Congreso Argentino en Ciencias de la Computacion -CACiC 2006

Adaptando OpenCA para implementar una PKI para e-Science

Lic. Francisco Javier Díaz
C.C. Viviana Miriam Ambrosi (*)
Lic. Miguel Angel Luengo
Lic. Nicolás Macia
Ms. Lía Molinari
Lic. Paula Venosa
{javierd, vambrosi, mluengo, nmacia, lmolinari, pvenosa}@info.unlp.edu.ar
Calle 50 y 115 – 1er Piso – Edificio Bosque Oeste
L.I.N.T.I.. - Facultad de Informática – U.N.L.P.
(*) Profesional Principal - CICBA

Workshop: Arquitectura, Redes y Sistemas Operativos

Resumen

En el presente artículo se describe la experiencia obtenida en la implementación de una PKI, que emitirá certificados de usuario, de host y de servicio para proyectos de E-science, utilizando OpenCA.

Se detallan aquí las ventajas ofrecidas por OpenCA así como los puntos débiles que este software presenta en el momento de poner en marcha la PKI. Se presentan además otros puntos relevantes relacionados con la instalación y configuración de la misma.

Palabras claves

PKI (Public Key Infraestructure) – CA (Certification Authority) – RA (Registration Authority) - certificados X509- Opensource - etokens

Introducción

Hoy en día existen 69 iniciativas de PKI[1] que ofrecen acceso a diferentes aplicaciones GRID para E-science, cumpliendo con las políticas establecidas por el IGTF[2]. Estas aplicaciones promueven adelantos en investigación a partir de la posibilidad de compartir recursos globalmente. En estos casos es importante fortalecer los mecanismos de seguridad subyacentes a fin de garantizar que los mismos sean usados únicamente por grupos autorizados y de una manera adecuada. En este sentido, son muchos los avances que la tecnología ha ido realizando, entre ellos el uso de los certificados X.509 [1] y las infraestructuras PKI, que soportan su creación y uso.

PKI abarca tecnologías de seguridad y políticas a través del uso de la criptografía y los estándares para proveer una infraestructura que brinda servicios de autenticación, firma digital y encripción. Las autoridades de certificación son el "corazón" de las PKI. Las mismas se encargan de emitir certificados firmados para los usuarios, hosts y servicios los cuales tienen como fin establecer una correspondencia entre la identidad del poseedor del certificado con su clave pública. [2]

La definición de políticas que dan lugar a los mecanismos que se implementarán es un capítulo imprescindible para proyectos de estas características, no sólo porque guían las tareas de implementación y generan confianza en el usuario, sino que dan las pautas para las tareas de auditoría que se recomiendan hacer con posterioridad.

Cada Autoridad de Certificación, de ahora en adelante CA, posee su CP/CPS (Certificate Policy and the Certification Practice Statement) definido en la RFC 3647. El CP/CPS define un conjunto de reglas y prácticas que la CA debe respetar en el manejo de certificados.

Al construir una CA, la misma debe brindar la funcionalidad que se detalla en el CP/CPS, por ello resulta de vital importancia la evaluación y adaptación del software a utilizar.

¿Qué es OpenCA?

OpenCA es un proyecto opensource que implementa una Autoridad de Certificación robusta, utilizando los servicios provistos por otras aplicaciones opensource como OpenLDAP, OpenSSL, Apache y Apache mod_ssl.

Este producto se ha utilizado para poner en marcha la CA raíz de una Infraestructura de clave pública en el CeSPI-UNLP para proyectos Grid de E-science [3] y se ha adaptado para cumplir con las reglas definidas en el CP/CPS de la PKI a implementar, el cual cumple con las pautas mínimas definidas por EugridPMA. [4]

Desde el nacimiento del Proyecto OpenCA en el año 1998, éste ha evolucionado notablemente, siguiendo siempre estos principios básicos:

- Adhesión a los estándares IETF[3].

