemas usando tarjetas criptográficas, asegurar conexiones VPN y muchos otros. EJBCA necesita de un servidor de aplicaciones y una base de datos para realizar su actividad correctamente en un entorno de producción.

El proyecto EJBCA fue iniciado por Tomas Gustavsson y Philip Vendil en el año 2001, en Noviembre de ese mismo año apareció la primera versión de EJBCA, la 1.0. Desde entonces se han publicado más de 50 nuevas versiones, actualmente, la última versión estable es la 4.0.15. Tiene un equipo de desarrolladores activo, y está alojado en el repositorio de proyectos opensource más grande del mundo: SourceForge, en donde cuenta con gran actividad.

En este sitio mantiene dos listas de correos y cuatro foros públicos de discusión bastante utilizados donde se puede encontrar rápida respuesta a cualquier duda específica. En sí, cuenta con gran cantidad de información disponible en su página web oficial, en su sitio wiki, en su blog y en los mismos paquetes de instalación.

EJBCA como solución PKI ha crecido y se ha popularizado en los últimos años existiendo una wiki y un blog como fuentes de información relevantes al momento de tener que solucionar problemas. De aquí

en adelante la aparición de nuevas ayudas e información sobre el tema se generarán al ritmo de aparición de nuevos usuarios.

Características y funcionalidades.

EJBCA tiene tres funcionalidades principales: la de AC, la de AR y la de publicación de CRL. También brinda funcionalidades adicionales como las de contestardor de solicitudes OCSP, cliente OCSP, almacenamiento y publicación de certificados y CRL. También ofrece administración por línea de comando, notificaciones a usuarios por correo electrónico, informes del sistema, gestión y firma de logs, integración con HSM, recuperación de claves, soporte para claves ECDSA, servicios que soportan los protocolos CMP (Certificate Management Protocol), XKMS y SCEP (Simple Certificate Enrollment Protocol) entre otros.

Una funcionalidad importante que tiene la herramienta es la administración remota por Web Services, para esto tiene una interfaz llamada WSCLI, y a través de ella se puede acceder a su componente CA desde una RA externa.

Dos de las características principales de la herramienta son su flexibilidad y su independencia de la plataforma, la primera gracias a que su arquitectura está basada en componentes y la segunda como consecuencia de estar desarrollada totalmente en JAVA. EJBCA puede utilizarse tanto para operar aislada, ya que no tiene dependencias con otras herramientas, o integrada en cualquier aplicación J2EE utilizando los componentes de la herramienta que se consideren necesarios.

Interfaces de la herramienta.

Se tienen tres interfaces básicas de interacción con la herramienta: la “public web”, la “admin web” y la “Command Line Interface” (CLI). La primera es una interfaz gráfica pública que es accesible por todos los usuarios en general y les permite obtener sus certificados, las CRLs de la AC y verificar la validez de cualquier certificado. La segunda es también una interfaz gráfica pero que es solo accesible por los administradores de la herramienta. A través de esta interfaz ellos pueden realizar tareas de administración. La tercera interfaz es una línea de comandos que permite realizar algunas operaciones de administración y es especialmente útil cuando se necesita realizar alguna acción por medio de scripts.

Interfaz de administración flexible.

EJBCA ofrece un fácil y sencillo control de los privilegios de administración de los administradores de la arquitectura de certificación ya que es posible establecer grupos de administradores de AC, administradores de AR, superadministradores y supervisores.

Esto aumenta el nivel de seguridad de la infraestructura ya que las funciones de administración están diferenciadas por el tipo de administrador que se sea. Esta interfaz gráfica también permite controlar el formato de los certificados gracias a la definición de perfiles de certificado y de entidad final.

Unidad de firma.

EJBCA permite elegir el tipo de unidad de firma según la política de certificación que se quiera implementar. Se puede usar un módulo software para almacenar las claves de firma o se puede requerir el uso de HSM o de tarjetas criptográficas.

Almacenamiento de certificados.

Los certificados que emite EJBCA se almacenan en la base de datos a la que está ligada, y a través de unos componentes publicadores llamados publishers se publican en directorios específicos. Esta funcionalidad es completamente configurable.

Obtención de los certificados de usuario.

Luego de que se ha dado de alta un usuario se le tiene que emitir el certificado, antes en el proceso de registro se le ha debido crear un nombre de usuario y una contraseña. Con estos datos el usuario desde su explorador accede a la “public web” y se autentica, crea su par de claves y envía la clave pública junto con sus datos de registro a la AC en formato PKCS10 (Certification Request Standard). Luego de esto la

AC le crea el certificado al usuario y se lo envía. El mecanismo de autenticación descripto en este proceso se puede reemplazar por otro que se adecue a los requerimientos de seguridad del prestador de servicios de certificación.

