Instituto Politécnico Nacional

Escuela Superior de
Ingeniería Mecánica y Eléctrica
Unidad Culhuacán

Sección de Estudios de Posgrado e
Investigación

Modelo de Evaluación de Canales Ocultos para Establecer una Comunicación Segura

Tesina
Que para obtener el grado de

Especialidad en Seguridad Informática
y Tecnologías de la Información

Presenta

Lic. Gunnar Eyal Wolf Iszaevich

Asesor:
MSI. Pablo Ramón Mercado Hernández

México, Distrito Federal, marzo de 2015

SIP-14-E

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESINA

En la Ciudad de México, D. F. siendo las 18:00 horas del día 20 del mes de marzo del 2015 se reunieron los miembros de la Comisión Revisora de la Tesina, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de SEPI-ESIME-CULHUACAN para examinar la tesina titulada:

"Modelo de Evaluación de Canales Ocultos para Establecer una Comunicación Segura"

Presentada por el alumno:

Wolf	**Iszaevich**	**Gunnar Eyal**
Apellido paterno	Apellido materno	Nombre(s)

Con registro: **A 1 4 0 3 3 4**

aspirante de:

ESPECIALIDAD EN SEGURIDAD INFORMÁTICA Y TECNOLOGÍAS DE LA INFORMACIÓN

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESINA***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director(a) de tesina

M.S.I Pablo Ramón Mercado Hernández

Dra. Mariko Nakano Miyatake

Dr. Rubén Vázquez Medina

INSTITUTO POLITÉCNICO NACIONAL S.E.P. SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION ESIME CULHUACAN

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Gonzalo Isaac Duchen Sánchez

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 20 del mes de marzo del año 2015, el que suscribe **Gunnar Eyal Wolf Iszaevich**, alumno del Programa de **Especialidad en Seguridad Informática y Tecnologías de la Información**, con número de registro **A014334**, adscrito a la **Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacán**, manifiesta que es el autor intelectual del presente trabajo de Tesina bajo la dirección del **MSI. Pablo Ramón Mercado Hernández,** y cede los derechos del trabajo titulado **Modelos de evaluación de canales ocultos para establecer una comunicación segura**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: **gwolf@gwolf.org**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Lic. Gunnar Eyal Wolf Iszaevich

Resumen

El presente trabajo aborda la creación de un modelo que describa a los *canales ocultos*, permitiendo compararlos en sus distintos aspectos. Por muchos años, el principal enfoque en la literatura académica al abordarlos ha sido *adversarial*, esto es, tomarlos como una amenaza, como un abuso a las políticas de uso de un sistema o red de cómputo. Hay, sin embargo, una gran cantidad de escenarios para los cuales resultan no únicamente legítimos, sino que indispensables.

Dado el poco acercamiento formal que hay a los canales ocultos, se argumenta respecto al por qué debe profundizarse en su estudio formal. Una vez establecido esto, se presenta un modelo descriptivo que permite discernir los componentes principales de un canal oculto y, presentando al canal en cuestión desde diferentes ángulos, permite evaluar su aplicabilidad, fortalezas y debilidades para su aplicación en escenarios particulares.

Por último, a modo de ejemplo, se presenta la aplicación del modelo desarrollado a las 14 propuestas de canal oculto consultadas como parte de la revisión bibliográfica, así como a una propuesta adicional enfocada a los escenarios planteados.

Abstract

This work presents the creation of a model describing *covert channels*, allowing the reader to compare their different aspects. For many years, academic literature has mainly focused on them in an *adversarial* way, this means, they are taken as a threat, as an abuse on a computer system or network usage policy. There are, however, a large amount of scenarios for which they are not only legitimate, but necessary.

Given the little amount of formal analysis on covert channels, this text argues as to why there is need to further its formal study. Once this is established, a descriptive model is presented. This model allows discerning the main components of a covert channel and, by dissecting it from different angles, allows evaluating its applicability, strengths and weaknesses in particular scenarios.

Finally, as working example, this model is applied to the 14 different covert channel proposals that were studied as part of the bibliographic review, as well as to one further proposal focused on the stated scenarios.

Dedicatoria

A Alan y a Elena, quienes seguramente están preparando un canal oculto de comunicación que me tomará años (¡y eso siendo optimista!) detectar. ¿Comprender o descifrar? No, no lo creo posible.

Apenas comienza el disfrute de nuestra vida juntos.

A Regina, mi amada cómplice y compañera de camino. Por muchos años más de seguir encontrando significados, disfrutando del uso y del abuso del lenguaje. Y claro está, por enfrentar juntos la aventura de la vida, presentando con el mundo a los dos aquí ya citados.

A mis padres, con quienes aprendí a buscar siempre espacios para la redundancia en el lenguaje y transmutarlos en carcajadas. Y gracias a quienes fui eligiendo el camino de vida por el que he andado. Imperfecto cual todo camino, pero muy disfrutado.

Agradecimientos

A mi asesor, Mtro. Pablo Mercado. Los comentarios que hizo al desarrollo de este proyecto y las atinadas observaciones a mi andar *por las ramas* me han enseñado mucho más que lo directamente aplicable a esta obra. Si me llevo un aprendizaje a lo largo de mi paso por la Especialidad, más allá de los conocimientos técnicos específicos a cada materia y proyecto, es el necesario rigor del ordenamiento de las ideas, la correcta aplicación del método científico, a pensar, fundamentar y presentar mis ideas de una forma ordenada y validada.

A mis profesores y compañeros de estudios al paso por la Especialidad. Por muchos años rehuí del aprendizaje formal, y me han brindado una gratísima experiencia en este programa, si bien corto, indudablemente intenso.

Al Instituto Politécnico Nacional en particular, y en general al sistema público de educación superior en nuestro país. Nací cercano a nuestras universidades públicas, y toda la vida he estado relacionado con una u otra de sus facetas. No imagino la vida (o la sociedad en la que vivimos) sin su contribución.

Índice general

Capítulo 1

Introducción

1.1. Definición del problema

La realidad de la seguridad en redes de datos a lo largo de los años ha ido cambiando, creciendo en sus alcances y en sus implicaciones ante la sociedad. Algunos temas, como el que en el presente trabajo se abordará, comienzan siendo anatema de la cultura formal de la seguridad informática, pero al paso del tiempo, va haciéndose claro que deben ser abordados si no queremos pecar de míopes.

La literatura académica formal que aborda a los canales ocultos los presenta, en líneas generales, como amenazas. Muchos de los trabajos consultados reconocen la inevitabilidad de su existencia, pero centran su análisis en facilitar su detección, restricción o posible eliminación — Pero siempre enfocándose en propiedades particulares, en implementaciones específicas.

La realidad de las redes de datos hoy en día, sin embargo, apunta en una dirección muy distinta. Los canales ocultos son muchas veces *necesarios*, como necesaria es también una profundización en su estudio para que puedan utilizarse mejor *a favor* de la seguridad de la información. Pueden brindar un espacio fundamental para facilitar tareas de legítima administración de redes, como se ejemplificará al describir algunos escenarios en la sección 1.5.1.

Los canales ocultos presentan una complejidad muy particular: Al ser necesariamente *huéspedes inesperados* de comunicaciones de otro tipo, dependen y derivan fuertemente del ingenio y de la inventiva de sus desarrolladores. Parte importante de la fortaleza de un canal oculto radica en qué tan inesperado es su planteamiento. Los canales ocultos no pueden, pues, ser correctamente clasificados siguiendo la lógica con la cual se estudian otros canales de comunicación.

Esta tesina, abordada desde un enfoque exploratorio (Hernández Sampieri, Fernández Collado y Baptista Lucio 2006), presenta un *modelo descriptivo* que permitirá estudiar y comparar mejor a distintas propuestas de canal oculto. Esto facilitará a un administrador de sistemas o redes elegir cuál mecanismo se ajusta mejor a sus necesidades particulares, o a un desarrollador encontrar si hay áreas

particulares para las cuales su propuesta podría recibir mejoras — O puntos de tensión que pueden llevar a que, si mejora en un sentido, necesariamente empeore en otro.

Como resultado del modelo presentado, y a modo de síntesis de conceptos, se presentará también una propuesta de canal oculto enfocado a resolver los escenarios ya referidos.

1.2. Objetivo general

El presente trabajo presenta el desarrollo, a partir de la comparación de distintas implementaciones existentes y validado ante un grupo de profesionales, de un modelo descriptivo que permita caracterizar los distintos canales ocultos de comunicación, y evaluarlos sobre sus distintos ejes.

El modelo generado permitirá tanto que administradores de infraestructura puedan comparar distintos canales para sus necesidades particulares, como auxiliar para que desarrolladores e investigadores puedan enfocarse en los puntos requeridos para diseñar una nueva propuesta.

1.3. Objetivos particulares

Para desarrollar el presente trabajo, se cubrieron los siguientes objetivos particulares:

1. Caracterizar escenarios de comunicación legítima empleando canales ocultos.
2. Realizar una revisión bibliográfica de las diferentes implementaciones de canales ocultos existentes, tanto en la literatura formal científica/académica como entre las comunidades de práctica.
    a) Explorar y contrastar los distintos espacios de ocultamiento empleados.
    b) Determinar la validez de un término único para las distintas implementaciones.
3. Examinar los fundamentos teóricos sobre los cuales se construyen las implementaciones abordadas.
    a) Explorar los planteamientos formales respecto al uso de canales de comunicación, espacios de ocultamiento de información, estimación de capacidad de canales y autenticación.
4. Analizar y validar la necesidad de este trabajo entre un grupo amplio de especialistas por medio de la aplicación y posterior análisis de una encuesta.

5. Integración de un modelo que permita la descripción y comparación de los esquemas resultantes del inciso 2.

   a) Aplicar el modelo a todos los esquemas presentados, retroalimentándolo de forma iterativa.

6. Desarrollo de una propuesta de canal seguro que resuelva a los supuestos planteados en el inciso 1, considerando los aspectos obtenidos del inciso 4.

   a) Aplicar el modelo integrado en el inciso 5 a la propuesta desarrollada.

7. Sintetizar la aplicación del modelo a todos los esquemas presentados (incluyendo al propuesto) en un cuadro comparativo global.

1.4. Hipótesis

El desarrollo de un modelo que documente los componentes y principales interacciones para la implementación de mecanismos de comunicación sobre canal oculto contribuirá a su mejor comprensión y uso, y contribuirá a que los canales ocultos dejen de verse desde un punto de vista meramente *adversarial* para convertirse en un objeto del estudio formal.

1.5. Justificación

Los esquemas para administración remota de servidores sobre canales cifrados (como `ssh`, *Secure Shell*) resultan hoy en día fundamentales, pero insuficientes, para la gestión de servicios y para una correcta respuesta a incidentes.

El modelo descrito por el presente trabajo busca resolver una serie de necesidades que no ha sido cubierta por las herramientas más ampliamente difundidas. Este modelo se enfoca a los *canales ocultos*: Mecanismos de comunicación que típicamente han sido aprovechados por los intrusos; la mayor parte de la bibliografía que se abordará en la sección 3.2 se enfoca a la *prevención* del uso de canales ocultos, siendo que cada vez más se vuelven necesarios para una gestión completa y proactiva de la seguridad.

Dado que este tema –o, cuando menos, este enfoque– no ha sido abordado con suficiente profundidad, resulta necesario el desarrollo de un modelo delineando los principales componentes con que un canal de esta naturaleza debe cumplir. Mucho se ha escrito acerca de la detección de su existencia desde un punto de vista *adversarial*, pero el desarrollo de implementaciones para su aprovechamiento para propósitos legítimos sufre al carecer de un modelo que presente los puntos que deben considerarse.

En la siguiente sección se plantean algunos escenarios legítimos[1] para los

[1] Claro está, el modelo adversarial y el uso común están llenos de escenarios ilegítimos: Desde los modelos teóricos de comunicación en que *Alice*, *Bob*, *Eve* y *Walter*, omnipresentes actores de la literatura en criptografía, buscan engañarse, espiarse y comunicarse entre sí, hasta los diferentes mecanismos de monitoreo y señalización de *bot-nets* que tanto preocupan

cuales resulta importante contar con un canal para la administración remota sobre un canal oculto.

Y si bien es cierto que existe una amplia cantidad de propuestas de canal oculto que podrían ser adoptadas por los grupos de usuarios que caen en los escenarios supuestos (así como en todos los otros que lleven a la búsqueda de canales ocultos), elegir cuál es la que mejor responde a determinada necesidad, o encontrar qué aspectos hace falta fortalecer de una nueva propuesta en desarrollo, resulta prácticamente imposible hacerlo sin contar con un modelo que aborde los distintos aspectos de cada una.

En cada uno de los siguientes escenarios se plantea no únicamente una situación para la cual convendría el empleo de canales ocultos, sino la razón de por qué no resulta suficiente el uso de los canales criptográficos comunmente utilizados hoy en día.

1.5.1. Escenarios

Hay una gran cantidad de razones que pueden llevar a un administrador de sistemas a considerar el establecimiento de un canal oculto. Al no contar con un modelo desarrollado, quienes busquen cubrir esta necesidad se enfrentan a ir desarrollando sobre la marcha, probablemente obviando pasos importantes, que deberían ser cubiertos para que su solución elegida tuviera la fuerza requerida.

Se presentan a continuación tres escenarios desde los cuales puede justificarse, a modo de ejemplo y desde el enfoque del administrador *legítimo* de un sistema de cómputo, la creación de un canal oculto; resultará obvio que, si se consideran los usos no legítimos que normalmente se da a estos canales, encontraremos una mucho mayor cantidad de escenarios.

Administración desde redes públicas o poco confiables Un administrador de sistemas debe poder responder a los incidentes graves en todo momento y donde sea que esté. La creciente ubicuidad del acceso a Internet se ha vuelto uno de sus mayores aliados; gracias a ella, cada vez es más fácil poder responder de inmediato a un reporte de fallo por parte de los usuarios. Sin embargo, ante una falta de cuidado, esto puede tornarse en contra del administrador.

Si bien la mayor parte de los ataques registrados en Internet no son *dirigidos* (esto es, son lanzados en simultáneo y paralelo a grandes cantidades de equipos buscando vulnerabilidades, en pos de *recursos genéricos* a ser empleados, por ejemplo, para construir una red para el envío de *spam* o el envío de ataques de negación de servicio distribuídos (Negroni 2005); al atacante no le importa la información o el servicio en particular de la organización víctima), la peligrosidad de los que sí lo son dirigidos es mucho mayor: El daño sufrido por una organización ante el robo de su información confidencial o la modificación o destrucción deliberada, sea de

y ocupan a los administradores de sistemas en todo el mundo. Este modelo debe poder aplicarse también a todos ellos.

la información que expone al público o de la configuración de sus equipos, resulta sensiblemente mayor que el mero mal uso de sus recursos.

Ahora, ¿en qué consiste un escenario de ataque que implique a un administrador diligentemente conectándose a su equipo? En que dicha respuesta perfectamente podría revelar información valiosa. Por poner un ejemplo, si el atacante crea meramente una distracción que requiera de la intervención del administrador, como podría ser una simple negación de servicio (CERT, Software Engineering Institute 1997).

Teniendo control un potencial atacante de alguna red por donde pasen los paquetes del administrador, puede causar una negación de servicio e iniciar una *captura de datos* (Tanase 2002) buscando registrar un flujo de paquetes con las credenciales de autenticación del administrador, para con ellas tener acceso completo a los datos que busca obtener.

Y si bien prácticamente la totalidad de los administradores conoce la importancia de emplear mecanismos de administración sobre canal cifrado (precisamente por esta amenaza), la conjunción de esta con una vulnerabilidad de implementación en mecanismos criptográficas como ya ha habido tantas (Arnbak y col. 2014; Codenomicon Defensics 2014; Marlinspike 2009; Violet 2014; Weimer 2008), esto puede ser suficiente para darle al atacante toda la información necesaria para entrar con privilegios avanzados.

Por otro lado, un administrador de sistemas no sólo debe preocuparse por cuidar sus credenciales de ser reveladas: Cuando detecta un patrón irregular en el servicio, el primer reto a que se enfrenta es a encontrar qué intenta hacer el atacante *sin alertarlo* — ¿Quién es? ¿Qué busca? ¿Qué tan experto es? ¿Ha colocado alguna *bomba lógica* para cubrir sus pasos? Asumiendo que el atacante puede haber logrado pleno control de la red, el administrador debe poder llevar a cabo las acciones básicas sin que el atacante *se sienta observado*.

El administrador inter-jurisdiccional La universalización del acceso a Internet ha llevado a que organizaciones de todo tipo requiera de la "red de redes" para cualquier tarea imaginable. Al inicio del 2013 había más de 1,200,000,000 computadoras conectadas a la red (ISC 2014), y la tendencia al aumento se mantiene lineal por lo menos desde el 2003. El incremento del uso de la red desde dispositivos móviles, así como la promesa de la *Internet de las Cosas*, indican que la pendiente muy probablemente se inclinará más aún en el futuro cercano-medio.

Los Estados Unidos son país con mayor cantidad de equipos en línea, con 505,000,000 (CIA 2014) (esto es, aproximadamente 1.6 por cada uno de sus 300 millones de habitantes, y casi la mitad de los equipos conectados a nivel mundial). Esta tremenda cantidad de computadoras conectadas en un sólo país se explica no sólo por la supremacía tecnológica, sino porque al haberse desarrollado con tal antelación Internet en este país con respecto al resto del mundo, casi todos los enlaces internacionales (a excepción de los

que interconectan a la Unión Europea) van hacia este país (Telegeography 2012).

Muchos administradores de infraestructura de cómputo han decidido *hospedar* a sus servidores en empresas proveedoras de servicio, principalmente en los Estados Unidos, pero en líneas generales, alrededor de todo el mundo. Alojar los servidores en un *centro de datos* reduce fuertemente los costos de administración de red, y racionaliza el uso de recursos.

Esto, sin embargo, no queda libre de problemas: El alojar los servidores en un centro de datos compartido implica que, compartiendo la red física con ellos, habrá probablemente cientos o miles de equipos pertenecientes a terceros, no necesariamente bienintencionados, que pueden aprovechar esta situación para buscar con mayor facilidad acceso no legítimo a los activos de dicha organización.

Además de esto, el cruce de jurisdicciones legales lleva a muchos a tener un temor justificado de las *escuchas* por parte de entidades gubernamentales, que tienen tanto suficientes puntos de control sobre las redes de datos como disponibilidad de recursos de cómputo para capturar inteligencia a gran escala; hemos visto en los últimos años cómo este supuesto salió por completo del ámbito de las *teorías de la conspiración* para convertirse en la realidad aceptada gracias a las filtraciones de información de inteligencia, particularmente las realizadas por el grupo *Wikileaks* y por Edward Snowden (Shane y Lehren 2010; Greenwald, MacAskill y Poitras 2013).

Gestión de un *honeypot* Una de las tareas que puede realizar como parte de su trabajo de análisis y reacción un equipo de investigación y respuesta de seguridad operativa es la operación de los *equipos señuelo*, también llamados *honeypot*: Equipos conectados a red, aparentemente ejecutándose con software vulnerable, dedicados a captar ataques para poder analizarlos posteriormente. Un *honeypot* puede brindar muy valiosa información para la prevención, detección y reacción ante ataques informáticos, sin embargo, también introduce riesgos inocultables (Spitzner y Roesch 2001).

La tecnología de la virtualización se ha popularizado enormemente en la última década, y ofrece importantes ventajas al operador de un *honeypot*; facilita además la creación económica de una red virtual de equipos vulnerables que pueden ser monitoreados con mucho mayor control incluso que las computadoras reales (Clark 2001; Provos 2003). Sin embargo, se han documentado ya diversos *exploits* que permiten, una vez habiendo obtenido privilegios elevados en una máquina virtual, un atacante puede *escapar* del hipervisor y hacerse del control del sistema físico (Schwartz 2012; Allar 2012; Kortchinsky 2009). Si el atacante obtiene este tipo de acceso antes de haber sido detectado por el administrador, podría instalar una *escucha* en el sistema anfitrión; el atacante podría monitorear toda la comunicación de dicho sistema para reaccionar rápidamente en caso de ser detectado.

Nuevamente, contar con un canal oculto permitirá al administrador obtener más información de la interacción que esté realizando el atacante sin alertarlo, y reduciendo su exposición a pérdidas de información.

Los escenarios recién descritos son sólo ejemplos de las muchas maneras por medio de las cuales un atacante podría aprovechar su ubicación para hacer un *olfateo* de tráfico (en inglés, *sniffing*), y los escenarios descritos son sólo unos de muchos puntos de partida posibles.

1.6. Organización de la tesina

La tesina está estructurada en siete capítulos, siguiendo la estructura delineada en la figura 1.1. Al final de cada capítulo se mostrará un mapa similar al aquí referido, pero presentando únicamente los conceptos que fueron cubiertos en dicho capítulo, y los conceptos con los que guarda relación directa en otras secciones secciones de la obra.

Los capítulos de la obra son:

1. **Introducción** Presenta el planteamiento base del problema, su justificación, y delínea el plan de acción para su solución.

2. **Estado del arte** Examina las principales implementaciones que se han desarrollado de mecanismos encaminados a resolver problemas relacionados con el planteado en la introducción, para identificar y revelar los detalles en que se diferencian y sus puntos en común, y sirviendo como punto de partida específico para el desarrollo del modelo.

3. **Marco teórico** Revisa los fundamentos teóricos sobre los cuales se ubica el problema, a saber: Los sistemas de comunicaciones, mecanismos de ocultamiento de la información, características de la estimación de capacidad de un canal, y mecanismos de autenticación seguros.

4. **Detección de requisitos** Presenta la encuesta que fue aplicada para validar las observaciones hechas en el transcurso del desarrollo del proyecto, y hace un breve análisis sobre los resultados obtenidos.

5. **Integración del modelo** El núcleo del trabajo desarrollado: Tras hacer precisiones básicas a los conceptos presentados en el capítulo 3 para ceñirlos al problema abordado y delimitar los alcances del modelo que se propone, desarrolla y explica cada uno de los puntos que lo componen.

6. **Aplicación del modelo** Como ejercicio para verificar la aplicabilidad del modelo presentado en el capítulo 5, lo aplica a cada uno de los canales que fueron presentados como parte de la revisión hecha en los capítulos 3 y 2. Los resultados se presentan, primero, en forma de prosa y detallando en cada punto y, posteriormente, como un cuadro comparativo.

7. Conclusiones Presenta las conclusiones de la obra: Las apreciaciones del autor respecto al trabajo realizado, y varias ideas de cómo éste podría extenderse para lograr mayores alcances.

Figura 1.1: Relaciones conceptuales: Mapa general de la tesina

Capítulo 2

Estado del arte

El desarrollo de este capítulo cubrirá algunas de las diferentes mecanismos que se han propuesto e implementado de canales ocultos, tanto en la literatura académica formal como en las comunidades de práctica. Esto es, si bien algunos de los mecanismos abordados (particularmente las más antiguas) describen escenarios teóricos que permiten la comunicación donde debería estar limitada, otras describen implementaciones específicas desarrolladas por sus autores (o incluso observadas *en el campo*).

Este capítulo presenta a cada uno de estos mecanismos de forma separada; el capítulo 6 los presenta, tras haber desarrollado el modelo propuesto en el capítulo 5, comparados sobre los diferentes aspectos que éste cubre.

Cabe mencionar que el término *canal oculto* se emplea a lo largo de la presente obra para referirse a canales que históricamente se han relacionado con distintos nombres: canales subliminales, ocultamiento esteganográfico de información, canales ocultos. La sección 3.2.2 explica por qué se eligió emplear para todo el trabajo el término *canal oculto*.

2.1. Canales ocultos

Los canales ocultos (*covert channels*) no pueden en definitiva ser vistos como una novedad: Ya en 1993, el National Computer Security Center dedicó uno de sus libros de la *Serie del Arcoíris* a su análisis y comprensión (National Computer Security Center 1993). Incluso 20 años antes de dicha publicación ya se caracterizó el uso de los canales ocultos como un ataque contra los activos confidenciales de una organización o un usuario, desde un planteamiento mayormente teórico en (Lampson 1973), y de forma mucho más aplicada a un entorono real y observable, en (Schaefer y col. 1977). Sin embargo, esta literatura temprana los aborda exclusivamente como una vulnerabilidad que debe ser evitada y controlada en la medida de lo posible — y no como una oportunidad para que el administrador del sistema administre sus equipos.

Resultará comprensible que buena parte de la literatura respecto a estos

temas se enfocara a los canales ocultos *en host*; aunque el modelo de red abierta y mundial que implementa Internet existía ya en forma germinal desde los setenta, no fue hasta ya bien entrados los noventa que se mostró como un mecanismo de comunicación universal.

Algunos de los canales descritos, por tanto, basan su planteamiento en la comunicación entre dos procesos sin permiso administrativo para hacerlo dentro de un mismo host. Estos serán considerados para el presente trabajo, ya que, como se menciona en la sección 3.3.1, en muchos casos puede presentarse una equivalencia que permite realizar un mapeo entre ambos espacios.

2.1.1. Port knocking

La presente investigación inició enfocándose a los primeros mecanismos ampliamente conocidos orientados a ocultar el inicio de una comunicación sobre un canal oculto *para propósitos de administración de sistemas*: El mecanismo hoy conocido como *port knocking* (traducción aproximada, *golpe de puerto*), descrito por primera vez en (Krzywinsky 2003). Este mecanismo basa su operación en que los servicios *sensibles* de un servidor permanecerán cerrados por completo, imposibilitando a un atacante el siquiera descubrir que dicho servicio existe, hasta que un usuario autorizado no emita una secuencia de *golpes*, aparentes intentos de conexión a puertos TCP donde residirían servicios inválidos. Una vez que el administrador emite dicha secuencia, el sistema le abre el puerto sensible, con lo que le permite conectarse.

El mecanismo de *port knocking* causó bastante interés y fue ampliamente explorado entre el 2003 y el 2006, incluyendo críticas a estos esquemas como (Izquierdo Manzanares y col. 2005); en líneas generales, y en lo concerniente al trabajo aquí desarrollado, las principales desventajas encontradas en el *port knocking*:

- El planteamiento original brinda protección suficiente únicamente ante un atacante sin capacidad de *olfatear* el tráfico de red, que probablemente está intentando vulnerar al servidor por medio de la *fuerza bruta*. Si el atacante puede obtener un volcado del tráfico de red, podrá observar que hay tráfico por un puerto que no consideraba abierto; haciendo una simple correlación, podrá encontrar los paquetes que fueron enviados previos a la conexión y replicarlos, abriendo de este modo el puerto.

  Existen refinamientos posteriores al concepto de *port knocking* que abordan este problema, generando secuencias únicas y no susceptibles a la repetición, como *fwknop* (Rash 2007; Rash 2007–2014), que incorpora autenticación firmada criptográficamente sobre un sólo paquete TCP enviado a un puerto cerrado, sin embargo la implementación resulta estrechamente ligada al esquema propuesto, y no puede ser directamente aplicado en esquemas de comunicación oculta que empleen transportes diferentes del (aparente) establecimiento de una sesión TCP.

- El mecanismo empleado por *port knocking* es frágil: Dado que su operación se basa directamente en el protocolo IP (sin emplear la capa orientada a

conexión que brinda TCP), no hay garantía de entrega ni ordenamiento de ninguno de los paquetes transmitidos. Es común que las implementaciones dejen un tiempo entre cada uno de los paquetes. Sin embargo, esta alternativa únicamente sirve cuando lo que se pretende implementar es un *port knocking* clásico — Resultaría demasiado lento (y demasiado obvio) para la transmisión de un mensaje completo.

- Además de esto, el *port knocking* resulta medianamente vulnerable a ataques de negación de servicio: Si un atacante conoce la dirección desde donde se está intentando conectar el administrador, puede evitar que éste envíe la secuencia correcta de *golpes* inundando la red de paquetes con dirección de origen falsificada (deGraaf 2007), lo cual *ahogaría* en el ruido a la comunicación legítima.

- No puede obviarse la falta de asociación entre la autenticación y la conexión. Citando a (deGraaf 2007) (traducción propia):

  > En la mayoría de los sistemas basados en *port knocking*, no hay asociación lógica entre la secuencia de autenticación y la conexión que se abre subsecuentemente. Esto significa que después de una autenticación exitosa, cualquiera que tenga la dirección IP del cliente puede conectarse al servidor (a lo que en adelante se referirá como *ataque de carrera*) (...) Este problema resulta especialmente severo en presencia de NAT; para un servidor que obtuvo la dirección pública de un cliente, todos los equipos que comparten la dirección pública del cliente parecen el mismo.

Varios de los puntos mencionados son abordados en (Bo, Jia-zhen y De-Yun 2007), y si bien dicho trabajo propone un mecanismo de ocultamiento de información en los encabezados de paquetes TCP/IP, la sobrecarga derivada de su utilización y el esquema mismo propuesto se alejan de lo aquí perseguido. Las estrategias seguidas en (Kundur y Ahsan 2003) muestran la enorme sobrecarga en que se debe incurrir para establecer comunicación esteganográfica sobre paquetes TCP/IP completamente legales; Murdoch y Lewis (2005) presentan estrategias más eficientes, pero con irregularidades que las hacen mucho más susceptibles de llamar la atención del adversario por su estructura.

Cabe mencionar que al día de hoy el *port knocking* sigue siendo utilizado para enviar señales de forma discreta. Por ejemplo, en octubre de 2014 la empresa de consultoría de seguridad *NCC Group* dio a conocer el resultado del análisis del *root kit* empleado para mantener una vía de acceso a la red de servidores IRC *Freenode*, y el mecanismo inicia con tres *golpes* que se esconden mediante la suma del número de puerto y de secuencia TCP (Cannings 2014).

2.1.2. Web knocking

Una respuesta muy creativa a las debilidades en el *port knocking* es la que presenta de forma informal (Lebelt 2005). Partiendo de que el *port knocking* basa

su funcionamiento, a fin de cuentas, en esconder información en la solicitud al transporte y directamente no transportar datos relevantes al tipo de conexión que se estaría estableciendo, Lebelt implementa un sistema que solicita a un servidor Web una determinada secuencia de páginas, misma que –para evitar ataques de repetición– resulta en la ejecución de la acción preestablecida. Puede apreciarse una lógica similar en (Briganti 2012).

Bello (2008) presenta una implementación que avanza en el sentido de los canales ocultos que estudia el presente trabajo, diseñando un sistema que, dependiendo de los parámetros especificados mediante el *web knocking*, efectuará distintas acciones; la implementación de Bello, sin embargo, está orientada más a la comodidad de administración que a la comunicación sobre canal oculto.

2.2. Ocultamiento esteganográfico

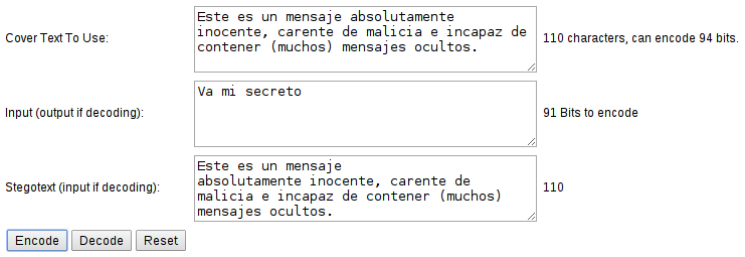
A lo largo de este trabajo ya se han presentado varias referencias a la esteganografía. Resulta claro que para el tema que aquí se aborda es necesario considerar el ocultamiento esteganográfico. Citando a (Johnson 1995) (traducción propia):

> La palabra *esteganografía* significa literalmente *escritura cubierta*, y es derivada del griego. Incluye una amplia gama de métodos de comunicación secreta que esconden la misma existencia del mensaje. (. . .)
>
> La esteganografía es el arte de ocultar la existencia de información dentro de portadores aparentemente inócuos. La esteganografía puede verse como cercana a la criptografía. Ambas han sido empleados a lo largo de la historia con fines de proteger la información. A veces, estas dos tecnologías parecen converger, si bien sus objetivos difieren. Las técnicas criptográficas "revuelven" mensajes de forma que, de ser interceptados, no se puedan entender. La esteganografía, en esencia, "camuflajea" un mensaje para ocultar su existencia y hacer que parezca ser "invisible", ocultando así del todo el hecho de que un mensaje está siendo enviado. Un mensaje cifrado puede levantar sospechas, en tanto que un mensaje invisible no.

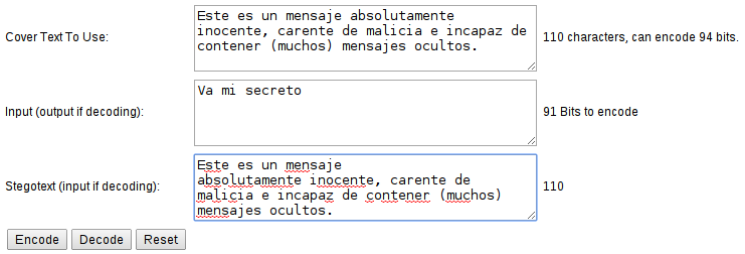
Ahora bien, es necesario hacer algunas precisiones: La esteganografía es un medio de cierto modo *ruidoso* para el envío de información: Por cada símbolo de mensaje oculto que se quiera enviar, típicamente se enviarán por lo menos datos *cubierta* por unas diez veces tanto en el canal visible; el ancho de banda resultante (o el tamaño total de los mensajes que puedan ser transmitidos) resulta por tanto fuertemente limitado.

La mayor parte del desarrollo de la esteganografía se realiza sobre flujos u objetos de naturaleza *binaria* (en contraposición con los de naturaleza *textual*): El área de confort de la esteganografía es la codificación por diferentes mecanismos en imágenes y audios, como lo describe (Codr 2009). La mayor parte de los

ejemplos que se analizan en este trabajo se ocultan en un canal que brinda mucho menos espacio: En los espacios redundantes que ofrecen distintos protocolos de red o en formas de *acomodar* textos.



(a) Ocultando un mensaje por homoglifos.



(b) La revisión ortográfica delata a los caracteres empleados para ocultar el mensaje: Invisible para el humano, pero trivial para un análisis automatizado.

Figura 2.1: Ejemplos de un mensaje oculto en esteganografía por homoglifos, empleando caracteres extendidos Unicode (Crenshaw 2012). La mayor parte de los caracteres del alfabeto latino pueden esconder un bit del mensaje secreto; el mensaje se oculta a la vista del humano, pero resulta obvio para la computadora.

Casi todos los métodos revisados se enfocan a crear o detectar técnicas esteganográficas diseñadas para esconderse *del humano*; un ejemplo digno de mención es (Crenshaw 2012), una implementación desarrollada en el lenguaje *JavaScript* de esteganografía por *homoglifos*, esto es, con caracteres Unicode prácticamente indistinguibles *a simple vista* de los caracteres estándar (ilustrado en las figuras 2.1(a) y 2.1(b)).

Este mecanismo, sin embargo, no puede ser utilizado sobre un protocolo de comunicación, dado que el reemplazo de un caracter por su homoglifo crearía solicitudes ilegales. No sólo no se obtendría una respuesta válida, sino que seguramente activaría las alarmas del potencial atacante. Visto desde las definiciones que se presentan en la sección 3.3, el ocultamiento por homoglifos puede ser *invisible*, pero no es *indetectable*.

