Implementación de un sistema de monitoreo para obtener estadísticas del estado en un servidor FreeRADIUS utilizando SNMPv3

Implementation of a Monitoring system to gather status statistics on a FreeRADIUS server using SNMPv3

Andres Mijail Leiva Cochachin, Javier Richard Quinto Ancieta, José Luis Quiroz Arroyo

INICTEL-UNI, San Borja - Lima 41 – Perú

RESUMEN

Hoy en día, el desarrollo constante de nuevas herramientas informáticas ha hecho posible el mejoramiento de los mecanismos de seguridad y el descubrimiento de muchas de sus vulnerabilidades; lo que obliga a los administradores de red a mejorar sus sistemas continuamente. Por tal motivo, varias organizaciones están implementando modelos con soporte IEEE 802.1X [1] con el objetivo de mejorar la seguridad en cuanto a la autenticación y el registro de los usuarios que acceden diariamente a su red. Para minimizar los problemas en seguridad, existen mecanismos que utilizan un sistema de autenticación en capa 2 del modelo de referencia OSI y basado en el modelo AAA [2], el cual es un estándar implementado en todos los sistemas RADIUS [3] tal como el sistema FreeRADIUS basado en software libre. A su vez, este sistema debería tener un mecanismo para el control de los diversos eventos que ocurren durante los procesos AAA, sin embargo, la versión actual de FreeRADIUS no cuenta con algún soporte de protocolo de gestión como SNMP [4] para que pueda ser monitoreado de manera remota con la integridad y confidencialidad garantizada. En este articulo se presenta la implementación de un sistema de monitoreo que permite obtener reportes de datos estadísticos de un servidor de autenticación RADIUS en forma segura, utilizando SNMPv3 [5] como protocolo de gestión y FreeRADIUS como Servidor RADIUS. Para ello se realizó consultas a los atributos [6] que registran el conteo de los eventos de autenticación en FreeRADIUS. Dichas cifras son almacenadas temporalmente en una Base de Información de Administración (MIB) [7] que un servidor de monitoreo consulta en un intervalo de tiempo constante, utilizando el protocolo SNMPv3. Los datos recogidos, son almacenados en una base de datos MySQL [8] y presentados en gráficos en series de tiempo, utilizando la herramienta RRDTool [9] en una interfaz web. Para validar el funcionamiento, se realizaron pruebas en un escenario LAN con un cliente RADIUS, un servidor FreeRADIUS, 2 usuarios y un servidor de monitoreo. Se hicieron varios intentos de autenticación con usuarios registrados y no registrados para obtener cifras significativas del conteo de eventos durante los procesos de autenticación. La implementación se llevó a cabo utilizando software libre, tanto para el servidor FreeRADIUS como para el servidor de monitoreo. Además, se consideró el monitoreo de todos los atributos disponibles en FreeRADIUS, a diferencia de otros sistemas propietarios que sólo muestran información limitada [10].

Descriptores: *Autenticación de usuarios, FreeRADIUS, Monitoreo, RRDTool, SNMPv3.*

