Aspectos de seguridad en el *provisioning* de servidores

Lanfranco, Einar
einar@info.unlp.edu.ar

Benencia, Raúl
rbenencia@linti.unlp.edu.ar

Macia, Nicolás
nmacia@linti.unlp.edu.ar

Venosa, Paula
pvenosa@linti.unlp.edu.ar

1. Resumen

En este trabajo se presentará la problemática que supone el mantenimiento de configuraciones y actualizaciones de múltiples sistemas informáticos en un ambiente descentralizado, dando en lo posible, un enfoque de solución que utilice productos de software libre [1] como premisa inicial.

Entre los alcances esperados de esta línea de I/D/I están los de analizar los distintos mecanismos y herramientas de *provisioning*, minimizando los problemas tanto de seguridad como de disponibilidad a los que se expone un sistema informático.

Entre las precondiciones que tomamos se puede mencionar que los mecanismos tienen que servir tanto para reconfigurar sistemas ya puestos en producción como para realizar nuevas instalaciones, teniendo como objetivo máquinas con sistema operativo tipo `*nix` [2] y considerando que las las herramientas a utilizar deben ser software libre.

Se incorpora en este trabajo, además del análisis de mecanismos y herramientas, el relato de la implementación realizada hasta el momento y los objetivos alcanzados.

En base al conocimiento adquirido, se espera establecer un mecanismo que permita cumplir buenas prácticas de configuración o actualización que habiliten, por ejemplo, el cambio de perfiles o anulación de los accesos de un usuario, debido tanto a un compromiso de sus credenciales de acceso como a la baja o reubicación laboral del mismo.

Palabras clave: provisioning, software libre, gestión centralizada, seguridad

2. Contexto

La línea de investigación presentada está inserta en el proyecto de incentivos Redes, Seguridad y Desarrollo de Aplicaciones para e-educación, e-salud, e-gobierno y e-inclusión"del LINTI [3] de la Facultad de Informática de la Universidad Nacional de La Plata (UNLP) [4].

En el marco de este proyecto un grupo de trabajo de docentes/investigadores del LINTI se avocan al estudio y aplicación de estándares y tecnologías vinculadas a la seguridad y privacidad en redes y aplicaciones, incluyendo en esta línea también la concientización y capacitación en temas vinculados con la seguridad de la información en el ámbito académico.

3. Introducción

Administrar un conjunto de servidores dentro de una organización implica la necesidad de mantener la consistencia entre ciertas configuraciones comunes a éstos. En este sentido, resulta frecuente la necesidad de aplicar reglas de seguridad como, por ejemplo, filtros de *firewall* que permitan el acceso a servicios sólo desde determinadas subredes; definición de usuarios junto con sus contraseñas y/o claves públicas; definición de privilegios dependiendo del servidor, a determinados usuarios; servicios que deben ejecutarse por defecto para mantener la consistencia de los *logs*, como la sincronización horaria con el servicio de NTP [5]; configuraciones de envío de logs hacia servidores de logs; etc.

El precepto de mantener la consistencia de la configuración se aplica también cuando un usuario deja de formar parte de la organización o es movido de función. Tanto su usuario como sus credenciales deben ser revocadas y deshabilitadas de los servidores donde ya no formará parte. Fallar en cumplir esta consigna puede provocar severas consecuencias en la organización puesto que se abre una ventana para una posible brecha de seguridad.

Mantener la coherencia entre la configuración de un conjunto de servidores conlleva trabajo manual donde el operador habitualmente requiere conectarse a cada dispositivo para aplicar una serie de reglas genéricas. Esta actividad, secuencial y repetitiva, es propensa a errores debido al factor humano involucrado en la tarea. Por lo tanto, surge la necesidad de encontrar una solución para automatizar estas tareas con el objetivo de optimizar el uso del tiempo de los recursos humanos y minimizar las amenazas de seguridad no intencional que implica la realización manual de las mismas.

El problema descripto, con el que conviven a diario los administradores de sistemas, actualmente se soluciona utilizando técnicas de *provisioning*. El *provisioning* consiste en establecer una serie de reglas o propiedades que un *host* debe cumplir. Luego, mediante algún mecanismo propio de la herramienta utilizada, estas propiedades se transmiten al dispositivo pertinente y se asegura su subordinación a las reglas establecidas. De esta forma se puede asegurar que la totalidad de servidores de una organización se rige bajo un único conjunto de reglas de configuración.

El *provisioning* también favorece el ciclo de mejora contínua de los procesos de administración y del sistema de gestión de seguridad de la organización, dado que resulta más sencillo el aplicar cambios requeridos en el marco del PDCA [6].

En toda organización es fundamental mantener algún esquema de usuarios y permisos. Dicho esquema facilita la separación de privilegios y permite controlar de forma granular el acceso a los recursos compartidos. La carencia de un sistema de este tipo provoca que todos los miembros de la organización mantengan el mismo nivel de privilegios, lo cual claramente genera un grave riesgo de seguridad.

