Resolución de consultas anónimas sobre DNS

Joaquín García-Alfaro[†,‡] y Sergio Castillo-Pérez[†]

***Resumen*—La utilización de DNS como mecanismo base de nuevos servicios telemáticos basados en resolución de nombres puede suponer riesgos en seguridad y privacidad. La información intercambiada entre clientes y servidores viaja sin ningún tipo de protección. Dicha información puede ser capturada por *malware* o servidores de acceso deshonestos y acabar siendo vendida para su utilización en técnicas de *spamming* o de minería de datos. Motivados por la utilización de DNS en el servicio ENUM y su proceso de traducción de números de teléfono tradicionales para su uso en redes de voz IP, analizamos en este trabajo dos propuestas para incorporar anonimato en consultas de DNS. La primera propuesta se basa en redirigir consultas y respuestas de DNS a través de infraestructuras de anonimato de baja latencia. La segunda propuesta plantea la introducción de ruido en consultas de DNS. Una carencia presente en el protocolo clásico de DNS es la falta de garantías en cuanto a integridad y autenticidad del servicio. Esta carencia puede ser solventada mediante la utilización de las extensiones de seguridad para DNS conocidas como DNSSEC, e incorporada en las dos alternativas aquí recogidas. Evaluamos en este trabajo las limitaciones y beneficios introducidos por cada una de las propuestas, así como el impacto en el rendimiento introducido por el uso de consultas basadas en DNSSEC.**

***Palabras Clave*—Seguridad en redes informáticas (*Computer network security)*, Privacidad (*Privacy*), Anonimato (*Anonymity*).**

I. Introducción

La principal motivación de nuestro trabajo proviene de las implicaciones en seguridad y privacidad que tiene la utilización del protocolo DNS como tecnología base de nuevos servicios telemáticos como, por ejemplo, el servicio ENUM (*tElephone NUmber Mapping*). ENUM engloba un conjunto de protocolos utilizado en servicios de VoIP (voz sobre IP). Una de las principales características de ENUM es la traducción de números de teléfono tradicionales asociados al sistema E.164, a URIs de proveedores de telefonía IP, así como otros servicios asociados a Internet (por ejemplo, correo electrónico, Web, etc.). En esta sección, detallamos algunas de las características de este servicio, así como implicaciones de seguridad y privacidad que tiene la utilización del protocolo DNS como tecnología base de ENUM.

A. ENUM

El servicio ENUM agrupa un conjunto de protocolos utilizados en VoIP para la comunicación de teléfonos basados en el sistema de telefonía tradicional E.164 a través de la red IP de Internet. ENUM es una propuesta del IETF (*Internet Engineering Task Force (IETF)*. A través de un proceso de traducción de números de teléfono asociados con el sistema E.164 (utilizado hoy en día por la mayoría de operadores de telefonía tradicionales), ENUM reutiliza la infraestructura ya existente de DNS y sus servicios de delegación a través de registros de tipo NS (NameServer). La información de ENUM es almacenada en registros de tipo NAPTR (*Naming Authority Pointer*) [1]. A largo plazo, ENUM debería convertirse en una alternativa descentralizada del sistema E.164. Para una visión más completa y detallada del servicio ENUM, animamos al lector a consultar [2].

A grandes rasgos, el servicio de traducción de ENUM es meramente un proceso de conversión de números de teléfono tales como +34-012345678, a cadenas de texto asociadas a identificadores de VoIP, direcciones de correo, etc. Para ello, ENUM propone la traducción del número E.164 al completo (incluyendo el identificador de país) a un identificador URI (*Uniform Resource Identifier*) asociado a DNS a través de la siguiente convención: (1) símbolos especiales como '+' y '-' son eliminados (e.g., +34-012345678 se convierte en 34012345678); (2) la cadena de dígitos resultantes es invertida de izquierda a derecha (e.g., 87654321043); (3) un símbolo '.' es intercalado entre cada dos dígitos (e.g., 8.7.6.5.4.3.2.1.0.4.3); (4) el dominio .e164.arpa es concatenado a la cadena resultante (e.g., 8.7.6.5.4.3.2.1.0.4.3.e164.arpa). La cadena final puede así ser utilizada por consultas DNS tradicionales. En el lado del servidor, el URI de cada número de teléfono registrado por ENUM es almacenado junto a información asociada (por ejemplo, propietarios o usuarios de dicho teléfono). Dicha información es almacenada en registros de tipo NAPTR. La estructura de estos registros permite a ENUM un amplio espacio de almacenamiento y gran flexibilidad en su uso (por ejemplo, a través del uso de expresiones regulares).

Veamos a continuación la utilización de ENUM para la obtención de información vinculada al número de teléfono +34-012345678 asociado a un usuario U_1. Un usuario U_2 desea contactar con el usuario U_1. En primer lugar, U_2 convierte el número de teléfono al formato 8.7.6.5.4.3.2.1.0.4.3.e164.arpa y utiliza la cadena anterior para realizar una consulta DNS de tipo NAPTR a través de la herramienta *dig*:

```
dig @$NS -t NAPTR 8.7.6.5.4.3.2.1.0.4.3.e164.arpa
```

Como resultado, U_2 obtiene la siguiente información:

Order	Pref.	Flags	Service	Regexp.	Rep.
100	10	u	sip+E2U	!^.*$!sip:u1@sip.es!'	.
101	10	u	mailto+E2U	!^.*$!mailto:u1@mail.es!	.
102	10	u	http+E2U	!^.*$!http://www.u1.es!	.
103	10	u	tel+E2U	!^.*$!tel:+34-012345678!	.

[†]Universitat Autònoma de Barcelona.
[‡]Universitat Oberta de Catalunya.

Este trabajo está financiado por el Ministerio de Ciencia y Educación, a través de los proyectos *CONSOLIDER CSD2007-00004* y *TSI2006-03481*, y el programa de becas de la Fundación "la Caixa".

Analizamos a continuación la respuesta obtenida. Como indicábamos anteriormente, los registros de tipo NAPTR soportan la utilización de expresiones regulares. El primer parámetro de cada línea, *Order*, es utilizado para dar prioridad a cada uno de los resultados. El valor de la primera línea, 100, indica que de los cuatro servicios asociados a la consulta, *SIP* es el que tiene mayor prioridad. En el caso de tener más de un registro con el mismo *order*, el siguiente parámetro *Pref.* es utilizado. El parámetro *Flag* de cada línea, inicializado con el valor 'u' en cada respuesta, indica que la cadena asociada al parámetro *Regexp* contiene una URI. El parámetro *Rep.* (*Replacement*) indica el post-proceso que hay que aplicar a la expresión regular contenida en el parámetro *Regexp*. En nuestro ejemplo, el operador *'.'* indica que el URI final corresponde a la cadena que hay entre los delimitadores *'!^.*$!'* y *'!'* del parámetro *Regexp*. El parámetro *Service* indica el protocolo asociado a cada servicio resultante de la traducción de E.164 a URI (E2U). Por ejemplo, el parámetro *Service* de la primera línea indica que el protocolo a utilizar es SIP (*Session Initiation Protocol*) [3]. Las otras opciones indicadas en la respuesta para contactar a U_1 son (1) la utilización de su dirección de correo electrónico, (2) la utilización de su página Web y (3) la utilización de su servicio de telefonía tradicional.

Es importante hacer notar que la respuestas no incluyen la dirección IP vinculada a cada uno de los URIs asociados con el usuario U_1. Por lo tanto, una consulta adicional de DNS de tipo 'A' deberá ser utilizada para finalizar el proceso de traducción y contactar con uno de los servicios obtenidos. Una vez obtenida dicha dirección IP tras la segunda consulta a DNS, el usuario U_2 tratará de contactar con uno de los servicios devueltos a través de la consulta a ENUM. En nuestro ejemplo, suponiendo que U_2 decide contactar mediante el servicio con mayor prioridad, se establecerá un inicio de sesión SIP empleando el URI u1@sip.u1.es.

