R. Rios, and J. A. Onieva, "Clasificación de canales encubiertos. Un nuevo canal: Covert_DHCP", *X Reunión Española de Criptología y Seguridad de la Información (RECSI08)*, pp. 325-336, 2008.
NICS Lab. Publications: `https://www.nics.uma.es/publications`

Clasificación de canales encubiertos. Un nuevo canal: Covert_DHCP

Rubén Ríos José A. Onieva

Resumen

Los canales encubiertos son una forma de comunicación oculta que puede vulnerar la integridad de los sistemas. Desde sus inicios en sistemas de seguridad multinivel a principios de los años 70 han evolucionado considerablemente, apareciendo soluciones para redes de computadores debido a la especificación de algunos protocolos. Por este motivo, se hace un estudio sobre las técnicas que se han utilizado para crear los canales, así como sobre las distintos obstáculos que han tratado de mermar su actividad. Asimismo, se presenta una nueva clasificación que trata de albergar la mayor cantidad de canales encubiertos existentes en la actualidad. Por último, se analiza un protocolo ampliamente extendido en la actualidad, DHCP, en busca de posibilidades de albergar información encubierta. A partir de este análisis se implementan distintas versiones de un canal encubierto haciendo uso de este protocolo.

Palabras clave: Canales Encubiertos, Seguridad en Sistemas de Información, Control de Accesos y Detección de Intrusos, Seguridad en Redes.

1 Introducción

La evolución experimentada por las redes de computadores en los últimos años ha propiciado el desarrollo de nuevos servicios, pero de manera simultánea ha supuesto la aparición de nuevas amenazas para los sistemas que se encuentran interconectados.

Este trabajo se centra en una de las subdisciplinas de la Ocultación de Información, los canales encubiertos o *covert channels*. Esta forma de ocultación ha sido normalmente considerada una amenaza para la seguridad, tanto en sistemas centralizados (ejm. [1, 2]) como en entornos de red (ejm. [3, 4]). Sin embargo, también existe la posibilidad de utilizarlos como alternativa a la criptografía, o para la generación de nuevos servicios [5]. Así pues, los canales encubiertos se definen como una forma de transmitir información oculta (que pase inadvertida a los ojos de un posible observador) aprovechando características del protocolo de comunicación que no se encuentran debidamente definidas. Por tanto, según lo establecido, debería ser posible la consecución de este tipo de canales a todos los niveles de la pila OSI, desde la capa de enlace hasta la de aplicación.

Tradicionalmente, los canales encubiertos han sido agrupados en dos categorías principales: canales de almacenamiento (*storage*) y de temporización (*timing*). Esta clasificación se refiere a la forma en la que se oculta la información a transmitir. En los primeros, el emisor oculta los datos en zonas de memoria a las que el receptor tiene acceso; estas zonas de memoria son ciertos campos en la cabecera de los paquetes de red que o bien están en desuso o que su modificación no afecta al correcto funcionamiento del protocolo. Por otra parte, los canales de temporización se basan en la modulación del comportamiento del emisor para codificar la información, lo cual se traduce, por norma general, en variaciones de la tasa de envío de paquetes.

A pesar de que esta es la taxonomía más conocida, es posible observar en la bibliografía que no existe un amplio consenso entre los diferentes autores. Así pues, han surgido rechazos a esta

clasificación, ampliaciones de la misma por considerarla incompleta, y también nuevos criterios de clasificación.

Por otra parte, existen multitud de herramientas capaces de crear flujos de información oculta entre dos o más hosts. Éstas suelen utilizar protocolos ampliamente utilizados en la mayoría de las redes actuales, como TCP/IP [6], HTTP [7] o DNS [8]. Sin embargo, aún existen multitud de protocolos que no han sido explorados y que podrían servir para la implementación de canales encubiertos.

El resto de este artículo se organiza de la forma que se especifica a continuación. La sección 2 da una visión de la evolución histórica de esta forma de comunicación, desde sus orígenes en sistemas de seguridad multinivel hasta la actualidad. La sección 3 presenta una nueva clasificación de canales encubiertos que trata de abarcar el mayor abanico de canales existentes. En la sección siguiente se realiza un exhaustivo análisis sobre las posibilidades de ocultación en un protocolo de uso muy extendido en las redes actuales, como es el protocolo de configuración dinámica de equipos, DHCP. Asimismo, se presenta una implementación basada en el estudio anterior. Finalmente, la sección 5 albergará las conclusiones así como posibles líneas de trabajo para el futuro.

2 Estado del Arte

Se comenzará proporcionando una visión general del ámbito en el que surgen los canales encubiertos y seguidamente se prestará mayor atención a la evolución experimentada en las redes de computadores.

2.1 Sistemas Multinivel

El concepto de canal encubierto fue introducido por Butler W. Lampson [1] en 1973 para sistemas de seguridad multinivel con la finalidad de señalar las posibilidades que tiene un programa para transmitir información a otro de manera clandestina. Tras Lampson otros autores siguieron utilizando el concepto de canal encubierto, sin embargo, con el tiempo los términos propuestos por éste fueron adquiriendo otros matices. Así por ejemplo, en [9] se comienza a notar que lo que Lampson definió como canal encubierto es lo que más tarde se conocerá como canal de temporización. Sin embargo, Lipner nunca llega a llamarlos así, para encontrar esta distinción habrá que esperar hasta 1977 [10].

Años más tarde, en 1985, el Departamento de Defensa de los Estados Unidos (*DoD*), publica los criterios de evaluación de seguridad para sistemas de computación (*TCSEC*), también conocido como "The Orange Book" [11]. En este documento se clasifican los diferentes sistemas según su nivel de seguridad, para lo cual se tiene en cuenta la presencia de canales encubiertos. Más adelante, en 1993, se publica un nuevo tomo conocido como "The Light Pink Book" [2], que se dedica exclusivamente al análisis de canales encubiertos, presentando diversos métodos para la identificación de estas formas de ocultación (ejm. el método de la matriz de recursos compartidos o *SRM*).

En ese mismo año, Kang y Moskowitz [12] proponen *Pump* para reducir las posibilidades de señalizar información entre procesos pertenecientes a distintos niveles de seguridad mediante el tiempo de llegada de los ACKs. Pump es un dispositivo (sistema de buffers) que se interpone entre los procesos comunicantes e introduce retrasos aleatorios reduciendo las posibilidades de comunicación oculta. Esta idea será más adelante adoptada para combatir canales encubiertos en redes.

Por último, en [13] se dedica un capítulo al análisis y detección de los canales encubiertos, de manera similar a lo propuesto en "The Light Pink Book". Además, se definen los canales

encubiertos desde un punto de vista que podría ser aplicado no sólo a sistemas multinivel sino también a entornos de red.

2.2 Protocolos de Red

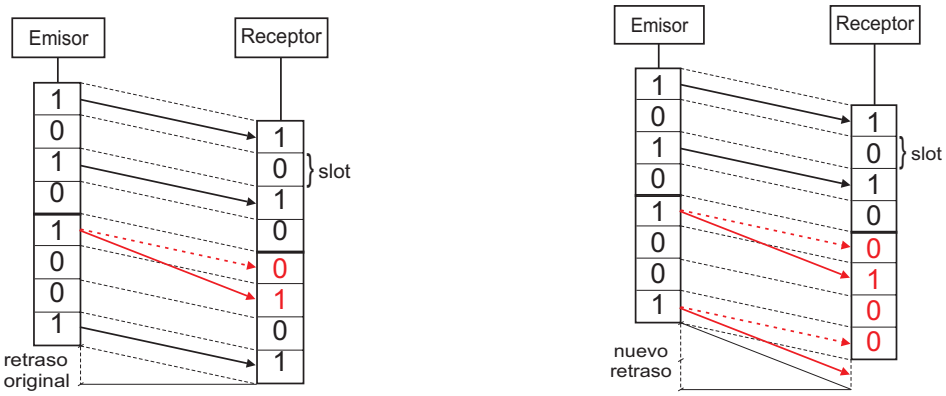
La evolución experimentada por las redes propició la aparición de canales encubiertos también en estos entornos. Así pues, los primeros estudios se dedicaron a analizar la propias características de las redes. En 1987 Girling [3] identifica dos canales de almacenamiento y uno de temporización que podrían ser explotados en la mayoría de las redes convencionales. Estos no son muy sofisticados y su ancho de banda podría verse fácilmente reducido. Sin embargo, este trabajo sirvió para abrir el camino a nuevos estudios como [14], en el que se estudia la familia de protocolos 802.2 hasta 802.5, y [15] en el que se analiza el modelo de referencia OSI en su totalidad.

