UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

"DISEÑO E IMPLEMENTACIÓN DE UN PENTEST"

TRABAJO ESCRITO BAJO LA MODALIDAD DE

SEMINARIOS Y CURSOS DE ACTUALIZACIÓN Y

CAPACITACIÓN PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

NOÉ RAFAEL GONZÁLEZ ARIAS

ASESOR:

M. EN C. LEOBARDO HERNÁNDEZ AUDELO

SAN JUAN DE ARAGÓN, EDO. DE MÉXICO, 2010

Dedicatorias y Agradecimientos

“A mis padres,
Noé Andrés González Rojas
Heribertha Arias Rosales,

por darme una carrera, por creer en mí,
por todos sus consejos y orientación
y principalmente por todo su apoyo y amor incondicional”

“A mis hermanos,
Erika González Arias
Rodolfo González Arias,

por todos los momentos que hemos compartimos juntos”

“A mi maestro,
M. en C. Leobardo Hernández Audelo,

por la dirección y orientación en este trabajo”

“A todos mis amigos,

por todas las experiencias que hemos vivido,
y la dicha de haberlos conocido”

“A la UNAM,

por los conocimientos que adquirí
y por la formación que me brindo”

Índice general

Índice de figuras

Índice de Tablas

Prólogo

Hoy en día, un equipo de cómputo se vuelve indispensable para realizar muchas de las actividades cotidianas que realizamos en nuestra vida diaria ya sea con fines laborales o recreativos. Son tantas las tareas que dependen de estos sistemas, al grado que han dejado de ser una herramienta para convertirse en una necesidad.

Muchas de las tareas que dependen de estos equipos son ejecutadas en combinación con la Internet como por ejemplo utilizar correo electrónico, dar a conocer nuestro negocio o actividad al mundo mediante un sitio web, buscar información para hacer una tarea o trabajo, leer artículos en publicaciones electrónicas de todo tipo, descargar juegos, tener acceso en línea a movimientos de la bolsa, servicios de banca electrónica, comprar artículos de todo tipo, etc. Además mucha de nuestra información personal o de negocio esta almacenada en un equipo de cómputo, desde las fotos y películas familiares hasta información de suma importancia como cuentas bancarias.

Son notorias las enormes posibilidades que ofrece un equipo de cómputo combinado con la Internet, sin embargo, la facilidad y comodidad que se obtiene realizando nuestras actividades a través de estos medios muchas veces nos hace olvidar el riesgo que conllevan si no son llevadas a cabo con la debida precaución.

Basta con mirar la televisión o leer los periódicos para enterarse de los muchos problemas de seguridad relacionados con sistemas de cómputo, estos problemas, también son llamados incidentes de seguridad informática.

Algunos incidentes de seguridad informática son por ejemplo: infecciones a nuestros sistemas por virus informáticos, fraude electrónico, robo de información de tarjetas de crédito, suplantación de identidad, etc.

En muchos casos ser víctima de un ataque informático es consecuencia no de imprudencia sino de ignorancia. Mi intención en este reporte es dar a conocer la facilidad con la que se puede irrumpir en un sistema de cómputo que no está bien protegido y a su vez mostrar la importancia que tiene la seguridad informática ya sea para uno mismo o para una organización. No pretendo en ningún momento hacer de este reporte un curso de seguridad, el contenido del mismo hace referencia tanto a las amenazas y vulnerabilidades como a los ataques a los que se está expuesto al utilizar sistemas de cómputo así como los medios que se tienen para protegerse de ellos. Intentare ir más allá de lo teórico mediante la simulación de algunos de estos ataques.

Por todo lo anterior, este reporte cuenta con los siguientes temas:

- Capítulo 1. En este capítulo se explica de manera general el objetivo de la seguridad informática así como los servicios que esta ofrece.
- Capítulo 2. Muchas de las vulnerabilidades, amenazas y ataques a los que los sistemas de cómputo están expuestos serán descritos de forma breve en este capítulo.
- Capítulo 3. Este capítulo hará énfasis en los mecanismos y técnicas más comunes de las que hace uso la seguridad informática para prevenir y combatir las amenazas y ataques.
- Capítulo 4. Una de las formas que se tiene para saber que tan expuesto está nuestro equipo de cómputo o la red de una empresa es a través de una prueba de penetración (Pentest). Este capítulo hace referencia al objetivo de estas pruebas.
- Capítulo 5. La manera más clara de entender cómo la información que almacenamos en nuestros equipos de cómputo, la seguridad de estos mismos y la red de una empresa están expuestos y pueden ser víctimas de tantas amenazas es mediante un ejemplo práctico. En este capítulo se realizará una pequeña prueba de penetración a una aplicación web.
- Capítulo 6. En este capítulo se darán las conclusiones obtenidas de todo lo mencionado en los capítulos anteriores, así como algunas recomendaciones generales de seguridad informática.
- Anexo. Por último se muestra un anexo enfocado al aseguramiento de un equipo de cómputo.

Capítulo 1 .- Introducción

Debido a la rápida evolución que las computadoras han tenido desde que fueron creadas, su propósito original ha cambiado desde entonces. Inicialmente las computadoras fueron diseñadas para facilitar la investigación sin poner mucho énfasis en la seguridad. Con la penetración de las computadoras en los hogares y oficinas se desarrollo una increíble dependencia a estas que llego a significar que una falla en el equipo podía derivar en la pérdida de tiempo o dinero, por lo tanto surgió la necesidad de proteger estos equipos pero no sólo su seguridad física sino también la información que estos almacenan así como aquella que es enviada a otros equipos. Por lo anterior es que surge la seguridad informática. Este capítulo dará a conocer de forma breve la finalidad de la seguridad informática.

1.1. Antecedentes

Originalmente las computadoras no tuvieron sistemas operativos: los propios usuarios les daban sus programas en binario a través de switches; cada usuario tenía acceso exclusivo a los recursos de la computadora durante lapsos de tiempo previamente establecidos; los usuarios cargaban sus propias rutinas de soporte tales como ensambladores, compiladores, etc., eliminando todos sus datos después de terminar.

Los primeros antecedentes de los sistemas operativos actuales eran utilerías llamadas ejecutivos que ayudaban a los programadores y hacía más amable la transición cuando otro usuario requería usar la computadora. Estos programas manejaban un único programa a la vez para su ejecución.

Con el desarrollo de la multiprogramación, los sistemas operativos asumieron un papel distinto. Cuando más de un usuario y más de un programa podían acceder a recursos de las computadoras, hubo necesidad de desarrollar conceptos tales como scheduling[1], compartimiento de recursos y concurrencia, entre otros.

Los primeros sistemas operativos multiprogramados eran conocidos como monitores y se encargaban de supervisar la ejecución de los programas. La diferencia entre estos dos sistemas operativos era que mientras un ejecutivo proporcionaba servicios por demanda, el monitor supervisaba todas las computaciones y asignaba recursos a los usuarios.

[1] Scheduling: Repartir el tiempo disponible de un microprocesador entre todos los procesos que están disponibles para su ejecución.

En términos de seguridad, los ejecutivos aceptaban un solo usuario y programa, y la única amenaza existente eran los propios usuarios: ningún usuario podía afectar intencionalmente el cómputo de otro. Con los monitores, que manejaban multiprogramación, un programa de un usuario podía causar un efecto negativo en la ejecución de un programa de otro usuario. Por esta razón, la protección de un usuario sobre otro se convirtió en un importante problema de seguridad en los sistemas multiprogramados.

Para poder dar la protección antes mencionada es necesario saber exactamente qué es lo que se va a asegurar, es por eso que a continuación se hace referencia a los activos relacionados con la seguridad informática.

1.2. Activos a Asegurar

De acuerdo a lo señalado en el tema anterior, el surgimiento de la multiprogramación trajo consigo la necesidad de proteger, por parte del sistema operativo, objetos o recursos del sistema de cómputo que se administraban de modo compartido y controlado, tales como:

- Memoria.
- Dispositivos compartidos de E/S, tales como discos, impresoras, etc.
- Programas y utilerías compartidas.
- Datos compartidos.