B. Implicaciones en la seguridad y privacidad del servicio

La utilización del protocolo DNS para la resolución de consultas ENUM implica ciertos riesgos asociados con la seguridad y privacidad del servicio. La explotación de vulnerabilidades de seguridad ya existentes en DNS es una forma clara de comprometer la seguridad de ENUM. En [4] podemos encontrar un buen análisis de seguridad sobre DNS. Algunos de los riesgos apuntados en [4] son los siguientes: (1) amenazas a la autenticidad e integridad del servicio, a través de ataques activos sobre el canal de comunicación entre clientes y servidores de DNS; (2) amenazas a la disponibilidad del servicio a través de ataques de denegación de servicio; (3) escalada de privilegios a través de deficiencias de seguridad en aplicaciones de DNS; (4) amenazas a la confidencialidad del servicio a través de escuchas pasivas sobre el canal durante el intercambio de información entre clientes y servidores de DNS. Las extensiones de seguridad de DNS (conocidas como DNSSEC), tratan de solventar algunas de estas amenazas. Más concretamente, autenticidad e integridad de DNS son tratadas en DNSSEC. Remitimos al lector a la sección V para más información al respecto.

Por otro lado, es importante recordar que el protocolo DNS se basa en operaciones sobre texto en claro. Por ello, amenazas a la privacidad de la información intercambiada pueden aparecer igualmente. Aunque esta fragilidad en la privacidad del servicio puede no ser crítica (aunque si molesta) en la utilización normal de DNS para la resolución de nombres de dominio, la situación puede ser diferente al utilizar los mismos mecanismos utilizados en DNS para un servicio como ENUM sobre VoIP. Los identificadores de ENUM apuntan a personas e instituciones físicas de una forma global. Un acceso no autorizado a los datos intercambiados entre clientes y servidores de ENUM podría ser utilizado tanto por *spammers* y campañas de marketing deshonestas, como en procesos de minería de datos interesados en recolectar información de ciudadanos e instituciones públicas o privadas. Algunas medidas propuestas por el IETF para reducir riesgos en la privacidad del servicio incluyen la limitación de información personal almacenada por ENUM, así como la solicitud del consentimiento de personas e instituciones a aparecer en bases de datos de ENUM. Aún así, más allá de limitar la cantidad de información almacenada en bases de datos asociadas con ENUM, ningún mecanismo parece haber sido propuesto con el objetivo de garantizar la privacidad de consultas relacionadas por terceras partes hacia ENUM. La posibilidad de que *malware* o servidores de acceso a Internet deshonestos puedan aprovechar consultas realizadas a ENUM para vulnerar la privacidad de sus usuarios no parece haber sido ampliamente tratada en la literatura.

C. Organización del artículo

Este articulo está estructurado de la siguiente manera. Los antecedentes utilizados en nuestro trabajo son introducidos en la sección II. La sección III presenta más en detalle algunas de las ventajas y limitaciones de utilizar la infraestructura de anonimato del proyecto Tor. La sección IV presenta la utilización de rangos y PIR (*Privacy Information Retrieval*) para la introducción de ruido en consultas de DNS. La sección V presenta algunas de las características de las extensiones de seguridad para DNS recogidas en DNSSEC. La sección VI presenta una evaluación de las herramientas utilizadas en nuestro trabajo. La sección VII cierra el artículo con una serie de conclusiones sobre los resultados obtenidos.

II. Antecedentes

Una primera solución al problema de privacidad planteado en la sección anterior es la utilización de una infraestructura de comunicaciones anónima. De esta manera, el emisor de las consultas DNS tratará de esconder su identidad a través de una red de *proxies* o *mixes* [5] que redirijan consultas y respuestas entre clientes y servidores de DNS. La utilización de un anonimato fuerte para un servicio como DNS puede suponer una gran latencia, incompatible con el rendimiento requerido por el servicio. Sin embargo, la utilización de una infraestructura de anonimato de baja latencia basada en *Onion Routing* [6] puede ayudar a mejorar la privacidad del servicio con un compromiso entre anonimato y rendimiento. Desde esta perspectiva, analizamos en la sección III algunos de las beneficios y limitaciones de la red de anonimato del proyecto Tor (*The second generation Onion Router*) [7].

Una segunda solución presentada recientemente en [8], [9] propone mejorar la privacidad en consultas de DNS perturbándolas con ruido aleatorio. Inspirada en la utilización de técnicas PIR (*Privacy Information Retrieval*) [10], el modelo presentado propone la construcción de distintos rangos de consultas enviadas hacia múltiples servidores, y donde tan sólo uno de los rangos incluye el objetivo real de la consulta. A través de la inserción de la consulta válida i dentro del intervalo $[1, n]$ de consultas distribuidas sobre múltiples servidores de DNS, e incluso asumiendo que los distintos servidores cooperasen para intercambiar información sobre las consultas recibidas, la probabilidad de que una tercera parte (servidor o atacante con acceso al canal entre usuario y servidores) pueda satisfactoriamente acertar la consulta deseada por el usuario es teóricamente $P_i = \frac{1}{n}$. El principal inconveniente de este modelo, al igual que las propuestas basadas en PIR [11], es el incremento en el coste de las comunicaciones asociadas a las consultas. En la sección IV analizamos más en detalle algunos de las beneficios y limitaciones de esta propuesta.

Aunque ambas soluciones (uso de la red de anonimato de Tor y uso de perturbaciones basadas en un enfoque PIR) podrían teóricamente beneficiar la privacidad del servicio, éstas propuestas no ofrecen sin embargo ninguna mejora en la integridad, autenticidad ni confidencialidad del tráfico intercambiado. La redirección de tráfico DNS a través de una infraestructura como la red del proyecto Tor puede suponer un riesgo a la integridad de consultas y respuestas. Ataques basados en técnicas de *man-in-the-middle* en los nodos de salida de la red, por ejemplo, pueden afectar al tráfico redirigido a través de ésta infraestructura (ver la sección III para más información). La misma técnica puede afectar a la propuestas presentadas en [8], [9] si ninguna contramedida es aplicada. Una de las propuestas del IETF para la protección de la integridad y autenticidad de tráfico DNS es la utilización de las extensiones de seguridad presentadas en [4] y conocidas como DNSSEC. En la sección V analizamos algunas de las características de estas extensiones y los beneficios que pueden aportar en combinación con las dos soluciones anteriores.

Ninguna de estas propuestas garantiza sin embargo la confidencialidad del servicio. Se podría pensar en la utilización de IPSec [12] para garantizar dicha confidencialidad entre clientes y servidores de DNS. Sin embargo, consideramos que la utilización de IPSec no es apropiado para solucionar cuestiones de confidencialidad en DNS. En primer lugar, el incremento en el ancho de banda y tiempo de proceso en servidores de DNS podría repercutir altamente en su rendimiento y causar problemas de disponibilidad y tolerancia [13]. En segundo lugar, la utilización de técnicas basadas en *cache* de información durante el proceso de resolución de consultas de DNS entre servidores intermedios plantean problemas en el esquema de cifrado de IPSec. Por último, recordamos que la motivación principal de nuestro trabajo no trata sobre la protección de la información intercambiada entre clientes y servidores de DNS, sino en la ocultación del origen de las consultas, o la consulta en si misma. Por ello, consideramos que la combinación de DNSSEC con cada una de las propuestas analizadas en las secciones III y IV es suficiente para nuestro estudio.

III. Realización de consultas anónimas a través de la infraestructura del proyecto Tor

Multitud de infraestructuras orientadas a reforzar el anonimato del tráfico dirigido hacia y por Internet han sido propuestas en la literatura. El principal objetivo de estas infraestructuras es la ocultación de la identidad de sus usuarios. Desde simples redes de *proxies* hasta complejos sistemas criptográficos, éstas infraestructuras ayudan a reforzar tanto el anonimato de servicios de alta latencia (por ejemplo, correo electrónico) como de baja latencia (por ejemplo, aplicaciones y servicios Web). Una de las infraestructuras más utilizadas en la actualidad para navegar de forma anónima a través de la Web es la infraestructura del proyecto Tor (*The second generation Onion Router*) [7]. Basada en la utilización de un esquema criptográfico conocido como *onion routing* [6], los diferentes componentes del proyecto Tor se distribuyen actualmente en modo de software libre y disponibles para gran multitud de plataformas y sistemas operativos.

