Mecanismos de seguridad en redes activas sobre arquitectura SARA

Marcelo Bagnulo[1], María Calderón[2], Bernardo Alarcos[3],David Larrabeiti[4]
[1,2,4] Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid
Av. Universidad 30 - 28911 LEGANES (MADRID)
[3] Área de Ingeniería Telemática, Universidad de Alcalá de Henares
28871 Alcalá de Henares (MADRID)
{marcelo,maria,dlarra}@it.uc3m.es, bernardo@aut.alcala.es

Abstract. *Active network technology enables fast deployment of new network services tailored to the specific needs of end users, among others features. Nevertheless security issues still are a main concern when considering the industrial adoption of this technology. In this article we describe SARA (Simple Active Router-Assistant) architecture, an active network platform deployed in the context of the IST-GCAP project, and then consider security requirements detected in this architecture, concerning confidentiality, integrity, authentication, no repudiation and retransmission. Later, we present the security protocol proposed which intents to cover all imposed requirements, and finally we will address implementation perspectives using available technologies such as IPSec and SSL.*

1 Introducción.

Como evolución de los modelos de red tradicionales, la comunidad científica ha propuesto un nuevo modelo identificado por el término *redes activas* [6], [9]. La idea fundamental es añadir programabilidad a las redes. Las redes activas constituyen una arquitectura de red en la que los nodos de la misma pueden realizar procesamiento "a medida" sobre los paquetes que los atraviesan. Las redes activas producen un cambio en el paradigma de red: de nodos capaces exclusivamente de transportar octetos de forma pasiva, a nodos capaces de procesar los paquetes en cualquier capa de la pila de protocolos.

Las redes activas introducen el concepto de procesamiento específico de los paquetes en base a código móvil que se ejecuta en los nodos de la red. Esto quiere decir que los nodos de la red no son sistemas de procesamiento especializados en un protocolo de red, como sucede en la actualidad, sino que son plataformas de ejecución genéricas en las que se puede descargar dinámicamente código específico para el procesamiento de los distintos tipos de paquetes que se desee definir.

Enmarcado en el contexto del proyecto europeo IST-GCAP [3] (2000-2001) se ha desarrollado en la Universidad Carlos III, una plataforma básica de redes activas denominada SARA que opera sobre redes IPv4 e IPv6.

En esta plataforma, la descarga de código de una aplicación activa (AA) en un nodo activo se hace de forma dinámica cuando llega el primer paquete activo que hace referencia a dicha AA, y se realiza desde uno o varios servidores de código administrados por el proveedor de red. Esta aproximación asegura que en la red activa solamente se ejecutarán aquellas AA que han sido previamente validadas antes de su habilitación por el proveedor o administrador de la red activa, y no el código inyectado por cualquier usuario anónimo como preconizan otras plataformas [4].

Otro elemento importante en esta arquitectura es el concepto de Router-Asistente (Fig. 1), desarrollado por primera vez en la plataforma SARA. En esta arquitectura el router delega las funciones de procesado activo en un asistente presente en una red local de alta velocidad como si de un coprocesador externo se tratara. Esta decisión de diseño tiene implicaciones sustanciales, en primer lugar, permite que la ejecución de aplicaciones activas en la red tenga un impacto mínimo en las prestaciones ofrecidas por el router que debe poder transportar también paquetes convencionales (no activos) con el máximo rendimiento, por otro lado permite dimensionar la capacidad de proceso del nodo activo sustituyendo únicamente el Asistente si fuera preciso. Por último, permite que el router esté blindado frente a potenciales errores en la programación de las AA que podrían afectar al Asistente pero no al router.