En la Figura 2 se muestra un diagrama más detallado del proceso de obtención de certificados. El usuario desde el explorador interactúa con el servidor web, que es el que contiene los servlets de la aplicación, luego éste interactúa con el servidor de aplicaciones que contiene a la aplicación EJBCA y a sus componentes AC y AR.

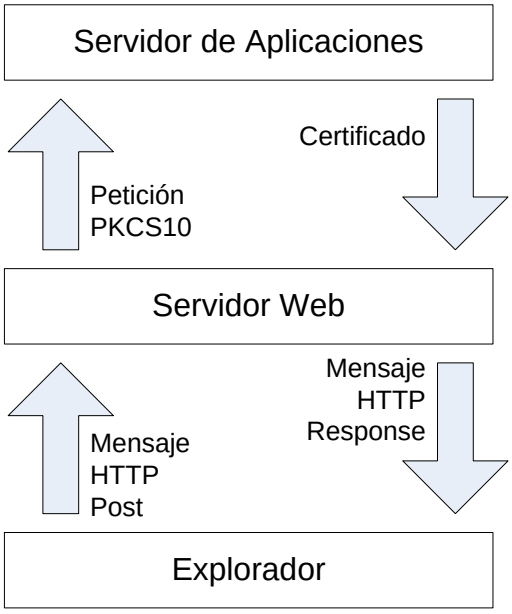

Figura 2: Proceso de obtención de CD

Un ejemplo de arquitectura propuesta.

En la Figura 3 se muestra un ejemplo de implementación de EJBCA. Con él se pretende mostrar la flexibilidad de la configuración de la herramienta. Se ve que la herramienta se ha instalado en un servidor que hace solo de AC y está conectado a un servidor de backup en el que almacenan los "logs" de eventos y al que copia periódicamente toda su base de datos. Se tiene una AR externa que recibe las peticiones de certificado y un contestador OCSP externo que recibe las peticiones OCSP; evitando de esta manera que la AC reciba tráfico directamente. Se tiene también un firewall que está configurado para que solo permita el tráfico saliente de la AC y no el tráfico entrante, lo que hará la AC será tomar periódicamente las peticiones que necesite procesar tanto de la AR como del contestador OCSP. Otro aspecto a resaltar es que se ha implementado un "cluster" de contestadores OCSP externos. Esto es útil cuando se espera emitir pocos certificados, pero recibir numerosas peticiones OCSP, por lo que no es necesario implementar un cluster de AC. El balanceador de carga recibe el tráfico de Internet y lo redirige al contestador OCSP que tenga menos carga en ese momento.

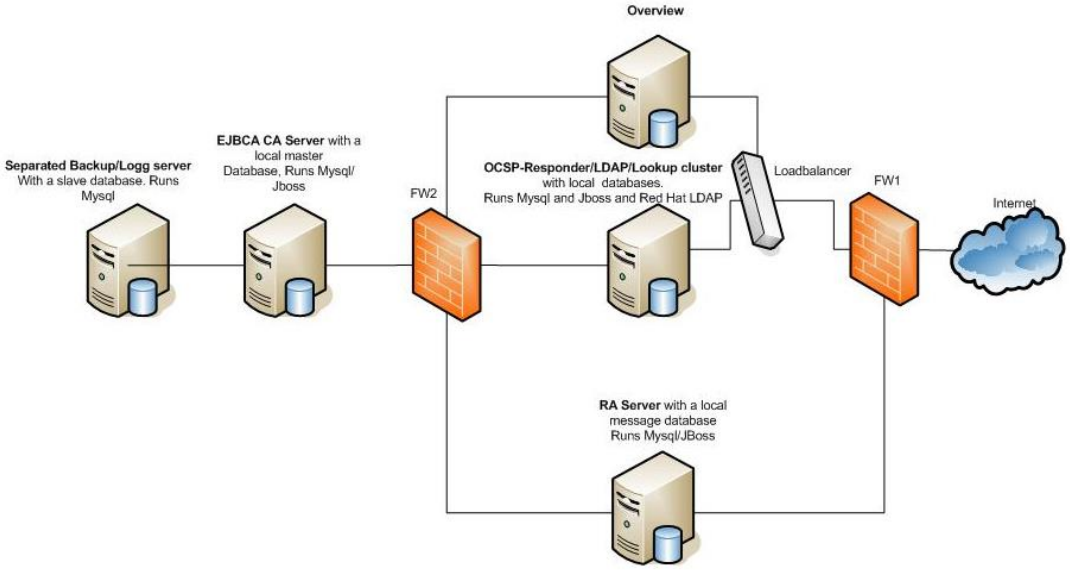

Figura 3: Arquitectura de una PKI propuesta por EJBCA

3.2 El proyecto Vyatta

Como se puede observar en la Figura 3, la opción de construir una infraestructura de clave pública con todos sus componentes lleva a tener que evaluar la posibilidad de contar con más de media docena de componentes de hardware, entre servidores y firewall. Esto trae como consecuencia no solo disponer de los mismos, sino del espacio que ocupan en sala de cómputos, el tiempo que demanda su mantenimiento, los esfuerzos para la realización de copias de seguridad y la confiabilidad al sumar componentes.