Todos los estudios hasta la fecha se habían dedicado al análisis desde un punto de vista teórico de las posibilidades de ocultación. Sin embargo, no hubo que esperar mucho más hasta la aparición de las primeras implementaciones. Craig H. Rowland presenta *Covert_TCP* [6], en el que se utilizan tres métodos para ocultar información en la cabecera de TCP e IP. Poco después Phrack Magazine saca a la palestra el proyecto Loki [16], que utiliza el payload de paquetes ICMP para crear su canal encubierto. Estas implementaciones, aunque tienen múltiples deficiencias que pueden alertar de su existencia, fueron las precursoras de muchas otras.

En [17] se presenta una nueva forma de utilizar la cabecera de TCP para enviar información encubierta. En este trabajo se analizan las ventajas e inconvenientes de la utilización de diferentes técnicas para el envío de información oculta dentro de protocolos de red. La utilización de datos de aplicación como portador de esta información requería un conocimiento detallado de la información enviada por éstas para conseguir modificarla sin que levantara sospechas. Este motivo impulsa al uso de campos de la cabecera. Sin embargo, algunos de estos tienen valores característicos determinados por el sistema operativo utilizado y si son modificados de cualquier manera podían ser detectados fácilmente. Por ello se elige como portador a los *timestamps* o marcas de tiempo del campo opciones en TCP, ya que la realización de pequeñas modificaciones en los bits menos significativos puede resultar difícilmente detectables.

Poco después, también en 2002, Kamran Ahsan [18] presenta en la Universidad de Toronto una tesis digna de mención sobre la que posteriormente se basará su artículo *"Practical Data Hiding in TCP/IP"*, publicado a finales del mismo año. En ésta se realiza un análisis de las distintas posibilidades de ocultar información en la pila de protocolos TCP/IP. Para ello se centra en dos enfoques diferentes, la ya conocida manipulación de cabeceras y un nuevo esquema basado en la reordenación de paquetes. El primero de ellos muestra nuevos canales en las capas de transporte y red, en concreto, en los protocolos TCP, IGMP, ICMP e IP. El segundo enfoque se basa en la posibilidad de realizar múltiples ordenaciones diferentes de los elementos de un conjunto, lo cual puede ser utilizado como medio para transportar información; n paquetes pueden transportar $log_2(n!)$ bits de datos. Para tal fin utiliza paquetes IPSec, ya que este protocolo está diseñado para evitar ataques por duplicación de paquetes, con lo que se puede identificar un orden natural de los paquetes, lo cual no ocurre con los números de secuencia de TCP e IP. Otra aportación interesante de Ahsan es que no sólo considera lo canales encubiertos como un medio para transmitir información oculta burlando las políticas de seguridad, sino que además los plantea como una posibilidad para mejorar la efectividad de la propia red y de los servicios que en ella se prestan aprovechando un ancho de banda que en la actualidad está en desuso.

Dos años más tarde se presenta la primera implementación de canales de temporización [19]. Los autores presentan el diseño del protocolo creado para llevar a cabo la comunicación, así como



(a) Errores producidos por un retraso eventual

(b) Errores producidos por un retraso duradero

Fig. 1: Errores producidos en canales de temporización a causa de diferentes retrasos en la red

los problemas que tuvieron que afrontar debido a la ausencia de un reloj común de precisión que suministrara el sincronismo necesario. Para hacer frente a los problemas de sincronización tuvieron que hacer uso de diferentes mecanismos dedicados. Sin embargo, a pesar de todas las técnicas utilizadas, los resultados demuestran que el estado de la red resulta determinante en la correcta recepción de los datos (véase Figura 1) o el ancho de banda del canal se ve gravemente reducido.

Antes de que se haya convertido en estándar de facto y haya conseguido desbancar a la versión 4, en la versión 6 del protocolo IP se han detectado 22 posibles canales encubiertos [20]. En la cabecera se identifican 6 canales de almacenamiento y los 16 restantes se reparten entre las 6 cabeceras de extensión del protocolo. Algunos de estos canales son bastante simples de eliminar o detectar incluso con cortafuegos tradicionales, sin embargo, lo que resulta interesante es que se hayan señalado tal número de vulnerabilidades en un protocolo que aún no ha sido completamente implantado.

Poco después, en [5] se presenta un protocolo que utiliza canales encubiertos y técnicas de *watermarking* (marcas de agua) para proveer de nuevas características de seguridad a una tecnología que está teniendo una gran aceptación en Internet, voz sobre IP (*VoIP*). Las mayores ventajas de este sistema son que además de proporcionar autenticación e integridad a la comunicación no necesitan utilizar ancho de banda adicional sino que hacen uso de campos en desuso, como propuso [18].

Ese mismo año, en Agosto de 2006, ve la luz un estudio que crea canales de temporización a partir de un dispositivo al que se bautiza con el nombre de *Jitterbug* [21]. Estos pueden ser tanto software como hardware y su finalidad es la de reconocer información sensible (ejm. passwords) y modularla a través de la red. Son mecanismos semipasivos que no generan nuevos eventos sino que utilizan otros para transmitir la información. En este caso Shah et al. implementan un jitterbug hardware para teclado. Éste se aprovecha de que la mayoría de las aplicaciones de red interactivas (ejm. telnet) envían un paquete de datos tras cada pulsación de teclado realizada. Así pues, jitterbug añadirá pequeños retrasos, inapreciables a nivel de usuario, a las pulsaciones de teclado para retrasar a su vez el envío de los respectivos paquetes. Estos retrasos serán observados por la entidad receptora con el fin de recuperar la información oculta (véase Figura 2).

No todas la implementaciones señaladas están disponibles para hacer uso de ellas. Algunas implementaciones de interés que se encuentran disponibles son *Firepass* [7], *PingTunnel* [22] y *Ozyman* [8]. Éstas crean canales encubiertos utilizando protocolos básicos para el

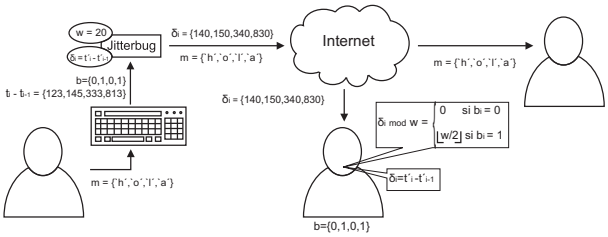


Fig. 2: Ejemplo de uso de un dispositivo jitterbug para teclado.

funcionamiento de las redes actuales, HTTP, ICMP y DNS respectivamente.

2.2.1 Métodos de prevención/eliminación

Al mismo tiempo que iban surgiendo estudios sobre técnicas de ocultación, otros autores creaban métodos o protocolos para evitar sus posibles efectos nocivos. Antes de que surgieran las primeras implementaciones, estos estudios se centraron en los canales de temporización ya que tenían gran similitud con la mayoría de canales identificados en sistemas de seguridad multinivel. Así por ejemplo, cabe destacar el estudio realizado por la Universidad de Arizona [23] en el que se analizan, tanto de forma teórica como práctica, varios métodos existentes y se propone uno nuevo.

Con la aparición de las primeras implementaciones de canales encubiertos, comienza un nuevo resurgimiento de los artículos dedicados a detección y prevención, dedicándose especial atención al estudio de casos concretos.

Así pues, en el año 2002, se publica un artículo de gran interés en lo relativo a la eliminación de canales encubiertos. Fisk et al. [24] presentan la primera implementación de "Active Warden" para redes, casi dos décadas después de que fueran definidos por primera vez por Simmons en su Problema de los Prisioneros. Esta implementación se basa en el análisis de las cabeceras de los protocolos, dejando a un lado el estudio de portadores no estructurados, como pueden ser fotografías. La técnica utilizada por este guardián consiste en la introducción de ruido en el portador, lo cual podría afectar gravemente a la información de usuario si se aplicase a la zona de datos de los paquetes.

En [19] se dedica parte del trabajo no sólo a la creación de un canal de temporización sino a investigar métodos que permitan la detección de dichos canales. En concreto se presentan dos métodos basados en el tiempo interpaquete, ya que al estar los envíos regidos por temporizadores deben seguir un patrón de llegada regular, lo cual no ocurre con el tráfico normal generado por usuarios. Ambos métodos fueron utilizados con éxito para canales sencillos, sin embargo para canales más complejos, con ruido y que realizaban pausas con el fin de imitar el funcionamiento normal de un protocolo, ambos métodos cometían errores.