El objetivo principal de Tor es proteger la privacidad de los usuarios que redirigen tráfico a través de sus componentes. Por ello, Tor construye circuitos criptográficamente protegidos a través de los cuales los mensajes son redirigidos. Por cada circuito, los componentes involucrados en la redirección de los mensajes tratan de realizar su reenvío de manera impredecible. El contenido de cada mensaje es además protegido mediante un cifrado independiente para cada nodo de la red de Tor, de manera que la existencia de adversarios controlando de forma parcial componentes de la red no puedan comprometer el origen de los mensajes. Tan pronto como un componente de la red recibe un nuevo mensaje, éste descifrará la capa que le atañe a través de las claves que se han establecido durante la construcción del circuito. Una vez descifrada, el componente comprobará si ha de entregar el mensaje al exterior, o si ha de redirigirlo a otro componente del circuito. Los diferentes caminos por los que los mensajes de un usuario serán redirigidos los establece su propio proceso local, en el lado del cliente. Ningún otro componente conoce la ruta al completo. Tan sólo el siguiente componente al que ha de redirigir el mensaje, en el caso de un nodo intermedio; o el destinatario final, en el caso de un nodo de salida.

La madurez del proyecto Tor y su bajo impacto en el rendimiento de servicios *on-line* lo posicionan como un candidato ideal para nuestro estudio sobre privacidad en protocolos de resolución de nombres. Aún así, todo y ofrecer excelentes prestaciones, Tor influye de forma evidente sobre el rendimiento de un servicio crítico como es DNS. Motivados por el impacto que Tor puede tener sobre la resolución de consultas de tipo NAPTR introducidas al inicio de este artículo, exponemos en la sección VI los resultados de una serie de experimentos orientados a analizar dicha penalización. Analizamos también en esta serie de experimentos cual es el grado de anonimato que cabe esperar de nuestra utilización de Tor. El conjunto de tests al completo fue procesado a través de Tor correctamente, sin experimentar serios problemas ni pérdida de mensajes. La desconexión de nodos en la red de Tor causó sin embargo algunas fluctuaciones en los tiempos analizados en nuestras pruebas. Es importante recordar que Tor

es una red de servidores basada en operadores voluntarios. Su servicio no es garantizado en ningún momento. Durante nuestros experimentos, se realizaron medidas acerca de la fiabilidad de los nodos y túneles construidos en nuestros escenarios. El resultado obtenido es de una fiabilidad en los nodos del 88%, lo cual nos lleva a una fiabilidad del 68% en cada túnel (asumiendo que los túneles se construyen con la longitud por defecto de tres nodos por circuito). Dada la naturaleza del protocolo DNS — y por lo tanto, del servicio de resolución de nombres que motiva el presente trabajo — consideramos estos resultados como aceptables.

Para obtener este impacto tan bajo sobre el tráfico redirigido a través de sus componentes, Tor basa su modelo de seguridad en un esquema realmente muy pragmático. En primer lugar, Tor asume la existencia de adversarios activos en la red. Dichos adversarios aparecen en el modelo de Tor con el objetivo de comprometer la identidad de los emisarios que envían mensajes a través de sus componentes. Estos adversarios pueden no sólo observar, sino también manipular parte de los mensajes redirigidos en Tor. Una primera implicación del modelo asumido por Tor es la completa visión del contenido de los mensajes que tendrán los componentes de salida (nodos finales de un circuito). De hecho, si ninguna contramedida es aplicada, es posible llevar a cabo ataques de tipo *man-in-the-Middle* en nodos terminales. Estos ataques pueden suponer, por ejemplo, la manipulación de respuestas a consultas de DNS redirigidas a través de Tor. Como resultado, un adversario controlando nodos de salida, y manipulando las respuestas de un servicio basado en DNS, podría redirigir al emisor de las consultas hacia servicios de información ilegítimos, o denegar la existencia de una consulta específica. Como introducíamos en la sección II, una solución eficiente a este problema es la combinación de Tor con el uso de consultas basadas en las extensiones para DNS propuestas en DNSSEC (ver la sección V para más información acerca de la utilización de estas extensiones). De esta manera, podemos garantizar no sólo la autenticidad e integridad de las consultas, sino también la no existencia de nombres o registros solicitados a través de DNS. Tal y como mostramos en la sección VI, el impacto en la latencia del servicio a través de la combinación de Tor con DNSSEC es mínimo (dado el número moderado de consultas analizadas en nuestro trabajo). Así pues, consideramos igualmente que esta primera limitación en el modelo de seguridad de Tor puede ser solventado a través de consultas basadas en DNSSEC.

Una segunda implicación del modelo de seguridad asociado a Tor es la posibilidad de sufrir ataques basados en análisis de tráfico. De nuevo, el objetivo del adversario podría ir dirigido a obtener la identidad del emisor de los mensajes que pasan por los nodos que controla. Varios ataques de este tipo han sido reportados en la literatura asociada. El ataque tratado en [14], [15], a menudo abreviado como *predecessor-attack*, asume que uno o varios adversarios controlan nodos de entrada y de salida en multitud de circuitos de la red de Tor. La cooperación entre estos nodos puede ser especialmente efectiva para degradar servicios ocultos proporcionados por la red de Tor. Aparte de proporcionar anonimato a sus usuarios, Tor puede posibilitar el anonimato a servicios que quieren permanecer ocultos tras su red. Sin embargo, la posibilidad de confabulación entre nodos de Tor para correlacionar de forma cooperativa información asociada a estos servicios (inicios de sesión, por ejemplo) puede degradar considerablemente el anonimato de Tor. Consideramos que este no es el caso de nuestro estudio. Pensamos además que las comunicaciones de un servicio de resolución de nombres como DNS no son fáciles de enlazar de manera tan evidente como podría ser la correlación de tráfico Web o SSH, a través de la utilización de *cookies* o identificadores de sesión asociados con los servicios ocultos tras la red de Tor. No consideramos por tanto como relevante para nuestro estudio otros ataques similares descritos en [16], [17] y que van especialmente dirigidos a la degradación del anonimato de servicios ocultos de la red de Tor.

Un ataque que ha tenido bastante repercusión entre los usuarios de Tor es el presentado en [18], donde los autores proponen la utilización de los límites en el ancho de banda ofrecido por los nodos y tratar de descubrir así los componentes específicos de un mismo circuito. Este ataque funciona además sin necesidad por parte del atacante de comprometer nodos en la red de Tor. Este ataque está especialmente orientado a descubrir el nodo de entrada de circuitos dirigidos hacia un mismo destinatario. El ataque asume por lo tanto que un adversario ha de tener completo control sobre el destinatario de los mensajes. La efectividad de la técnica propuesta no parece sin embargo escalar correctamente con el tamaño actual de la red de Tor. Los autores proponen adicionalmente en su trabajo un conjunto de mejoras que dificultan el ataque. Una técnica más apropiada, e inspirada por el ataque anterior, ha sido recientemente presentada en [19]. En este trabajo, los autores proponen un nuevo ataque basado en análisis de tráfico y promete una gran eficiencia incluso con un número de nodos comprometidos más bien limitado. De nuevo, el ataque propone la cooperación entre nodos de entrada y de salida en la red de Tor. Los componentes, conectados con el servicio de directorio (DS) de Tor, tratarán de inyectar información fraudulenta respecto al ancho de banda del que disponen y del rendimiento que tienen como componentes de Tor. De esta forma, cuando un cliente de Tor solicite información a DS para construir sus circuitos, los componentes bajo control del atacante aparecerán a menudo en posiciones privilegiadas de la lista de nodos de Tor. Si esto sucede, los componentes controlados por el atacante incrementarán sus probabilidades de actuar como nodos de entrada y de salida de multitud de circuitos y, por lo tanto, incrementarán la probabilidad de correlacionar información que permita al atacante identificar al emisor de los mensajes. En nuestra sección de evaluación (sección VI), analizamos más en detalle que repercusión en el anonimato de nuestros escenarios de pruebas podría tener el ataque reportado en [19].