Pero la seguridad informática no sólo se enfoca en proteger los recursos que se comparten dentro de un sistema. Hoy en día la seguridad informática se vuelve importante no sólo para las redes empresariales sino para cualquier persona que haga uso de un equipo de cómputo, esto debido a la existencia de personas ajenas a la información, también conocidas como piratas informáticos o hackers[2], que buscan tener acceso a la información pública o privada para modificar, sustraer o borrar datos con fines poco éticos.

De este punto en adelante se hace referencia en repetidas ocasiones a algunos conceptos que deben tenerse claros para no confundirlos. Por lo tanto el siguiente tema hace mención de algunos conceptos básicos (pero no los únicos) de la seguridad informática.

[2] La palabra hacker se utiliza normalmente para describir a alguien que con conocimientos suficientes para eludir o desactivar las medidas de seguridad de un sistema.

1.3. Conceptos Básicos

Los conceptos que a continuación se señalan son primordiales para tener un panorama claro de lo que conlleva la seguridad informática por lo tanto es necesario entenderlos de manera correcta:

- **Amenaza.** Es cualquier circunstancia con el potencial suficiente para causar pérdida o daño al sistema. Existen diferentes tipos de amenazas: amenazas naturales (terremotos, inundaciones, huracanes, etc.), humanas, (huelgas, amenazas de bomba, empleados descontentos, etc.), y de tecnologías (hardware y software).
- **Vulnerabilidad.** Consiste en cualquier debilidad que puede explotarse para causar pérdida o daño al sistema. Indica que un activo es susceptible a recibir un daño a través de un ataque. Por ejemplo, falta de contraseñas en los equipos de la empresa, falta de registro de entradas y salidas en las instalaciones tanto de personal interno como externo, falta de políticas dentro de la empresa, falta de actualizaciones de los sistemas operativos y/o de las aplicaciones, etc.
- **Ataque.** Consiste en cualquier acción que explota una vulnerabilidad. Existen dos tipos de ataques:
    - Ataques pasivos (sólo afectan la confidencialidad). Un ataque pasivo consiste sólo en monitorear la red, identificar como está constituida y obtener información importante a cerca de una empresa u organización sin alterar ni el estado del sistema ni la información. Aunque en esta etapa en sí, no se comete ninguna acción que afecte a la empresa, si se pasa a la vida real, es como si de repente se encontrase a una persona que está afuera de una empresa revisando qué puertas o qué ventanas están abiertas y por dónde se podría acceder a la empresa. Aunque en sí, no está ingresando a las instalaciones de la empresa, son actividades que hacen sospechar sobre las intenciones que se tienen al revisar los puntos de acceso a la empresa. Normalmente un ataque pasivo es realizado antes de realizar un ataque activo.

    - Ataques activos (afectan la confidencialidad, integridad y autenticidad). Un ataque activo tiene la capacidad de modificar o afectar la información, o el estado del sistema o ambos. Consiste en traspasar los mecanismos de seguridad implementados en una empresa, acceder a información confidencial y la posterior destrucción, divulgación o modificación de la misma.

Con la información a la que se ha hecho referencia hasta el momento, problemática inicial, los activos a asegurar y los conceptos recién mencionados se puede dar un concepto más amplio de seguridad.

1.4. Concepto de Seguridad Informática

La seguridad informática puede ser entendida como una disciplina con las siguientes características:

- Se encarga de proteger la información de amenazas, ataques y vulnerabilidades reales y potenciales.
- Garantiza propiedades de la información en todos sus estados: creación, modificación, transmisión y almacenamiento.
- Son técnicas, procedimientos y herramientas para proteger y resguardar información en medios electrónicos.
- Protege la información almacenada en los equipos así como la que se transfiere e intercambia por canales públicos.

Entonces:

“La seguridad es una característica que se le pretende dar a un determinado sistema con el fin de afrontar y vencer las vulnerabilidades, amenazas y ataques a los que pueda estar expuesto sin que este se rompa”.

Para lograr seguridad sobre un activo hay que tener claro qué es lo que se desea, es decir, qué tipo de protección es la que se quiere lograr (servicios de seguridad) y de acuerdo a esto la manera es que la seguridad será implementada (mecanismos de seguridad).

1.5. Servicios de Seguridad Informática

Como se ha señalado anteriormente la seguridad informática es un problema que se plantea tanto para cualquier persona como para grandes redes empresariales. En este sentido, los servicios de seguridad son los siguientes:

1.5.1. Integridad

Estipula que la única forma de crear, modificar y eliminar los datos en un sistema así como los datos enviados entre varios sistemas, sea de una manera controlada y por elementos autorizados.

1.5.2. Confidencialidad

La confidencialidad se asegura que la información existente en un sistema o aquella que es transmitida, sea leída únicamente por personas o entidades autorizadas.

1.5.3. Autenticación

La autenticación se logra asegurando que el origen de un mensaje o documento digital es correctamente identificado, lo cual asegura que la identidad provista en el documento de quien fue su creador, no es falsa.

1.5.4. No Repudio

Previene que un emisor niegue haber remitido un mensaje (cuando realmente lo ha emitido) y que un receptor niegue su recepción (cuando realmente lo ha recibido).

1.5.5. Control de Acceso

Es una protección contra el uso no autorizado del sistema. Se aplica a todo tipo de acceso a la información como transferencia, escritura, lectura y ejecución. Cada acceso debe ser verificado en función de los privilegios del sujeto y los atributos de la información.

1.5.6. Disponibilidad

La disponibilidad indica que los datos u objetos estarán accesibles por entidades autorizadas cuando estás las requieran, confiando en la autenticidad que dichos objetos son los que dicen ser.

Los primeros cinco servicios se estandarizan por ISO[3] en la norma ISO 7498-2 y deben ser implementados para disminuir el riesgo de sufrir indisponibilidad.

Los servicios de Integridad, Confidencialidad y Disponibilidad son conocidos como el triángulo de oro de la seguridad (Figura 1.1).

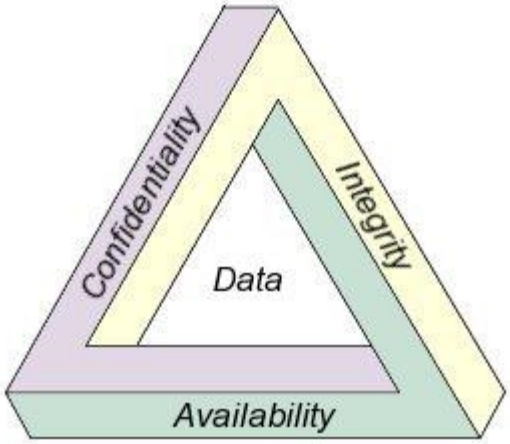

Figura 1.1 Triángulo de Oro de la Seguridad Informática

La implementación de estos servicios se hace a través de diferentes técnicas conocidas como mecanismos de seguridad.

[3] La Organización Internacional para la Estandarización o ISO, es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional

1.6. Mecanismos de Seguridad

Las técnicas que se utilizan para implementar los servicios de seguridad antes mencionados son llamadas mecanismos de seguridad. Con excepción del servicio de disponibilidad todos los servicios de seguridad informática son implementados mediante diferentes mecanismos de seguridad como se observa en la siguiente tabla:

Servicio	Mecanismo
Confidencialidad	Cifrado
Integridad	Funciones Hash
Autenticación	Protocolos Criptográficos
Control de acceso	Esquemas de control de acceso
No repudio	Firma digital

Tabla 1.1 Servicios y Mecanismos de Seguridad

Como se observa en la tabla 1.1 la mayoría de los mecanismos de seguridad están basados en criptografía por lo tanto no es posible hablar de seguridad informática sin hablar de criptografía, el siguiente tema aborda tanto su concepto como su funcionalidad.