La mejor solución pasa por la virtualización de componentes. Lo cual no representaba un problema para el caso de EJBCA, pero si para los componentes de comunicaciones.

Para solucionar este problema se consideró el trabajo final “Implementación de un laboratorio virtual con acceso remoto para la enseñanza de redes de computadoras” [16]. En este trabajo los autores plantean la necesidad de contar con soluciones virtualizadas para diferentes equipos de comunicaciones como routers y firewall, necesarios para un laboratorio de redes de computadoras.

A partir de los resultados de este trabajo, para virtualizar los dos firewall se optó por el proyecto Vyatta [17]. Es un producto de código abierto, pensado para crear un router de altas prestaciones que nos permita ofrecer cualquier servicio de red, entre ellos el de firewall. El proyecto está basado Debian [17]. Contiene numerosas aplicaciones de red y de manejo de interfaces. Existe una versión comercial, cuyo modelo de negocio se basa en una suscripción mensual a cambio de servicios de actualizaciones, soporte técnico y formación.

El sistema Vyatta se diseñó con el propósito de reemplazar versiones comerciales de Cisco, desde la línea 1800 hasta la 7200, apoyándose en el bajo coste y flexibilidad de las soluciones open source, y que además al estar basado en Linux funcionaría en la mayoría de sistemas basados en la tecnología x86. Vyatta permite la actualización del sistema mediante repositorios y la posibilidad de añadir nuevos paquetes. De esta forma, vyatta se convierte en un sistema totalmente configurable que nos ofrece un gran abanico de servicios. Entre los cuales destacamos:

- Protocolos de enrutado avanzados: BGP/OSPF/RIP usando Quagga.
- Seguridad: antivirus ClamAV, Sistemas de Prevención de Intrusos (IPS), Sistema de Detección de Intrusos (IDS) utilizando Snort.
- Monitorización de Redes: SNMP, Wireshark y tcpdump.
- QoS y limitaciones de ancho de banda.
- Firewall avanzado: NAT, QoS, DMZ, balanceo de carga, filtrado de tráfico, VLAN, IPSec, etc.
- Servicios de acceso remoto: SSH y Telnet.

- Servidor DHCP.
- Cliente y servidor VPN (PTPP y OpenVPN).
- Encapsulamiento y armado de túneles: Cisco HDLC, Frame Relay, PPP, PPPoE, Multilink.
- Clustering.
- DNS forwarding, Dynamics DNS.
- URL filtering.
- Web caching.
- Otros.

3.3 Resumen de componentes utilizados

En la construcción de la PKI se utilizaron las siguientes herramientas de software libre:

- Virtual Box
- GNU Linux Ubuntu Server
- MySQL
- JBoss
- EJBCA
- Vyatta

4 Experiencias realizadas

A finales del año 2012 se tuvo la primera versión operativa de una autoridad de certificación. A partir de ello se han realizado diferentes experiencias creando jerarquías de AC, perfiles de AR y perfiles de entidades finales. Se documentaron los procesos de construcción e instalación y se definió la arquitectura que se pretenden construir para la AC de producción.

Se participó en la muestra de Arte, Ciencia y Tecnología de la UNC Cuatrociencia, donde se desplegó una Autoridad de Certificación denominada "ACCuatrociencia" junto a un perfil de usuario final con datos mínimos, para mostrar el proceso de emisión de certificados digitales y su aplicación en la firma de documentos.

Al momento de escribir este trabajo se ha comenzado a trabajar con la Cátedra de Tecnología de Información I de la FCE para definir el procedimiento que permita emitir certificados digitales a los primeros 500 alumnos y mostrar su utilización en la firma de documentos y correo electrónico. Para el segundo semestre de este mismo año 2013 se pretende sumar a la Cátedra de Criptografía y Seguridad Informática de la FCEFyN el desarrollo de prácticos con la herramientas EJBCA que contemple la construcción de autoridades de certificación con diferentes perfiles de entidades finales, la emisión de certificados digitales para las mismas y su aplicación para brindar servicios de seguridad complejos para múltiples usuarios.