IV. Utilización de rangos para perturbar consultas de DNS

Una alternativa al uso de infraestructuras de anonimato para incrementar la privacidad de consultas de DNS podría ser la introducción de ruido en las consultas realizadas. Aunque esta propuesta no parece haber sido ampliamente estudiada en la

literatura asociada, podemos encontrar en [8] ideas iniciales sobre una posible implementación de esta técnica. De hecho, el modelo propuesto por los autores de este trabajo está inspirado en la utilización de técnicas de PIR (*Privacy Information Retrieval*) [20], [10] utilizadas para la realización de consultas privadas en bases de datos distribuidas.

La propuesta presentada en [8] funciona de la siguiente manera: un usuario U, en lugar de realizar una única consulta a un servidor de DNS NS, construye un conjunto de consultas $Q\{H_i\}_{i=1}^{n}$. Suponiendo consultas de DNS de tipo A, el rango de consultas anterior incluirá hasta n nombres de dominio diferentes a resolver. Tan sólo la consulta $Q\{H_i\}$ contiene el nombre de dominio deseado por el usuario U. El resto de consultas $Q\{H_1\} \dots Q\{H_{i-1}\}$ y $Q\{H_{i+1}\} \dots Q\{H_n\}$ son elegidos al azar de una base de datos DB con nombres de dominio aleatorios. A través de este modelo, los autores afirman poder incrementar la privacidad de las consultas del usuario U. La única información divulgada por el usuario U a terceras partes (es decir, servidor NS y posibles atacantes con acceso al canal entre U y NS) es que la consulta real $Q\{H_i\}$ está dentro del intervalo $[1, n]$. Esto supone que la probabilidad que una de estas terceras partes pueda satisfactoriamente acertar la consulta $Q\{H_i\}$ requerida por U es igual a $P_i = \frac{1}{n}$. Animamos al lector a acudir a [8] para una descripción más detallada de la propuesta.

Pese a que los autores de la propuesta anterior garantizan que la probabilidad de acierto viene expresada por $P_i = \frac{1}{n}$, ésta podría ser reducida a valores inferiores si un atacante es capaz de interaccionar con el canal. Concretamente, en [8] no se expone el comportamiento ante situaciones en las que la resolución de $Q\{H_i\}$ ha sido fallida. Si suponemos que el atacante es capaz de manipular el tráfico (ya sea a través de ataques RST, o envío de tráfico ICMP adecuado, por ejemplo) de forma que $Q\{H_i\}$ no es resuelta satisfactoriamente, el cliente volverá a lanzar un nuevo rango que incluiría la petición $Q\{H_i\}$ nuevamente. Si este nuevo rango no es construido de forma adecuada, la probabilidad puede ser entonces reducida si el intruso es capaz de calcular la intersección de estos dos rangos. De forma particular, sea $Q_j\{H_i\}_{i=1}^{n}$ el j-ésimo rango consecutivo enviado para resolver la misma petición $Q\{H_i\}$, la probabilidad de acierto vendrá dada por la expresión:

$$P_{ij} = \frac{1}{|Q_1\{H_i\}_{i=1}^{n} \cap Q_2\{H_i\}_{i=1}^{n} \cap \dots \cap Q_j\{H_i\}_{i=1}^{n}|}$$

Una de las principales limitaciones del modelo anterior es el consumo elevado de ancho de banda provocado por las múltiples consultas del rango. Por ello, en [9], los autores presentan una mejora al modelo propuesto en [8] inspirándose en un esquema PIR básico de dos servidores y recogido en [20]. Esta propuesta se basa en la construcción de dos rangos $Q_1\{H_i\}_{i=1}^{n}$ y $Q_2\{H_i\}_{i=1}^{n+1}$, donde $H_{n+1} \in Q_2$ se corresponde a la petición deseada por el usuario U. Tras la construcción de Q_1 y Q_2, cada uno de estos rangos es enviado por U a un servidor de DNS distinto, esto es Q_1 enviado a NS_1 y Q_2 enviado a NS_2. A continuación, cada uno de los servidores resuelve cada una de las peticiones asociadas a su rango, esto es X_i es la dirección IP de resolución asociada a la consulta H_i. Seguidamente NS_1 calcula $A_1 = \sum_{i=1}^{n} \otimes X_i$ y NS_2 calcula $A_2 = \sum_{i=1}^{n+1} \otimes X_i$. Tanto el valor A_1 como el valor A_2 son enviados entonces al cliente, el cual obtiene la resolución asociada a H_{n+1} a partir de la expresión $X_{n+1} = A_1 \otimes A_2$. Como podemos observar, el consumo de ancho de banda se ve ampliamente reducido, ya que sólo se envían dos respuestas (A_1 y A_2) en comparación a las n respuestas de rango construido en la propuesta anterior.

A diferencia del uso de una infraestructura de anonimato o la utilización de un único rango, esta última alternativa provee privacidad desde el punto de vista del servidor. A pesar de esto y de reducir el consumo de ancho de banda, esta propuesta presenta diversos problemas relacionados con el anonimato en el canal. Así, un atacante con acceso al canal que interceptase los dos rangos por completo podría deducir la petición del cliente dado que $Q_1 \setminus Q_2 = H_{n+1}$. De forma similar, en el caso de interceptar las respuestas A_1 y A_2 enviadas por los servidores NS_1 y NS_2, el usuario malintencionado puede obtener la IP del mismo modo que lo hace el usuario legítimo al computar $X_{n+1} = A1 \otimes A2$ y, tras esto, deducir la petición original al realizar una resolución inversa de DNS sobre X_{n+1}. De forma análoga, la posibilidad de interferir en el canal puede conducir a un cliente a realizar una resolución errónea forzada por un usuario malintencionado, especialmente si la resolución está basada en UDP. Para esto, el atacante debería interceptar una respuesta de los dos servidores, por ejemplo A_1, para luego calcular $A_2^* = A_1 \otimes A_3$ (donde A_3 es una dirección IP no legítima) y enviar esta respuesta haciéndose pasar por el segundo servidor NS_2. A partir de aquí, el cliente resolvería de forma errónea la IP según la expresión $A_1 \otimes A_2^* = A_1 \otimes A_1 \otimes A_3 = A_3$. Al margen de la problemática asociada a la privacidad desde el punto de vista del canal, esta alternativa supone un problema adicional basado en la necesidad de modificar las aplicaciones servidor de DNS, dado que su comportamiento difiere al de cualquier servidor de DNS convencional.

Dadas las limitaciones, desde el punto de vista del anonimato, y de los problemas asociados a las propuestas basadas en perturbación de consultas expuestas anteriormente, proponemos una alternativa inspirada en las dos opciones anteriores. La idea principal es la distribución de la carga resultante de los rangos de consultas lanzadas por el usuario U a múltiples servidores $NS_1 \dots NS_m$. A diferencia de los esquemas anteriores, en nuestro caso nos basamos en la construcción de distintos rangos de consultas para cada uno de los servidores en $NS_1 \dots NS_m$. Los rangos de consultas irán así distribuidos de $Q\{H_1^{NS_1}\} \dots Q\{H_{\frac{n}{m}}^{NS_1}\}$ a $Q\{H_1^{NS_m}\} \dots Q\{H_{\frac{n}{m}}^{NS_m}\}$. Una vez obtenidas las respuestas de cada uno de los servidores, U comprobará que la consulta que incluye el nombre de dominio H_i deseado ha sido efectivamente resuelta, y descartará el resto de la información. En caso de que la respuesta no sea resuelta de forma correcta, se volverán a lanzar nuevos rangos construidos a partir de las mismas peticiones. Sin embargo, en esta ocasión, el rango enviado a un determinado servidor será totalmente distinto al que previamente recibió. De esta manera, un atacante no puede inferir cual fue la petición enviada mediante la intersección de los rangos consecutivos. Cabe notar que, en nuestro esquema,

la privacidad viene dada desde el punto de vista del canal y del servidor, resolviendo las deficiencias que aparecían en las alternativas previas. Con el objetivo de evitar ataques basados en la alteración de las respuestas por parte de un usuario malintencionado ya vistos anteriormente, nuestra propuesta garantiza la autenticidad e integridad empleando DNSSEC en cada una de las peticiones resueltas. A pesar de esto, nuestra propuesta adolece de un elevado consumo de ancho de banda y tiempo de respuesta (tal y como exponemos en la sección VI) a medida que el tamaño del rango aumenta.