1.7. Criptografía

Para dar una posible solución a cuatro de los problemas de la seguridad en el manejo de los datos (confidencialidad, autenticación, no repudio e integridad), es necesario conocer los métodos de transformación de los mismos de manera controlada, mediante el empleo de funciones matemáticas, de manera que se pueda garantizar en primer lugar el secreto en la comunicación entre dos puntos y en segundo lugar asegurar que la información que se envía es autentica en un doble sentido (de manera que el remitente sea quien dice ser y que su contenido haya llegado sin alteraciones durante su tránsito).

Al estudio de estos sistemas que ofrecen medios seguros de comunicación se les conoce como criptografía.

Estos sistemas de comunicación están basados en mecanismos donde el lado emisor oculta o cifra el mensaje para que sólo el receptor autorizado pueda descifrarlo. En la actualidad la criptografía ha extendido sus áreas a métodos de autenticación de información digital, comúnmente llamada firma digital (posteriormente descrita).

El procedimiento o método utilizado para cifrar datos se realiza mediante la implementación de un algoritmo conocido, el cual puede ser considerado como una función matemática donde entra información en texto claro y es desordenada de manera que la misma se transforme en algo incomprensible a partir de la utilización de

al menos un código o llave. En muchos casos esta función de transformación posee su contraparte, la que permite descifrar el mensaje (mediante el ingreso de un texto cifrado se logra como salida el texto claro), este sistema criptográfico es conocido como de doble vía o reversible.

Estos métodos de cifrados se dividen en dos categorías: de sustitución y de transposición. Los primeros se logran mediante el intercambio de una letra o grupo de letras, por otra letra o grupo de letras, para disfrazarlas. Un ejemplo de cifrado por sustitución es el conocido Cifrado de Cesar[4].

Los cifrados de transposición reordenan las letras en lugar de disfrazarlas. Esto se logra mediante el cambio de posición de los caracteres del mensaje, por ejemplo, reescribiendo un texto corrido, una cantidad específica de lugares, o mediante uso de una llave se define el tamaño para un vector de cambios.

Para abordar el tema de la criptografía de manera más profunda es necesario hablar sobre las funciones Hash y algunas de sus implementaciones.

1.7.1. Funciones Hash

Una función hash, es un método que se aplica a un documento para garantizar la autentificación del mismo mediante una secuencia de bits de pequeña longitud adjuntando al mensaje en lugar de tener que utilizarlo por completo. Como características fundamentales para un buen uso de esta Función Hash, se destacan:

- Que el resultado de la función sea de longitud física, independientemente de la longitud del mensaje.
- Que sea fácil de calcular sobre cualquier mensaje.
- Que sea computacionalmente intratable recuperar el mensaje a partir de la función hash.
- Que sea computacionalmente intratable lograr generar la función hash desde un mensaje distinto. De aquí deriva una condición implícita que determina que el tamaño del resultado de la función hash debe ser de al menos 128 bits para poder reducir las probabilidades de que dos mensajes aleatorios den el mismo valor.

[4] Cifrado de Cesar. Atribuido a Julio Cesar, donde la letra A se representaba con una D, la B por una E, la C con una F y así con cada una de las letras del alfabeto donde son sustituidas por la que se encuentra tres lugares delante de ella (de manera circular, donde luego de la Z vuelve a comenzar con la A). Una variante de este método es permitir desplazar n lugares cada letra donde n es convertida en la clave de cifrado.

A continuación una breve descripción de 2 de las funciones Hash más conocidas (pero no las únicas).

1.7.1.1. MD5

MD5[5] es un algoritmo de reducción criptográfico ampliamente difundido en la actualidad, diseñado por el profesor Ronald Rivest[6], en el año 1991, luego de que el mismo año, Den Boer y Bosselaers publicarán ciertas debilidades en su antecesor MD4.

La codificación del MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal. El siguiente texto es tratado con MD5 y se observa su correspondiente hash de salida:

MD5 ("prueba 1 de md5") = c4a555c8c36be0cc9998af19cf131db6

Un simple cambio en el mensaje nos da un cambio total en la codificación hash, en este caso cambiamos un número, el «1» por un «2».

MD5 ("prueba 2 de md5") = c3b4dde8034f9939bf274e52507abaeb

En el año de 1996, Hans Dobbertin[7] anunció que el MD5 tenía problemas de colisión de hash. Quedó demostrado que un hash de 128 bits era muy pequeño por lo que se recomendó sustituir al MD5 por algoritmos alternativos como SHA-1.

1.7.1.2. SHA

El algoritmo SHA (Secure Hash Algorithm, Algoritmo de hash seguro) fue desarrollado por la NSA en el año 1993 y evolucionó a versiones posteriores publicadas con los nombres SHA-1 (ambas de 160 bits) y SHA-2 (esta última con variantes en el rango de salida, llamadas SHA-224, SHA-256, SHA-384, y SHA-512). La importancia que representa el hash de salida y la característica fundamental de que sea teóricamente única, es imprescindible para garantizar que no se encuentren mensajes con firmas

[5] MD5: Acrónimo de Message Digest Alorithm 5, es un algoritmo de reducción criptográfico de 128 bits.

[6] Profesor Ronald L. Rivest, nacido en 1947 en Schenectady, (Nueva York). Criptógrafo y profesor de ciencias de la computación en el departamento de ingeniería eléctrica y ciencias de la computación del MIT. Es muy conocido por su trabajo con el cifrado de clave pública junto con Len Adleman y Adi Shamir, específicamente el algoritmo RSA, con el que ganaron en el año 2002 el premio ACM Turing. También es el inventor de los algoritmos de llaves de cifrado simétrico RC2, RC4, RC5, y co-inventor de la RC6. «RC» viene de Rivest Cipher o bien de Ron's Code. El RC3 fue roto mientras se desarrollaba en el RSA Security y, al igual que el RC1, nunca fue publicado. También ha sido el autor de las funciones criptográficas de hash MD2, MD4 y MD5.

[7] Hans Dobbertin (1952 - 2006) fue un criptógrafo alemán que realizó criptoanálisis de los algoritmos MD4, MD5, y en las funciones hash del RIPEMD original, así como en su participación en el diseño de la función de hash de la nueva versión del RIPEMD

iguales. Por ello el incremento de longitud de las funciones SHA de hasta 512 bits. Como se menciona anteriormente, el SHA-1 (al igual que el MD5) son algoritmos de una sola vía y por ello no es posible obtener el mensaje original a partir de su hash al utilizar fuerza bruta[8].

Aquí un ejemplo de resultados de las funciones de SHA para la frase: “este es mi hash SHA”.

> ***SHa1:*** *60d127db6ef5ef4e329bacb0b80c52e471af4ec8*
> ***SHA256:*** *2e7ed0e51fab3a3bc7105477b59ce5992945ab4976d86e46a7d8b937fb565eb2*
> ***SHA384:***
> *2b994e861d40a331405aade0bf18d18834e32410f2b35d919b6b8910d13101f0286fb511d3f4b75c2 01f00146b85d576*
> ***SHA512:***
> *3d2a37ac16dd69b886bc1d7cad3dc197c9ddf47bf168f436459222b01a441f443764c7a3cbdf8b166 a423c27ebb12090edfe928237b9e9b1d2d9b7cc75bda829*

Cabe señalar que las funciones mencionadas no son las únicas existentes. A continuación se hace mención de dos métodos criptográficos que implementan el uso de funciones hash: criptografía simétrica y criptografía asimétrica.

