Sistema para Evitar Ataques al Protocolo ARP en Redes de Área Local

Marcos, Xavier; Ortega, Andre; Abad, Cristina Ms.Sc.
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral
Campus Gustavo Galindo, Km 30.5 vía Perimetral,
Apartado 09-01-5863. Guayaquil-Ecuador
{xmarcos,aortega,cabad}@fiec.espol.edu.ec

Resumen

Se conoce que las redes de área local presentan problemas de confiabilidad en sus comunicaciones y que hay un gran riesgo de que muchas veces la información que circula sea interceptada o manipulada por terceras personas no deseadas. Consideramos que la protección contra el robo de la información por los ataques generados contra ARP es un problema de suma importancia tanto para los usuarios de redes hogareñas, como para aquellos de grandes empresas. Por esta razón presentamos una solución que impide la realización de ataques que aprovechan las vulnerabilidades que posee el protocolo ARP. En este documento se plantea una posible solución que a diferencia de otros esquemas no depende de técnicas criptográficas, alteraciones al protocolo, o modificaciones en cada uno de los computadores que conforman la red de área local. La solución presentada es eficaz, económica y asequible, y muestra claramente ventajas sobre los otros esquemas de detección actualmente desarrollados.

Palabras Claves: *protocolo, ARP, switch, firmware, trama, paquete, dirección MAC, dirección IP, broadcast.*

Abstract

We already know that nowadays the LANS have confidentiality problems and that there is a big risk that data that travels in our network might be intercepted or manipulated by third party persons. We consider that protection and security against the loss of valuable information caused by taking advantage of the Address Resolution Protocol (ARP) it's a huge problem from owners of big enterprises to even users of SOHO LANS. This is the reason why we would like to present a solution that doesn't allow this kind of attacks and doesn't permit to take advantage of vulnerabilities that the ARP has. Not like many other approaches that implement cryptographic methods, re-implementation in the ARP or changes in every single computer that belong to the LAN. The solution proposed is effective and not expensive and clearly shows advantages among the other detection methods that already exists.

1. Introducción

Para que la comunicación se lleve a cabo entre dos computadores de una red, ambos deben conocer las direcciones físicas (MAC) correspondientes para que a nivel de la capa de enlace de datos se pueda completar la trama y el paquete pueda ser enviado a su destino correspondiente. Para obtener esta dirección MAC se utiliza el protocolo ARP (Address Resolution Protocol), el cuál es el encargado de conseguirla a partir de una dirección IP.

La manera como trabaja este protocolo es muy sencilla. Cuando un computador desea transmitirle datos a otro del cual se desconoce su dirección MAC se envía una petición ARP de tipo broadcast para que sea recibido por todos los computadores de la red. Así, el computador que tiene la dirección física que se está buscando, en el momento en el que recibe esta consulta contesta con una respuesta ARP y le permite saber su dirección física al computador inicial. Cada computador tiene lo que se denomina caché ARP, tabla donde se guardan las direcciones MAC de las respectivas direcciones IP recientemente utilizadas para no tener la necesidad de producir excesivo tráfico de consultas ARP.

El problema del funcionamiento de este protocolo ocurre cuando un computador malicioso de la red responde a una consulta ARP que no le corresponde, ya que ARP es un protocolo sin estado, el computador que recibiera esta clase de respuesta ARP modificada no tendría manera de verificarlo y añadiría en su caché ARP un mapeo incorrecto. Esto se conoce como envenenamiento a la caché y es la manera como se obligaría a un computador víctima, enviarle tramas a otro nodo que no necesariamente es el destinatario pero se hace pasar por él, pudiéndose llegar a obtener de esta manera información que podría ser considerada importante en una red local.