V. DNSSEC: EXTENSIONES DE SEGURIDAD PARA DNS

DNSSEC (*Domain Name System SECurity extensions*) es un conjunto de especificaciones del IETF orientadas a reforzar la autenticidad e integridad de registros de DNS (abreviados generalmente por RR, del inglés *Resource Records*) como, por ejemplo, registros de tipo NAPTR. DNSSEC se basa en la utilización de algoritmos de criptografía de clave pública y firmas digitales. DNSSEC es criticado a menudo por no haber sido desplegado después de más de diez años de discusiones y revisiones [21]. Sin embargo, es la mejor, y quizás la única, solución para reducir el riesgo de ataques contra la integridad y autenticidad de la información obtenida a través de servicios basados en DNS — tales como ataques de tipo *man-in-the-middle* y envenenamiento de caché [4]. Aunque las primeras propuestas de DNSSEC plantearon ciertos problemas de gestión asociados con el manejo de claves, las últimas revisiones de DNSSEC solventan este problema con el modelo de delegación de firmas propuesto en los RFCs (*Request for Comments*, disponibles en `http://www.rfc-archive.org/`) 3658 y 3755. DNSSEC está siendo desplegado actualmente en multitud de zonas (algunas aún de forma experimental), tales como Suecia, Puerto Rico, Bulgaria y Méjico (ver `http://www.xelerance.com/dnssec/` para una representación gráfica de las zonas ya desplegadas). En el momento de escribir el presente artículo, más de diez mil zonas de DNS tienen activadas las extensiones de DNSSEC (ver `http://secspider.cs.ucla.edu/`). El despliegue sobre las zonas raíz de DNS está siendo discutido en la actualidad. Algunos problemas, más bien de carácter político, parecen entorpecer el despliegue de DNSSEC a este nivel [22].

Las principales características de DNSSEC están descritas en los siguientes RFCs: 3658, 3755, 4033, 4034 y 4035. Un análisis de riesgos sobre DNS y su tratamiento a través de DNSSEC está disponible en [4]. Aparte de ofrecer autenticación de origen e integridad para registros de DNS (tales como registros de tipo A, CNAME, MX y NAPTR), DNSSEC también ofrece pruebas de la no existencia de registros. Si un registro de tipo NAPTR es requerido y éste no se encuentra en ninguna base de datos de DNS, una prueba de no existencia firmada por el servidor responsable del dominio será enviada a la aplicación que realizó la consulta. Como indicábamos anteriormente, DNSSEC permite dos estrategias diferentes a la hora de gestionar firmas y claves. Por un lado, los administradores de una o varias zonas pueden firmar digitalmente sus registros empleando su propia conjunto de claves. Por otro lado, los administradores pueden decidir también utilizar un esquema de cadenas de confianza entre zonas y subzonas. De esta manera, a partir de una consulta de registros de una subzona determinada, el usuario puede validar las firmas a partir de información recibida de zonas de nivel superior. Para ello, DNSSEC cuenta con cuatro tipo de registros adicionales respecto a DNS: (1) registros de tipo RRSIG (*Resource Record Signature*), utilizados para almacenar la firma asociada a cada registro de una zona determinada; (2) registros de tipo DNSKEY (*DNS Public Key*) que contienen la llave pública asociada a una zona y que permite a las aplicaciones que realizan consultas de DNS poder validar las firmas obtenidas; (3) registros de tipo DS (*Delegation Signer*), añadidos en zonas de nivel superior con el objetivo de permitir funciones de delegación en sus subzonas; y (4) registros de tipo NSEC (*Next Secure*) que contienen información acerca del siguiente registro en la zona, con el objetivo de permitir la verificación de no existencia de registros consultados. DNSSEC utiliza adicionalmente dos nuevos bits de bandera (del inglés, *bit flags*) ya existentes (pero no utilizados) en las cabeceras de mensajes DNS. Estos bits son utilizados en DNSSEC para indicar (1) que la aplicación que realiza la consulta acepta información sin autenticación y (2) que los registros incluidos en una respuesta han sido previamente autenticados por el servidor.

Respecto al conjunto de claves utilizado en DNSSEC para la firma de registros, uno o dos pares de claves deberán ser generados. Como indicábamos más arriba, si un administrador decide firmar los registros de sus zonas sin utilizar cadenas de confianza, el conjunto de registros será firmado únicamente a partir de un par de claves ZSK (*Zone Signing Keys*). De lo contrario, si el administrador decide utilizar cadenas de confianza entre zonas de nivel superior e inferior, dos conjuntos de claves deberán ser generados por cada zona: un par de claves KSK (*Key Signing Keys*) es generado para firmar el registro DNSKEY de cada zona; y un par de claves ZSK es utilizado para firmar el resto de registros. Diversos algoritmos son soportados por DNSSEC para la generación de claves y la firma de registros. Entre ellos, podemos encontrar la utilización de RSA, DSA (*Digital Signature Algorithm*), y ECC (*Elliptic Curve Cryptosystem*). Estos algoritmos son utilizados únicamente para generar firmas, en combinación con funciones *hash* de tipo MD5 o SHA1. La combinación RSA/SHA1 es de obligatoria implementación para cualquier cliente y servidor compatible con DNSSEC. El tipo y longitud de clave escogidas durante el proceso de firmas debe ser escogido cuidadosamente, ya que afecta de manera significativa al tamaño de las respuestas y a la carga de servidores que realizan verificación de firmas. Los resultados presentados en [23] apuntan a un impacto del 3% al 12% sobre el rendimiento en servidores de DNS gestionando zonas con firmas generadas a partir de clave KSK/ZSK basadas en RSA/SHA1 de longitud 1200/1024 bits; y de un 2% a un 6% para zonas firmadas a partir de ECC con clave de longitud 144/136 bits.

El periodo de validez de las firmas y claves debe ser también definido con cautela. La utilización de caches intermedias durante el proceso de validación de firmas puede ocasionar algunos conflictos durante el mantenimiento y actualización de claves. Los parámetros de sincronización en DNSSEC son

por tanto críticos a la hora de hacer funcionar el proceso de verificación. Otras limitaciones referidas a menudo en la literatura respecto a DNSSEC es la posibilidad de utilizar el registro NSEC para la obtención del total de registros de una zona protegida. Como indicamos anteriormente, este registro es utilizado para poder la generación de pruebas de no existencia de recursos. Aunque algunos grupos de trabajo de DNSSEC no ven está posibilidad como una limitación real del protocolo (pues apuntan a que por definición, cualquier información contenida en una zona debería ser pública), un nuevo registro, llamado NSEC3, ha sido propuesto para almacenar resúmenes en lugar de nombres de registro. De esta manera, se evita la posibilidad de utilizar malintencionadamente los registros NSEC de una zona para recorrer la base de nombres al completo. Por último, el almacenamiento seguro de la clave utilizada para los esquemas de confianza de DNSSEC ha recibido también fuertes críticas en la literatura asociada [21]. Sin embargo, y a diferencia de soluciones similares basadas en PKI (*Public key infrastructure*), el concepto de cadenas de confianza de DNSSEC parece ofrecer mayores beneficios en comparación al uso de certificados X.509 en PKIs tradicionales, ya que el número de claves a tratar es generalmente mucho menor en DNSSEC. Invitamos al lector a acudir a RFC 3658 y RFC 3755 para más información al respecto.

