Protocolos de Comunicación

Material didáctico

Protocolos de Comunicación: Material didáctico

Revisión: 2021-07-26 22:34:27

Parte I. Guías Prácticas

Convenciones

Para todos aquellos ejercicios marcados con la etiqueta*{W}* se debe utilizar Wireshark para analizar los mensajes intercambiados por las aplicaciones.

Introducción y repaso

Revisión: 2021-07-26 22:34:27

Resumen

El objetivo de esta guía es aplicar conocimientos generales obtenidos en otras materias con la intención de reforzarlos para la aplicación en esta materia e introducir algunos otros que servirán como herramientas para el desarrollo de otros trabajos prácticos.

Entorno UNIX

E1. Encuentre todos los archivos alojados en el directorio `/var/log` (incluyendo subdirectorios) que contenga el texto `boot`. Quédese observando nuevas entradas en dichos archivos.

Sistemas operativos

E2. El programa que se describe en `copy.c` copia el contenido de un archivo en otro; permitiéndole al usuario especificar el tamaño del buffer utilizado.

Mida el tiempo de ejecución (utilizando **time(1)**) cuando se utiliza un buffer de tamaño 0 (`sendfile`), 1, 512, 1024, 2048, 4096.

¿A qué se debe la diferencia de tiempos? ¿Cual es la principal ventaja de la llamada de sistema `sendfile`?

Sugerencia

Puede generar un archivo de casi 100 megabytes ejecutando **dd if=/dev/urandom of=archivo-in bs=4096 count=25600**

Sobre codificaciones generales

E3. ¿Cómo se codifica los caracteres en un documento texto plano?¿Qué diferencias entre las codificaciones `ASCII`, `ISO-8859-1`, `UNICODE`? ¿Cuales son su ventajas y desventajas?

E4. ¿Es lo mismo hablar de `UNICODE` y de `UTF-8`? ¿son conceptos que refieren a diferentes cosas?

E5. ¿Qué criterio utilizaría para decidir utilizar `UTF-8`, `UTF-16` o `UTF-32`?

E6. Ejecute las siguientes directivas Java:

```
System.out.println("\u1F355");
System.out.println("\u1F355".length());
```

¿De qué *code point* se trata?

¿Cuantos caracteres reporta el programa? ¿Considera que es correcto? ¿Por qué?

Realizar la misma prueba en una consola Javascript (En Google Chrome presionar F12, seleccione el menú Console y ejecute `"\u1F355".length`. ¿Obtiene el mismo resultado?

E7. La sección 2.2 título *Efficiency* del [UNICODE9] se lee:

> All Unicode encoding forms are self-synchronizing and non-overlapping. This makes randomly accessing and searching inside streams of characters efficient.

Siendo `UTF-8` una forma de codificar Unicode. ¿Por qué una codificación que se auto sincroniza es una ventaja en un acceso aleatorio de datos?

E8. ¿En qué consisten la discusión de *Little-Endian contra Big-Endian*?

E9. Cuando decimos que un archivo es de *texto plano* o un archivo binarios, ¿de qué estamos hablando?

Brinde ejemplos de cada cada categoría.

En el caso de archivos de *texto plano*: provea ejemplos donde la codificación de caracteres sea `ASCII`, donde sea `UTF-8`, y donde el mismo formato permite especificar la codificación dentro del mismo archivo. En este último caso describa como lo hace.

E10. ¿De qué técnicas disponemos al momento de serializar estructuras de datos en archivos?

Puede tomar de base para la discusión el siguiente extracto de código C:

```
#include <time.h>

typedef enum {
   TRACE,
   DEBUG,
   INFO,
   WARN,
   ERROR,
} level;

struct logentry {
   time_t when;
   char   *msg;
   level  l;
};
```

¿De qué técnicas dispone para codificar tipo de datos de ancho variables (como por ejemplo ser un *string* o una lista variables de elementos)?

En particular al codificar un "*string*" de ancho variable. ¿Qué ventaja presenta una codificación *Null-terminated* contra una codificación *chunked*?

Nota

En un contexto donde estamos limitados a caracteres ASCII por ejemplo podríamos entender una codificación *chunked* a una codificación tal que provee primero la cantidad de bytes del string y luego provee los bytes.

La cantidad de bytes a su vez podría estar codificar con un ancho fijo (ej: 16 bits) o con una codificación null-terminated.

Ejercicios de Diseño/Programación

E11. Dado la siguiente interfaz Java que especifica una librería de reporte de mensajes de error y progreso

```
/** Registra mensajes */
public interface Log {
    /** registra un mensaje con el nivel warning */
    void warning(String msg);

    /** registra un mensaje con el nivel info */
    void info(String msg);

    /** dado un input stream consume uno a uno los mensajes */
    void read(java.io.InputStream input,
              final java.util.function.Consumer<Message> consumer);

    enum Level {
        warning,
        info,
    }

    interface Message {
        java.util.Date when();
        String getMessage();
        Level getLevel();
    }
}
```

1. Diseñe un formato *de archivo* para poder almacenar en disco los mensajes y que luego los poderlos leer de forma secuencial.

   No se olvide de describir el domino de caracteres que se pueden utilizar y qué verificaciones se deben realizar (por ejemplo ¿existen tamaños máximos de mensajes?)

El formato de archivo debe poderse implementar en múltiples arquitecturas, y en múltiples lenguajes de programación.

2. Describa un documento de texto formalmente dicho formato, ya sea utilizando ABNF [RFC5234] o algún otro mecanismo similar.

   La intención es que con dicho documento otro desarrollador pueda realizar una implementación compatible con su formato, aún utilizando otro lenguaje de programación.

3. Escriba dos programas en algún lenguaje (C/Java) que utilizando el formato anterior uno que genere registros y termine; y otro que lea dichos mensajes y los imprima a la salida estándar.

Lecturas recomendadas

[RFC20] *ASCII format for Network Interchange [https://tools.ietf.org/html/rfc20]*. Vint Cerf. The Internet Engineering Task Force. October 16, 1969.

[IEN137] *On holy wars and plea for peace [https://www.ietf.org/rfc/ien/ien137.txt]*. Danny Cohen. The Internet Engineering Task Force. 1 April 1980.

"The root of the conflict lies much deeper than that. It is the question of which bit should travel first, the bit from the little end of the word, or the bit from the big end of the word? The followers of the former approach are called the Little-Endians, and the followers of the latter are called the Big-Endians. "

[RFC2978/BCP19] *IANA Charset Registration Procedures [https://tools.ietf.org/html/rfc2978]*. Ned Freed y Jon Postel. The Internet Engineering Task Force. October 2000.

Multipurpose Internet Mail Extensions (MIME) and various other Internet protocols are capable of using many different charsets. This in turn means that the ability to label different charsets is essential. This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements.

[RFC3629] *UTF-8, a transformation format of ISO 10646 [https://tools.ietf.org/html/rfc3629]*. F. Yergeau. The Internet Engineering Task Force. November 2003.

ISO/IEC 10646-1 defines a large character set called the Universal Character Set (UCS) which encompasses most of the world's writing systems. The originally proposed encodings of the UCS, however, were not compatible with many current applications and protocols, and this has led to the development of UTF-8, the object of this memo. UTF-8 has the characteristic of preserving the full US-ASCII range, providing compatibility with file systems, parsers and other software that rely on US-ASCII values but are transparent to other values. This memo obsoletes and replaces RFC 2279.

[RFC4648] *The Base16, Base32, and Base64 Data Encodings [https://tools.ietf.org/html/rfc4648]*. Simon Josefsson. The Internet Engineering Task Force. October 2006.

This document describes the commonly used base 64, base 32, and base 16 encoding schemes. It also discusses the use of line-feeds in encoded data, use of padding in encoded data, use of non-alphabet characters in encoded data, use of different encoding alphabets, and canonical encodings. [STANDARDS-TRACK]

[RFC5234] *Augmented BNF for Syntax Specifications: ABNF [https://tools.ietf.org/html/rfc5234]*. Dave Crocker y Paul Overell. The Internet Engineering Task Force. January 2008.

Internet technical specifications often need to define a formal syntax. Over the years, a modified version of Backus-Naur Form (BNF), called Augmented BNF (ABNF), has been popular among many Internet specifications. The current specification documents ABNF. It balances compactness and simplicity with reasonable representational power. The differences between standard BNF and ABNF involve naming rules, repetition, alternatives, order-independence, and value ranges. This specification also supplies additional rule definitions and encoding for a core lexical analyzer of the type common to several Internet specifications. [STANDARDS-TRACK]

[UNICODE9] *Unicode Standard, Version 9.0.0 [http://www.unicode.org/versions/Unicode9.0.0/]. Capítulo 1, y Capítulo 2 (de la sección 2.4 a la sección 2.7)*. The Unicode Consortium. July 2016.

The Unicode Standard and its associated specifications provide programmers with a single universal character encoding, extensive descriptions, and a vast amount of data about how characters function. The specifications and data describe how to form words and break lines; how to sort text in different languages; how to format numbers, dates, times, and other elements appropriate to different languages; how to display languages whose written form flows from right to left, such as Arabic and Hebrew, or whose written form splits, combines, and reorders, such as languages of South Asia. These specifications include descriptions of how to deal with security concerns regarding the many “look-alike” characters from alphabets around the world. Without the properties and algorithms in the Unicode Standard and its associated specifications, interoperability between different implementations would be impossible, and much of the vast breadth of the world’s languages would lie outside the reach of modern software.

Direccionamiento y HTTP

Revisión: 2021-07-26 22:34:27

Direccionamiento

E12. En la siguiente URI `https://tools.ietf.org/html/rfc2616#section-14.19` ¿Cómo se llama la parte "`section-14.19`"? ¿Se envía por la red al hacer el request? ¿Por qué? Piense usos.

E13. ¿Cuál es la diferencia entre una URI y una URI reference?

E14. Dada la URI base `http://a/b/c/d;p?q` resuelva las referencias enumeradas en la sección 5.4.1. del [RFC3986]

E15. ¿Según la interpretación que tiene HTTP sobre las URIs: las siguientes URL son equivalentes?

a. `http://abc.com:80/~smith/home.html`
b. `http://ABC.com/%7Esmith/home.htm`
c. `http://ABC.com:/%7esmith/home.htm`

E16. En el siguiente elemento XML ¿Cuales de las siguientes URL tienes sentido que sean propiamente URL y cuales pueden ser URNs?

```
<foo xmlns="http://www.springframework.org/schema/beans"
     xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
     xmlns:util="http://www.springframework.org/schema/util"
     xsi:schemaLocation="http://www.springframework.org/schema/beans
     http://www.springframework.org/schema/beans/springbeans.xsd
     http://www.springframework.org/schema/util
     http://www.springframework.org/schema/util/springutil.xsd">
```

netcat

E17. Comunique dos terminales utilizando **netcat**.

E18. {W} HTTP/1.1 y otros protocolos utilizan las secuencias `CR LF` como marcas de fin de línea. ¿Como usted puede lograr enviar esta secuencia con netcat desde una terminal donde se consumen caracteres desde la entrada estándar?

E19. {W} Con netcat escuche en el puerto TCP 9090. Ingrese la URL `http://localhost:9090/` en un *User Agent*:

a. Analizar detalladamente la estructura del request y en particular los headers que el user agent envió
b. Desde el server (netcat) retorne una respuesta que no siga las reglas de `HTTP`. ¿Qué sucede en el *user agent*?

c. Vuelva a realizar el request y desde el server (netcat) retorne una respuesta de formato válido. Verifique comportamiento en el *user agent*.

E20. {W} Con netcat conéctese a `www.google.com.ar` al puerto 80, y envíe de header `GET / HTTP/1.1\r\n\r\n` (sin ningún otro header).

a. ¿Qué código de respuesta retorna el servidor? ¿Qué significa? ¿Qué header es requerido que esté presente en dicho código?

b. Realizar los cambios necesarios en el request para que el servidor envíe como representación a nuestro pedido el formulario de búsqueda (es un html).

c. Analizar todos los headers que aparecen en la respuesta y su impacto en los UA (Ejemplo: `Date`, `Cache-Control`, `Content-Type`, `Set-Cookie`).

E21. Investigue el comando **curl** y el comando **wget**. ¿En que situación utilizaría cada una?. Preste atención a las opciones `-0 --http1.1 --http2 --compressed -d -H -i -s --socks5` del comando **curl**.

E22. {W} Con la utilidad wget descargue un archivo "*grande*". En el medio de la descarga cancele la misma; y luego intente resumir la descarga (`--continue`).

¿Qué headers participan para resumir las descargas?

¿Funciona con todas las páginas? Si no funciona; busque un contra ejemplo.

HTTP: Funcionalidades que provee el protocolo

E23. Los siguientes headers HTTP... ¿Son equivalentes? `Accept: text/plain` y `AccEpT: text/plain`

E24. 1. ¿Qué significa en la respuesta del servidor un header `Transfer-Encoding: chunked`?

2. ¿Qué otras codificaciones existen?

3. ¿Para que sirven?

4. ¿En que ocasiones utilizaría cada uno?