Actualmente para este tipo de ataques existe una serie de técnicas de detección, prevención y mitigación del problema[1], sin embargo estos esquemas suelen no ser muy efectivos por no combatir todos los tipos de ataques existentes o porque generalmente presentan cambios sobre el funcionamiento original del protocolo ARP. Por esta razón se desea presentar una solución nueva que sea capaz de evitar por completo toda clase de ataques a este protocolo, sin modificar su comportamiento original y que se adapte de manera natural a las redes que actualmente lo usan.

2. Diseño

El diseño de la solución que realizamos consiste principalmente en bloquear en su totalidad toda clase de respuestas ARP no deseada. Es decir es una solución centralizada en la cual el único capaz de poder producir tráfico ARP será el servidor de nuestro sistema. Entonces para llevar a cabo una solución de este tipo vamos a requerir de dos elementos esenciales. Primero el ruteador/switch que conforma nuestra red de área local, y segundo el servidor que responderá a las consultas ARP.

3. Implementación

Para que el servidor pueda llegar a responder estas consulta ARP de una manera correcta tendrá que saber cuando se produce una consulta ARP en la red y de alguna manera también estar constantemente al tanto de los mapeos de las direcciones MAC con las direcciones IP de los computadores que estén conformando la red para poder contestar con la respuesta ARP correspondiente.

Entonces las dos funciones principales que realiza el ruteador/switch de nuestra solución son primero redireccionar todo el tráfico dhcp hacia el servidor para que de esta manera pueda conocer las direcciones MAC de los computadores de la red y así poder formar las respuestas ARP, y segundo redireccionar las consultas ARP hacia el servidor para que este se encargue de responderlas y bloquear el resto de tráfico ARP que no sea deseado. Así de esta manera se podrá evitar toda clase de ataques de tipo ARP en la red.

Ahora para que el ruteador/switch pueda llevar a cabo esta tarea es esencial el uso de iptables y ebtables que son funcionalidades que son parte del firmware opensource OpenWRT [2]del ruteador/switch. Las reglas iptables son muy importantes ya que nos permiten filtrar el tráfico de la red para así por medio de su modulo ULOG poder capturar todos las tramas dhcp en un archivo, el cual será próximamente enviado hacia el servidor y así cumplir con la primera función que se mencionó anteriormente. Estas son las reglas que se usan para hacerlo:

- Para guardar las tramas DHCP recibidas por el ruteador

iptables -t mangle -A INPUT -s IP_RED_LOCAL -j MARK --set-mark 100

iptables -A INPUT -m mark --mark 100 -p udp --sport 68 -j ULOG

- Para guardar las tramas DHCP enviadas por el ruteador

iptables -t mangle -A OUTPUT -d IP_RED_LOCAL -j MARK --set-mark 200

iptables -A OUTPUT -m mark --mark 200 -p udp --dport 68 -j ULOG

Cuando estas reglas se ejecutan en el ruteador lo que se hace es que por medio de la tabla mangle de iptables se marcan los paquetes de toda la red. Luego de estos paquetes marcados se procede a almacenar en un archivo ulog.pcap los paquetes udp del puerto 68 es decir los paquetes dhcp de la red. Una vez hecho esto ahora hay que enviar estas tramas que se encuentran guardadas en el archivo hacia el servidor para que este los interprete y pueda guardar los mapeos MAC, IP en su caché y así poder generar las respuestas ARP. Para realizar esta sencilla tarea se creó un pequeño programa que usa la librería libpcap y que se encuentra constantemente corriendo en el servidor. El programa se encarga de leer del archivo ulog.pcap las tramas que se almacenan y luego reenviarlas hacia el servidor haciendo uso de su dirección MAC previamente conocida.