Existen muchas otras herramientas de esteganografía sobre texto, pero prácticamente todas se limitan a codificar la información en los espacios entre palabras; esta técnica no sólo resulta trivial de detectar, sino que muy frágil (susceptible a ser eliminada o alterada).

Tres implementaciones mucho más cercanas a lo que el presente trabajo explora son StegoSpam (Harvey 2013), TextHide (*TextHide* 1999) y SpamMimic (McKellar 2000-2014), que codifican mensajes arbitrarios sobre lo que parece ser fragmentos de *spam* (correo comercial no solicitado). Todos ellos fundamentan su elección de canal partiendo de que los *spammers* buscan generalmente alejar a sus mensajes de patrones reconocibles por los filtros automáticos de correo; el correo *spam* contiene grandes cantidades de redundancia y comportamiento aleatorio, mecanismos que permiten la inserción de contenido oculto, e inyectan suficiente reduncancia y variación gramatical como para que a un análisis automatizado le resulte tan dificil como sea posible reconocerlos como indeseables.

En estos casos, sin embargo, la implementación resulta relativamente ingenua y demasiado fácil de detectar ante un análisis sobre el protocolo, y puede ser calificada más de divertimento que de verdadera esteganografía — En el caso de *StegoSpam*, dado que el mensaje se divide en bloques del mismo tamaño y cada bloque se substituye por una cadena de texto, la salida resultante no presenta la menor coherencia ni hilación sintáctica (vamos, a grados mucho menores incluso que el *spam*, lo cual lo es poco decir), y determinados patrones en el texto llevarán a la generación de líneas repetidas; *TextHide* es un producto comercial, por lo que no resultó posible para el presente trabajo hacer un análisis de su operación real, pero basa su funcionamiento en la codificación del significado empleando la redundancia inherente al lenguaje natural — listas de sinónimos y el ordenamiento de los componentes de cada frase.

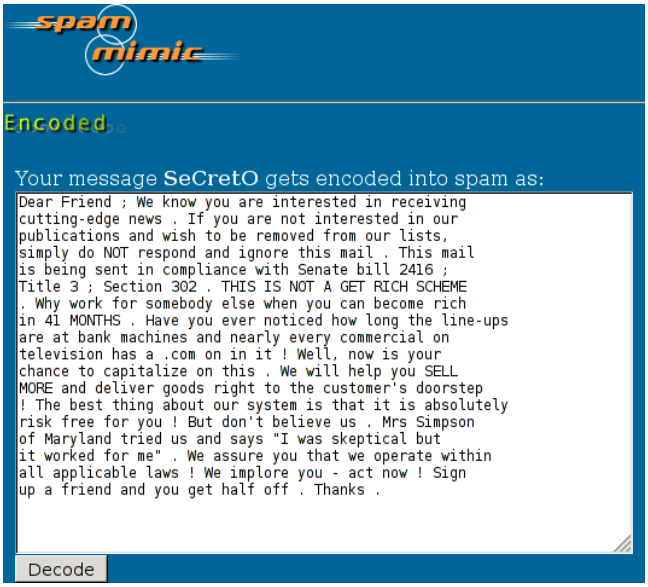


Figura 2.2: Codificación de la cadena «SeCretO» como spam empleando el servicio *SpamMimic*, basado en una gramática regular

SpamMimic, cuya interfaz se muestra en la figura 2.2, es un servicio prestado por de un sitio Web que presenta varios mecanismos para codificar un mensaje, que si bien (al igual que *TextHide*) no brinda información detallada acerca de su implementación, sí presenta una liga a (Wayner 2009) como la fuente de uno de sus modos de funcionamiento. Profundizando sobre los principios que dan origen a SpamMimic resultó claro que, si bien constituye esteganografía sobre un canal de texto, pertenece más bien al tema que aborda la sección 2.3: Las funciones mímica.

Hay dos implementaciones notorias más que caen en este apartado, a pesar de ser fundamentalmente distintas a las descritas anteriormente: (van Hauser 1999) propone un protocolo codificado sobre una pareja solicitud-respuesta HTTP para la conformación de una *bot-net*,[1] Por su parte, (Bauer 2003) presenta un mecanismo para la comunicación anónima y la creación de una *red superpuesta* que se comunique sobre canal oculta sin que haya comunicación directa, siempre que los nodos participantes puedan actuar como servidores Web de contenido *inocente*. El canal que emplean para su comunicación es el espacio que puede controlarse a lo largo de varias interacciones[2] de uno (o varios) terceros: Clientes Web que irán visitando de forma alternada a ambos sitios, probablemente utilizando servicios anonimizadores.

2.2.1. Evaluación del ocultamiento

Dado que este trabajo aborda la creación de un modelo para los canales ocultos, resulta importante comparar con los modelos existentes relacionados.

Olaniyi y col. (2014) presentan la evaluación de mecanismos esteganográficos sobre imágenes como parte de un proceso de votación electrónica, buscando cumplir con el requisito de secrecía en la emisión del voto electrónico desde equipos controlados por el usuario y conectados a redes no confiables; la esteganografía (al igual que el cifrado) en su propuesta se emplea para ocultar de posibles escuchas el hecho de que se está transmitiendo información electoral. La evaluación de la esteganografía que realizan, sin embargo, se reduce a presentar la salida dicotómica de algunos programas ampliamente disponibles para determinar si hay contenido esteganográfico en las imágenes evaluadas.

El trabajo de Kumar (2014) lleva a la categorización de distintos mecanismos de esteganografía en seis niveles consecutivos. Buena parte de los criterios empleados en esta obra, sin embargo, la hacen específica y exclusiva para la esteganografía sobre imágenes, razón por la cual no puede aplicarse al caso que aquí abordamos.

En los artículos de Bailey y Curran (2006) y Li y col. (2011) pueden encontrarse revisiones a profundidad de mecanismos de esteganografía y esteganálisis sobre imágenes; si bien ya se mencionó que un enfoque específico a las imágenes

[1] Un conjunto de computadoras a los cuales un atacante ha tenido acceso y de las cuales se ha *apropiado*, mismas a las que puede controlar para que ejecuten acciones de forma coordinada, como envío masivo de *spam* o ataques de negación de servicio distribuidos.

[2] Principalmente, redirecciones, galletas (*cookies*), encabezado *referido por* (*referer*) y contenido activo (código ejecutado por el navegador).

no resulta de utilidad para abordar el tipo de comunicaciones que atañe a este trabajo, vale la pena citar el siguiente texto de Li (traducción propia):

> A diferencia de los métodos esteganalíticos que requieren conocer los detalles de los métodos esteganográficos objetivo, el esteganálisis universal requiere menos o incluso ninguna información a priori. Un acercamiento estegoanalítico universal típicamente emplea una estrategia basada en el aprendizaje que implica una etapa de entrenamiento y una etapa de prueba. (. . .) [durante las cuales] se emplea un paso de extracción de características. Su función es mapear una imagen de entrada de un espacio de imágenes de alta dimensionalidad a un espacio de características de baja dimensionalidad.

Trasladando esto al ámbito de comunicación en red en el cual se está trabajando, indicaría que para poder evaluar un mecanismo de ocultamiento tendría que hacerse un análisis estadístico del patrón común de tráfico *en la red donde será aplicado*. Esto respalda lo dicho desde la sección 1.1, en el sentido de que el planteamiento de un canal oculto conlleva una importante dosis de *ingenio específico a la situación destino*.

Pevný, Fridrich y Ker (2012) presentan un enfoque interesante orientado a la estimación de la capacidad de un canal esteganográfico. Menciona en su resumen (traducción propia):

> Un estegoanalizador cuantitativo es un estimador del número de cambios embebientes introducidos por una operación de embebimiento específica. Dado que para la mayor parte de los algoritmos el número de los cambios embebientes se correlaciona con la longitud del mensaje, los estegoanalizadores cuantitativos son herramientas forénsicas importantes.

El desarrollo de la obra de Pevný es exhaustivo y técnicamente sólido, sin embargo, también orienta el análisis por completo al ocultamiento de información en imágenes.

En suma, si bien es imposible afirmar que no haya trabajos publicados abordando una evaluación del ocultamiento esteganográfico en los términos que el presente trabajo lo requiere, sobre transportes variados (mayormente textuales o en encabezados redundantes), se hace necesario afirmar que –para el ámbito de desarrollo actual del proyecto– es imposible presentar un modelo que estime la fortaleza o la eficiencia del ocultamiento esteganográfico; esto sin duda es un aspecto relevante a desarrollar, pero excede el ámbito de desarrollo disponible.

Por este motivo, al no encontrar en la literatura otros modelos de evaluación aplicables al problema que este trabajo aborda, la investigación se presenta con un alcance exploratorio-descriptivo (Hernández Sampieri, Fernández Collado y Baptista Lucio 2006).

2.3. Funciones mímica: Gramáticas y lenguajes libres de contexto

Buscando implementaciones esteganográficas eficientes y orientadas al texto, una lectura fundamental resultó ser la propuesta expuesta por Wayner (2009), referido en la sección anterior. Este libro es un texto relativamente de divulgación acerca de métodos esteganográficos. Toma la idea básica de ocultar la comunicación sobre correo *spam*, pero lo hace por medio de herramientas mucho más fuertes y resistentes que sus antecesores: Gramáticas que definen a lenguajes *libres de contexto*. Hasta donde resultó posible investigar, Wayner ha sido el único autor que ha abordado a profundidad esta técnica; la primer implementación ubicada es también suya (Wayner 1991), y presentó un análisis más formal acerca de la *tratabilidad* y la fuerza del esquema que presenta en (Wayner 1999), originalmente bajo el título de *funciones mímicas*. Citando de esta última (traducción propia):

> Las funciones mímicas están diseñadas para esconder información transformándola a otro formato. Pueden ser tan simples como convertir los datos en un archivo con un perfil estadístico particular o tan complejas como dar al nuevo archivo cualquier estructura computable.

El trabajo de Wayner conecta con uno completamente distinto: El paquete *Polygen*, catalogado dentro de la sección de *Juegos* en la distribución Debian GNU/Linux, toma como entrada una gramática regular y construye textos aleatorios que cumplen con ella; el mantenedor del paquete ha publicado un conjunto de ejemplos ilustrando su uso (Zini 2005-2009). La intención de Zini al presentar este paquete en el congreso *DebConf5* fue lúdica y social: Dado que en las listas de discusión del proyecto Debian se presentaban discusiones en las que los participantes comenzaban a repetir sus argumentos, Zini sugirió que si ante una discusión desarrolla una gramática regular que ante *semillas* aleatorias generara mensajes similares a los que aparecían en la lista, la discusión debía darse por zanjada.

En esta misma línea puede ubicarse a (Stribling, Krohn y Aguayo 2005), un generador aleatorio de artículos científicos. Presenta a *SciGen*, un generador de artículos aleatorios de apariencia académica; fue creado por sus autores para demostrar que diversas revistas y congresos de corte académico mentían acerca de sus procesos de dictaminación; los autores lograron la aceptación de varios artículos fabricados aleatoriamente en revistas y congresos.

Hasta donde tenemos noticia, ni Polygen ni SciGen han sido empleados para la creación de un canal oculto para la comunicación — Pero perfectamente podrían emplearse con este fin, como será presentado en la sección 6.2.

2.4. Resumen del capítulo

El objetivo de este capítulo es abordar las *implementaciones específicas*, así como el desarrollo histórico de los canales ocultos, enfocándose particularmente a los diseñados para llevar comunicación sobre una red (en contraposición de los de uso local en un sistema — sin perder de vista la convertibilidad entre ambos, que será abordada en la sección 3.3.1).

El desarrollo del presente trabajo inició con una revisión particularmente a los mecanismos de *port knocking* y avanzó hacia las funciones mímicas/esteganográficas, razón por la cual estos son abordados con particular detalle.

El lugar de este capítulo en el trabajo como un todo es el de presentar y señalar las virtudes y carencias una serie de implementaciones; fue en buena medida esta tarea la que hizo notoria la necesidad de dar al estudio el enfoque que tomó. Además de servir esta revisión como fundamento para el desarrollo del modelo en el capítulo 5, las implementaciones de canales ocultos que aquí se abordaron son evaluadas por el modelo en el capítulo 6.

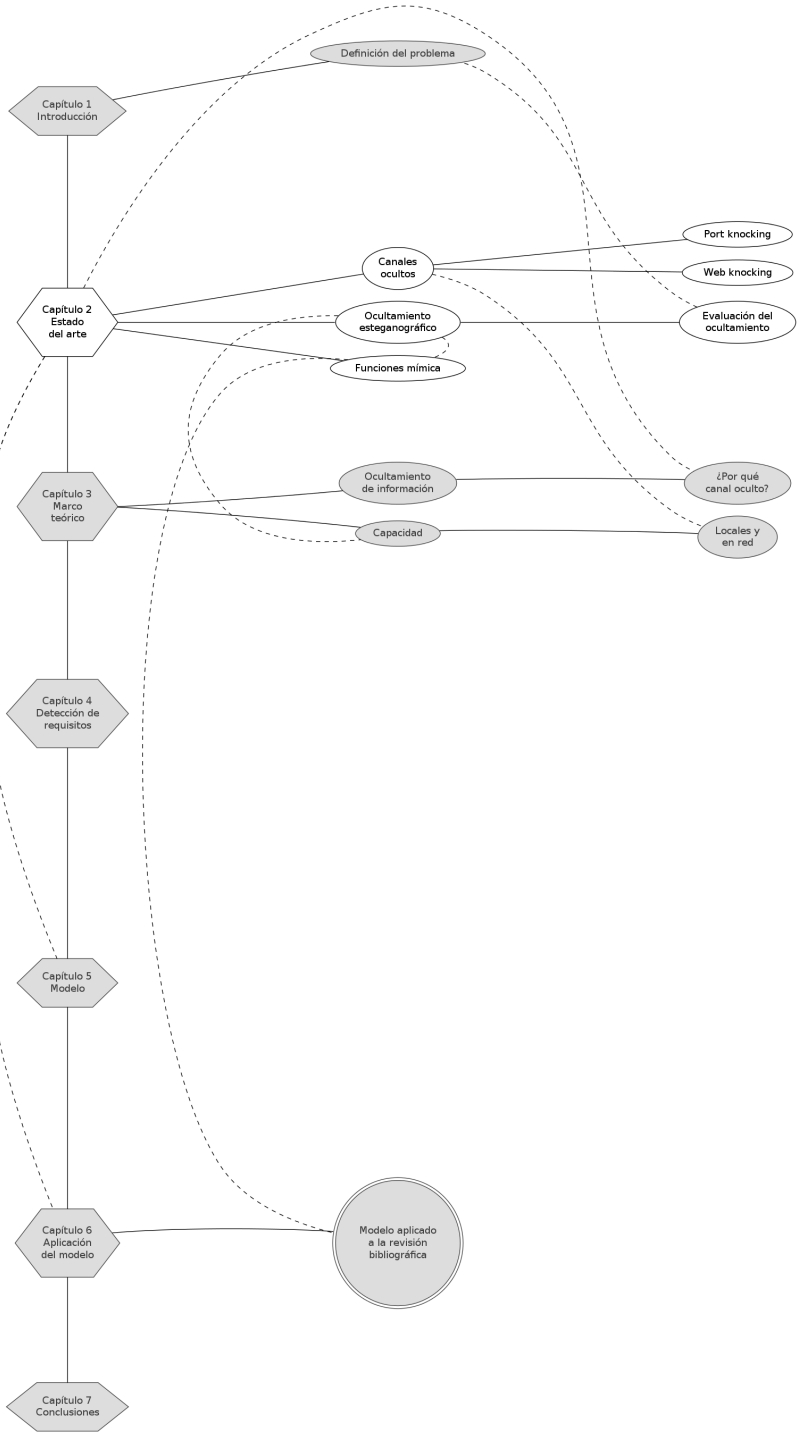


Figura 2.3: Relaciones conceptuales: Temas abordados en el capítulo 2

Capítulo 3

Marco teórico

En este capítulo se aborda el fundamento teórico requerido y empleado por las distintas propuestas expuestas en el capítulo anterior, sirviendo como base teórica para desarrollar la propuesta que se presentará en el capítulo 5.

3.1. Sistema de comunicaciones

En primer término, si se pretende por medio del presente trabajo modelar la comunicación sobre un canal oculto, no puede obviarse que éste debe ser visto como un sistema de comunicaciones. La referencia obligada para comenzar una discusión sobre ellos es Shannon (1948). En esta, se detallan los principales componentes que todo sistema de comunicaciones presentará; el presente modelo construye sobre su diagrama esquemático general; en la sección 5.1 se profundizará al respecto. Los componentes de un sistema de comunicaciones son los que ilustra la figura 3.1.

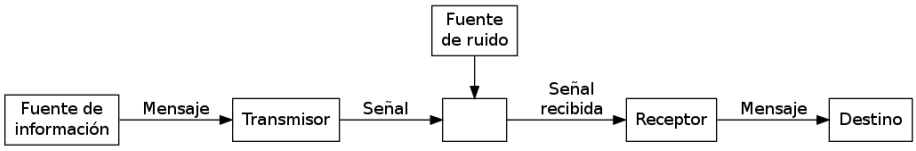


Figura 3.1: Diagrama esquemático de un sistema de comunicaciones (Shannon 1948)

3.1.1. Funciones de la comunicación

Resulta interesante evaluar también a un sistema de comunicaciones desde una perspectiva *humana*. Esto permitirá evaluar cuáles características de la comunicación (en un sentido muy amplio) cumple un sistema de comunicaciones,

sea entre dos personas o entre programas que intercambian datos de forma encubierta.

Roman Jakobson identificó en 1960 las seis funciones que conforman a todo acto de *comunicación verbal* o *evento lingüístico* a partir de los seis *factores fundamentales* de la comunicación (Waugh 1980; Hébert 2011).

Emisor El que habla, codifica o emite el mensaje; poeta, autor, narrador.

Receptor Quien decodifica, escucha, lee o interpreta el mensaje.

Código Sistema o lengua empleado para el mensaje.

Mensaje El texto o discurso mismo.

Contexto Referente necesario para la interpretación del mensaje.

Canal El canal físico, así como la conexión psicológica, entre emisor y receptor.

Un mensaje dado tendrá como eje principal a una de estas seis funciones, como lo muestra la figura 3.2. Estas funciones son, respectivamente:

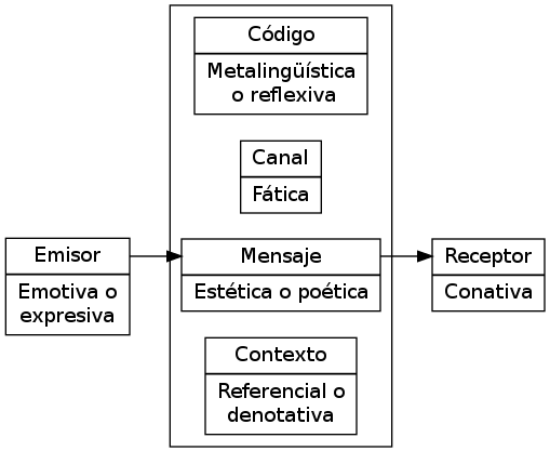


Figura 3.2: Los seis factores de la comunicación, así como sus funciones correspondientes, según la descripción de Jakobson.

Emotiva Se refieren al emisor, autoreflexivos. Son mensajes que no alteran el *significado denotativo* de la comunicación, pero agregan información acerca del estado interno del emisor. Por ejemplo, *«Estoy satisfecho»*, *«¡Qué buena vista!»*.

Conativa Solicitan acción directa al receptor (esto es, emplean voz imperativa, ruegos, o preguntas con sentido de provocar acción). Por ejemplo, *«Haz tu tarea»*, *«¿Me traerías mi mochila?»*.

Metalingüística Empleo del lenguaje para describirse a sí mismo. Ejemplos de esto incluirían cuando uno de los interlocutores pregunta al otro, *«¿A qué te refieres con "metalingüístico"?»*.

Estética / poética Enfoca al mensaje *en el mensaje mismo*. Puede apreciarse cuando la forma del mensaje es parte fundamental del mensaje, como en la poesía o en los *slogans* de la mercadotecnia.

Referencial Describe una situación, objeto, o estado mental dentro del cual debe enmarcarse el mensaje para ser mejor comprendido.

Fática La comunicación realizada buscando la interacción, clave para abrir, mantener, verificar o cerrar el canal de comunicación. Por ejemplo, iniciar una llamada telefónica con *«¿Bueno?»*[1]

Cada mensaje (incluso, fracciones de un mismo mensaje) transmitido sobre un canal de comunicación dado puede ubicarse como un punto en el espacio definido por estas seis dimensiones. La intencionalidad de una frase puede resultar particularmente alta o significativa en una de ellas, pero eso no necesariamente significa que resulte cero en todas las demás (como lo ilustra la nota al pie).

Y si bien esta clasificación está claramente enfocada a modelar la comunicación entre personas, determina también las diferentes etapas de una interacción por computadora. Se presentarán algunos ejemplos de esto en el capítulo 6.

3.2. Ocultamiento de información, canales ocultos y subliminales

El estudio formal de los canales ocultos se origina con Lampson (1973), que argumenta que, por más que se confine la información que pueden comunicarse legalmente dos procesos, el conjunto de interacciones que tengan *con el sistema* a lo largo del tiempo, así como medios no diseñados para fungir como canal de comunicación, pueden emplearse como conductos para la fuga de información. Lampson señala algunas medidas que, en el sistema hipotético que presenta, pueden instrumentarse para evitar esta fuga, aunque termina el artículo reconociendo que cerrar la fuga por completa es tan complejo que una alternativa más realista es limitar la capacidad de fuga restringiendo al máximo la desivación de patrones rígidos de comunicación.

La mayor parte de la literatura respecto a los canales ocultos se centra en cómo defenderse de ellos; citando a Millen (1999) (traducción propia),

> Los canales ocultos son medios de comunicación establecidos entre dos procesos que no tienen permitido comunicarse y aún así lo hacen,

[1] Frecuente en México, pero no en otros países de habla hispana, que iniciarían la llamada con *«¿Aló?»* o *«¿Hola?»* Nótese que la presente nota al pie sería *referencial*, dado que se refiere al *contexto* en que se realiza la comunicación, y dado que forma parte de una descripción acerca de las funciones de la comunicación, también sería *metalingüística*.

> de a pocos bits, afectando a los recursos compartidos. El ocultamiento de información es un poco diferente: Los dos interlocutores tienen permiso de comunicarse, pero el contenido es censurado y restringido a ciertos temas. El truco está en "montar" algunos datos de contrabando de forma invisible en contenido legítimo.

Más aún, Cabuk (2006) caracteriza a un canal oculto como (traducción propia):

> Los canales ocultos aparecen en los sistemas en que el acceso directo está prohibido por política. Por tanto, los canales ocultos son una construcción de ocultamiento de información relacionada pero distinta a la criptografía y la esteganografía.

El trabajo de Cabuk se centra en prevenir la aparición y el abuso de canales ocultos vistos exclusivamente como una *subversión de políticas de uso*. En su introducción presenta varios ejemplos abordados históricamente, incluyendo a la clasificación hecha por Schaefer y col. (1977) en canales *basados en almacenamiento* (aquellos que ocultan la información empleando el almacenamiento de datos no relacionados) y *basados en tiempo* (que emplean los patrones de uso de recursos del sistema como señalización).

Los *canales subliminales* se definen de forma mucho más rígida, a partir de la publicación del *Problema del Prisionero* en (Simmons 1983; Simmons 1985): Se refieren a los casos de canal oculto en que la comunicación se esconde en las firmas criptográficas de un mensaje cubierta (inocente en apariencia) enviado en claro.

Los escenarios presentados en la sección 1.5.1 mantienen como hilo conductor la importancia de ocultar la *identidad administrativa* del usuario que inicia la comunicación: Para no revelar que es un usuario administrativo, *disfrazará* su interacción de la que efectuaría un usuario cualquiera, incluso –si la naturaleza del sistema sobre del cual se implemente lo permite– de un usuario anónimo.

En la revisión de bibliografía académica realizada destaca un documento que explora el espacio para desarrollar canales ocultos sobre las diferentes capas del modelo de redes OSI (Handel y Sandford 1996), reconociendo –como también este trabajo lo hace– que es imposible para propósitos prácticos evitar la comunicación entre dos actores con acceso a un medio común; de especial relevancia para este trabajo, en sus conclusiones menciona que: (traducción propia)[2]

> La vulnerabilidad de un sistema de información es más una función de las habilidades de Alice y Bob para descubrir e implementar funcionalidad desconocida o ignorada dentro de la arquitectura del sistema. La seguridad del sistema de información es más una función

[2] El texto aquí citado hace referencia a los nombres informales mediante los cuales es común referirse en la literatura a los participantes de un criptosistema: Los usuarios legítimos *Alice* y *Bob*, el guardián *Walter*, la espía de comunicaciones *Eve*, la atacante activa *Mallory*, la intrusa *Trudy*, etc.

de las habilidades que Walter pueda aplicar para detectar anomalías sutiles en la comunicación entre Alice y Bob.

Rediseñar el sistema para contrarrestar a Alice y Bob puede llevar, y en muchos casos, llevará a la creación de otras oportunidades de explotación.

3.2.1. Espacios para el ocultamiento

Este trabajo abordará distintas técnicas que permiten ocultar la información. Ocultar la información incluye, pero excede, a las áreas tradicionalmente abordadas, criptografía y esteganografía — Citando a Kahn (1967) (traducción propia), el campo de estudio de la *criptología* incluye:

- Seguridad de las comunicaciones:
  - Esteganografía (tintas invisibles, códigos abiertos, mensajes en tacones vacíos) y seguridad de la transmisión (sistemas de radio en ráfagas).
  - Seguridad del tráfico (Cambios en señales de llamada, mensajes "maniquí", silencio de radio).
  - Criptografía (Códigos y cifrados, cifonía[3], cifax [4])
- Inteligencia en las comunicaciones:
  - Análisis de tráfico (ubicación de correcciones en la dirección, estudios al flujo de mensajes, huellas digitales de radio)
  - Criptoanálisis
- Seguridad electrónica:
  - Seguridad de emisiones (alteración de frecuencias de radar).
  - Contra-contramedidas ("leer a través" de un radar bajo interferencia).
- Inteligencia electrónica:
  - Reconocimiento electrónico (escuchar emisiones de radar)
  - Contramedidas (interferencia, falsos ecos de radar)

Desde el mismo léxico empleado, resulta claro que el campo abordado (y su comprensión en círculos no militares) ha cambiado fuertemente en los casi 50 años desde la publicación de este libro,[5] pero las definiciones presentadas

[3] El proceso de cifrar las señales de telecomunicaciones, para prevenir que la información sea interceptada por un enemigo o competidor.

[4] Comunicación facsimilar cifrada, en la que la salida de un generador de pulsos es combinada con la salida del *scanner* del fax.

[5] Kahn reeditó este libro en 1996, agregándole un capítulo, pero el contenido principal se mantuvo sin cambios. Es necesario apuntar que se trata de un libro con mayor énfasis en ser un estudio histórico que técnico.

resultan un muy útil punto de partida: El ámbito de desarrollo del presente trabajo es la seguridad de las comunicaciones, aunque no puede dejar de prestar atención a aspectos de la inteligencia en las comunicaciones.

Dentro de la clasificación presentada por Kahn es posible ubicar a distintas técnicas no expresamente abordadas. Por poner un ejemplo, el *esteganálisis* (la detección de mensajes esteganográficos) podría abordarse tanto dentro del *análisis de tráfico* en búsqueda de patrones u otros factores que delaten la presencia de comunicación oculta, aunque (de forma mucho más limitada, pero no por ello menos importanet) como parte de la seguridad de emisiones.

La asociación natural para muchos al plantear un estudio relacionado con la creación de un canal oculto es valerse de la esteganografía como principal herramienta. Sin embargo, y como se verá tanto al abordar a Wang y Lee (2005) como al desarrollar la sección 5.3, una amplia proporción de las implementaciones viajan más bien ocultas en la seguridad del tráfico.

3.2.2. ¿Por qué *canal oculto*?

En líneas generales, para el presente trabajo se entiende como *canal oculto* la comunicación que se realiza sobre un canal originalmente diseñado para comunicar información de distinta naturaleza, de distinto *nivel administrativo*. Dado que lo que se busca es presentar un modelo, reconoce que éste podría implementarse indistintamente sobre canal oculto, canal subliminal o mediante ocultamiento de información.

Más aún, si bien en ninguno de los ejemplos delineados se presenta una violación administrativa (en todos los casos se presentan ejemplos de un administrador de sistemas ingresando a un equipo en el que tiene legítimos derechos administrativos, por medio de una red pública que no establece expresamente políticas de tráfico aceptable), el mecanismo presentado requiere del sigilo, y es por ello que los mecanismos que emplean los canales ocultos, subliminales y esteganográficos se terminan hermanando.

Las diferencias entre dichos términos pierden relevancia, y se elige el término de *canal oculto* por ser el más natural desde un nivel lingüístico; en las primeras versiones de este trabajo se proponía el término de *canal discreto* para hermanar de forma menos ambígua a estos conceptos, pero caía en una polisemia peor al confundirse con el significado ampliamente utilizado en las matemáticas de *discreto* en contraposición de *contínuo*.

3.3. Capacidad del canal

Un canal de comunicaciones con las restricciones que impone la operación como *canal oculto* conlleva la necesidad de evaluar si el canal elegido brinda suficiente *espacio* para transmitir el mensaje requerido de emisor a receptor.

Resulta claro que un canal oculto necesariamente ofrecerá sólo una fracción de la capacidad disponible para el canal en claro; es ya conocido que los

canales ocultos pueden codificar la información sobre el *tiempo* o sobre el *almacenamiento* (Shieh 1996). En ambos casos, la información a transmitir ocuparía notablemente menos tiempo y espacio si se transmitiera en claro. Wang y Lee (2005) elabora sobre esta clasificación, llegando a cuatro clases de canales ocultos: Canal espacial basado en valores, canal espacial basado en transiciones, canal temporal basado en valores, y canal temporal basado en transiciones.

Una característica importante para no revelar la presencia de actividad administrativa en el modelo propuesto es el requerir del mínimo de transferencia de información para hacer llegar un mensaje; la capacidad del canal oculto que será propuesto para el modelo debe ser medible. La única medida que puede aplicarse, siguiendo las ideas presentadas ya desde Shannon (1948), es de la entropía que se puede codificar por símbolo empleando el mecanismo de codificación elegido. El inverso de dicho resultado indicará la longitud del mensaje requerido para codificar un mensaje del tamaño que se requiera transmitir.

Citando a Millen (1989) (traducción propia):

> El ritmo máximo de información de un canal oculto sin ruido de estado finito es la capacidad del canal, la cual usualmente puede calcularse utilizando la técnica sugerida por Shannon. La técnica de Shannon es suficientemente poderosa para dar una respuesta correcta cuando el tiempo de transición entre estados del canal no es uniforme. El modelo del canal sin ruido de estado finito es apropiado cuando el mecanismo del canal y los elementos temporales han sido identificados a detalle, y la estimación de este ritmo es obtenido en condiciones de "peor caso", en las cuales el canal no es sujeto a interferencia o ruido de procesos no confinados. Un ritmo de información cercano a la capacidad del canal puede lograrse por medio de la codificación, y puede ser significativamente mayor que un ritmo estimado en supuestos simplificantes, como podría ser el esperar una frecuencia igual de 0's y 1's.

Conviene rescatar la idea del *triángulo mágico* (Sehgal y Goel 2014), que indica que un mecanismo esteganográfico de comunicación debe ubicarse en el *triángulo mágico* (ilustrado en la figura 3.3) definido por tres vértices: *Imperceptibilidad*, *Robustez* y *Capacidad*; un canal oculto muy robusto y muy imperceptible naturalmente tendrá muy poca capacidad, mientras que si se da mayor peso a la capacidad e imperceptibilidad, el canal oculto resultante será muy poco robusto. Estas consideraciones entran particularmente en juego al evaluar la prioridad relativa de las propiedades de un canal propuesto, como será ilustrado al abordar los *requisitos de comunicación* en la encuesta realizada (véase la sección 4.1).

El triángulo mágico de Sehgal y Goel guarda una muy cercana correspondencia con la definición de *buena esteganografía* presentada en Codr (2009), inspirada originalmente en Salomon (2003): Los *objetivos* de la esteganografía. Como ilustra la figura 3.4, si bien el objetivo directo de la esteganografía es ocultar los datos, esto se logra óptmiamente mediante los siguientes seis subobjetivos:

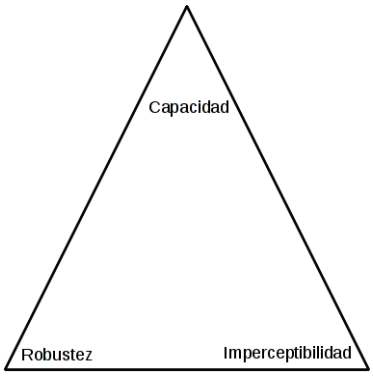


Figura 3.3: Triángulo mágico de los requisitos para el ocultamiento de información (Sehgal y Goel 2014)

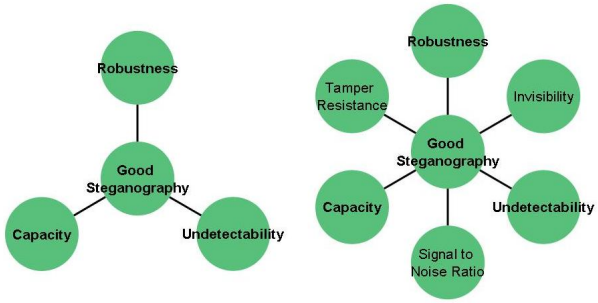


Figura 3.4: Los seis objetivos de la *buena esteganografía* (Codr 2009). A la izquierda, los tres principales — Guardando una muy cercana correspondencia con el triángulo de Sehgal y Goel.

Capacidad Cuánta información puede ser ocultada.

Indetectabilidad Imposibilidad a una computadora para diferenciar entre la *cubierta* y el *estego-objeto* por medio de estadísticas u otros métodos computacionales.

Robustez La capacidad de un mensaje de sobrevivir a la compresión u otras modificaciones comunes.

Invisibilidad o imperceptibilidad Imposibildad a los humanos para detectar distorsiones en el *estego-objeto*.

Resistencia a la modificación La capacidad de un mensaje de sobrevivir a medidas intencionales de destruirlo.

Relación señal a ruido Cuántos datos pueden ser codificados contra cuántos datos no relacionados forman parte del mensaje.

Los tres primeros (que *casi* coinciden con el triángulo de Sehgal y Goel) son calificados como primarios. La tensión existente entre ellos también es mencionada explícitamente en esta obra.

Puede apreciarse una importante diferencia entre las figuras 3.3 y 3.4: Al separar a mayor detalle los tres vértices principales, Codr hace claro la facilidad de enfocarse en un sentido equivocado, lo cual llevaría a menor fortaleza. Un esquema esteganográfico podría ser muy *imperceptible*, por ejemplo, como el uso del bit menos significativo en imágenes (Wayner 2009, págs. 157-172). Este esquema resulta, sin embargo, muy poco *indetectable*. Dado el modelo de amenazas que aquí se aborda implica a un contrincante con capacidad de automatización de procesos, la *indetectabilidad* resulta mucho más importante que la *imperceptibilidad*, sin embargo, bajo otros modelos de amenazas, la evaluación podría ser distinta.