1.7.2. Criptografía Simétrica

La criptografía simétrica es un método criptográfico que utiliza una única llave para cifrar y descifrar mensajes. Si dos individuos desean comunicarse establecen un valor para la llave, este valor permanece secreto para cualquier otro individuo lo que permite a las dos partes ser las únicas que puedan descifrar el mensaje. La criptografía simétrica tiene tres problemas principales:

- Si un tercer individuo obtiene la llave, el mensaje enviado por los dos individuos originales sería comprometido. Aunado a esto, el tercer individuo puede enviar mensajes cifrados utilizando la llave y así engañar a los otros dos.
- La llave debe ser distribuida de forma secreta.
- Se necesitan demasiadas llaves para permitir la comunicación secreta entre pares de de individuos. El número de llaves sería $(n^2-n)/2$. Esto quiere decir que si fueran 10 individuos sólo 45 llaves serían necesarias, pero si fueran 100 individuos 4950 serían requeridas.

A continuación dos implementaciones de criptografía simétrica: DES y AES

[8] En informática se denomina fuerza bruta a la forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso.

1.7.2.1. DES

El algoritmo DES (Data Encryption Standard) está basado en el algoritmo Lucifer[9], desarrollado por IBM a principios de los setenta y luego adoptado como estándar por el gobierno de los Estados Unidos para comunicaciones no clasificadas.

DES está diseñado para cifrar y descifrar bloques de datos de 64 bits utilizando una llave de 64 bits (56 bits utilizados por el algoritmo y ocho bits usados para la detección de errores).

Tiempo después se encontraron fallas en este algoritmo debido a la corta longitud de la llave. Los usuarios al verse amenazados por la aplicación exitosa de fuerza bruta, comenzaron a utilizar 3DES, que consiste en aplicar DES tres veces consecutivas con diferentes llaves. Seguro por algún momento pero con costos computacionales muy altos. Empezaron a aparecer alternativas como DES-X o GDES. En el 2001, tras un concurso internacional, el NIST (National Institute for Standards and Technology) escogió un nuevo algoritmo, el AES (Advance Encryption Standard).

1.7.2.2. AES

El AES o algoritmo de Rijndael fue anunciado como nuevo estándar avanzado de cifrado para el empleo de aplicaciones criptográficas no militares, potente, eficiente y fácil de utilizar. Creado por los Belgas Joan Daemen[10] y Vincent Rijmen[11], es un sistema de cifrado por bloques, diseñado para manejar longitudes de llave variable comprendidas entre 128 y 256 bits. El método más común de ataque hacia un cifrador por bloques consiste en intentar varios ataques sobre versiones del cifrador con un número menor de rondas. El AES tiene 10 rondas para llaves de 128 bits, 12 rondas para llaves de 192 bits, y 14 rondas para llaves de 256 bits. Hasta 2005, los mejores ataques conocidos son sobre versiones reducidas a 7 rondas para llaves de 128 bits, 8 rondas para llaves de 192 bits, y 9 rondas para llaves de 256 bits.

---

[9] El algoritmo original Lucifer de Horst Feistel tenía una llave de 112 bits de longitud pero fue reducido a petición de la Agencia de Seguridad Nacional de E.U. a los 56 bits actuales.

[10] Joan Daemen (nacido en 1965) es un criptógrafo belga y uno de los diseñadores de Rijndael, el algoritmo elegido para ser el estándar criptográfico, junto con Vincent Rijmen. Fue también diseñador de los algoritmos de cifrado por bloques MMB, Square, SHARK, NOEKEON y 3-Way. Daemen nació en Hamont-Achel, provincia de Limburgo de Bélgica. Trabajó en el diseño y criptoanálisis de cifrados por bloques, cifrados de flujo y de funciones hash.

[11] Vincent Rijmen (16 de octubre de 1970) es un criptógrafo diseñador de Rijndael, el Advanced Encryption Standard. Rijmen es co-diseñador de la función de hash WHIRLPOOL y de los algoritmos de cifrado por bloques Anubis, KHAZAD, Square, NOEKEON y SHARK. Rijmen nació en Leuven, cerca de Bruselas (Bélgica).

Algunos criptógrafos muestran preocupación sobre la seguridad del AES. Ellos sienten que el margen entre el número de rondas especificado en el cifrador y los mejores ataques conocidos es muy pequeño. El riesgo es que se puede encontrar alguna manera de mejorar los ataques y de ser así, el cifrado podría ser roto. En el contexto criptográfico se considera "roto" un algoritmo si existe algún ataque más rápido que un ataque por fuerza bruta. De modo que un ataque contra el AES de llave de 128 bits que requiera 'sólo' 2120 operaciones sería considerado como un ataque que "rompe" el AES aun tomando en cuenta que por ahora sería un ataque irrealizable.

1.7.3. Criptografía Asimétrica

La criptografía asimétrica o criptografía de llave pública[12] es el método criptográfico que utiliza un par de llaves (una pública y otra privada) para la transmisión de mensajes. Ambas llaves son generadas en el mismo momento, la llave pública se entrega a las terceras partes, y la llave privada se deberá guardar de modo que nadie tenga acceso a ella. Estos sistemas fueron creados para evitar el problema de intercambio de llaves que contenían los sistemas criptográficos simétricos.

Existen dos formas en que puede ser empleado el sistema de llaves públicas. La llave pública puede ser usada para cifrar un texto en claro, o puede ser usada para descifrar un texto cifrado. Un sistema que trabaja en cualquiera de las dos formas es llamado criptosistema de llave pública irreversible, en este método la comunicación se realiza a partir de cifrar un mensaje con la llave pública del destinatario, de manera que sólo él, con la llave privada, pueda descifrar el envío. Este método se utiliza en el caso que se desee proteger la información desde un emisor (el que conoce la llave pública) a un receptor (que obviamente posee su llave privada).

Un sistema que trabaja en ambas direcciones es conocido como criptosistema de llave pública reversible. Esta forma se utiliza para la autentificación de mensajes. Mediante el uso de firmas digitales, el emisor (en este caso el que posee ambas partes, la llave pública y la privada), tiene que generar un hash del mensaje y codificarlo con su llave privada. El receptor (parte que desea comprobar la autenticidad del mensaje y conocedor de la llave pública del emisor) puede ahora descifrar y comparar el criptograma. Si coinciden, el mensaje es auténtico ya que el único que posee la llave privada para codificar el mismo es el emisor.

El proceso de llave pública puede ser expresado de la siguiente manera:

Texto cifrado = Cifrar [Llave1] (texto en claro)
Texto en claro = Descifrar [Llave2] (Texto cifrado)

A continuación 2 ejemplos de criptografía asimétrica: RSA y Firma Digital.

[12] El concepto de criptografía de llave pública fue introducido por Whitfield Diffie y Martin hellman en 1976.

1.7.3.1. RSA

RSA[13] es un algoritmo que se basa en la dificultad de factorizar grandes números. Las claves pública y privada se obtienen a partir de grandes números primos. Aunque el algoritmo RSA es bastante seguro conceptualmente, existen algunos puntos que se deben tener en cuenta, como el tamaño de la clave, que no debería ser menor a 1024 bits; y no firmar el mensaje después de codificarlo, ya que existen algoritmos que permiten manipular con éxito mensajes primero codificados y luego firmados.

Existen otros algoritmos de criptografía asimétrica como el cifrado ElGamal[14], el DSA[15] o el protocolo de Diffie-Hellman[16] o Digital Signature Algorithm (algoritmo de firma digital).

1.7.3.2. Firma Digital

La firma digital es un método criptográfico que se aplica a un documento, el cual permite asegurar la identidad del firmante y la integridad del mensaje a partir de una huella digital o secuencia de bits añadida a la pieza original de la información. Este método presenta una analogía directa con la firma autógrafa, por ende debe cumplir con tres propiedades fundamentales:

- Estar ligada directamente a un mensaje y no ser válida para un documento diferente.
- Sólo puede ser generada por su emisor original, de la misma manera que una firma autógrafa se encuentra vinculada con una única persona.
- Es públicamente verificable, cualquier persona podría comprobar su identidad de manera sencilla.

[13] El sistema criptográfico con clave pública RSA es un algoritmo criptográfico asíncrono descrito en 1977 por Ron Rivest, Adi Shamir y Len Adleman del MIT, pero hecho público recién en 1977 ya que el mismo fue confidencial hasta entonces.