Ahora para la segunda función importante del ruteador se hace uso de las reglas ebtables, las cuales las cargamos cuando se enciende el ruteador. Las reglas ebtables que usamos son las siguientes:

ebtables -A FORWARD -p ARP --arp-opcode 2 -i ! vlan3 -j DROP

ebtables -t nat -A PREROUTING -p ARP --arp-opcode 1 -s ! MAC_SERVIDOR -d ff:ff:ff:ff:ff:ff -j dnat --to-destination MAC_SERVIDOR

ebtables -t nat -A PREROUTING -p ARP --arp-opcode 1 -s ! MAC_SERVIDOR -d ! ff:ff:ff:ff:ff:ff -j dnat --to-destination MAC_SERVIDOR

De estas reglas la primera es la que nos permite bloquear, las tramas ARP con código de operación 2 o todas las respuestas ARP, menos las que provengan de la interface vlan3 debido a que en esa interface es donde se va a conectar el servidor que va a generar todas las respuestas ARP. Y por último están las últimas dos reglas que son las que reenvían todas las consultas ARP hacia el servidor para que se encargue de responderlas.

Hasta aquí tenemos lista la primera parte de la solución eficaz y económica que se quieren llegar a tener bloqueando todos las tramas ARP no deseadas. Ahora bien para terminar la solución solo falta que se produzcan las respuestas ARP, las cuales van a ser generadas por el servidor. Para lograr esto se implementaron dos programas en él. El primero es un programa que toma las tramas dhcp que el ruteador envía hacia la interface del servidor y con ellas actualizar la caché ARP del mismo. Para hacer esta manipulación de la caché ARP el programa usa la librería libdnet. Una vez que se tiene actualizada la caché entonces se podría generar las respuestas ARP de la red usando estos mapeos de direcciones MAC, IP para lo cual se tiene un último programa en ejecución en el servidor que por cada trama ARP que el ruteador le envía este genera su correspondiente respuesta ARP obteniendo la respectiva dirección MAC que se desea conocer de su caché ARP ya actualizada y la envía al computador que la requiere haciendo uso de la librería libpcap. De esta manera es como el servidor aprovecharía los mensajes dhcp y generaría las respuestas ARP necesarias en la red. Es importante recalcar que estos programas deberán ser ejecutados con una muy alta prioridad debido a que es necesario que su tiempo de respuesta sea lo más rápido posible.

Ahora la última cosa que debemos asegurar en el servidor para que la solución sea segura es bloquear el paso de consultas ARP falsificadas hacia el servidor debido a que como este es el que ahora contesta a todas las peticiones ARP de la red tenemos que evitar que sea envenenado. Para lograrlo hacemos uso de la siguiente regla de arptables

```
arptables -A INPUT --opcode 1 -j DROP
```

la cual bloquea todos las peticiones ARP eliminando el riesgo de ser envenenado y así lograr obtener nuestra solución segura.

La Figura 1 ilustra la solución propuesta.