VI. Evaluación de las propuestas

Presentamos en esta sección una evaluación práctica del impacto que supone la utilización de las dos técnicas analizadas en este trabajo (modelo basado en la infraestructura anónima del proyecto Tor y utilización de rangos) sobre un escenario de resolución de consultas DNS y DNSSEC de tipo NAPTR. El montaje preparado para nuestra experimentación consta de los siguientes componentes. Un equipo R corriendo sobre una plataforma Intel Core 2 Duo 2 GHz y con 1 GB de memoria realiza las consultas a un servicio de resolución de nombres global G simulado por medio de los tres servidores siguientes: S_1, funcionando sobre una plataforma AMD Duron 1 GHz con 256 MB de memoria; S_2, funcionando sobre una plataforma Intel PIII 1 GHz con 512 MB de memoria; y un servidor S_3 funcionando sobre una plataforma Intel Xeon 2.4 GHz con 1 GB de memoria. Cada uno de los servidores se encuentra localizado en una zona geográfica distinta. El servicio de DNS configurado en cada uno de estos equipos contiene las extensiones de DNSSEC activadas y se basa en BIND 9.4.2 (disponible en `http://www.isc.org/products/BIND/`). La configuración de cada servidor consiste en una base de datos $\mathcal{N}$ que contiene más de veinte mil registros de tipo NAPTR. Cada uno de estos registros contiene el conjunto de firmas especificadas en DNSSEC. Las distintas zonas en $\mathcal{N}$ fueron firmadas a través de la herramienta *dnssec-signzone* que viene con el paquete BIND 9.4.2. El tamaño inicial de la base de datos $\mathcal{N}$ es de 6MB. El incremento en el tamaño de la base de datos al incluir las firmas es de 16MB. El tamaño final de $\mathcal{N}$ es por tanto de 22MB. La generación de claves se realizó con *dnssec-keygen*, también proveniente del paquete BIND 9.4.2. El tamaño de las claves es de 1200 bits para el par de claves KSK (*Key Signing Keys*) y 1024 bits para el par de claves ZSK (*Zone Signing Keys*). Las claves generadas se basan en RSA y las firmas en RSA/SHA1. Aunque el uso de criptografía de curva elíptica en DNSSEC promete un incremento en el espacio de la base de datos inferior al espacio requerido para firmas basadas en RSA o DSA [23], los algoritmos necesarios no se encuentra implementados aún en la versión de BIND utilizada para nuestros experimentos.

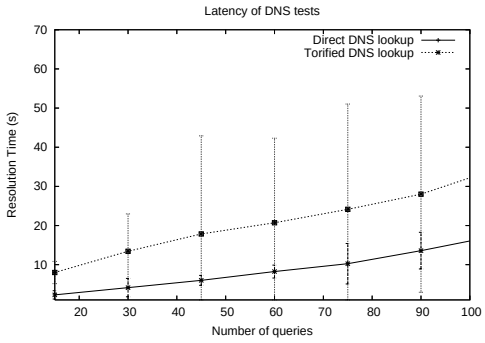
A. Evaluación del modelo basado en Tor

Cuatro conjuntos de tests son configurados en esta primera evaluación para simular un intercambio directo e indirecto (a través de Tor) entre R y G: (1) consultas/respuestas DNS; (2) consultas/respuestas DNSSEC; (3) consultas/respuestas DNS a través de Tor; y (4) consultas/respuestas DNSSEC a través de Tor. Para el intercambio de mensajes entre R y G, un enlace directo es utilizado en los dos primeros conjuntos de tests. Etiquetamos estos tests como *Direct DNS tests* y *Direct DNSSEC tests*. Un enlace indirecto basado en el protocolo SOCKS4a es utilizado en los dos últimos tests. Un cliente de Tor basado en la versión 0.1.2.18 (disponible en `http://torproject.org`) se ejecuta en el equipo R y redirige el tráfico de las consultas a través de mensajes SOCKS4a al conjunto de servidores en G. Etiquetamos estos dos últimos tests como *Torified DNS tests* y *Torified DNSSEC tests*. La gestión de consultas y respuestas DNS y DNSSEC en R se realiza a través de una aplicación basada en la librería *NET::DNS* (disponible en `http://www.net-dns.org/`) y desarrollada en *Perl*. Cada consulta es ejecutada como un proceso independiente en R. La ejecución de n consultas supone por lo tanto la ejecución de n procesos independientes en R.

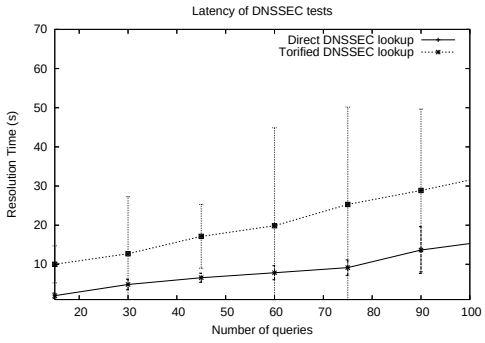
Bandwidth class					
996KB/s	*621KB/s*	*111KB/s*	*59KB/s*	*29KB/s*	*<29KB/s*
131	130	338	315	406	158

Cuadro I
Nodos disponibles en la red de Tor durante los experimentos, clasificados según su ancho de banda.

La actividad y estado de la red de Tor al inicio de nuestros experimentos es analizada a través de TorFlow, un conjunto de *scripts* desarrollados en *Python* y disponibles en `http://torproject.org/svn/torflow/`. El cuadro I muestra un resumen del conjunto de nodos disponibles en la red de Tor durante nuestros experimentos, y clasificados según el ancho de banda reportado por los distintos nodos al servicio de directorio (DS) de la red de Tor. Podemos observar que más de mil cuatrocientos nodos están disponibles para redirigir el tráfico de nuestros experimentos. El cliente de Tor instalado en R está configurado por defecto. Por lo tanto, la longitud de los circuitos generados a partir de este cliente es la longitud por defecto definida en Tor (es decir, tres nodos por circuito). El porcentaje de desconexiones reportado por TorFlow, con un margen de error del 8%, es del 12%. De acuerdo con [24], la fiabilidad de los circuitos en la red de Tor puede ser estimada de la siguiente manera. Sea l la longitud de los circuitos (tres nodos por circuito en nuestro caso). Sea f la fiabilidad de cada nodo en el sistema



(a) Direct vs. torified DNS tests



(b) Direct vs. torified DNSSEC tests

Fig. 1. Resultados de la evaluación del modelo basado en Tor.

(88% según hemos visto más atrás). La fiabilidad de cada circuito puede ser estimada como f^l. Así pues, asumimos un 68% de fiabilidad por cada uno de los circuitos de Tor que se utilizan en nuestros experimentos.

La Fig. 1 muestra los resultados de ejecutar los conjuntos de tests definidos en este apartado. La Fig. 1(a) muestra la ejecución de los tests etiquetados como *direct DNS* y *torified DNS*. La Fig. 1(b) muestra la ejecución de los tests etiquetados como *direct DNSSEC* y *torified DNSSEC tests*. Cada grupo de tests es ejecutado múltiples veces con el objetivo de generar diferentes series de consultas generadas de forma aleatoria a partir del conjunto de nombres en $\mathcal{N}$. Cada serie es almacenada de manera persistente en la ejecución del primer conjunto de tests (*direct DNS tests*) y cargada en el resto de tests — con el objetivo de facilitar la comparación de resultados. Podemos observar a través de la primera curva de la Fig. 1(a) y la Fig. 1(b) que las diferencias en los tiempos de resolución entre consultas DNS o DNSSEC es prácticamente mínima. Por otro lado, la ejecución de cada serie de tests de los conjuntos *torified DNS* y *torified DNSSEC* es lanzada sobre un circuito de Tor diferente. Como ya indicamos en la sección III, la desconexión dinámica de nodos en la red de Tor forzó la creación de más de un circuito por serie en algunos casos. Esta desconexión se ve reflejada en Fig. 1(a) y Fig. 1(b) con los peores tiempos de resolución mostrados en los intervalos de confianza de cada serie. Algunas de estas desconexiones forzaron a la aplicación en R a repetir algunas de las consultas. Sin embargo, nos gustaría hacer constar que incluso en estos casos extremos, los tiempos totales son considerablemente aceptables. Además, como ya adelantamos en la sección III, todas las consultas fueron evaluadas y contestadas. No experimentamos en ningún momento pérdida de información. Consideramos por lo tanto estos resultados como positivos. Pensamos que la combinación con DNSSEC y su impacto en el servicio es igualmente aceptable y necesaria, ya que garantiza propiedades de seguridad no contempladas en Tor, tales como integridad, autenticidad y prueba de no-existencia. Estas propiedades son esenciales para detectar y prevenir ataques de tipo *man-in-the-middle* que podrían ser perpetrados por nodos de salida de la red de Tor. No experimentamos durante la ejecución de nuestros tests ninguna alteración del conjunto de firmas de los registros de $\mathcal{N}$ consultados a través de Tor sobre G.