En sistemas operativos tipo `*nix` el esquema de separación de privilegios se basa en usuarios, grupos y permisos de lectura, escritura y ejecución. Cuando la organización o el grupo de administradores es pequeño, es normal mantener un único usuario privilegiado con una contraseña maestra compartida. Indudablemente esto es una mala práctica puesto que si uno de los administradores deja el grupo, habrá que cambiar la contraseña maestra en todas las máquinas afectadas. Además, utilizar un esquema de este estilo no permite mantener la trazabilidad de las acciones efectuadas por cada administrador.

Es entonces esencial mantener una relación unidireccional entre usuarios y miembros de la organización. Inevitablemente esta actividad conlleva un mayor esfuerzo de administración, puesto que por cada servidor se debería configurar cada usuario junto con su contraseña. Por este motivo, el mencionado problema habitualmente se resuelve utilizando un servidor centralizado de autenticación, o algún árbol de directorios como LDAP [7] en conjunto con algún esquema de permisos como PAM [8].

Estas soluciones, que son aceptables desde el punto de vista de la seguridad de la información, siempre y cuando estén correctamente implementadas, poseen dos problemas que impulsan a buscar otras alternativas. En primer lugar, el grupo de trabajo que lleva a cabo esa investigación utiliza SSH [9] con autenticación por clave pública para conectarse remotamente a sus servidores. Configurar cada usuario con una clave pública no es algo que se pueda implementar de forma sencilla con un servidor centralizado de autenticación. Tampoco existe un método directo para hacerlo utilizando servicios como LDAP. En segundo lugar, es preferencia de nuestro grupo de trabajo no mantener servicios que impliquen un único punto de falla. Si por algún motivo el hipotético servidor de autenticación centralizada no estuviera disponible, entonces el resto de la infraestructura se vería inutilizada puesto que no sería posible autenticarse, provocando efectivamente una denegación de servicio [10] no solo al servicio comprometido sino a la totalidad de los servicios de la organización.

Ante las decisiones de diseño tomadas, entonces, se optó por seguir una metodología de *provisioning* de usuarios y credenciales. De esta forma, los usuarios y claves públicas se configuran en un único punto, pero luego esta configuración es distribuida e instalada en cada servidor involucrado. De esta forma se obtienen los beneficios de un servicio de autenticación centralizada, pero se elimina el problema de que sea un único punto de falla.

Existen diversas herramientas licenciadas como software libre para realizar *provisioning*. En particular, se evualuaron aquellas que funcionan desde y para sistemas operativos tipo `*nix`, siendo requerimiento fundamental que tengan buen soporte para aquellas distribuciones basadas en Debian GNU/Linux como ser el propio Debian [11], Ubun-

tu [12] o Lihuen GNU/Linux [13], la distribución de la Facultad de Informática. Las herramientas más destacadas que cumplen con estos requisitos son Chef [14], Puppet [15] y Ansible [16]. Sin embargo, para efectuar las experiencias realizadas por este grupo de trabajo se decidió utilizar Propellor [17], una herramienta para hacer *provisioning* que aún está dando sus primeros pasos, pero no por ello es menos efectiva.

El paradigma de distribución de configuraciones de Propellor es interesante de analizar y merece una breve descripción en este artículo. Para empezar, Propellor es una librería escrita utilizando Haskell [18], un lenguaje de programación funcional puro. El usuario de Propellor debe escribir su programa donde cada *host* sobre el que se requiera hacer *provisioning* es representado por una lista de propiedades. Este aspecto no es muy distinto a Chef con sus recetas, Puppet con sus módulos o Ansible con sus *playbooks*. Tanto el programa escrito como la librería se versionan en un repositorio `git` [19]. Cuando se desea hacer *deploy* de las configuraciones se disponen de dos alternativas. En primer lugar se puede realizar una conexión directa vía SSH al *host* involucrado, transmitiendo de esta forma los cambios o actualizaciones de la configuración vía un *commit* en el repositorio `git`. La otra alternativa consiste en que los mismos *hosts* sobre los que se hace *provisioning* intenten cada cierto tiempo descargar nuevos *commits* de un repositorio `git` centralizado con las configuraciones. Cabe destacar que, si se elige esta alternativa, los *commits* del repositorio deberán estar firmados con una clave GPG [20] previamente configurada para tal fin, con el objetivo de asegurar la integridad y autenticidad de los datos.

4. Líneas de Investigación, Desarrollo e Innovación

Este artículo presenta la línea de trabajo que está siguiendo el grupo de seguridad del LINTI respecto al *provisioning* de configuraciones con el objetivo de garantizar la seguridad e integridad de los datos administrados.

Sobre los ejes de investigación inicialmente planteados se realizaron evaluaciones comparaciones sobre distintas herramientas de *provisioning* de configuraciones. Se designó una de ellas, Propellor, como herramienta clave para realizar y escribir configuraciones genéricas de servicios de sistemas operativos de tipo `*nix`.