[14] El algoritmo de ElGamal fue descrito por Taher Elgamal en 1984. La seguridad del algoritmo se basa en la suposición que la función utilizada es de un sólo sentido y la dificultad de calcular un logaritmo discreto.

[15] DSA (Digital Signature Algorithm, en español Algoritmo de Firma digital) es un estándar del Gobierno Federal de los Estados Unidos de América o FIPS para firmas digitales. DSA se hizo público el 30 de agosto de 1991, este algoritmo como su nombre lo indica, sirve para firmar y no para cifrar información. Una desventaja de este algoritmo es que requiere mucho más tiempo de cómputo que RSA

[16] El protocolo Diffie-Hellman (debido a Whitfield Diffie y Martin Hellman) permite el intercambio secreto de llaves entre dos partes que no han tenido contacto previo, utilizando un canal inseguro, y de manera anónima (no autenticada). Se emplea generalmente como medio para acordar llaves simétricas que serán empleadas para el cifrado de una sesión. Siendo no autenticado, sin embargo provee las bases para varios protocolos autenticados. Su seguridad radica en la dificultad de calcular logaritmos discretos en un campo finito.

Como se muestra en la Figura 1.2 la forma más sencilla de generar una firma digital consiste en combinar cifrado asimétrico y funciones hash.

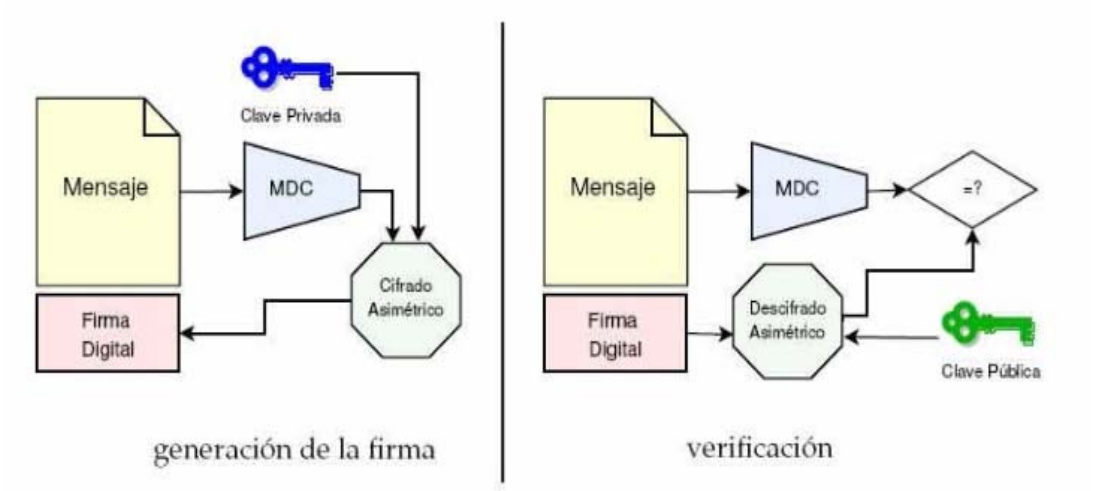

Figura 1.2 Firma Digital Utilizando Cifrado Asimétrico y Funciones Hash

Además de la criptografía existen otras ramas de las criptología empleadas para lograr los servicios de seguridad, una de estas ramas es llamada esteganografía la cual será descrita a continuación.

1.8. Esteganografía

La esteganografía es la rama de la criptología que trata sobre la ocultación de los mensajes para que no se perciba ni siquiera su propia existencia. En contraste con la criptografía, donde está claro que el mensaje existe pero el contenido se encuentra codificado, los mensajes en la estenográfica se ocultan en sí mismos, por ejemplo, ocultando texto manipulando el tamaño de la letra, el espaciado entre caracteres, o una muy utilizada, ocultándolo dentro de una imagen. Para esto existen varias aplicaciones que permiten codificar mensajes dentro de imágenes de una manera muy sencilla, donde se recibe como parámetro una imagen cualquiera, una contraseña y una frase que desea codificar y basado en ello se genera una imagen de aspecto idéntico con la frase oculta, a la cual hay que aplicarle una función para recuperarla.

Un ejemplo sencillo para entender la esteganografía es el siguiente:

Gerardo envía un mensaje a Rodolfo Figura1.3. De ser interceptado por un tercero, en este caso Oscar, Oscar al leer el mensaje piensa que no hay nada extraño y lo deja pasar a Rodolfo.

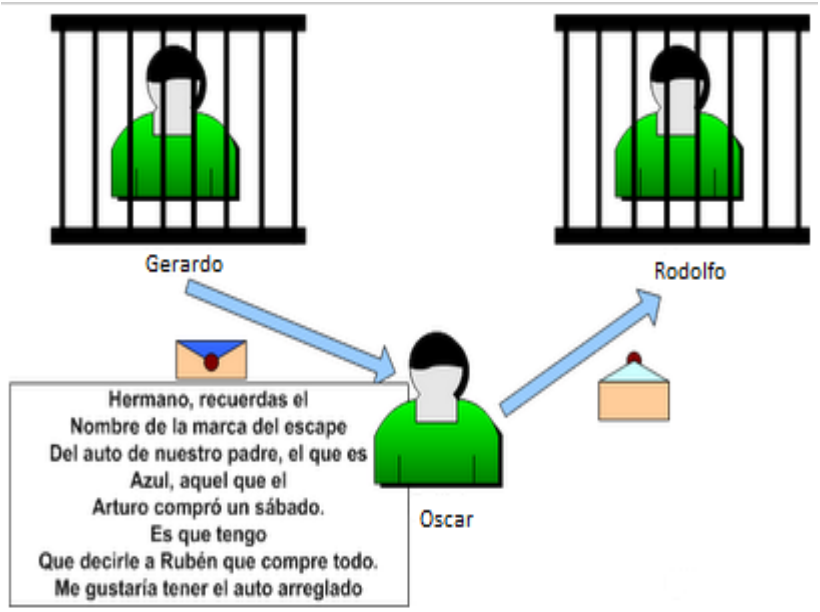

Figura 1.3 Mensaje Utilizando esteganografía

Sin embargo, esta nota tiene un mensaje oculto el cual se puede observar en la Figura 1.4.

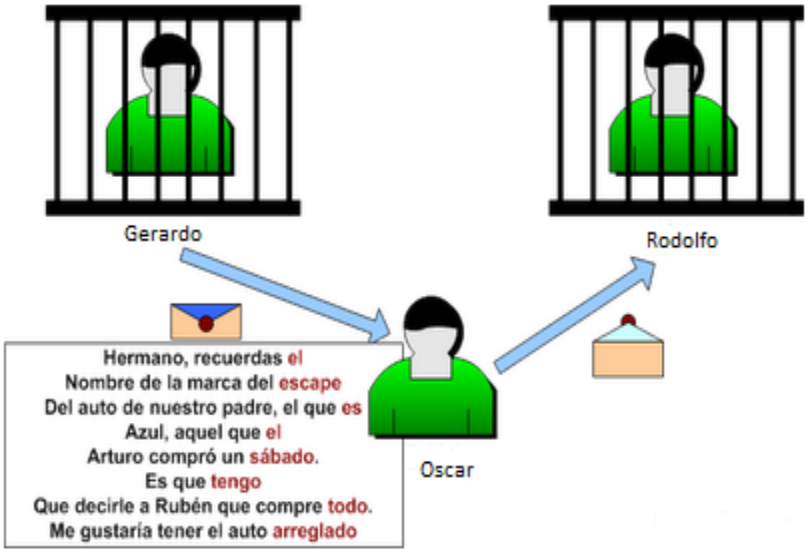

Figura 1.4 Mensaje Oculto con Esteganografía.