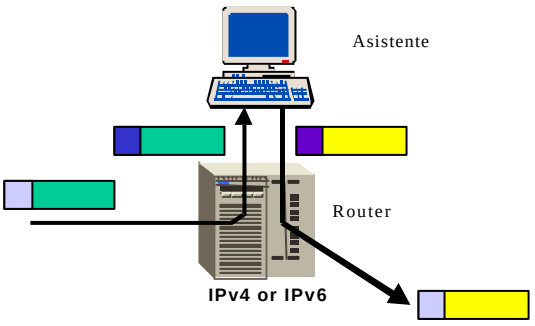

Figura 1. Arquitectura Router-Asistente

La plataforma desarrollada ofrece servicios activos transparentes. Este concepto tiene varias vertientes; por un lado es totalmente transparente para los paquetes tradicionales (paquetes pasivos) el hecho de que coexistan en la red con los paquetes activos y que existan en la red determinados nodos con soporte de redes activas. Por otro lado la topología de red activa es transparente para los sistemas finales, esto significa que estos últimos no están obligados a conocer la ubicación física de los nodos activos para poder hacer uso de sus servicios. Serán los propios nodos activos los que se encarguen de capturar los paquetes activos que los atraviesan. Esta funcionalidad se ha implementado haciendo uso de la opción *router alert* de IP.

En la actualidad SARA es una plataforma de redes activas de alto rendimiento consistente en un prototipo de experimentación que soporta las funcionalidades básicas de una plataforma de este tipo (disponible en: http://matrix.it.uc3m.es/~sara).

Si bien el esquema de funcionamiento de las redes activas ofrece una importante flexibilidad, que permite ofrecer una diversa gama de servicios de forma dinámica, presenta también, por su propia estructura de funcionamiento, una serie de riesgos de seguridad, algunos de los cuales pueden ser muy graves y poner en riesgo el correcto desempeño de toda la red, por lo que intentaremos proponer una solución a los mismos.

En el presente artículo se describirá el intercambio básico de paquetes activos en una red activa con arquitectura SARA para luego analizar los riesgos existentes y los requisitos de seguridad que estos imponen así como las condiciones de borde impuestas por restricciones de otro orden como ser la escalabilidad del sistema. En una segunda instancia se presentará una solución que pretende cumplir con los requisitos detectados para luego evaluar la implementación de la misma utilizando tecnología existente y así dar paso a las conclusiones extraídas.

2. Entorno de trabajo.

2.1 Intercambio básico de paquetes activos en SARA.

2.1.1 Elementos que participan en el intercambio.

Origen: ordenador del usuario de la red activa que genera tráfico en la misma y utiliza las facilidades activas que esta ofrece.

Destino: Ordenador al cual está dirigido el tráfico generado por *Origen*.

RACT: router activo (router + asistente) capaz de interpretar paquetes activos que circulan en la red y obtener el código necesario para procesarlos.

Servidor de código: Repositorio del código ejecutable utilizado por los routers para procesar las distintas clases de paquetes que circulan en la red.

2.1.2 Mecanismo básico.

Para hacer uso de las prestaciones ofrecidas por la red activa para el procesamiento particular de una clase de paquetes, *Origen* debe solicitar dicho servicio a la red (ver Figura 2). Esta solicitud se realiza mediante el envío de un paquete activo (ACT[1]), dirigido hacia el destino final deseado (*Destino*), y que adicionalmente indica a los routers activos del camino el código necesario para el procesamiento de los paquetes de esta clase.

Cuando un router activo recibe un paquete de este tipo, verifica la disponibilidad local del código solicitado. En caso que no posea el código necesario para procesar los paquetes, solicita el mismo al Servidor de Código (CODREQ [2]). El Servidor de Código envía entonces la información solicitada al router activo (COD[3]), quien puede ahora procesar el paquete activo y encaminarlo hacia el próximo salto (ACT[4])).

Los siguientes routers activos del camino realizarán un procedimiento análogo hasta que el último lo encaminará hasta *Destino*, quien ignora la información concerniente al tratamiento del paquete y extrae la información para las capas superiores.

Una vez que se ha establecido el camino y todos los routers contienen el código necesario para procesar los paquetes como lo ha solicitado *Origen*, este debe informar a la red la intención de continuar utilizando este código de procesamiento de paquetes. Para ello, *Origen* debe enviar periódicamente paquetes de refresco (Refresh) del código en uso. Los routers activos verifican el código solicitado y extienden su tiempo de vida en el router.