[1] PKI: Public Key Infraestructure [http://www.pki.gov.ar]
[2] IGTF: International Grid Trust Federation [http://www.gridpma.org/]
[3] IETF: Internet Engineering Task Force [http://www.ietf.org]

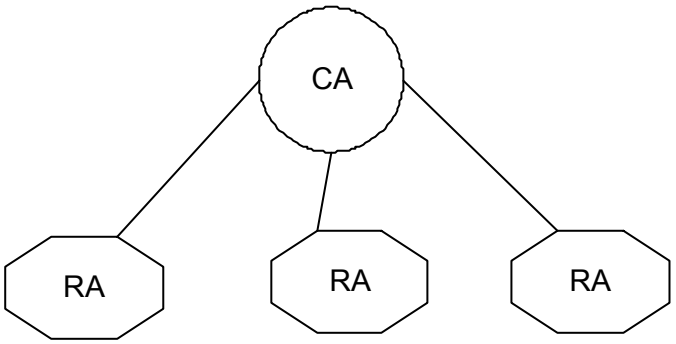
- Evolución en base al feedback dado por los usuarios y desarrolladores.
- Disponibilidad del código manteniéndolo lo más "abierto" posible para que toda la comunidad colabore, tanto a través de su opinión como aportando posibles modificaciones al mismo.
- Interoperabilidad, ya sea por adherirse a los estándares así como para ajustarse a distintas plataformas y ambientes (Por ej: inclusión de Applets de JAVA para soportar Microsoft Internet Explorer, soporte para SCEP, entre otros).
- Uso de lenguajes de programación simples en la medida de lo posible, basándose en el principio de que la seguridad no está basada en la oscuridad, de modo tal que el código sea fácilmente legible (PERL fue elegido por su simplicidad y portabilidad).

La versión utilizada en la experiencia aquí descripta es OpenCA 0.9.2.5, que es la última versión estable que se encuentra disponible hoy en día, en el sitio del Proyecto[5].

Implementación de la CA

Existen diferentes formas de "organizar" la confianza cuando se crea una PKI. Los modelos principales de confianza que existen son: modelo de confianza jerárquico,modelo de confianza de "certificación cruzada" y modelo de confianza "Bridge CA" [6].

El modelo que se ha implementado ha sido un modelo jerárquico con una única CA, que es la CA raíz, bajo la cual existirá un conjunto de RAs subordinadas. Inicialmente el modelo implementado cuenta con una única RA con la cual se comenzará a trabajar, extendiéndose en un futuro al modelo mencionado, que se representa en el siguiente gráfico:



OpenCA provee los siguientes componentes [7]:

- CA, que incluye la funcionalidad completa de una Autoridad de Certificación
- RA, que incluye la funcionalidad completa de una Autoridad de Registración
- PUB, que incluye la funcionalidad relacionada con la interacción de los usuarios y la PKI, permitiendo a los mismos requerir certificados, consultar los certificados válidos, solicitar revocación de sus certificados, consultar información pública como el CP/CPS, entre otros, a través de una interfaz WEB.

A fin de proveer un alto nivel de seguridad, la CA fue instalada en una máquina off-line, completamente aislada de la red, cumpliendo con lo establecido en la CP/CPS definida para nuestra PKI. Por otra parte, la RA y el Sitio público residen en otra máquina conectada a la red, que solamente brinda esos servicios (se han instalado y habilitado únicamente los paquetes necesarios para que la RA y el Sitio público funcionen) y que se encuentra conectada a una DMZ[4] altamente monitoreada.

Instalación de la CA

Desarrollar un sistema por capas contribuye a la seguridad del producto final. Siguiendo este principio se tuvo como eje garantizar la seguridad en cada una de las etapas de instalación de la CA.

La CA se instaló en una máquina dedicada donde solamente correrán los servicios necesarios para su funcionamiento. En dicha máquina se ha instalado una distribución de GNU Linux, siguiendo un procedimiento de instalación seguro plausible de ser auditado.

Como dicho servidor no cuenta con un módulo HSM[5] (Hardware Security Module), el mismo trabajará en forma offline.

La versión de OpenCA instalada provee una opción específica para instalar la CA offline, a través de la ejecución del comando *make install-offline*.

Una característica interesante que provee esta versión de OpenCA es la flexibilidad que brinda para establecer la configuración de la CA. Durante la instalación se crean en algunos directorios templates para los archivos de configuración. Estos templates poseen algunas referencias a valores que se encuentran definidos en el archivo config.xml.

El último paso de la instalación de la CA consiste en armar el archivo config.xml completando el mismo con los valores necesarios y ejecutar el script configure_etc.sh. El configure_etc.sh carga el config.xml y utiliza los valores allí establecidos para crear los archivos de configuración definitivos a partir de los archivos *.templates.

Este procedimiento de configuración anteriormente descripto también hace que la instalación del módulo pueda replicarse fácilmente.

Configuración de la CA

En cuanto a la operación de los módulos:

A fin de lograr seguridad y "modularización" de las tareas de administración se definen roles, los cuales se describen en el manual de procedimiento de la PKI.

OpenCA permite identificar roles, módulos y funciones, de manera tal que cada rol tenga claramente establecidos los permisos de acceso que determinan en qué módulo puede actuar un operador y cuáles son las tareas que puede realizar en el mismo.