INTRODUCCIÓN

La seguridad es un aspecto de suma importancia en cualquier infraestructura de red. Una buena implementación de equipos y políticas podrían reducir considerablemente el riesgo de sufrir ataques por algún malware. Actualmente existen diversas redes que cuentan con mecanismos eficientes de seguridad que, en cierta medida, les protege de ataques externos por parte de usuarios no autorizados. Una práctica eficiente para garantizar la seguridad en el acceso de usuarios a una red, consiste en utilizar un sistema de autenticación, autorización y contabilidad controlado por servidores dedicados cuya función principal es otorgar permisos a usuarios confiables utilizando cuentas de acceso propias de cada uno. Una solución a este problema es el uso de un servidor AAA (Authentication, Authorization and Accounting) basado en el protocolo RADIUS. Una implementación de software de éste protocolo es FreeRADIUS, cuyo rendimiento es uno de los mejores y es altamente configurable ya que soporta distintos tipos de protocolos de autenticación y acceso a diferentes tipos de bases de datos. FreeRADIUS cuenta con 2 versiones, con características similares, sin embargo, a pesar de que en la segunda versión podemos disfrutar de mayores ventajas, ésta versión no cuenta con soporte para activar un agente SNMP en el servidor y poder ser monitoreado de manera remota. Esto induce a utilizar otros mecanismos sencillos y poco seguros para poder obtener información del servidor, tal como realizar consultas externas o crear una secuencia de comandos adecuados. Utilizando estos mecanismos se puede extraer la información y obtener resultados fiables, no obstante, tienen la desventaja de ser soluciones con un método específico y poco seguro y su despliegue en otros sistemas es un tanto complicado ya que no incluye un protocolo estandarizado.

Sin embargo, contar con dicha herramienta no es suficiente para garantizar la seguridad ya que en muchas implementaciones seguras siempre ha existido la necesidad de poder contar con información relacionada a los eventos que reporta nuestra herramienta, especialmente si esta herramienta es de seguridad. Implementar un sistema de monitoreo de este tipo no solamente es necesario, también nos permite administrar registros que facilitan la generación de reportes sobre el estado de los servidores en el tiempo.

En el presente artículo se presenta una propuesta de implementación, rentable y eficiente, de un sistema de monitoreo de los eventos ocurridos entre un servidor RADIUS y sus clientes. La metodología de implementación consiste en extraer información de eventos de un servidor RADIUS utilizando el protocolo simple de gestión de red (SNMP) en versión 3, el cual garantiza la autenticación y privacidad de los datos extraídos y facilita su administración a nivel de usuario. Esta información es recopilada y almacenada utilizando un servidor de monitoreo que se encarga de gestionar los eventos de un servidor RADIUS en forma centralizada. La información es almacenada en una base de datos MySQL y es presentada en una interfaz web mediante gráficos en series temporales sintetizados con RRDTool. Las pruebas del escenario implementado se realizaron utilizando un servidor FreeRADIUS, un servidor de monitoreo, un Access Point como cliente RADIUS y una computadora portátil como un usuario.

RADIUS y Sistema de Gestión de Acceso usando AAA

Proceso AAA usando paquetes RADIUS

El modelo AAA fue desarrollado después de la creación del RADIUS con la finalidad de crear un estándar en los métodos de autenticación y autorización para la validación de los usuarios a la red. Los servidores RADIUS utilizan el modelo AAA como framework, existiendo otros protocolos que usan este modelo, por ejemplo Tacacs de Cisco [6].

El uso más general del AAA es en escenarios de autenticación en donde es necesario centralizar las consultas de acceso y llevar la cuenta de los eventos en un solo servidor y poder ser administrado por éste.

El modelo AAA se basa en 3 aspectos fundamentales: autenticación, autorización y contabilidad. La autenticación es el proceso en

donde el usuario envía sus credenciales a un cliente RADIUS para su posterior acceso a la red. Dichas credenciales podrían ser de diversos tipos (usuario y clave, certificados digitales) usando algoritmos de cifrado (PAP, CHAP) o un protocolo extensible de autenticación (EAP). También indica el protocolo a usar. La autorización define el tipo de acceso a la red a la cual el usuario debe conectarse, para ello se definen una serie de políticas para los usuarios [10]. La contabilidad es más que tener un estado de cuentas de cada usuario, existe una variedad de aplicaciones que también lo definen como por ejemplo: auditoria, asignación de costos y los análisis de tendencias [2].

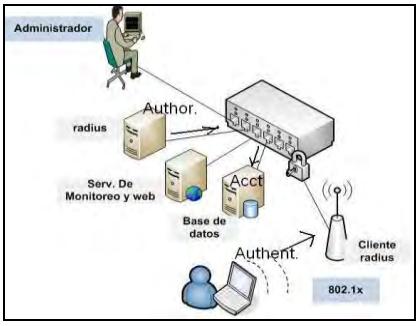

Figura 1: Atributos y Seguridad usando RADIUS.