3. Análisis de riesgos y requisitos de seguridad.

Resulta claro a partir de la descripción previa, que el principal tema a resolver se vincula al control de acceso. Ya sea el control de acceso de los distintos *Origene*s a los códigos solicitados así como el control de cuales routers activos tienen permisos para acceder a los distintos códigos solicitados. También resulta de principal importancia controlar que servidores de código pueden introducir código en los routers activos. Estos requisitos y otros que se detallarán más adelante redundan en los puntos detallados a continuación.

3.1 Autenticación e integridad.

El riesgo más evidente que parece amenazar a la arquitectura descrita, es la posibilidad que alguna parte no deseada pueda cargar código en los routers

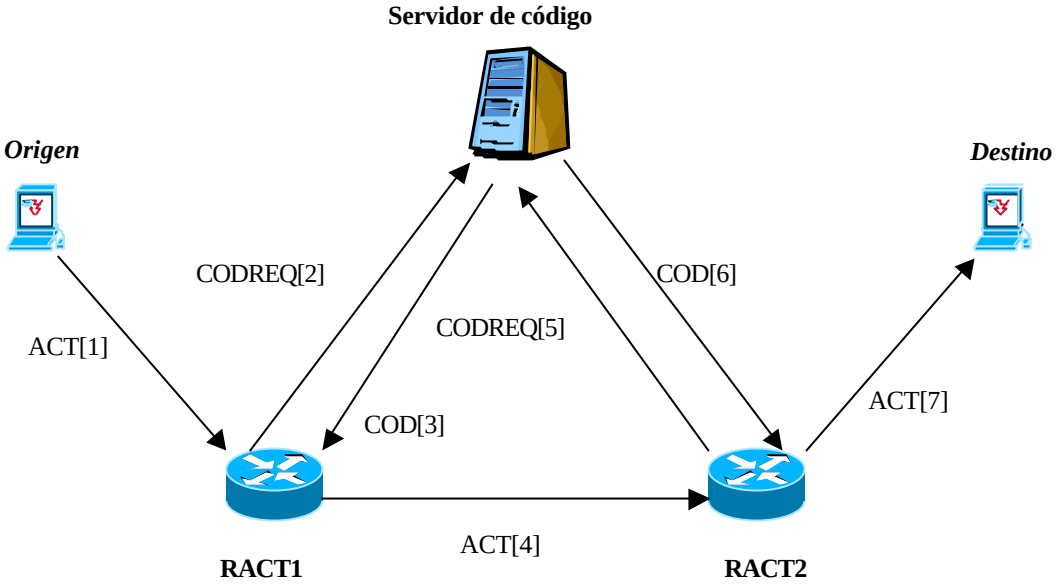

Figura 2

activos de la red, por lo que parece imprescindible que los mensajes que contienen el código (COD) a ser introducido, sean autenticados por parte del router activo de forma de estar seguros que fue el servidor quien generó dichos paquetes. Otro riesgo existente, íntimamente vinculado al anterior, es la posibilidad que un intruso altere el código contenido en el paquete (COD) mientras viaja hacia el router, por lo que también resulta necesario que dicho paquete posea una verificación de integridad asociada a la autenticación.

Adicionalmente, por razones de servicio, resulta deseable que sólo partes autorizadas sean capaces de ejecutar ciertos códigos, ya sea porque los mismos son potencialmente peligrosos para la red, (por ejemplo requieren muchos recursos del router), por lo que también es necesario poder identificar a *Origen* cuando solicita la ejecución de un código particular (ACT). Análogamente al caso anterior, resulta necesaria una verificación de la integridad de la solicitud realizada.

Finalmente, podemos pensar que algunos de los códigos existentes en el servidor no sean de dominio público por lo que sería deseable que sólo routers activos autenticados pudieran acceder al mismo.