Al emitir un certificado, OpenCA permite seleccionar un rol asociado al mismo. Esa información no forma parte del certificado pero si se incluye como información interna para OpenCA.

---

4 DMZ: Zona desmilitarizada de una red. [http://es.wikipedia.org/wiki/DMZ]

5 HSM es un dispositivo criptográfico basado en hardware que almacena y protege claves. [http://es.wikipedia.org/wiki/HSM]

Hay distintos niveles de seguridad para controlar el acceso a cada interfaz de administración de un módulo (CA, RA, etc). La configuración del control de acceso está completamente basada en XML [7].

El control de acceso consiste en:

- Verificación del canal: se encarga de chequear los parámetros de una conexión entrante para detectar clientes obsoletos o mal configurados.
- Login: establece el modo de loguearse a la interfaz de OpenCA, que puede ser sin mecanismo alguno, a través del mecanismo de usuario/clave o a través del uso de certificados.
- Manejo de sesión: establece algunas opciones relacionadas con el manejo de sesiones CGI.
- ACLs: permite definir las reglas de control de acceso a la interfaz de cada módulo, especificando si se usará una lista de control de acceso, cómo se utilizará el certificado de la CA con el que se verificarán los certificados usados para autenticarse, si se habilitará el mapeo de certificados con sus correspondientes roles, si se habilitará el mapeo de nombres de scripts con operaciones.

Además de ser posible definir que para autenticarse en un módulo se requieren certificados (lo cual se define en la sección login), se puede establecer qué roles tienen permiso a ese módulo y a qué operaciones (lo cual se define en la sección acl_config).

En nuestro caso particular se ha definido que sólo un usuario que presenta un certificado con rol de CA Operator puede acceder a la interfaz de administración de la CA y que sólo un usuario que presenta un certificado con rol de RA Operator puede acceder a la interfaz de administración de la RA.

A fin de resguardar las claves de los operadores, se decidió almacenar las mismas en dispositivos criptográficos que las protejan de usos indebidos. Esto es posible puesto que tales dispositivos, luego de validar al poseedor del mismo, permiten realizar operaciones de firma/verificación/encriptación a través de una API. La seguridad está dada porque la clave privada nunca sale del dispositivo, y ante un intento de fuerza bruta sobre el mismo, éste se bloquea sin posibilidad de recuperar los datos allí almacenados.

Como dispositivos criptográficos se utilizaron los provistos por la empresa Aladdin [8], llamados etokens pro de 32K. Los mismos permiten operaciones de 1024 bits por lo cual es posible almacenar claves como máximo de 1024 bits. Se los probó con éxito tanto en Windows como en Linux, utilizando dirvers abiertos y propietarios, y testeando la posibilidad de usar un etoken en Windows y en Linux indistintamente

En particular, la utilización de estos dispositivos brinda movilidad al operador de la RA, que posee el etoken, ya que el mismo puede administrar la RA desde cualquier host, siempre y cuando las políticas del firewall de la red lo permitan.

En cuanto a los certificados que la CA emite:

En el CP/CPS se define el formato de los certificados que emitirá la CA, lo cual significa que se definen allí los campos que se incluirán como extensiones en los mismos. En el CP/CPS se establecen también otras reglas que se deben tenerse en cuenta para definir el valor que tendrán alguno de los campos del certificado (por ejemplo la URL donde está publicado el CP/CPS) [9].

Para realizar la definición de perfiles de certificados (valores de los campos y extensiones) en OpenCA hubo que trabajar sobre:

- Adaptar el contenido del directorio OpenCADIR[6]/etc/openssl/openssl, de manera tal que para cada perfil de certificados a emitir exista un archivo "perfil".conf, en el cual se definan los valores de los campos obligatorios del certificado X509. Estos archivos se usan para armar los requerimientos en el módulo de la RA.
- Adaptar el contenido del directorio OPENCADIR/etc/openssl/extfiles, de manera tal que para cada perfil de certificados a emitir exista un archivo "perfil".ext, en el cual se definan los campos que se incluirán como extensiones y sus valores. Estos archivos se usan para construir los certificados en el módulo de la CA.

  El hecho de poder agregar nuevos perfiles de certificados y de poder manejar sus campos es una facilidad presente en la versión usada de OpenCA, la cual no estaba incluída en versiones anteriores.

También se debió adaptar la configuración de OpenCA para que el formato del certificado de la CA estuviera acorde a lo definido en el CP/CPS.