3.3.1. Canales ocultos locales y en red

En las décadas de los setenta y ochenta, numerosos trabajos analizaron la capacidad de crear canales ocultos en sistemas operativos multiusuario con enfoque de separación de privilegios y política de seguridad obligatoria, como los que describe el *Libro Naranja* (*Trusted Computer System Evaluation Criteria* 1985), y los escenarios que describen son para la comunicación oculta *local*, dentro de un mismo sistema de cómputo. Varios sistemas certificados en las categorías propuestas por el Libro Naranja aparecieron en las décadas de los ochenta y noventa, pero –presumiblemente por su rigidez, así como por su falta de adecuación ante la evolución del cómputo profundamente ligado a las redes– han ido desapareciendo de la vista general en la comunidad de profesionales en seguridad en cómputo.

Varios textos se enfocan en la creación de canales ocultos dentro de sistemas como *Trusted XENIX*; un interesante ejemplo puede ser Tsai y Gligor (1988), que presenta varios canales –tanto basados en tiempo como basados en almcenamiento, e incluso hibridizándolos– sobre la semántica POSIX extendida ofrecida por dicho sistema. Y no porque el enfoque principal hoy en día sea claramente orientado a redes deben dejar de importar dichos trabajos. Por ejemplo, citando a Handel y Sandford (1996) (traducción propia):

> El problema (...) es complicado dado que debe diseñar varias medidas para analizar el contenido en busca de datos ocultos. Una bitácora de transacciones sólo tiene sentido si alguien está dispuesto a analizar e interpretar dicha bitácora. Hay ciertos grados posibles de automatización, pero los sistemas automatizados existentes están basados en la detección heurística de eventos anómalos. La comunicación oculta es posible porque las transacciones son ordinarias, y no lanzan ninguna alarma para dichos detectores.

Lo abordado en el texto anterior describe muy de cerca el funcionamiento de las primeras implementaciones de *port knocking*, abordadas en la sección

2.1.1: Los *golpes de puerto* a puertos cerrados son registrados por el firewall a la bitácora del sistema, y un proceso local trabaja leyendo a dicho archivo *del disco local* para llevar a cabo las acciones configuradas. Esto es, si bien la distancia entre un canal oculto local y uno basado en red puede parecer muy grande, convertir entre ellos es una acción a fin de cuentas trivial. El presente trabajo se enfoca a los canales basados en red, pero nada impide fundamentalmente que se aplique a canales locales.

3.4. Autenticación

Presentar un sistema que no realice una autenticación suficientemente rigurosa equivale a abrir una puerta al abuso indiscriminado del sistema, esto es, instalar un *backdoor* en el propio equipo que se pretende proteger. Y si bien todo programa escrito por humanos es falible, por su naturaleza repetitiva (la sección 3.4.3 presenta una mayor discusión al respecto) debe enfocarse especial atención a que no se convierta en el punto débil del sistema

3.4.1. Usuario y contraseña

El mecanismo de autenticación más empleado del mundo, pese a sus grandes insuficiencias, es el basado en nombre de usuario y contraseña. Este mecanismo, si bien es fácil de implementar y no requiere ningún hardware adicional, es recooncido como muy débil ya desde la década de los setenta —la predictibilidad y desidia humana llevan al uso de contraseñas demasaido débiles (Morris y Thompson 1979).

Se han efectuado análisis de frecuencia sobre contraseñas obtenidas tras los casos de bases de usuarios filtradas, encontrando que un 40 % de las contraseñas empleadas caen en las primeras 100, y un 91 % en las primeras 1000 (Burnett 2011).

Figura 3.5: Nube con las contraseñas más frecuentemente utilizadas, con tamaños asignados según su frecuencia relativa (Burnett 2011)

Por otro lado, los mecanismos más fuertes de autenticación, como la basada en datos biométricos o multifactorial, requieren no sólo de una mayor cantidad de

bits que un par usuario-contraseña, sino que de hardware que el administrador no siempre tendrá disponible, particularmente, no en varios de los escenarios descritos en la sección 1.5.1.

Distintas situaciones requieren respuestas acorde a su realidad. Es por ello que el NIST contempla cuatro niveles de seguridad (Burr y col. 2013), cada uno de ellos empleando mecanismos más seguros. Pero correspondientemente, cada nivel de seguridad resulta más caro e inconveniente que los inferiores para su uso casual; resulta claro por qué que la recomendación de NIST busca (citado de su resumen, traducción propia, énfasis agregado):

> Proveer lineamientos técnicos para *agencias federales* implementando autenticación electrónica, y no pretende constreñir el desarrollo o uso de estándares fuera de este propóisto. La recomendación cubre la autenticación remota de usuarios (...) que interactúen con sistemas tecnológicos *del gobierno* sobre redes abiertas".

3.4.2. Fortaleza de la autenticación

Los diferentes esquemas de autenticación pueden sintetizarse a *cuántos bits de entropía* emplean para la información de autenticación. Por ejemplo, en un esquema tradicional usuario-contraseña, donde el nombre de usuario pueda asumirse como públicamente divulgado y la contraseña sea de hasta 8 caracteres alfanuméricos (Burnett 2005, pág. 28) llevan a un espacio muestral de apenas 48 bits ante una búsqueda ciega de fuerza bruta.[6] Cabe recalcar que, como lo ilustra la figura 3.5, las contraseñas muy rara vez siguen una distribución aleatoria, por lo cual la complejidad real de búsqueda es mucho menor.

Siguiendo las recomendaciones de niveles de seguridad (Smart 2012, pág. 44), resulta claro que la gran mayoría de las contraseñas en uso hoy en día quedan *muy por debajo* de la *protección a muy corto plazo contra organizaciones pequeñas*. De ahí la recomendación que la elección de un mecanismo de autenticación para un esquema de comunicación como los que describe este modelo busque la mayor fuerza criptográfica posible.

Ahora bien, aunque la recomendación indudable sea usar mecanismos de autenticación tan largos como sea posible, como se aborda en la sección 3.3, el espacio disponible en un canal oculto típicamente resulta muy limitado; resultará necesario encontrar un justo medio que equilibre fortaleza y brevedad.

La elección de las fuentes de entropía y de una adecuada forma de combinarlas para lograr números aleatorios suficientemente fuertes, son abordados por Eastlake, Schiller y Crocker (2008).

En resumen, entre más bits de entropía puedan expresarse como parte de la cadena de autenticación, más segura será esta ante ataques; la recomendación al día de hoy sería tomar como mínimo absoluto el empleo de 72 bits, y de preferencia 80.

[6] Prácticamente la totalidad de sistemas que emplean autenticación usuario-contraseña hoy en día permitirían una fuerza mucho mayor que esta, sin embargo, estas características aún tenidas como válidas por una gran proporción de los usuarios del mundo.

Debe ser posible estimar de la forma más clara posible la entropía y resistencia alcanzables; estas no son únicamente función de la longitud de la cadena a transmitir, sino del proceso mediante el cual se obtuvo dicha cadena. Una recomendación general es el uso de funciones *hash* criptográficamente fuertes. Persiguiendo, por lo expuesto en la sección 3.3, un balance entre fortaleza y tamaño, hoy en día resulta recomendable emplear las familias de hashes SHA1,[7] SHA2 y SHA3 (NIST 2012; Dang 2012).

3.4.3. Ataques de reproducción

Cualquier esquema de autenticación empleado debe ser resistente a los ataques de reproducción (*replay attacks* o *playback attacks*). Esto es, aunque el adversario capturara el tráfico de red, lo analizara, y determinara que cierta conexión es un comando administrativo, esto no debería servirle para establecer una nueva sesión.

Hay muchos mecanismos para prevenir ataques de reproducción. Cuando el canal es bidireccional y sus condiciones permiten el establecimiento de una sesión, resulta atractivo el establecimiento de una llave de sesión empleando mecanismos criptográficos como los muchos derivados del trabajo de Diffie y Hellman (1976).

En circunstancias donde no es posible hacer un intercambio de información suficiente para calcular una llave de sesión, como muchas veces resultarán los escenarios a los que se enfoca el presente trabajo (en que la identidad debe establecerse con un intercambio mínimo de mensajes), la cadena de autenticación debe ser modificada. Si el equipo a verificar la autenticación es el mismo que el que la recibe (como una computadora que recibe usuario y contraseña desde la consola, o cuando hay un canal confiable sobre del cual se transmite la contraseña), la estrategia más frecuentemente utilizada es el agregarle *sal* a la cadena de autenticación: Un valor aleatorio y conocido, que al agregarse a la cadena de autenticación entregará siempre un resultado distinto pero predecible para ambos participantes legítimos de la comunicación (Morris y Thompson 1979). Sin embargo, este esquema funciona únicamente si la cadena de autenticación llega en claro al sistema que pueda verificarla, o si es posible enviar la *sal* al equipo remoto.

Cuando el cliente debe transmitir de una sola vez sus credenciales al servidor, cuidando de no ser vulnerables a un ataque de reproducción, pueden utilizarse distintos mecanismos *cambiantes* y *predecibles* de modificación de la cadena de autenticación: *vectores de inicialización* o *nonces*, cadenas pseudoaleatorias que *no* tienen el requisito de permanecer ocultas — únicamente el de *nunca repetirse* (Rogaway 2004).

La definición del protocolo SSL (Dierks y Rescorla 2008, págs. 90–96) incluye una revisión de los conceptos aquí presentados, abordados desde la perspectiva

[7] Se conocen ya debilidades en el algoritmo SHA1 que llevan a que su fortaleza inicial, supuesta de 80 bits, sea en realidad de sólo 69 bits (Schneier 2005), colocándolo únicamente como *marginalmente seguro*. (Smart 2012)

del establecimiento de una sesión, que es sólo una de las modalidades abordadas en el modelo propuesto (véase la sección 5.3).

3.5. Resumen del capítulo

Este artículo presentó los fundamentos teóricos para construir sobre las nociones de un sistema de comunicaciones y presentó la noción de *canal oculto*, así como las definiciones relacionadas de *canales subliminales* y *esteganografía*, indicando la razón por la cual se eligió el primero de estos términos para el desarrollo del trabajo. Dado que estos puntos son el fundamento de todo análisis que se haga a un canal de comunicación, estos conceptos son la base sobre la cual se construyen las implementaciones reseñadas a lo largo del capítulo 2, y serán retomados todo a lo largo del capítulo 5.

Al hablar de la capacidad del canal, se mencionan además las características que definen al *triángulo mágico* dentro del cual se enmarca cualquier comunicación oculta. Estos conceptos se abordan nuevamente al definir el modelo, en la sección 5.3.

Por último, se hizo una breve revisión acerca de la autenticación, las desventajas del sistema de autenticación más empleado a nivel mundial (usuario/-contraseña), la medición de la fortaleza de un esquema de autenticación, y de los importantes riesgos (particularmente si se habla, como lo hace el presente trabajo, de comunicación en red) de los ataques de reproducción. Las consideraciones de autenticación, además de ser señaladas como fundamentales por los participantes de la encuesta aplicada (sección 4.1), se consideran en el transcurso de los esquemas que presenta el capítulo 2 y las secciones 5.4 y 5.5 — y, claro está, al aplicar el modelo a todos los mecanismos de canal oculto que aborda el capítulo 6.

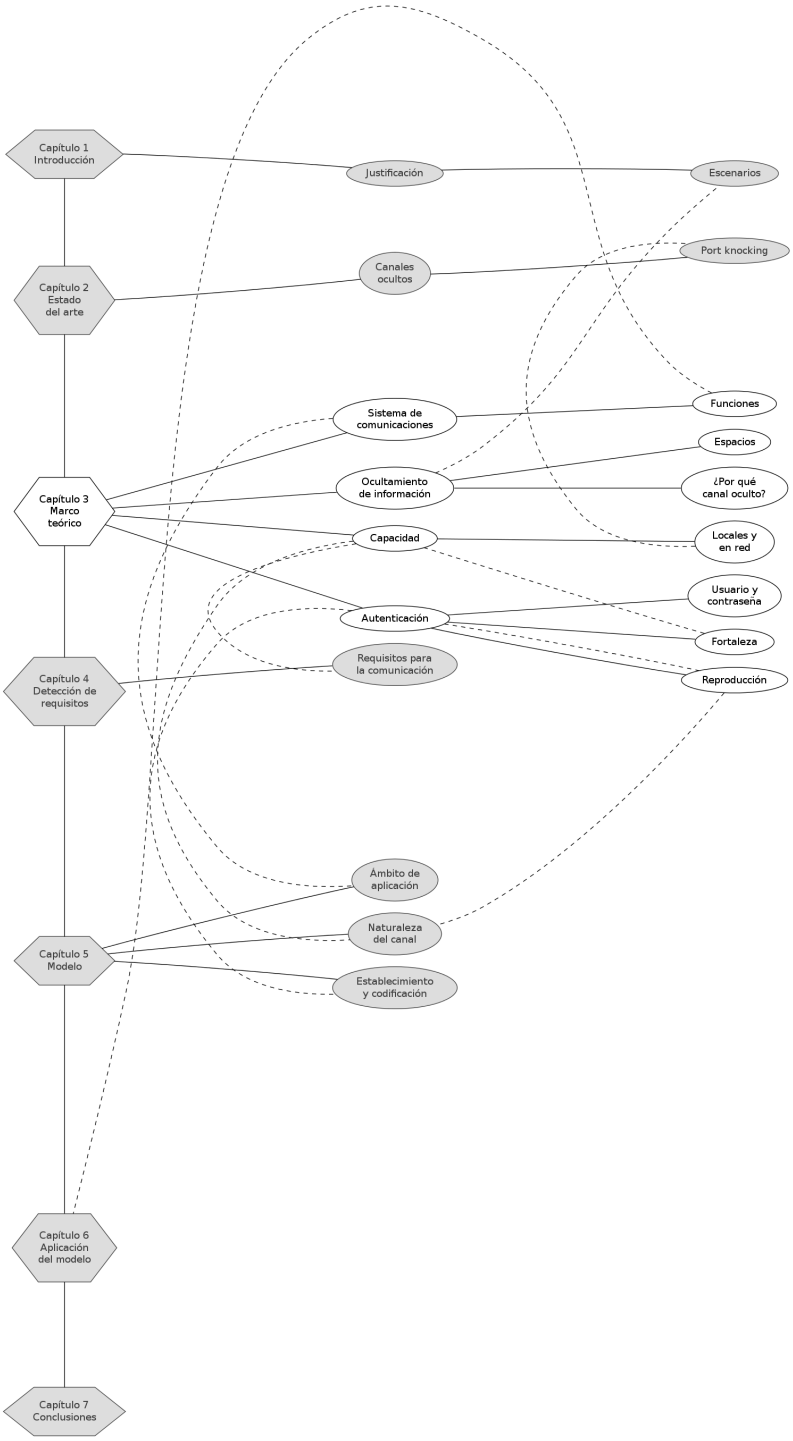


Figura 3.6: Relaciones conceptuales: Temas abordados en el capítulo 3

Capítulo 4

Detección de requisitos

El planteamiento completo de este trabajo parte de la suposición de validez de las afirmaciones que discute el capítulo 1. Si bien la revisión bibliográfica realizada a lo largo de los siguientes capítulos ayuda a sustentar lo afirmado, se estimó importante descubrir la opinión colectiva de un grupo de expertos en la administración de sistemas, redes y seguridad informática para resolver si las hipótesis de inicio son válidas.

Se aplicó una encuesta abierta de diseño transeccional descriptivo (Hernández Sampieri, Fernández Collado y Baptista Lucio 2006), enviando invitaciones para la participación a diversos grupos de desarrolladores de software y administradores de sistemas de habla hispana.[1] Esta encuesta buscó validar los puntos que formaron parte del modelo y comprender las prioridades relativas que los participantes daban a sus distintas opciones.

La encuesta fue presentada a los participantes sin dar un preámbulo personalizado, invitando a participar con el siguiente texto:

> ## Encuesta sobre las necesidades para la implementación de un canal oculto
>
> El objetivo de la presente encuesta es validar y adecuar un proyecto de desarrollo, enfocado a la creación de un modelo para la administración remota de servidores transportado sobre un canal oculto.
>
> Los esquemas para administración remota de servidores sobre canales cifrados (como `ssh`, Secure Shell) resultan hoy en día fundamentales, pero insuficientes, para la gestión de servicios y para una correcta respuesta a incidentes.
>
> El modelo que el presente trabajo pretende desarrollar busca resolver una serie de necesidades que no ha sido cubierta por las herramientas más ampliamente difundidas. Este modelo se enfoca a los

[1] Esta encuesta no puede tomarse como estadísticamente significativa; al haber sido aplicada a personas relacionadas con la ocupación laboral e ideológica del autor, es comprensible que presente un inevitable sesgo.

> canales ocultos: Mecanismos de comunicación que típicamente han sido aprovechados por los intrusos; la mayor parte de la bibliografía al respecto se enfoca a la prevención de la existencia de canales ocultos, siendo que cada vez más se vuelven necesarios para una gestión completa y proactiva de la seguridad.
>
> Dado que este tema no ha sido abordado con suficiente profundidad, resulta necesario el desarrollo de un modelo delineando los principales componentes con que un canal de esta naturaleza debe cumplir.
>
> Hay una gran cantidad de razones que pueden llevar a considerar el establecimiento de un canal oculto. Al no contar con un modelo desarrollado, quienes busquen cubrir esta necesidad se enfrentan a ir desarrollando sobre la marcha, probablemente obviando pasos importantes que deben ser cubiertos.
>
> Hay varios escenarios que podrían requerir de un medio de comunicación oculto, pero en líneas generales pueden resumirse en que hay diversas situaciones en que el administrador debe realizar un cambio en sus equipos sin alertar a terceros que puedan estar monitoreando la comunicación (sea en la red desde la cual se conecta o en el sistema destino a administrar).
>
> *Nota:* Donde el sistema presente opciones a priorizar, trate al 1 como "absolutamente en desacuerdo", al 3 como "neutral o indiferente", al 5 con "absolutamente de acuerdo".

La encuesta constó de seis preguntas, separadas en tres grupos (se indica entre paréntesis la sección en que se abordan):

- Requisitos para la comunicación (4.1)
- Relevancia del trabajo (4.2)
- Datos demográficos (4.3)

A excepción de la segunda pregunta del primer inciso (en que se pide a los encuestados proporcionar un listado de otros puntos a considerar), todas fueron de opción múltiple, para facilitar su análisis cuantitativo.

La encuesta fue iniciada por 210 personas (y completada únicamente por 97) a lo largo de 20 días. Para el análisis, se consideran únicamente las 97 respuestas completas.

El autor envió la invitación a participar en esta encuesta a:

- Grupo de administradores de sistemas de la Universidad Nacional Autónoma de México.
- Lista de discusión general de administradores de sistemas, con población principalmente chilena.

- Grupos de usuarios interesados en la privacidad y el anonimato en línea, basado en México pero con participación internacional.
- Lista en español de usuarios del sistema operativo Debian, de participación internacional.
- De forma personalizada, al círculo social del autor.

Sin embargo, siendo una encuesta de participación abierta, no hay garantía de que todos los participantes provinieran de estos grupos.

El breve análisis que a continuación se describe es meramente descriptivo; un trabajo a mayor profundidad podría abordar cruzamientos de datos y visiones parciales, estudiando las respuestas resultantes de la aplicación de criterios de filtrado. Sin embargo, el autor estima que dicho nivel de análisis sólo tendría sentido en una encuesta aplicada a una población más significativa y menos sesgada que la que aquí se aborda.

4.1. Requisitos para la comunicación

La primera sección se encamina a servir de base para fundamentar los criterios sobre los cuales se desarrolla el modelo (véase el capítulo 5). En esta sección se pregunta a los encuestados cuáles serían sus prioridades para el establecimiento y uso de un canal oculto.

La primera pregunta es «¿Qué tan importante le parece cada uno de los siguientes aspectos para la creación de un canal seguro?», con los siguientes once incisos:

a. Baja latencia

b. Disponibilidad de amplio ancho de banda

c. Resistencia a la generación de falsos eventos administrativos positivos

d. Capacidad de bloqueo selectivo

e. Capacidad de viajar sobre transportes anonimizantes (p.ej. TOR)

f. Resistencia a ataques de denegación de servicio

g. Imperceptibilidad del mecanismo de comunicación

h. Capacidad de pasar inadvertida por firewalls, IDS

i. Utilización de distintos canales

j. Cantidad de información que pueda ser transmitida

k. Fortaleza del mecanismo de autenticación

Los resultados se presentan en el cuadro 4.1 y en la figura 4.1.

Se solicitó a los participantes responder calificando a cada uno de estos aspectos del 1 (*absolutamente en desacuerdo*) al 5 (*absolutamente de acuerdo*). Para la encuesta, se definió la respuesta a cada uno de los incisos como de caracter obligatorio.

De las respuestas a estas preguntas se obtienen las siguientes observaciones:

- El aspecto más importante para los participantes es indiscutiblemente el *k*, con 76 participantes calificándolo con 5, y sólo 9 con 1.
- Le sigue el *f*, con 56 por 5 y sólo 7 por 1.
- Ninguno de los incisos muestra una preferencia por excluirlo (en ningún caso hay más respuestas por 1 o 2 que por 4 o 5).
    - Hay, sin embargo, cuatro casos en que el sentimiento general muestra una tendencia hacia la indiferencia, en que los votos por 3 son iguales o mayores a los votos por 5: *a*, *b*, *i* y *j*.
    - Cabe recalcar que en ningún caso los votos por 3 son superiores a la suma de los positivos (4 y 5); en el caso de *b*, la suma de las preferencias 4 y 5 es igual a la obtenida por la 3.

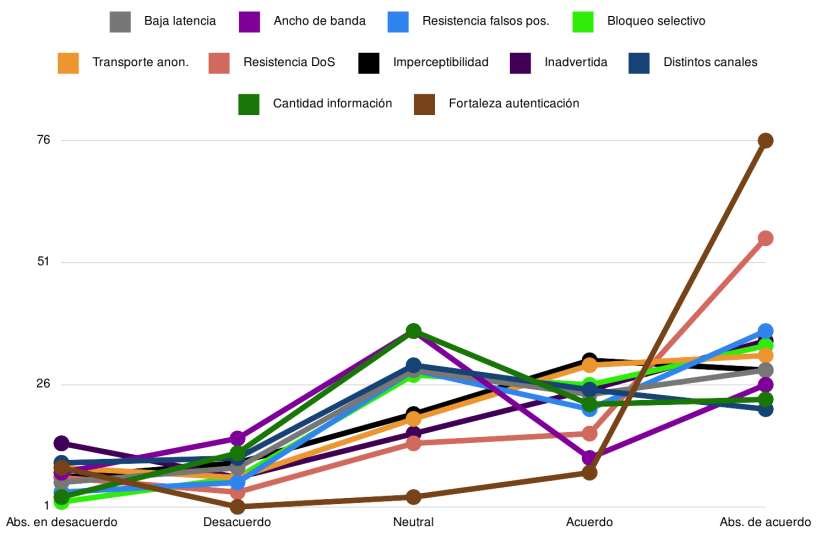


Figura 4.1: Resumen de las respuestas de la pregunta 1, «¿Qué tan importante le parece cada uno de los siguientes aspectos para la creación de un canal seguro?»

Cuadro 4.1: Respuestas a cada inciso de la pregunta 1, «¿Qué tan importante le parece cada uno de los siguientes aspectos para la creación de un canal seguro?»

	Baja latencia	Disponibilidad de amplio ancho de banda	Resistencia a la generación de falsos eventos administrativos positivos	Capacidad de bloqueo selectivo
Abs. en desacuerdo	6	8	4	2
En desacuerdo	9	15	6	7
Neutral o indiferente	29	37	29	28
De acuerdo	24	11	21	26
Abs. de acuerdo	29	26	37	34
	Capacidad de viajar sobre transportes anonimizantes (p.ej. TOR)	Resistencia a ataques de denegación de servicio	Imperceptibilidad del mecanismo de comunicación	Capacidad de pasar inadvertida por firewalls, IDS
Abs. en desacuerdo	9	7	7	14
En desacuerdo	7	4	10	7
Neutral o indiferente	19	14	20	16
De acuerdo	30	16	31	25
Abs. de acuerdo	32	56	29	35
	Utilización de distintos canales	Cantidad de información que pueda ser transmitida	Fortaleza del mecanismo de autenticación	
Abs. en desacuerdo	10	3	9	
En desacuerdo	11	12	1	
Neutral o indiferente	30	37	3	
De acuerdo	25	22	8	
Abs. de acuerdo	21	23	76	

Como segunda pregunta, aunque supeditada a la anterior, se solicitó a los encuestados responder (a texto abierto) a la pregunta «¿Hay algún punto adicional que le parezca importante para el establecimiento de un canal oculto que no haya sido considerado?»

Catorce de los encuestados respondieron a esta pregunta; eliminando cuatro respuestas negativas ("no", "ninguno", etc.), se obtuvo:

1. Cifrado de datos dependiendo de la importancia de la información enviada.
2. Comunicacion en ambas direcciones, como el tunel inverso de SSH.
3. Estabilidad del servicio; mecanismos de comprobacion anti spoofing o MITM.
4. Que sea distribuido. Que no haya que pasar por un punto central.
5. Capacidad de persistencia pasando por redes de intercambio; Reducir lo mas posible la demanda de recursos necesarios para ejecutarse.
6. Control de acceso discrecional, revocación de acceso.
7. No creo tener un punto adicional, sin embargo creo que un sistema. que pase de forma transparente a través de un FW, un IDS, un IPS, etc., representaría un vehículo para algún atacante con habilidad.
8. Que el código fuente este disponible bajo una licencia que respete las libertades del usuario.
9. No dejar rastros en los logs del servidor; No necesitar ser superusuario para instalarlo (similar a mosh); mantenerse conectado a traves de cambios de IP (similar a mosh).[2]
10. La posibilidad de ser recuperado en caso de ser compremetido.

De las respuestas obtenidas, cuatro van relacionadas con la robustez de la conexión (respuestas 3, 4, 5 y 10). Algunas abordan facilidades o conveniencia específica ante el usuario (respuestas 2, 6, 8, 9). La #1 expresa nuevamente la preocupación en un cifrado acorde a la sensibilidad de la información, y la #7 expresa su preocupación acerca de la utilidad del canal oculto para un atacante.

4.2. Relevancia del trabajo

La segunda sección de la encuesta busca averiguar qué tan relevante resultará la propuesta del presente trabajo a los encuestados, así como sentir el pulso de

[2] El programa `mosh` (*Mobile Shell*), es una aplicación terminal remota implementando prestaciones comparables con las de `ssh`, pero que tolera conectividad intermitente a red, sostener una sesión aún si el cliente *migra* de una dirección IP a otra, y otras características comunes de las conexiones desde dispositivos móviles (Winstein y Balakrishnan 2012).

la importancia que el grupo encuestado da a las diversas formas de publicacíón en el desarrollo de soluciones de seguridad informática.

Estando el presente trabajo orientado a la obtención de un grado académico, el autor reconoce que buena parte de los trabajos citados en este no son publicaciones científicas formales y arbitradas; este punto abona a comprender este hecho como parte de los usos de la comunidad de profesionales de la seguridad.

La primera pregunta de esta sección es: «¿Qué tan de acuerdo está con las siguientes afirmaciones, en el contexto de la utilidad para la realización de su trabajo diario u otras actividades cotidianas?»

Esta pregunta obedece a la misma lógica que la primera: Cinco opciones, de *absolutamente en desacuerdo* a *absolutamente de acuerdo*, y con respuesta obligatoria. Los resultados se muestran en el cuadro 4.2 y en la figura 4.2.

a. Considero que el olfateo (sniffing) de redes es un problema importante

b. Basta con el empleo de mecanismos criptográficos fuertes para administrar sistemas

c. Al responder ante un incidente, es más importante la agilidad y simplicidad que el sigilo

d. Utilizaría un mecanismo de administración remota sobre un canal oculto

e. Puedo caracterizar en 5 o 10 acciones los comandos que típicamente daría a un servidor en emergencias

Un análisis a las respuestas a esta pregunta brinda los siguientes resultados:

- Las respuestas más positivas se obtuvieron, en órden de preferencia, a los incisos *d* (39 por 5, 9 por 1, y una pendiente de preferencias muy clara), *c* (29 por 5, 4 por 1, pero con el máximo en 4, con 34) y *a* (30 por 5, 5 por 1, y un leve pico en 4, con 31).
- La respuesta a *b* es la única en toda la encuesta en que los participantes se manifestaron mayoritariamente en contra: Resulta claro (y, recalco, sin que los encuestados conocieran más respecto al presente trabajo) que no basta con una capa criptográfica fuerte para la administración de sistemas.
- Las preferencias respecto a *e* resultan más complicadas: Si bien es mayoritariamente positivo (46 para 4 y 5 contra 24 para 1 y 2), el pico es neutral (27 para 3). Nuevamente, esto puede deberse a que la encuesta se realizó en abstracto, no aterrizando a ningún tipo de canal oculto en particular.

Cuadro 4.2: Respuestas a cada inciso de la pregunta 3, «¿Qué tan de acuerdo está con las siguientes afirmaciones, en el contexto de la utilidad para la realización de su trabajo diario u otras actividades cotidianas?»

	Considero que el olfateo (sniffing) de redes es un problema importante	Basta con el empleo de mecanismos criptográficos fuertes para administrar sistemas	Al responder ante un incidente, es más importante la agilidad y simplicidad que el sigilo
Abs. en desacuerdo	5	13	4
En desacuerdo	11	26	11
Neutral o indiferente	20	28	19
De acuerdo	31	18	34
Abs. de acuerdo	30	12	29
	Utilizaría un mecanismo de administración remota sobre un canal oculto	Puedo caracterizar en 5 o 10 acciones los comandos que típicamente daría a un servidor en emergencias	
Abs. en desacuerdo	9	13	
En desacuerdo	11	11	
Neutral o indiferente	14	27	
De acuerdo	24	23	
Abs. de acuerdo	39	23	

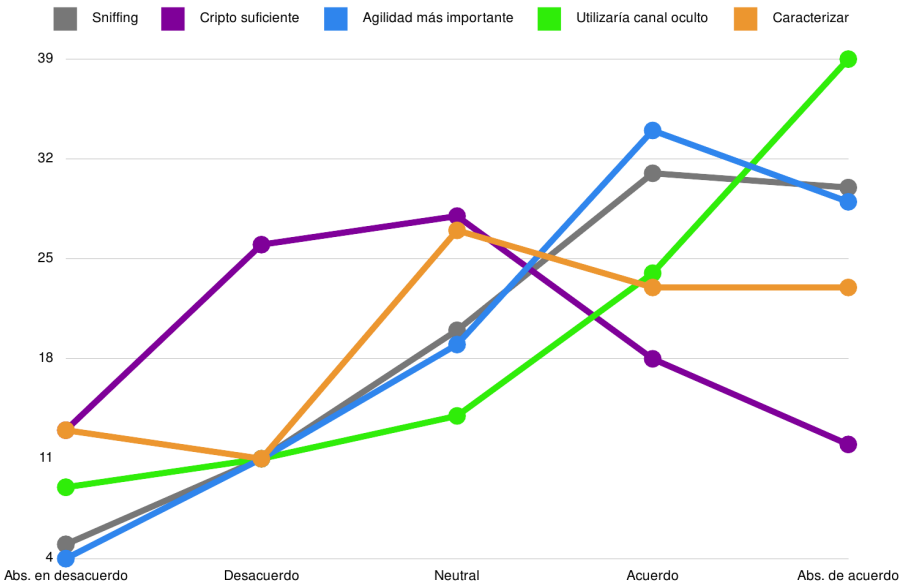


Figura 4.2: Resumen de las respuestas a la pregunta 3, «¿Qué tan de acuerdo está con las siguientes afirmaciones, en el contexto de la utilidad para la realización de su trabajo diario u otras actividades cotidianas?»

La segunda pregunta de esta sección (y cuarta de la encuesta) fue «Al desarrollar o implementar una solución relativa a seguridad informática, ¿qué tan importante le resulta seguir los pasos que otras personas han delineado? ¿Qué tan frecuente es que consulte los siguientes tipos de documentos?»

Esta pregunta también pedía indicar, de *absolutamente en desacuerdo* a *absolutamente de acuerdo*, cada respuesta. A diferencia de las preguntas anteriores, las respuestas a esta pregunta son opcionales, indicándoselo a los encuestados por medio de una columna *Ningun valor* y del texto «Si usted no desarrolla o implemena soluciones de seguridad, puede dejar estas respuestas en blanco.»

Los resultados se muestran en el cuadro 4.3 y en la figura 4.3.

Los incisos a responder fueron:

a. Publicaciones no formales: Páginas Web, blogs, etc.

b. Comienzo a desarrollar sin mucha investigación previa, voy resolviendo las necesidades sobre la marcha

c. Código fuente de otros sistemas que implementen algo similar

d. Documentación provista por mis proveedores de infraestructura

e. Artículos en revistas académicas, tesis, ponencias en congresos

f. Estándares oficiales (FIPS, NIST, NOM, etc.)

Esta pregunta fue respondida únicamente por cerca de dos tercios de los encuestados. Algunos puntos que llaman la atención en esta encuesta son:

- Presenta un comportamiento muy curioso: Es la única donde, en 5 de 6 casos, la opción 4 "vence" a la 5. Una posible explicación para esto es que, dado que esta pregunta de cierto modo *cuestiona* las buenas prácticas en el ejercicio profesional del encuestado, sus respuestas tienden a ser más cautas.

  Esto contrasta, claro está, con el comportamiento descendente del inciso *b*, en el que relativamente pocos (7 por el 5, 8 por el 4 — Pero aún así, cerca del 25 % de los que respondieron) admitieron lanzarse al desarrollo de una solución en seguridad sin realizar investigación previa.

- Los resultados para *c*, *d* y *e* son muy cercanos, y dependen fuertemente de dónde se haga el corte para leerlos:

  - Si se toman únicamente las respuestas de 5, *d* domina por 1 sobre *c*, y éste por dos sobre *e*.
  - Si se toman las de 4 y 5, *c* domina por 1 sobre *d* y éste por 2 sobre *e* (y los tres sobrepasan ligeramente el 50 % de su participación).
  - Si se toman todas las respuestas no negativas (3, 4 y 5), *e* domina por 2 sobre *d*, pero *c* cae fuertemente hasta seis registros por detrás.

- Aunque *a* y *f* quedan en los primeros lugares en 5, su comportamiento es mucho más conservador para todos los demás puntos.