Los tipos de paquetes enviados en un proceso AAA utilizando RADIUS contienen valores predefinidos para diversos eventos de autenticación y contabilidad. Cada valor predefinido contiene una serie de atributos en un par Valor-Atributo (AVPs) desde un cliente a un servidor RADIUS. Los atributos limitan que tipo de paquete debe ser enviado en el proceso de autenticación y contabilidad.

En el proceso de autenticación, existen 10 tipos de paquetes RADIUS que son fundamentales (RFC 2865) entre los más usados son Access-request, Access-challenge, Access-accept, Access-reject. Para el proceso de contabilidad existen solo 7 tipos de paquetes RADIUS que son fundamentales (RFC 2866) entre los más usados son Accounting-request, Accounting-response, estos 17 tipos de paquetes se diferencian por un código en el formato del paquete RADIUS.

La RFC 1865 muestra una tabla del total de atributos para el proceso de autenticación y un valor aleatorio de 1 byte en el campo identificador por cada tipo de paquete, éste último otorga seguridad a un paquete RADIUS dificultando algún tipo de ataque por clonación de paquetes. Para esto existe dos tipos de paquete llamado Auth-Duplicate-Requests y Acct-Duplicate-Requests que detecta cualquier intento de duplicidad de paquete mandando mensajes de alerta al RADIUS usando un puerto definido por el administrador, también existen los paquetes RADIUS mal formados que se originan debido a un incorrecto shared-secret, así también existen los paquetes inválidos y los paquetes rechazados tanto para la autenticación y contabilidad en un proceso AAA.

Para mejorar la seguridad, el atributo User-Password solo es enviado en el tipo de paquete Access-request y su valor es cifrado durante la autenticación usando Hash MD5 [11], su longitud esta en el rango de 18 a 130 bytes del tipo String (cadena de texto).

Otra medida de seguridad es el máximo número de atributos permitidos en un paquete RADIUS, si éste valor es muy pequeño entonces los paquetes RADIUS no serán aceptados, si éste valor es muy grande entonces facilita al atacante para poder enviar más atributos de lo permitido siendo esto una vulnerabilidad en el paquete RADIUS. El valor máximo de atributos permitido en un paquete RADIUS es 200. Otro problema de seguridad es el valor reject_delay, lo que significa un retardo en los paquetes rechazados, si éste valor es 0, entonces es vulnerable a ataques de denegación de servicio o ataques de fuerza bruta, ya que estos pueden enviar varios paquetes de tipo reject sobre cargando al servidor RADIUS, para un valor reject_delay muy grande, entonces el servidor RADIUS procesaría más tiempo dicha consulta no siendo eficaz para la seguridad en el servidor.

Estrategia de monitoreo del servidor RADIUS

Para poder obtener las estadísticas de un Servidor FreeRADIUS utilizando SNMPv3, se tuvo en cuenta mecanismos de extracción de datos de FreeRADIUS y almacenamiento de información en una MIB de Net-SNMP [12].

FreeRADIUS tiene la opción de mostrar estadísticas de autenticación y contabilidad de las consultas de usuarios hechas en una red local, así como también puede mostrar estadísticas de consultas que son reenviadas a un servidor proxy de mayor jerarquía. FreeRADIUS cuenta con un diccionario en donde podemos encontrar en total 56 atributos que registran diferentes eventos del Servidor. En las tablas 1 a 4 se muestra parte del diccionario mostrando la declaración de 34 atributos de interés para el presente trabajo.

Tabla 1: Estadísticas de autenticación globales para paquetes recibidos por el servidor.