Con esto, Gerardo y Rodolfo planean el escape de la prisión sin levantar sospecha alguna. Ahora imaginemos un escenario pero ahora referente a informática. Oscar es un Hacker y está analizando todos los mensajes que envían Gerardo y Rodolfo. Si se utiliza la criptografía Oscar trataría de romper dicha codificación para obtener la

información oculta, en cambio si se utiliza la esteganografía Oscar no sabría que existen mensajes encubiertos en dicha comunicación, manteniendo el secreto a salvo.

En base a los servicios y mecanismos de seguridad es posible implementar y adoptar el modelo de seguridad que más se ajuste a las necesidades o prioridades de una empresa. Algunos modelos de seguridad son mencionados en el siguiente tema.

1.9. Modelos de Seguridad

El objetivo de un modelo de Seguridad Informática es mejorar la seguridad de la información y de los equipos, buscando asegurar que el trabajo que se realiza perdure. A continuación una breve explicación de los modelos de seguridad más comunes.

1.9.1. Seguridad por Oscuridad

A veces llamada seguridad por ocultación, la seguridad por oscuridad, es un método que utiliza el secreto de diseño o implementación para asegurar que, por desconocimiento, no “se encontrarán” los puntos débiles de dicho sistema.

1.9.2. Seguridad del Perímetro

La seguridad basada en la defensa perimetral apunta a reforzar los puntos de acceso o conexión de nuestra red privada con la red externa. Para ello se tiene que evaluar y planear qué tipos de acceso requiere el sistema, implementar sistemas de seguridad para bloquear el resto del tráfico, proteger esos únicos puntos vulnerables y ahí mismo ubicar sistemas de monitoreo y detección de intrusos para que den aviso al administrador del sistema y así poder ejecutar acciones defensivas a tiempo.

1.9.3. Seguridad en Profundidad

La seguridad en profundidad asume que cada una de las medidas tomadas pueden ser rotas por algún atacante. Sin embargo, a medida que se agreguen capas en el sistema de seguridad, la probabilidad de que el atacante pueda esquivar todas y cada una de ellas sin ser descubierto disminuye proporcionalmente. Esta metodología está basada en un conjunto de reglas cada vez más restrictivas a medida que el objeto a defender se encuentre más cercano. Estas medidas están delimitadas por áreas o zonas consideradas de la siguiente manera:

- **Área de Influencia:** Es la zona más externa del sistema, donde es factible realizar acciones contra la integridad de esta área.
- **Área de Exclusión:** Es el espacio concéntrico exterior al área protegida, de utilización restringida o acceso limitado.
- **Área Protegida:** Es el espacio delimitado por barreras físicas en el que se ejerce un cierto control de movimientos y permanencia.

- **Área Crítica:** Es el espacio delimitado por barreras físicas, en el interior del área protegida, cuyo acceso y permanencia son objeto de especiales medidas de control.

Ahora que se tiene un concepto más claro de seguridad informática es posible hablar de todas aquellas amenazas, vulnerabilidades y ataques a los que se enfrenta, los cuales son descritos en el siguiente capítulo.

Capítulo 2 .- Principales Amenazas, Ataques y Vulnerabilidades

Ya se ha hablado de manera general de lo que es la seguridad, lo que ofrece y lo que pretende proteger. Para este capítulo y haciendo referencia a los conceptos dados en el capítulo uno se pretende dar a conocer una visión más amplia de aquellas vulnerabilidades, amenazas y ataques que pudieran comprometer nuestros equipos o sistemas.

2.1. Amenazas Físicas

Este tipo de amenazas son producidas tanto por el hombre como por la naturaleza. Básicamente, las amenazas físicas que pueden poner en riesgo un sistema informático son:

- Desastres naturales, incendios accidentales, humedad e inundaciones.
- Amenazas ocasionadas involuntariamente por personas.
- Acciones hostiles deliberadas como robo.

En contraste con estas amenazas existen las amenazas lógicas, las cuales hacen referencia a fallos en software.

2.2. Amenazas Lógicas

Estas amenazas se refieren a todo tipo de programas que de una forma u otra pueden dañar a nuestro sistema, creados de forma intencionada para ello (software malicioso, también conocido como malware) o simplemente por error (bugs o agujeros). Muchas veces los protocolos de comunicación utilizados carecen en su mayoría de seguridad o ésta ha sido implementada en forma de "parche" tiempo después de su creación. Se puede señalar que existe una amenaza lógica cuando:

- Existen agujeros de seguridad en los sistemas operativos.
- Existen agujeros de seguridad en las aplicaciones.
- Existen errores en las configuraciones de los sistemas.
- Los usuarios carecen de información respecto al tema.

Esta lista podría seguir extendiéndose a medida que se evalúen mayor cantidad de elementos de un Sistema Informático.

La materialización de la mayoría de las amenazas lleva a un acceso y/o uso no autorizado de recursos. Por lo tanto es necesario tener claro los siguientes conceptos.

2.3. Acceso - Uso - Autorización

La identificación de estas palabras es importante ya que el uso de algunas implica un uso desapropiado de las otras.

Específicamente "Acceso" y "Hacer Uso" no son el mismo concepto cuando se estudian desde el punto de vista de un usuario y de un intruso. Por ejemplo:

- Cuando un usuario tiene acceso autorizado, implica que tiene autorizado el uso de un recurso.
- Cuando un atacante tiene acceso desautorizado está haciendo uso desautorizado del sistema.
- Pero, cuando un atacante hace uso desautorizado de un sistema, esto implica que el acceso fue autorizado (simulación de usuario).

Luego un ataque será un intento de acceso, o uso desautorizado de un recurso, sea satisfactorio o no. Un Incidente envuelve un conjunto de ataques que pueden ser distinguidos de otro grupo por las características del mismo (grado, similitud, técnicas utilizadas, tiempos, etc.).

Con los conceptos dados hasta el momento en este capítulo se pueden clasificar las amenazas de manera muy general, sin embargo para tener una protección adecuada contra ellas es necesario saber identificarlas desde un punto más granular.

2.4. Identificación de las Amenazas

La identificación de amenazas requiere conocer los tipos que existen y sus consecuencias.

Las consecuencias de los ataques se podrían clasificar en:

- **Data Corruption**: la información que no contenía defectos pasa a tenerlos.
- **Denial of Service (DoS)**: servicios que deberían estar disponibles no lo están.
- **Leakage**: los datos llegan a destinos a los que no deberían llegar.

Los anteriores son algunos ejemplos de las consecuencias que puede traer un ataque y estos dependen del tipo de amenaza a la que se está expuesto.

2.4.1. Tipos de Amenazas

Los tipos de amenazas pueden ser separados según su taxonomía en: interrupción, intercepción, modificación y fabricación.

- **Interrupción:** Existe cuando un recurso de un sistema es destruido o se hace indisponible o inusable. Es un tipo de ataque a la disponibilidad del recurso.
- **Intercepción:** Ocurre cuando una parte no autorizada logra capturar el objeto siendo éste, un receptor no autorizado. Esta parte podría ser una persona, un programa o una computadora. Ejemplos incluyen conectar a una red de datos privada para capturar paquetes de transmisión, o el copiado ilegal de archivos y programas.
- **Modificación:** Esto se logra cuando además de ganar el acceso al recurso, se modifica y reenvía al destino. Este ataque es a la integridad de los datos, donde por ejemplo, se podrían cambiar valores en un archivo o una base de datos, alterar un programa para que funcione distinto, o modificar los mensajes transmitidos en una red.
- **Fabricación:** Es cuando una parte no autorizada genera objetos en el sistema. Se considera un ataque a la autenticidad y como ejemplos se puede enumerar el agregado de filas o registros a una base de datos, la inserción de paquetes a una red, o en mayor escala, el envió de mensajes desde un servidor de correo electrónico controlado, mediante el reemplazo de identidad de origen del remitente.

Estas amenazas se pueden materializar aprovechando distintas vulnerabilidades en cualquier parte del sistema. El siguiente tema describe de manera breve las vulnerabilidades más conocidas.