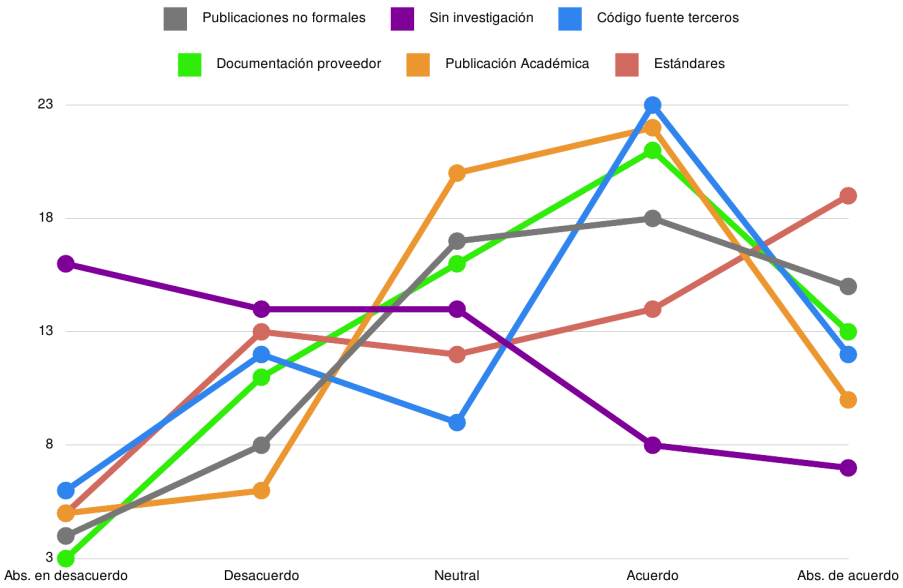


Figura 4.3: Resumen de las respuestas a la pregunta 4, «Al desarrollar o implementar una solución relativa a seguridad informática, ¿qué tan importante le resulta seguir los pasos que otras personas han delineado? ¿Qué tan frecuente es que consulte los siguientes tipos de documentos?»

Cuadro 4.3: Respuestas a cada inciso de la pregunta 4, «Al desarrollar o implementar una solución relativa a seguridad informática, ¿qué tan importante le resulta seguir los pasos que otras personas han delineado? ¿Qué tan frecuente es que consulte los siguientes tipos de documentos?»

	Publicaciones no formales: Páginas Web, blogs, etc.	Comienzo a desarrollar sin mucha investigación previa, voy resolviendo las necesidades sobre la marcha	Código fuente de otros sistemas que implementen algo similar
Abs. en desacuerdo	4	16	6
En desacuerdo	8	14	12
Neutral o indiferente	17	14	9
De acuerdo	16	8	23
Abs. de acuerdo	15	7	12
Respuestas totales	60	59	62
	Documentación provista por mis proveedores de infraestructura	Artículos en revistas académicas, tesis, ponencias en congresos	Estándares oficiales (FIPS, NIST, NOM, etc.)
Abs. en desacuerdo	3	5	5
En desacuerdo	11	6	13
Neutral o indiferente	16	20	12
De acuerdo	21	22	14
Abs. de acuerdo	13	10	19
Respuestas totales	64	63	63

4.3. Datos demográficos

El último inciso se enfocó a conocer al público que respondió a la encuesta, en dos sentidos particulares: Su área de especilización laboral (detallado en el cuadro 4.4 y la figura 4.4) y los grupos de interés personal particular a los que pertenece (detallado en el cuadro 4.5)y la figura 4.5.

En ambos casos, la pregunta fue de selección múltiple sobre una lista (permitiendo la respuesta en más de un rubro por cada participante).

Estas dos preguntas pretenden medir dos aspectos que guardan una fuerte correlación, aunque no son idénticos: Para la pregunta 5, el área de especialización laboral del encuestado, y para la pregunta 6, qué es lo que lo lleva a interesarse por cuestiones de seguridad en cómputo.

La pregunta 5 muestra que las respuestas provienen efectivamente del grupo poblacional correcto:[3] 75 de los encuestados son administradores de sistemas, 40 pertenecen a la categoría que a últimos años se ha popularizado como *dev-ops* (desarrolladores y administradores a la vez), 40 realizan actividades de soporte técnico, 35 son administradores de red, y de ese punto hacia abajo la densided de respuestas baja drásticamente.

Cuadro 4.4: Respuestas a la pregunta 5, «¿Cuál o cuáles de las siguientes opciones describen mejor su área de especialización laboral?»

	Cantidad	Porcentaje
Administrador de sistemas (sysadmin)	75	77.32 %
Desarrollador y operador de sistemas (dev-op)	40	41.24 %
Soporte técnico / atención a usuarios	40	41.24 %
Administrador de redes (netadmin)	35	36.08 %
Investigador / académico	25	25.77 %
Administrador de bases de datos (DBA)	21	21.65 %
Usuario final	18	18.56 %
Soy responsable de un grupo de expertos	16	16.49 %
Consultor en seguridad	16	16.49 %
Pen-tester	9	9.28 %
Perito informático	3	3.09 %

Por último, el texto presentado para la pregunta 6 llevó la siguiente introducción:

> Si está respondiendo el presente estudio, muy probablemente tiene conciencia de las necesidades de seguridad. Ahora bien, las necesidades percibidas probablemente serán distintas dependiendo del por qué de su interés en el tema. Indique si siente pertenencia con alguno

[3] Recordemos que las preguntas 5 y 6 permiten múltiples respuestas. Esto es, para cada inciso, debería leerse, *de los 97 encuestados, x se dedican a...*

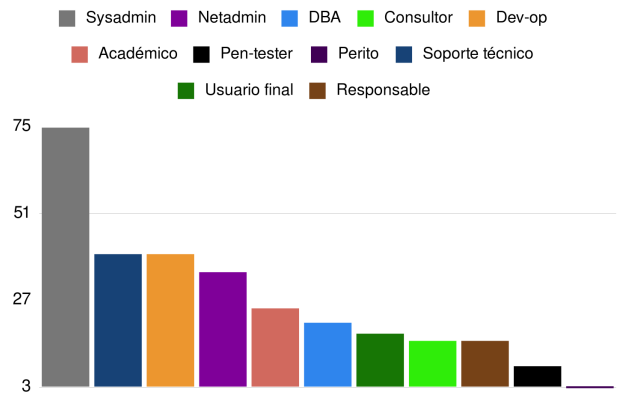


Figura 4.4: Resumen de las respuestas a la pregunta 5, «¿Cuál o cuáles de las siguientes opciones describen mejor su área de especialización laboral?»

(o varios) de los grupos presentados a continuación, o a alguno no contemplado.

Para esta pregunta se siguió la misma lógica que con la pregunta 5, permitiendo más de una respuesta por encuestado.

Una amplísima mayoría (87.63 %) de los encuestados indican que, independientemente del trabajo que realizan, al ser especialistas en cómputo es necesario tener conciencia de las implicaciones de seguridad en los sistemas diseñados, en las arquitecturas de red, en las acciones recomendadas a los usuarios.

Quedan en lejano segundo lugar, y conformando una amplia meseta con apenas variación, quienes tienen a la seguridad informática como trabajo pago, para quienes es un hobby, los académicos y los activistas de grupos sociales (todos ellos con 27 a 30). Con 17, los encuestados que trabajan para la seguridad informática del Estado quedan bastante por detrás, seguidos sólo por los cinco que indicaron *Otro*.

Los encuestados que detallaron su actividad como *Otro* manifestaron en el campo de captura correspondiente:

1. Atención a incidentes de seguridad.
2. La seguridad de la informacion y la comunicacion va de la mano en cualquier aspecto de la vida.
3. Es interesante.
4. Apoyo para la obtención de información y documentación científico-técnica a usuario final.

Cuadro 4.5: Respuestas a la pregunta 6, «Indique si siente pertenencia con alguno (o varios) de los grupos presentados a continuación, o a alguno no contemplado.»

	Cantidad	Porcentaje
Mi trabajo es relacionado con el cómputo, la conciencia en la seguridad va de la mano	85	87.63 %
La seguridad informática es mi trabajo pago	30	30.93 %
Es mi hobby	28	28.87 %
Soy docente, investigador, académico	27	27.84 %
Soy activista de un grupo social	27	27.84 %
Participo en las labores de seguridad informática del Estado (cualquier nivel)	17	17.53 %
Otro	5	5.15 %

5. La seguridad o privacidad personal son muy importantes para mi.

Cada participante tendrá, claro está, su propia motivación para responder como lo hizo; el autor, sin embargo, sugeriría agregar a estas cinco participaciones como:

1. Trabajo pago.
2. Conciencia de la mano.
3. Hobby.
4. Académico.
5. Activista.

4.4. Resumen del capítulo

La encuesta presentada en este capítulo fue realizada para dar una primera validación a las ideas desarrolladas en el proyecto y para "sentir el pulso" de una comunidad de expertos respecto a la temática abordada.

La primera y segunda preguntas de la encuesta fueron consideradas al desarrollar algunos de los puntos del capítulo 5, enfatizando ante la importancia que revisten la autenticación y la resistencia a denegaciones de servicio.

La tercera pregunta mide la importancia de los temas que motivaron al desarrollo de este trabajo; si bien no tienen un impacto definitorio en el desarrollo, sí se alínean con la justificación presentada en la sección 1.5.

La cuarta pregunta busca anticiparse a una posible de las críticas que probablemente reciba el presente trabajo: Entre las referencias citadas en estas páginas hay muchas que no provienen de publicaciones académicas acreditadas.

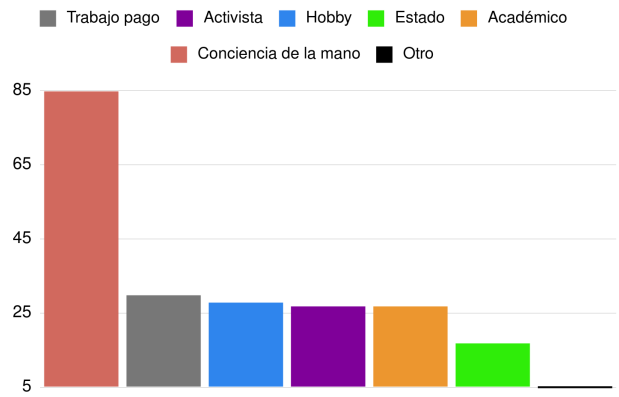


Figura 4.5: Respuestas a la pregunta 6, «Indique si siente pertenencia con alguno (o varios) de los grupos presentados a continuación, o a alguno no contemplado.»

Esta pregunta pretende demostrar cuáles son los usos en la comunidad de expertos en seguridad informática; buena parte de las técnicas y mecanismos tanto de ataque como de defensa son presentados de forma informal y sin el fundamento que en la academia sería considerado como validado por pares. Sin que esto signifique que el presente trabajo dejó de guiarse por publicaciones académicas formales, se incluye para validar las muchas fuentes no formales empleadas.

Por último, los datos demográficos solicitados en la quinta y sexta preguntas se encaminan a conocer el perfil de quienes respondieron.

Cabe mencionar, como inescapable realidad, que tras un análisis muy posterior a la aplicación de la encuesta, el sustentante reconoce que por múltiples factores (incluyendo el sesgo derivado del espacio muestral de aplicación de esta encuesta, un diseño del instrumento no suficientemente planificado, y un desconocimiento personal de la metodología de la investigación en lo relativo a la obtención de información a partir de encuestas anónimas), la validez y objetividad de los datos aquí presentados pueden estar comprometidas (Hernández Sampieri, Fernández Collado y Baptista Lucio 2006, págs. 277-292), y esta carencia aparece como un punto a abordar en el trabajo futuro (véase la sección 7.2). La confiabilidad fue medida *a posteriori* por el método de mitades partidas (*split-halves*), arrojando una correlación global del 95 %, con una correlación mínima en preguntas específicas del 83 %, por lo cual la confiabilidad puede asumirse como elevada.

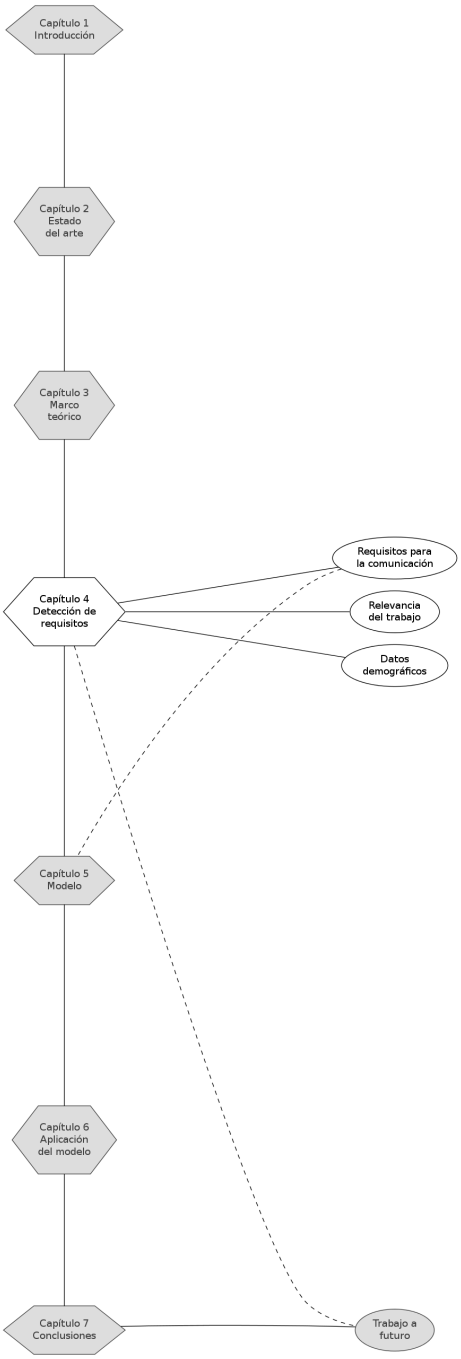


Figura 4.6: Relaciones conceptuales: Temas abordados en el capítulo 4

Capítulo 5

Integración del modelo

En el transcurso del presente capítulo se abordarán los siguientes puntos que definen la interacción modelada; en este punto se presentan los puntos que se desarrollarán, así como la sección en que se abordan:

1. Naturaleza del canal (5.3)
    a) Canal visible sobre el cual viaja
    b) Técnica de ocultamiento
        1) Fortaleza del ocultamiento
        2) Vector para el ocultamiento
    c) Mecanismo de corrección de errores
        1) Resistencia al ruido
    d) Dúplex
    e) Longevidad
    f) Capacidad
2. Establecimiento y codificación (5.4)
    a) Inicialización
        1) Acuse de establecimiento
        2) Autenticación
    b) Codificación y envío de datos
    c) Costo computacional
3. Reconocimiento y decodificación (5.5)
    a) Identificación de mensaje oculto
        1) Evento *disparador*
        2) Autenticación

b) Decodificación del mensaje

1) Codificación de respuestas

c) Costo computacional

El modelo que a continuación se presenta pretende ser general; un canal oculto puede no implementar alguno o varios de los incisos aquí descritos.

5.1. Caracterización básica: Sistema de comunicaciones

Partiendo de que el canal oculto es un caso específico de un sistema de comunicaciones, y si el modelo presentado busca caracterizar una serie de transformaciones para viajar sobre de uno diferente, debe comenzar construyendo sobre los elementos fundamentales descritos por Shannon (1948), obra abordada en la sección 3.1. Se presenta a continuación la adecuación de sus definiciones al campo específico de implementación:

Fuente de información La persona o proceso generador de la información a ser transmitida. Debe estar en la misma red confiable (o en el mismo equipo) que el *Emisor*.

Mensaje Información que debe ser transmitida del *emisor* al *destino* sobre un canal oculto.

Emisor Entidad que *codifica* al mensaje empleando las transformaciones seleccionadas para ocultar al *mensaje* en la *señal* transmitida.

Señal Flujo de bits aparentemente *inocuos* que transitarán por redes públicas. Esta señal lleva codificado dentro de sí al *Mensaje*.

Fuente de ruido Modificaciones a la *señal* producidas por un medio no controlado. Se elaborará al respecto en la sección 5.1.1.

Señal recibida Flujo de bits recibido por el *Receptor*. lleva codificado dentro de sí al *Mensaje*.

Receptor Entidad que *decodifica* la *Señal recibida* para obtener de ella al *Mensaje* privado. Su elección debe considerar la detección, corrección o evasión del ruido, acorde a lo abordado en la seccción 5.1.1.

Destino La persona o proceso destinatario del mensaje. Debe estar en la misma red confiable (o en el mismo equipo) que el *Receptor*.

Como se verá en breve, el modelo propuesto se inserta en el marco general de comunicación aquí descrito en el *emisor* y el *receptor*.

5.1.1. Canales ruidosos y libres de ruido: precisiones

En la literatura específica pueden encontrarse dos acepciones de *ruido* muy distintas, y es preciso encontrar un punto en común para quedar libres de ambigüedad.

El modelo de Shannon (1948) fue presentado en un tiempo en que la transmisión de información era sobre canales analógicos,[1] y fue hecha de forma que permite su aplicación a muy distintos medios físicos. El modelo asume que el canal descrito susceptible a ruido. Al aplicar dicho modelo a una red de datos, el primer gran cambio es que se trata necesariamente de señales digitales; la naturaleza del ruido no desaparece — Pero cambia.

En la realidad en la que se desarrolla el presente trabajo, asumiendo ya una red TCP/IP (como lo es Internet) el efecto del ruido en alguno de los medios sobre los cuales viaja la información se manifiesta en:

1. Entrega de paquetes fuera de órden
2. Corrupción de datos
3. Pérdida de paquetes
4. Demora en la entrega de paquetes
5. Fallo en el establecimiento de sesión

Los primeros tres se presentarán más al emplear como transporte los protocolos orientados a *datagramas*, como UDP, ICMP o directamente IP,[2] como mencionan (Bo, Jia-zhen y De-Yun 2007; Kundur y Ahsan 2003; Murdoch y Lewis 2005), mientras que los últimos dos afectarán a aquellos orientados a *conexión* o *circuitos virtuales*, como TCP o SCTP.

Sin embargo, es necesario considerar particularidades que afectan a los canales ocultos, que resultarían obviables en canales visibles. Dado que el canal oculto aprovecha los espacios aptos para la *redundancia*, el ruido puede, más que distorsionar u ocultar porciones del mensaje, destruirlo por completo — Y sin siquiera ser un mecanismo dedicado a tal fin, como se detallará al comparar diversos canales ocultos en el capítulo 6.

Cabuk 2006, pág. 19 hace referencia a una definición muy distinta de ruido. Citando (traducción propia),

> Si un recurso compartido no es únicamente empleado por los participantes del canal oculto sino por otros usuarios legítimos, el canal resultante es un canal de comunicación ruidoso. Específicamente, un *canal ruidoso* es un canal donde se observan tráfico tanto normal

[1] El modelo de Shannon es perfectamente aplicable a la transmisión digital; numerosos mecanismos netamente digitales existían ya cuando esta fue publicada.

[2] Un mecanismo basado en *port knocking*, a pesar de poder dirigirse a puertos TCP, emplea como canal únicamente al protocolo IP: No existe una sesión TCP aún sobre la cual se transmita información, sólo hay un intercambio de paquetes iniciales. Es por esto que el *port knocking* es vulnerable al desordenamiento de los paquetes.

> como oculto. En contraste, en un *canal oculto sin ruido* el recurso compartido es utilizado exclusivamente por los participantes en la comunicación oculta. El ruido en el canal aquí descrito no es *ruido en la señal* sino que *ruido de contención* causado por la competencia por un recurso compartido.

Cabe recordar que el trabajo de Cabuk fue citado en la sección 3.2, y enfoca su estudio de los canales ocultos en una visión *adversarial*, describiéndolos exclusivamente como agentes que buscan romper una política de aislamiento de datos. Explora, sí, el uso de diversos protocolos que podrían emplearse para comunicación como la que este trabajo modela, pero caracterizándola siempre como el puerto de control para *zombies* en ataques DDoS o para el manejo de puertas traseras.

La manera de hermanar ambas definiciones de *ruido* resulta el determinar que puede considerarse como *ruido* tanto el efecto de variaciones en los medios físicos que la comunicación requiera como los efectos –accidentales o intencionales– de terceros que empleen al mismo canal; esto sin perder en cuenta que la red es en prácticamente la totalidad de los casos un recurso compartido, pero dado que la capa significativa para la transmisión del canal oculto puede estar a diferentes niveles, esto no implica necesariamente que el canal sea ruidoso: Dependiendo la capa de red sobre la cual se cree el canal, éste puede resultar libre de ruido a pesar de que lo haya en capas superiores o inferiores.

5.2. Ámbito de aplicación del modelo

Es necesario mantener en mente el ámbito de desarrollo dentro del que fue desarrollado el proyecto: Una tesina de *especialización*, un proyecto necesariamente corto. Esto define inescapablemente la profundidad del alcance del modelo; resulta claro que hay mucho trabajo con el que podría continuarse desarrollando lo aquí presentado, y algunas direcciones posibles son planteadas en la sección 7.2.

El modelo que a continuación se propone es meramente *descriptivo*. Si bien sería importante y conveniente presentar métricas cuantitativas que permitan comparar objetivamente a los diversos esquemas, así como un conjunto de ponderaciones facilitando la localización del mejor entre varios esquemas de canal oculto para las circunstancias particulares del escenario específico, esto claramente excede el ámbito del desarrollo actual.

La aplicación del modelo lleva a una *comprensión general* del funcionamiento de cada uno de los canales que se estudien, y permite hacer comparaciones *cualitativas* entre ellos. En el capítulo 6 se presentarán dos enfoques posibles: el desarrollo a detalle de cada uno de los esquemas de forma consecutiva, y la presentación en tablas comparativas.

5.3. Naturaleza del canal

Los siguientes ejes primarios definen la naturaleza del canal, esto es, definen qué tipo de comunicación es posible realizar sobre del medio establecido:

Canal visible sobre el cual viaja Un canal oculto se define empleando interacciones legales sobre un canal visible. Este punto describe las características *visibles* del canal sobre del cual éste se instalará.

Técnica de ocultamiento El punto medular de la definición de un canal oculto: ¿qué aspecto del canal visible se emplea para establecer este canal?

Es previsible que en este punto se encuentre una gran inventiva que imposibilite la delimitación a una lista preacordada de técnicas; siguiendo lo delineado por Kahn (1967), el ocultamiento puede realizarse por seguridad en el tráfico o por esteganografía. El ocultamiento en el tráfico puede clasificarse siguiendo las categorías delineadas por (Wang y Lee 2005): canal espacial basado en valores, canal espacial basado en transiciones, canal temporal basado en valores, canal temporal basado en transiciones.

Fortaleza del ocultamiento El planteamiento de un canal oculto ocurre dentro de un contexto técnico determinado, y respondiendo a un dado modelo de amenaza. Siempre que sea posible, conviene evaluar qué tan resistente es el ocultamiento ante diferentes amenazas: Bajo qué supuestos sirve, y cuándo deja de hacerlo.

Vector para el ocultamiento Este punto detalla el espacio del mensaje que será empleado para su codificación. Puede emplearse para esto la clasificación de comunicación presentada en la sección 3.1.1.

Mecanismo de corrección de errores En un canal oculto, que por su naturaleza puede sufrir de las interferencias descritas en la sección 5.1.1, puede ser deseable emplear una codificación que permita la detección y corrección de errores. ¿Qué mecanismo de detección o corrección incorpora el canal, si es que lo tiene, y cómo reacciona el canal en caso de detectar un error superior a la capacidad de corrección?

Resistencia al ruido Dada la importancia que reviste a la resistencia ante ataques de denegación de servicio (véase la sección 4.1), conviene evaluar en este punto qué tan susceptible resulta el canal oculto ante los estos, lo cual, como se abordó en la sección 5.1.1, constituye parte importante de lo que puede medirse como el *ruido* al que estará sujeto el canal.

Dúplex La naturaleza del canal puede permitir únicamente que la información fluya en un sentido (*unidireccional* o *símplex*) o en ambos (*bidireccional* o *dúplex*); en caso de que el canal oculto se defina en torno a más de dos participantes, puede también ser *multidireccional*.

En el caso de un canal bidireccional, el mecanismo empleado para el flujo de la información en un sentido puede ser el mismo (*simétrico*) o distinto (*asimétrico*) al del sentido opuesto.

Un canal unidireccional tiene los papeles de *emisor* y *receptor* claramente definidos; en el caso de un canal bidireccional, denominaremos *emisor* al iniciador de la comunicación y *receptor* al destinatario, aún si mensajes individuales transitan en sentido contrario.

Longevidad Un canal oculto puede establecerse para la transmisión de un comando específico (una pieza mínima de información útil), o mantenerse establecido sobre una suerte de sesión, de modo que puede hablarse de canales orientados, para el primer caso, a la *señalización* o, para el segundo, al *establecimiento de sesión*. Este inciso está necesariamente relacionado muy de cerca con el siguiente.

Capacidad Un canal puede resultar o no adecuado para determinado fin dependiendo de su capacidad. Hay distintas maneras en que puede expresarse la capacidad de un canal oculto:

- Ancho de banda, si es un canal *de establecimiento de sesión*. Detallar también qué tan dependiente es este de actividad visible o de terceros; (Wang y Lee 2005) detalla acerca de la estimación de la capacidad de estos canales.
- Tamaño o complejidad del mensaje, si es un canal de *señalización*.
- En caso de un canal *bidireccional asimétrico*, la medida de capacidad será distinta para cada dirección.

Un canal oculto, sea de *señalización* o de *establecimiento de sesión*, debería ofrecer por lo menos un espacio suficiente para transmitir la información de autenticación, el comando a ejecutar, y probablemente los argumentos. Esto es, el modelo impone una cota mínima en el eje de la *capacidad*. Retomando la idea del *triángulo mágico* abordado en la sección 3.3 (Sehgal y Goel 2014) , el mínimo en los dos ejes restantes (imperceptibilidad y robustez) dependerá del modelo de interacción y la complejidad de la amenaza esperadas por cada implementación.

5.4. Establecimiento y codificación

Esta sección aborda las acciones típicamente realizadas por el *emisor* para el establecimiento y empleo del canal: El establecimiento del mismo y la codificación de la información. Cabe mencionar que cuando el canal es bidireccional, corresponderá también al *receptor* codificar mensajes para su transmisión.

Dentro de este apartado encontramos:

Inicialización El mecanismo por medio del cual el *emisor* notifica al *receptor* que va a iniciar una comunicación sobre canal oculto.

Acuse de establecimiento Al establecer un canal oculto (o como parte de la transmisión de un mensaje, si el canal es de *señalización*), es deseable que el *emisor* reciba confirmación de que el *receptor* reconoció el mensaje. Claro está, no siempre es posible dadas las restricciones del canal. Debe indicarse si el canal maneja un acuse explícito o no, y si lo hace *en banda* o *fuera de banda* (por ejemplo, modificando un valor aparentemente no relacionado, o realizando alguna tarea de algún modo monitoreable.

Autenticación Una vez que el *receptor* reconoce el patrón que lleva al *establecimiento* de un canal, el paso inmediato siguiente debe ser la *autenticación*, siguiendo los parámetros delineados en la sección 3.4.[3]

Buena parte de los canales ocultos descritos en la literatura consultada abordan al canal oculto sin considerar la autenticación o la implementación de un mecanismo criptográfico sobre de éste. En dado caso, este punto puede indicarse como no implementado, pero dadas tanto las consideraciones presentadas en la justificación del presente trabajo como los resultados de la encuesta que se abordan en la sección 4.1, este punto resulta de gran importancia para los usos esperados.

Atendiendo a lo abordado en la sección 3.4, es esperable que la autenticación implique una identificación criptográfica; si el canal permite el acuerdo y renegociación de llaves, debe especificarse.

Codificación y envío de datos ¿Cómo se lleva a cabo la *técnica de ocultamiento* descrita en la sección 5.3? Esto incluye:

- Cómo se realiza la codificación necesaria.
- Costo computacional de la codificación.
- Requisitos particulares para su transmisión (por ejemplo, para ambos tipos de *canal temporal*, resolución temporal requerida).

Costo computacional La factibilidad de un canal oculto para ciertas situaciones puede depender del costo computacional de su implementación. ¿Qué tan complejo es para el emisor realizar las transformaciones al *canal visible* para que transmita al mensaje oculto? ¿Cómo debe transformarse el *mensaje oculto* para ser codificado?

5.5. Reconocimiento y decodificación

Aquí se abordan las acciones típicamente (si bien no exclusivamente) realizadas por el *receptor*:

[3] Cabe mencionar que en implementaciones como (Rash 2007) el establecimiento del canal y la autenticación se realizan de forma simultánea. El proceso de reconocimiento y autenticación, a pesar de viajar en el mismo paquete de red, puede verse como compuesto por esta secuencia de acciones.

Identificación de mensaje oculto El *receptor* debe poder identificar un intento de establecer un canal oculto dentro del tráfico normal de red (de forma, claro está, que no resulte obvio para un tercero). Este mecanismo puede dividirse en dos partes, correspondientes a lo descrito en la sección 5.4:

Evento disparador El primer paso para que el *receptor* establezca un canal es la identificación de un evento disparador, la notificación por parte del *emisor* de que intenta establecer un canal oculto.

La identificación de dicho disparador puede o no estar seguida de un *acuse* enviado de vuelta al *emisor*, dependiendo de las características del medio y del canal.

Puede ser importante considerar en este punto el costo computacional, no sólo del reconocimiento de un evento disparador, sino que también de la sobrecarga que implica el procesamiento de las comunicaciones en claro, hasta descartar la presencia de un evento disparador de este canal oculto.

Autenticación El *receptor* debe asegurarse de que el *emisor* es efectivamente un usuario autorizado para el establecimiento del canal oculto administrativo en cuestión; para no reiterar en la explicación, se remite al lector al inciso correspondiente de la sección anterior.

En caso de ser un canal *bidireccional*, cabe mencionar si se brinda al *emisor* un acuse de autenticación exitosa.

Cabe recalcar que, si bien varios de los canales analizados no contemplan siquiera a un mecanismo de autenticación (probablemente descargando esa tarea a capas superiores), este tema resultó de principal importancia entre las respuestas a la encuesta presentada en la sección 4.1.

Decodificación del mensaje Teniendo ya a un canal establecido y a un usuario autenticado, el siguiente paso es el medular de este canal, la razón de su establecimiento: La recepción y decodificación del mensaje. En los casos de canales de *señalización*, el evento disparador podría incluir la totalidad de la comunicación, pero en el caso de un canal de *establecimiento de sesión*, la decodificación se mantendrá activa hasta la finalización del canal.

Codificación de respuestas En el caso de tratarse de un canal *bidireccional*, las respuestas deben ser codificadas y transmitidas al *emisor*. Esto puede ser realizado por el mismo mecanismo que el ya descrito o por un mecanismo diferente; en caso de ser diferente, se sugiere tratar al canal de vuelta como un segundo canal oculto, y desarrollarlo de forma independiente.

Costo computacional El *receptor* también deberá realizar un trabajo computacional para detectar y descifrar a un mensaje oculto. ¿Cuánto se penalizará a su rendimiento por estar constantemente a la espera de un mensaje

oculto? ¿Qué tan complejo resultará descifrar el mensaje oculto una vez detectado?

A diferencia del criterio presentado en la sección anterior, en caso de que la definición del canal evaluado incluya el paso de la identificación de un mensaje oculto, deberá considerarse el procesamiento adicional que el *receptor* deba hacer en todos los mensajes que *no incluyan* mensajes ocultos para detectar su presencia.

5.6. Cuantificación de valores

La mayor parte de los ejes que describe el modelo propuesto son cualitativos. Sin embargo, para poder presentar un reporte comparativo (véase la sección 5.7), algunos de ellos presentarán un criterio básico de cuantificación. Esto debe tomarse sin menoscabo de lo expuesto en la sección inmediata anterior; el modelo es descriptivo y los criterios sugeridos para la cuantificación no están formalmente validados. Este aspecto se menciona, pues, como trabajo pendiente en la sección 7.2.

En caso de manejarse valores discretos con mejoría cualitativa (por ejemplo, *dúplex*: Unidireccional, bidireccional, multidireccional), a cada uno de ellos se asigna un valor determinado en el rango de cero a tres.[4]

En caso de tratarse de un eje que represente un continuo, la cuantificación puede realizarse a una escala con cuatro valores (0, 1, 2, 3), correspondiente a etiquetas descriptivas (típicamente *Nulo*, *Bajo*, *Mediano*, *Alto*), y representarse numéricamente en el intervalo mencionado.

Si se describe un atributo que no puede presentarse en escala, sino meramente de forma descriptiva (como el *canal visible sobre el cual viaja*), se excluye de la cuantificación y se maneja únicamente como dato nominativo. La sección 5.7 detalla cómo corresponde evaluar a cada uno de los ejes del modelo.

Las cuantificaciones sugeridas son:

Fortaleza del ocultamiento Puede discretizarse hacia la siguiente escala:

Nula (0) La comunicación es abierta y en claro.

Baja (1) El canal puede ser detectado por un análisis casual y no dirigido.

Algunos ejemplos podrían ser el empleo de canales *laterales* que resultan obvios ante un análisis trivial (por ejemplo, variando el contenido de los campos *reservados* de un protocolo binario), la generación de patrones de tráfico demasiado repetitivos y delatores, o la apertura explícita de canales de comunicación cifrada, donde el contenido se oculta, pero la intencionalidad de la comunicación entre los actores involucrados no.

[4] De ser necesario, podrían ser valores fraccionales.

Mediana (2) El canal no disparará alarmas derivadas de desviaciones del comportamiento normal, pero puede ser develado tras un análisis enfocado; está codificado en patrones similares a los que se presentarían en la operación normal del sistema.

Algunos ejemplos incluyen al *abuso* del tráfico aparentemente normal en la red llegando a umbrales de repetición notables o el uso válido pero con anomalías detectables de recursos del sistema.

Alta (3) La presencia de comunicación oculta no únicamente se basa en el desconocimiento de su existencia, sino que resulta indetectable incluso ante su búsqueda explícita.

Algunos puntos que podrían apuntar a un canal con fuerza de ocultamiento alta pueden ser: evita la comunicación directa entre *emisor* y *receptor*; el mensaje oculto se transmite cifrado o procesado de forma que lleva a una distribución no detectable estadísticamente; emplea algoritmos de cifrado considerados como estado del arte.

Corrección de errores y resistencia al ruido Se presentan como un campo único, con los siguientes valores:

Nula (0) El canal no contempla corrección de errores alguna; la presencia de ruido fácilmente lleva a la pérdida o modificación del mensaje.

Baja (1) El canal no contempla corrección de errores explícitamente, pero viaja sobre un transporte que hace poco probable que el ruido conlleve pérdida o modificación de datos.

Mediana (2) Contempla corrección de errores ante la modificación del mensaje, pero es susceptible a su pérdida.

Alta (3) Especifica mecanismos robustos para asegurar la entrega e integridad de los mensajes.

Dúplex Siendo ya tres valores de inicio, su representación es directa:

Unidireccional (1) Este canal permite únicamente el flujo de información del emisor hacia el receptor.

Bidireccional (2) Este canal contempla explícitamente la posibilidad de respuesta del receptor hacia el emisor.

Multidireccional (3) El canal gestiona la comunicación oculta no únicamente entre dos entidades, sino que entre varias.