ATTRIBUTE FreeRADIUS-Total-Access-Requests 128 integer
ATTRIBUTE FreeRADIUS-Total-Access-Accepts 129 integer
ATTRIBUTE FreeRADIUS-Total-Access-Rejects 130 integer
ATTRIBUTE FreeRADIUS-Total-Access-Challenges 131 integer
ATTRIBUTE FreeRADIUS-Total-Auth-Responses 132 integer
ATTRIBUTE FreeRADIUS-Total-Auth-Duplicate-Requests 133 integer
ATTRIBUTE FreeRADIUS-Total-Auth-Malformed-Requests 134 integer
ATTRIBUTE FreeRADIUS-Total-Auth-Invalid-Requests 135 integer
ATTRIBUTE FreeRADIUS-Total-Auth-Dropped-Requests 136 integer
ATTRIBUTE FreeRADIUS-Total-Auth-Unknown-Types 137 integer

Tabla 2: Estadísticas globales para paquetes de autenticación enviados hacia otros servidores.

ATTRIBUTE FreeRADIUS-Total-Proxy-Access-Requests 138 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Access-Accepts 139 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Access-Rejects 140 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Access-Challenges 141 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Auth-Responses 142 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Auth-Duplicate-Requests 143 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Auth-Malformed-Requests 144 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Auth-Invalid-Requests 145 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Auth-Dropped-Requests 146 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Auth-Unknown-Types 147 integer

Tabla 3: Estadísticas globales para paquetes de contabilidad enviados por el servidor.

ATTRIBUTE FreeRADIUS-Total-Accounting-Requests 148 integer
ATTRIBUTE FreeRADIUS-Total-Accounting-Responses 149 integer
ATTRIBUTE FreeRADIUS-Total-Acct-Duplicate-Requests 150 integer
ATTRIBUTE FreeRADIUS-Total-Acct-Malformed-Requests 151 integer
ATTRIBUTE FreeRADIUS-Total-Acct-Invalid-Requests 152 integer
ATTRIBUTE FreeRADIUS-Total-Acct-Dropped-Requests 153 integer
ATTRIBUTE FreeRADIUS-Total-Acct-Unknown-Types 154 integer

Tabla 4: Estadísticas globales para paquetes de contabilidad enviados hacia otros servidores.

ATTRIBUTE FreeRADIUS- Total-Proxy-Accounting-Requests 155 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Accounting-Responses 156 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Acct-Duplicate-Requests 157 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Acct-Malformed-Requests 158 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Acct-Invalid-Requests 159 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Acct-Dropped-Requests 160 integer
ATTRIBUTE FreeRADIUS-Total-Proxy-Acct-Unknown-Types 161 integer

Las estadísticas de los eventos en los servidores FreeRADIUS pueden ser consultadas desde un cliente previamente autorizado usando el tipo de paquete radclient para el envío de paquetes de consultas al servidor, para ello es necesario quitar los comentarios en los modulos “authorize” y “accounting” en las líneas relacionadas al status-server y en el archivo “default” escribir status-server=no, lo cuál hará que el servidor RADIUS ignore los paquetes Status-server para los puertos “auth” y “acct” y solo responda para aquellas consultas al puerto “status”, todo esto por razones de seguridad [7].

Status es uno de los tipos de paquetes RADIUS según RFC 2865 con valor código igual a 12. En el proceso de consulta Status-Server envía 2 atributos como mínimo siendo el atributo “Message-Authenticator” indispensable para las consultas del tipo status, el atributo Message-Authenticator con valor 0x50 y el atributo Vendor-Specific con valor 0x1a son enviados en la consulta de estadísticas de eventos al servidor RADIUS, y éste devuelve los atributos pedidos de cada uno con su respectivo valor en hexadecimal. En la figura 2 se observa la captura de un paquete del tipo de consulta al servidor RADIUS usando “status-server”.