Al año siguiente, Murdoch y Lewis [25] critican la forma tradicional de utilizar las cabeceras para crear canales de almacenamiento sobre TCP e IP. Éstas solían rellenarse con información aleatoria sin tener en cuenta que los campos utilizados contienen unos valores dependientes del sistema operativo en que hayan sido creados. Así pues, desarrollan 14 tests, de los cuales 4 están dedicados a detectar anomalías en el campo de identificación IP, 7 al número de secuencia de TCP y los 3 restantes a la detección de otras anomalías.

3 Clasificación de Canales Encubiertos

Aunque existen multitud de trabajos dedicados al análisis de los canales encubiertos, no existe una nomenclatura común que sea ampliamente aceptada sino que los diferentes autores han hecho uso de las taxonomías que más convenían a su estudio. Así pues, a partir de éstas se desarrollará una nueva categorización que se ajuste al estado actual de la tecnología.

Lampson [1] además de introducir el concepto de canal encubierto propone la primera taxonomía. Aunque ésta no puede considerarse estrictamente una clasificación, pues lo que Lampson llama canal encubierto será más adelante conocido por otros como canal de temporización. Esta tendencia comienza a tomar forma en el trabajo de Lipner [9], aunque será en [10] donde se encuentre la primera referencia a una clasificación que realice una distinción clara entre canales de almacenamiento y temporización, dejando de lado otra categoría propuesta en el trabajo de Lampson, los canales legítimos.

En el Libro Naranja (TCSEC) se sigue haciendo uso de la clasificación en canales de almacenamiento y temporización. Sin embargo, según Virgil D. Gligor [2], no existe una clara distinción teórica entre estos canales. La razón para rechazar esta categorización es que en cualquier forma de comunicación las partes implicadas requieren algún mecanismo de sincronización para que ésta tenga éxito.

Así pues, Gligor propone dos nuevas formas de clasificar los distintos canales. La primera de ellas está basada en la cantidad de ruido que afecta al canal y tiene que ver, por tanto, con la fiabilidad del mismo:

- Canales sin ruido (*Noiseless Channels*): se dice que un canal es sin ruido si los símbolos que son transmitidos por el emisor son los mismos que los recibidos por el receptor con probabilidad 1.
- Canales con ruido (*Noisy Channels*): son canales en los que existe cierta probabilidad de que cualquier símbolo enviado por el emisor sea recibido de manera incorrecta por el receptor.

Aunque creemos acertado el realizar una distinción entre canales con y sin ruido, consideramos que este tipo de clasificación no es apropiada en el ámbito de las redes. La aplicabilidad de esta división parece propia de estudios sobre la capacidad máxima de los canales.

La segunda de las taxonomías propuestas se basa en la cantidad de procesos que se comunican simultáneamente mediante la utilización de zonas de memoria (variables):

- Canales desagrupados (*Non-Aggregated Channels*): son aquellos en los que las variables utilizadas para la creación de un canal encubierto son accedidas únicamente por un par de procesos, el emisor y el receptor.
- Canales agrupados (*Aggregated Channels*): son canales creados por varias parejas de procesos que acceden a un mismo conjunto de variables. Es posible distinguir:
  - En serie: un proceso P_i accede a su variable o grupos de variables y hasta que éste no ha terminado, el proceso siguiente, P_j, no obtendrá el control.
  - En paralelo: varios procesos acceden simultáneamente a las variables sin necesidad de esperar a que otro proceso les ceda el control.
  - Mixtos: surgen de la combinación de canales agrupados en serie y paralelo.

Años antes, Wray [26] también cuestiona la validez de la clasificación clásica ya que establece que en ocasiones existen canales en los que no se observa una clara distinción entre canales

de almacenamiento y de temporización porque poseen aspectos de ambas categorías. Además, Wray establece otra división con canales de proceso (*Process Channels*), donde la parte receptora del mensaje es un proceso ejecutándose en el sistema y canales directos (*Direct Channels*) en los cuales la información transmitida por el canal es volcada directamente a un dispositivo de salida, por ejemplo un monitor.

En [27] se define una categoría de canales encubiertos que se denominan canales discretos sin memoria (*DMC*). Por "discreto" se hace referencia a que el número de elementos del alfabeto utilizado es finito, mientras que por "sin memoria" se quiere decir que no hay restricciones en qué símbolo puede ser transmitido basándose en un estado anterior del canal. Así pues, existen tanto DMCs de almacenamiento como de temporización. Además, en Mayo de ese mismo año, en [28] se definieron los canales de temporización simples (*STC*), que son DMCs de temporización sin ruido. Estos tipos de canales fueron definidos principalmente con el fin de simplificar los estudios realizados sobre la capacidad de los canales encubiertos.

También en 1994, James W. Gray [29] identifica un tipo de canal de temporización que se aprovecha del número de ocurrencias de determinados eventos para codificar la información, por ello recibe el nombre de canal de contabilización (*Counting Channel*). Es decir, se trata de una clase de canal de temporización en el que lo relevante no es el tiempo que tarda en tener lugar un determinado evento sino el número de eventos que tienen lugar en un determinado intervalo de tiempo.

Un año más tarde se propone, en [30], una nueva distinción:

- Canales espaciales (*Spatial Channels*): son aquellos que utilizan como medio para transmitir información las variaciones en el volumen de datos que se comunican entre parejas de equipos de la red.

- Canales temporales (*Temporal Channels*): son creados a partir de variaciones en las características de la transmisión a lo largo del tiempo. Es decir, manipulan el tamaño de los paquetes, el orden, la frecuencia o incluso la duración de la transmisión con el fin de crear un canal de datos oculto.

Nótese que no existe una relación directa entre canales espaciales y canales de almacenamiento, ni tampoco entre canales temporales y de temporización. La única mención que los autores hacen a los canales de almacenamiento y de temporización es para referirse a la transmisión entre procesos dentro de un mismo sistema, por tanto, da la impresión de que no consideran apropiada tal taxonomía para las comunicaciones en redes.

Si bien es cierto que la clasificación propuesta parece comprender un conjunto de canales no englobados en la clasificación clásica, parece posible establecer una relación entre estos. Así pues, los canales espaciales podrían considerarse como una especie de canal de contabilización (temporización) en el que la información se oculta en el número de eventos que tienen lugar en un determinado intervalo de tiempo, que en este caso se corresponde con el número de paquetes enviados. Los canales temporales aquí propuestos, desde nuestro punto de vista podrían ser considerados también como canales de temporización, ya que se refiere a la variación de ciertas características a lo largo del tiempo.

Un año más tarde, Meadows y Moskowitz [31] proponen una nueva clasificación basada en el contexto en el que tienen lugar los canales encubiertos. Así pues, identifican canales de servicio alto a bajo (*high-to-low*), de servicio bajo a alto (*low-to-high*), de servicio compartido (*shared service*) y de servicio incomparable (*incomparable service*).

La mayor ventaja que presenta esta clasificación, según los autores, es que aquellos canales pertenecientes a una misma clase pueden ser tratados de forma similar. Sin embargo, lo que Meadows y Moskowitz consideran el mayor mérito de su nueva clasificación puede resultar algo

familiar. Por ejemplo, existe un amplio abanico de métodos para reducir el ancho de banda de los canales de temporización clásicos, entre los cuales será posible elegir el que mejor se ajuste a una situación específica. Así pues, desde nuestro entender, lo más interesante de esta clasificación es que propone un cambio de perspectiva. La categorización no se basa en las técnicas utilizadas para transmitir información sino en el ámbito en el que los canales tienen lugar, y por ello la consideramos digna de mención.

Por otra parte, Kamran Ahsan [18] propone una nueva forma de canal encubierto conocida como canal de ordenación (*Sorting Channel*). Se trata de un canal que oculta la información en las distintas ordenaciones posibles de los paquetes que atraviesan la red. Este tipo de canales son considerados por [19] como canales de temporización por ocultar la información en el patrón de llegada de los paquetes. Sin embargo, consideramos que este tipo de canal no debe ser considerado como tal ya que la forma de llegada de los paquetes no está regida por temporizadores. Incluso las técnicas de reducción del ancho de banda para canales de temporización, como Pump, no tendrían ningún efecto sobre este tipo de canales ya que no modifican el orden de los paquetes.