3.2 Confidencialidad.

Como ya se ha mencionado en el párrafo anterior, parecería interesante que el código intercambiado no pudiera ser accedido por terceras partes que tienen acceso a la red, por lo que dicha comunicación (COD) debería ser confidencial. Los otros intercambios podrían ser confidenciales por razones menos críticas, como privacía de las intenciones de los clientes de los distintos servicios pero creemos que este es un requisito opcional para el resto de los paquetes.

3.3 No repudio.

Es posible imaginar que los operadores de las redes activas deseen cobrar de forma diferenciada en función de la forma de procesamiento de paquetes que sea solicitada a los routers activos mediante los paquetes ACT, por lo que dichos paquetes tendrán implicaciones comerciales y contractuales, específicamente jugará el rol de solicitud de servicio. Por lo tanto es interesante poder garantizar el no repudio de dichos paquetes por parte de *Origen*, ya que la tarificación del servicio se realizará en función de esto.

3.4 Retransmisiones.

El proceso de carga de código en los routers activos, es exigente en memoria y en capacidad de procesamiento, por lo que es posible imaginar un ataque a los routers de la red simplemente retransmitiendo paquetes válidos de un origen auténtico, lo que aumentaría la demanda en el router e incluso podría saturarlo. Por ello es importante que los paquetes que pueden cargar código (ACT) posean una validez temporal de forma que luego de un lapso ya no sean más válidos y sean descartados por todas las partes. Asimismo sucede con las solicitudes de código al servidor (COD) ya que se podría imaginar un ataque al mismo solicitando múltiples veces código ya pedido.

3.5 Requisitos adicionales.

3.5.1 Conocimiento nulo de usuarios por parte de los routers activos.

Resulta imprescindible diseñar el sistema de forma escalable, considerando una cantidad importante de potenciales usuarios, por lo que no resulta posible considerar una administración de permisos de usuarios local en cada router activo. Es necesario por consiguiente, la utilización de un servicio de directorio que contenga toda la información de usuarios y permisos, donde no sea necesaria la información de usuarios en los routers.

3.5.2 Transparencia frente a los routers activos que forman el camino.

Resulta importante que los distintos *Origene*s no deban conocer los routers que formen el camino hasta el destino, para brindar transparencia de la red e independencia de la topología de la misma, así como asegurar la escalabilidad, por lo que los paquetes activos enviados (ACT) no deben depender de los routers por los que deban transitar. Este requisito es particularmente fuerte parta un sistema de seguridad donde se requiere autenticación y confidencialidad con una parte a la cual no se conoce.

3.5.3 Eficiencia y velocidad en el procesamiento.

Recordemos que el objetivo principal de la subred es la transmisión de la información por lo que la velocidad en el procesamiento de los paquetes debe ser óptima y si bien los requisitos de seguridad son relevantes, la eficiencia en el uso es la razón de ser.

4. Arquitectura de seguridad.

4.1 Infraestructura necesaria.

Para el funcionamiento de la solución propuesta, resulta necesario contar con los siguientes elementos:

Origen: un par de claves asimétricas, pública (PubO) y privada (PrO) y un certificado digital que asocie dichas claves a *Origen* (opcional; ver 4.5).

RACTs: par de claves asimétricas, pública (PubR1 y PubR2) y privada (PrR1 y PrR2) y certificado digital que asocie dichas claves al router correspondiente (opcional; ver 4.5).

Servidor de Código: par de claves asimétricas, pública (PubSC) y privada (PrSC) y un certificado digital que asocie dichas claves al Servidor de Código (opcional;ver 4.5).

4.2 Definiciones.

Firma digital: calculamos el HASH de lo que deseamos firmar y luego encriptamos el mismo con la clave privada del firmante.

Verificar firma digital: calculamos el HASH de la información firmada y luego lo comparamos con el resultado de desencriptar con la clave pública del fimante el HASH recibido con la información recibida. En caso que coincidan, decimos que la firma ha sido verificada; en caso que no coincidan el paquete será descartado.