E25. a. ¿Qué es un media type?

b. ¿Para que se usa?

c. ¿Cómo está estructurado?

d. ¿Cuales son los media type asociados a …?

i. Páginas web (ej: html)

ii. Archivos de texto plano

iii. Estilos de páginas webs (CSS)

iv. Javascript (archivos de extensión .js)

v. Imágenes jpeg, gif, png

vi. archivos pdf

vii. un archivo .exe o de formato no estandar

E26. a. ¿Cuales son los requisitos para que un cliente pueda utilizar el esquema de conexiones persistentes de HTTP?
b. ¿Qué ventajas brinda frente al esquema de pedidos tradicional?
c. ¿De qué se trata el pipelining?

E27. a. ¿Qué métodos provee HTTP?
b. ¿Qué significa que un método es idempotente?
c. Piense del efecto de la idempotencia en el siguiente caso:

> Se cuenta con una extensión al user-agent (ej: plugin de Firefox) que una vez cargada las página web (de formato HTML) el mismo recolecta todos los links en el documento, y con el objetivo de acelerar la experiencia del usuario, comienza a descargar todos los links recolectados. De esta forma el usuario cuando siga algún link de interés no tendrá que esperar. Suponga que el usuario visita un sitio web que tiene una página web que presenta un listado de cosas, y cada ítem, además de presentar su nombre, presenta un link para editar su contenido, y otro link para eliminar el mismo (es decir el clásico ABM). El desarrollador decidió que realizando un GET a los link de eliminación (`/items/123/delete`).

- ¿Qué sucederá cuando el usuario que tiene el acelerador de internet activado acceda al ABM?
- ¿Quién tiene la culpa de ese comportamiento?
- ¿Como se podría solucionar el problema?

d. ¿Por qué se dice que `DELETE` es idempotente si es que el método borra el recurso de la misma forma que el comando **rm** borra un archivo?
e. ¿Cuando utilizaría `POST`? (Además del RFC, revise [whenToUseGet])

E28. Utilice el método `TRACE` contra www.google.com.ar para detectar si hay un proxy entre su computadora y el servidor de Google. Si ocurre algún error elabore una hipótesis de por que sucede.

E29. ¿Que código de retorno utilizaría desde su aplicación web para las siguientes situaciones?
a. El recurso al que se está accediendo ya no se encuentra en la dirección a la que se está intentando acceder. Se conoce cual es la nueva dirección que se debe utilizar a partir de este momento.
b. El cliente no tiene permiso para acceder a la página (ejemplo un usuario anónimo encontró cual es la URL del backoffice de administración)
c. El cliente no se encuentra autenticado
d. El recurso que se está editando (por ejemplo una entrada de un blog) fue editado desde el momento que nosotros lo comenzamos a editar, y apretamos el botón guardar (no queremos perder los cambios recién guardados)
e. ¿Cuando usaría el código 410? ¿En qué se diferencia en su opinión del 404?
f. Faltó en la URL de un `GET` un parámetro (ej `/foo/bar?param=123`)

g. El formato de un parámetro es incorrecto (ej `/foo/bar?param=abc` donde `param` debía ser un número entero).
h. Se está intentando subir un recurso cuyo formato no es soportado en el servidor
i. El recurso al que se está intentando acceder requiere de otro sistema externo que no se encuentra por el momento disponible (ej: está caído).
j. El cliente está enviando un `DELETE` al recurso pero esa operación no es soportada (solo soporta el `GET`).

E30. Si se encuentra snifeando la red y se encuentra con el siguiente pedido:

```
GET / HTTP/1.1
Authorization: Basic YWxndW51c3VhcmlvOmFsZ3VuYXBhc3N3b3Jk
```

a. ¿De qué se trata el header `Authorization`?
b. ¿Podría recuperarse el usuario y la password? Si es así...¿cuales son?

E31. ¿Qué estrategias provee HTTP para minimizar el tráfico de red y la utilización de recursos del servidor?

HTTP: Entity tags, y caching

E32. {W} Con la herramienta **cURL**, y el recurso `http://foo.leak.com.ar/` responda las siguientes preguntas:
a. Haga un `GET` condicional utilizando la fecha de última modificación
b. Haga un `GET` condicional utilizando los "*entity tags*"

E33. ¿Qué interpretará un UA que tiene soporte de caching si recibe en una respuesta el header `Cache-Control: max-age=3600, must-revalidate`?

E34. Si hablamos de *Shallow Etag*... ¿De qué hablamos? [deep-tag]

E35. {W} Acceda utilizando Google Chrome a https://campus.itba.edu.ar/.

a. ¿Cómo se envía los datos del formulario? ¿Cómo está codificado?
b. Una vez que está logueado haga click en la materia Protocolos de Comunicación. Si HTTP busca no tener estado...¿Como hace el servidor para saber que es usted el que está haciendo el pedido?
c. Haga el pedido a esta URL usando el cURL ¿Visualiza el mismo contenido que en el browser? ¿Por qué?
d. Utilizando cURL, obtenga el HTML visualiza el desde el browser.

Nota

Debido a que HTTP se trasporta utilizando TLS si utiliza un sniffer no podrá analizar el tráfico. Sin embargo puede utilizar las *Developers Tools* (**F12**/ Network) para analizarlo.

HTTP: Configuración de servidor http, virtual hosts

Instalar un servidor nginx.

E36. El servidor proveerá hosting para el sitio web `foo` y para el sitio `bar`.

- Cuando un usuario se conecta a `foo` debe ver el mensaje "*Bienvenido a Foo*"
- Cuando un usuario se conecta a `bar` debe ver el mensaje "*Bienvenido a Bar*"
- Cuando un usuario se conecta con cualquier otro nombre al servidor (como ser utilizando sólo la dirección `IP`) el usuario debe ver el mensaje "*What?*"

Aviso

Podría encontrarse con un problema al acceder http://foo/ desde un user agent avanzado (como Chrome o Firefox). Investigue a que se debe.

E37. ¿Como prueba que el servidor está bien configurado aún cuando su computadora no resuelva los nombre `foo` y `bar`?

¡Comprobarlo!

Luego haga los cambios necesarios en el archivo `/etc/hosts` [hosts(5)] para poder acceder con el navegador.

E38. {W} Analizando los headers de pedido y respuesta, investigar las diferencias que hay al acceder desde un browser a un recurso versus recargar dicho recurso. Para la recarga discriminar entre el uso de la tecla F5 y la combinación de teclas `Ctrl+F5` (en Firefox).

E39. Proxy reverso:

a. Descargue la última versión del Apache Tomcat y pongalo a correr (descomprimir, y ejecutar **bin/startup.sh**)

b. Verificar que está corriendo `http://localhost:8080/`

c. Haga los cambios necesarios en el nginx para que cuando acceda al host `foo` ve el contenido servido por el Tomcat

d. ¿Qué ventajas piensa que tiene éste esquema de despliegue?

> A "gateway" (a.k.a. "reverse proxy") is an intermediary that acts as an origin server for the outbound connection but translates received requests and forwards them inbound to another server or servers. Gateways are often used to encapsulate legacy or untrusted information services, to improve server performance through "accelerator" caching, and to enable partitioning or load balancing of HTTP services across multiple machines.
>
> —RFC7230 2.3 Intermediaries

E40. En el sitio `foo` y en el `bar` configure el nginx para que comprima de forma transparente los recursos que sirve.

a. ¿Que ventajas tiene este esquema?

b. ¿Afecta la performance?

c. ¿Salva ancho de banda y aumenta positivamente la experiencia de usuario?

d. `foo` estaba sirviendo contenido que era traído desde tomcat. Vuelva a pensar sobre las ventajas de éste esquema de despliegue.

Lecturas recomendadas

[DesignIssues] *Axioms of Web Architecture: 0 [https://www.w3.org/DesignIssues/Model]* . Tim Berners-Lee. World Wide Web Consortium. January 1998.

[Addressing] *Naming and Addressing: URIs, URLs, ... [https://www.w3.org/Addressing/]* . Tim Berners-Lee. World Wide Web Consortium. 1993.

[URI-Style] *Cool URIs don't change [https://www.w3.org/Provider/Style/URI]* . Tim Berners-Lee. World Wide Web Consortium. 1998.

[CURIE-S] *CURIE Syntax 1.0: A syntax for expressing Compact URIs [https://www.w3.org/TR/curie/]* . W3C Working Group Note. World Wide Web Consortium. 16 December 2010.

[selfDescribingDocuments] *The Self-Describing Web [https://www.w3.org/2001/tag/doc/selfDescribingDocuments]* . . World Wide Web Consortium. 07 February 2009.

[http-perf96] *Analysis of HTTP Performance [http://www.isi.edu/touch/pubs/http-perf96/]* . Joe Touch, , y . Information Sciences Institute. Aug. 16, 1996.

We discuss the performance effects of using per-transaction TCP connections for HTTP access, and the proposed optimizations of avoiding per-transaction re-connection and TCP slow-start restart overheads. We analyze the performance penalties of the interaction of HTTP and TCP. Our observations indicate that the proposed optimizations do not substantially affect Web access for the vast majority of users, who typically see end-to-end latencies of 100-250 ms and use low bandwidth lines. Under these conditions, there are only 1-2 packets in transit between the client and server, and the optimizations reduce the overall transaction time by only 11%. Rates over 200 Kbps are required in order to achieve at least a 50% reduction in transaction time, resulting in a user-noticeable performance enhancement.

[whenToUseGet] *URIs, Addressability, and the use of HTTP GET and POST [http://www.w3.org/2001/tag/doc/whenToUseGet.html]* . . World Wide Web Consortium. 21 March 2004.

An important principle of Web architecture is that all important resources be identifiable by URI. The finding discusses the relationship between the URI addressability of a resource and the choice between HTTP GET and POST methods with HTTP URIs. HTTP GET

promotes URI addressability so, designers should adopt it for safe operations such as simple queries. POST is appropriate for other types of applications where a user request has the potential to change the state of the resource (or of related resources). The finding explains how to choose between HTTP GET and POST for an application taking into account architectural, security, and practical considerations.

This finding does not discuss URI schemes other than "http" or protocols other than HTTP/1.1 [RFC2616].

[RFC2046] *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types [https://tools.ietf.org/html/rfc2046]* . N. Freed y N. Borenstein. The Internet Engineering Task Force. November 1996.

This second document defines the general structure of the MIME media typing system and defines an initial set of media types.

[RFC7230] *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing [https://tools.ietf.org/html/rfc7230]* . R. Fielding y J. Reschke. The Internet Engineering Task Force. June 2014.

The Hypertext Transfer Protocol (HTTP) is a stateless application- level protocol for distributed, collaborative, hypertext information systems. This document provides an overview of HTTP architecture and its associated terminology, defines the "http" and "https" Uniform Resource Identifier (URI) schemes, defines the HTTP/1.1 message syntax and parsing requirements, and describes related security concerns for implementations.

[RFC3987] *Internationalized Resource Identifiers (IRIs) [https://tools.ietf.org/html/rfc3987]* . M. Duerst y M. Suignard. The Internet Engineering Task Force. January 2005.

This document defines a new protocol element, the Internationalized Resource Identifier (IRI), as a complement of the Uniform Resource Identifier (URI). An IRI is a sequence of characters from the Universal Character Set (Unicode/ISO 10646). A mapping from IRIs to URIs is defined, which means that IRIs can be used instead of URIs, where appropriate, to identify resources. The approach of defining a new protocol element was chosen instead of extending or changing the definition of URIs. This was done in order to allow a clear distinction and to avoid incompatibilities with existing software. Guidelines are provided for the use and deployment of IRIs in various protocols, formats, and software components that currently deal with URIs.

[RFC3986] *Uniform Resource Identifier (URI): Generic Syntax [https://tools.ietf.org/html/rfc3986]* . Roy Fielding y Larry Masinter. The Internet Engineering Task Force. January 2005.

[RFC6266] *Use of the Content-Disposition Header Field in the Hypertext Transfer Protocol (HTTP) [https://tools.ietf.org/html/rfc6266]* . Reschke, J.. The Internet Engineering Task Force. June 2011.

RFC 2616 defines the Content-Disposition response header field, but points out that it is not part of the HTTP/1.1 Standard. This specification takes over the definition and regis-

tration of Content-Disposition, as used in HTTP, and clarifies internationalization aspects. [STANDARDS-TRACK]

[RFC6585] *Additional HTTP Status Codes [https://tools.ietf.org/html/rfc6585]* . M. Nottingham y R. Fielding. The Internet Engineering Task Force. April 2012.

This document specifies additional HyperText Transfer Protocol (HTTP) status codes for a variety of common situations. [STANDARDS-TRACK]

[RFC7540] *Hypertext Transfer Protocol Version 2 (HTTP/2) [https://tools.ietf.org/html/rfc7540]* . Mike Belshe y Roberto Peon. The Internet Engineering Task Force. May 2015.

This specification describes an optimized expression of the semantics of the Hypertext Transfer Protocol (HTTP), referred to as HTTP version 2 (HTTP/2). HTTP/2 enables a more efficient use of network resources and a reduced perception of latency by introducing header field compression and allowing multiple concurrent exchanges on the same connection. It also introduces unsolicited push of representations from servers to clients. This specification is an alternative to, but does not obsolete, the HTTP/1.1 message syntax. HTTP's existing semantics remain unchanged.

[mnot_cache] *Caching Tutorial for Web Authors and Webmasters [https://www.mnot.net/cache_docs/]* . Mark Nottingham. 6 May, 2013.