En Septiembre de 1995, Wang y Lee [32] presentan una nueva clasificación para los distintos tipos de canales encubiertos, aunque el enfoque utilizado no es completamente nuevo. Esta clasificación se basa en la dimensión en la que se codifican los datos, es decir, espacio o tiempo. Aunque esto pueda parecer una repetición de la clasificación tradicional, los autores dan un paso más y capturan en ésta el paradigma utilizado para realizar la codificación de los datos. Así pues, la taxonomía propuesta incluye canales espaciales (almacenamiento) y temporales (temporización) que a su vez pueden estar basados en valor o en transición. Esto indica la posibilidad de ocultar información no sólo gracias al valor de una zona de datos, sino en la modificación o transición de un valor a otro.

3.1 Una Nueva Clasificación

En la sección 3 se han tratado tanto clasificaciones creadas para sistemas multinivel como para entornos de red. Las primeras pueden ser en la mayoría de casos aplicables también a sistemas en red y, por tanto, la clasificación aquí propuesta tomará aspectos de ambas.

Así pues, se proponen dos posibles criterios de clasificación de los canales encubiertos: según el número de procesos que intervienen en la comunicación y según la forma de ocultar la información. En el primer caso es posible distinguir:

- Canales desagrupados: son aquellos canales encubiertos creados para el envío de información entre, únicamente, dos equipos de la red.
- Canales agrupados en serie, paralelo o mixtos: se refiere a los canales de comunicación ocultos en los que la información transmitida puede ser aprovechada por más de una pareja de equipos de la red. Así, el tipo de agrupación podría depender de la topología de la red, por ejemplo, las redes en anillo sólo permitirían el agrupamiento en serie, las redes en bus posibilitarían además la agrupación en paralelo y las redes en árbol podrían dar lugar a agrupaciones mixtas.

El segundo criterio, que se basa en el medio que se utiliza para ocultar la información, daría lugar a:

- Canales de almacenamiento: son aquellos que utilizan una zona dentro del paquete, bien la cabecera o bien la zona de datos.
  - Basados en valor: son canales que codifican la información en el propio valor de una zona determinada de los paquetes enviados.

  - Basados en transición: son aquellos canales en los que la información se codifica en las variaciones de un campo concreto de los paquetes. La información contenida no es el mensaje en sí, sino que éste viene dado por el cambio de un valor a otro.

- Canales de temporización: son aquellos que ocultan la información en los patrones de llegada de los paquetes al receptor. La recepción o no de paquetes está regida por relojes, lo cual codifica la información.

  - Canales de contabilización: son canales de temporización que codifican la información en la cantidad de eventos que tienen lugar durante un determinado periodo de tiempo.

- Canales de ordenación: son aquellos canales que ocultan la información en el orden de llegada de los paquetes.

- Canales combinados o híbridos: son aquellos canales que resultan de la combinación de las técnicas empleadas por algunos de los tipos anteriores.

Estos criterios de clasificación no son excluyentes. Así, por ejemplo, sería posible la existencia de un canal agrupado de contabilización.

4 Un Nuevo Canal Encubierto: Covert_DHCP

En esta sección se plantea un escenario de uso ficticio a partir del cual se desarrolla una nueva forma de comunicación oculta basada en un protocolo que hasta el momento no había sido explotado para tales fines. Tras la identificación de las necesidades se realiza un análisis sobre las posibilidades del protocolo y se desarrollan e implementan distintos métodos de ocultación de información: Covert_DHCP.

4.1 Escenario Ejemplo

Supongamos, por ejemplo, un par de individuos, *Alicia* y *Benito*. Alicia, trabaja y tiene acceso privilegiado a parte de una red dentro de una embajada en un país en el que la libertad de expresión se encuentra limitada o se quiere comunicar una información de extrema importancia de la que nadie debe saber su existencia, como el día de comienzo de una guerra. Supongamos, además, que Benito se encuentra en la embajada debido, por ejemplo, a una cumbre internacional y que a estos individuos no se les puede ver comunicarse físicamente o se levantarían sospechas de que tienen algo entre manos.

A partir de este escenario es posible identificar las siguientes propiedades para el canal:

- Dificultad de identificar la existencia del canal. Esta característica se ve facilitada por el hecho de hacer uso de un protocolo de comunicación que no ha sido utilizado con tales fines anteriormente.

- Ancho de banda moderado. La capacidad del canal no se considera un factor esencial ya que la intención es el envío de pequeñas cantidades de información, como por ejemplo, la comunicación de una clave criptográfica.

- Robustez. Si los datos enviados son sensibles, como puede ocurrir en el envío de claves criptográficas, es necesario que estos sean recibidos correctamente.

- Actuación entre equipos próximos. Nuestro objetivo no es el de realizar envíos a través de Internet, sino más bien en redes LAN o PAN.

TABLA 1: Tipos de Mensaje DHCP

Paquete	**Sentido**	**Descripción**
Discover	C → S	Mensaje de broadcast para localizar servidores.
Offer	C ← S	Mensaje de respuesta al DHCPDiscover, que contiene una oferta de configuración.
Request	C → S	Mensaje que confirma la aceptación de los parámetros ofrecidos por el servidor o de renovación de una configuración previa.
Ack	C ← S	Mensaje de aceptación de lo acordado con el cliente, que incluye tales parámetros acordados, además de la dirección IP.
Nak	C ← S	Mensaje indicando al cliente que no acepta la configuración, bien porque la dirección no es correcta o porque su contrato ha expirado.
Decline	C → S	Mensaje que indica que la dirección de red ya está asignada a otro cliente de la red.
Release	C → S	Mensaje que sirve para rechazar la dirección asignada, cancelando el contrato actual.
Inform	C → S	Mensaje solicitando más información sobre la configuración; el cliente ya posee dirección IP.

- Sentido de la comunicación. Basta con crear un canal unidireccional pues el propósito no es realizar un intercambio de datos.

Estas propiedades permiten elegir a DHCP como canal para ocultar información. Para ello se analizará el protocolo en busca de posibilidades de ocultación.

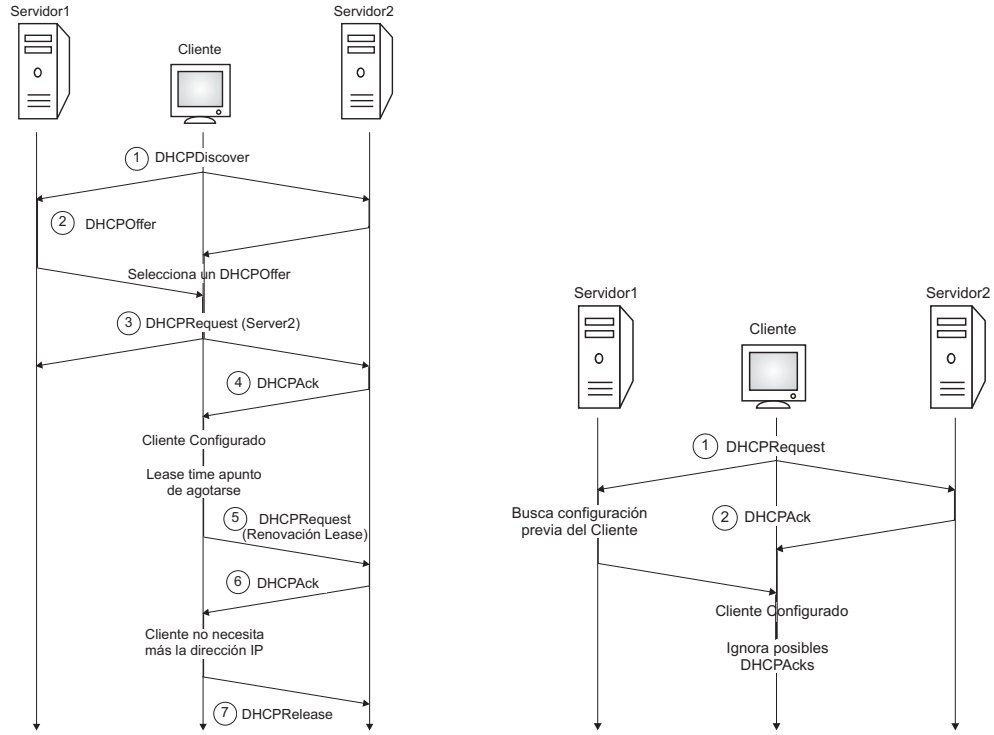
4.2 Análisis

El protocolo de configuración dinámica de equipos (*Dynamic Host Configuration Protocol*) puede considerarse una extensión del protocolo BOOTP (*Bootstrap Protocol*). Es un protocolo de nivel de aplicación que trabaja sobre UDP, utilizando los puertos 67 (servidor) y 68 (cliente). A pesar de utilizar un servicio de datagramas para llevar a cabo las labores que tiene asignadas no es habitual la pérdida de mensajes entre cliente y servidor puesto que el ámbito de trabajo suele encontrarse dentro de la propia red de área local. A pesar de ello proporciona algunos medios de recuperación frente a la pérdida de paquetes, como el reenvío de éstos si no recibe respuesta tras un tiempo determinado.