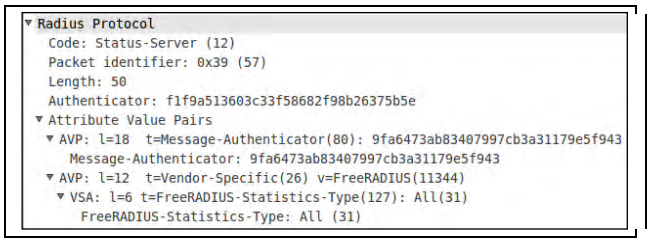

Figura 2: Estructura de un paquete de consulta de estado al servidor RADIUS.

En la figura 3 se observa la respuesta del servidor RADIUS al Tipo de consulta FreeRADIUS-Statistics-Type: All.

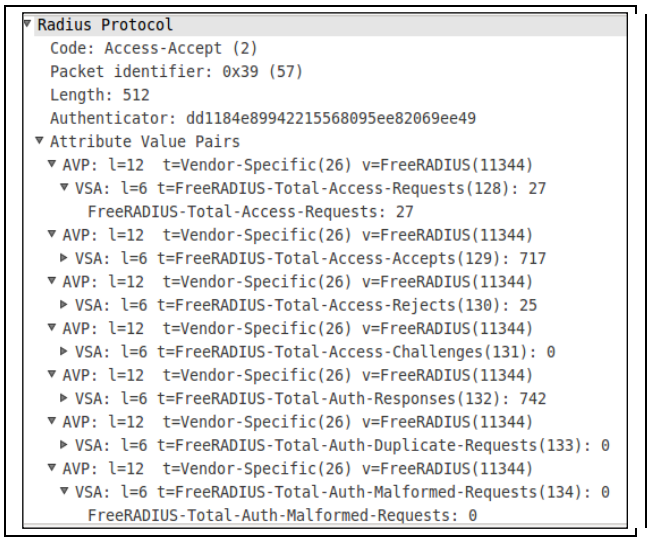

Figura 3: Estructura de un paquete de respuesta de estado al servidor RADIUS.

Mecanismo de Almacenamiento de la Información en una MIB

Los valores de los atributos de FreeRADIUS fueron almacenados dentro de una MIB extendida bajo el número 8072 de la lista de números de empresa de la IANA [13]. Según la lista, este número corresponde a Net-SNMP el cual es el programa que

implementa al agente SNMP en el servidor FreeRADIUS. La figura 4 muestra un diagrama de árbol que nos ilustra la ubicación en donde se ha almacenado la información de los eventos del FreeRADIUS en la estructura de la MIB del Servidor.

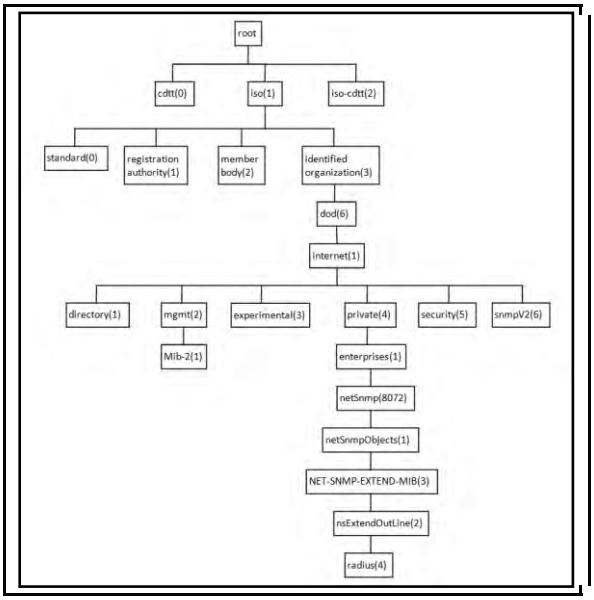

Figura 4: Localización de la rama FreeRADIUS en la MIB del Servidor.