Motivados por conocer el grado de anonimato que cabría esperar durante estos experimentos, adoptamos la misma estrategia presentada en [19] y estimamos el nivel de anonimato a partir de la distribución de probabilidades asociada a los nodos en la red de Tor [25], [26]. Sea N el número de nodos en la red de Tor. Sea $p(x_i)$ la probabilidad de un nodo de ser seleccionado para aparecer en un circuito. Calculamos la siguiente métrica basada en el concepto de entropía [27]:

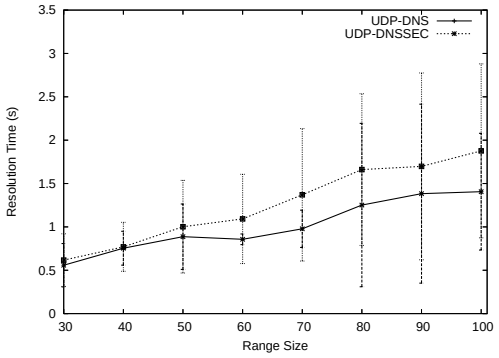
$$H(N) = -\sum_{x_i \in N} p(x_i) \log_2 (p(x_i));$$

Si cada nodo en la red de Tor tuviera la misma probabilidad de ser incluido en un circuito, y normalizando el valor obtenido de $H(N)$ dividiéndolo por $log_2(|N|)$, obtendríamos una entropía ideal con valor de 1. Sin embargo, el algoritmo para la creación de circuitos en Tor da prioridad a aquellos nodos de la red con mejores prestaciones (tales como ancho de banda disponible, tiempos de conexión en la red, protocolos aceptados en sus políticas de salida, etc.). No podemos asumir por lo tanto que la probabilidad de todos los nodos será la misma. A través de las herramientas incluidas en TorFlow aproximamos el valor de $H(N)$ agrupando el conjunto general de nodos a partir de su ancho de banda, dividiéndolos en diferentes segmentos y creando con cada uno de los segmentos múltiples circuitos. La estimación obtenida es de 0.89.

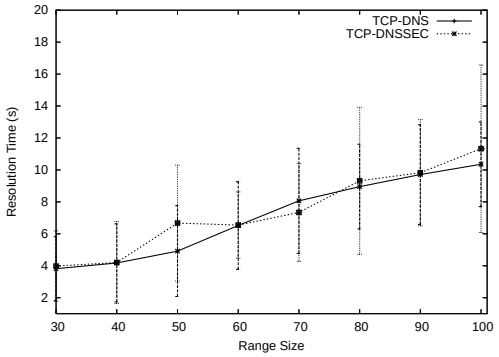
Como ya adelantamos en la sección III, el modelo de seguridad de Tor presenta ciertas vulnerabilidades que podrían ser explotadas por un atacante con el objetivo de degradar el valor anterior. Si el atacante tiene bajo su control un elevado número de nodos en la red, y consigue hacer cooperar nodos de entrada y de salida de un mismo circuito, podrá obtener la localización del emisor que construyó dicho circuito, el destinatario de los mensajes y el contenido del mensaje. El principal requisito para el atacante es por lo tanto asegurar que los nodos que están bajo su control tengan alta probabilidad de ser seleccionados en el máximo número de circuitos posibles. De acuerdo con los diseñadores de Tor en [7], si un adversario controla $m > 1$ nodos del total de N nodos en la red, podría llegar a correlacionar como máximo $(\frac{m}{N})^2$ del tráfico. Este modelo asume nuevamente que los distintos nodos tienen igual probabilidad de ser seleccionados para participar en un mismo circuito. Los autores en [19] muestran en su trabajo que a través de la inyección de información falsa sobre el servicio de directorio (DS) de Tor, es posible para un atacante incrementar

la probabilidad de que sus nodos sean seleccionados. Los autores describen que el modelo anterior debería ser sustituido por $(\frac{m}{N})(\frac{m-1}{N-1})$, afirmando que éste nuevo modelo tiene en cuenta la posibilidad de que un nodo sea utilizado únicamente una vez por circuito.

Usando esta segunda métrica, los autores muestran en su trabajo que la implementación práctica de su propuesta sobre una red alternativa de Tor desplegada sobre la red PlanetLabs resultó en una mejora de casi setenta veces la expectación analítica, siendo el número de nodos controlado por el atacante de un 5% a un 10% sobre el total de nodos de su red de Tor. El mismo porcentaje de nodos comprometidos sobre la red real de Tor utilizada durante nuestros experimentos, supondría que un hipotético atacante controlase de setenta a más de cien nodos de la red. La predicción analítica a partir del modelo propuesto en [19] indicaría que el número de caminos potencialmente comprometidos en ese caso es entre un 0.21% y un 0.67% del total de caminos construidos en la red de Tor. Asumiendo que el ataque presentado en [19] escalara correctamente sobre la red real de Tor, y manteniendo la mejora reportada por los autores, deberíamos esperar teóricamente que el número real de caminos comprometidos fueran de un 15% a un 48%. Esto supondría una degradación del anonimato de casi un 50%. Así pues, deberíamos considerar que el grado de anonimato obtenido ofrecido por Tor durante nuestros experimentos estaría en el mejor de los casos cercano a un valor H de 0.89; y en el peor de los casos, asumiendo un ataque como el presentado en [19], cercano a un 0.45.



(a) UDP tests



(b) TCP tests

Fig. 2. Evaluación del modelo basado en PIR.

B. Evaluación del modelo basado en PIR

El prototipo de pruebas empleado para la evaluación de nuestro modelo basado en PIR ha sido programado en el lenguaje *Python*, empleando el módulo *dnspython* [28] para las resoluciones, y *m2crypto* [29] (una interfaz de acceso a la librería *OpenSSL* [30]) para la verificación de las firmas digitales definida en DNSSEC.

El conjunto de pruebas para nuestra propuesta se ha basado en evaluar los tiempos de respuesta en función de los tamaños de los rangos. Para esto, se han lanzado tres rangos del mismo tamaño dirigidos a los tres servidores de DNS configurados, de forma que sólo uno de estos contenía la petición deseada. Dicho tamaño ha sido modificado para obtener resultados diferentes. Asimismo, las pruebas se han realizado usando tanto DNS bajo TCP, como UDP. Simultáneamente, se han considerado los tiempos de respuesta empleando DNSSEC (garantizando la integridad y autenticidad de las respuestas) o desactivándolo. En Fig. 2(a) y Fig. 2(b) se pueden observar los resultados de los *tests* realizados para rangos de tamaño comprendido entre treinta y cien peticiones por cada uno de los tres servidores.

Por un lado, y como es de esperar, los tiempos de respuesta aumentan de forma prácticamente lineal a medida que se incrementa el tamaño de peticiones por rango, viéndose especialmente afectado de este hecho las consultas basadas en TCP debido a la sobrecarga que supone este protocolo frente a UDP. Ante estas dos alternativas, UDP parece ser más adecuado para nuestro propósito ya que, para un tiempo máximo aceptable de 1.5 segundos aproximadamente, tenemos una probabilidad de acierto de $P_i = \frac{1}{3 \cdot 80} = \frac{1}{240} \simeq 0.004167$, valor que consideramos suficientemente satisfactorio. De forma genérica, la probabilidad de acierto P_i para un rango de tamaño n y m servidores distintos viene determinada por la expresión $P_i = \frac{1}{n \cdot m}$.

A pesar de la dificultad en la predicción de la petición original, tanto los resultados basados en UDP como los basados en TCP adolecen de un elevado consumo de ancho de banda. Sin embargo, desde nuestro criterio, este coste adicional puede ser tolerado al considerar el nivel de privacidad conseguido. En este sentido, en contextos donde el ancho de banda sea un factor crítico, es posible reducir el tamaño del rango, disminuyendo eso sí, la probabilidad de predicción de la consulta válida.

En términos generales, el consumo del ancho de banda es inversamente proporcional a la probabilidad de acierto: a menor probabilidad necesitamos mayor número de peticiones por rango y servidor. De hecho, la probabilidad de acierto se comporta de forma lineal respecto al tamaño del rango. Así, si reducimos en un 50% las peticiones por rango y servidor, la probabilidad de acierto se ve incrementada por dos.