This is an informational document. Although technical in nature, it attempts to make the concepts involved understandable and applicable in real-world situations. Because of this, some aspects of the material are simplified or omitted, for the sake of clarity. If you are interested in the minutia of the subject, please explore the References and Further Information at the end.X

[deep-tag] *REST Tip: Deep etags give you more benefits [http:///bitworking.org/news/150/REST-Tip-Deep-etags-give-you-more-benefits]* . Joe Gregorio. 2007-03-22.

ETags, or entity-tags, are an important part of HTTP, being a critical part of caching, and also used in "conditional" requests. So what is an etag?

[REST] *Architectural Styles and the Design of Network-based Software Architectures [http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm]* . Roy Fielding. University of California, Irvine. 2000.

The World Wide Web has succeeded in large part because its software architecture has been designed to meet the needs of an Internet-scale distributed hypermedia system. The Web has been iteratively developed over the past ten years through a series of modifications to the standards that define its architecture. In order to identify those aspects of the Web that needed improvement and avoid undesirable modifications, a model for the modern Web architecture was needed to guide its design, definition, and deployment.

Software architecture research investigates methods for determining how best to partition a system, how components identify and communicate with each other, how information is

communicated, how elements of a system can evolve independently, and how all of the above can be described using formal and informal notations. My work is motivated by the desire to understand and evaluate the architectural design of network-based application software through principled use of architectural constraints, thereby obtaining the functional, performance, and social properties desired of an architecture. An architectural style is a named, coordinated set of architectural constraints.

This dissertation defines a framework for understanding software architecture via architectural styles and demonstrates how styles can be used to guide the architectural design of network-based application software. A survey of architectural styles for network-based applications is used to classify styles according to the architectural properties they induce on an architecture for distributed hypermedia. I then introduce the Representational State Transfer (REST) architectural style and describe how REST has been used to guide the design and development of the architecture for the modern Web.

REST emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems. I describe the software engineering principles guiding REST and the interaction constraints chosen to retain those principles, contrasting them to the constraints of other architectural styles. Finally, I describe the lessons learned from applying REST to the design of the Hypertext Transfer Protocol and Uniform Resource Identifier standards, and from their subsequent deployment in Web client and server software.

[URI-2001] *URIs, URLs, and URNs: Clarifications and Recommendations 1.0 [https://www.w3.org/TR/uri-clarification/]* . W3C/IETF URI Planning Interest Group. 21 September 2001.

This paper addresses and attempts to clarify two issues pertaining to URIs, and presents recommendations. Section 1 addresses how URI space is partitioned and the relationship between URIs, URLs, and URNs. Section 2 describes how URI schemes and URN namespace ids are registered. Section 3 mentions additional unresolved issues not considered by this paper and section 4 presents recommendations.

[LinkedDataTutorial] *How to Publish Linked Data on the Web [http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/]* . 2007-07-11.

[LinkedData] *Linked Data [https://www.w3.org/DesignIssues/LinkedData.html]* . World Wide Web Consortium. 2006-07-27.

[LinkedData2] *Linked Data [https://www.w3.org/standards/semanticweb/data]* . World Wide Web Consortium.

[nginx] *nginx documentation [http://nginx.org/en/docs/]* . nginx.org.

[ReverseProxy] *Nginx/ReverseProxy [https://help.ubuntu.com/community/Nginx/ReverseProxy]* . nginx.org.

[nc(1)] *nc — arbitrary TCP and UDP connections and listens*. BSD General Commands Manual. February 7, 2012.

The nc (or netcat) utility is used for just about anything under the sun involving TCP, UDP, or UNIX-domain sockets. It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scan# ning, and deal with both IPv4 and IPv6. Unlike telnet(1), nc scripts nicely, and separates error messages onto standard error instead of send# ing them to standard output, as telnet(1) does with some.

[curl(1)] *curl - transfer a URL*. November 30, 2014.

curl is a tool to transfer data from or to a server, using one of the supported protocols (DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET and TFTP). The command is designed to work without user interaction.

curl offers a busload of useful tricks like proxy support, user authen# tication, FTP upload, HTTP post, SSL connections, cookies, file trans# fer resume, Metalink, and more. As you will see below, the number of features will make your head spin!

[wget(1)] *Wget - The non-interactive network downloader*. June 14, 2016.

GNU Wget is a free utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies.

Wget is non-interactive, meaning that it can work in the background, while the user is not logged on. This allows you to start a retrieval and disconnect from the system, letting Wget finish the work. By contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data.

[hosts(5)] *hosts - static table lookup for hostnames*. Linux Programmer's Manual. March 29, 2015.

This manual page describes the format of the /etc/hosts file. This file is a simple text file that associates IP addresses with hostnames, one line per IP address.

Domain Name System

Revisión: 2021-07-26 22:34:27

Glosario

Las siguientes definiciones fueron tomadas del [RFC7719], sección 2, 4 y 6

Domain name	Section 3.1 of [RFC1034] talks of "*the domain name space*" as a tree structure. "Each node has a label, which is zero to 63 octets in length. ... The domain name of a node is the list of the labels on the path from the node to the root of the tree. ... To simplify implementations, the total number of octets that represent a domain name (i.e., the sum of all label octets and label lengths) is limited to 255. Any label in a domain name can contain any octet value.
Fully qualified domain name (FQDN)	This is often just a clear way of saying the same thing as "domain name of a node", as outlined above. However, the term is ambiguous. Strictly speaking, a fully qualified domain name would include every label, including the final, zero-length label of the root: such a name would be written "`www.example.net.`" (note the terminating dot). But because every name eventually shares the common root, names are often written relative to the root (such as "`www.example.net`") and are still called "fully qualified". This term first appeared in [RFC819].
Label	The identifier of an individual node in the sequence of nodes identified by a fully qualified domain name.
Alias	The owner of a CNAME resource record, or a subdomain of the owner of a DNAME resource record [RFC6672]. See also "canonical name".
Canonical name	A CNAME resource record "*identifies its owner name as an alias, and specifies the corresponding canonical name in the RDATA section of the RR.*" (Quoted from [RFC1034], Section 3.6.2) This usage of the word "canonical" is related to the mathematical concept of "canonical form".
CNAME	"*It is traditional to refer to the owner of a CNAME record as 'a CNAME'. This is unfortunate, as 'CNAME' is an abbreviation of 'canonical name', and the owner of a CNAME record is an alias, not a canonical name.*" (Quoted from [RFC2181], Section 10.1.1)
RR	An acronym for resource record. ([RFC1034] Section 3.6.)

RRset

A set of resource records with the same label, class and type, but with different data. (Definition from [RFC2181]) Also spelled RRSet in some documents. As a clarification, "same label" in this definition means "same owner name". In addition, [RFC2181] states that "the TTLs of all RRs in an RRSet must be the same". (This definition is definitely not the same as "the response one gets to a query for `QTYPE=ANY`", which is an unfortunate misunderstanding.)

TTL

The maximum "time to live" of a resource record. "*A TTL value is an unsigned number, with a minimum value of 0, and a maximum value of `2147483647`. That is, a maximum of `2^31 - 1`. When transmitted, the TTL is encoded in the less significant 31 bits of the 32 bit TTL field, with the most significant, or sign, bit set to zero.*" (Quoted from [RFC2181], Section 8) (Note that [RFC1035] erroneously stated that this is a signed integer; that was fixed by [RFC2181].)

The TTL "*specifies the time interval that the resource record may be cached before the source of the information should again be consulted*". (Quoted from [RFC1035], Section 3.2.1) Also: "*the time interval (in seconds) that the resource record may be cached before it should be discarded*". (Quoted from [RFC1035], Section 4.1.3). Despite being defined for a resource record, the TTL of every resource record in an RRset is required to be the same ([RFC2181], Section 5.2).

The reason that the TTL is the maximum time to live is that a cache operator might decide to shorten the time to live for operational purposes, such as if there is a policy to disallow TTL values over a certain number. Also, if a value is flushed from the cache when its value is still positive, the value effectively becomes zero. Some servers are known to ignore the TTL on some RRsets (such as when the authoritative data has a very short TTL) even though this is against the advice in [RFC1035].

There is also the concept of a "default TTL" for a zone, which can be a configuration parameter in the server software. This is often expressed by a default for the entire server, and a default for a zone using the $TTL directive in a zone file. The $TTL directive was added to the master file format by [RFC2308].

Apex

The point in the tree at an owner of an `SOA` and corresponding authoritative NS RRset. This is also called the "*zone apex*". [RFC4033] defines it as "*the name at the child's side of a zone cut*". The "apex" can usefully be thought of as a data-theoretic

> description of a tree structure, and "origin" is the name of the same concept when it is implemented in zone files. The distinction is not always maintained in use, however, and one can find uses that conflict subtly with this definition. [RFC1034] uses the term "top node of the zone" as a synonym of "apex", but that term is not widely used. [...]

Clientes DNS

E41. *{W}* Utilizando herramientas de línea de comandos determine cuál es el servidor de nombres que es autoridad para cada uno de los `FQDN`:

- `google.com.`
- `itba.edu.ar.`
- `pampero.it.itba.edu.ar.`

Revise los paquetes enviados, y sus respuestas.

E42. Determine cuales son los servidores que manejan el correo electrónico entrante para cada uno de los siguientes dominios FQDNS:

- `google.com.`
- `itba.edu.ar.`
- `it.itba.edu.ar.`

E43. ¿Cuales son todos los servidores de nombre que entran en juego para resolver `pampero.it.itba.edu.ar` partiendo desde un *root server*?

E44. **dig** tiene una opción `trace`. ¿para qué sirve dicha opción? Contrastar su resultado con el punto anterior.

E45. *{W}* Utilizar **whois** para obtener información de los dominios: `clarin.com`, `google.com`, `facebook.com`, `lanacion.com.ar`, `itba.edu.ar`, `apple.com`.

DNS y HTTP

E46. Realizar los cambios necesarios para que al ingresar en un *User Agent HTTP* (**curl**, Google Chrome, ...) a `http://foo.pdc.lab`, en realidad se acceda `http://foo.leak.com.ar/` (no se debe ver modificada la URL en el browser). ¿Qué sucede y por qué?

Aviso

Puede obtener diferentes resultados si realiza las pruebas conectado desde la red del laboratorio y fuera de ésta. Se requiere una solución estable.

E47. Suponga que en el ejercicio anterior se permite que la URL en el browser pueda ser modificada. ¿Puede encontrar otra solución?

Configuración de DNS Server

Se quiere configurar un servidor DNS que sea autoridad de la zona `foo.pdc.lab`. Dicha zona deberá mantener nombres para su host y algunos otros hosts de la red en la que se encuentra. Configurar también un alias para cada uno de estos hosts.

E48. De existir, borrar los archivos `/etc/named.caching-server.conf`, `/var/named/chroot/etc/named.caching-server.conf`.

E49. Crear la zona correspondiente a su dominio (forward mapping zone y reverse zone). La configuración de named se encuentra en `/etc/bind`. Establezca una variedad de registros, con diferentes TTL.

E50. Indicarle al daemon que reinicie. Observar el mensaje almacenado en el log, elaborar conclusiones en base al mismo.

E51. Comprobar usando las utilidades **dig**, **host** y **nslookup** que su dominio esté configurado correctamente.

E52. Asegurarse que su host esté utilizando al servidor DNS configurado como su servidor DNS primario

E53. {W} Investigar en named que son los forwarders. Sniffear paquetes haciendo uso y sin hacer uso de forwarders para ver los paquetes que generan el servidor DNS cuando le hacen consultas.

Bibliografía

[RFC819] *The Domain Naming Convention for Internet User Applications, [https://tools.ietf.org/html/rfc819]*. The Internet Engineering Task Force. August 1982.

This RFC is an attempt to clarify the generalization of the Domain Naming Convention, the Internet Naming Convention, and to explore the implications of its adoption for Internet name service and user applications.

[RFC1033] *Domain Administrators Operations Guide [https://tools.ietf.org/html/rfc1033]*. The Internet Engineering Task Force. November 1987.

This RFC provides guidelines for domain administrators in operating a domain server and maintaining their portion of the hierarchical database. Familiarity with the domain system is assumed

[RFC1034] *Domain names - concepts and facilities [https://tools.ietf.org/html/rfc1034]*. The Internet Engineering Task Force. November 1987.

This RFC is the revised basic definition of The Domain Name System. It obsoletes RFC-882. This memo describes the domain style names and their used for host address look up and electronic mail forwarding. It discusses the clients and servers in the domain name system and the protocol used between them.

[RFC1035] *Domain names - implementation and specification [https://tools.ietf.org/html/rfc1035]*. The Internet Engineering Task Force. November 1987.

This RFC is the revised specification of the protocol and format used in the implementation of the Domain Name System. It obsoletes RFC-883. This memo documents the details of the domain name client - server communication.

[RFC2181] *Clarifications to the DNS Specification [https://tools.ietf.org/html/rfc2181]*. The Internet Engineering Task Force. July 1997.

This document considers some areas that have been identified as problems with the specification of the Domain Name System, and proposes remedies for the defects identified.

[RFC2308] *Negative Caching of DNS Queries [https://tools.ietf.org/html/rfc2308]*. The Internet Engineering Task Force. March 1998.

RFC1034 provided a description of how to cache negative responses. It however had a fundamental flaw in that it did not allow a name server to hand out those cached responses to other resolvers, thereby greatly reducing the effect of the caching. This document addresses issues raise in the light of experience and replaces RFC1034 Section 4.3.4.