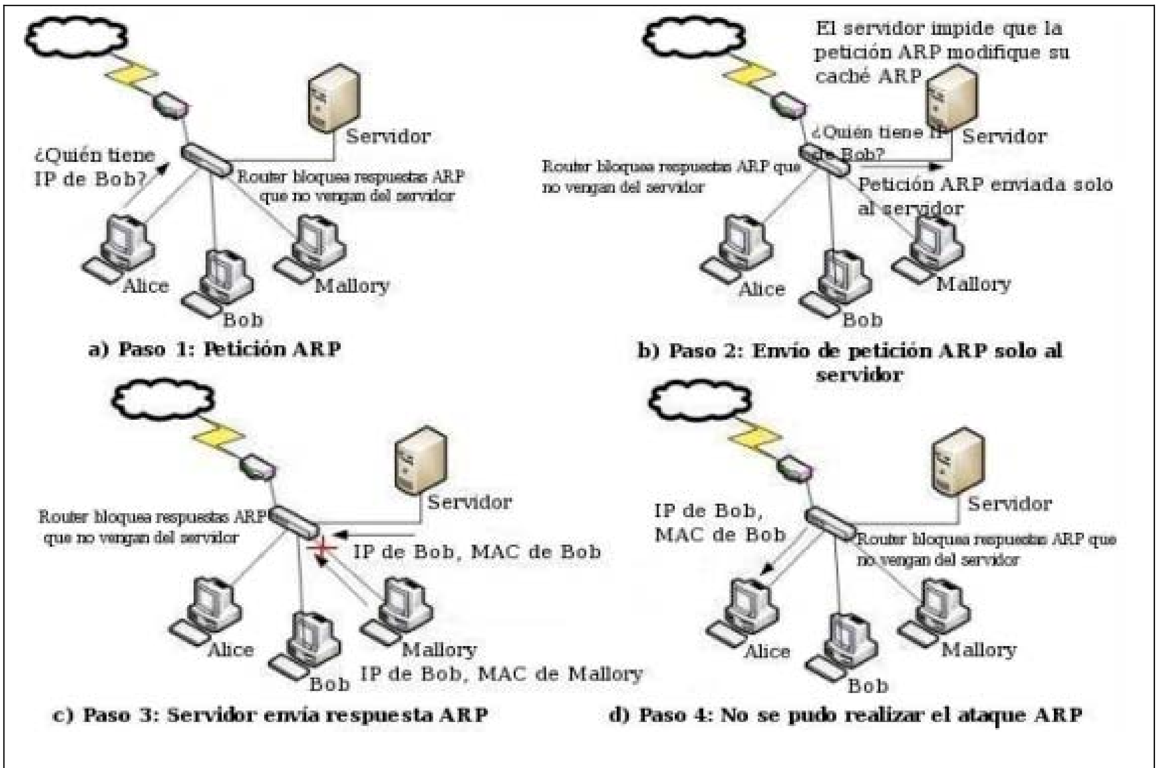

Figura 1. Ejemplo de funcionamiento de nuestro mecanismo para evitar ataques al protocolo ARP.

4. Pruebas y rendimiento

Una vez terminada la solución tuvimos que realizar las pruebas de funcionamiento para corroborar que de verdad evite por completo toda clase de ataques al protocolo ARP. Los ataques utilizados para las pruebas fueron desarrollados por Luis Chiang como parte de su proyecto de tesis[3]. Todas las pruebas consistían en enviar grandes cantidades de paquetes falsificados ARP, pero enviándolos en diversos escenarios para tratar de lograr encontrar en alguno de éstos, una vulnerabilidad. Ninguna prueba tuvo éxito al tratar de envenenar la caché ARP de un computador de la red.

También se realizaron pruebas de rendimiento para comprobar que el sistema sea una solución eficiente. Para lo cual se compararon las mediciones de tiempo de respuesta de tráfico en la red, sin solución y con solución. Al hacer esta comparación se pudo observar que el tráfico de datos con nuestra solución no es tan rápido como el tráfico sin ella pero es un retardo que puede ser tolerable como se puede ver en la Figura2. Luego se observó los tiempos de respuestas de las contestaciones ARP generadas por el servidor y estas no fueron tan eficientes como se esperaba debido al uso de la librería libpcap[4], sin embargo se resolvió que esto no era un gran problema debido a que la frecuencia con la que se producen de estas consultas-respuestas ARP en el día a día en una red de área local es muy mínima, esto fue comprobado con un estudio del tráfico de una red que se hizo por 5 días en el cual se observó que el tráfico ARP representa aproximadamente el 0.06% del tráfico total. Por lo cual a pesar de que nuestra solución no sea ideal representa igual una gran ventaja con respecto a los otros esquemas de detección y mitigación ya existentes.

5. Conclusión

Sabemos que en las redes que actualmente manejamos en nuestras empresas, universidades y hogares todavía existe el problema de que son susceptibles a que se produzca una manipulación de la información o que simplemente su información confidencial sea visible para personas no deseadas, lo cual es un problema muy grave que no debería dejarse olvidado y sin solución. Por estos motivos se presenta esta solución que aunque no es cien por ciento ideal por afectar el rendimiento del protocolo ARP es muy provechosa porque presenta grandes ventajas al no producir cambios en el protocolo y ser transparente a los usuarios de la red, sobretodo produce beneficios en lo que se refiere al aspecto económico y técnico que para muchos son factores muy determinantes en el momento de adquirir e implementar una nueva solución.