Capacidad Es particularmente dificil brindar una guía clara para este inciso, dadas las diferencias entre los diferentes canales reseñados (y buscando la generalidad necesaria para este trabajo). Se sugiere la siguiente escala:

Mínima (0) Para establecimiento de sesión, aquellos canales que puedan aprovechar menos de $\frac{1}{100}$ de la capacidad del canal visible independientemente de la demora en que incurran por el tráfico normal generado por terceros.

Para señalización, si el canal permite enviar un único mensaje.

Baja (1) Para establecimiento de sesión, aquellos canales que puedan aprovechar entre $\frac{1}{10}$ y $\frac{1}{100}$ de la capacidad del canal visible independientemente de la demora generada por el tráfico normal entre terceros, o aquellos canales que puedan aprovechar hasta $\frac{1}{100}$ del ancho de banda, pero que no sean susceptibles a demora por tráfico.

Para señalización, si el canal permite enviar entre dos y diez mensajes distintos.

Mediana (2) Para establecimiento de sesión, aquellos canales que, sin sufrir demoras inducidas por terceros actores más allá de los parámetros normales en una red, puedan aprovechar más de $\frac{1}{10}$ de la capacidad del canal visible sobre el cual viaja independientemente de la demora generada por el tráfico normal entre terceros, o aquellos canales que puedan aprovechar entre $\frac{1}{10}$ y $\frac{1}{100}$ del ancho de banda, pero que no sean susceptibles a demora por tráfico.

Para señalización, si el canal permite enviar mensajes arbitrarios, aunque de tamaño limitado.

Alta (3) Para establecimiento de sesión, sólo aquellos que no sean susceptibles a la demora por la espera inducida por terceros actores más allá de los parámetros normales en una red, y que puedan aprovechar por lo menos $\frac{1}{10}$ de la capacidad de comunicación en claro.

Para señalización, si el canal permite enviar mensajes arbitrarios, sin límite de tamaño.

Acuse de establecimiento (Establecimiento y codificación) Dado que al presentarse como reporte cuantificado la información resultaría redundante, este campo y el de *identificación de mensaje oculto* (de *Reconocimiento y decodificación*) se presentan como un campo único.

Nulo (0) No especifica ningún mecanismo de inicialización, describe únicamente la interacción sobre un canal ya establecido, o se asume que el canal siempre estará listo.

Bajo (1) Especifica un mecanismo de notificación de inicio de canal, que será esperado por el *receptor*, pero no contempla ninguna manera de entregar un acuse de establecimiento: El canal podría fallar silenciosamente.

Alto (2) Define un mecanismo de inicialización con acuse de establecimiento.

Autenticación (Establecimiento y codificación) La autenticación en el momento de establecimiento y codificación (esto es, del emisor ante el receptor).

Nulo (0) No contempla autenticación.

Bajo (1) El establecimiento del canal incluye un paso de autenticación, aunque es muy débil acorde a lo abordado en la sección 3.4.

Alto (2) Contempla un mecanismo confiable (fuerte) de autenticación.

Costo computacional (Establecimiento y codificación) Cabe recalcar que la escala en el caso del costo computacional va *en órden inverso* de como se maneja con los demás incisos. Esto se explica al ver qué valores resultan más deseables: A menor costo mejor (mientras que, por ejemplo, a mayor fortaleza de ocultamiento, mejor).

Alto (0) El mensaje debe someterse a más de una capa de cifrado, o a alguna otra transformación de complejidad similar o superior; la transmisión del mensaje requiere de reiteradas llamadas al sistema operativo.

Como ejemplo, los esquemas de anonimato criptográfico (firmas ciegas o en anillo) o el uso de árboles de parseo.

Mediano (1) El mensaje se cifra con un mecanismo de llave pública, o es autenticado por *hashes* empleados de forma apta contra la repetición; la transmisión del mensaje requiere la generación de estructuras de bajo nivel.

Como ejemplos, un mensaje que sea cifrado bajo esquemas DSA o RSA, o cuya verificación sea basada en HMAC.

Bajo (2) El mensaje debe sufrir una recodificación trivial; su transmisión emplea mecanismos directos y estándar.

Como ejemplos, mensajes enviados con una codificación directa y procesada caracter a caracter, como Base64 o uudecode, o cuyos caracteres son insertados u ocultados en campos o posiciones específicas de otros flujos de datos; mensajes procesados empleando códigos simples de detección y corrección de errores.

Nulo (3) Se entrega el mensaje al sistema operativo tal como se recibió del emisor.

Autenticación (Reconocimiento y decodificación) La verificación correspondiente que asegura estar hablando con el equipo adecuado y no con un impostor (esto es, del receptor ante el emisor).

Nulo (0) El establecimiento de canal no contempla autenticación.

Bajo (1) El canal contempla autenticación únicamente en un sentido, pero el *emisor* no autentica la identidad del *receptor*.

Mediano (2) El *emisor* autentica la identidad del *receptor*, aunque por mecanismos débiles acorde a lo abordado en la sección 3.4.

Alto (3) El *emisor* autentica al *receptor* por medio de un mecanismo confiable (fuerte) de autenticación.

Costo computacional (Reconocimiento y decodificación) Corresponde la misma nota que en el costo computacional de establecimiento y codificación: El órden presentado es el inverso respecto a los demás puntos porque el dato más deseable es también otro.

Alto (0) El mensaje tiene más de una capa de cifrado, o alguna otra transformación de complejidad similar o superior. La carga computacional para reconocer o descartar la presencia de comunicación oculta en el canal resulta considerable.

Mediano (1) El mensaje se cifra con un mecanismo de llave pública, o es autenticado por *hashes* empleados de forma apta contra la repetición; el sistema debe aplicar varias operaciones para detectar si cada mensaje contiene un mensaje oculto.

Bajo (2) El mensaje sufrió una recodificación trivial; su detección –sabiendo del esquema y en su caso las claves necesarias– requiere un esfuerzo mínimo.

Nulo (3) El sistema operativo entrega el mensaje de forma que no requiera procesamiento adicional.

5.7. Construcción del reporte

Este modelo pretende ser útil para asistir en la comparación de cara tanto a la utilización como al desarrollo de canales ocultos; para lograrlo, se sugiere la construcción de los datos resultantes tras la integración de datos de un canal evaluado en un reporte sucinto.

Este reporte está orientado a facilidad de comparación únicamente, por lo queno incluirá la información completa recabada en la aplicación del modelo, limitándose a los siguientes datos:

Datos cuantificables Aquellas variables que denoten un valor que pueda representarse como una magnitud escalar, representando una progresión de menor a mayor o de peor a mejor, se representarán en una *gráfica de radar* (Croarkin 2003-2012, sección 1.3.3.29).

Dada la naturaleza de esta gráfica, los datos más cercanamente relacionados deben presentarse juntos. La gráfica resultante describirá la fortaleza relativa por área; los valores a representar se homologarán sobre una misma escala.

Los ejes que se representan en la gráfica de radar son:

1. Fortaleza del ocultamiento
2. Corrección de datos y resistencia al ruido
3. Dúplex
4. Inicialización y acuse de establecimiento
5. Autenticación (*receptor*)
6. Autenticación (*emisor*)
7. Costo computacional (*receptor*)
8. Costo computacional (*emisor*)

9. Capacidad

   Al leer esta gráfica es indispensable recordar que presentar la información sobre estos nueve diferentes ejes *de ninguna manera implica* que estén a escala; cada dimensión descrita tiene criterios propios, y su importancia relativa variará según el caso para el cual se esté analizando.

Datos nominativos Varios de los campos que comprende el modelo no son cuantificables, sino únicamente nominativos. Esto no significa que carezcan de valor, sino que describen al canal de una forma no numérica. Estos datos se reportarán como campos de texto, facilitando al usuario distibguir cuál se adecúe mejor a sus necesidades. Los datos nominativos son:

- Canal visible sobre el cual viaja
- Técnica y vector de ocultamiento
- Longevidad
- Codificación y envío de datos
- Decodificación del mensaje y codificación de respuestas

5.8. Resumen del capítulo

El presente capítulo constituye la médula del trabajo desarrollado como tesina, y se espera que forme la aportación al desarrollo del campo: Construyendo fundamentalmente sobre lo presentado en los capítulos 3 y 2, y considerando los puntos afirmados en el capítulo 4, este capítulo procede con el desarrollo del modelo.

La primera sección (5.1) hace las precisiones necesarias a considerar en los modelos de comunicación presentados en la sección 3.1 para ser aplicados al modelado de canales ocultos, particularmente al contexto de las redes de datos TCP/IP.

La sección 5.2 enmarca el ámbito esperado de aplicación y alcance del modelo, y apunta algunas ideas para su desarrollo a futuro.

Las secciones restantes del capítulo describen las tres áreas primarias en que está dividido el modelo: La naturaleza del canal (sección 5.3), el establecimiento y codificación (5.4) y el reconocimiento y decodificación (5.5). Claro está, este capítulo es referido reiteradamente a lo largo del capítulo 6.

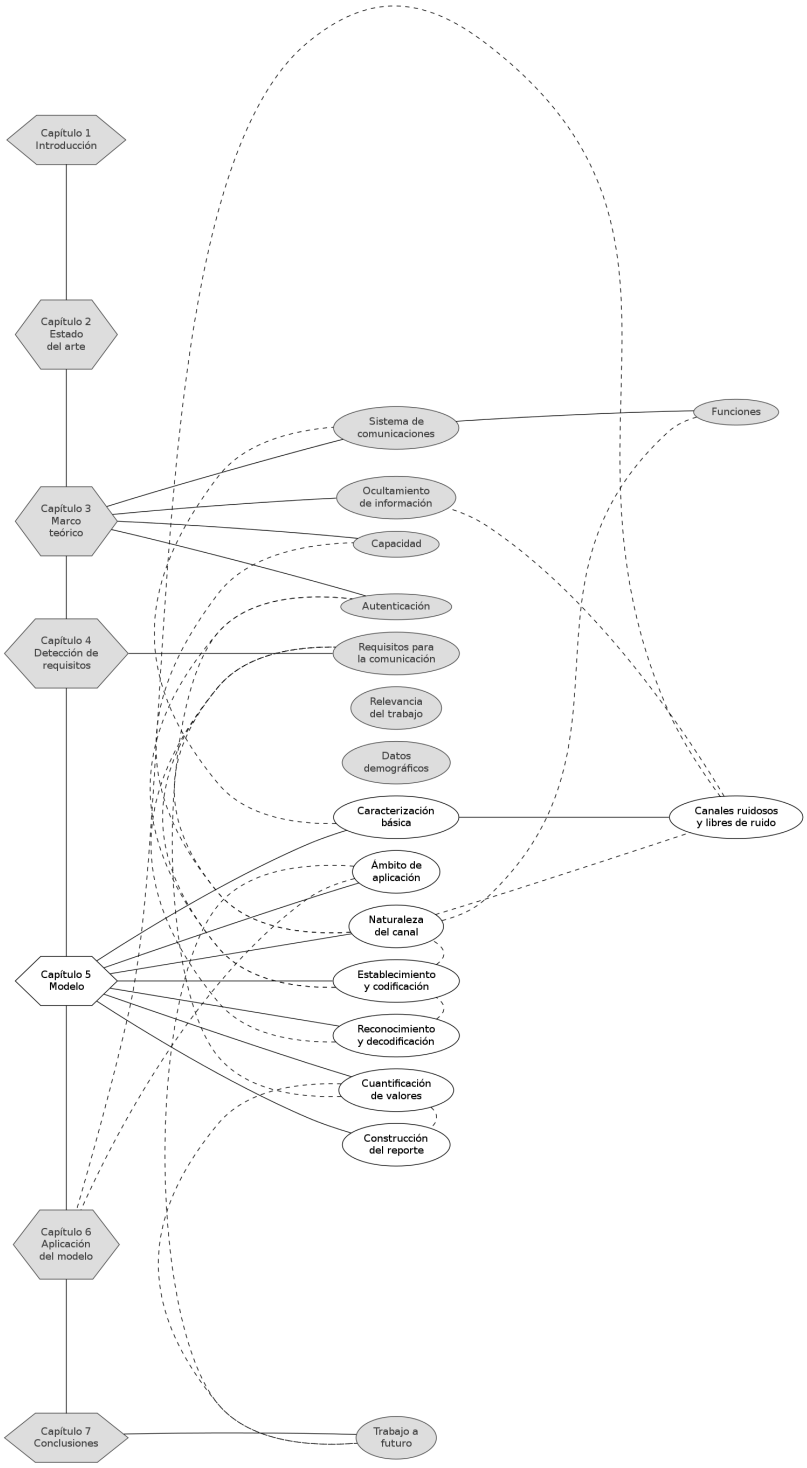


Figura 5.1: Relaciones conceptuales: Temas abordados en el capítulo 5

Capítulo 6

Aplicación del modelo

En este capítulo se presenta, a modo de ejemplo, la aplicación del modelo descrito a varios de los canales propuestos por la literatura citada. En la sección 6.1 se abordan los distintos canales presentados a lo largo de la presente obra, particularmente en las secciones 3.2, 3.3.1, 2.1.1, 2.1.2, 2.2 y 5.1.1.

Además de los canales citados en la literatura, en la sección 6.2 se presenta brevemente una propuesta que se plantea como respuesta a los escenarios planteados en la sección 1.5.1. Los detalles técnicos de esta propuesta se basan en buena medida en las ideas presentadas en la sección 2.3.

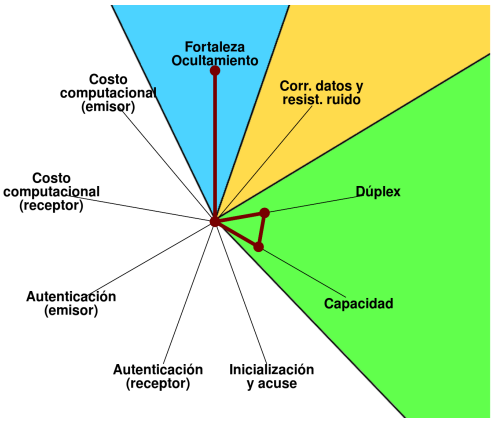
6.1. Modelo aplicado a la revisión bibliográfica

A continuación, cada una de las subsecciones aborda a uno de los canales ocultos abordados en la revisión bibliográfica previa.

Todos los análisis inician con la representación del análisis siguiendo el formato de reporte detallado en la sección 5.7; los datos cuantificables se presentan empleando una *gráfica de radar*, y los datos nominativos se detallan en una tabla. Después del reporte, se presenta un análisis detallado del funcionamiento del canal, detallando punto por punto sus características.

Como dato adicional, en las gráficas de radar se presentan tres áreas sombreadas, correspondientes a los vértices de los triángulos presentados en las gráficas 3.3 y 3.4: En azul, el sector de la *fortaleza del ocultamiento* (correspondiente a *indetectabilidad*), en amarillo, la *corrección de datos y resistencia al ruido* (correspondiente con la *robustez*), y en verde, *dúplex* y *capacidad* (correspondiente a *capacidad*).

6.1.1. El problema del confinamiento



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Bloqueos en acceso a archivos en el sistema operativo	Canal espacial basado en transiciones	Conativo	Sesión	Señalizado por acceso a tres archivos con bloqueo exclusivo	Señalizado por acceso a tres archivos

Como primer ejemplo se analiza la propuesta de (Lampson 1973) (ilustrado en la figura 6.1), que abre la discusión académica acerca de la inevitabilidad de la existencia canales ocultos. A continuación se aborda su ejemplo #5, único de los ejemplos en el artículo acompañado por pseudocódigo ilustrando el funcionamiento.

1. Naturaleza del canal

   Canal visible Señalización de error por parte de un sistema operativo que no permite una doble apertura a determinado archivo.

   Técnica de ocultamiento Canal espacial basado en transiciones.

   Los dos procesos implicados en la comunicación no pueden comunicarse directamente; el canal se implementa con tres archivos, cuyo contenido es irrelevante. El manejo de dichos archivos lleva una semántica similar a la de *mutexes*.

   Fortaleza del ocultamiento Alta.

   El canal se establece en base al uso *externamente observable* de llamadas al sistema operativo. En caso de sospechar un administrador de la existencia de un canal de este tipo (y de permitirlo el sistema operativo en cuestión), podría registrar en bitácora estas llamadas, y revelar al canal por su alta frecuencia en los periodos en que se presente intercambio oculto de información.

Given a procedure open (file, error) which does **goto** error if the file is already open, the following procedures will perform this simulation:

```
procedure settrue (file); begin loop 1: open (file, loop 1) end;
procedure setfalse (file); begin close (file) end;
Boolean procedure value (file); begin value := true;
  open (file, loop 2); value := false; close (file); loop 2: end
```

Using these procedures and three files called data, sendclock, and receiveclock, a service can send a stream of bits to another concurrently running program. Referencing the files as though they were variables of this rather odd kind, then, we can describe the sequence of events for transmitting a single bit:

```
sender:    data := bit being sent; sendclock := true
receiver:  wait for sendclock = true; received bit := data;
             receive clock := true;
sender:    wait for receive clock = true; sendclock := false;
receiver:  wait for sendclock = false; receiveclock := false;
sender:    wait for receiveclock = false;
```

Figura 6.1: Pseudocódigo ilustrando el mecanismo usado para el problema del confinamiento (Lampson 1973)

Esto, sin embargo, llevaría a probables falsos positivos, dado que el patrón de interacción sería muy parecido al de los mecanismos de sincronización que necesariamente emplean los sistemas distribuidos.

Vector para el ocultamiento Conativa.
Solicitar el uso de un recurso con bloqueo exclusivo constituye una llamada (una *órden*) al sistema operativo.

Mecanismo de corrección de errores No contemplado.

Resistencia al ruido Otros procesos en el sistema podrían solicitar alguno de los archivos, ralentizando la comunicación o interfiriendo en su significado.

Dúplex Unidireccional

Longevidad Establecimiento de sesión

Capacidad Ancho de banda limitado por la frecuencia del cambio de contexto del sistema operativo, con diez llamadas al sistema y cuatro cambios de contexto por bit transmitido.

2. Establecimiento y codificación

Inicialización No contemplado, sin embargo, la semántica de bloqueo/espera implica que el emisor que haya iniciado la transmisión esperará al receptor para continuar.

Acuse de establecimiento No contemplado.

Autenticación No contemplado.

Codificación y envío de datos Conversión de los datos a transmitir a un vector de bits; para cada bit a ser enviado debe seguirse el protocolo delineado en la figura 6.1.

Costo computacional Alto.

Según el esquema propuesto, cada bit transmitido requiere de diez llamadas al sistema y cuatro cambios de contexto.

3. Reconocimiento y decodificación

Identificación de mensaje oculto No contemplado, sin embargo, la semántica de bloqueo/espera implica que el receptor que haya entrado a la subrutina de decodificación quedará en espera del emisor.

Evento disparador No contemplado.

Autenticación No contemplado.

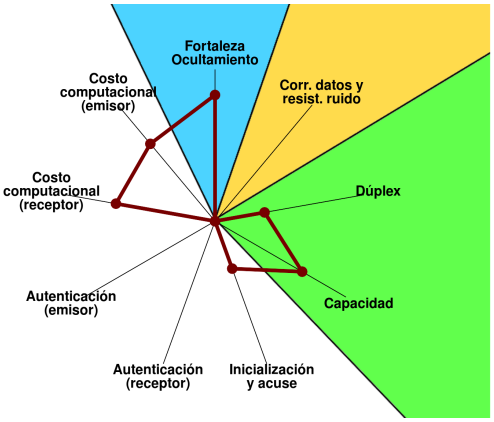
Decodificación del mensaje Recepción de datos bit por bit; para cada bit recibido debe seguirse el protocolo delineado en la figura 6.1.

Costo computacional Alto.

Según el esquema propuesto, cada bit transmitido requiere de diez llamadas al sistema y cuatro cambios de contexto.

Lampson abordó únicamente el cómo se intercambiaría la información entre dos participantes que ya tienen una comunicación establecida y operan en la misma ventana de tiempo. Toda autenticación y acuerdo de inicio de sesión deben ser realizados en otras capas o componentes. El esquema *asume* una sesión, por lo que su *longevidad* es de establecimiento de sesión, pero no aborda expresamente cómo iniciarla y terminarla.

6.1.2. COS sobre IP por almacenamiento



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Conexiones IP arbitrarias entre los participantes	Canal temporal basado en valores	Estética	Sesión	Mensaje paquetizado, codificado con el algoritmo elegido, enviando (1) o no (0) mensajes en el tiempo τ establecido	Recibir un número dado de paquetes, verificar integridad. Se entrega a capa superior.

Cabuk (2006) presenta varios potenciales canales ocultos; analizamos como ejemplo a sólo uno de ellos, el *Canal Oculto Simple sobre IP por almacenamiento*. Su funcionamiento está ilustrado en la figura 6.2.

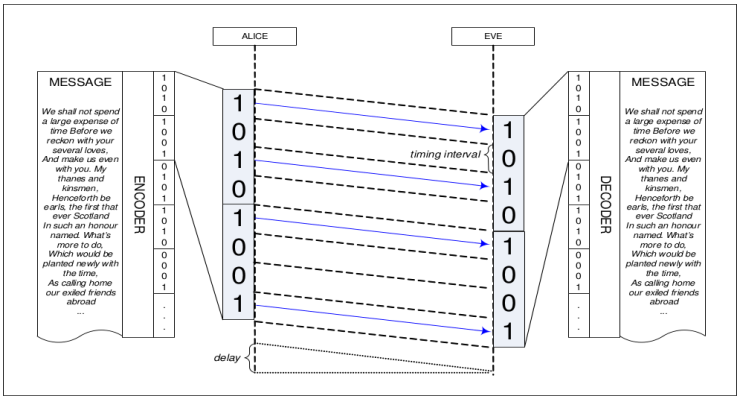


Figura 6.2: Diagrama ilustrando el mecanismo propuesto para un canal oculto simple sobre IP por almacenamiento (Cabuk 2006).

1. Naturaleza del canal

   Canal visible Conexiones arbitrarias sobre IP entre *Alice* y *Bob*

   Técnica de ocultamiento Canal temporal basado en valores.

   Se define un intervalo τ. Para cada bit a ser enviado, *Alice* envía (1) o no envía (0) un paquete de datos arbitrario a *Bob*.[1] *Eve* observa el tráfico en el canal.

   Fortaleza del ocultamiento Mediana / alta.

   La mera transmisión de una gran cantidad de paquetes IP no debe constituir base para la sospecha de tráfico oculto (y mucho menos, como lo sugiere Cabuk, en el caso de que *Bob* sea un servidor de contenido). La naturaleza de consultas podrían llamar más la atención de un *Walter* (por ejemplo, solicitar de

[1] Cabuk sugiere a *Bob* como un servidor de contenido HTTP.

forma reiterada la misma página Web podría delatar la presencia de *algo más*), pero puede elaborarse una serie más *creíble* de solicitudes.
Posiblemente incluso resultaría más adecuado el empleo de un protocolo que se emplee para comunicación interactiva, como un túnel SSL (convirtiendo la comunicación en HTTPS).

Vector para el ocultamiento Estética.
Al ser este un mecanismo temporal (no esteganográfico), el contenido del canal visible resulta irrelevante. El mensaje oculto viaja *modulado* sobre la *forma* de la comunicación.

Mecanismo de corrección de errores Cabuk no especifica qué mecanismo de detección y corrección de errores deba emplearse, pero apunta a que (véase la figura 6.3), siendo este canal susceptible al ruido por las demoras (e incluso pérdidas) de paquetes, debería emplearse uno.

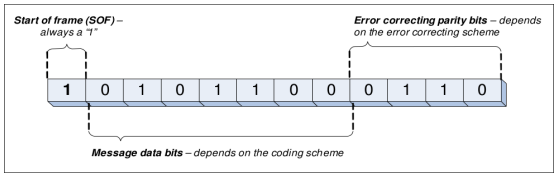


Figura 6.3: Cabuk sugiere el uso de algún mecanismo de detección y corrección de errores, sin detallar cuál deba de ser.

Resistencia al ruido Cualquier evento que cause que el equipo de *Alice* realice solicitudes a *Bob* fuera de las estrictamente controladas introducirá ruido al sistema.
Una alta saturación de la red podría causar que *Eve* pierda de vista algunos paquetes, llevando también a pérdida de información; este punto depende fuertemente de la topología y configuración particular de la red.

Dúplex Unidireccional

Longevidad Establecimiento de sesión

Capacidad Ancho de banda limitado por la predictibilidad de la demora de paquetes enviados por *Alice*. Si *Alice* y *Eve* están en la misma red local, el ancho de banda será superior; entre más *saltos* haya entre ellas, menor será el τ mínimo aceptable.

2. Establecimiento y codificación

Inicialización Un mensaje se divide en cuantos *marcos* (*frames*) sean necesarios. Normalmente no hay tráfico entre *Alice* y *Bob*; un paquete enviado de *Alice* a *Bob* constituye un *inicio de marco* (*Start Of Frame*).

Acuse de establecimiento No contemplado.

Autenticación No contemplado.

Codificación y envío de datos El mensaje es convertido en un vector de bits y codificado con el algoritmo de detección y corrección de errores elegido. Una vez enviado un paquete como *inicio de marco*, durante un intervalo τ *Alice* enviará un paquete de datos a *Bob* para indicar un 1, o no lo enviará para indicar un 0.

Costo computacional Bajo.

La codificación depende únicamente de generar tiempos de espera acorde.

3. Reconocimiento y decodificación

Identificación de mensaje oculto En circunstancias normales (cuando no se está enviando un mensaje oculto), no habrá paquetes de *Alice* a *Bob*. Un paquete constituye el *inicio de marco*, a continuación de lo cual será transmitida una serie de bits.

Evento disparador Un paquete de datos de *Alice* a *Bob*.

Autenticación No contemplado.

Decodificación del mensaje Al completar un marco, *Eve* verifica la integridad del mismo (y de ser necesario, intenta corregirlo) empleando el mecanismo de detección y corrección de errores determinado. Una vez validado el marco, se entrega a la capa o componente de nivel superior.

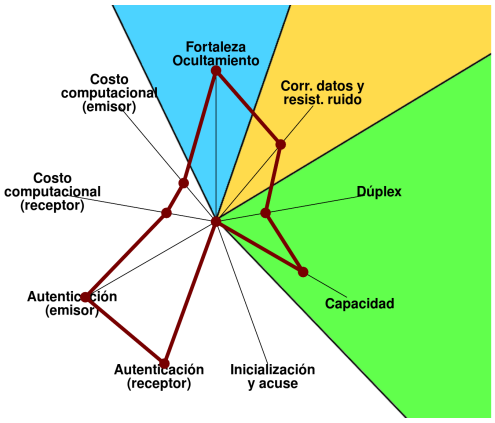
Codificación de respuestas No contemplado.

Costo computacional Bajo.

Toda actividad generada de *Alice* hacia *Bob* indicará a *Eve* que está iniciando una transmisión, y la codificación es trivial de recuperar.

Este mecanismo presenta una implementación más completa, aunque se limita a transportar los datos que serán empleados por alguna otra capa, por lo tanto no contempla ni autenticación ni la interpretación de un mensaje completo (más allá de entregar los bits de un marco como una sola entidad).

6.1.3. El canal subliminal y las firmas digitales



Canal visible	Técnica ocultamiento	Vector ocultamiento	Longitud	Codificación y envío	Decodificación y respuesta
Comunicación abierta y firmada (comprobable por terceros)	Esteganográfica	Fática	Señalización	El firmado del mensaje se altera entre dos *sub-llaves* relacionadas para señalizar 1 o 0.	Conociendo u y con un mensaje recibido, la sub-llave elegida indica un bit.

Esta propuesta parte de un planteamiento completo de ocultamiento de información (Simmons 1985), por lo cual es el primero de los esquemas abordados en cubrir la parte de autenticación. El canal oculto, dentro de la firma criptográfica de un mensaje cubierta (inocente en apariencia), está ilustrado en la figura 6.4.

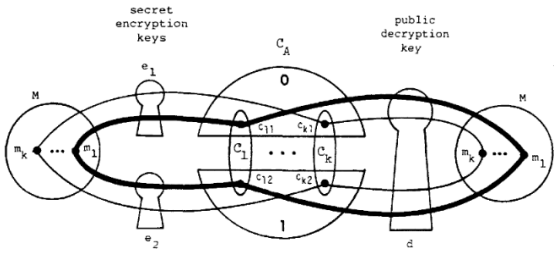


Figura 6.4: Representación de un mecanismo que permite ocultar un bit por mensaje empleando el espacio del firmado (Simmons 1985).

1. Naturaleza del canal

   Canal visible Comunicación abierta de mensajes *cubierta* con firma criptográfica comprobable por terceros.

Técnica de ocultamiento Esteganográfica.

Partiendo de un canal que permita el intercambio de mensajes en claro de m bits, con una firma de r bits, el firmado de los mensajes puede *pervertirse* para reducir la fortaleza del firmado en l bits, y emplear dichos l bits para la creación de un canal oculto.

Fortaleza del ocultamiento Alta.

El modelo de amenaza está basado en que el vigilante *Walter* media y vigila toda la comunicación.

Vector para el ocultamiento Fática.

El canal visible está definido parcialmente por su autenticación (un mensaje sin firmar no será aceptado); la esteganografía se realiza dentro de la comunicación *relativa al canal*.

Mecanismo de corrección de errores No contemplado. expresamente, sin embargo, el mensaje sigue siendo (verificablemente) firmado criptográficamente.

Resistencia al ruido Al estar los mensajes *cubierta* intercambiados firmados por la llave que hace posible la existencia del canal oculto, la integridad de éstos está garantizada: Mientras no tenga la llave privada, es extremadamente poco probable que *Walter* pueda firmar exitosamente un mensaje modificado.

Lo mismo vale a la inversa: Podría verse de cierto modo como que el mensaje *cubierta* garantiza la integridad de la firma (pues si ésta fuera modificada, el mensaje no validaría correctamente). Esto contempla únicamente la detección de errores, no su corrección.

Dúplex Unidireccional

Longevidad Señalización.

El canal oculto no es presentado como orientado a la comunicación en red; cada mensaje es visto como una unidad y es tenido por independiente de todos los demás. En capas superiores podría implementarse un mecanismo que maneje secuencias de mensajes, convirtiéndolo en más adecuado para operar como *establecimiento de sesión*, pero no es parte del planteamiento.

Capacidad El artículo presenta mecanismos que permiten la comunicación oculta de hasta $l = \frac{r}{3}$ bits por mensaje, y menciona en sus conclusiones que hay un mecanismo más complejo que el presentado que permite que *fluya igual cantidad de información por el canal oculto que por el canal abierto* empleando firmas Ong-Schnorr-Shamir. Presenta, sin embargo, como *postscript* que, una semana después de presentarse el trabajo, se presentó un criptoanálisis al criptosistema de Ong-Schnorr-Shamir.

2. Establecimiento y codificación

Inicialización Se delínea el esquema básico presentado por el artículo. Cabe mencionar que, al no estar este esquema formalmente propuesto para una comunicación *en red*, la inicialización es más bien tomada como el paso de generación de llaves, pero no puede verse necesariamente como una inicialización de *cada comunicación*.

Para generar las llaves, el *emisor* calcula un n de factorización computacionalmente infeasible. Elige un u aleatorio tal que $(u, n) = 1$, y calcula $k = u^{-2} (\text{mod } n)$. Publica n y k como su llave de autenticación.

La existencia de u es mantenida como secreta, pero el *receptor* debe conocerla.

Acuse de establecimiento No contemplado.

Autenticación Toda la comunicación fluye sobre el espacio redundante de una firma criptográfica; requiere establecer de antemano la fortaleza relativa que sus usuarios están dispuestos a ceder para la creación del canal.

Codificación y envío de datos Dado un mensaje $m, (m, n) = 1$ a ser firmado, y un mensaje $m*, (m*, n) = 1$ a ser comunicado de forma oculta, el emisor calcula $s_1 = \frac{1}{2}(\frac{m}{m*} + m*) (\text{mod } n)$ y $s_2 = \frac{u}{2}(\frac{m}{m*} - m*) (\text{mod } n)$, y transmite la tripleta (m, s_1, s_2) como mensaje firmado.

Costo computacional Mediano.

El esquema propuesto se construye *abusando* de firmas realizadas bajo un modelo de criptografía de llave pública.

3. Reconocimiento y decodificación

Identificación de mensaje oculto Se asume que todo mensaje firmado con la llave *trucada* lleva un mensaje oculto.

Evento disparador Recepción de un mensaje firmado.

Autenticación Un mensaje (m, s_1, s_2) recibido tal que $a \equiv s_1^2 + k \cdot s_2^2 (\text{mod } n)$ es tenido por auténtico si $a = m$.

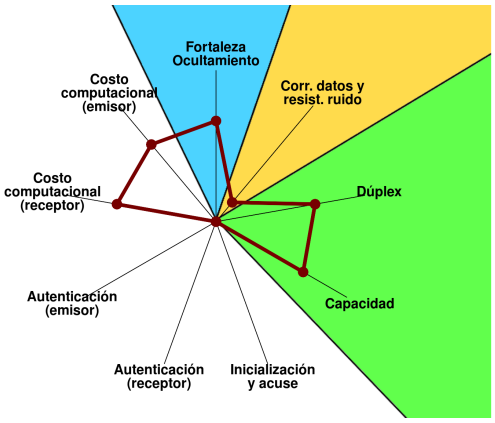
Decodificación del mensaje Conociendo u, y habiendo recibido el mensaje oculto consistente en (m, x, y), el *receptor* obtiene el mensaje oculto $m* = y^{-1}(m - ux) (\text{mod } p - 1)$.

Codificación de respuestas No contemplado.

Costo computacional Mediano.

El esquema propuesto se construye *abusando* de firmas realizadas bajo un modelo de criptografía de llave pública.

6.1.4. Capa 1 OSI: Disciplina serial



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Canal serial punto a punto	Canal temporal basado en valores	Estética	Sesión	Basado en la modulación de un flujo de datos sobre el tiempo, permitiendo (1) o deteniendo (0) el flujo.	El *receptor* envía un flujo de información irrelevante; el *emisor* modula dicho flujo con CTS/RTS, marcando intervalos determinados. La respuesta se codifica por este mismo proceso, con los actores invertidos.

Pueden crearse canales ocultos sobre medios ubicados a cualquier nivel de la pila de red; Handel y Sandford (1996) proponen formas simples de implementar canales ocultos en cada una de las capas del modelo OSI. En esta sección se aborda la capa 1 del modelo OSI (capa física). Los autores proponen "modular" una señal sobre un puerto serial de modo que, empleando las señales de control de contención *Clear to Send* (CTS) y *Ready to Send* (RTS), transmita un mensaje oculto (véase la figura 6.5. El escenario planteado para este ejemplo es que *Alice* y *Bob* quieren transmitirse un mensaje secreto, y sospechan que *Walter* tiene una escucha sobre el canal de comunicaciones (que en este caso podría ser un modem telefónico).

1. Naturaleza del canal

   Canal visible Comunicación de datos arbitrarios sobre un canal serial punto a punto.

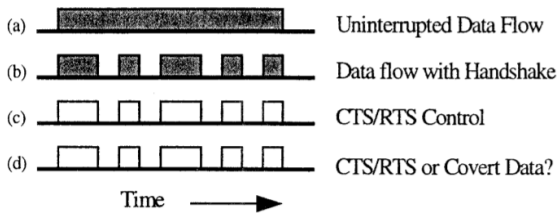


Figura 6.5: Uso de las señales CTS/RTS para "modular" datos sobre una transmisión serial (Handel y Sandford 1996).