[RFC4033] *DNS Security Introduction and Requirements [https://tools.ietf.org/html/rfc4033]*. The Internet Engineering Task Force. March 2005.

The Domain Name System Security Extensions (DNSSEC) add data origin authentication and data integrity to the Domain Name System. This document introduces these extensions and describes their capabilities and limitations. This document also discusses the services that the DNS security extensions do and do not provide. Last, this document describes the interrelationships between the documents that collectively describe DNSSEC.

[RFC6672] *DNAME Redirection in the DNS [https://tools.ietf.org/html/rfc6672]*. The Internet Engineering Task Force. June 2012.

The DNAME record provides redirection for a subtree of the domain name tree in the DNS. That is, all names that end with a particular suffix are redirected to another part of the DNS. This document obsoletes the original specification in RFC 2672 as well as updates the document on representing IPv6 addresses in DNS (RFC 3363)

[RFC7719] *DNS Terminology [https://tools.ietf.org/html/rfc7719]*. The Internet Engineering Task Force. DECEMBER 2015.

The DNS is defined in literally dozens of different RFCs. The terminology used by implementers and developers of DNS protocols, and by operators of DNS systems, has so-

metimes changed in the decades since the DNS was first defined. This document gives current definitions for many of the terms used in the DNS in a single document.

[RFC3912] *WHOIS Protocol Specification [https://tools.ietf.org/html/rfc3912]*. The Internet Engineering Task Force. September 2004.

This document updates the specification of the WHOIS protocol, thereby obsoleting RFC 954. The update is intended to remove the material from RFC 954 that does not have to do with the on-the-wire protocol, and is no longer applicable in today's Internet. This document does not attempt to change or update the protocol per se, or document other uses of the protocol that have come into existence since the publication of RFC 954.

[ISC-DNS] *DNS RFC [https://www.isc.org/community/rfcs/dns/]*. Internet Systems Consortium.

The Domain Name System protocols are more than 20 years old, and many of the older RFCs are obsolete, but there still exist clients running software implementing the very oldest protocols. Here are the RFCs pertaining to DNS.

[domainsroot] *Domain Name Services [https://www.iana.org/domains]*. Internet Assigned Numbers Authority.

IANA is responsible for management of the DNS root zone. This role means assigning the operators of top-level domains, such as .uk and .com, and maintaining their technical and administrative details.

[dig(1)] *DNS lookup utility*. BIND9.

dig (domain information groper) is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. Most DNS administrators use dig to troubleshoot DNS problems because of its flexibility, ease of use and clarity of output. Other lookup tools tend to have less functionality than dig.

[nslookup(1)] *nslookup - query Internet name servers interactively*. BIND9.

nslookup is a program to query Internet domain name servers. Nslookup has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode is used to print just the name and requested information for a host or domain.

[nsswitch.conf(5)] *`nsswitch.conf` - Name Service Switch configuration file*. Linux Programmer's Manual.

The Name Service Switch (NSS) configuration file, `/etc/nsswitch.conf`, is used by the GNU C Library to determine the sources from which to obtain name-service information in a range of categories, and in what order. Each category of information is identified by a database name.

[resolv.conf(5)] *`nsswitch.conf` - Name Service Switch configuration file*.

> The Name Service Switch (NSS) configuration file, `/etc/nsswitch.conf`, is used by the GNU C Library to determine the sources from which to obtain name-service information in a range of categories, and in what order. Each category of information is identified by a database name.

Correo Electrónico

Revisión: 2021-07-26 22:34:27

SMTP

E54. Si usted desea entregar de forma directa un correo electrónico a una casilla del dominio `alu.itba.edu.ar` ¿A qué servidores SMTP deberá conectarse?

E55. Intente conectarse a otros servidores SMTP que no sean administrado por el ITBA o a la red en la que se encuentra (por ejemplo `smtp.speedy.com.ar`).

a. ¿Puede?

b. Si no puede establecer la conexión: ¿Por qué cree que el administrador del ITBA configuró la red de esa manera?

c. Si le es posible intente enviar un correo electrónico a su cuenta del ITBA. ¿Es posible?

d. ¿Qué significa la configuración de relay?

e. Teniendo en cuenta todo el análisis anterior. ¿Con qué políticas configuraría su servidor SMTP?

E56. Instale **dovecot** y **postfix**. Configurarlo para que los correos enviados a su casilla puedan ser obtenidos mediante el protocolo POP3.

Envíe un correo a su cuenta de correo del ITBA con título `"PDC Ejercicio 1"`, y cuerpo `"hola mundo"`.

a. Use en el `MAIL FROM`: su cuenta de correo del ITBA. Valide que el correo llegue. Revise desde el MUA el "fuente de correo". ¿El correo fue modificado?

b. Haga los cambios necesarios para asegurar que en el campo `To:` que muestra el MUA se vea la dirección del destinatario.

c. Manteniendo su dirección de correo del ITBA en el `MAIL FROM:` agregue en el mensaje una cabecera From con la dirección `alguien@example.org`.
   i. ¿Con qué dirección se muestra en el MUA el remitente?
   ii. ¿Qué ve en las cabeceras del mensaje fuente?

d. Manteniendo los cambios realizados. ¿Que sucede si en el mensaje (`DATA`) se escribe una cabecera To: `pepe@example.org`. ¿El mensaje le llega a usted?

e. Manteniendo los cambios realizados anteriormente, en `RCPT TO:` use una dirección inexistente.
   i. ¿Se le notifica que no se pudo entregar el correo?
   ii. ¿Qué dirección se utiliza para notificar los rebotes?

Sugerencia

Es posible que en ciertos casos los correos no lleguen a destino, o lleguen con retraso. Verifique los logs, y comprenda las razones. Puede utilizar **mailq** y **postsuper** para verificar el estado de la cola de salida.

E57. Repita el ejercicio anterior, pero utilizando como destino una dirección servida por su SMTP local.
- Revise los correos electrónicos desde el filesystem (*maildir* o *mailbox*).
- Revise los correos electrónicos utilizando el protocolo POP3 utilizando netcat.

E58. *{W}* Utilizando un MUA envíe un correo electrónico que contenga de mensaje el texto

```
hólá múndó ñ
```

y contenga como adjunto la imagen ubicada en `https://www.itba.edu.ar/wp-content/uploads/2020/07/Marca-ITBA.jpg`.

E59. Verifique como realiza la transacción SMTP el MUA. Detecte diferencias con las que usted ha realizado en los ejercicios anteriores.

Utilizando el MUA también obtenga utilizando POP3 y verifique la transacción.

E60. Analizar el funcionamiento de las siguientes técnicas anti-spam. ¿Cuales son sus ventajas y desventajas?
- SPF
- Domain keys
- Grey Listing
- Filtros Bayesianos

E61. Según la configuración SPF de `itba.edu.ar`: ¿desde qué direcciones IP se pueden enviar correos a servidores que implementen SPF?

Codificaciones BASE

E62. Codifique en `base64` de forma manual (sin usar ningún programa de computadora) el siguiente mensaje: `HO LA` (el mensaje está compuesto por 5 caracteres: ‘H’, ‘O’, ‘ ‘, ‘L’, ‘A’).

E63. Decodifique de forma manual (sin usar ningún programa de computadora) el siguiente mensaje que fue codificado con `base64` `Y2hhdQ==`

E64. Decodifique con el comando **base64** el siguiente mensaje

```
/9j/4AAQSkZJRgABAgAAZABkAAD/7AARRHVja3kAAQAEAAAALwAA/
+4ADkFkb2JlAGTAAAAAAf/bAIQACQYGBgcGCQcHCQ0JBwkNDwsJCQsPEg4ODw4OEhEODw4
ODw4RERQVFxUUERsbHR0bGycmJiYnLCwsLCwsLCwsLAEKCQkKCwoMCgoMDw0ODQ8TDg4OD
```

```
hMVDg4QDg4VGxQRERERFBsYGhcXFxoYHh4bGx4eJiYkJiYsLCwsLCwsLCws/8AAEQgALwB
4AwEiAAIRAQMRAf/EAJQAAAIDAAMBAAAAAAAAAAAAAAAGBAUHAQIIAwEAAgMBAAAAAAAA
AAAAAAAAgMAAQQFEAACAQMDAgUBBQYHAQAAAAABAgMRBAUAEgYhEzFBIhQHMlFhI7M2cYG
RQjNTocFysnN0FSQRAAEDAwIEBAUFAAAAAAAAAAERAgMAIRIxQVFhEwSBoSIUkbEyYgXh8
VIjFf/aAAwDAQACEQMRAD8A3HRrrJIkNJIWNAW5PyLkObu+NYi4nsbH
YI1y0QO9ZG9cdxVT/AEHClar9ta+WmxQukJQgBoVzjoBQveGpuTYAVorOifUwWvhU01yCC
KjqNefuZW9tJYvjbKSacYrKezElzu7jNcwRiRzv60aaByPu1Kg5Bf4vk2WzMN1LFisXts4
7ZXPbuJYk9vb2+3wp+GXY+SjWv/OJYHNkuQShanDEa7rSPcoULfFfj8K3jRpX4Fy+XkmIi
nvYBaX53eitFmRKAzwKx3bKmh+/z1nPzDmcvZ8uWGzvri3h9pE3bildFqWkqaKRpMPaPkn
MBIa4AruLUx8zWxiQBQa27RrFr+35vleF8Ynwr3lxL2rg3csMrBie7RO428E+HTSXksnzL
F3j2WQvr63uowC8TzvUBhuHg58Rp8f47qKBMzIFwLd/SU0pb+5x1YUIF9rha9O6NYJxzFf
Jk2Rxl44yL455oJWkadijQllYsQZPDbqL8h57OW3M8rDb5C5ihSUBI45nVVGxD0UNTVD8d
lJ02zMd6S4kXRCiVD3KNyLCLpevQujXn3keP5zxe2tr65z7SCdtsa295Kzg7d1SrU6a0z4
o5NluQcflkyh7k9rMYFuSKd1dqt6qADctaHS5uyMcXWbI17VS1qNk4c/AtLSi3p20axD5f
5jeTchXE466khgxy7Zmhdk3TvQsCVIrsFB+2uuPiLmN5ByM4vJXUk1vkV2RGZ2fZOlSlN5
NNwqv7aaL/Pk9v11H05YJdP2ofct6nTTdF51uGjRo1hrRSX8q5eysuNGyu55rZcm/t+9bo
sjKoG99ysyVU02mhr11nfHchPj8XdY+wuDlmulNta3Fo6x3VlDJ/U7UM6rNVmpVVO3p416
60v5BTBm0s3zFmt5GJSsSmNpCpYdWVUliPgPv0p2Fvw97hbjG4WB5bdoJYpAjL/UI7cibr
zqA1NdPtnNHb4ljirsjYYqtr2NZZQTKuQFk5pTJhvjWxSAT5uV7zJzNbzXTq5WMzWtRFIB
9W7afVU9dKPN+FLgTbTWsT5DHb5XhtZiFjF1IS8k97KSgZdoUKOlaUJ+1xg5tdykBoliLq
hh3Rr+IZBbMqLS68St0h/j9mo82TtuUi3xWUs4pIZZAyRzR1AlVZWTpHd7vpRvUOg0Mcnc
MkzkKt1cBwTarc2MtxaL7frWaYLLR4zk9rncrmRPeo6o1vZJ3gY2/D7TSfhwogB8ErTy1L
+bP1mv/AE4f90mr9MfxATGN8DBFIoDKSjEVMrW6DcLzaKyIRUnoOupmWu+NZnJn/wBHHQ3
WURRGN8e0FEAf0y+9ERUbm618RT7Naus0TslDHelhafp02RNhSSwmMsLhcgjWmb4v/QWI/
wCN/wA2TWQfLv67yH+iD8pdafj+UWGMw9omNWOPHATbIUhIaKKKVY3Yq91uJLSAhRVtU19
NxXN35ub3G29zkJ6bndQtVWMOpd/eqi/h+Vfu0jt3GPuZZ3MJa7Kw1CuW9MlAfG2MEKE+V
PfEf0tiP+nB+WusD+S/1zmKf3l/LTWw2PJ+zjCLJI1t7T20NvD2wN0c7m3h9RuyE6oahyC
NUFzHxXMxZLPXOKhnaHtvNK8MgeVpCsY2gXVOlRqu0cYZpJHNJDvTbVXEEVJgHsawEAi9+
VIPJo/jpcZG3HZrx8nuXes4Pb2U9dd6j91NM3xjyq9wvE87dXRL4ywCNZK3h7mXcO0h89z
bSR5amdjgix2s0WHt5GuXKJH2XJUhY29a+7b+5/gdWE2a41kcaMXJYQnH2rGV7UW/ZSPtt
2zIFF0m8+sEAVJBrrJIHxCIske0uBLnoXAA7fKltaj8w5oKIA3Ssrw1vn8nlzkLGyfJ3k
Mou51CGRS5ffWUCnQtrjNW2exuWF9f2T4y8nlN3AmwxqGD76xA16Bta/g8tgMFixe46CDH
RXt4tnLGIGL71Fe5JW6b0qrV/fqPkOT4fPyW8ORx0V6yuFt+9bbgvdBLGvuOn0ddH7x/UJ
EP9YGP3J8tarotxHr9Rvyp14rnYc/gLPKx0BnjHdQfyyr6ZF/cw0ag8FlxcmNuFxlrHZwx
3DI8EUYiG/ah3lRJL9SkeejXIwb7jHE45fTvjqlbMj01UKmvOry7soroJ3GZTGdylSAa/b
1B1FTA2cZLI0iFtu4qVBO36a0Xy8tWWjSg5wCA0aDhVVNhrRRET3pNsiFAu07G6IJBUdNo
8x5aqFuYVeCdMfkO60/su4qAPErFdzsdoIj3P1I6eemzRomyEahfFKos4WpTuvbKZGfHXs
rR2zzhtituUsUaAEr9TEk7fs66+dxJbqtxC2LvZYYbVbhaKjJIT229ulU+paj+H3acNGiE
32+ZqsOflSm3tFnt2GPvC800R7mxQUeREfuk7PFe5Qn7jroHsiFjXG32x7toCrRBQaEFbk
Ap1Vq1BOm/Rqdb7T8amHPypZyEEVrLd26WF3cwCH3UzRbCk0jMQI9pX1udtdfL/5zCr/
+dek3XYSaPatazBqCX0+EZWjE/Tpr0aoS208zUw5+VJ8jWB3gYy9elyLOjRAb40HomWsfW
PxC/wCWptlb2V3LJGbW8i9tc9mJpYwqnaDtuIyV+joQD4/x0x6NQy2sCPGoGc/KqxMBZIn
bRnWP1ehSoX1/X02/zeeuH47j3VVfcyp9AO0hf2DZ01aaNB1H/wAjRYt4VGsrCCyjaOCu1
2LkGn1HxPpA8dGpOjVZFclvxqIESv/Z
```

a. ¿Qué representa el mensaje original?