2.5. Vulnerabilidades

Haciendo referencia al concepto dado en el capítulo uno, una vulnerabilidad es la materialización de una amenaza. La presencia de una falla, ya sea en la fase de diseño o implementación de un sistema, producto o componente, puede conducir de manera imprevista a comprometer la seguridad. A continuación se hará mención de las vulnerabilidades más comunes:

- **Bugs:** El bug es un error de software generado durante el proceso de creación del mismo cuando no se contemplan todos los posibles estados que el sistema puede tomar en tiempo de ejecución. Los errores más comunes pueden ser: división por cero, un ciclo infinito, desbordamiento de buffer (buffer overflow y underflow), utilización de variables no inicializadas, acceso a un área de memoria restringida, desbordamiento de stack, entre otros.

- **Backdoors:** También llamado “puerta trasera”, es una secuencia especial en el código, generada por el programador, para saltear el normal flujo del sistema.
- **Acceso Físico:** Otra forma de acceder a los datos es de forma directa a la terminal, con herramientas de crackeo de contraseñas de login como “John the ripper” con los que se puede fácilmente obtener el nombre de cuenta de administrador y sus contraseñas, y así, lograr tener el control completo de la computadora.
- **Bomba Lógica:** Es una rutina que hace que un programa que funciona adecuadamente, ‘explote’m en una fecha, momento determinado o condición especifica o generalmente deje de funcionar adecuadamente, pudiendo de manera optativa dañar información.
- **Caballo de Troya:** Es un programa útil o aparentemente útil que tiene comandos o procedimientos ocultos y que cuando es invocado, realiza funciones no queridas o dañinas, generalmente para tomar el control de la máquina.
- **Bacterias:** Son programas que no corrompen explícitamente archivos sino que su función es replicarse a sí mismas, realizando como principal daño, que el sistema se sature (disminuya su potencia computacional, espacio libre de memoria RAM, espacio libre en disco, etc.).
- **Virus:** Son programas ocultos que se replican y se liberan en determinadas fechas o momentos, en carga letal, que generalmente involucra la destrucción de información en el sistema infectado, desde algún archivo de manera aleatoria, hasta todos los discos del sistema infectado.
- **Spyware:** Los programas espías o spyware son aplicaciones que recopilan información sobre una persona u organización sin su conocimiento. La función más común que tienen estos programas es la de recopilar datos sobre el usuario y distribuirlo a empresas publicitarias. Dado que el spyware usa normalmente la conexión de una computadora a Internet para transmitir información, consume ancho de banda y por lo tanto puede verse afectada la velocidad de transferencia de datos de dicha computadora.
- **Rootkits:** Es una herramienta o grupo de ellas que tiene como finalidad esconderse a sí misma y esconder a otros programas, procesos, archivos, directorios, llaves de registro y puertos, que permiten al intruso mantener el acceso a un sistema para remotamente comandar acciones o extraer información sensible.
- **Worms:** Los gusanos o worms son semejantes a los virus en el sentido de que se esparcen, pero están orientados a otro tipo de contagio, intentando detectar debilidades en determinados programas que atienden servicios, como pueden ser servidores Web, Ftp, mails, entre otros. Una vez que encuentran una vulnerabilidad, envían un mensaje que produce un mal funcionamiento en el

servidor, donde queda una copia del worm, el cual comienza a escanear desde ese servidor otras redes.

- **Spam:** Esta técnica se basa en la recolección de grandes bases de datos de correo electrónico por parte de robots que recorren la web en busca de emails. Es fuente común de posteriores infecciones con virus y otras plagas masivas.
- **Scam:** Es una técnica especifica de Spam que normalmente sirve para obtener información privada y claves.
- **Denial of Service (DOS) o Negación de servicio:** Los ataques de DOS o Negación de servicio ocurre cuando se le impide a un usuario realizar la operación del sistema. Algunas de las maneras más comunes de realizar ataques de DOS son:
    - Consumo de ancho de banda – inundando una red con información
    - Escases de recursos – agotando los recursos de un sistema
    - Ataques de ruteo y DNS – manipulando las tablas de ruteo para que apunten a una dirección IP alternativa.
- **Ingeniería Social:** Es el uso de influencia y persuasión para engañar a las personas con el propósito de obtener información o realizar alguna acción. Todas las medidas de seguridad que la empresa adopta son en vano cuando un empleado es víctima de ingeniería social por algún extraño. Algunos ejemplos de ingeniería social incluyen el responder inconscientemente a las preguntas de un extraño o responder e-mails spam.
- **Secuestro de Sesión:** Secuestro de sesión significa tomar control sobre una sesión TCP intercambiada entre dos máquinas. Debido a que la autenticación únicamente ocurre al inicio de la sesión de TCP, un atacante puede obtener acceso a la máquina.

La mayoría de estas vulnerabilidades afectan al sistema operativo o software utilizado en el sistema expuesto pero además de estas amenazas existen también algunas que pueden afectar directamente a una aplicación web.

2.5.1. Ataques a Aplicaciones Web

Es importante señalar los diferentes tipos de vulnerabilidades que pueden ser descubiertas en las aplicaciones web, ya que estas se vuelven una parte importante de una empresa porque son la forma en la que ésta es representada en internet y pueden significar una vía de acceso hacia los activos de esta misma.

Las conductas de ataques a sitios web son las siguientes:

- **Deformación de Sitios Web:** es la más común y frecuente forma de cyber vandalismo y existen herramientas que pueden ser descargadas de internet para explotar estas vulnerabilidades.

- **Robo de Información de Tarjetas de Crédito:** Después de que un atacante obtiene acceso a la red de la empresa, este puede escanear las bases de datos en busca de cualquier archivo que contenga información valiosa como archivos de clientes que contengan información de tarjetas de crédito. Estos archivos pueden ser descargados a la máquina del atacante.
- **Ataque al Servidor:** Es posible corromper el servidor debido a:
    - Ejecución de comandos en el servidor web
    - Lectura de archivos del servidor web
    - Modificación de archivos en el servidor web
- **Buffer Overflows:** Es posible tirar (dejar fuera de servicio) un servidor ejecutando comandos arbitrarios en el sistema de la víctima haciendo que un programa escriba en el buffer más información de la que puede manejar el espacio asignado.
- **Ataques de Domain Name Server (DNS):** DNS es un protocolo mediante el cual la web traduce direcciones web (www.ejemplo.com) a direcciones IP (ej. 201.210.20.2). Errores en la programación y diseño permiten a un atacante envenenar el servidor DNS con información incorrecta y así desviar a los usuarios a otro destino.
- **Ataques de Distributed Denial of Service (DDOS):** En los ataques de DDOS o de negación de servicio distribuida, muchas computadoras son comprometidas para actuar como esclavos y después utilizadas para inundar un sitio con paquetes o peticiones de información y de esta manera negar el servicio a los usuarios del sistema víctima.
- **Utilización de Código Malicioso:** Utilizar código maliciosos para esparcir virus, gusanos y amenazas.

Al igual que con los ataques realizados al sistema operativo de un equipo, los ataques a aplicaciones web sólo pueden llevarse a cabo mediante la explosión de las vulnerabilidades a las que están expuestas. A continuación se señalan algunas de estas vulnerabilidades.

2.5.2. Vulnerabilidades de Aplicaciones Web

Este tipo de vulnerabilidades no se limitan a ataques basados sobre URL y puerto 80. A pesar de utilizar puertos, protocolos y capaz de OSI [17], la integridad de las aplicaciones de misión crítica debe ser protegida de posibles ataques futuros.

Algunos tipos de vulnerabilidades en aplicaciones web son los siguientes:

[17] OSI (Open System Interconnection) o modelo de referencia de Interconexión de Sistemas Abiertos. Es un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones creado por Organización Internacional para la Estandarización.