Se trata de un modelo de petición-respuesta en el que el cliente es el encargado de iniciar la comunicación. La interacción entre el cliente y el servidor tiene lugar mediante el intercambio de los mensajes que se muestran en la Tabla 1.

Existen dos modelos de intercambio de mensajes. El primero tiene lugar la primera vez que el cliente trata de obtener una configuración de red, o si el contrato (*lease*) ha dejado de ser válido (véase Figura 3a). El segundo modelo se utiliza si el contrato sigue vigente y se está tratando de renovar tal configuración con el servidor (véase Figura 3b). La secuencia normal de mensajes es: Discover, Offer, Request, Ack y, posiblemente, Release. Los dos primeros sólo tienen cabida en el primer modelo.

Todos los mensajes comparten la misma estructura, ya procedan del servidor como del cliente (véase Figura 4). La primera posibilidad que se presenta es el *xid* o identificador de



(a) Designación de una configuración. (b) Renovación de una configuración previa.

Fig. 3: Modelos de Intercambio de Mensajes entre cliente y servidor DHCP.

transacción, que con su longitud de 32 bits puede ser un importante portador de información. Lo más interesante es que según la especificación este valor será generado por el cliente de forma aleatoria. Esto supone que no existen algoritmos prefijados, como ocurre con el generador de números de secuencia en TCP, hecho que facilitaría el descubrimiento del canal de datos oculto.

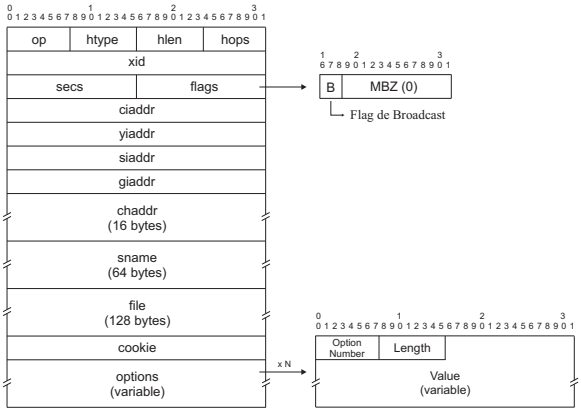


Fig. 4: Estructura de los Paquetes DHCP.

El campo *secs*, con sus 2 bytes, podría ser utilizado de forma similar a lo propuesto en [17] para ocultar información en la opción Timestamp de TCP. La ocultación de información sin hacer uso de una técnica como la propuesta en ese trabajo podría levantar sospechas o hacer que un servidor determinado no funcionara correctamente al recibir nuevos mensajes "atrasados" en el tiempo. Por otra parte, al ser un protocolo que hasta el momento no ha sido explotado para la creación de canales encubiertos, los sistemas detectores de intrusos probablemente pasarían

por alto este tipo de detalles.

El campo *chaddr* presenta al menos dos posibilidades diferentes. En primer lugar se podría hacer uso de una técnica semejante a la utilizada por Rowland en el método 3 de Covert_TCP, utilizando servidores DHCP de rebote. Para ello se introduciría la dirección MAC de la entidad con la que se quiere contactar en el campo *chaddr*, y en otro, por ejemplo el *xid*, ya que se mantiene inalterado durante toda la transacción, se introducirían los datos que se desean transmitir. Esto provocaría que el servidor DHCP respondiera, en lugar de al emisor del mensaje, a aquel cuya dirección hardware se encuentra especificada en el mensaje que recibió. El inconveniente se presenta en que esto podría resultar sospechoso en un análisis del datagrama si la MAC real es diferente a la que aparece en este campo, aunque sería posible utilizar la misma dirección MAC del destinatario sin provocar incovenientes en el funcionamiento de la red. Esto podría ser igualmente detectado por un IDS con reglas para detectar técnicas de *MAC spoofing*. La otra oportunidad se debe al tamaño del campo en cuestión. Se trata de un campo de 16 bytes cuando en la mayoría de ocasiones sólo se utilizan 6 bytes (tarjetas de red Ethernet). En este caso podría aprovecharse que el tamaño de chaddr viene determinado por *hlen* y ocultar información en los bytes restantes, que en el caso de Ethernet asciende a 10 bytes.

Los campos *sname* y *file* por su gran tamaño, 64 y 128 bytes respectivamente, son potencialmente excelentes portadores de información. Ambos contienen una cadena acabada con el carácter terminador; así pues, podrían introducirse nuestros datos tras este carácter, de manera que un cliente o servidor sin modificar creyese que tales campos no contienen información relevante para ellos. Aunque los bytes de ambos campos suelen ser nulos cuando no llevan sus propios datos, existe la posibilidad de que alberguen opciones, que por razones de tamaño del paquete no pueden ser añadidas a la zona dedicada para ello. En tal caso es necesario incluir una opción indicándolo, ésta es la opción 52, que recibe el nombre de *"Overload"*. El principal problema para utilizar dichos campos como portador de información es que normalmente, aunque no se hace ninguna referencia en la especificación del protocolo, suelen tener el valor cero cuando la opción *Overload* no está activa; lo cual puede resultar sospechoso ante analizadores del tráfico DHCP.

Por último, también el campo *options* presenta características interesantes que podrían ser aprovechadas para ocultar información. En primer lugar, el hecho de que sea un campo de longitud variable proporciona la posibilidad de ocultar información, bien en el número de opciones utilizadas (lo cual puede verse determinado por el tamaño del paquete) como en la ordenación que éstas presentan, o bien en el número de opción (entre 0 y 255) de una o varias opciones. Por ejemplo, supóngase que se pretende codificar la cadena de caracteres "HOLA", podría optarse por alguna de las formas siguientes:

1. En este caso, por simplicidad, podría suponerse que solamente se enviarán caracteres en mayúsculas, delimitando en gran medida el número de opciones que sería necesario incluir. Así por ejemplo, en lugar de utilizar una codificación ASCII usual podría utilizarse una codificación propia, de tal manera que la letra 'A' se correspondiese con el valor 2 (ya que al menos debe aparecer la opción *"DHCP Message Type"* y la *"End"*), la 'B' con el 3 y así sucesivamente. Por tanto, para enviar el mensaje "HOLA" sería necesario enviar 4 mensajes, el primero con 9 opciones ('H'), el segundo con 16 ('O'), el tercero con 13 ('L') y el último con 2 ('A').

2. La ordenación de las opciones dentro de un mismo mensaje, y no de diferentes paquetes, como propuso Kamran Ahsan, podría servir como medio de ocultación de información. Si se tiene un alfabeto de 255 caracteres (ejm. ASCII), será necesario tener la capacidad de codificar 8 bits. Por tanto, debería tenerse un mensaje con $n = 8$ opciones y con una ordenación determinada, que sirva para comparar con los mensajes recibidos. Así pues,

para cada opción del mensaje recibido, si la opción ocupa la misma posición que en un mensaje de referencia, se considerará que tal bit está a 1 y en caso contrario será 0. Por ejemplo, supóngase que tenemos cuatro opciones A, B, C y D en ese orden y se recibe un mensaje con las opciones en el orden C, B, A, D; entonces el mensaje oculto sería 0101.

3. Supóngase que el número de la opción que se encuentra en segunda posición codifica un valor entre 0 y 255, es decir, 8 bits. Para enviar el mensaje "HOLA" sería necesario enviar cuatro mensajes y en cada uno de ellos la segunda opción debería contener respectivamente la opción número 72 (*"WWW-Server"*), la número 79 (*"Service Scope"*), la 76 (*"STDA-Server"*) y la 65 (*"NIS-Server-Addr"*). También sería posible disminuir el número de mensajes enviados y al mismo tiempo aumentar el ancho de banda utilizando, en lugar de una única opción encargada de codificar 1 carácter hacer uso de varias en un mismo mensaje. Por ejemplo, codificar los 4 caracteres de "HOLA" en las opciones 2, 3, 4 y 5.