Por otro lado, podemos observar como la penalización introducida por DNSSEC no es especialmente significativa en los tiempos de respuesta, resolviendo así los problemas de integridad y autenticidad a los que eran susceptibles las otras propuestas. Es por esto que consideramos la activación de DNSSEC como un factor decisivo para evitar ataques de manipulación del tráfico de respuesta.

VII. Conclusiones

El uso del protocolo DNS (*Domain Name System*) como tecnología base de nuevos servicios de resolución de nombres puede suponer riesgos en su seguridad y privacidad. La explotación de vulnerabilidades y debilidades ya existentes en DNS pueden permitir a un atacante comprometer estos nuevos servicios. En este trabajo, nos hemos centrado en el problema de privacidad que puede suponer la utilización de DNS como infraestructura base del conjunto de protocolos utilizado en VoIP para la traducción de números de teléfono E.164 a nombres de servicio y direcciones IP. Hemos analizado dos posibles propuestas que podrían mejorar el problema de privacidad planteado: utilización de la infraestructura de anonimato del proyecto Tor [7] y utilización de perturbaciones y rangos de consultas múltiples.

La realización de consultas de DNS a través de la infraestructura del proyecto Tor tiene por objetivo esconder el origen de las consultas. Un conjunto de encaminadores se encargan de preservar la privacidad del emisor que realiza las consultas mediante la utilización de circuitos protegidos criptográficamente. De forma experimental, hemos analizado el coste en la latencia del servicio que supone la utilización de Tor para la realización de consultas de DNS de tipo NAPTR, así como el grado de anonimato que cabría esperar tras la realización de nuestros experimentos. Teniendo en cuenta el modelo de seguridad de Tor, consideramos los resultados obtenidos como muy satisfactorios. Adicionalmente, y con el objetivo de garantizar la integridad y autenticidad de las respuestas recibidas, hemos analizado también la implicación de combinar el anonimato ofrecido por Tor junto con la utilización del conjunto de extensiones de seguridad para DNS (conocidas como DNSSEC). Los resultados obtenidos son igualmente satisfactorios, siendo mínima la penalización introducida por DNSSEC a los resultados anteriores.

Por otro lado, hemos analizado la implementación de una propuesta propia basada en un modelo PIR (*Privacy Information Retrieval*) para la introducción de ruido en consultas de DNS. El objetivo de la propuesta es ofrecer anonimato tanto a nivel de canal (e.g., para evitar que un atacante con acceso al medio puede inferir las consultas en las que un usuario está interesado) como a nivel de servidores de DNS (e.g., para evitar que un servidor malintencionado pueda realizar análisis de perfiles y preferencias). Nuestra propuesta se inspira en dos trabajos existentes reportados en [8] y [9], mejorando deficiencias de seguridad detectadas en ambas contribuciones, tales como posibilidad de manipulación de las respuestas e intersección de rangos enviados por parte de un atacante con acceso al medio. La utilización de DNSSEC en nuestra propuesta garantiza además la autenticidad e integridad de las consultas realizadas, así como pruebas de no existencia. El principal inconveniente de nuestra contribución es el elevado incremento tanto en el ancho de banda como en los tiempos de resolución a medida que los rangos utilizados aumentan. Actualmente estamos trabajando en una mejora de nuestros algoritmos para solventar estas limitaciones.

Referencias

[1] Mealling, M. and Daniel, R. The Naming Authority Pointer (NAPTR) DNS Resource Record. *Request for Comments, RFC 2915*, IETF, 2000.

[2] Faltstrom, P. and Mealling, M. The E.164 to Uniform Resource Identifiers Dynamic Delegation Discovery System Application. *Request for Comments, RFC 3761*, IETF, 2004.

[3] Rosenberg J., et al. Session Initiation Protocol. *Request for Comments, RFC 3261*, IETF, 2002.

[4] Atkins, D. and Austein, R. Threats analysis of the domain name system (DNS). *Request for Comments, RFC 3833*, IETF, 2004.

[5] Chaum, D. Untraceable electronic mail, return addresses, and digital pseudonyms. *In: Communications of the ACM*, 24(2):84–88, 1981.

[6] Reed, M. G., Syverson, P. F., and Goldschlag, D. M. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1998.

[7] Dingledine, R., Mathewson, N., and Syverson, P. F. Tor: The second-generation Onion Router. *In: 13th conference on USENIX Security Symposium*, 2004.

[8] Zhao, F., Hori, Y., and Sakurai, K. Analysis of Privacy Disclosure in DNS Query. *In: IEEE Int'l Conf. on Multimedia and Ubiquitous Engineering*, pp. 952–957, 2007.

[9] Zhao, F., Hori, Y., and Sakurai, K. Two-Servers PIR Based DNS Query Scheme with Privacy-Preserving. *In: IEEE Int'l Conf. on Intelligent Pervasive Computing*, pp. 299–302, 2007.

[10] Ostrovsky, R. and Skeith, W.E. A survey of single database PIR: Techniques and applications. *Proceedings of Public Key Cryptography (PKC-2007)*, 2007.

[11] Sion, R. and Carbunar, B. On the Computational Practicality of Private Information Retrieval. *Network and Distributed Systems Security Symposium (NDSS)*, 2007.

[12] IETF IPsec. Available from: http://www.ietf.org/ids.by.wg/ipsec.html

[13] Meenakshi, S.P. and Raghavan, S.V. Impact of IPSec Overhead on Web Application Servers. *Advanced Computing and Communications (ADCOM2006)*, pp. 652–657, 2006.

[14] Wright, M. K., Adler, M., Levine, B. N., Shields, C. An analysis of the degradation of anonymous protocols. *In: Network and Distributed Security Symposium - NDSS'02*, February 2002.

[15] Wright, M. K., Adler, M., Levine, B. N., Shields, C. Passive-Logging Attacks Against Anonymous Communications Systems. *ACM Transactions on Information and System Security (TISSEC)*, Vol. 11, No. 2, Article 7, 1–33, Pub. date: May 2008.

[16] Overlier, L. and Syverson, P. Locating hidden servers.. *IEEE Symposium on Security and Privacy)*, pp. 100–114, 2006.

[17] Murdoch, S. J. Hot or not: Revealing hidden services by their clock skew.. *In: ACM Conference on Computer and Communications Security.*, pp. 27–36, 2006.

[18] Murdoch, S. J. and Danezis, G. Low-cost traffic analysis of Tor. *IEEE Symposium on Security and Privacy)*, pp. 183–195, 2005.

[19] Bauer, K., McCoy, D., Grunwald, D., Kohno, T., and Sicker, D. Low-resource routing attacks against Tor. *2007 ACM workshop on Privacy in electronic society*, pp. 11–20, 2007.

[20] Chor, B., Kushilevitz, E., Goldreich, O., and Sudan, M. Private information retrieval *In: Journal of the ACM*, pp. 965–981, New York, USA, 1998

[21] Seltzer, L. DNSSEC Is Dead, Stick a Fork in It. Available from: http://www.eweek.com

[22] DNSSEC Deployment Initiative. Available from: http://dnssec-deployment.org/

[23] Ager, B., Dreger, H., and Feldmann, A. Predicting the DNSSEC overhead using DNS traces. *40th Annual Conf. on Information Sciences and Systems*, pp. 1484–1489, 2006.

[24] Borisov, N., Danezis, G., Mittal, P., and Tabriz, P. Denial of service or denial of security?. *In: 14th ACM conference on Computer and communications security*, pp. 92–102, New York, USA, 2007.

[25] Diaz, C., Seys, S., Claessens, J., and Preneel, B. Towards measuring anonymity. *In: Privacy Enhancing Technologies Workshop*, 2002.

[26] Serjantov, A. and Danezis, G. Towards an information theoretic metric for anonymity. *In: Privacy Enhancing Technologies Workshop*, 2002.

[27] Shannon, C. A Mathematical Theory of Communication. *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[28] Nomium Inc. A DNS toolkit for Python http://www.dnspython.org/

[29] Ng Pheng Siong, Toivonen, H. Mee Too Crypto http://chandlerproject.org/bin/view/Projects/MeTooCrypto.

[30] Eric A. Young, and Tim J. Hudson OpenSSL: The Open Source toolkit for SSL/TLS http://www.openssl.org/