b. ¿Cuantos bytes ocupa el mensaje original?

c. ¿Cuantos bytes ocupa el mensaje codificado?

d. Dado un mensaje cualquiera: ¿Puede predecir cuanto ocupará codificado en base64?

e. Teniendo en cuenta todo esto, si su proveedor de correo electrónico le comunica que podrá recibir correos electrónicos de hasta 10MB y alguien le quiere enviar un correo con una imagen cuyo tamaño es de 9MB. ¿Llegará?

E65. El ejercicio anterior muestra que es posible transportar en una hoja impresa `*contenido binario*'. ¿Elegiría la codificación *base64* para ése medio de transporte? Compare los criterios de selección del alfabeto de la Sección 3.4 del [RFC4648].

E66. Se encuentra diseñando una aplicación que utiliza HTTP. Desea exponer un recurso que contiene un identificador de un ancho fijo de 160 bits. La *URL Template* [RFC6570] podría ser `http://example.org/docid/{id}`. La intención del ejercicio es discutir diferentes formas de codificación del parámetro `{id}`.

Los 160 bits están contenidos en 20 bytes. Un ejemplo de este identificador representado en bytes sin signo podría ser:

```
0x42 0xa4 0x4e 0xe4 0x3f 0x1d 0x63 0xa9 0xf6 0x57 0x13 0xcb
0xa7 0x6d 0x42 0x30 0x3c 0xcc 0x35 0xa7
```

Que podría transportarse en una URL sin problema utilizando el *Percentage Encoding* (sección 2.1 de [RC3986]) de la siguiente forma:

`%42%a4%4e%e4%3f%1d%63%a9%f6%57%13%cb%a7%6d%42%30%3c%cc%35%a7` (60 bytes/caracteres ASCII)

Dichos bytes se podrían interpretar como un número entero con signo. Si se interpretan los bytes con una codificación *big endian*, complemento a 2, y se representa dicho número en base 10 se podría representar como:

`-380457585513508727835503478909238945443373397415` (48 bytes/caracteres ASCII)

Una representación hexadecimal más compacta podría ser:

`42a44ee43f1d63a9f65713cba76d42303ccc35a7` (40 bytes/caracteres ASCII)

Decide aprovechar la codificación *base64* logrando una codificación de ancho fijo de 28 bytes:

QqRO5D8dY6n2VxPLp21CMDzMNac=

Quedando satisfecho con esta longitud decide implementarlo (un ejemplo de URL podría ser `http://example.org/docid/QqRO5D8dY6n2VxPLp21CMDzMNac=`.

a. ¿El uso de base64 en URLs es una buena decisión o debe tomar algún recaudo?

b. Utilizando esta codificación, ¿puede ahorrase algún byte más al sintetizar las URLs?

MIME

E67. ¿Por qué necesitamos de métodos de codificación como *Quoted Printable* o *Base64* en el envío de correos?

E68. Decodifique el mensaje codificado en el siguiente correo

```
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="Boundary-01=_bHUWOieIFHJ6aoh"
Content-Transfer-Encoding: 7bit
--Boundary-01=_bHUWOieIFHJ6aoh
Content-Type: text/plain;
charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

h=F3l=E1 c=F3m=F3
t=E9 v=E1
=2D-=20
Buenos Aires, Argentina
--Boundary-01=_bHUWOieIFHJ6aoh
```

E69. Utilizando un MUA, cree un correo electrónico cuyo mensaje principal esté formateado en HTML (agregue links; e imágenes externas). Adjunte también un archivo TXT, una imagen JPG, y una imagen PNG. Guárdelo como Draft, y vea su “código fuente”.

a. Revise su estructura MIME

b. ¿Qué es cada parte MIME? Si el mensaje presenta una versión txt/plain del texto que usted escribió analice porque lo incluye si usted escribió un HTML.

c. Fowardee el correo draft como un adjunto MIME. ¿Cuales partes MIME que contienen otras partes?

d. ¿Para que sirve el header `Content-ID`?

e. ¿Para que sirve el header *Content-Disposition*?

f. Investigue el formato de los siguientes Media-Type:
   i. `multipart/digest`
   ii. `multipart/mixed`
   iii. `message/rfc822`
   iv. `multipart/alternative`
   v. `multipart/related`

vi. `multipart/report`

Bibliografía

[dovecot(1)] *dovecot - a secure and highly configurable IMAP and POP3 server*.

Dovecot is an open source IMAP and POP3 server for Linux/UNIX-like systems, written with security primarily in mind. Dovecot is an excellent choice for both small and large installations. It's fast, simple to set up, requires no special administration and it uses very little memory.

[postfix(1)] *postfix - Postfix control program*.

The postfix(1) command controls the operation of the Postfix mail system: start or stop the master(8) daemon, do a health check, and other maintenance.

[base64(1)] *base64 encode/decode data and print to standard output*.

Base64 encode or decode FILE, or standard input, to standard output.

[RFC1939] *Post Office Protocol - Version 3 [https://tools.ietf.org/html/rfc1939]* . The Internet Engineering Task Force. May 1996.

The Post Office Protocol - Version 3 (POP3) is intended to permit a workstation to dynamically access a maildrop on a server host in a useful fashion.

[RFC2045] *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies [https://tools.ietf.org/html/rfc2045]* . The Internet Engineering Task Force. November 1996.

This initial document specifies the various headers used to describe the structure of MIME messages

[RFC3986] *Uniform Resource Identifier (URI): Generic Syntax [https://tools.ietf.org/html/rfc3986]* . The Internet Engineering Task Force. January 2005.

A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource. This specification defines the generic URI syntax and a process for resolving URI references that might be in relative form, along with guidelines and security considerations for the use of URIs on the Internet. The URI syntax defines a grammar that is a superset of all valid URIs, allowing an implementation to parse the common components of a URI reference without knowing the scheme-specific requirements of every possible identifier. This specification does not define a generative grammar for URIs; that task is performed by the individual specifications of each URI scheme.

[RFC4648] *The Base16, Base32, and Base64 Data Encodings [https://tools.ietf.org/html/rfc4648]* . The Internet Engineering Task Force. October 2006.

This document describes the commonly used base 64, base 32, and base 16 encoding schemes. It also discusses the use of line-feeds in encoded data, use of padding in encoded data, use of non-alphabet characters in encoded data, use of different encoding alphabets, and canonical encodings.

[RFC5321] *Simple Mail Transfer Protocol [https://tools.ietf.org/html/rfc5321]* . The Internet Engineering Task Force. October 2008.

This document is a specification of the basic protocol for Internet electronic mail transport. It consolidates, updates, and clarifies several previous documents, making all or parts of most of them obsolete. It covers the SMTP extension mechanisms and best practices for the contemporary Internet, but does not provide details about particular extensions. Although SMTP was designed as a mail transport and delivery protocol, this specification also contains information that is important to its use as a "mail submission" protocol for "split-UA" (User Agent) mail reading systems and mobile environments.

[RFC5322] *Internet Message Format [https://tools.ietf.org/html/rfc5322]* . The Internet Engineering Task Force. October 2008.

This document specifies the Internet Message Format (IMF), a syntax for text messages that are sent between computer users, within the framework of "electronic mail" messages. This specification is a revision of Request For Comments (RFC) 2822, which itself superseded Request For Comments (RFC) 822, "Standard for the Format of ARPA Internet Text Messages", updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.

[RFC6570] *URI Template [https://tools.ietf.org/html/rfc6570]* . The Internet Engineering Task Force. March 2012.

A URI Template is a compact sequence of characters for describing a range of Uniform Resource Identifiers through variable expansion. This specification defines the URI Template syntax and the process for expanding a URI Template into a URI reference, along with guidelines for the use of URI Templates on the Internet.

[RFC7208] *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1, [https://tools.ietf.org/html/rfc7208]* . The Internet Engineering Task Force. April 2014.

Email on the Internet can be forged in a number of ways. In particular, existing protocols place no restriction on what a sending host can use as the "MAIL FROM" of a message or the domain given on the SMTP HELO/EHLO commands. This document describes version 1 of the Sender Policy Framework (SPF) protocol, whereby ADministrative Management Domains (ADMDs) can explicitly authorize the hosts that are allowed to use their domain names, and a receiving host can check such authorization. This document obsoletes RFC 4408.

Otros protocolos de aplicación

Revisión: 2021-07-26 22:34:27

Telnet

E70. *{W}* Exponer un servidor de telnet utilizando **xinetd.**

Para esto debe tener instalado **xinetd** y **in.telnetd** y crear el archivo `/etc/xinetd.d/telnet` con el siguiente contenido y luego reiniciarlo:

```
service telnet
{
    disable = yes
    id = telnet
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    server = /usr/sbin/in.telnetd
}
```

Establezca una sesión con telnet y pase la autenticación. Con wireshark:

- Verifique la negociación entre el cliente y el servidor.
- ¿Cuál es la técnica para que la contraseña ingresada por el usuario no se muestre en la terminal?

Bibliografía

[xinetd(8)] ***xinetd*** - *the extended Internet services daemon*.

xinetd performs the same function as **inetd**: it starts programs that provide Internet services. Instead of having such servers started at system initialization time, and be dormant until a connection request arrives, xinetd is the only daemon process started and it listens on all service ports for the services listed in its configuration file. When a request comes in, xinetd starts the appropriate server. Because of the way it operates, xinetd (as well as inetd) is also referred to as a super-server.

[telnetd(8)] *telnetd — DARPA telnet protocol server*.

The **telnetd** program is a server which supports the DARPA telnet interactive communication protocol. Telnetd is normally invoked by the internet server (see **inetd**(8)) for requests to connect to the telnet port as indicated by the `/etc/services` file (see services(5)). The -debug option may be used to start up telnetd manually, instead of through inetd(8). If started up this way, port may be specified to run telnetd on an alternate TCP port number.

[RFC854] *TELNET PROTOCOL SPECIFICATION [https://tools.ietf.org/html/rfc854]*. The Internet Engineering Task Force. May 1983.

The purpose of the TELNET Protocol is to provide a fairly general, bi-directional, eight-bit byte oriented communications facility. Its primary goal is to allow a standard method of interfacing terminal devices and terminal-oriented processes to each other. It is envisioned that the protocol may also be used for terminal-terminal communication ("linking") and process-process communication (distributed computation).

Protocolos de transporte

Revisión: 2021-07-26 22:34:28

Resumen

El objetivo de esta práctica es comprender cómo funcionan ciertos aspectos de la capa de transporte de TCP/IP y UDP.

E71. Suponiendo que no tiene acceso a su host Linux, determinar desde otro host si están activos los siguientes servicios:

a. daytime
b. time
c. smtp
d. telnet

E72. *{W}* Elegir un servicio de los anteriores que se encuentre inactivo y analizar los paquetes que se generan e intercambian cuando se realiza un intento de conexión.

E73. {W} Utilizando netcat transfiera un archivo entre dos hosts utilizando TCP, UDP, y SCTP. Prestar atención a las diferencias de uso (UDP) y analizar con wireshark los diferentes flujos de información.

E74. {W} Exponer un servidor `echo TCP` utilizando **xinetd**. Debe asegurarse que el servidor `echo UDP` no se encuentra habilitado. De ser posible ejecutar el cliente (**netcat**) desde un host diferente (pero con conectividad). Analizar los paquetes en la interfaz del red donde el cliente deja los datagramas cuando:

a. Se realiza una conexión al servicio echo.

b. Se envía un único caracter.

c. Se envían varios caracteres dejando un pequeño intervalo entre cada uno (por ejemplo 1 segundo).

d. Se envían varios caracteres simultáneamente (por ejemplo escribiendo rápidamente o manteniendo presionada una tecla).

e. Se desconecta el servicio (se mata el proceso servidor)

También analice los cambios de estados que publica la salida de **netstat** en las diferentes etapas. Puede servirse de la Figura 1, “Maquina de estados de TCP” para el seguimiento.