Se analizó el funcionamiento de la mencionada herramienta desde un punto de vista de la seguridad de la información y su forma de hacer *deploy* de configuraciones.

5. Resultados y Objetivos

Como resultados de la línea de trabajo presentada en este artículo se logró, mediante *provisioning*, automatizar la creación e instalación de credenciales en todos los servidores involucrados. También se automatizó la configuración de servicios básicos como NTP, instalación de actualizaciones de seguridad de forma automática y configuración de un MTA (*Mail Transfer Agent*) funcional. Cabe destacar que cada una de estas configuraciones es genérica, motivo por lo cual se puede aplicar directamente en un servidor recién instalado sin necesidad de realizar posteriores configuraciones manuales.

A partir de los resultados obtenidos se trabajará en seguir haciendo *provisioning* sobre distintos aspectos de la configuración de un servidor `*nix`. Se tiene pensado trabajar sobre configuraciones genéricas de agentes SNMP, *backups* de la información administrada, automatización de la centralización de *logs* del sistema operativo, entre otros aspectos de la administración de sistemas. Adicionalmente se trabajará en la integración del *provisioning* de sistemas operativos con el gestor de máquinas virtuales con el objetivo de generar de forma automatizada servidores preparados para el ámbito de trabajo.

6. Formación de Recursos Humanos

La línea de investigación en aspectos de seguridad en el *provisioning* de servidores esta siendo abordada en conjunto por los docentes Nicolás Macia, Paula Venosa, Einar Lanfranco y Raúl Benencia quienes también forman parte del grupo de seguridad del LINTI de la Facultad de Informática de la UNLP, el CERT.unlp [21] y las cátedras de grado y postgrado Seguridad y privacidad en redes. La temática a su vez esta muy relacionada con la tesis en desarrollo para obtener el título de Magister en Redes

de Datos, denominada: “Optimización del control y administración de la seguridad en una red de servidores. Su implementación como software libre para contribuir con MatFel” la cuál esta siendo desarrollada por Einar Lanfranco.

El grupo de seguridad del LINTI de la Facultad de Informática de la UNLP trabaja hace varios años realizando varias experiencias tanto en Seguridad de la Información como en desarrollo e implementación en Software Libre, ambas experiencias dirigidas a distintos perfiles.

Por otra parte, este grupo de investigadores representa a la UNLP en el Centro de excelencia en el tema “Ciberseguridad” de la UIT, durante el transcurso del año 2015 [22] donde el mismo comenzó a participar en el año 2014 de la comisión de estudio ITU-T SG17:Security de la UIT [23]

Bibliografía

[1] ¿Qué es el software libre? - Proyecto GNU - Free Software Foundation. http://www.gnu.org/philosophy/free-sw.html.

[2] Category:Unix - Wikimedia Commons. http://commons.wikimedia.org/wiki/Category:Unix.

[3] LINTI. http://www.linti.unlp.edu.ar/linti.

[4] Facultad de Informática - UNLP. http://www.info.unlp.edu.ar/.

[5] ntp.org: Home of the Network Time Protocol. http://www.ntp.org/.

[6] Ciclo PDCA (Planificar, Hacer, Verificar y Actuar): El círculo de Deming de mejora continua | PDCA Home. http://www.pdcahome.com/5202/ciclo-pdca/.

[7] K. Zeilenga. Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map. RFC 4510 (Proposed Standard), June 2006.

[8] PAM (Pluggable Authentication Modules). http://tldp.org/HOWTO/User-Authentication-HOWTO/x115.html.

[9] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Authentication Protocol. RFC 4252 (Proposed Standard), January 2006.

[10] Understanding Denial-of-Service Attacks | US-CERT. https://www.us-cert.gov/ncas/tips/ST04-015.

[11] Debian – el sistema operativo universal. http://www.debian.org/.

[12] The leading OS for PC, tablet, phone and cloud | Ubuntu. http://www.ubuntu.com/.

[13] Sitio oficial de Lihuen. http://lihuen.linti.unlp.edu.ar/index.php.

[14] Chef | IT automation for speed and awesomeness | Chef. https://www.chef.io/chef/.

[15] Puppet Labs: IT Automation Software for System Administrators. https://puppetlabs.com/.

[16] Ansible is Simple IT Automation. http://www.ansible.com/home.

[17] Propellor | deploying properties to hosts with haskell. https://propellor.branchable.com/.

[18] Haskell Language. https://www.haskell.org/.

[19] Git. http://git-scm.com/.

[20] Werner Koch. Gnupg: Gnu privacy guard, 1998.

[21] CeSPI - UNLP - CERT - Seguridad informática. http://www.cespi.unlp.edu.ar/cert.

[22] UIT: Comprometida para conectar el mundo. http://www.itu.int/es/Pages/default.aspx.

[23] ITU-T SG17: Security. http://www.itu.int/en/ITU-T/studygroups/2013-2016/17/Pages/default.aspx.