Teniendo estos valores registrados en la MIB, el servidor de monitoreo puede realizar consultas mediante SNMPv3 hacia el servidor FreeRADIUS dirigido hacia los objetos almacenados en la MIB. Los valores del conteo de cada uno de los 34 atributos son almacenados directamente en la MIB utilizando la propiedad de extensión de la funcionalidad del agente incluida en Net-SNMP [14]. Los datos de los objetos son actualizados por un programa basado en una secuencia de comandos, el cual ha sido configurado en el agente Net-SNMP. La figura 5 muestra un esquema que representa el proceso de actualización de la MIB del agente SNMP en el servidor FreeRADIUS.

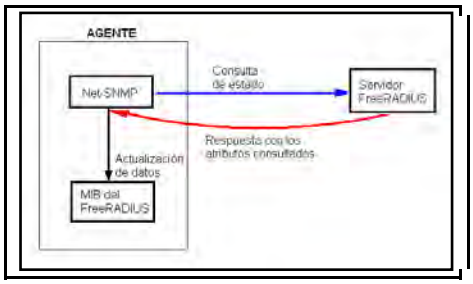

Figura 5: Proceso de actualización de la MIB del FreeRADIUS.

Mecanismo de Extracción y Actualización de datos

Para extraer la información del agente, se utilizó el programa cliente de Net-SNMP en el servidor de monitoreo. El agente (FreeRADIUS) utiliza Net-SNMP configurado para operar según modelo de seguridad USM (User-based Security Model) y el mecanismo de control de acceso VACM (View-based Access Control Mode) de SNMPv3 [5]. En este caso, el usuario que monitorea es autenticado utilizando el protocolo MD5 y la información es encriptada utilizando el algoritmo de cifrado DES. Asimismo, se ha creado una vista de acceso de sólo lectura a la MIB de FreeRADIUS para permitir el acceso al usuario autorizado que en este caso es únicamente el servidor de monitoreo.

Se ha observado que durante una consulta se genera un conteo adicional en los atributos “Total-Access-Accepts” y “Total-Auth-Responses” ya que estos atributos reportan la cantidad total de accesos que fueron aceptados y respondidas por el servidor respectivamente. Cada consulta corresponde a un acceso aceptado y a la vez, la respuesta que el servidor emite a esta consulta es una respuesta de autenticación. Esto es un inconveniente ya que cada vez que el servidor de monitoreo realiza una consulta, el valor de estos atributos aumenta una unidad y como las consultas se realizan cada cierto intervalo de tiempo, siempre nos muestra resultados aumentados en 1.

Para aliviar este inconveniente, se utilizó un programa escrito en php alojado y ejecutado en el servidor de monitoreo. Este programa se encarga de realizar la consulta a la MIB del FreeRADIUS para obtener los datos actuales de cada atributo. Una vez obtenido los datos, el programa realiza la corrección del error de adición que se genera al realizar la consulta. Para ello, resta una unidad a la diferencia del resultado actual con el resultado anterior de cada uno de los 2 atributos mencionados. Luego el valor de esta diferencia es almacenada en una base de datos MySQL [8] local para tener un pleno registro de los datos en el tiempo. Asimismo, se actualiza el archivo rrd generado con la herramienta RRDTool [9], la cual nos permite generar gráficos en series temporales. Por último, se genera una gráfica por cada atributo utilizando los datos en el archivo rrd.

En la figura 6, se ilustra el proceso general de monitoreo del Servidor FreeRADIUS representando

en forma simbólica cada parte del proceso mencionado anteriormente.