E75. {W} Exponer un servicio `echo UDP` utilizando **xinetd**. Debe asegurarse que el servidor echo TCP no se encuentra habilitado. De ser posible ejecutar el cliente (**netcat**) desde un host diferente (pero con conectividad). Analizar los paquetes en la interfaz del red donde el cliente deja los datagramas cuando:

a. Se ejecuta `nc -u <direccion IP> echo`

b. Se envía un único caracter (presione **Enter**)

c. Se envían varios caracteres

d. Se cierra abruptamente alguno de los procesos (cliente o del servidor)

¿Por qué se muestra en la salida estándar de **netcat** el mismo contenido que se envió? Describa el mecanismo.

E76. {W} Utilizar **nmap** para determinar cuales son los servicios (TCP y UDP) para alguna de las direcciones IP que resultan de resolver el nombre `pampero.itba.edu.ar`. Analizar los paquetes generados.

a. ¿Cómo funciona el escaneo TCP? Verificar empíricamente de forma directa las diferencia el *TCP SYN scan* y el *TCP connect scan*. ¿En que situación es conveniente usar cada uno?

b. ¿Cómo funciona el escaneo UDP?

E77. {W} Utilizar **nmap** para determinar estimativamente el sistema operativo de varios hosts. De ser posible probar con impresoras de red, celulares, "*access points*" hogareños, etc.

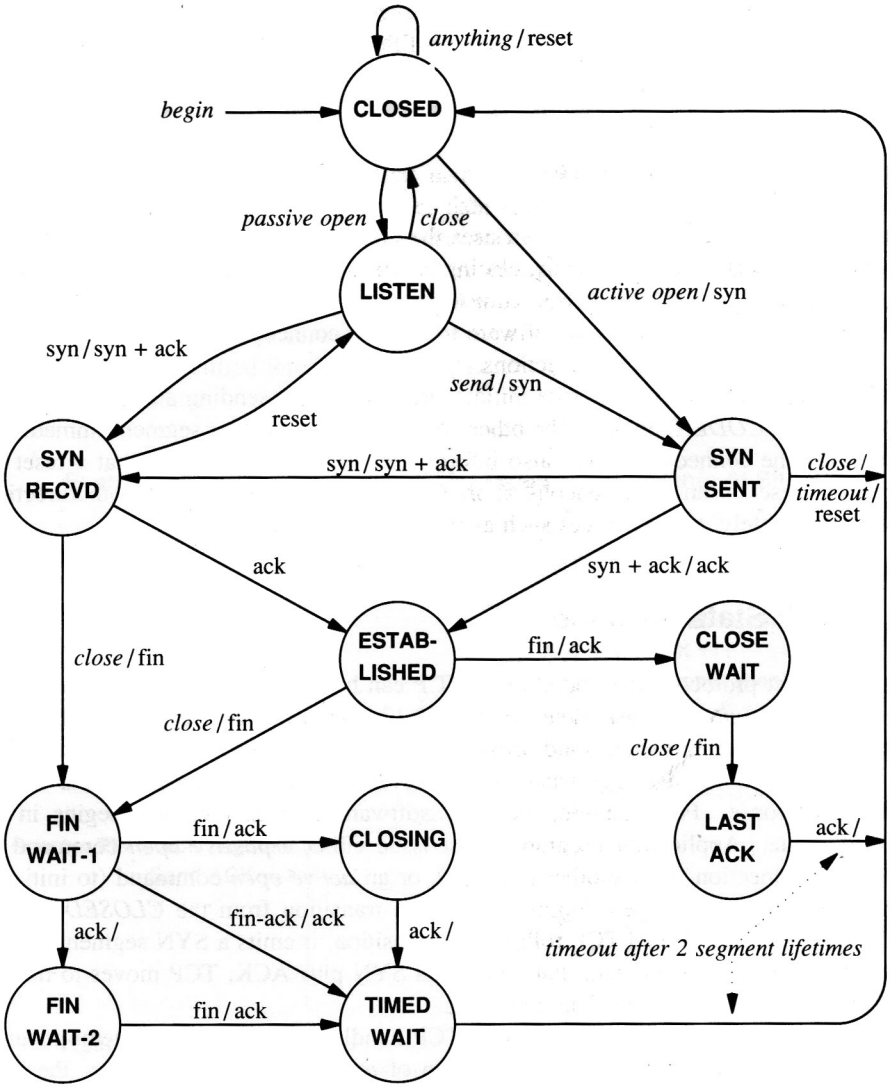

Figura 1. Maquina de estados de TCP

Bibliografía

[xinetd(8)] ***xinetd*** - *the extended Internet services daemon*.

> **xinetd** performs the same function as **inetd**: it starts programs that provide Internet services. Instead of having such servers started at system initialization time, and be dormant until a connection request arrives, xinetd is the only daemon process started and it listens on all service ports for the services listed in its configuration file. When a request comes in, xinetd starts the appropriate server. Because of the way it operates, xinetd (as well as inetd) is also referred to as a super-server.

[services(7)] ***services*** - *Internet network services list*.

services is a plain ASCII file providing a mapping between human- friendly textual names for internet services, and their underlying assigned port numbers and protocol types. Every networking program should look into this file to get the port number (and protocol) for its service. The C library routines getservent(3), getservbyname(3), getservbyport(3), setservent(3), and endservent(3) support querying this file from programs.

[nc(1)] *nc — arbitrary TCP and UDP connections and listens*. BSD General Commands Manual. February 7, 2012.

The nc (or netcat) utility is used for just about anything under the sun involving TCP, UDP, or UNIX-domain sockets. It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scan# ning, and deal with both IPv4 and IPv6. Unlike telnet(1), nc scripts nicely, and separates error messages onto standard error instead of send# ing them to standard output, as telnet(1) does with some.

[nmap(1)] *Network exploration tool and security / port scanner*.

Nmap (“Network Mapper”) is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

[netstat(8()] *Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships*.

Netstat prints information about the Linux networking subsystem. The type of information printed is controlled by the first argument, as follows:...

Internet Protocol

Revisión: 2021-07-26 22:34:28

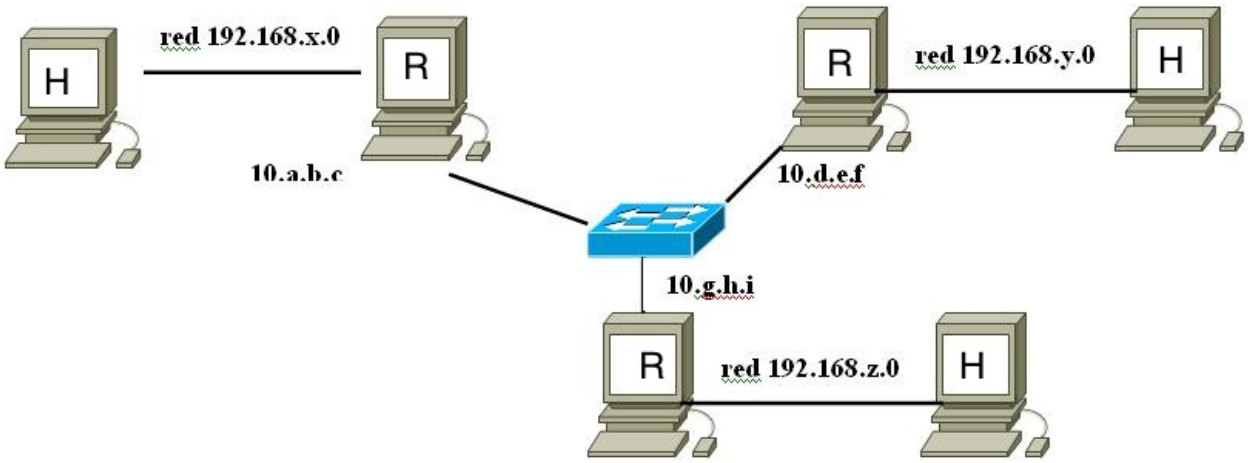

Figura 1. Red a configurar

E78. {W} Ejecutar el comando `traceroute www.google.com`. Analizando el tráfico generado por la herramienta y la información obtenida, explicar cómo determina la información que muestra.

E79. {W} Repetir el ejercicio anterior pero con las opciones `-M icmp` y `-M tcp`. Analizar si se producen cambios en la forma de determinar la ruta y conjeturar sobre las razones de existencia de estas estas opciones.

E80. {W} Ejecutar el comando `traceroute www.google.com 4000`. Analizando el tráfico generado por la herramienta y la información obtenida responder los siguiente:

a. ¿Se fragmentaron los datagramas IPs?

b. De generarse fragmentos ¿Cuantos fragmentos por datagrama? ¿Por qué se produjo la fragmentación?

c. ¿Qué información en el header IP indica que se produjo fragmentación?

E81. {W} Se desea configurar los diferentes hosts R y H descriptos en la Figura 1, “Red a configurar”, para que tengan conectividad entre ellos.

Sugerencia

Un *setup* posible es utilizar una máquina virtual como host R y su máquina real como host H. En ése caso asegúrese que la configuración de red en el gestor de maquinas virtuales sea de tipo *host only*.

Antes de comenzar verifique identifique cada interface de red en cada host, verifique su correcta configuración, y compruebe conectividad con la red a la cual está conectado.

a. El archivo `/proc/sys/net/ipv4/ip_forward` controla el funcionamiento de R como router. Si contiene un `0`, el host no funcionará como router, si contiene un `1` sí tratará de forwardear los paquetes IP que lleguen a él.
   i. Asegurarse que está deshabilitado el forwarding de paquetes IP en R.

Sugerencia

Para mas detalle consulte [ip-sysctl]. Con el comando **cat** puede observar su contenido.

Si se encuentra prendido puede apagarlo ejecutando con el usuario `root` el comando:

echo 0 > /proc/sys/net/ipv4/ip_forward

   ii. Iniciar la captura de paquetes en ambos hosts. Para R capturar el tráfico de la interfaz que se conecta con los demás routers.
   iii. Para cada uno de los siguientes incisos responder las siguientes preguntas:
      - ¿Se genera tráfico en la red al enviar el paquete pedido? ¿En qué segmentos y por qué?
      - ¿Se produce algún tipo de respuesta? ¿Por qué?
      - Analizar los paquetes generados en ambos segmentos.
   iv. Enviar un paquete ICMP Echo Request desde H a R utilizando como destino la dirección IP de la interfaz de red que se conecta con su host H.
   v. Enviar un paquete ICMP Echo Request desde H a R utilizando como destino la dirección IP de la interfaz de red de la interfaz conectada a la red de laboratorio.
   vi. Enviar un paquete ICMP Echo Request desde H a otro host R.
   vii. Enviar un paquete ICMP Echo Request desde H a otro host H.
   viii Habilitar el forwarding de paquetes IP y R, y repetir desde el inciso iv.

E82. Sin utilizar protocolos de routing realizar los cambios que sean necesarios para poder enviar un paquete ICMP Echo Request desde H a un host H de otro grupo y recibir la respuesta.

E83. Realizar los cambios necesarios para que tanto H y R puedan enviar paquetes y obtener respuesta de todos los otros hosts H y R.

E84. Suponga que existe una red `192.168.14.0/24` que está dividida en 8 subredes con máscara de 27 bits. Por un requerimiento especifico, se necesita que ningún host de su red pueda comunicarse con una de estas subredes (`192.168.14.0/27`, siguiendo con el ejemplo).

Se pide realizar los cambios necesarios para cumplir con este requerimiento. Hacerlo de dos maneras distintas.

Sugerencia

Investigar la opción `reject` del comando **route**.

E85. Asigne múltiples direcciones IP a la única interfaz de su host H. ¿Cambia su tabla de ruteo? ¿Cambia su tabla de ruteo en los otros hosts?

Sugerencia

Revise la documentación de ifconfig(1) sobre *alias*.

E86. Existen los hosts A y B de los cuales se sabe que la dirección IP de A es `192.168.0.18` y la dirección IP de B es `192.168.0.13`. Ambos están conectados al mismo hub/switch pero no se pueden comunicar: al hacer enviar un ICMP Request desde A hacia B y viceversa no se obtiene respuesta. Las placas de red funcionan correctamente.

a. De ser posible, reproducir el problema
b. Si se pudo reproducir el problema, hallar dos soluciones distintas para que estos hosts se puedan comunicar conservando sus direcciones IP (una sin utilizar ifconfig).

Bibliografía

[ifconfig(8)] ***ifconfig*** *- configure a network interface*.

ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

[route(8)] ***route*** *- show / manipulate the IP routing table*.

route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the **ifconfig**(8) program.

[ip(8)] ***ip*** *- show / manipulate routing, devices, policy routing and tunnels*.

[ip-sysctl] *ip-sysctl.txt [https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt]* .

Describe las variables `/proc/sys/net/ipv4/*`

[RFC1812] *Requirements for IP Version 4 Routers, [https://tools.ietf.org/html/rfc1812]* . The Internet Engineering Task Force. June 1995.

This memo defines and discusses requirements for devices that perform the network layer forwarding function of the Internet protocol suite.