Técnica de ocultamiento Canal temporal basado en valores.

Modulación del flujo de datos pausando la información por medio de CTS/RTS.

Fortaleza del ocultamiento Mediana.

Una frecuencia demasiado alta de transiciones CTS/RTS podría llamar la atención de *Walter*; el modelo de amenaza de este canal implica que un vigía que monitorea una línea serial.

Si disminuye la frecuencia de transiciones, aumenta la fortaleza del ocultamiento, pero disminuye correspondientemente la capacidad del canal.

Vector para el ocultamiento Estética.

Al ser este un mecanismo temporal (no esteganográfico), el contenido del canal visible resulta irrelevante. El mensaje oculto viaja *modulado* sobre la *forma* de la comunicación.

Mecanismo de corrección de errores No contemplado. Considerando comunicación serial, prácticamente la totalidad de modems hoy en día incorporan mecanismos de corrección de errores siguiendo el estándar V.42bis (CCITT 1990), sin embargo los autores en ningún momento asumen la implementación de este estándar.

Resistencia al ruido Siendo una conexión punto a punto, el ruido físico podría constituir ruido, alterando los símbolos CTS o RTS, y llevando a la pérdida de datos.

En caso de viajar la línea de comunicación sobre un canal que incorpore corrección de errores, estos símbolos no serán alterados, pero sí puede perderse la sincronía necesaria para el canal propuesto.

Dúplex Bidireccional.

Longevidad Establecimiento de sesión.

Capacidad Dependiendo de la codificación empleada. Los autores apuntan a que con una conexión a 9600bps es muy posible poder señalizar CTS/RTS creando un canal de 300bps.

2. Establecimiento y codificación

Inicialización No contemplado.

Acuse de establecimiento No contemplado.

Autenticación No contemplado; canal punto a punto.

Codificación y envío de datos Se elige una codificación de datos, y se emplea el control de contención de datos para permitir (1) o detener (0) el flujo de datos.

El *receptor* envía un flujo de datos (cuyo contenido es irrelevante para este propósito). El *emisor* le señaliza que detenga (CTS) o reanude (RTS) la transmisión por intervalos que codifiquen el mensaje oculto.

Mencionan los autores, "Los datos señalizados por CTS/RTS no tienen por qué ser ASCII, pueden seguir cualquier esquema de codificación (como podría ser el código Morse); un diseño cuidadoso del canal oculto y el uso de criptografía harían al descubrimiento e identificación de este canal más dificil."

Costo computacional Bajo.

La señalización requiere únicamente la inserción de los símbolos CTS/RTS en un flujo de datos, el mensaje no requiere ser modificado.

3. Reconocimiento y decodificación

Identificación de mensaje oculto No contemplado.

Evento disparador No contemplado.

Autenticación No contemplado; canal punto a punto.

Decodificación del mensaje El *receptor* envía un flujo de información (cuyo contenido es irrelevante). El *emisor* modula el flujo solicitándole que detenga (CTS) o continúe (RTS) la transmisión. Para cada intervalo dado, si el flujo está permitido el *receptor* registra 1, y si está detenido registra 0.

Codificación de respuestas La comunicación serial es típicamente bidireccional (aunque, dependiendo de la tecnología empleada, puede ser *dúplex completo* o *semi-dúplex*, y la velocidad de ambos canales puede ser asimétrica. Para el presente ejemplo, se asume dúplexcompleto y un canal simétrico). El canal de respuesta se maneja exactamente como su contraparte.

Costo computacional Bajo.

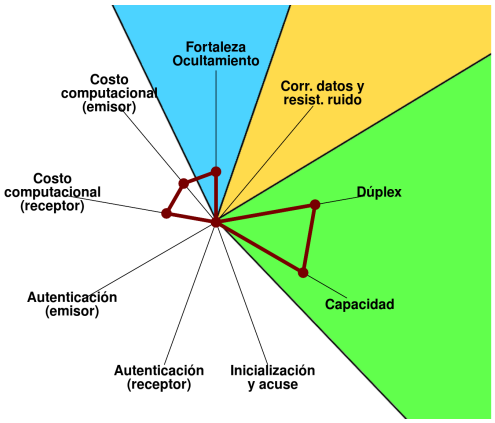
La sobrecarga computacional por monitorear las transiciones CTS/RTS es casi trivial; posiblemente sería conveniente que cada mensaje contara con un encabezado y finalización reconocibles para facilitar más esta detección, pero eso reduciría su fuerza de ocultamiento.

Este mecanismo está orientado a conexiones directas punto a punto, no a su uso en redes de datos, lo que limita su aplicabilidad. Mencionan los autores que un mecanismo similar puede emplearse en una red TCP/IP, en la capa de red,

enviando paquetes ICMP de tipo *source quench*. Esto, sin embargo, sería mas susceptible a los tiempos de entrega de paquetes poco previsibles inherentes a Internet, lo cual reduce fuertemente el ancho de banda alcanzable.

Cabe mencionar que si bien el *canal temporal basado en valores*, mecanismo básico empleado en este trabajo, sigue siendo propuesto en nuevas implementaciones. En los últimos días previos a la entrega del presente trabajo se dio a conocer (Caviglione y Mazurczyk 2014), que presenta posibles vectores de inyección a un teléfono celular *Apple iPhone* para poder modular la operación de su servicio de consulta de datos por voz *Siri*, obteniendo un canal oculto con una capacidad aproximada de un byte por cada dos segundos.

6.1.5. Capa 4 OSI: Manipulación del paquete TCP



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Conexión TCP/IP cualquiera	Esteganografía	Fática	Sesión	Cada byte se descompone en sus bits. Dos bits se codifican en el encabezado IP, seis en el TCP. Requiere ejecutarse con privilegios de administrador.	Se configura la interfaz en modo *promiscuo*; todos los paquetes con valores en los campos reservados van al buffer. Requiere ejecutarse con privilegios de administrador. La respuesta se codifica por este mismo proceso, con los actores invertidos.

Otro de los esquemas propuesto por (Handel y Sandford 1996) emplea la capa de transporte, que en redes IP reside en el protocolo TCP. El funcionamiento

de este esquema se basa en la estructura del encabezado, como lo ilustra la figura 6.6. Emplea los seis bits *reservados* del encabezado TCP, mas los dos bits no empleados en el campo *tipo de servicio* de TCP, para codificar un byte por paquete transmitido.

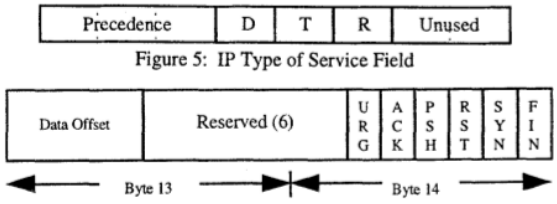


Figura 6.6: Ocultamiento de la información aprovechando los 8 bits redundantes en los encabezados TCP/IP (Handel y Sandford 1996).

1. Naturaleza del canal

   Canal visible Una conexión TCP/IP cualquiera.

   Técnica de ocultamiento Esteganografía.

   Uso de 8 bits *reservados* (no utilizados) en los encabezados de cada paquete, 2 en los encabezados de IP y 6 en los encabezados de TCP.

   Fortaleza del ocultamiento Baja.

   Un sistema de detección de intrusos (IDS) puede reportar la actividad inusual en campos reservados de los paquetes TCP/IP.

   Vector para el ocultamiento Fática.

   La información se oculta en el espacio que sería empleado para la descripción de la conexión misma.

   Mecanismo de corrección de errores No contemplado. Los autores explicitan que estos campos pueden ser descartados por los ruteadores, y que debe verse como un esquema débil.

   Los autores mencionan el aprovechamiento de los ocho bits para transmitir un byte como paquete — Esto indica que su esquema base no contempla detección ni corrección de errores.

   Resistencia al ruido Diversas configuraciones ampliamente utilizadas en equipos de ruteo *limpian* los campos reservados de TCP/IP, lo cual imposibilitaría sostener una comunicación como la propuesta. Incluso si se establece el canal, dada la naturaleza dinámica del cálculo de rutas IP, es posible que un cambio en los pesos de las rutas lleve a la pérdida de información.

   Dúplex Bidireccional. Ambos participantes de una conexión TCP/IP pueden utilizar el mismo esquema.

   Longevidad Establecimiento de sesión.

Capacidad Un byte por paquete TCP/IP. El paquete mínimo TCP/IP es de 41 bytes (20 de encabezado IP, 20 de encabezado TCP, 1 de carga útil mínima).

2. Establecimiento y codificación

Inicialización No contemplado.

Acuse de establecimiento No contemplado.

Autenticación No contemplado.

Codificación y envío de datos Los datos a transmitirse son tomados byte por byte. El *emisor* requiere ejecutarse con privilegios de administrador de sistema para poder crear *paquetes en crudo*; no establece la conexión empleando la pila TCP/IP del sistema operativo, sino que crea el paquete tal como será *depositado* en el medio físico.

Cada uno de los bytes a ser enviados se codifica con dos bits en el espacio reservado del encabezado IP, y los seis bits restantes en el espacio reservado del encabezado TCP.

Costo computacional Mediano.

El *emisor* tiene que fabricar y enviar a la red paquetes TCP/IP en crudo. Si bien este proceso por sí solo no es complejo, el hacerlo de forma que parezca una comunicación legítima y coherente puede ser demandante.

3. Reconocimiento y decodificación

Identificación de mensaje oculto No contemplado.

Evento disparador No contemplado.

Autenticación No contemplado.

Decodificación del mensaje El *receptor* requiere ejecutarse con privilegios de administrador de sistema para poder hacer una captura de paquetes en crudo de la red. De esta captura, filtra para su procesamiento únicamente aquellos paquetes que tengan información en los espacios reservados de los encabezados de TCP e IP; en caso de haberlos, obtiene el byte correspondiente, y lo va agregando al buffer con los datos ocultos recibidos.

Codificación de respuestas No contemplado explícitamente; el *emisor* y *receptor* únicamente intercambiarían su papel, y el mismo esquema podría emplearse en sentido inverso.

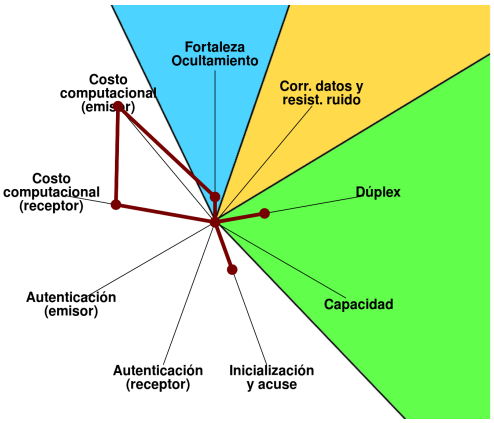
Costo computacional Mediano.

La tarjeta de red del sistema *receptor* debe configurarse en *modo promiscuo* y efectuar un filtrado de paquetes en crudo. La carga adicional de este procesamiento dependerá del tamaño y la arquitectura de la red en que se ubique.

Los autores mencionan que este mecanismo es relativamente débil: (traducción propia)

> *Walter* puede descubrir el uso de este espacio reservado si tiene activado el monitoreo de paquetes para detectar el uso de áreas reservadas. Algunos ruteadores pueden descartar esta información, dependiendo de cómo es la implementación del software de ruteo. Sin embargo, estos datos quedarán ocultos de un análisis normal.

6.1.6. *Port knocking*: puerto único, mapeo fijo



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Tráfico normal de red	Esteganografía	Conativa	Señalización	Intentos de conexión (`SYN`) a puertos TCP cerrados en secuencia predeterminada	Identificar la secuencia de paquetes con `SYN` de entre las configuraciones

El primer abordaje de un canal oculto como esquema para la legitima administración de sistemas, como se expuso en la sección 2.1.1, fue el *golpe de puerto* o *port knocking*, descrito por Krzywinsky (2003). La primera implementación de esta idea, si bien hoy en día aparece cruda y simplista, presenta un importante cambio en cómo se enfoca la atención los canales ocultos, por primera vez no vistos como una violación a las políticas de uso de red. La figura 6.7 ilustra el funcionamiento básico.

Esta sección y la siguiente presentan la evaluación sobre el modelo de dos modalidades que presenta el artículo citado. Además de estas dos, el artículo menciona a una tercera modalidad, *mapeo con cifrado*, pero no lo desarrolla lo suficiente como para poder evaluarlo.

1. Naturaleza del canal

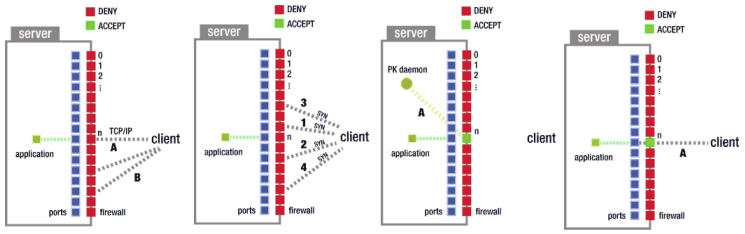


Figura 6.7: Esquema de operación del *port knocking*.

Canal visible Tráfico normal de red.

Técnica de ocultamiento Esteganografía.

La información viaja por una secuencia de solicitudes de conexión (paquetes `SYN`) a puertos de red cerrados, a los cuales se responde indicando que no hay ningún servicio a la escucha (`RST`).

Fortaleza del ocultamiento Muy baja.

Los paquetes SYN han sido reportados por los sistemas de detección de intrusos (IDS) desde bastante antes de la popularización del *port knocking* por su potencial para la denegación de servicio (CERT, Software Engineering Institute 1996). Un patrón, o un grupo de patrones, repetidos y con efectos observables (la disponibilidad de un puerto que aparentaba estar cerrado) pueden llevar a un vigilante a descubrir al canal.

Vector para el ocultamiento Conativa.

Las solicitudes de conexión (paquetes `SYN`) son la solicitud de iniciar una sesión TCP sobre el puerto determinado.

Mecanismo de corrección de errores No contemplado.

Resistencia al ruido Un *paquete IP perdido* causará que un comando no sea recibido; la demora en la entrega de un paquete IP puede romper el órden y por tanto alterar el significado del mensaje enviado. Muchos *firewalls* están configurados para no permitir la conexión a puertos no autorizados, con lo que el usuario puede no tener una ruta *libre* por la cual enviar sus *golpes de puerto*.

Dúplex Unidireccional.

Longevidad Señalización.

Capacidad Baja. Los ejemplos que presenta el artículo llevan hasta el equivalente a un total de tres comandos. Nada limita inherentemente a la capacidad del mensaje a esta longitud, pero un tamaño mayor pondría en riesgo la secrecía del esquema. El autor señala que sería posible emplear campos adicionales del paquete TCP para enviar información adicional, pero la implementación no lo hace.

2. Establecimiento y codificación

Inicialización No contemplada.

Acuse de establecimiento No contemplado explícitamente; por la naturaleza de las acciones propuestas para el *port knocking*, un acuse de establecimiento sería comprobar que el puerto solicitado esté ya abierto.

Autenticación No contemplado. El artículo presenta esquemáticamente un tercer modo, *mapeo con cifrado*, pero no lo detalla a un nivel suficiente.

La secuencia de *golpes* puede ser interceptada por un escucha que esté *olfateando* la red y repetida, no hay protección contra ataques de repetición.

Codificación y envío de datos El *emisor* realiza intentos de inicio de conexión TCP (paquetes `SYN`) a la secuencia de puertos correspondiente al mensaje a señalizar.

Costo computacional Nulo.

El universo de mensajes que pueden enviarse bajo este esquema es muy bajo, y el costo de generar un pequeño número de conexiones TCP es despreciable.

3. Reconocimiento y decodificación

Identificación de mensaje oculto Recepción de un paquete TCP/IP con la bandera `SYN` (intento de conexión) a alguno de los puertos cerrados determinados.

Evento disparador Recepción de una secuencia de intentos de conexión válida.

Autenticación No contemplada.

Decodificación del mensaje El *firewall* del sistema *receptor* registrará los intentos de conexión a los puertos determinados en la bitácora. Un segundo programa monitorea la bitácora, en espera del registro de estos intentos de conexión.

Si se observa uno de los patrones establecidos en la configuración de dicho programa,[2] se ejecuta la acción correspondiente. Las acciones forman parte de una suerte de diccionario.

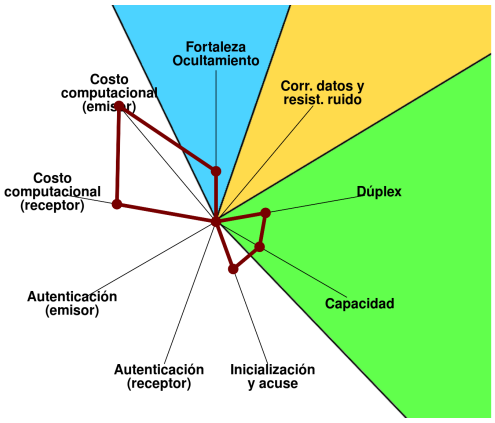
Codificación de respuestas No contemplada.

Costo computacional Bajo.

El *firewall* del sistema *receptor* está integrado al sistema operativo y diseñado de forma muy eficiente para lidiar con grandes cantidades de tráfico; el servicio que espera estos *golpes de puerta* únicamente monitorea la bitácora del sistema, una tarea muy poco demandante.

[2] El artículo presenta como ejemplo que la secuencia (31, 32, 30) causa que se abra el puerto 22 a la dirección IP origen, que la secuencia (32, 30, 31) causa que se cierre el puerto 22 a la dirección IP origen, y que la secuencia (31, 30, 32) causa que se cierre la conexión y se ignoren todos los intentos futuros de la dirección origen, para prevenir ataques de reproducción.

6.1.7. *Port knocking*: puertos múltiples, mapeo dinámico



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Tráfico normal de red	Esteganografía	Conativa	Señalización	Encabezado fijo, seguido de una serie de *golpes* que codifican acorde a la configuración el puerto a abrir.	Detección de secuencia de paquetes `SYN` (encabezado, puerto, verificación, finalización) en la bitácora del firewall.

1. Naturaleza del canal

   Canal visible Tráfico normal de red.

   Técnica de ocultamiento Esteganografía.

   La información viaja por una secuencia de solicitudes de conexión (paquetes `SYN`) a puertos de red cerrados, a los cuales se responde indicando que no hay ningún servicio a la escucha (`RST`).

   Fortaleza del ocultamiento Baja.

   Los paquetes SYN han sido reportados por los sistemas de detección de intrusos (IDS) desde bastante antes de la popularización del *port knocking* por su potencial para la denegación de servicio (CERT, Software Engineering Institute 1996). Un patrón, o un grupo de patrones, repetidos y con efectos observables (la disponibilidad de un puerto, por más que este varíe, que aparentaba estar cerrado) pueden llevar a un vigilante a descubrir al canal. A diferencia del caso anterior, en este esquema no siempre se transmitirá la misma secuencia. Sin embargo, la variabilidad es bastante menor.

   Vector para el ocultamiento Conativa.

Las solicitudes de conexión (paquetes `SYN`) son la solicitud de iniciar una sesión TCP sobre el puerto determinado.

Mecanismo de corrección de errores No contemplado.

Resistencia al ruido Un *paquete IP perdido* causará que un comando no sea recibido; la demora en la entrega de un paquete IP puede romper el órden y por tanto alterar el significado del mensaje enviado. Muchos *firewalls* están configurados para no permitir la conexión a puertos no autorizados, con lo que el usuario puede no tener una ruta *libre* por la cual enviar sus *golpes de puerto*.

Dúplex Unidireccional.

Longevidad Señalización.

Capacidad Baja. Los ejemplos que presenta el artículo llevan hasta el equivalente a cinco bytes de carga útil. Nada limita inherentemente a la capacidad del mensaje a esta longitud, pero un tamaño mayor pondría en riesgo la secrecía del esquema. El autor señala que sería posible emplear campos adicionales del paquete TCP para enviar información adicional, pero la implementación no lo hace.

2. Establecimiento y codificación

Inicialización El mensaje inicia con una secuencia de intentos de conexión (`SYN`) predefinida a modo de encabezado.

Acuse de establecimiento No contemplado explícitamente; por la naturaleza de las acciones propuestas para el *port knocking*, un acuse de establecimiento sería comprobar que el puerto solicitado esté ya abierto. Este acuse se presenta, sin embargo, sólo después de la finalización del canal.

Autenticación No contemplada.

La secuencia de *golpes* puede ser interceptada por un escucha que esté *olfateando* la red y repetida, no hay protección contra ataques de repetición.

Codificación y envío de datos El autor sugiere el envío, en hasta cuatro paquetes, del puerto destino, seguido de una suma de verificación (*checksum*) validando su recepción correcta.[3]

Costo computacional Muy bajo.

El universo de mensajes que pueden enviarse bajo este esquema es limitado, y el costo de generar un pequeño número de conexiones TCP es despreciable.

3. Reconocimiento y decodificación

[3] El ejemplo presentado asume que se monitorearán los puertos 100–109. Si el *emisor* desea conectarse al puerto 143, podría codificarlo como 10**0** 10**1** 10**4** 10**3**, seguido de la verificación: $(0 + 1 + 4 + 3) \bmod 10 = 8$, codificado como puerto 108.

Identificación de mensaje oculto Recepción de intentos de conexión (`SYN`) a los puertos en la secuencia que conforma al encabezado.

Evento disparador Recepción de una secuencia de intentos de conexión válida.

Autenticación No contemplada.

Decodificación del mensaje El *firewall* del sistema *receptor* registrará los intentos de conexión a los puertos determinados en la bitácora. Un segundo programa monitorea la bitácora, en espera del registro de estos intentos de conexión.

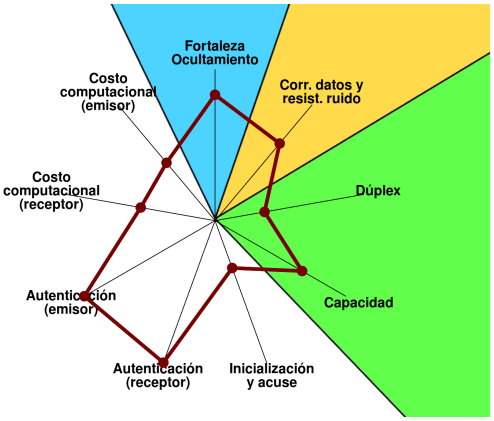
Si se observa en la bitácora una secuencia (encabezado, puerto, verificación, finalización) válida, se ejecuta la acción correspondiente (abrir el puerto indicado).

Codificación de respuestas No contemplada.

Costo computacional Muy bajo.

El *firewall* del sistema *receptor* está integrado al sistema operativo y diseñado de forma muy eficiente para lidiar con grandes cantidades de tráfico; el servicio que espera estos *golpes de puerta* únicamente monitorea la bitácora del sistema, una tarea muy poco demandante.

6.1.8. Autenticación por un solo paquete: fwknop



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Tráfico normal de red	Esteganografía, criptografía	Referencial	Señalización	Cadena de texto que incluye *token*, nombre de usuario, *timestamp*, acción preconfigurada deseada y otros datos, cifrados y validados por hash	

La idea del *port knocking* resultó inovadora e inspiradora, pero no tardaron mucho en aparecer carencias en su planteamiento original, como puede apreciarse en el trabajo de Izquierdo Manzanares y col. (2005). Parte importante de la debilidad del *port knocking* deriva del limitado espacio que un paquete TCP ofrece para ocultar información. Uno de los puntos más frecuentemente citados como debilidad de dicho esquema, derivados de lo limitado del espacio, es la vulnerabilidad ante los ataques de repetición y la falta de un mecanismo expreso de autenticación.

El siguiente paso en este camino lo constituyen las implementaciones de *autenticación por un solo paquete* (*Single Packet Authentication*); la implementación que a continuación se aborda es *fwknop*. (Rash 2007; Rash 2007–2014)

1. Naturaleza del canal

   Canal visible Tráfico normal de red.

   Técnica de ocultamiento Esteganografía, criptografía.

   Se envía un solo paquete (no necesariamente `SYN`) a un puerto cerrado. El paquete incluye como carga útil un mensaje cifrado indicando los datos de autenticación y la acción a realizar.

   Fortaleza del ocultamiento Mediana / alta.

   Si bien se trata de un paquete no asociado a ninguna conexión establecida como en el caso del *port knocking*, en este caso es únicamente un paquete el que se envía (y no una secuencia). Al presentar mayor riqueza de configuración en lo relativo a sus acciones desde su planteamiento, *fwknop* resulta al fin más enfocado al establecimiento de un verdadero *canal oculto* que como meramente un auxiliar de reconfiguración del firewall.

   Cabe mencionar, sin embargo, que los paquetes de *fwknop* siguen siendo atípicos: Un sistema de detección de intrusos configurado de forma restrictiva podría identificar y hasta bloquear a estos paquetes sin estar explícitamente configurado para ello, aún si, no puede determinar su función.

   Vector para el ocultamiento Referencial.

   El paquete único enviado se *esconde* en su contexto: el hecho de ser apenas una mota de ruido en la red, siendo demasiado

pequeño para disparar alarmas, y no estar relacionado a nada más.

Mecanismo de corrección de errores El mensaje incluye un hash (MD5) del mensaje dentro del texto cifrado.

Resistencia al ruido Un *paquete IP perdido* causará que un comando no sea recibido. Muchos *firewalls* están configurados para no permitir la conexión a puertos no autorizados, con lo que el usuario puede no tener una ruta *libre* por la cual enviar el paquete.

Dúplex Unidireccional.

Longevidad Señalización.

Capacidad Hasta 1500 bytes (el MTU más común en red). En su configuración típica, un mensaje lleva 128 bytes de carga útil.

2. Establecimiento y codificación

Inicialización El *emisor* envía un paquete (por omisión, al puerto UDP 62201, aunque puede configurarse para no ser en un puerto específico) con el mensaje cifrado como carga útil.

Acuse de establecimiento No contemplado.

Autenticación En su configuración por omisión, el mensaje se compone por un token aleatorio de 16 bytes (para prevenir ataques de repetición), un nombre de usuario, *timestamp*, y va cifrado empleando Rijndael (clave simétrica).
Adicionalmente, puede emplearse un cifrado de llave pública GPG.

Codificación y envío de datos El *emisor* genera una línea de texto con los valores a enviar (token aleatorio, nombre de usuario, *timestamp*, versión del cliente, modo de acceso, acción deseada, y verificación MD5 del mensaje) separados por el caracter `:`. Se cifra la línea empleando Rijndael con la clave especificada, y se envía codificado Base64 como *carga útil* del mensaje..

Costo computacional Bajo / mediano.

Si bien el mensaje se cifra y firma, la longitud del mensaje es tan corta que su procesamiento resulta prácticamente trivial.

3. Reconocimiento y decodificación

Identificación de mensaje oculto Decodificación y verificación exitosa del mensaje recibido.

Evento disparador Recepción de un paquete del tipo correcto al puerto (o rango de puertos) especificado.

Autenticación Verificación de que el mensaje esté cifrado con la llave correcta, por omisión bajo el algoritmo Rijndael. Alternativamente, puede especificarse que emplee una llave asimétrica con GPG.

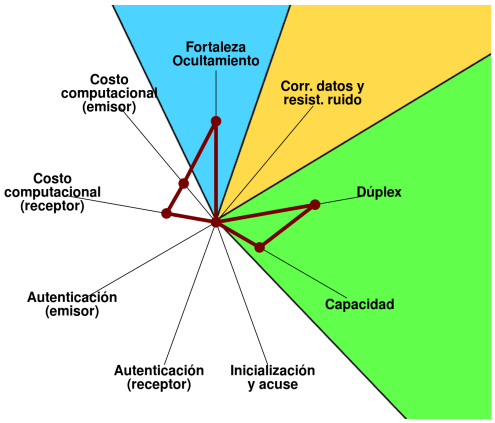
Decodificación del mensaje El *receptor* valida que el mensaje esté bajo una codificación correcta Base64, que la autenticación sea correcta, y que la verificación MD5 resulte exitosa. Ejecuta la acción configurada para el quinto campo (acción).

Codificación de respuestas No contemplado.

Costo computacional Bajo / mediano.

Si bien el mensaje se cifra y firma, la longitud del mensaje es tan corta que su procesamiento resulta prácticamente trivial.

6.1.9. Esteganografía práctica en Internet



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Una conexión TCP/IP cualquiera	Esteganografía	Fática	Sesión	Bits enviados secuencialmente, empleando el campo DNF de TCP en paquetes cortos.	Captura de paquetes en crudo; busca paquetes chicos pertenecientes a la conexión portadora, obtiene la bandera DNF de cada paquete, armando el mensaje bit a bit.

La propuesta de Kundur y Ahsan (2003) es muy parecida a la abordada en la sección 6.1.5: Emplea el espacio redundante en los encabezados de paquetes TCP para la creación del canal oculto. La principal diferencia radica en que este

texto los autores ponen énfasis en que el canal no sea fácilmente detectable o susceptible a perderse por configuración de los ruteadores, manteniendo paquetes TCP completamente válidos — Aunque esto, como se indicó en la sección 3.3, supone una disminución del ancho de banda disponible.

ID field	Flags	Frag.Offset	Total Length
XXX..XX	010	000...0	472

ID field	Flags	Frag.Offset	Total Length
XXX..XX	000	000...0	472

Figura 6.8: Codificación propuesta por (Kundur y Ahsan 2003) para viajar sobre un paquete TCP legal. El primer cuadro presenta los encabezados de un datagrama de valor oculto "1", el segundo los de otro de valor oculto "0".

1. Naturaleza del canal

   Canal visible Una conexión TCP/IP cualquiera.

   Técnica de ocultamiento Esteganografía.

   Uso de la bandera *Do Not Fragment* del encabezado TCP en paquetes estrictamente menores al MTU observado en la red (o las redes) en cuestión.

   Fortaleza del ocultamiento Mediana.

   Al encontrar la redundancia en el uso completamente legal de un campo, este mecanismo de ocultamiento no suena alarmas en los sistemas de detección de intrusos (IDS). La restricción de manejar únicamente paquetes cortos hace que se pueda esconder perfectamente en protocolos típicamente interactivos, como `ssh`. Una inspección humana y a profundidad de los paquetes puede llamar la atención a los valores cambiantes en *Do Not Fragment*, pero únicamente para quien conoce (y espera) el uso de este mecanismo.

   Vector para el ocultamiento Fática.

   La información oculta viaja sobre el espacio del canal visible dedicado a señalizar acerca de su propia estructura.

   Mecanismo de corrección de errores No contemplado.

   Resistencia al ruido No contemplado. Los autores mencionan que todos los paquetes generados serán válidos, y el protocolo TCP garantiza su llegada en la secuencia correcta.
   La gran cantidad de mensajes requeridos para la transmisión de cualquier mensaje puede alertar a los opoeradores de la red. Sin embargo, en un contexto en que las transferencias de gigabytes son ya cosa rutinaria, es altamente improbable que esto llame la atención.

Dúplex Bidireccional. Ambos participantes de una conexión TCP/IP pueden utilizar el mismo esquema.

Longevidad Establecimiento de sesión.

Capacidad Un bit por paquete TCP/IP. El paquete mínimo TCP/IP es de 41 bytes (20 de encabezado IP, 20 de encabezado TCP, 1 de carga útil mínima).

2. Establecimiento y codificación

Inicialización No contemplado.

Acuse de establecimiento No contemplado.

Autenticación No contemplado.

Codificación y envío de datos Los datos a transmitirse son convertidos a un vector de bits, y cada uno de ellos es enviado secuencialmente. El *emisor* requiere ejecutarse con privilegios de administrador de sistema para poder crear *paquetes en crudo*; no establece la conexión empleando la pila TCP/IP del sistema operativo, sino que crea el paquete tal como será *depositado* en el medio físico.

Cada uno de los bits a ser enviados se codifica en el encabezado *Do Not Fragment* del protocolo TCP en paquetes cortos (que no serían fragmentados de cualquier forma).

Costo computacional Mediano.

El *emisor* tiene que fabricar y enviar a la red paquetes TCP/IP en crudo. Si bien este proceso por sí solo no es complejo, el hacerlo de forma que parezca una comunicación legítima y coherente puede ser demandante.

3. Reconocimiento y decodificación

Identificación de mensaje oculto No contemplado.

Evento disparador No contemplado.

Autenticación No contemplado.

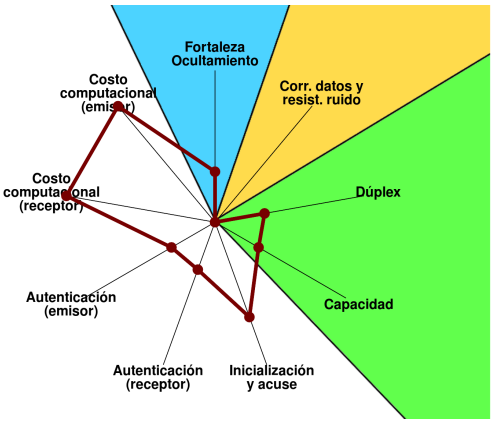
Decodificación del mensaje El *receptor* requiere ejecutarse con privilegios de administrador de sistema para poder hacer una captura de paquetes en crudo de la red. De esta captura, filtra para su procesamiento únicamente aquellos paquetes que pertenezcan a la conexión TCP que ya se conoce como portadora de un canal oculto. De cada mensaje recibido, obtiene el bit correspondiente, y lo va agregando al buffer con los datos ocultos recibidos.

Codificación de respuestas No contemplado explícitamente; el *emisor* y *receptor* únicamente intercambiarían su papel, y el mismo esquema podría emplearse en sentido inverso.

Costo computacional Mediano.

La tarjeta de red del sistema *receptor* debe configurarse en *modo promiscuo* y efectuar un filtrado de paquetes en crudo. La carga adicional de este procesamiento dependerá del tamaño y la arquitectura de la red en que se ubique.

6.1.10. *Webknocking*: Golpea diferente



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Solicitudes Web (HTTP o HTTPS)	Esteganografía	Conativa	Señalización	No hay envío posterior, la solicitud va codificada en los parámetros `GET` o `POST` de la solicitud a la página *maestra*.	No hay envío posterior, la solicitud se codifica en los parámetros `GET` o `POST` de la solicitud a la página *maestra*.

El *Webknocking* nació motivado por las debilidades del *port knocking* (Lebelt 2005). Este concepto, presentado de forma informal por el autor publicando una breve descripción y el código de su implementación en su sitio Web, aprovecha las ventajas de una conexión establecida TCP/IP, y decide ocultar o *ahogar* el tráfico administrativo en una capa superior de red, que le brinda varias ventajas adicionales: Las solicitudes Web.

1. Naturaleza del canal

   Canal visible Solicitudes Web (HTTP o HTTPS).

   Técnica de ocultamiento Esteganografía.

   De forma análoga a funcionamiento del *port knocking*, el *emisor* hará una serie de solicitudes a páginas Web al servidor del *receptor*.

Fortaleza del ocultamiento Baja.