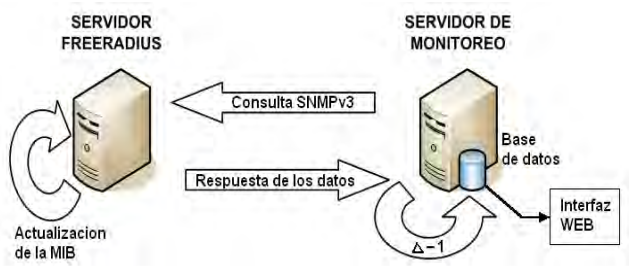

Figura 6: Proceso de monitoreo

Interfaz web de reporte con RRDTool

El servidor de monitoreo se encarga de realizar las consultas de los datos hacia el servidor FreeRADIUS cada 5 minutos para no saturar la red y para tener una ventana de observación relevante que nos permita identificar los minutos donde se producen el conteo máximo de cada atributo. Los gráficos nos permiten, por ejemplo, identificar el intervalo de tiempo donde se produce la cantidad máxima de accesos rechazados por el servidor, esta cantidad es importante para un administrador de red ya que le puede ayudar a reconocer el posible ataque de algún usuario.

Para facilitar el acceso a la información gráfica, se ha implementado un servidor web Apache en el servidor de monitoreo. La página web presenta los gráficos de cada atributo por separado. Asimismo, se presentan 6 gráficas adicionales donde se apilan las gráficas de atributos clasificados según accesos, autenticaciones, y estado de cuentas tanto locales como externas. RRDTool nos permite apilar gráficas y mostrarlas en un intervalo de tiempo escogido arbitrariamente.

Estos gráficos nos permiten realizar comparaciones visuales de cada atributo. La figura 7 muestra el entorno de la interfaz web con los 6 gráficos apilados.

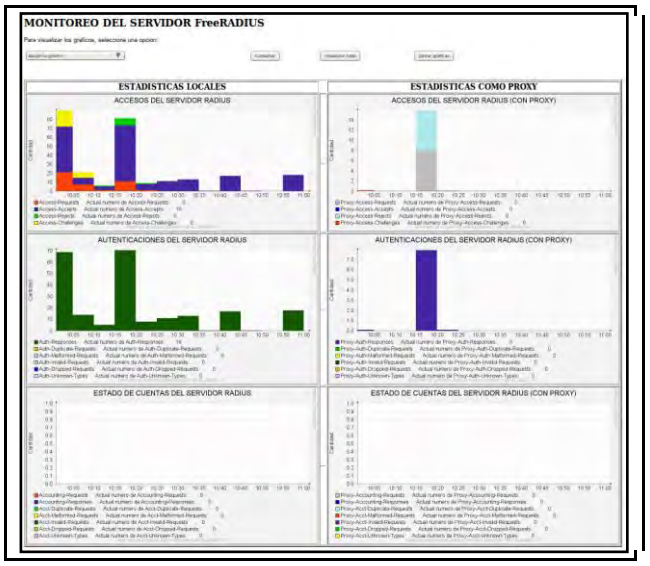

Figura 7: Interfaz web de reporte.

La figura 8 muestra uno de los 6 gráficos apilados. En este caso se observa los gráficos de accesos del servidor mostrados uno encima de otro mostrados en diferentes colores. Esta representación permite realizar comparaciones visuales en una ventana de tiempo de 5 minutos. Aquí se puede observar que existen accesos rechazados sólo en algunos intervalos mientras que existen accesos aceptados en casi todos los intervalos. Asimismo se observa inactividad en algunos intervalos.

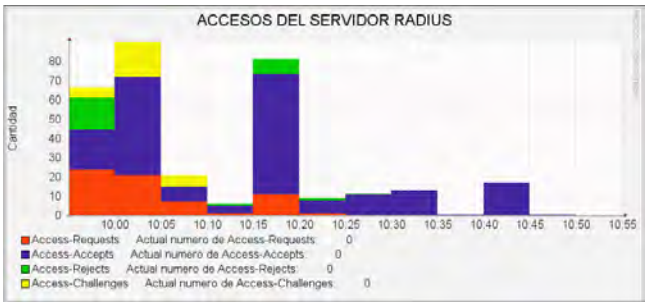

Figura 8: Estadísticas de accesos en gráficos apilados.