DHCP

Revisión: 2021-07-26 22:34:28

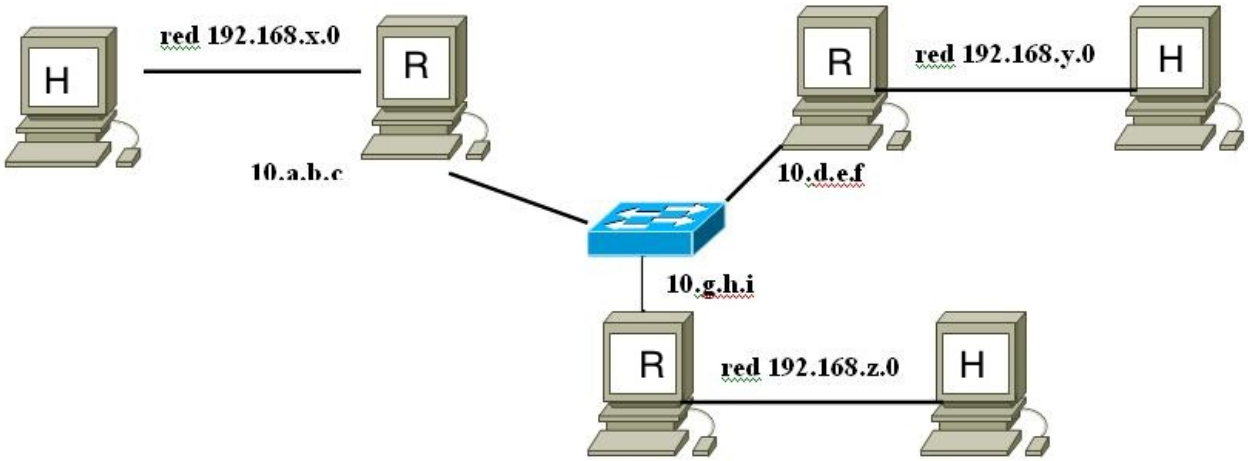

Figura 1. Red a configurar

E87. {W} Configure su host R (Figura 1, “Red a configurar”) para que un servidor `DHCP` provea configuración únicamente en la interfaz de red `192.168.x.0/24`. El servidor DHCP deberá informar como mínimo una dirección IP, la máscara de red y la dirección IP del gateway.

Aviso

Asegúrese que el servidor DHCP únicamente provea configuración en la red indicada y no interfiera con la red del laboratorio.

Sugerencia

Para evitar errores en la inicialización agregar la siguiente línea en el archivo de configuración del servidor DHCP: `ddns-update-style none;`

Si no se le especifica, el servidor DHCP atiende requerimientos en todas sus interfaces de red. Para evitar conflictos con la red del laboratorio especifique en el archivo `/etc/default/dhcp3-server` la interfaz que se conecta a la red del grupo.

Si se detectan errores de inicialización, revisar `/var/log/messages`

Verifique el correcto funcionamiento desde el host H utilizando **dhclient**(8).

E88. Modifique la configuración del servidor DHCP de forma de asegurarse que H reciba siempre la misma dirección IP. Contemplar que puede haber otros hosts en la red, no sólo H.

Sugerencia

Puede asignar direcciones IP basado en las direcciones MAC.

Comenzar la captura de paquetes en H (quitar la opción Enable MAC name resolution})

a. En H renueve la configuración IP
b. Analizar los paquetes DHCP intercambiados entre H y R
   i. ¿Cuántos paquetes se intercambian?
   ii. ¿Cuál es la información pedida o informada en cada caso?
   iii. Observar el Transaction ID. ¿Mantiene su valor a lo largo de la negociación? ¿Por qué?
   iv. ¿Los paquetes son enviados en forma broadcast o unicast? ¿Cómo es posible que el cliente DHCP reciba el paquete que envia el servidor si todavía no ha configurado su dirección IP? (Revisar RFC 2131 Sección 2).

E89. {W} Configurar el servidor DHCP para que el *lease time* sea de dos minutos.

a. Comenzar la captura en el host H
b. Informar que ya no se usa la dirección IP (*release*)
c. Volver a adquirir una dirección IP.
d. Dejar pasar tres minutos y analizar la secuencia de paquetes DHCP que se genera.

E90. {W} Repetir el ejercicio anterior, pero una vez que el host H haya adquirido una dirección IP detener el servidor DHCP

a. Dejar pasar tres minutos y analizar la secuencia de paquetes DHCP que se genera.
b. Repetir el ejercicio pero volver a levantar el servidor DHCP al haber trascurrido un minuto y medio.

E91. {W} Realice los cambios para denegar la configuración de red via DHCP al host H y analizar como es que lo resuelve el servidor DHCP. Restablecer la configuración del servidor.

Bibliografía

[dhclient(8)] ***dhclient*** - *Dynamic Host Configuration Protocol Client*.

The Internet Systems Consortium DHCP Client, dhclient, provides a means for configuring one or more network interfaces using the Dynamic Host Configuration Protocol, BOOTP protocol, or if these protocols fail, by statically assigning an address.

[dhcpd(8)] ***dhcpd*** - *Dynamic Host Configuration Protocol Server*.

The Internet Systems Consortium DHCP Server, **dhcpd**, implements the Dynamic Host Configuration Protocol (DHCP) and the Internet Bootstrap Protocol (BOOTP). DHCP

allows hosts on a TCP/IP network to request and be assigned IP addresses, and also to discover information about the network to which they are attached. BOOTP provides similar functionality, with certain restrictions.

[dhcpd.conf(5)] *dhcpd.conf* - ***dhcpd*** *configuration file*.

The dhcpd.conf file contains configuration information for **dhcpd**, the Internet Systems Consortium DHCP Server. The dhcpd.conf file is a free-form ASCII text file. It is parsed by the recursive-descent parser built into **dhcpd**.

The file may contain extra tabs and newlines for formatting purposes. Keywords in the file are case-insensitive. Comments may be placed anywhere within the file (except within quotes). Comments begin with the # character and end at the end of the line.

[isc-dhcp] *ISC DHCP Server 4.3.1 Distribution Documentation [https://www.isc.org/wp-content/uploads/2014/08/DHCP-4.3.1-Distribution-Documentation-Aug-4-14.pdf]* .

The documentation for dhcpd, the ISC DHCP server, has been assembled from the various man pages included in the ISC DHCP distribution and should not be considered comprehensive. Instead, it is intended to serve as an introduction and overview of ISC DHCP – specifically, the process of setting up a basic DHCP server. For a complete listing of supported commands and variables, please consult the man pages available once the software is installed. This documentation assumes a basic familiarity with networking and DNS.

Routing Information Protocol

Revisión: 2021-07-26 22:34:28

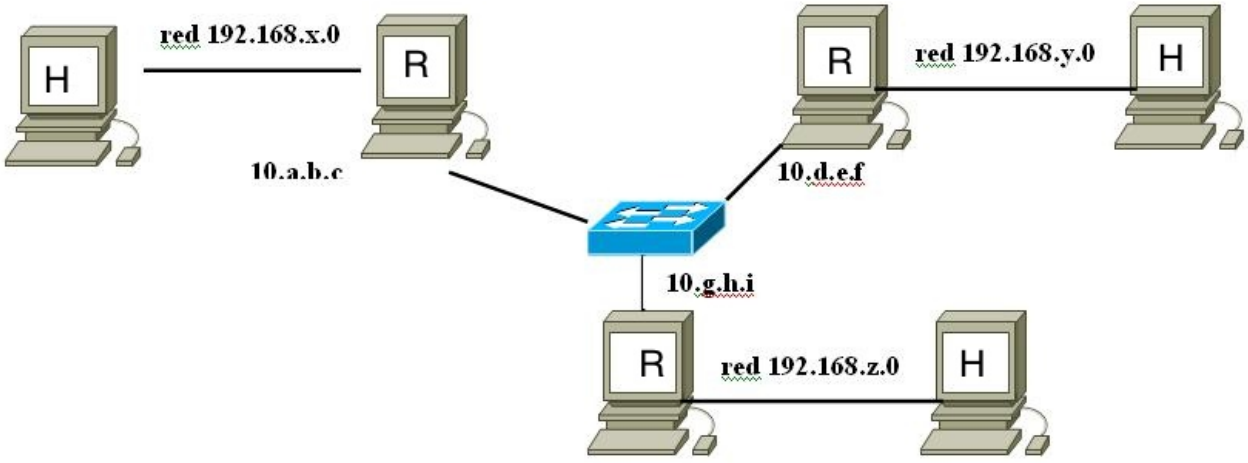

Figura 1. Red a configurar

E92. {W} Asegurarse que está habilitado el forwarding de paquetes IP en los hosts R (Figura 1, “Red a configurar”). Asegurarse que las únicas entradas en la tabla de ruteo de R sean aquellas que se agregan al levantar las interfaces de red.

Activar RIP en los hosts R en modo *background*, para ello levantar el servicio **routed**.

a. ¿Qué protocolo de transporte y qué puerto utiliza **routed**?
b. Iniciar (o continuar) la captura de paquetes desde R
c. Analizar paquetes RIP recibidos y enviados. ¿De cuántos routers recibe información RIP?
d. Analizar los campos más significativos de las capas de red, transporte y aplicación, observando semejanzas y diferencias entre los paquetes recibidos de cada grupo.
e. Una vez que haya recibido paquetes RIP volver a analizar la tabla de ruteo tanto de R como H. ¿Hubo cambios? Explicar.
f. Verifique conectividad con servicios UDP/TCP de otras capas.

E93. {W} Asegurarse que **routed** no está corriendo en ningún host R y realice las modificaciones para lograr un esquema como el que se describe en la Figura 2, “Red 10.x.0.0/16 a configurar”.

a. Asegurarse que las únicas entradas en la tabla de ruteo de R sean aquellas que se agregan al levantar las interfaces de red y que H tenga como default gateway a R.
b. Iniciar el servicio **routed**.
c. Analizar paquetes RIP recibidos y enviados. ¿De qué routers recibe información su router?

d. Una vez que haya recibido paquetes RIP vuelva a analizar la tabla de ruteo tanto de R como de H. ¿Hubo diferencias con respecto al ejercicio 2? Explicar.
e. Enviar un paquete ICMP Echo Request desde su host R a otro host R usando el IP de la red grupal de éste. ¿Se generó tráfico? ¿Hubo respuesta? Explicar
f. Enviar un paquete ICMP Echo Request desde su host R a un host L de otro grupo. ¿Se generó tráfico? ¿Hubo respuesta? Explicar
g. Restablecer la configuración de red de forma tal como estaba antes de empezar esta práctica.

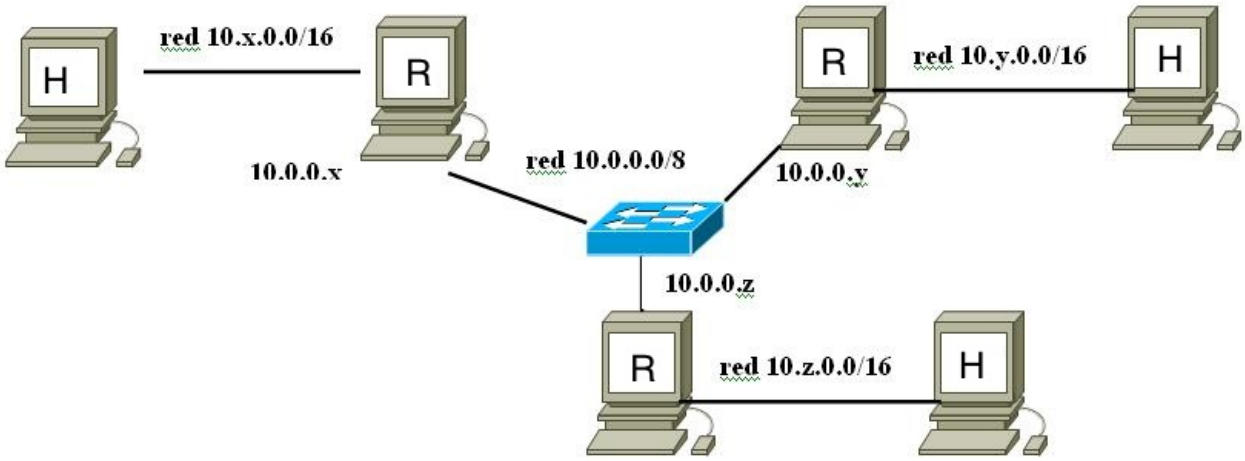

Figura 2. Red 10.x.0.0/16 a configurar

Bibliografía

[routed(8)] ***routed*** *— network routing daemon*.

routed is invoked at boot time to manage the network routing tables. The routing daemon uses a variant of the Xerox NS Routing Information Proto# col in maintaining up to date kernel routing table entries. It used a generalized protocol capable of use with multiple address types, but is currently used only for Internet routing within a cluster of networks

[RFC2453] *RIP Version 2 [https://tools.ietf.org/html/rfc2453]* . Gary Scott Malkin. The Internet Engineering Task Force. November 1998.

This document specifies an extension of the Routing Information Protocol (RIP), as defined in RFC 1058, to expand the amount of useful information carried in RIP messages and to add a measure of security. A companion document will define the SNMP MIB objects for RIP-2. An additional document will define cryptographic security improvements for RIP-2.