4.3 Protocolo de seguridad en el intercambio básico de paquetes activos.

ACT[1]: Como ya hemos dicho, este paquete es especialmente crítico y debe ser autenticado, no repudiable, íntegro y con validez temporal para evitar retransmisiones, por lo que se propone que el mismo contenga un sello temporal (timestamp) que establezca la hora de creación del mismo y se firme digitalmente el contenido activo del paquete con la clave privada de *Origen* (PrO) de forma de asegurar autenticidad, integridad y no repudio. Adicionalmente, este paquete debe transmitir una clave simétrica (K), previamente generada por *Origen*, que será luego utilizada por los routers para autenticar los mensajes de Refresh. Dicha clave debe ser transmitida de forma secreta y como *Origen* solo conoce al Servidor de Código (no puede conocer a los routers por requisito establecido previamente) K se encripta con la clave pública del servidor (PubSC).

Contenido de ACT
- Dirección origen: *Origen*
- Dirección destino: *Destino*
- Identificación de código deseado
- Clave simétrica K
- Timestamp

Firmado por *Origen*
Confidencialidad: clave simétrica K encriptada por PubSC

CODREQ[2]: El router activo analiza el paquete activo, pero por el requisito impuesto de no conocimiento de *Origen* por los routers, este no puede verificar si el paquete no ha sido alterado. Verifica si posee o no el código solicitado y reenvía el paquete recibido al Servidor de Código encapsulándolo en otro paquete e indicando si posee o no el código solicitado. En caso de que el router ya posea el código, este intercambio es utilizado para obtener autorización de acceso al código por parte de *Origen*.

Contenido de CODREQ
- Dirección origen: RACT1
- Dirección destino: Servidor de Código
- ACT
- Indicación si desea o no el código

COD[3]: Cuando el Servidor de Código recibe la solicitud (CODREQ), este verifica la firma de *Origen* utilizando su clave pública (PubO) para luego comprobar si *Origen* posee permisos para realizar la presente solicitud. En caso afirmativo, comprueba el timestamp, de forma que el momento de creación del paquete esté dentro de un entorno aceptable del momento presente. Adicionalmente verifica que el router activo solicitante posea los permisos para ejecutar el código solicitado. Una vez realizadas las verificaciones, desencripta la clave K utilizando PrSC. Genera entonces el paquete COD en el cual incluye el código en caso que el router activo así lo solicite y un timestamp. Firma luego la información utilizando PrSC para luego encriptarlo con la clave K. Finalmente adjunta la clave K encriptada con la clave pública del router (PubR1) de forma que sólo éste pueda leer la información enviada.

Contenido de COD
- Dirección origen: Servidor de código
- Dirección destino: Router activo 1
- Código solicitado
- Clave simétrica K encriptada con PubR1
- Timestamp

Firmado por SC
Confidencialidad para Router Activo 1 (encriptado por K, clave K encriptada con PubR1)

ACT[4]: Una vez que el Router activo recibe COD, este desencripta la clave K utilizando su clave privada (PrR1) para luego desencriptar el resto del contenido del paquete utilizando la clave K. Posteriormente verifica la firma digital del servidor utilizando PubSC. Una vez superadas estas corroboraciones verifica que el tiempo de generación del paquete (timestamp) se encuentre dentro de un entorno admisible del instante actual. En caso afirmativo, obtiene el código del paquete en caso que lo hubiera solicitado y asocia la clave K a dicho código, para luego cursar el paquete activo hacia el próximo salto.

Los siguientes routers activos del camino realizarán un procedimiento análogo al descrito anteriormente hasta que finalmente el último de los routers encaminará el paquete hacia *Destino*. Este simplemente descartará la cabecera de seguridad, ya que no es capaz de comprenderla y la información contenida no le presenta interés alguno.

Paquetes de Refresh: Una vez que se ha notificado a todos los routers activos del camino el código a utilizar, es necesario enviar paquetes que indiquen la extensión de la validez del código utilizado en el tiempo de forma que este permanezca en los routers. Dichos paquetes son generados por *Origen* y serán dirigidos a *Destino*; contienen la identificación del código deseado, una marca temporal y están firmados por *Origen*. Para que el proceso de verificación de firma sea menos exigente para los routers, se utiliza la clave K para autenticar los paquetes de refresh, adjuntando el hash del contenido del paquete concatenado con K.