Estudio de vulnerabilidades de la PKI

Recientemente se ha comenzado un análisis de vulnerabilidades conocidas de los componentes de la arquitectura construida. En el marco de una Práctica Supervisada, dentro del Laboratorio de Arquitectura de Computadoras del Departamento de Computación de la FCEFyN de la UNC se ha iniciado un estudio de la seguridad de los componentes individuales de la arquitectura de la PKI construida. Esto persigue el objetivo de poder diagnosticar y mejorar el nivel de seguridad de sus componentes y del conjunto.

De este modo se pretende elaborar una serie de recomendaciones respecto a las configuraciones o actualizaciones que sean necesarias para mejorar la seguridad de la infraestructura de firma digital de producción que se pretende construir como continuidad del proyecto para los próximos años. Por otro lado, adquirir también el conocimiento y elaborar los procedimientos para una evaluación permanente de su nivel de seguridad.

5 Conclusiones

En un tiempo relativamente corto, con los elementos y recursos que teníamos a nuestro alcance, con un presupuesto absolutamente limitado y contando con la capacitación realimentada de trabajos realizados en otros proyectos integradores, de alumnos de nuestra propia carrera de Ingeniería en Computación, hemos construido una PKI de desarrollo. Hoy podemos experimentar con la construcción de nuestra propia autoridad de certificación (AC) y emitir nuestros propios certificados digitales para su utilización en varias cátedras de diferentes unidades académicas de la Universidad Nacional de Córdoba.

Si bien la PKI hoy levantada es experimental y totalmente de laboratorio, estamos avanzando sobre la definición de requerimientos y diseño de una PKI de producción. Que contemple políticas de funcionamiento de acuerdo a la Ley de Firma Digital argentina y a las Políticas de Seguridad de la UNC. Esto nos permitirá aproximar una solución que estaría muy próxima a poder brindar servicios de emisión de Certificados Digitales.

Si bien esto es un objetivo en sí mismo, creemos que el mayor valor agregado se alcanzará de ahora en adelante. Con los CD emitidos es posible avanzar sobre el desarrollo de software y aplicaciones que utilicen certificados digitales y la infraestructura de clave públilca. Esto nos permitirá realimentar los aspectos (funcionalidades transversales a todos los requerimientos) de las aplicaciones a las condiciones de emisión y estructura de los CD en nuestra propia AC. Esto generará nuevos proyectos que partirán de la infraestructura de firma digital como un componente básico a tener en cuenta en la construcción de aplicaciones que puedan brindar servicios de seguridad complejos en ambientes distribuidos y con numerosos usuarios.

Sin lugar a dudas la construcción relatada en este trabajo, su análisis, los resultados hasta ahora obtenidos y sobre todo los trabajos a futuro, no serían posibles sin la ayuda de estándares abiertos y software libre. Por lo que es fundamental continuar fomentando este tipo de prácticas especialmente desde los ambientes públicos, especialmente desde las universidades.

6 Bibliografía

1. Applied Cryptography, Bruce Schneier, John Wiley & Sons, 1996.
2. OPENCA; http://www.openca.org/; consultado en Mayo 2013.
3. PKIGrid CA; http://www.pkigrid.unlp.edu.ar/; consultado en Mayo 2013.
4. EJBCA; http://www.ejbca.org/; consultado en Mayo 2013.
5. Primekey; http://www.primekey.se/; consultado en Mayo 2013.
6. JBoss; http://www.jboss.org; consultado en Mayo 2013.
7. GlassFish; https://glassfish.java.net/; consultado en Mayo 2013.
8. Orace WebLogic Server; http://www.oracle.com/technetwork/middleware/weblogic/overview/index.html; consultado en Mayo 2013.
9. Oracle Containers for J2EE (OC4J); http://www.oracle.com/technetwork/middleware/ias/index-099846.html; consultado en Mayo 2013.
10. MySQL; http://www.mysql.com/; consultado en Mayo 2013.
11. PostGreSQL; http://www.postgresql.org.es/; consultado en Mayo 2013.
12. DB2 Express Edition; http://www-03.ibm.com/software/products/us/en/db2express-edition?S_CMP=ecddww01&S_ TACT=wikies; consultado en Mayo 2013.
13. SQL Server; http://www.microsoft.com/en-us/sqlserver/default.aspx; consultado en Mayo 2013.
14. Autoritat de Certificació de la Comunitat Valenciana; http://www.accv.es/; consultado en Mayo 2013.
15. Wikpedia; http://en.wikipedia.org/wiki/EJBCA; consultado en Mayo 2013.
16. Giraudo N., Veneranda G.; “Implementación de un laboratorio virtual con acceso remoto para la enseñanza de redes de computadoras”; Proyecto Integrador Ing.en Computación; FCEFyN – UNC; 2012.
17. Vyatta.org Community; http://www.vyatta.org/; consultado en Mayo 2013.
18. Debian; http://www.debian.org/index.es.html; consultado en Mayo 2013.