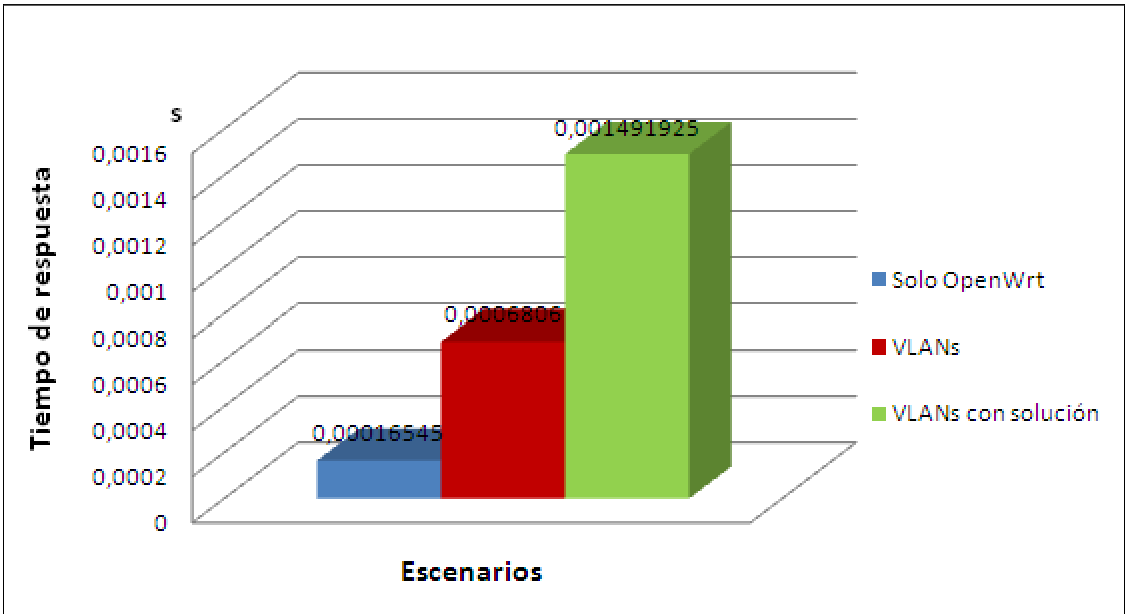

Figura 2. Medición de tiempo de transmisión de paquetes.

6. Agradecimientos

Este trabajo ha sido posible gracias al financiamiento del programa VLIR-ESPOL y a la donación de equipos de la FIEC. Especialmente gracias al apoyo y orientación de la Ing. Cristina Abad.

7. Referencias

[1] Abad, Cristina y Bonilla, Rafael. “An Analysis of the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks”. En Proceedings of the IEEE International Conference on Distributed Computing Systems Workshops (ICDCS 2007 Workshops), Toronto, Canadá, Junio 2007, ISBN: 0-7695-2838-4.

[2] “OpenWrt Wireless Freedom”, <http://downloads.openwrt.org> [Consulta: Viernes, 13 de junio de 2008].

[3] Chiang, Luis y Abad, Cristina. “Hacia Un Mejor Entendimiento de los Ataques al Protocolo ARP, a Través de la Identificación de Árboles de Ataque en una Red Cerrada.”. Enviado para su revisión a ESPOLCIENCIA 2007

[4] Weigle, Eric, Feng, Wu-chun. “TICKETing High-Speed Traffic with Commodity Hardware and Software”. En Passive & Active Measurement Workshop (PAM2002), Fort Collins, Colorado, Marzo 2002.