Si bien el inicio de la interacción puede ser completamente oculta (pasando por una serie de páginas no relacionadas entre sí), la penúltima y última páginas solicitadas serán siempre `webknocking.php`, un nombre por demás obvio, y presentando los argumentos en texto plano. El script podría renombrarse, pero seguirá respondiendo de forma estática y requiriendo una invocación expresa.

Si se despliega sobre un servidor Web cifrado (HTTPS), el ocultamiento mejora fuertemente.

Vector para el ocultamiento Conativa.

La información viaja sobre las distintas solicitudes de páginas sobre HTTP.

Mecanismo de corrección de errores No contemplado.

Resistencia al ruido No contemplado.

Dúplex Unidireccional.

Longevidad Señalización.

Capacidad El mecanismo propuesto es similar al presentado en la sección 6.1.6: La secuencia correcta presentada una vez abre el acceso a determinado puerto para el host solicitante, y una segunda vez lo cierra.

El mecanismo podría ser modificado o extendido de muchas maneras; se analiza la versión tal cual fue publicada.

2. Establecimiento y codificación

Inicialización La configuración del servicio indica una lista estática de páginas a visitar a modo de encabezado (`$neededpages`). Estas páginas tienen que visitarse en el órden y periodo máximo establecido.

Acuse de establecimiento Tras abrir el puerto, el servidor Web entrega una página con únicamente la palabra `OPENING`; tras cerrarlo, entrega `CLOSING`.

Autenticación La autenticación es muy básica, un reto-respuesta. El autor la presenta como un *juego de preguntas*, que pueden ser una respuesta aritmética o preguntas predefinidas por el administrador. La pregunta es presentada en claro al hacer la primera solicitud a la página `webknocking.php` posterior al encabezado.

Codificación y envío de datos Tras el encabezado y la autenticación no hay envío posterior de datos; el sistema conoce sólo el estado *abierto* y *cerrado*, y alterna entre ellos.

Costo computacional Nulo.

El mensaje se transmite en claro, siempre que no emplee HTTPS.

3. Reconocimiento y decodificación

   Identificación de mensaje oculto Cuando la página `webknocking.php` es llamada sin argumentos, este programa busca en la bitácora del servidor Web (en sus últimas líneas, según el tiempo configurado) la secuencia estática de páginas preestablecida.

   Evento disparador La visita a la página *maestra*, `webknocking.php`

   Autenticación Muy básica, un reto-respuesta. El autor la presenta como un *juego de preguntas*, que pueden ser una respuesta aritmética o peguntas predefinidas por el administrador. Las respuestas son transmitidas como parte de la siguiente solicitud Web.

   Decodificación del mensaje No hay envío posterior de datos; el sistema conoce sólo el estado *abierto* y *cerrado*, y alterna entre ellos. El estado del sistema es determinado por la existencia de un archivo `/tmp/open`.

   Codificación de respuestas No contemplado.

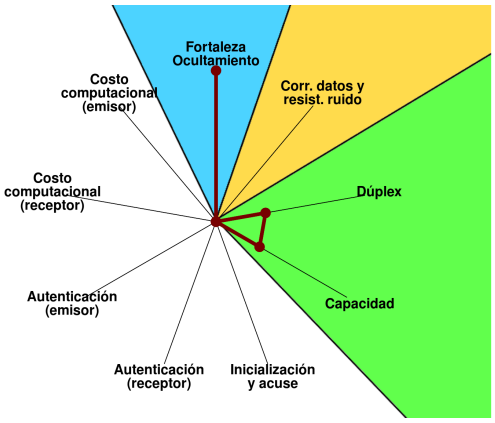
   Costo computacional Nulo.

   El mensaje se transmite en claro, siempre que no emplee HTTPS.

Queda claro que este es un enfoque muy débil desde muchos de los puntos de vista descritos en este trabajo; se presenta por ser un relativo precursor. Indudablemente la implementación podría enriquecerse para fortalecer muchos de sus puntos débiles; detallar en el modelo sus fortalezas y carencias ayuda a comprender qué aspectos podrían mejorarse.

El esquema aquí abordado se parece mucho a la implementación independiente desarrollada (también informalmente) por Briganti (2012), con la diferencia de que Briganti enfrenta a la debilidad de la *autenticación* enviando como parámetro de la solicitud HTTP al MD5 de una contraseña concatenada con la fecha y hora actual. Esto mejora sensiblemente la fortaleza de la autenticación, pero la elección de MD5 resulta desafortunada por su haber sido demostrado vulnerable a colisiones (Kaminsky 2004); adecuarlo a otros esquemas resulta, afortunadamente, trivial. El sistema de Briganti, que emplea como *nonce* a la fecha y hora actual (con resolución de minutos) resulta medianamente vulnerable a ataques de repetición, al no implementar verificación alguna de que un *nonce* sea efectivamente de uso único.

6.1.11. Escondiéndose en el spam



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Correo electrónico no solicitado	Esteganografía (función mímica)	Emotiva, conativa, referencial, fática	Señalización	Separado en bloques, cada uno se pasa por un *parser inverso* que entrega la cadena que representa al árbol de parseo en cuestión.	Mensaje entregado al *parser* bloque a bloque; éste construye un árbol de parseo, obtiene su representación numérica, que es el mensaje oculto.

En la sección 2.2 se presentaron algunos ejemplos de canales ocultos que emplean como cubierta al correo *spam*: Dado que el envío de spam se ha vuelto un arte que busca engañar a los filtros automatizados, estos mensajes de correo indeseado contienen una gran cantidad de redundancia y de *aparentes errores* en su formación. Además, tienden a tener texto repetitivo y lleno de exclamaciones de todo tipo.

El sitio Web *SpamMimic* (McKellar 2000-2014) implementa las funciones mímica basadas en gramáticas regulares descritas por Wayner (2009).

El servicio que ofrece *SpamMimic* no constituye un canal oculto como los que aborda este trabajo, dado que cada mensaje requiere ser reconocido y procesado por un humano. Asumiendo un mecanismo (no determinado) de reconocimiento para su decodificación, procedemos con su análisis por presentar un acercamiento poco explorado.

1. Naturaleza del canal

   Canal visible Correo electrónico no solicitado (*spam*).

   Técnica de ocultamiento Esteganográfica.

El mensaje visible es generado eligiendo un valor que represente a la cadena oculta a transmitir, y construido mediante la *caminata* del árbol de parseo inverso de una gramática regular.

Fortaleza del ocultamiento Alta.

El mensaje visible no guarda correspondencia alguna (ni siquiera longitud del texto generado) con el oculto, y las modificaciones incluso más pequeñas a la gramática generadora cambian por completo los mensajes. El *spam* emplea suficientes mecanismos de engaño para que los filtros le permitan el paso que resultaría muy complicado apuntar a *un mensaje* particular (es por esto que los principales mecanismos *anti-spam* operan basados en modelos markovianos de probabilidad).

Vector para el ocultamiento Emotiva, conativa, referencial, fática.

El mensaje cubierta incluye componentes que van sobre diferentes vectores, desde presentar al supuesto remitente (emotivo), el famoso "¡Compre ya!" (conativo), testimonios de clientes satisfechos (referencial), promesas de que es un correo único y no hace falta responder (fático), etcétera.

Mecanismo de corrección de errores No contemplado.

Resistencia al ruido No contemplado.

Dúplex Unidireccional.

Longevidad Señalización.

Capacidad Empíricamente, el mensaje fuente se separa en bloques de hasta 13 caracteres, generando cada uno un bloque de 800 a 1000 caracteres. Esto es, una eficiencia máxima cercana al 1.5 %.

2. Establecimiento y codificación

Inicialización No contemplado.

Acuse de establecimiento No contemplado.

Autenticación No contemplado.

Codificación y envío de datos El mensaje fuente es separado en bloques, cada uno de los cuales es convertido a su representación numérica. Se aplica entonces un *parseo* inverso con la gramática (la cual no está disponible públicamente) obteniendo la cadena que generaría dicho número al ser *parseado*. Se transmite la cadena obtenida.

Costo computacional Alto.

Si no se emplea un *parser* altamente optimizado, la complejidad de recorrer la gramática para crear el mensaje correspondiente es muy alta.

3. Reconocimiento y decodificación

Identificación de mensaje oculto No contemplado.

Evento disparador No contemplado.

Autenticación No contemplado.

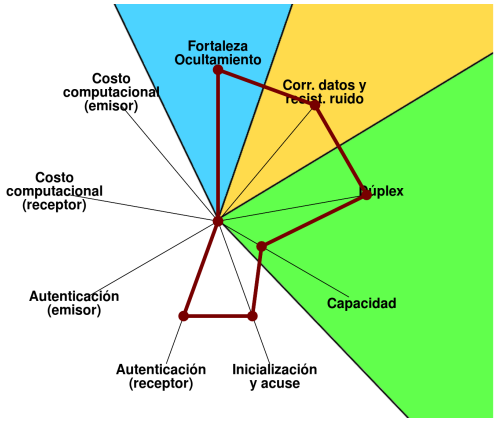
Decodificación del mensaje El mensaje recibido es entregado al *parser*, se construye un árbol de parseo, obteniendo su representación numérica. Cada uno de estos números representa un bloque del mensaje oculto.

Codificación de respuestas No contemplado.

Costo computacional Muy alto.

Además del costo computacional del *parser*, por la mecánica de *backtracking* normalmente utilizada, la sobrecarga para detectar mensajes que *no* lleven un mensaje oculto puede ser superior al de cuando sí lo hay.

6.1.12. Comunicación oculta entre servidores HTTP



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Comunicación estándar HTTP	Esteganografía	Referencial, conativa	Señalización	Para cada mensaje, se calcula una secuencia aleatoria. Es comunicado de servidor a servidor, empleando parámetros y *cookies* de las solicitudes Web de los clientes (no participantes en el esquema). No especifica la codificación específica a realizar sobre HTTP.	No especifica mecanismo de decodificación (sólo el espacio donde se envían los datos codificados). Para la respuesta, emplea el mismo esquema multidireccional.

Bauer 2003 presenta una propuesta que implementa la *inligabilidad* (*unlinkability*) y la *inobservabilidad* (*unobservability*) mediante *mezcladores* descrita en Chaum (1981). Presenta un protocolo para crear una *red superpuesta anónima* basada en la navegación Web de usuarios regulares. El modelo de amenaza presentado por Bauer se enfoca en mantener comunicación que no sea detectada dentro de las capacidades legales de monitoreo del FBI.[4]

La principal diferencia del trabajo de Bauer con otros esquemas es que éste plantea la comunicación oculta entre dos *servidores* HTTP, empleando como medio a los *clientes* (navegadores), sin que necesariamente éstos estén al tanto de que están siendo empleados para la comunicación.

1. Naturaleza del canal

   Canal visible Comunicación estándar HTTP.

   Técnica de ocultamiento Esteganografía.

   Codificación de datos en determinados encabezados y elementos de HTTP y HTML:

   - Redirecciones
   - Galletas (*cookies*)
   - Encabezado *referido por* (*referer*)
   - Elementos HTML que solicitan contenido de terceros
   - Contenido activo (código ejecutable en el navegador)

   Fortaleza del ocultamiento Alta.

   Los mensajes intercambiados no son inherentemente distintos de los que llevaría el tráfico común HTTP, particularmente con la presencia

[4] Doce años después de la publicación del trabajo, es ya bien sabido que ha habido muy numerosas escuchas extralegales.

de *banners* y demás anuncios comperciales. Esta fortaleza crece naturalmente entre más nodos participen en la red, dado que no se verá un intercambio notable de *banners* entre el mismo grupo de servidores.

El planteamiento incluye un componente aleatorio enfocado a no enviar la información pendiente de inmediato, de forma que sea más dificil hacer un rastreo entrada-salida de los paquetes.

Vector para el ocultamiento Referencial / conativa.
La información va oculta en los elementos que definen el contexto (galletas, redirecciones, *referer*), y su propagación va mediada por los elementos que solicitan al navegador realizar determinadas acciones (elementos HTML, contenido activo).

Mecanismo de corrección de errores Todos los mensajes, así como todas las confirmaciones de respuesta, van criptográficamente firmados.

Resistencia al ruido El primer protocolo descrito en el trabajo resulta vulnerable a potenciales negaciones de servicio por parte de un navegador adversario,[5] pero corrigen la debilidad con un segundo protocolo que requiere confirmaciones de recepción para cada mensaje enviado.
El autor cita entre las razones para emplear HTTP como transporte el que típicamente no es bloqueado ni modificado por firewalls o traducción de direcciones de red (NAT).

Dúplex Multidireccional — El canal oculto no es punto a punto, sino que una *red superpuesta* entre todos los equipos participantes.

Longevidad Señalización. Cada mensaje es enviado como una entidad discreta. Un mensaje largo puede separarse en varios mensajes y concatenarse al recibirlo.

Capacidad Menciona el autor que para un mensaje de 4K, el tráfico resultante es similar al de un *banner* de anuncio, 16KB. El tiempo de entrega de un mensaje, sin embargo, puede ser muy largo si no hay un flujo suficiente de navegadores visitando los diversos sitios de esta red.

2. Establecimiento y codificación

Inicialización Cuando el *emisor* tiene un mensaje para enviar, espera la conexión de un cliente Web, y envía como parte de la página generada una liga que le haga comunicarse a otro de los nodos de la red (empleando un elemento HTML como `frame`, `iframe`, `img`, `script`, `link`, etc.)

[5] Configuración que sería cada vez más común de encontrar hoy en día, en que muchos usuarios –particularmente los conscientes de temas de seguridad y privacidad– bloquean de forma selectiva la ejecución de Javascript y despliegue de anuncios por medio de extensiones al navegador como *NoScript*, *AdBlock* y similares.

El mensaje a transmitir es enviado empleando alguno de los mecanismos descritos anteriormente. Este mensaje se mantiene en la *cola de envío* hasta recibir confirmación del *receptor*.

Acuse de establecimiento Tras recibir y verificar un mensaje se genera una *confirmación*. Un nodo de la red intentará reenviar el mensaje repetidamente hasta recibir su confirmación.

Autenticación Todos los mensajes transmitidos de un nodo a otro son firmados empleando un esquema de llave pública. Además de estas firmas por cada *salto*, el mensaje es además cifrado con la llave pública del destinatario.

Codificación y envío de datos No se especifica el mecanismo de codificación (sólo el espacio en que se envían los datos codificados).

Para realizar el envío de un mensaje m a un destino δ, el *emisor*:

- Calcula una secuencia aleatoria de $S = (1..\delta)$ para el envío, terminada con el nodo destino.
- Cifra el mensaje m con la llave pública del destino δ: $m_\delta = E_\delta(m)$
- Cifra el mensaje (ya cifrado) m_δ, la secuencia restante $S_r = S - S_1$ y el hash del mensaje m_δ (para la confirmación) con la llave pública del primer salto s_1: $m_{s_1} = E_{s_1}(\text{To :}||S_r||\text{Ack :}||h(m_\delta)||m_\delta)$
- Envía m_{s_1} a S_1 por medio de la interacción ya descrita con un cliente.

Costo computacional Alto.

Todos los mensajes deben cifrarse doblemente (con la llave del destinatario y con la del siguiente *salto*) y firmarse a cada paso hacia su entrega.

3. Reconocimiento y decodificación

Identificación de mensaje oculto Toda solicitud de un recurso por parte de un cliente Web que pueda descifrarse o validarse con la llave privada de un nodo n identifica la presencia de un mensaje oculto.

Evento disparador La solicitud de un recurso Web de la naturaleza especificada.

Autenticación No contemplada. El mensaje viaja cifrado para que únicamente el *receptor* lo pueda obtener, pero el esquema descrito no contempla autenticación de origen.

Decodificación del mensaje No se especifica el mecanismo de codificación/decodificación (sólo el espacio en que se envían los datos codificados).

Codificación de respuestas Mismo esquema que el ya descrito.

Costo computacional Alto.

Todos los mensajes deben descifrarse a cada salto y en el destinatario último, y su firma es verificada a cada paso hacia su entrega.

Tras un análisis descriptivo de Bauer (2003) debe quedar clara su similitud con las redes anonimizadoras basadas en *ruteo cebolla*, la más popular de las cuales hoy en día es el *Proyecto TOR*. Uno de los primeros ejemplos de análisis al ruteo cebolla es (Reed, Syverson y Goldschlag 1996), del cual se presenta un esquema básico en la figura 6.9.

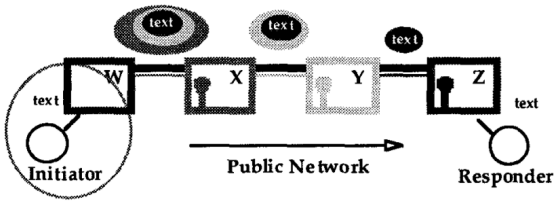


Figure 6: Moving Data Forward

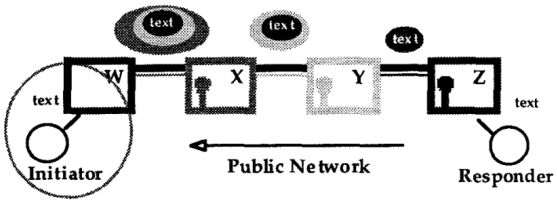
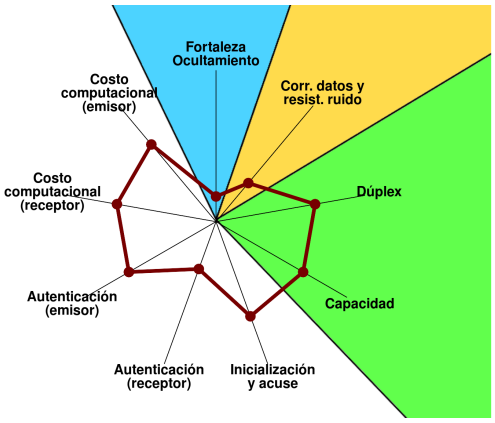


Figure 7: Moving Data Backward

Figura 6.9: Representación del *ruteo cebolla* en la obra de Reed, Syverson y Goldschlag (1996).

No se aborda a mayor detalle ninguna otra de las implementaciones de ruteo cebolla dado que exceden el ámbito del presente trabajo: Si bien se encamina a la preservación del anonimato dentro de las comunicaciones y emplea una lógica muy cercana a la ya descrita, no implementan un *canal oculto* sino que un *canal cifrado* — siendo su principal contribución el ocultar en todo punto de su tránsito las direcciones origen y destino de la comunicación.

6.1.13. Puertas traseras para atravesar firewalls



Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Solicitud sobre conexión HTTP	Esteganografía	Referencial	Señalización	Codificación de la solicitud como Base64, enviado como parámetro HTTP	

Con un artículo informal publicado en la revista *The Hacker's Choice*, van Hauser (1999) describe su implementación para poder controlar fácilmente a los sistemas que ha vulnerado sin tener que repetir el camino tedioso y específico cada vez que quiera poner a *trabajar* a un ejército de *bots*. Este texto claramente es presentado desde el punto de vista de un atacante; se analiza independientemente de su intencionalidad como un ejemplo más de un canal que debe permanecer oculto — En este caso, sí, por ir claramente en contra de las políticas de uso aceptable de los recursos. El autor asume un servidor de contenido Web protegido por firewall, al cual sólo se puede llegar por el puerto 80 TCP, y en el cual el atacante encontró una vulnerabilidad explotable, como lo ilustra la figura 6.10.

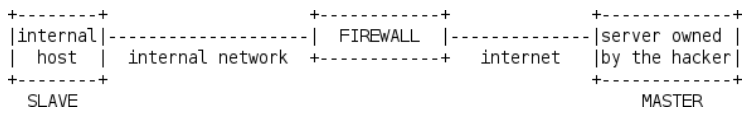


Figura 6.10: Esquema del escenario propuesto por (van Hauser 1999).

1. Naturaleza del canal

   Canal visible Solicitud sobre una conexión HTTP.

Técnica de ocultamiento Esteganografía.

Ofuscación de la carga útil de la información[6] empleando codificación Base64.

Fortaleza del ocultamiento Muy baja .

Vector para el ocultamiento Referencial.

El ocultamiento realizado en este escenario es tan básico que incluso resulta dificil clasificarlo: van Hauser apunta que es tan frecuente que un equipo conectado a red haga solicitudes HTTP que esto no lanzará ninguna alarma.

El vector se califica de *referencial* puesto que depende de ocultarse *en el contexto* de la actividad normal de un equipo en red.

Mecanismo de corrección de errores Muy básica: Al decodificar las cadenas de solicitud y respuesta, éstas van precedidas de una contraseña; citando del código, esto es "para prevenir que el administrador envíe datos raros, no es seguridad real".

Resistencia al ruido No contemplada.

Dúplex Bidireccional.

Longevidad Señalización.

Capacidad Un intercambio solicitud-respuesta por periodo (preconfigurado a 28 segundos).

2. Establecimiento y codificación

Inicialización Conexión periódica del *esclavo* (receptor) al *amo* (*emisor*).

Acuse de establecimiento Obtención de la respuesta al comando ejecutado.

Autenticación Muy débil: La repetición (en cuasi-claro) de la contraseña.

Codificación y envío de datos Codificación de la solicitud, primero a Base64, y posteriormente envío como parámetro HTTP.

Costo computacional Bajo.

El mensaje oculto o de control viaja únicamente con una codificación Base64.

3. Reconocimiento y decodificación

Identificación de mensaje oculto El *amo* (*emisor*) recibe una conexión periódica del *esclavo* (*receptor*).

Evento disparador Conexión HTTP.

[6] El autor menciona explícitamente y de forma sarcástica, "no diré que esto está cifrado, no soy Microsoft". Sin embargo, al emplear Base64 logra esconder –de los ojos del administrador humano, aunque sea– el contenido del mensaje. Siendo rigurosos, tampoco correspondería llamarle *esteganografía*; ofuscación puede ser una mejor descripción.

Autenticación Muy débil: La repetición (en cuasi-claro) de la contraseña.

Decodificación del mensaje Recepción de la cadena de parámetros HTTP, decodificación Base64 de la respuesta.

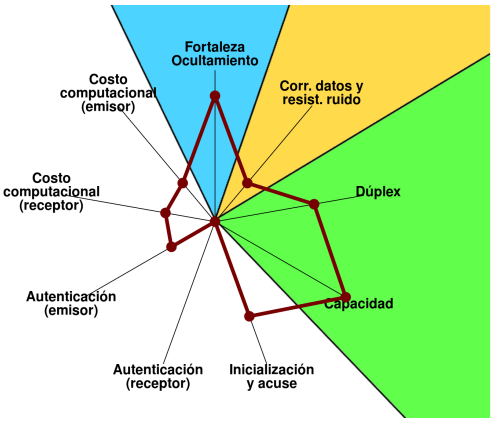
Codificación de respuestas No contempla respuesta posterior.

Costo computacional Bajo.

El mensaje oculto o de control viaja únicamente con una codificación Base64.

Este esquema, si bien hoy en día resulta francamente débil, tiene que entenderse como una herramienta de atacante aspirante hace 15 años. No tiene mucha sofisticación y muestra una fuerte falta de conocimiento de la literatura formal. Sin embargo, como lo ilustra la encuesta aplicada (véase el cuadro 4.3), una proporción no trivial de los profesionales enfrentarán al mismo problema: podría tratarse del cerca del 25 % de los casos que contestaron no hacer investigación previa en la encuesta (véase la sección 4.2).

6.1.14. El ataque a Freenode



Canal visible	Técnica ocultamiento	Vector ocultamiento	Longitud	Codificación y envío	Decodificación y respuesta
Conexiones TCP/IP estándar	Esteganografía	Conativa, fática	Sesión	Comunicación por llave compartida, RC4. Define varios comandos de 4 bytes preestableciendo acciones a realizar.	Cifrado por MD4.

En octubre del 2014 se hizo público el estudio de cómo un atacante externo logró establecer un canal oculto como puerta trasera en los servidores de la popular red de IRC orientada al software libre *Freenode* (Cannings 2014). Esta

puerta trasera combinó varias estrategias relativamente simples, logrando establecer un canal oculto que resultó dificil de detectar — Y resulta un buen caso a estudiar.

La parte de interés para el presente trabajo de este ataque consiste en dos partes: Un módulo del núcleo del sistema operativo Linux que permite a la puerta tracera permanecer oculta, y un proceso en espacio de usuario que inicia una conexión saliente y brinda al atacante el acceso al *shell* del sistema.

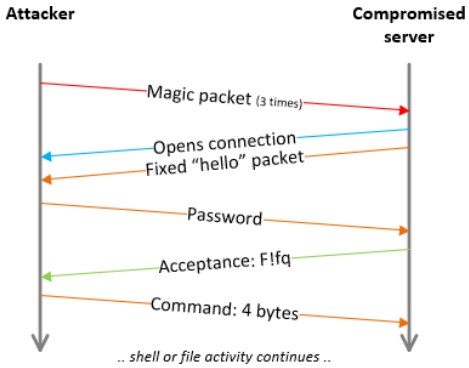


Figura 6.11: Inicialización del canal oculto descrito por (Cannings 2014).

1. Naturaleza del canal

   Canal visible Conexiones TCP/IP estándar.

   Técnica de ocultamiento Esteganografía.

   Componentes de *port knocking*, conexiones no relacionadas, carga útil cifrada.

   Fortaleza del ocultamiento Aparentaría ser *mediana*, empíricamente resultó ser *alta*.

   En la práctica, logró mantenerse un *largo tiempo* en un sitio de alto perfil, administrado por un grupo técnicamente muy capaz.

   Vector para el ocultamiento Conativa / fática.

   Las solicitudes de conexión (paquetes `SYN`) son la solicitud de iniciar una sesión TCP sobre el puerto determinado.

   Se clasifica también como *fática* porque el puerto a ser empleado para el canal a ser abierto *en sentido inverso* viaja oculto dentro de los puertos origen de los tres paquetes recibidos, como se describe a continuación.

   Mecanismo de corrección de errores Corrección no contemplada; la sesión ya establecida viaja sobre un canal TCP (la corrección de errores se realiza en capas inferiores).

   Resistencia al ruido No contemplada.

   Dúplex Bidireccional.

Longevidad Establecimiento de sesión.

Capacidad Ilimitada (establece una conexión TCP/IP estándar).

2. Establecimiento y codificación

   Inicialización El establecimiento del canal (esquematizado en la figura 6.11) comienza por lo que parecería ser un *port knocking*. El *emisor* (amo) envía un paquete inicial a *cualquier puerto* TCP del *receptor* (esclavo) tres veces, en el cual el *puerto origen* y *número de secuencia* deben sumar un valor particular (*número mágico*) no divulgado.

   El tercero de estos paquetes lleva un valor en el campo *ventana* TCP, del al cual se le resta 8192 y se obtiene el número de puerto al cual conectarse de vuelta en la dirección origen de estos paquetes (*emisor*).

   Acuse de establecimiento Una vez reconocida la inicialización, el *receptor* abre una conexión TCP al *emisor* al puerto especificado.

   Autenticación Al abrir la conexión, el *receptor* (esclavo) envía un paquete de *saludo*, a lo cual el *emisor* (amo) responde identificándose responde con una contraseña preestablecida procesada por `crypt()` (hash DES).

   Codificación y envío de datos Toda la comunicación es cifrada empleando RC4 (llave compartida). Hay varios comandos establecidos en el programa de espacio de usuario, todos ellos identificados por cuatro bytes; uno de ellos es abrir una sesión interactiva como administrador.

   Costo computacional Mediano.

   Una vez iniciado (y autenticado por un hash DES) el canal, la comunicación viaja cifrada bajo RC4.

3. Reconocimiento y decodificación

   Identificación de mensaje oculto Tras reconocer los tres *golpes de puerto* especiales (por la suma de su puerto origen y número de secuencia), el *receptor* (o *esclavo*) abre una conexión TCP al puerto indicado del *emisor* (o *amo*).

   Evento disparador Tres paquetes TCP con los valores especificados.

   Autenticación El *amo* (*emisor*) se autentica ante el *esclavo* (*receptor*), pero no hay autenticación en sentido inverso.

   Decodificación del mensaje Toda la comunicación es cifrada empleando RC4 (llave compartida).

   Codificación de respuestas Toda la comunicación es cifrada empleando RC4 (llave compartida).

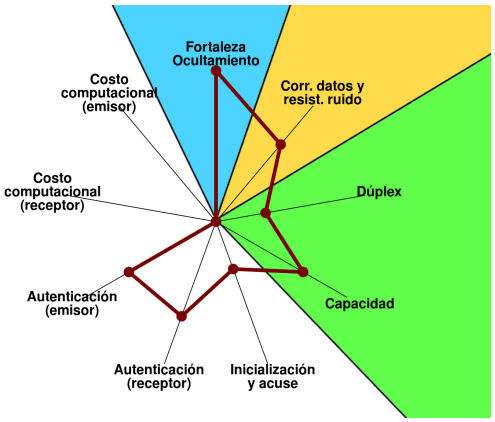
   Costo computacional Mediano.

   Una vez iniciado (y autenticado por un hash DES) el canal, la comunicación viaja cifrada bajo RC4.

El valor del análisis es por la superposición de técnicas simples que resultaron en un canal oculto sorprendentemente exitoso.[7] Citando del trabajo de Cannings,

> Mientras los mecanismos de acuerdo de sesión y de seguridad de los datos están aparentemente bien diseñados, el mecanismo de persistencia no es de ninguna manera silencioso. Este *rootkit* en particular podría haberse detectado por herramientas como *Tripwire* y el *Rootkit Hunter*.
>
> (. . .)
>
> Las técnicas empleadas están bien desarrolladas, pero no son de ninguna manera únicas. Por ejemplo, los *ganchos* sobre Netfilter se discutieron en el contexto de *rootkits* en el artículo de *Phrack* llamado *Kernel Rootkit Experiences*. Golpes de puerto similares y cifrado RC4 para ocultamiento y seguridad del transporte no son altamente sofisticados, pero sí son enfoques sólidos al desarrollar un *rootkit*.

6.2. *HttpSteg*: Función mímica basada en gramática sobre HTTP



[7] Por otro lado, podría verse como *sorprendentemente exitoso* a un canal que nunca sea descubierto. . . En tanto este fue encontrado y analizado.

Canal visible	**Técnica ocultamiento**	**Vector ocultamiento**	**Longitud**	**Codificación y envío**	**Decodificación y respuesta**
Solicitudes Web (HTTP o HTTPS)	Esteganografía (función mímica)	Emotiva, fática, conativa, referencial	Señalización	El hash y comando se procesan por un *parser inverso* que entrega la cadena que representa al árbol de parseo en cuestión	Mensaje entregado al *parser*, que construye un árbol de parseo, obtiene su representación numérica, que es el mensaje oculto.

Tras revisar y evaluar las distintas propuestas hasta aquí presentadas, y –aplicando el modelo desarrolado– determinar que ninguna de ellas cumple cabalmente con los escenarios iniciales descritos en la sección 1.5.1, esta sección presenta una propuesta de canal oculto, sintetizando los puntos que cubre el modelo. Este canal se aborda únicamente en forma de *propuesta*, y en buena medida por restricciones de tiempo no se persiguió una implementación funcional, pero claramente puede considerarse como una dirección de trabajo futuro.

A continuación se abordará la descripción, si bien muy esquemática, del canal hasta donde fue desarrollado, y se enmarca en el modelo para apuntar a sus principales fortalezas y carencias; la figura 6.12 presenta el mecanismo de construcción de la solicitud en la implementación ejemplo.

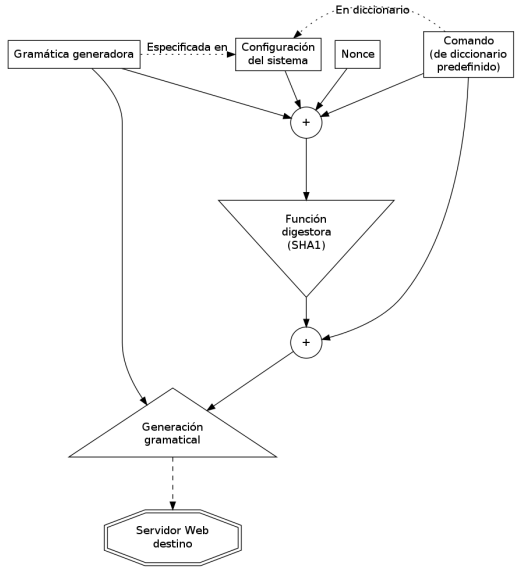


Figura 6.12: Construcción de la solicitud al servidor empleando el canal oculto propuesto.

El canal propuesto, al igual que el abordado en la sección 6.1.11, se basa en esteganografía sobre texto empleando las funciones mímica descritas en la

sección 2.3. La diferencia entre estos esquemas comienza a hacerse notar al examinar el ámbito de aplicación de uno y el otro: Como se mencionó al exponerlo, *SpamMimic* presenta parte de los bloques fundamentales para sostener un canal oculto, pero no contempla muchos de los aspectos definitorios para presentarlo como un esquema operacional: Prinicipalmente, la inicialización, autenticación e identificación de mensaje oculto quedan sujetos a como el implementador los aplique; no constituye, pues, una implementación completa de canal, sino que sólo la parte central.

El componente central del esquema propuesto sería un módulo del popular servidor Web *Apache*, dado que además de ser el servidor Web dominante desde 1996 (Netcraft 2014), ofrece una interfaz de programación que, a diferencia de los demás servidores, permite la evaluación de cualquier solicitud Web en diferentes momentos de su ciclo de vida, incluso antes de determinarse cuál será el recurso que solicita el cliente, como se aprecia en la figura 6.13. Esto lo hace ideal para transmitir información sobre un canal oculto.

El lenguaje elegido para la implementación fue *Perl*, por dos razones principales: La primera, la existencia de `mod_perl`, que expone el API completo de Apache y permite desarrollar *módulos manejadores* que se invoquen en cualquiera de los pasos del ciclo de vida de Apache, y la familia de módulos `Regexp::Grammar`, que permite la definición gradual y recursiva de gramáticas basadas en expresiones regulares.

Bajo `mod_perl`, al conectar al *módulo manejador* a la etapa `PostReadRequest` (véase la figura 6.13) se logra una mucho mayor versatilidad y posibilidad de ocultamiento de información que la que podrían presentar los esquemas abordados en la sección 2.1.2: El canal no se oculta únicamente en los parámetros de una página específica o el campo de *cookies*, sino que en la solicitud Web como un todo — Incluso en componentes que quedan típicamente ocultos al desarrollador, como el órden en que se especifican los campos o la versión del protocolo HTTP utilizado.

Además, el uso de `PostReadRequest` permite que el canal oculto sea procesado sin interferir directamente con la atención a la solicitud: Para un observador externo, la atención a la solicitud continuaría su curso de forma normal, sin evidenciar la existencia del canal oculto.

Un punto medular de esta implementación es que, al establecer una función mímica, esto no puede hacerse a ciegas — Dicho de otro modo, si el canal se define de forma que sea *parecido al tráfico normal*, es indispensable detallar qué constituye a dicho *tráfico normal*. La gramática propuesta presentaba un comportamiento modelado alrededor de la operación de una página basada en el sistema de administración de contenido *Drupal*.