La dificultad que presentan estas soluciones estriba en que dependiendo del tipo de mensaje existen una serie de opciones que son necesarias o que no están permitidas. La posibilidad de implementar cualquiera de éstas dependería de si la introducción de opciones no permitidas en ciertos paquetes llegaría a influir en el buen funcionamiento del protocolo. Otro inconveniente, específico de la tercera forma de ocultación, es que el intento de codificar mensajes con caracteres repetidos implicaría la aparición de opciones repetidas, lo cual aunque es posible en ciertos casos, podría levantar sospechas. Sin embargo, esto no sería difícil de solventar mediante el envío de caracteres siempre y cuando el siguiente carácter a enviar no se encuentre codificado previamente en ese mensaje DHCP. Así por ejemplo, para el mensaje "AHORA", habría que enviar al menos dos mensajes DHCP, conteniendo "AHOR" y "A".

Otra de las características del campo opciones que puede ser aprovechada es la existencia de ciertas opciones que bien no están definidas o lo están para uso privado. El uso de estas opciones (84, 96, 102-111, 115, 126, 127, 137-149, 151-174, 178-207, 212-219, 222 y 223 han sido eliminadas o están sin asignar; 224-254 se definen para uso privado), principalmente las definidas para uso privado, puede ser de gran interés ya que al no encontrarse definidas, cualquiera puede especificar su formato, de tal manera que el campo *value* albergue tanta información como sea necesario, hasta 255 bytes.

De entre todas las opciones de ocultación analizadas en DHCP, se ha decidido implementar tres de ellas, utilizando los campos *xid*, *Sname* y *File* y el campo *Options*. Mostraremos en las subsecciones sucesivas dos de éstas por cuestiones de espacio y legibilidad.

4.3 Implementación basada en xid

Se parte de un código existente desarrollado por el consorcio de sistemas de Internet (*ISC*) en su versión 3.1.0 [33]. Este se encuentra incluido en muchas distribuciones Linux, como Debian. Además, está escrita en lenguaje C.

En nuestra implementación, los procedimientos de mayor interés para la creación del canal basado en xid son *state_reboot* y *make_discover*, ya que en estos se modifica el xid del paquete. En la versión original del ISC, éste es cargado mediante la función *random()*.

El cliente DHCP modificado permite que el usuario pueda decidir si utilizar el canal encubierto en la solicitud de una configuración de red. Para que el servidor DHCP sea capaz de determinar que el xid que está recibiendo contiene información oculta se utiliza un delimitador de inicio y fin de transmisión. Este se encuentra predefinido en un fichero de cabecera lo que permite que sea fácilmente modificado, dificultando así su detección.

El servidor utiliza una lista con los posibles clientes que pueden enviarle información oculta. Esto unido a la llegada del delimitador hará que el servidor comience a almacenar la información

recibida. Este procedimiento se realiza en el momento que se recibe el DHCPRequest del cliente por ser un punto común en los dos modelos de intercambio de mensajes posibles en DHCP (véase Figura 3).

Asimismo, el cliente debe tener constancia de quién es el servidor con el que desea contactar pues podrían coexistir varios servidores en una misma red. Es decir, si el servidor DHCP que responde como consecuencia del mensaje DHCPDiscover no es el servidor deseado, no iniciará la ocultación de información en el canal DHCP. Sería inútil e incluso facilitaría la detección del sistema el hecho de enviar información a un servidor que no esperara tales datos.

4.3.1 Ventajas e Inconvenientes. Posibles Mejoras

El principal inconveniente que es posible apreciar es su reducido ancho de banda. Este inconveniente se ve aumentado por la necesidad de controlar la repetición de paquetes DHCP[1]. A pesar de tratarse de un campo de una extensión razonable para el envío de información (4 bytes), la capacidad del canal se ve limitada por el hecho de que el campo en cuestión es utilizado para identificar a todos los paquetes pertenecientes a una misma transacción, lo que significa que su valor no puede variar en cada uno de los paquetes enviados.

Cada transacción tiene lugar, por lo general, transcurrido un tiempo alrededor de la mitad de la duración del contrato en vigencia (sin tener en cuenta paquetes DHCPInform, que en este caso no han sido utilizados). El tiempo de vigencia del contrato suele estar determinado por el servidor DHCP. Sin embargo, es responsabilidad del cliente renovar tal contrato y es éste quien decide en qué momento llevar a cabo tal proceso de renovación. Por tanto, el tiempo transcurrido entre dos transacciones consecutivas puede ser modificado, no obstante, el establecimiento de nuevas transacciones excesivamente cercanas en el tiempo podría ser utilizado como medio para la identificación de canales encubiertos sobre DHCP.

Con el fin de mejorar el ancho de banda del canal se podría tratar de modificar la aplicación de tal manera que hasta que no se consiguiera enviar toda la información, o al menos gran parte de ella, el cliente no consiguiera una configuración. Es decir, se trataría de simular la pérdida de paquetes, o de la cobertura en caso de estaciones portátiles conectadas a través de un interfaz de red inalámbrico. Además, esto podría propiciar la creación de un canal de comunicación en ambos sentidos, de cliente a servidor y viceversa. Esto no es posible en la versión actual puesto que nuestra implementación se basa en que el cliente es el único creador del campo xid y éste debe permanecer inalterado durante una transacción completa. Sin embargo, al simular la existencia de circunstancias adversas en la red, que dificulten la comunicación, podría darse lugar a la aparición de paquetes procedentes del servidor DHCP con un valor de xid inesperado en un momento determinado. En cualquier caso, no se considera necesaria la creación de un canal con estas características según el escenario de uso propuesto en la sección 4.1.

Por otra parte, la cantidad de tráfico DHCP de una red suele verse influenciada por el número de usuarios que pueden acceder a ésta. Es decir, una red que cuente con un servidor DHCP y sea susceptible de acoger a muchos invitados diferentes tendrá un mayor número de solicitudes de configuración. Además, si la cantidad de invitados es muy amplia, como puede ocurrir en una red universitaria, el tiempo de duración del contrato deberá ser menor para liberar aquellas direcciones IP que no se encuentren en uso, y poder asignarlas a otros usuarios que la requieran. Esto posibilita un mayor número de transacciones y, por tanto, un aumento de la capacidad de envío del canal encubierto desarrollado. Del mismo modo, supone un mayor trabajo para un posible sistema detector de intrusos que tenga que analizar este tipo de tráfico.

[1]Según la especificación de DHCP es posible la repetición de paquetes (y por tanto del xid). Para solventar este problema puede utilizarse uno de los bytes del xid para indicar el número de secuencia del mismo y hacer uso de una lista con el número de secuencia actual de cada cliente encubierto. Esta solución, sin embargo, reduce a tres el número de bytes disponibles para enviar información.

Las mayores ventajas de la creación de un canal encubierto utilizando el identificador de transacción se encuentran en la naturaleza aleatoria de este campo en las implementaciones actuales y el amplio uso de este protocolo, que se encuentra implementado en la inmensa mayoría de redes, tanto en equipos dedicados como en puntos de acceso.

El uso del delimitador de inicio y fin de transmisión, además de reducir levemente la capacidad del canal, puede considerarse un posible medio de detección del canal. Sin embargo, el usuario del canal puede decidir modificar el delimitador a su antojo para hacer más difícil la detección. Además, el hecho de eliminar el delimitador de fin de transmisión en favor del uso del delimitador de inicio junto con el tamaño de la transmisión no supondría una mejora en cuanto a la detectabilidad, e imposibilitaría el envío de información de la que se desconociera su tamaño a priori.

4.4 Implementación basada en Options

Aunque ya se señalaron múltiples posibilidades para la ocultación de información mediante el uso del campo opciones (véase Sección 4.2), en este caso se ha optado por la utilización de aquellas opciones que se encuentran definidas dentro del protocolo como opciones de uso privado, es decir, las pertenecientes al rango 224 - 254. La elección se debe principalmente a que, de entre las posibilidades analizadas, es la que puede proporcionar un ancho de banda más considerable, paliando así el principal inconveniente encontrado en la implementación basada en xid.