Contenido de Refresh
- Dirección origen: *Origen*
- Dirección destino: *Destino*
- Identificación de código deseado
- Timestamp

Autenticación: hash del contenido del paquete concatenado con la clave simétrica.

Cuando los routers activos reciben el paquete de Refresh, verifican la firma comparando el hash contenido en el paquete con el hash resultante del contenido del paquete concatenado con la clave asociada al código, la cual obtienen de sus bases internas. Si además la marca temporal del paquete es correcta, extienden el tiempo de vida del código dentro del router y envían el paquete hacia el próximo router del camino. Una vez concluido el trayecto, *Destino* recibe el paquete, quien descarta la cabecera de seguridad.

4.4 Funcionalidades opcionales.

4.4.1 Confidencialidad.

Es posible modificar el protocolo propuesto para que todos los mensajes intercambiados sean confidenciales. Si bien esto puede ser deseable en ciertas situaciones, normalmente creemos que es una tarea muy exigente en cuanto a procesamiento, por lo que debe ser opcional su implementación dentro del protocolo. A continuación detallaremos como deberían encriptarse los paquetes intercambiados:

ACT: Como se ha descrito anteriormente, este paquete transporta una clave K encriptada con la clave pública del servidor de código (PubSC). El resto de información se encripta con la clave K para mejorar la velocidad de encriptado ya que los algoritmos de encriptado de claves simétricas son menos exigentes que los de claves asimétricas. El problema que surge en este punto es la imposibilidad del router de comprender la identificación del código solicitado por *Origen*. Las posibilidades entonces son o bien enviamos dicha información en texto en claro o bien el router no lo comprende y el servidor de código siempre envía el código solicitado independientemente de que el router ya lo posea. La primera opción presenta el inconveniente que la identificación de código no es confidencial y el encriptado entonces protege poca información y poco relevante por lo que es dudoso que justifique el esfuerzo. La segunda opción aumenta el overhead del protocolo de seguridad pero puede ser compatible con una red donde el tiempo de vida del código en los routers es corto.

Refresh: Los paquetes de refresh pueden ser encriptados por la clave K que ya es conocida por todos los routers del camino y corresponde a un algoritmo de clave simétrica.

4.5 Certificados digitales.

Hasta el momento hemos supuesto que el Servidor de Código conocía todas las claves públicas de todos los orígenes posibles y que además la administración de los permisos de acceso a los distintos códigos residentes en el servidor para cada usuario era realizada también en el Servidor de código. Si bien esto es posible, existen dificultades en el momento de escalar al solución, en particular cuando los usuarios pertenecen a diversas esferas administrativas.

Una solución posible es la utilización de certificados digitales para los orígenes. Para ello, cada vez que se da de alta un nuevo usuario, se debe generar un certificado digital que asocie la identificación de usuario con la clave pública del mismo. Ahora cuando dicho usuario desee enviar un paquete firmado, debe además de encriptarlo con su clave privada adjuntarle el certificado. Para verificar la firma del paquete, el servidor deberá verificar la autenticidad del certificado para luego verificar la firma. En este escenario, el Servidor de código solo debe conocer la clave pública de aquellos emisores de certificados digitales autorizados para usuarios del sistema. Adicionalmente, se podría incluir dentro de los certificados los permisos de usuario que informarían al Servidor de Código si el usuario posee o no permiso para ejecutar el código solicitado, de forma que estos permisos también fueran administrados por el emisor de certificados y no por el servidor de Código. El problema que puede presentar esta solución es la dinámica de los permisos de usuarios. Para ello se podría elaborar un sistema de certificados de atributos, es decir certificados que se generen dinámicamente a pedido en un servidor de certificación central presentando el certificado inicial.