Parte de la propuesta base incluye que, para facilitar la adecuación de este esquema a diferentes *mímicas*, el comportamiento fuera configurable fácilmente; se presenta a continuación, y únicamente a modo de ejemplo, la configuración que se empleó para este fin. La representación empleada en este punto es la de `Regexp::Grammar`; el esquema propuesto incluía la conversión de una notación más familiar en *Backus Naur From* de una gramática en *forma normal de Greibach*.

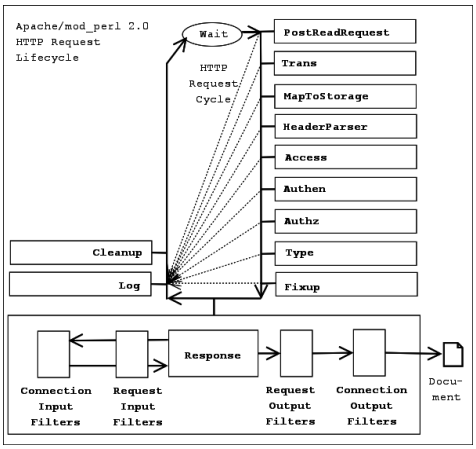


Figura 6.13: Puntos del ciclo de vida de una solicitud Web donde puede implementarse un *handler* con mod_perl. (Bekman 1996-2014)

```
qr[<grammar: DrupalSteg>
<extends: HttpSteg>

<rule: resource>
  /(?:|index.php|node/<nodeid>|admin/<module>|user|
    modules/<modulepart>-menu.css)

<token: nodeid>
  \d{1,3}

<token: module>
  (system|book|blog|filefield|aggregator|jquerymenu|user|node)

<rule: modulepart>
  <module>/<module>

<rule: PostData>
  PostArguments FormBuildId FormId

<token: FormBuildId>
  &form_build_id=form-<[Printable]>{43}

<token: FormId>
  &form_id=user_login

<token: Printable>
  (?:0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|
   v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z)
```

```
<token: PostArguments>
  <PostUsername><PostJoin><PostUserPass>

<token: PostUsername>
  name=<Username>

<token: PostJoin>
  &

<token: PostUserPass>
  pass=<Userpass>

<token: Username>
  (?:admin|gwolf)

<token: Userpass>
  (?:g%2B%23kS1s-3_MzA%7Bs)
];
```

El mensaje transmitido sería un *hash* SHA1, de 160 bits, incluyendo la información de autenticación (basada en la *posesión* de la gramática adecuada; se consideró agregar como *nonce* la hora del sistema o el mantenimiento de estado en conjunto con un esquema de *one-time pad*), además del comando especificado de entre los que forman parte de un diccionario definido; esta estrategia para limitar los comandos se refleja en el inciso *e* de la pregunta 3 de la encuesta (véase la sección 4.2).

Sin entrar demasiado en detalles con la implementación, una de las principales dificultades encontradas radicó en la complejidad del *parseo* de las solicitudes: Si el parser detectaba una cadena válida (esto es, si la solicitud incluía un mensaje sobre el canal oculto), su conversión a un árbol y la reconstrucción de la cadena original se realizaban en un tiempo aceptable. Sin embargo, dada la naturaleza del motor de expresiones regulares y la gramática resultante particular, una solicitud que *no* llevara canal oculto resultaba demasiado compleja. Si bien su procesamiento podía enviarse al fondo para no interferir con la respuesta a las solicitudes, en un servidor Web con tráfico real (necesario para que resulte un canal oculto) esto llevaría rápidamente a un agotamiento de recursos del sistema.

Habiendo expuesto el funcionamiento básico del canal propuesto, a continuación sigue –al igual que con todos los canales anteriores abordados– la aplicación del modelo.

1. Naturaleza del canal

   Canal visible Solicitudes Web (HTTP o HTTPS)

   Técnica de ocultamiento Esteganográfica.

   El mensaje visible es generado eligiendo un valor que represente a la cadena oculta a transmitir, y construido mediante la *caminata* del árbol de paseo inverso de una gramática regular.

Fortaleza del ocultamiento Alta.

El mensaje visible no guarda correspondencia alguna (ni siquiera longitud del texto generado) con el oculto, y las modificaciones incluso más pequeñas a la gramática generadora cmabian por completo los mensajes.

Vector para el ocultamiento Emotiva, fática, conativa, referencial.

El mensaje cubierta incluye componentes que van sobre diferentes vectores, desde presentar al iniciador de la solicitud por medio de campos de identificación (emotivo), presentar una solicitud por medio de los *verbos* HTTP (conativo), información de contexto para la solicitud (referencial), o información acerca del tipo de datos aceptable (fático).

Mecanismo de corrección de errores El valor diccionario del comando a transmitir es incluido en el hash de autenticación e incluido generación del árbol; se verifica la autenticación antes de realizar la acción indicada.

Cabe apuntar que este mecanismo constituye *detección*, no *corrección* de errores.

Resistencia al ruido No contemplado.

Dúplex Unidireccional.

Longevidad Señalización.

Capacidad La gramática presentada ofrece del órden de 60 bits de entropía. No se realizaron pruebas para verificar la profundidad homogénea del árbol. Se requiere aún trabajo sobre esta gramática para triplicar el espacio disponible para la transmisión del mensaje planteado, un hash de 160 bits.

2. Establecimiento y codificación

Inicialización Envío de una solicitud Web que cumpla con la gramática.

Acuse de establecimiento No contemplado.

Autenticación Hash incluyendo la configuración específica del sistema y otros posibles secretos.

Codificación y envío de datos Se concatena el hash autenticador con el comando solicitado, y se aplica un *parseo* inverso con la gramática, obteniendo la cadena que dicho número generaría al ser *parseado*. Se transmite la cadena obtenida.

Costo computacional Alto.

La complejidad de recorrer la gramática para crear el mensaje correspondiente puede ser muy alta.

3. Reconocimiento y decodificación

Identificación de mensaje oculto Todas las solicitudes recibidas por el servidor Web son procesadas por un módulo que busca correspondencias con la gramática generadora.

Evento disparador Un mensaje que cumpla con la gramática procede a la revisión del hash autenticador y, si procede, la ejecución del comando.

Autenticación El hash se verifica contra la configuración (que incluirá los *secretos* a verificar).

Decodificación del mensaje El mensaje recibido es generado al *parser*, se construye un árbol de parseo, obteniendo su representación numérica.

Codificación de respuestas No contemplado.

Costo computacional Muy alto.

Además del costo computacional del *parser*, por la mecánica de *backtracking* normalmente utilizada, la sobrecarga para detectar mensajes que *no* lleven un mensaje oculto puede ser superior al de cuando sí lo hay.

6.3. Resumen del capítulo

Este capítulo presentó la aplicación del modelo desarrollado en el capítulo 5 a los distintos canales ocultos presentados en los capítulos 3 y 2. Hubo varias razones para aplicar el modelo a todos estos esquemas:

- Al ir considerando los distintos canales fueron surgiendo necesidades de representación que hicieron ver algunas debilidades en el planteamiento original, y llevaron a madurar al modelo antes de su presentación formal.
- A pesar de que, como se explicitó al principio del capítulo, muchos de estos esquemas hacen un planteamiento incompleto ante lo requerido por el modelo (particularmente aquellos canales que son presentados meramente en forma conceptual y como parte de un trabajo teórico/académico), presentarlos junto con canales provenientes de la literatura informal (refiérase a la sección 4.4 para una mayor discusión al respecto) permite *unir los puntos* y plantear cómo podrían complementarse para implementar las partes faltantes.
- Y al plantearse, en contraposición, las claras debilidades en distintos ámbitos de los canales presentados por vía informal, resulta claro cómo muchos de ellos podrían beneficiarse de aspectos de las implementaciones formales.

El desarrollo central del capítulo aborda canal por canal, de forma secuencial. El presentar el reporte de cada uno de los canales permite, además, un vistazo comparativo que permite ubicar a las propuestas salientes, que presentan a uno de los puntos claramente distinto de las demás propuestas. Se sugiere referirse

a esta sección (o realizar un ejercicio similar) a cualquier implementador que busque combinar características para desarrollar una nueva propuesta.

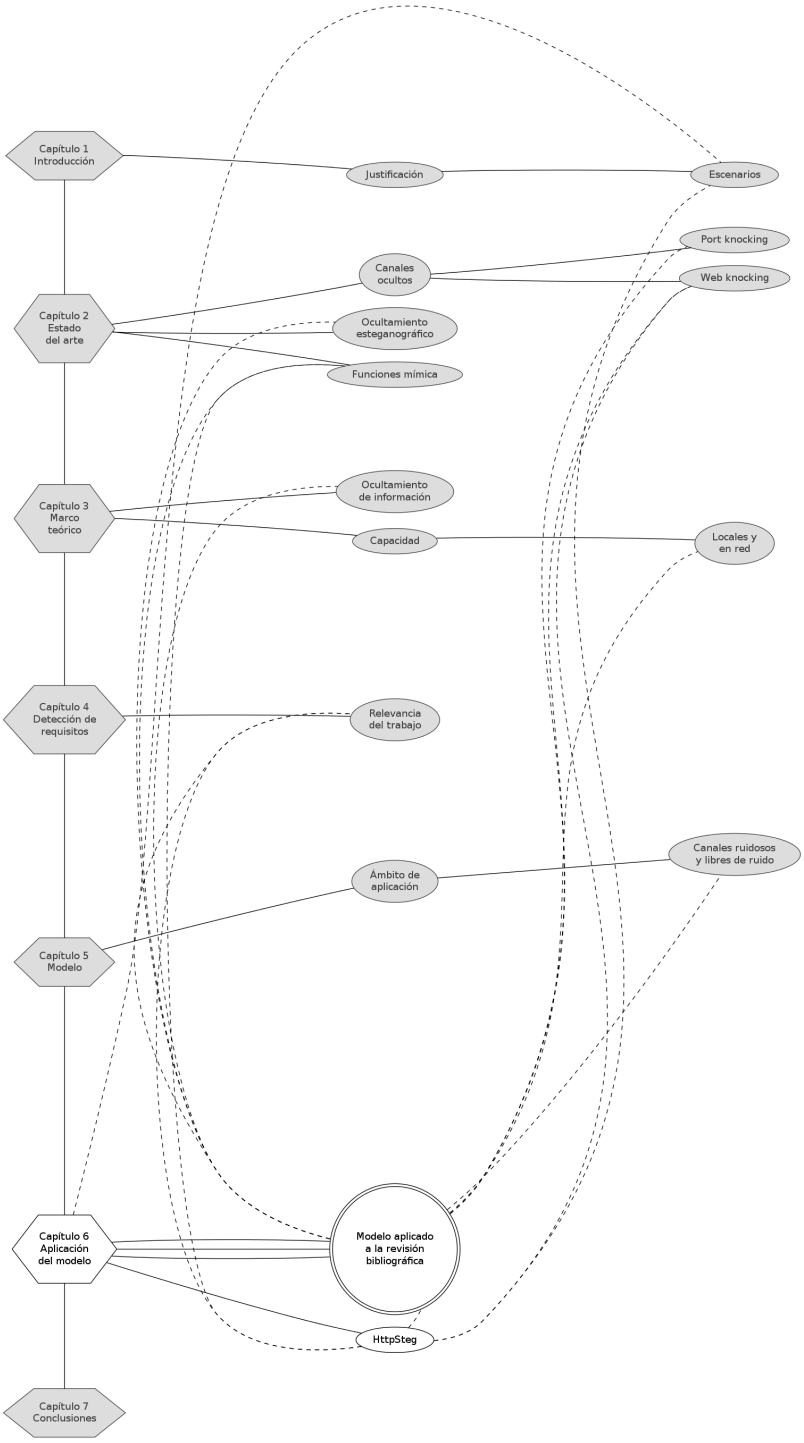


Figura 6.14: Relaciones conceptuales: Temas abordados en el capítulo 6

Capítulo 7

Conclusiones

7.1. Conclusiones generales

El estudio y desarrollo de los canales ocultos amerita que en la literatura académica se le dedique una mayor atención; como fue expuesto, no sólo es un área con gran riqueza de implementaciones, sino con una clara justificación, en lo técnico y en lo social.

El modelo aquí presentado es, sin duda, perfectible. La contribución que se pretende hacer es en dos sentidos:

1. Al brindar un esquema amplio y sistematizado, facilitar a los administradores de sistemas que deseen emplear un esquema ya existente la comparación de implementaciones.

2. Para los investigadores o desarrolladores que busquen crear un nuevo esquema de comunicación sobre canal oculto o perfeccionar uno existente, presentar de forma clara y sistematizada los principales puntos que deben ser considerados, avalados no únicamente por una comparativa de implementaciones preexistentes, sino por las opiniones de una comunidad de profesionales.

7.2. Trabajo a futuro

Queda claro que el tema abordado da para un análisis mucho más profundo, y para muchos desarrollos derivados. El tema, como se vio, no es novedoso, pero dada la creciente importancia y vulnerabilidad de las redes, muy probablemente sea abordado cada vez con mayor necesidad.

Un punto que complicó el desarrollo del modelo, y podría ser abordado en futuros refinamientos del presente trabajo, es la dificultad de cuantificar de forma inambigua los cualificadores presentados, particularmente la *resistencia al ruido* y –como se detalla en la sección 2.2.1– la *fortaleza del ocultamiento*.

El modelo propuesto debería poder ser *validado* por terceros. Para hacer esto, se sugiere solicitar a un grupo de administradores de sistemas y desarrolladores (esto es, el público objetivo del trabajo) analizar implementaciones o propuestas de canales ocultos, sean los abordados por este trabajo u otros. Se espera, naturalmente, que los resultados de dichos análisis presenten la mayor cercanía posible.

Del mismo modo, para sustentar más claramente los resultados presentados, la encuesta descrita en el capítulo 4 debería ser revisada y vuelta a aplicar, prestando un mayor cuidado a su validez y objetividad.

La cuantificación de valores sugerida en el capítulo 5 para facilitar el uso comparativo del modelo se realizó empleando métodos y líneas de corte que no fueron formalmente validadas. Queda como trabajo pendiente validarla o corregirla.

En las etapas finales del desarrollo de este trabajo, resultó claro que la *prueba de fuego* para el modelo es su aplicación a canales ocultos de distintas naturalezas; una aplicación más extensa seguramente ayudará a ubicar posibles carencias del mismo.

Y, claro está, continuar con las ideas hilvanadas en la propuesta de la sección 6.2: terminar con la implementación de este canal desarrollado basado en el modelo propuesto, actualmente existente sólo de forma descriptiva, y evaluar si logra hacer una aportación apta al campo.

Bibliografía

Allar, Jared (2012). *SYSRET 64-bit operating system privilege escalation vulnerability on Intel CPU hardware*. Inf. téc. CERT. URL: http://www.kb.cert.org/vuls/id/649219 (vid. pág. 13).

Arnbak, Axel y col. (2014). "Security collapse in the HTTPS market". En: *Communications of the ACM* 57.10, págs. 47-55. DOI: 10.1145/2660574. URL: http://cacm.acm.org/magazines/2014/9/178779-security-collapse-in-the-https-market/fulltext (vid. pág. 12).

Bailey, Karen y Kevin Curran (2006). "An Evaluation of Image Based Steganography Methods". En: *Multimedia Tools and Applications* 30.1, págs. 55-88. URL: http://link.springer.com/article/10.1007/s11042-006-0008-4 (vid. pág. 23).

Bauer, Matthias (oct. de 2003). "New Covert Channels in HTTP: Adding Unwitting Web Browsers to Anonymity Sets". En: *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2003)*. Washington, DC, USA. URL: http://www.gray-world.net/papers/bauer_wpes2003.pdf (vid. págs. 23, 107, 110).

Bekman, Stas (1996-2014). "mod_perl: HTTP Handlers". En: *Mod_perl documentation*. URL: https://perl.apache.org/docs/2.0/user/handlers/http.html (vid. pág. 119).

Bello, Luciano (2008). *CWKPF Cuasi-Web Knocking para Packet Filter*. URL: http://www.lucianobello.com.ar/webknocking/index.html (vid. pág. 20).

Bo, Xu, Wa Jia-zhen y Peng De-Yun (2007). "Practical Protocol Steganography: Hiding data in IP header". En: *Proceedings of the First Asia International Conference on Modelling & Simulation (AMS'07)*. Ed. por IEEE Computer Society, págs. 584-588. URL: http://www.computer.org/csdl/proceedings/ams/2007/2845/00/28450584-abs.html (vid. págs. 19, 61).

Briganti, Domenico (2012). *Port Knocking via web, Web Knocking!* URL: http://tipsaboutmywork.blogspot.com/2012/02/port-knocking-via-web-web-knocking.html (vid. págs. 20, 103).

Burnett, Mark (2005). *Perfect Passwords: Selection, protection, authentication*. Syngress. ISBN: 978-1597490412 (vid. pág. 38).

Burnett, Mark (2011). *10,000 Top Passwords*. URL: https://xato.net/passwords/more-top-worst-passwords/ (vid. pág. 37).

Burr, William E. y col. (2013). *Electronic Authentication Guideline*. NIST Special Publication 800-63-2. National Institute of Standards y Technology. URL: http://dx.doi.org/10.6028/NIST.SP.800-63-2 (vid. pág. 38).

Cabuk, Serdar (2006). "Network Cover Channels: Design, Analysis, Detection and Elimination". Tesis doct. Purdue University. URL: http://spaf.cerias.purdue.edu/Students/cabuk_thesis.pdf (vid. págs. 31, 61, 78).

Cannings, David (oct. de 2014). *Analysis of the Linux backdoor used in freenode IRC network compromise*. Consulta: Octubre 2014. NCC Group. URL: https://www.nccgroup.com/en/blog/2014/10/analysis-of-the-linux-backdoor-used-in-freenode-irc-network-compromise/ (vid. págs. 19, 113-114).

Caviglione, Luca y Wojciech Mazurczyk (2014). "How to covertly leak data from iOS?" En: *arXiv preprint arXiv:1411.3000*. URL: http://arxiv.org/abs/1411.3000 (vid. pág. 87).

CCITT (1990). *V.42bis : Data compression procedures for data circuit-terminating equipment (DCE) using error correction procedures*. International Telecommunications Union. URL: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-V.42bis-199001-I!!PDF-E&type=items (vid. pág. 85).

Central Intelligence Agency (2014). *The CIA World Factbook*. Consulta: Marzo 2014. URL: https://www.cia.gov/library/publications/the-world-factbook/rankorder/2184rank.html (vid. pág. 12).

CERT, Software Engineering Institute (1996). *TCP SYN Flooding and IP Spoofing attacks*. URL: https://www.cert.org/historical/advisories/CA-1996-21.cfm (vid. págs. 91, 93).

— (1997). *Denial of Service Attacks*. URL: https://www.cert.org/historical/tech_tips/denial_of_service.cfm (vid. pág. 12).

Chaum, David L. (feb. de 1981). "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". En: *Commun. ACM* 24.2, págs. 84-90. ISSN: 0001-0782. DOI: 10.1145/358549.358563. URL: http://doi.acm.org/10.1145/358549.358563 (vid. pág. 107).

Clark, Michael (nov. de 2001). *Virtual Honeynets*. SecurityFocus. URL: http://www.symantec.com/connect/articles/virtual-honeynets (vid. pág. 13).

Codenomicon Defensics (2014). *Heartbleed Bug*. URL: http://heartbleed.com/ (vid. pág. 12).

Codr, Jessica (2009). *Unseen: An Overview of Steganography and Presentation of Associated Java Application C-Hide*. URL: http://www.cse.wustl.edu/~jain/cse571-09/ftp/stegano/index.html (vid. págs. 20, 34-35).

Crenshaw, Adrian (2012). *Unicode Text Steganography Encoders/Decoders*. irongeek.com. URL: http://www.irongeek.com/i.php?page=

security/unicode-steganography-homoglyph-encoder (vid. pág. 21).

Croarkin, Carroll, ed. (2003-2012). *NIST/SEMATECH e-Handbook of Statistical Methods*. URL: http://www.itl.nist.gov/div898/handbook/ (vid. pág. 71).

Dang, Quynh (2012). *Recommendation for Applications Using Approved Hash Algorithms*. NIST Special Publication 800-63-2. National Institute of Standards y Technology. URL: http://csrc.nist.gov/publications/nistpubs/800-107-rev1/sp800-107-rev1.pdf (vid. pág. 39).

deGraaf, Reinderd Gordon Nathan (2007). "Enhancing Firewalls: Conveying User and Application Identification to Network Firewalls". Tesis de lic. The University of Calgary. URL: http://ciphertext.info/papers/thesis-degraaf.pdf (vid. pág. 19).

Dierks, Tim y Eric Rescorla (2008). *RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2*. URL: https://www.rfc-editor.org/rfc/rfc5246.txt (vid. pág. 39).

Diffie, Whitfield y Martin Hellman (1976). "New directions in cryptography". En: *IEEE Transactions on Information Theory* 22, págs. 644-654 (vid. pág. 39).

Eastlake, Donald E., Jeffrey I. Schiller y Steve Crocker (2008). *RFC 4086: Randomness Requirements for Security*. URL: https://www.rfc-editor.org/rfc/rfc4086.txt (vid. pág. 38).

Greenwald, Glenn, Ewen MacAskill y Laura Poitras (2013). *Edward Snowden: The whistleblower behind the NSA surveillance revelations*. URL: http://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance (vid. pág. 13).

Handel, Theodore y Maxwell Sandford (1996). "Hiding Data in the OSI Network Model". En: *Proceedings of the First International Workshop on Information Hiding*. London, UK, UK: Springer-Verlag, págs. 23-38. ISBN: 3-540-61996-8. URL: http://faculty.kfupm.edu.sa/COE/mimam/Papers/96%20Hiding%20Data%20in%20the%20OSI%20Network%20Model.pdf (vid. págs. 31, 36, 84-85, 87-88).

Harvey, Charles (2013). *Stegospam: Hiding messages in spam for fun and mischief*. URL: http://charlieharvey.org.uk/page/stegospam_steganography_with_perl_and_spam (vid. pág. 22).

Hébert, Louis (2011). "The Functions of Language". En: *Signo [online]*. URL: http://www.signosemio.com/jakobson/functions-of-language.asp (vid. pág. 29).

Hernández Sampieri, Roberto, Carlos Fernández Collado y Pilar Baptista Lucio (2006). *Metodología de la investigación*. 4.ª ed. McGraw Hill Interamericana. ISBN: 970-10-5753-8 (vid. págs. 8, 24, 42, 57).

Internet Systems Consortium (2014). *ISC Domain Survey*. Consulta: Marzo 2014. URL: https://www.isc.org/services/survey/ (vid. pág. 12).

Izquierdo Manzanares, Antonio y col. (2005). "Attacks on port knocking authentication mechanism". En: *Computational Science and its implications — ICCSSA 2005 Lecture Notes in Computer Science*. URL: http://

pdf.aminer.org/000/291/623/attacks_on_port_knocking_authentication_mechanism.pdf (vid. págs. 18, 96).

Johnson, Neil F. (1995). *Steganography*. Inf. téc. Center for Secure Information Systems, George Mason University. URL: http://www.jjtc.com/pub/tr_95_11_nfj/ (vid. pág. 20).

Kahn, David (1967). *The codebreakers: The story of secret writing*. Macmillan. ISBN: 0-684-83130-9 (vid. págs. 32, 63).

Kaminsky, Dan (2004). *MD5 To Be Considered Harmful Someday*. Cryptology ePrint Archive, Report 2004/357. URL: http://crppit.epfl.ch/documentation/Hash_Function/Examples/Code_Project/Documentation/md5_someday.pdf (vid. pág. 103).

Kortchinsky, Kostya (2009). *Cloudburst: Hacking 3D (and breaking out of VMware)*. URL: https://www.blackhat.com/presentations/bh-usa-09/KORTCHINSKY/BHUSA09-Kortchinsky-Cloudburst-SLIDES.pdf (vid. pág. 13).

Krzywinsky, Martin (jun. de 2003). "Port Knocking". En: *Linux Journal*. URL: http://www.linuxjournal.com/article/6811 (vid. págs. 18, 90).

Kumar, Pramod (oct. de 2014). "Evaluation criteria of stego system". En: *Proceedings of 10th IRF International Conference*. ISBN: 978-93-84209-56-8. URL: http://www.iraj.in/up_proc/pdf/106-141388611860-65.pdf (vid. pág. 23).

Kundur, Deepa y Kamran Ahsan (2003). "Practical Internet steganography: data hiding in IP". En: *Proceedings of the Texas workshop on security of information systems*. Vol. 2. URL: http://vanilla47.com/PDFs/Cryptography/Steganography/Practical%20Internet%20Steganography%20Data%20Hiding%20in%20IP.pdf (vid. págs. 19, 61, 98-99).

Lampson, Butler W. (1973). "A note on the confinement problem". En: *Communications of the ACM* 16.10, págs. 613-615. URL: http://dl.acm.org/citation.cfm?id=362375.362389&coll=portal&dl=ACM (vid. págs. 17, 30, 75-76).

Lebelt, Stefan (2005). *Webknocking: knock different*. URL: http://lebelt.info/old/?item=webknocking_en (vid. págs. 19, 101).

Li, Bin y col. (abr. de 2011). "A Survey on Image Steganography and Steganalysis". En: *Journal of Information Hiding and Multimedia Signal Processing* 2.2. ISSN: 2073-4212 (vid. pág. 23).

Marlinspike, Moxie (2009). *New tricks for defeating SSL in practice*. Black Hat DC 2009. URL: https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf (vid. pág. 12).

McKellar, Dave (2000-2014). *SpamMimic*. URL: http://www.spammimic.com/ (vid. págs. 22, 104).

Millen, Jonathan (jun. de 1989). "Finite-state noiseless covert channels". En: *Computer Security Foundations Workshop II, 1989., Proceedings of the*, págs. 81-86. DOI: 10.1109/CSFW.1989.40590. URL: http://

ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=40590 (vid. pág. 34).

Millen, Jonathan (1999). "20 years of Covert Channel Modelling and Analysis". En: *IEEE Symposium on Security and Privacy*. DOI: 10.1109/SECPRI.1999.766906 (vid. pág. 30).

Morris, Robert y Ken Thompson (1979). "Password Security: A Case History". En: *Communications of the ACM* 22, págs. 594-597. URL: http://cm.bell-labs.com/cm/cs/who/dmr/passwd.ps (vid. págs. 37, 39).

Murdoch, Steven J y Stephen Lewis (2005). "Embedding covert channels into TCP/IP". En: *Information Hiding*. Springer, págs. 247-261. URL: http://link.springer.com/chapter/10.1007/11558859_19 (vid. págs. 19, 61).

National Computer Security Center (1993). *A guide to understanding covert channel analysis of trusted systems*. The Rainbow Books NCSC-TG-030. National Computer Security Center. URL: http://fas.org/irp/nsa/rainbow/tg030.htm (vid. pág. 17).

Negroni, Andrea (2005). *Distributed Denial of Service*. Cisco Systems. URL: http://www.cisco.com/web/IT/events/pdf/iin2005/distributed_denial.pdf (vid. pág. 11).

Netcraft (2014). *Web Server Survey*. Consulta: Abril 2014. URL: http://news.netcraft.com/archives/category/web-server-survey/ (vid. pág. 118).

NIST (2012). *Secure Hash Standard (SHS)*. Federal Information Processing Standards Publication FIPS-180-4. National Institute of Standards y Technology. URL: http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf (vid. pág. 39).

Olaniyi, O. M. y col. (2014). "Performance Evaluation of modified Stegano-Cryptographic model for Secured E-voting". En: 3.1. ISSN: 2320-2610. URL: http://www.warse.org/pdfs/ijmcis01312014.pdf (vid. pág. 23).

Pevný, Tomáš, Jessica Fridrich y Andrew D. Ker (abr. de 2012). "From Blind to Quantitative Steganalysis". En: *IEEE Transactions on Information Forensics and Security* 7.2. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6081932 (vid. pág. 24).

Provos, Niels (2003). "A Virtual Honeypot Framework". En: *In Proceedings of the 13th USENIX Security Symposium*, págs. 1-14. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.9516&rep=rep1&type=pdf (vid. pág. 13).

Rash, Michael (2007). "Single Packet Authorization". En: *Linux Journal*. URL: http://www.linuxjournal.com/article/9565 (vid. págs. 18, 65, 96).

— (2007–2014). *Single Packet Authorization: A Comprehensive Guide to Service Concealment with fwknop*. Cypherdine. URL: http://www.cipherdyne.org/fwknop/docs/fwknop-tutorial.html (vid. págs. 18, 96).

Reed, Michael G., Paul F Syverson y David M. Goldschlag (dic. de 1996). En: *12th Annual Computer Security Applications Conference*. ISBN: 0-8186-7606-X. DOI: 10.1109/CSAC.1996.569678. URL: http://ieeexplore.

ieee.org/xpls/abs_all.jsp?arnumber=569678&tag=1 (vid. pág. 110).

Rogaway, Phillip (2004). "Nonce-Based Symmetric Encryption". En: *Proc. FSE 2004, volume 3017 of LNCS*. Springer, págs. 348-359. URL: http://web.cs.ucdavis.edu/~rogaway/papers/nonce.pdf (vid. pág. 39).

Salomon, David (2003). *Data Privacy and Security: Encryption and Information Hiding*. Springer Science y Business Media. ISBN: 978-0387003115 (vid. pág. 34).

Schaefer, Marvin y col. (1977). "Program Confinement in KVM/370". En: *Proceedings of the 1977 Annual Conference*. ACM '77. New York, NY, USA: ACM, págs. 404-410. ISBN: 978-1-4503-2308-6. DOI: 10.1145/800179.1124633. URL: http://doi.acm.org/10.1145/800179.1124633 (vid. págs. 17, 31).

Schneier, Bruce (2005). *Cryptanalysis of SHA-1*. URL: https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html (vid. pág. 39).

Schwartz, Matthew J. (2012). *New virtualization vulnerability allows escape to hypervisor attacks*. URL: http://www.darkreading.com/risk-management/d/d-id/1104823 (vid. pág. 13).

Sehgal, Nancy y Ajay Goel (2014). "Evolution in Image Steganography". En: *International Journal of Information and Computation Technology* 4.12, págs. 1221-1227 (vid. págs. 34-35, 64).

Shane, Scott y Andrew W. Lehren (2010). *Leaked Cables Offer Raw Look at U.S. Diplomacy*. URL: http://www.nytimes.com/2010/11/29/world/29cables.html?_r=0 (vid. pág. 13).

Shannon, Claude E. (jul. de 1948). "A mathematical theory of communication". En: *Bell System Technical Journal* 27, págs. 379-423, 623-656. URL: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf (vid. págs. 28, 34, 60-61).

Shieh, Shiuh-pyng (1996). "Estimating and measuring covert channel bandwidth in multilevel secure operating systems". En: *Journal of Information Science and Engineering* 15, págs. 91-106 (vid. pág. 34).

Simmons, Gustavus J. (1983). "The Prisoners' Problem and the Subliminal Channel". En: *Advances in Cryptology, Proceedings of CRYPTO '83* (vid. pág. 31).

— (1985). "The Subliminal Channel and Digital Signatures". En: *Advances in Cryptology*. Vol. 209. Lecture Notes in Computer Science, págs. 364-378. DOI: 10.1007/3-540-39757-4_25. URL: http://link.springer.com/chapter/10.1007%2F3-540-39757-4_25 (vid. págs. 31, 81).

Smart, Nigel (2012). *ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012)*. Inf. téc. 7th Framework Programme, European Commission. URL: http://www.ecrypt.eu.org/documents/D.SPA.20.pdf (vid. págs. 38-39).

Spitzner, Lance y Marty Roesch (ene. de 2001). SecurityFocus. URL: http://www.symantec.com/connect/articles/value-honeypots-part-one-definitions-and-values-honeypots (vid. pág. 13).

Stribling, Jeremy, Max Krohn y Dan Aguayo (2005). *Scigen-an automatic cs paper generator*. URL: http://pdos.csail.mit.edu/scigen/ (vid. pág. 25).

Tanase, Matthew (2002). *Sniffers: What they are and how to protect yourself*. Symantec. URL: http://www.symantec.com/connect/articles/sniffers-what-they-are-and-how-protect-yourself (vid. pág. 12).

Telegeography (2012). *2012 Global Internet Map*. Consulta: Marzo 2014. URL: http://www.telegeography.com/telecom-maps/global-internet-map/ (vid. pág. 13).

TextHide (1999). Website. URL: http://www.texthide.com/ (vid. pág. 22).

Trusted Computer System Evaluation Criteria (1985). Department of Defense. URL: http://csrc.nist.gov/publications/secpubs/rainbow/std001.txt (vid. pág. 36).

Tsai, C.-R. y V.D. Gligor (abr. de 1988). "A bandwidth computation model for covert storage channels and its applications". En: *Security and Privacy, 1988. Proceedings., 1988 IEEE Symposium on*, págs. 108-121. DOI: 10.1109/SECPRI.1988.8103. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8103 (vid. pág. 36).

van Hauser (1999). "Placing backdoors through firewalls". En: *The Hacker's Choice*. URL: https://www.thc.org/papers/fw-backd.htm (vid. págs. 23, 111).

Violet, Imperial (2014). *Apple's SSL/TLS bug*. URL: https://www.imperialviolet.org/2014/02/22/applebug.html (vid. pág. 12).

Wang, Zhenghong y RubyB. Lee (2005). "New Constructive Approach to Covert Channel Modeling and Channel Capacity Estimation". En: *Information Security*. Ed. por Jianying Zhou y col. Vol. 3650. Lecture Notes in Computer Science. Springer Berlin Heidelberg, págs. 498-505. ISBN: 978-3-540-29001-8. DOI: 10.1007/11556992_37. URL: http://dx.doi.org/10.1007/11556992_37 (vid. págs. 33-34, 63-64).

Waugh, Linda R. (1980). "The poetic function in the theory of Roman Jakobson". En: *Poetics Today* 2.1a, págs. 57-82. URL: http://www.jstor.org/stable/1772352 (vid. pág. 29).

Wayner, Peter (1991). *Mimic Functions: The Manual*. URL: http://www.nic.funet.fi/pub/crypt/old/mimic/Mimic-Manual.txt (vid. pág. 25).

— (1999). *Mimc Functions and Tractability*. URL: https://www.solver.io/external/cryptography/applied-crypto/mimic-two.ps.gz (vid. pág. 25).

— (2009). *Disappearing Cryptography*. 3.ª ed. Morgan Kaufmann. ISBN: 978-0-12-374479-1 (vid. págs. 23, 25, 36, 104).

Weimer, Florian (2008). *DSA-1571-1 OpenSSL — Predictable random number generator*. Inf. téc. Proyecto Debian. URL: https://www.debian.org/security/2008/dsa-1571 (vid. pág. 12).

Winstein, Keith y Hari Balakrishnan (2012). "Mosh: An Interactive Remote Shell for Mobile Clients". En: *2012 USENIX Annual Technical Conference*.

USENIX. URL: https://mosh.mit.edu/mosh-paper-draft.pdf (vid. pág. 47).

Zini, Enrico (2005-2009). *Debian polygen grammars*. Debian project. URL: https://people.debian.org/~enrico/polygen-debian/ (vid. pág. 25).