A pesar de que el rango de opciones elegido se encuentra definido para uso privado y, por tanto, se considera que cada organización puede utilizarlo como mejor le convenga, durante la realización de esta implementación se han detectado, mediante el uso de un analizador de tráfico, al menos dos opciones que son ampliamente utilizadas, aunque de manera no oficial. La primera de éstas se trata de la opción 249 o *"Classless static routes"*, y la segunda es la opción 252 o *"Proxy Autodiscovery"*. Esta última es más común al ser utilizada por el protocolo de descubrimiento del proxy de Web (*WPAD*, **W**eb **P**roxy **A**uto**d**iscovery), que es aprovechado por conocidos exploradores, como Microsoft Internet Explorer, para localizar un fichero en el que se indican los parámetros de configuración del servidor proxy.

En esta versión del código no ha sido necesario el uso de delimitadores que indicaran el inicio y el fin de la transmisión encubierta. Esto se debe a que se ha considerado como indicador de la existencia de una transmisión el mero hecho de que aparezca la opción 224 (en nuestro caso DHO_COVERT_CHANNEL) entre las incluidas en el campo de opciones. Se ha optado por un tamaño de 255 bytes por ser el máximo permitido por cada opción. En caso de que el tamaño de los datos fuese mayor del autorizado se incluirían nuevas opciones con el mismo "tag" (número de opción), hasta completar el tamaño máximo del paquete. En este momento se comenzaría a incluir nuevas opciones dentro del espacio reservado para los campos sname y file, en caso de no estar siendo utilizados.

El servidor DHCP tiene la misión de comprobar si el paquete DHCP que recibe contiene información oculta en el campo de opciones. Para ello procesa el campo de opciones en busca de la opción DHO_COVERT_CHANNEL.

La Figura 5 muestra una captura de pantalla de un cliente DHCP que se encuentra enviando un fichero con un tamaño de 1024 bytes utilizando como portador el campo de opciones. Para ello realizará el envío mediante diversos paquetes que contendrán un máximo de 255 bytes cada uno.

El servidor se encuentra en otra ubicación recibiendo simultáneamente datos del cliente anterior y de otro cliente adicional, que le envía una cantidad de información que es posible albergar en un único paquete de datos. Como podemos comprobar, se transmiten datos tanto

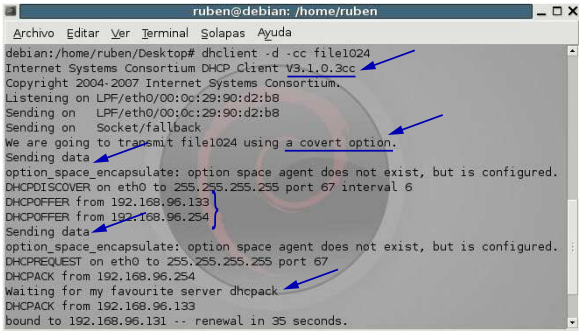


Fig. 5: Cliente transmitiendo 1024 bytes

en paquetes de descubrimiento (discover) como de solicitud (request) (véase Figura 6).

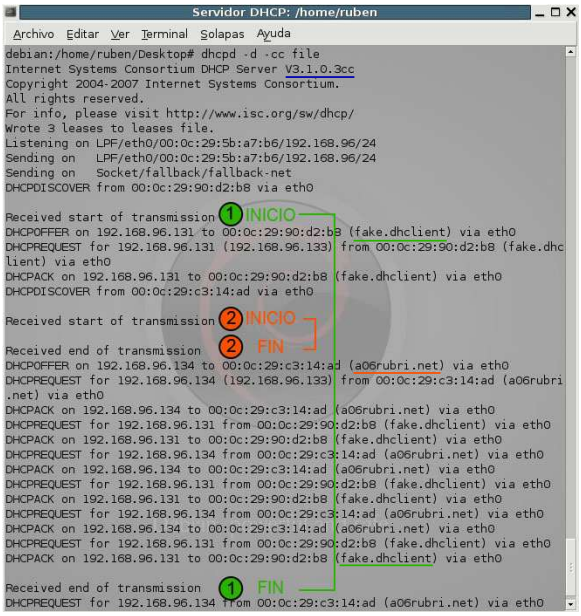


Fig. 6: Servidor recibiendo de dos clientes.

4.4.1 Ventajas e Inconvenientes. Posibles Mejoras

Con esta implementación, además de aumentar el ancho de banda, que era el principal problema que presentaba la versión con *xid*, existe la posibilidad de crear si se desease un canal de comunicación de doble sentido, en el que cliente y servidor DHCP pudieran enviar información a la otra entidad. Ésto se debe a que el campo de opciones, en concreto la opción de la que se hace uso para la creación del canal, no es utilizada para ningún otro motivo.

En cuanto a la detectabilidad de la aplicación se podrían realizar diversas mejoras. En primer lugar se ha optado por introducir opciones con hasta 255 bytes, pues es lo máximo permitido, sin provocar la aparición de varias opciones con el mismo "tag" o identificador, lo cual podría resultar sospechoso. Aún así, el uso de opciones de 255 bytes de longitud también podría llamar la atención puesto que el tamaño de los paquetes se ve aumentado considerablemente. Así pues, cuando los ficheros a transmitir son de un tamaño superior a los 255 bytes lo ideal sería poder optar por: un ancho de banda máximo o por el mayor sigilo. En el primer caso se procedería de la forma actual, enviando el máximo de 255 bytes permitido en cada opción,

pero sin exceder este valor puesto que creemos que el envío de un tamaño superior podría ser demasiado sospechoso, incluso para un red sin una vigilancia exhaustiva. En el segundo caso se procedería enviando datos en porciones más pequeñas ya que por lo general los paquetes DHCP suelen tener un tamaño de 300 bytes, de los cuales sólo 60 suelen dedicarse al campo de opciones.

5 Conclusiones y Trabajo Futuro

Existe una gran cantidad de bibliografía relativa a los canales encubiertos pero no existe un consenso claro en lo que a la clasificación de esta forma de comunicación oculta se refiere. En este artículo se propone una clasificación que trata de abarcar la totalidad de canales encubiertos existentes.

Además, existen multitud de estudios que demuestran que es posible descubrir el uso de éstas aplicaciones debido a patrones de uso, características de los datos enviados, etcétera. Los canales de temporización suelen ser más difíciles de detectar aunque la presencia de secuencias de acciones repetitivas pueden levantar sospechas. No existe el mejor canal de comunicación encubierto bajo cualquier circunstancia, y normalmente aquellos que se comportan adecuadamente en ciertos escenarios dejan de ser útiles en otros. Por lo general, el ancho de banda se encuentra reñido con la capacidad de pasar inadvertido. Además, otro factor determinante en la elección de un canal es la cantidad de bytes erróneos que el destino es capaz de soportar para poder recuperar la información transmitida de manera que ésta siga siendo inteligible. Así pues, se deberán sopesar estos factores a la hora de decantarse por una u otra implementación.

Normalmente los canales encubiertos han sido utilizados para burlar las políticas de seguridad, sin embargo, también podrían ser utilizados de manera beneficiosa aprovechando un ancho de banda que en la actualidad se encuentra en desuso. Esta es una línea de trabajo futuro interesante.

Aún existe un largo camino por recorrer en cuanto a la detección de estas formas de comunicación oculta. La detección mediante reglas es una posibilidad, sin embargo, no es una solución definitiva debido a las múltiples formas de ocultación existentes en la actualidad, y que sin duda surgirán más adelante. Asimismo, es necesario la depuración de las reglas para reducir al mínimo el número de falsos positivos, ya que la situación ideal sería que el propio IDS estuviese lo suficientemente preparado como para detectar la amenaza y actuar en consecuencia.

En la sección 4.2 se han señalado diferentes posibilidades de ocultación que también podrían ser llevadas a la práctica. Asimismo, se han analizado las ventajas e inconvenientes y posibles mejoras, algunas de las cuales ya han sido implementadas. Nuestros pasos actuales se encuentran en esta dirección.

De entre todas las propiedades de un canal encubierto en red, tres destacan como las más importantes: detectabilidad, ancho de banda y fiabilidad. Las dos primeras, han sido tratadas en este artículo para el canal encubierto implementado sobre DHCP. Las pruebas realizadas en una LAN pequeña, arrojan una fiabilidad del 100%. Sin embargo, sería conveniente realizar estudios sobre el porcentaje de errores que el canal encubierto creado proporcionaría en situaciones extremas del servicio DHCP. Esto podría suponer la repetición de paquetes, y por lo tanto el canal encubierto debe poder desechar los paquetes repetidos. Es el siguiente paso a realizar en nuestra implementación.