En cuanto a la interfaz de usuario

En lo que se refiere al módulo PUB, que representa la interfaz WEB de la PKI, resultó necesario adaptar la interfaz del mismo así como la funcionalidad para proveer los servicios descriptos en el CP/CPS definido. Para ello se adaptó la configuración y se modificaron scripts escritos en PERL a través de los cuales se implementan las operaciones de la PKI.

Todos aquellos puntos del CP/CPS a implementar que no se cubran desde la funcionalidad del producto pueden implementarse a través de procedimientos.

En cuanto a la comunicación de la CA con los demás módulos (RA y Sitio Público)

Es necesario que la CA y sus RAs se comuniquen a través de algún mecanismo para poder llevar a cabo la emisión de certificados: la RA debe aprobar los requerimientos, posteriormente a la verificación de identidad, y luego enviar dichos requerimientos aprobados a la CA para que la misma emita los certificados, luego de lo cual la CA debe enviar los certificados al sitio público de la PKI para que están disponibles, tanto para el usuario que los requirió como para la comunidad entera.

Al estar la CA offline por razones de seguridad, resulta de vital importancia establecer la forma en que va a comunicarse con las RAs subordinadas.

OpenCA permite definir los puntos de intercambio de datos entre la CA y las RAs. Ello se realiza a través de los nodos. El nodo es la interfaz de OpenCA que se comunica con la Base de Datos y que se encarga de implementar las operaciones de importanción y exportación.

En nuestra PKI se configuraron diferentes reglas para la sincronización de los nodos de los diferentes niveles de la jerarquía (nodo de la CA y nodo de la RA e interfaz pública), de manera tal de posibilitar el intercambio de datos entre los componentes a través de un medio off-line.

[6] OPENCADIR: nos referimos así al Directorio base donde se encuentra instalado OpenCA.

Conclusiones

Por su naturaleza opensource, OpenCA provee facilidades para modificar su funcionalidad de acuerdo a los requerimientos de cada PKI. Asimismo, como hemos mencionado anteriormente el proceso de configuración puede personalizarse gracias a la existencia del script configure_etc.sh que se encarga de armar los archivos de configuración en base a los templates y a las variables definidas en el archivo configure.xml. Este mecanismo permite armar una instalación que puede replicarse fácilmente.

Otro punto a tener en cuenta es la posibilidad de configurar OpenCA para restringir el acceso a la interfaz de administración de cada módulo (CA y RA) de manera tal que sólo la persona que presenta un certificado con el rol de operador de dicho módulo pueda acceder al mismo y realizar únicamente las operaciones que le son permitidas.

Vale destacar que hubo una serie de dificultades que se fueron resolviendo durante la experiencia aquí descripta. Una de ellas es que no todo es adaptable a partir de la modificación de los archivos de configuración sino que ante algunos requerimientos se debió recurrir a modificar el código de los scripts que implementan las operaciones de la PKI y su interfaz.

Además el proceso de personalización de los perfiles de los certificados implicó entender cuáles son los archivos de configuración y los parámetros de los mismos utilizados por la RA para armar los requerimientos y cuáles son los archivos de configuración y los parámetros de los mismos utilizados por la CA.

A lo descripto en el párrafo anterior se le debe sumar la adaptación de la interfaz Web de manera tal de permitir al usuario ingresar los datos que requiere la PKI en cuestión.

El proyecto concluyó con la implementación del sitio de la PKI de la UNLP: http://www.pkiunlprid.unlp.edu.ar .

Referencias:

[1] S. Chokani, W. Ford, R. Sabett, C. Merrill and S. Wu, " Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework" , RFC 3647, November 2003 [replaces RFC 2527] http://www.ietf.org/rfc/rfc3647.txt

[2] http://www.dartmouth.edu/~deploypki/

[3] Díaz J., Ambrosi V., Luengo M., Macia N., Molinari L., Venosa P. - Cuando la seguridad trasciende las fronteras o sobre como manejar el problema de la autenticación para el acceso internacional de recursos distribuidos. WICC 2006.

[4] David Groep " Profile for Traditional X.509 Public Key Certification Authorities with secured infraestructure version 4.0-03" http://www.eugridpma.org/igtf/IGTF-AP-classic-20050905-4-03.doc

[5] http://www.openca.org

[6] Tim Moses “PKI trust models”
http://www.it-c.dk/courses/DSK/F2003/PKI_Trust_models.pdf

[7] OpenCA Guide for Versions 0.9.2+ http://www.openca.org/openca/docs/online/

[8] http://www.aladdin.com/

[9] http://www.grid-support.ac.uk/content/view/23/55/