5. Consideraciones sobre la implementación del protocolo.

Una vez definidos los requisitos y el protocolo de seguridad deseado, debemos evaluar las posibles implementaciones del mismo. Para ello, lo más recomendable es el estudio de las diversas implementaciones de protocolos de seguridad existentes, como elementos funcionales sencillos sobre los que construir la solución propuesta. Los protocolos que nos resultaron más interesantes, por su amplia difusión, fueron SSL e IPSec, por lo que se evaluarán a continuación.

5.1 Secure Socket Layer.

SSL es un protocolo de capa de aplicación utilizado para el establecimiento de una sesión segura entre dos partes. Incluye una etapa de handshake en la que se intercambian claves de forma segura, basándose en que al menos una de las partes posee un certificado. Este protocolo parece un posible candidato para el transporte seguro de código entre el servidor y los routers activos. Sin embargo, cabe notar que la negociación de una sesión de seguridad antes mencionada puede añadir mucho overhead, tanto en el procesamiento como en el ancho de banda necesario.

5.2 IPSec.

IPSec es un conjunto de protocolos diseñados por el IETF para brindar seguridad a IP. Originalmente fue definido para IPv6 pero luego se extendió de forma que pueda ser utilizado con IPv4. El objetivo de IPSec es la comunicación segura entre dos partes remotas. Cabe notar en esta instancia que el objetivo de IPSec es proteger los paquetes intercambiados de los dispositivos intermedios, como pueden ser los routers, es decir exactamente lo contrario de lo que deseamos hacer, es decir, dar indicaciones a dichos dispositivos. Es razonable esperar entonces que se deban realizar al menos ciertos cambios en el espíritu de IPSec. En particular los mecanismos de seguridad son aplicados a todo lo que contiene el paquete IP, incluyendo toda la información de usuario, por lo que, para aplicarlo en nuestro modelo, además de los routers, *Destino* también debe poder comprender los paquetes. Para ello, se podría establecer un intercambio de claves inicial entre *Origen* y *Destino* de forma que ambos conozcan la clave K utilizada, este intercambio podría realizar utilizando Diffie-Hellman o haciendo uso de certificados digitales.

IPSec está compuesto esencialmente de AH, ESP y IKE. Estudiaremos a continuación los posibles usos de estos para el protocolo propuesto.

AH – Authentication Header, brinda facilidades de firma y control de integridad. AH podría utilizarse para todos los paquetes firmados del protocolo. En particular parece un formato idóneo para los paquetes de solicitud (ACT) y para los paquetes de refresh que son simplemente firmados.

ESP – Encapsulating Security Payload, brinda facilidades de firma, control de integridad y confidencialidad, por lo que resulta interesante su utilización para los paquetes que transportan código desde el servidor a los routers activos.

IKE – Internet Key Exchange, posibilita el intercambio de claves entre dos partes para el posterior uso de las mismas con AH y/o ESP. Este protocolo podría ser utilizado para el acuerdo de la clave K entre *Origen* y *Destino* previo al comienzo del protocolo en sí de forma que *Destino* pueda comprender los paquetes.