Referencias

[1] B. W. Lampson, "A Note on the Confinement Problem," *Commun. ACM*, vol. 16, no. 10, pp. 613–615, 1973.

[2] V. D. Gligor, "A Guide to Understanding Covert Channel Analysis of Trusted Systems, "The Light Pink Book"," U.S. National Computer Security Center, Tech. Rep., 1993, [online]. [Accedida el 27 de Febrero 2007]. Disponible a través de la World Wide Web: ⟨http://www.windowsecurity.com/whitepapers/NCSCTG030_Light_Pink_book_.html⟩.

[3] C. G. Girling, "Covert Channels in LAN's." *IEEE Trans. Software Eng.*, vol. 13, no. 2, pp. 292–296, 1987.

[4] P. Singh, "Whispers On The Wire - Network Based Covert Channels Exploitation & Detection," *Symposium on Security for Asia Network (SyScAN)*, 2005, [online]. [Accedida el 12 de Enero 2007]. Disponible a través de la World Wide Web: ⟨http://gray-world.net/papers/pukhrajsingh_covert.doc⟩.

[5] Z. Kotulski and W. Mazurczyk, "Covert Channel for Improving VoIP Security." in *Proceedings of Multiconference on Advanced Computer Systems (ACS)*. Berlin Heidelberg: Springer-Verlag, 2006, pp. 311–320, [online]. [Accedida el 27 de Agosto 2007]. Disponible a través de la World Wide Web: ⟨http://www.ippt.gov.pl/~zkotulsk/CovertChannelforImprovingVoIPSecurity.pdf⟩.

[6] C. H. Rowland, "Covert Channels in the TCP/IP protocol suite," 1996, [online]. [Accedida el 27 de Febrero 2007]. Disponible a través de la World Wide Web: ⟨http://www.firstmonday.org/issues/issue2_5/rowland/⟩.

[7] A. Dyatlov, "Firepass - Gray-World.net Team," 2003, [online]. [Accedida el 12 de Mayo 2007]. Disponible a través de la World Wide Web: ⟨http://gray-world.net/it/pr_firepass.shtml⟩.

[8] D. Kaminsky, "Tunneling Audio, Video, SSH and pretty much anything else over DNS," 2004, [online]. [Accedida el 14 de Mayo 2007]. Disponible a través de la World Wide Web: ⟨http://www.doxpara.com/⟩.

[9] S. B. Lipner, "A Comment on the Confinement Problem," in *SOSP '75: Proceedings of the fifth ACM symposium on Operating systems principles*. New York, NY, USA: ACM Press, 1975, pp. 192–196.

[10] M. Schaefer, B. Gold, R. Linde, and J. Scheid, "Program Confinement in KVM/370," in *ACM '77: Proceedings of the 1977 annual conference*. New York, NY, USA: ACM Press, 1977, pp. 404–410.

[11] NCSC, "Department of Defense Trusted Computer System Evaluation Criteria, "The Orange Book"," U.S. Department of Defense, Tech. Rep., 1985.

[12] M. H. Kang and I. S. Moskowitz, "A Pump for Rapid, Reliable, Secure Communication," in *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 1993, pp. 119–129.

[13] J. McHugh, "Covert Channels Analysis: A Chapter of the Handbook for the Computer Security Certification of Trusted Systems," Naval Research Laboratory, Washington, D.C., Tech. Rep., 1995.

[14] M. Wolf, "Covert Channels in LAN Protocols," in *LANSEC '89: Proceedings on the Workshop for European Institute for System Security on Local Area Network Security.* London, UK: Springer-Verlag, 1989, pp. 91–101.

[15] T. G. Handel and M. T. Sandford, "Hiding Data in the OSI Network Model," in *Proceedings of the First International Workshop on Information Hiding.* London, UK: Springer-Verlag, 1996, pp. 23–38.

[16] Daemon9, "Loki2 (the implementation)," 1997, [online]. [Accedida el 27 de Febrero 2007]. Disponible a través de la World Wide Web: ⟨http://www.phrack.org/archives/51/P51-06⟩.

[17] J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, "Covert Messaging Through TCP Timestamps," in *Proceedings of the Privacy Enhancing Technologies Workshop (PET)*, 2002, pp. 194–208.

[18] K. Ahsan, "Covert Channel Analysis and Data Hiding in TCP/IP," Ph.D. dissertation, University of Toronto. Department of Electrical and Computer Engineering, 2002, [online]. [Accedida el 13 de Febrero 2007]. Disponible a través de la World Wide Web: ⟨http://gray-world.net/papers/ahsan02.pdf⟩.

[19] S. Cabuk, C. E. Brodley, and C. Shields, "IP Covert Timing Channels: Design and Detection," in *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security.* New York, NY, USA: ACM Press, 2004, pp. 178–187.

[20] N. B. Lucena, G. Lewandowski, and S. J. Chapin, "Covert Channels in IPv6." in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, vol. 3856. Berlin Heidelberg: Springer-Verlag, 2006, pp. 147–166.

[21] G. Shah, A. Molina, and M. Blaze, "Keyboards and Covert Channels," in *USENIX-SS'06: Proceedings of the 15th Conference on USENIX Security Symposium.* Berkeley, CA, USA: USENIX Association, 2006, pp. 59–75.

[22] D. Stødle, "Ping Tunnel - Send TCP traffic over ICMP," 2005, [online]. [Accedida el 22 de Marzo 2007]. Disponible a través de la World Wide Web: ⟨http://www.cs.uit.no/~daniels/PingTunnel/⟩.

[23] N. Ogurtsov, H. Orman, R. Schroeppel, S. O'Malley, and O. Spatscheck, "Covert Channel Elimination protocols," Department of Computer Science, University of Arizona, Tucson, AZ, USA, Tech. Rep., 1996, [online]. [Accedida el 4 de Agosto 2007]. Disponible a través de la World Wide Web: ⟨ftp://ftp.cs.arizona.edu/reports/1996/TR96-14.pdf⟩.

[24] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating Steganography in Internet Traffic with Active Wardens," in *IH '02: Revised Papers from the 5th International Workshop on Information Hiding.* London, UK: Springer-Verlag, 2002, pp. 18–35.

[25] S. J. Murdoch and S. Lewis, "Embedding Covert Channels into TCP/IP," in *Proceedings of the 7th Information Hiding Workshop.* Barcelona, España: Springer-Verlag, 2005, pp. 247–261.

[26] J. C. Wray, "An Analysis of Covert Timing Channels," in *IEEE Computer Society Symposium.* Los Alamitos, CA, USA: IEEE Computer Society, 1991, pp. 2–7.

[27] I. S. Moskowitz and M. H. Kang, "Covert Channels – Here to Stay?" in *Compass'94: 9th Annual Conference on Computer Assurance.* Gaithersburg, MD: National Institute of Standards and Technology, 1994, pp. 235–243.

[28] I. S. Moskowitz and A. R. Miller, "Simple Timing Channels," in *SP '94: Proceedings of the 1994 IEEE Symposium on Security and Privacy.* Oakland, CA: IEEE Computer Society, 1994, pp. 56–64.

[29] J. W. Gray III, "Countermeasures and Tradeoffs for a Class of Covert Timing Channels," Hong Kong University of Science and Technology, Tech. Rep., 1994, [online]. [Accedida el 25 de Marzo 2007]. Disponible a través de la World Wide Web: ⟨http://citeseer.ist.psu.edu/361793.html⟩.

[30] B. R. Venkatraman and R. E. Newman-Wolfe, "Capacity Estimation and Auditability of Network Covert Channels," in *SP '95: Proceedings of the 1995 IEEE Symposium on Security and Privacy.* Washington, DC, USA: IEEE Computer Society, 1995, p. 186.

[31] C. Meadows and I. S. Moskowitz, "Covert Channels – A Context-Based View," in *Proceedings of the First International Workshop on Information Hiding.* London, UK: Springer-Verlag, 1996, pp. 73–93.

[32] Z. Wang and R. B. Lee, "New Constructive Approach to Covert Channel Modeling and Channel Capacity Estimation," in *8th Information Security Conference (ISC '05).* Berlin Heidelberg: Springer-Verlag, 2005, pp. 498–505.

[33] ISC, "Internet Systems Consortium, Inc." 2007, [online]. [Accedida el 21 de Noviembre 2007]. Disponible a través de la World Wide Web: ⟨http://www.isc.org⟩.