Capa de Enlace

Revisión: 2021-07-26 22:34:28

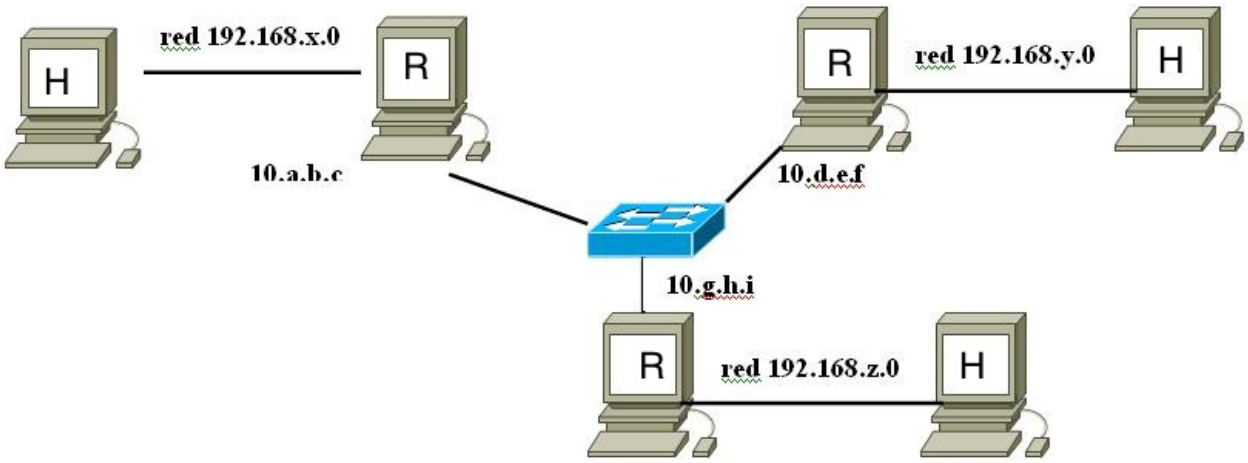

Figura 1. Red a configurar

Nota: se continúa en el escenario descripto en la figura Figura 1, “Red a configurar”.

E94. Averiguar la dirección MAC de las placas de red usadas en la conexiones entre las dos *hosts* de su red (*host* R / *host* H).

E95. Análisis del funcionamiento de ARP

a. Comenzar la captura de paquetes en H

b. En R agregar una entrada en forma estática a la tabla ARP donde tenga el número MAC del H y una dirección IP no utilizada (pero posible en la red IP en la que trabaja) que llamaremos X.

c. ¿Se generó tráfico al agregar esa entrada en la tabla ARP? ¿Por qué?

d. Envíe desde R un sólo paquete ICMP ECHO a la dirección IP X. ¿Se generó tráfico? ¿Por qué? En caso afirmativo, describir y analizar las características de él o los paquetes generados.

e. ¿Afecta lo hecho en el punto (b) al tráfico IP que envíe desde R a H? ¿Por qué?. Encontrar un ejemplo de cómo afecta (o no afecta).

f. Eliminar la entrada agregada en el punto (b)

g. En R agregar una entrada en forma estática a la tabla ARP donde tenga un número MAC no existente y la dirección IP de H. Envíe desde R un sólo paquete ICMP ECHO

al IP de H. ¿Se generó tráfico en la red? ¿Por qué? En caso afirmativo, analizar las características de él o los paquetes generados.

h. Desde H enviar paquetes ICMP ECHO al otro host. ¿Hay respuesta? ¿Por qué?

i. Eliminar la entrada agregada en el punto (g)

j. En R agregar una entrada en forma estática a la tabla ARP donde tenga el número MAC y la dirección IP de H. Agregar otra entrada con un número MAC inexistente y el IP de H. ¿Cuántas entradas ARP habrá para H? Encontrar una razón lógica para lo sucedido.

k. Elimine todas las entradas estáticas de la tabla ARP.

E96. Explique como se realiza el *forwarding* de paquetes IP y cuales son los campos de los headers de Ethernet e IP que entran en juego en este proceso. Puede realizar las siguientes pruebas:

a. Asegurarse que el forwarding de datagramas esté activado en R
b. Comenzar la captura de paquetes en H y R en segmentos de red distintos
c. Analizar las diferencias en los paquetes que se generan cuando en H se ejecuta un ping con destino a:

i. la dirección IP asignada a H en la interfaz que con conecta con R

ii. la dirección IP asignada a R en la interfaz que conecta con H

iii. la dirección IP asignada a R en la interfaz que conecta con otros hosts R

iv. la dirección IP asignada a otro R en la interfaz que conecta con otros hosts R

v. la dirección IP asignada a un host H (de otro R)

E97. Con consentimiento[ACM-ethics][IEEE-ethics] y sin afectar a otros hosts del laboratorio, capture el tráfico entrante y saliente IP de otro host de la red. Deduzca que sitios web visita el otro host, e intente obtener credenciales (POP3, HTTP). ¿Qué podría hacer para obtener las credenciales de campus.itba.edu.ar e impersonar al usuario del otro host?

Bibliografía

[ifconfig(8)] ***ifconfig*** - *configure a network interface*.

> **Ifconfig** is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.
>
> If no arguments are given, **ifconfig** displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

[arp(8)] ***arp*** *- manipulate the system ARP cache*.

> **arp** manipulates or displays the kernel's IPv4 network neighbour cache. It can add entries to the table, delete one or display the current content.
>
> ARP stands for Address Resolution Protocol, which is used to find the media access control address of a network neighbour for a given IPv4 Address.

[ACM-ethics] *ACM Code of Ethics and Professional Conduct*.

> Computing professionals' actions change the world. To act responsibly, they should reflect upon the wider impacts of their work, consistently supporting the public good. The ACM Code of Ethics and Professional Conduct ("the Code") expresses the conscience of the profession.
>
> https://www.acm.org/code-of-ethics

[IEEE-ethics] *IEEE Code of Ethics*.

> The following is from the IEEE Policies, Section 7 - Professional Activities (Part A - IEEE Policies).
>
> https://www.ieee.org/about/corporate/governance/p7-8.html

SSH

Revisión: 2021-07-26 22:34:28

E98. Cuando se realiza una conexión a un servidor `SSH` éste ofrece sus credenciales (clave pública) para que la aplicación cliente pueda determinar si realmente se está conectando al servidor que se desea. Cuando se utiliza el cliente `SSH` y se realiza una conexión por primera vez a un determinado servidor, el cliente no tiene información sobre las credenciales de dicho servidor y por lo tanto delega la responsabilidad de determinar si las credenciales son válidas al usuario. Si el usuario acepta dichas credenciales, el cliente las almacena en el archivo `$HOME/.ssh/known_hosts` para que en una nueva conexión no se le pida al usuario confirmación.

a. Elimine o mueva el archivo `$HOME/.ssh/known_hosts` de su computadora

b. Averiguar el *fingerprint* de la llave pública `RSA` usada por su servidor SSH. Para ello utilizar el comando `ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub`

c. Desde otro host conectarse a su servidor `SSH`. Antes asignar el nombre `foo` a su dirección. Acceder utilizando **ssh foo**. Verificar que el *fingerprint* presentado por dicho servidor coincida con el informado previamente. Desconectarse.

d. Revisar el archivo `$HOME/.ssh/known_hosts` y explicar los cambios producidos.

e. Modificar el nombre `foo` para que apunte a una dirección IP de otro host que tenga un servidor `SSH` (por ejemplo `pampero.itba.edu.ar`). Intente conectarse utilizando el comando **ssh foo**. Explique lo que sucede.

E99. Ejecute los cambios necesarios para poder autenticarse en su servidor SSH sin utilizar contraseñas.

E100. Transfiera el archivo `/etc/passwd` del host `pampero.itba.edu.ar` a su computadora.

E101. Realizar los cambios necesarios para desde su computadora poder navegar por Internet pero que el tráfico salga de `pampero.itba.edu.ar`.

Sugerencia

Recuerde el protocolo SOCKSv5

E102. Realizar los cambios necesarios para que un usuario del host pampero.itba.edu.ar pueda acceder a un servicio que corre en su computadora puerto TCP 9090 unido a la dirección `127.0.0.1`.

E103. Realizar los cambios necesarios para que un usuario de su computadora pueda acceder a un servicio TCP del puerto ubicado en pampero.itba.edu.ar.

Bibliografía

[ssh-keygen(1)] ***ssh-keygen*** *— authentication key generation, management and conversion*.

> **ssh-keygen** generates, manages and converts authentication keys for ssh(1). s**sh-keygen** can create keys for use by SSH protocol versions 1 and 2. Protocol 1 should not be used and is only offered to support legacy devices. It suffers from a number of cryptographic weaknesses and doesn't support many of the advanced features available for protocol 2.

[sshd(8)] ***sshd*** *— OpenSSH SSH daemon*.

> **sshd** (OpenSSH Daemon) is the daemon program for ssh(1). Together these programs replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network.

[ssh(1)] ***ssh*** *— OpenSSH SSH client (remote login program)*.

> **ssh** (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections, arbitrary TCP ports and UNIX-domain sockets can also be forwarded over the secure channel.

Programación con sockets

Revisión: 2021-07-26 22:34:28

E104. Implemente un proxy SOCKSv5 [RFC1928]:

- El servidor debe ser concurrente.
- Únicamente implemente el método de autenticación `NO AUTHENTICATION REQUIRED` (`0x00`).
- Únicamente implemente `CMD CONNECT` (`0x01`).
- Únicamente implemente *address type of following address* `IP V4 address` (`0x01`) y `DOMAINNAME` (`0x03`).

Sugerencia

No olvide de probarlo con protocolos donde el flujo sea requests/response, y con protocolos donde el flujo sea *full duplex*.

Puede utilizar **netcat** como cliente y server. Como cliente puede utilizar el proxy SOCSKS de la siguiente forma: `nc -v -X 5 -x localhost:1080 …`

Puede hacer que casi cualquier aplicación funcione mediante un proxy SOCKS aún cuando ésta no fue programada para ello. Para más información puede revisar [tsocks(8)] y [ld.so(8)].

E105. En el ejercicio anterior ¿Cómo afecta la tasa de transferencia la elección del tamaño del buffer? De ser posible hacer pruebas con un buffer de un byte. Debería observar una degradación el dicha tasa. ¿A qué se debe?

Sugerencia

Puede utilizar netcat combinado con **pv**[pv(1)] para obtener las mediciones. Adicionalmente puede utilizar **sha1sum** [sha1sum(1)] para verificar que no se modificaron los bytes transferidos.

E106. Verificar experimentalmente cuál es el tamaño máximo de un mensaje que se puede enviar y recibir utilizando UDP. ¿Puede hacer lo mismo con TCP?

E110. ¿De qué se trata el problema C10K (*C10K problem*)? ¿Cómo contribuye la utilización de llamadas de sistema no bloqueantes a solucionar dicho problema?

E107. Utilice JMeter y JConsole para verificar los límites operativos de su implementación.

E108. Suponiendo un único cliente que realiza pedidos en forma iterativa a una aplicación, cuál es la relación que se establece entre el *throughput* de la aplicación y el tiempo de respuesta del servicio?

E109. ¿Como se ve afectada la relación del punto anterior a medida que se incrementa la concurrencia de clientes?

Bibliografía

[RFC1928] *SOCKS Protocol Version 5 [https://tools.ietf.org/html/rfc1928]* . M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, y L. Jones. The Internet Engineering Task Force. March 1996.

This memo describes a protocol that is an evolution of the previous version of the protocol, version 4 [1]. This new protocol stems from active discussions and prototype implementations. [STANDARDS-TRACK]

[tsocks(8)] *tsocks - Library for intercepting outgoing network connections and redirecting them through a SOCKS server. [http://linux.die.net/man/8/tsocks]* . Linux man page.

tsocks is a library to allow transparent SOCKS proxying. It wraps the normal connect() function. When a connection is attempted, it consults the configuration file (which is defined at configure time but defaults to /etc/tsocks.conf) and determines if the IP address specified is local. If it is not, the library redirects the connection to a SOCKS server specified in the configuration file. It then negotiates that connection with the SOCKS server and passes the connection back to the calling program.

tsocks is designed for use in machines which are firewalled from then internet. It avoids the need to recompile applications like lynx or telnet so they can use SOCKS to reach the internet. It behaves much like the SOCKSified TCP/IP stacks seen on other platforms.

[ld.so(8)] *ld.so, ld-linux.so* - dynamic linker/loader [http://linux.die.net/man/8/tsocks]* . Linux man page.

The dynamic linker can be run either indirectly by running some dynamically linked program or shared object (in which case no command-line options to the dynamic linker can be passed and, in the ELF case, the dynamic linker which is stored in the .interp section of the program is executed) or directly by running:

```
/lib/ld-linux.so.* [OPTIONS] [PROGRAM [ARGUMENTS]]
```

[pv(1)] *pv - monitor the progress of data through a pipe [http://linux.die.net/man/8/tsocks]* . User Manual.

pv shows the progress of data through a pipeline by giving information such as time elapsed, percentage completed (with progress bar), current throughput rate, total data transferred, and ETA.

[sha1sum(1)] *sha1sum - compute and check SHA1 message digest [http://linux.die.net/man/8/tsocks]* . User Commands.

Print or check SHA1 (160-bit) checksums.

[jmeter] *Apache JMeter [http://jmeter.apache.org/]* .

The Apache JMeter™ application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

Apache JMeter may be used to test performance both on static and dynamic resources (*Webservices (SOAP/REST), Web dynamic languages - PHP, Java, ASP.NET, Files, etc. -, Java Objects, Data Bases and Queries, FTP Servers and more)*. It can be used to simulate a heavy load on a server, group of servers, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load.