RESULTADOS Y DISCUSIÓN

Los gráficos que el sistema reporta permiten tener un mayor control de los eventos globales que el servidor maneja. Esto mejora la calidad de la gestión por parte del administrador de red.

Las estadísticas de los eventos en un servidor Freeradius en su primera versión podía ser monitoreada sin implementaciones adicionales ya que tuvo soporte para SNMP, por lo tanto, el proceso de implementación de un sistema de monitoreo era más sencillo ya que el mismo agente

tenía la capacidad de extraer la información de estado del FreeRADIUS por defecto y no era necesario realizar alguna secuencia de comandos para poder monitorear el estado del servidor. La versión 2 del FreeRADIUS no tiene soporte SNMP nativo pues su código fue borrado antes de su lanzamiento con el objetivo de utilizar un programa separado para reportar las estadísticas del servidor [11]. En este caso, un servidor virtual interno es el encargado de manejar los paquetes de consulta de estado que los clientes realizan.

Actualmente existen otras implementaciones relacionadas al monitoreo de eventos en servidores RADIUS, una de ellas es el caso de F-Ticks el cuál utiliza el módulo linelog y cadenas de syslog para enviar a una base de datos centralizada las estadísticas más importantes como es el caso de Access-accept o Access-reject. Esta información representa las autenticaciones aceptadas o fallidas de la conexión de los usuarios al servidor FreeRADIUS de su institución. En este caso, nuestro sistema muestra información más detallada de los eventos ya que involucra a 34 atributos del FreeRADIUS por separado.

CONCLUSIONES

La generación de reportes de estadísticas en un servidor FreeRADIUS usando SNMPv3 nos permite monitorear de forma sencilla los distintos eventos que se producen en el servidor y tener una mejor administración a diferencia de otras técnicas de monitoreo que consisten en utilizar un registro de eventos en archivos de logs el cual permite monitorear sólo eventos puntuales del diccionario del FreeRADIUS, no siendo tan óptimo para un administrador de red.

AGRADECIMIENTO

Agradecemos al INICTEL-UNI, Área de Conmutación y Transmisión de la Dirección de Investigación y Desarrollo por apoyarnos constantemente en la investigación sobre software libre y darnos alcances precisos sobre temas de monitoreo, en especial al Ing. Fredy Chalco por su valioso aporte en el conocimiento teórico del tema y a los compañeros del área en general por su dedicación exclusiva en investigación.

REFERENCIAS

[1] Edwin Lyle Brown, 802.1X Port-Based Authentication, page 2.
[2] AAA and Network Security for Mobile Access by Madjid Nakhjiri and Mahsa Nakhjiri page 1.
[3] Yago Fernández Hansen, RADIUS / AAA / 802.1x, Sistemas basados en la Autenticación en Windows y Linux/GNU Seguridad Máxima
[4] Essential SNMP, 2nd Edition. Douglas Mauro, Kevin Schmidt. O'Reilly September 2005.
[5] SNMPv3: A Security Enhancement for SNMP. William Stalligns.
[6] RADIUS by Jonathan Hassell ,2002.
[7] RFC 1213 Management Information Base for Network Management of TCP/IP-based internets MIB-II. Network Working Group
[8] Mysql Tutorial By Luke Welling, Laura Thomson
[9] http://www.mrtg.org/rrdtool
[10] Deliverable DJ3.1.2,1: Roaming Developments by GN3-10-304, 24-02-2011.
[11] http://www.mail-archive.com/freeradius-users%40lists.freeradius.org/msg46012.html.
[12] http://www.net-snmp.org/docs/man/snmpd.conf.html
[13] http://www.iana.org/assignments/enterprise-numbers
[14] http://www.net-snmp.org/docs/man/snmpd.conf.html

E-mail:
aleiva@inictel-uni.edu.pe
jquinto@inictel-uni.edu.pe
jquiroz@inictel-uni.edu.pe