6. Trabajos relacionados.

Existen diversas arquitecturas de redes activas desarrolladas en distintos proyectos y todas ellas, en mayor o menor grado, intentan resolver los diferentes problemas de seguridad resultantes de la propia estructura. Cabe notar que los riesgos detectados en cada caso dependen fuertemente de la arquitectura de redes activas utilizada por lo que las soluciones propuestas pueden diferir considerablemente de un proyecto a otro. En el presente trabajo se han estudiado principalmente requisitos de tipo "dinámico" [6], es decir requisitos que deben realizarse durante el procesamiento de ciertos paquetes críticos. Las herramientas criptográficas utilizadas en la solución, es decir criptografía de claves asimétricas y simétricas, funciones de hash y certificados digitales son técnicas conocidas y utilizadas en diversas soluciones de seguridad en redes tanto activas como convencionales, IPSec por ejemplo. Las diferencias relevantes con otros esquemas de seguridad se encuentran principalmente en el protocolo de seguridad planteado el cual se ha diseñado para cumplir con los requisitos definidos como relevantes en el presente entorno de trabajo y que el mismo ha sido diseñado de forma de beneficiarse de las características particulares de la arquitectura SARA. Cabe destacar que el protocolo propuesto cumple naturalmente con los requisitos referentes a transparencia frente a los routers activos que forman el camino y al conocimiento nulo de usuarios por parte de los routers activos, basándose esencialmente en el papel desempeñado por el Servidor de código y no exige una negociación de una sesión de seguridad entre las partes ya que toda la información necesaria se encuentra en el paquete activo (ACT). Otras arquitecturas como SANE [1], al carecer de una figura central que tome el papel de administrador, requieren una negociación entre el origen de los paquetes y cada uno de los nodos activos del camino, dificultando así la escalabilidad de la solución así como la introducción de nuevas facilidades como agentes móviles. Soluciones alternativas para estos problemas conocidas como "single packet authentication", son costosas y presentas diversos inconvenientes [1]. Otras arquitecturas como ANTS, basan su esquema de seguridad en la restricción del entorno de ejecución de código de forma de un paquete activo no sea capaz de consumir todos los recursos de red [9] y no consideran requisitos adicionales como autenticación de origen o no repudio. Cuán idónea resulte dicha solución dependerá del ambiente de trabajo en el que se pretenda implantar el sistema.

7. Conclusiones.

En el presente trabajo se ha presentado un protocolo de seguridad cuyo objetivo es solucionar los riesgos existentes en la implementación SARA de redes activas, previamente analizados. La solución propuesta al cumplir con los requisitos identificados, particularmente con los que hemos llamado conocimiento nulo de usuarios por parte de los routers activos y transparencia frente a los routers activos que forman el camino, resulta escalable ya no requiere una negociación entre el origen y cada uno de los nodos activos del camino, permitiendo una autenticación mediante un solo paquete, a saber el paquete activo que se desea enviar. La escalabilidad está basada también en un esquema centralizado de administración de permisos de control de acceso, el cual es posible gracias a la arquitectura de SARA. Estas características implican que el protocolo impone una carga relativamente baja a los routers activos. Adicionalmente parece posible una implementación de la solución utilizando en parte tecnología disponible, IPSec y/o SSL. En particular es especialmente interesante la evaluación del nivel de exigencia de procesamiento que impone el protocolo al router, así como el overhead generado por el esquema de seguridad (especialmente cuando se utilizan certificados digitales).

Agradecimientos.

El presente trabajo se ha realizado en el marco de los proyectos TEL99-0988-C02 y GCAP IST-1999-10 504 de CICYT (Comisión Interministerial de Ciencia y Tecnología)

Referencias.

[1] D. Scott Alexander, William A. Arbaugh, Angelos D. Keromytis y Jonathan M. Smith, "A Secure Active Network Environement Architecture", IEEE Network, [1998]

[2] Doraswamy, Naganand, "IPSec : the new security standard for the Internet, Intranets and virtual private networks", Upper Saddle River (New Jersey) : Prentice Hall PTR , [1999]

[3] Servidor www proyecto IST GCAP (Global Communication Architecture and Protocols for new QoS services over IPv6 networks). http://www.laas.fr/GCAP/

[4] D. Wetherall, J. Guttag, and D.L. Tennenhouse. "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols". IEEE OPENARCH'98, San Francisco, CA, April 1998.

[5] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", Request for Comments: 2401, Network Working Group, [1998]

[6] Konstantinos Psounis, "Active networks: applications, security, safety and architectures", IEEE Communications Surveys, [1999]

[7] Schneier, Bruce, “Applied cryptography : protocols, algorithms and source code in C”, New York, John Wiley & Sons , [1996]

[8] Urueña, Manuel, “Diseño de una plataforma de redes activas basada en arquitectura router-asistente”, Facultad de Informática, UPM, [2001].

[9] David Wetherall, Ulana Legedza y John Guttag, “Introducing new Internet services: Why and How”, IEEE Network Magazine [1998]