2.5.2.1. Cross Site Scripting (XSS)

Vulnerabilidades que ocurren cuando un atacante utiliza aplicaciones web y envía código malicioso programado en Java Script[18] a diferentes usuarios. La figura 2.1 muestra el funcionamiento de esta vulnerabilidad. Estas vulnerabilidades se pueden encontrar en cualquier aplicación que tenga como objetivo final, el presentar la información en un navegador web. No se limita a sitios web, ya que puede haber aplicaciones locales vulnerables a XSS o incluso el navegador en sí. El problema está en que usualmente no se validan correctamente los datos de entrada que son usados en cierta aplicación. Esta vulnerabilidad puede estar presente de forma directa (también llamada persistente) o indirecta (también llamada reflejada).

- **Directa (Persistente):** este tipo de XSS comúnmente filtrado y consiste en invadir código HTML peligroso en sitios que así lo permiten; incluyendo así etiquetas como lo son <script> o <iframe>.
- **Indirecta (Reflejada):** este tipo de XSS consiste en modificar valores que la aplicación web utiliza para pasar variables entre dos páginas, sin usar sesiones y sucede cuando hay un mensaje o una ruta en la URL del navegador, en una cookie[19] o cualquier otra cabecera HTTP (en algunos navegadores y aplicaciones web, esto podría extenderse al DOM del navegador).

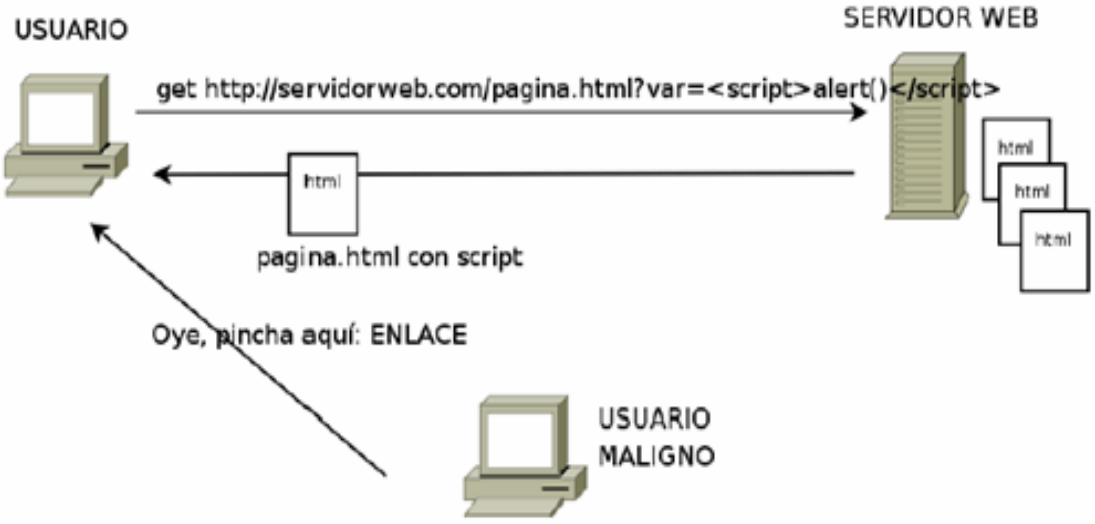

Figura 2.1 Funcionamiento de XSS

[18] JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

[19] Las cookies son pequeños archivos que los sitios web colocan en el disco duro del equipo cuando los visita por primera vez. Las cookies permiten almacenar preferencias y nombres de usuario, registrar productos y servicios, así como personalizar páginas.

2.5.2.2. SQL Injection

Estos ataques utilizan secuencias de comandos de sentencias SQL para controlar directamente la base de datos. Las aplicaciones frecuentemente utilizan sentencias de SQL para la autenticación de usuarios en la aplicación, validan roles y niveles de acceso, almacenamiento y obtienen información para la aplicación y el usuario. Utilizando métodos de SQL Injection un atacante puede utilizar una aplicación vulnerable para evadir las medidas de seguridad comunes y obtener acceso directo a información sensible. La razón por la que los ataques de SQL Injection funcionan es porque la aplicación no valida de forma adecuada los datos de entrada antes de convertirlos en una sentencia de SQL como se puede observar en la figura 2.2.

Aquí un ejemplo de SQL Injection:

*SELECT * FROM nombre_tabla WHERE id_usuario= 15*

Se convierte en lo siguiente con un simple ataque de SQL Injection

*SELECT * FROM nombre_tabla WHERE id_usuario=15 OR 1=1*

La expresión “OR 1=1” se evalúa como un valor verdadero, permitiendo la enumeración de los valores de id_usuario existentes en la base de datos. Los ataques de SQL Injection pueden permitir a un atacante:

- Acceder a la aplicación sin proporcionar credenciales válidas.
- Realizar búsquedas en la base de datos, incluso información a la que normalmente la aplicación no tiene acceso.
- Modificar el contenido de la base de datos o eliminar la base de datos completamente.
- Utilizar las relaciones de confianza establecidas entre los componentes de la aplicación web y acceder a otras bases de datos.

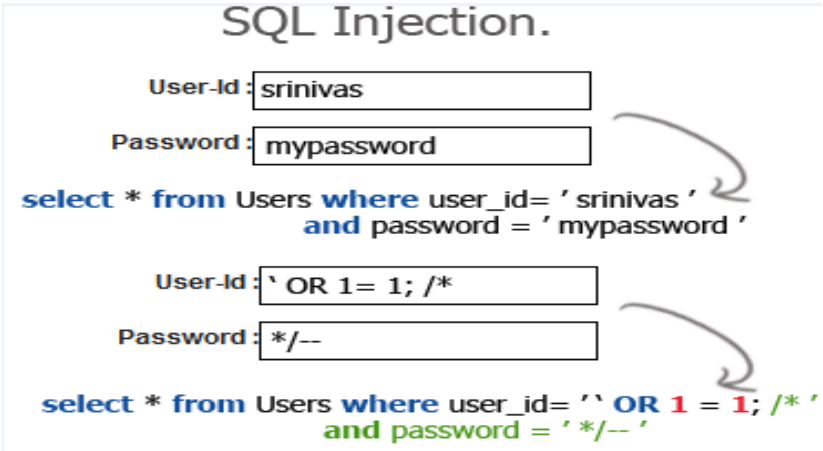

Figura 2.2 Funcionamiento de SQL Injection.

2.5.2.3. Errores de Inyección de Comando

Permite a los atacantes pasar código malicioso a diferentes sistemas vía la aplicación web. Los ataques incluyen utilizar comandos de Shell y llamadas a la base de datos mediante SQL.

2.5.2.4. Envenenamiento de Sesión/Cookie

Las cookies frecuentemente transmiten información de autenticación y pueden ser modificadas con facilidad para escalar privilegios o asumir la identidad de otro usuario.

2.5.2.5. Manipulación de Parámetros

La manipulación de parámetros es una manera sencilla de atacar directamente la lógica de negocio de la aplicación. Este ataque toma ventaja del hecho que muchos programadores confían en campos ocultos (como etiquetas ocultas o parámetros en una URL) como la única medida de seguridad para ciertas operaciones. Para evitar estos mecanismos de seguridad un atacante puede cambiar estos parámetros.

La manipulación de parámetro puede llevar a:

- Robo de servicios
- Escalación de privilegios
- Asumir la identidad de otro usuario y secuestro de sesión
- Los parámetros pueden permitir el acceso a información de código de la aplicación.

2.5.2.6. Recorrido de Directorio/Navegación Obligada

Cuando se otorga un acceso fuera del contexto definido de una aplicación, existe la posibilidad de revelar o modificar información de forma no deseada. Los componentes e información de una aplicación compleja típicamente son guardados en múltiples directorios. Una aplicación tiene la habilidad de recorrer estos directorios para localizar y ejecutar porciones de la aplicación. Un ataque de Recorrido de directorio/Navegación obligada ocurre cuando un atacante es capaz de explorar estos directorios y archivos fuera del acceso normal de la aplicación. Este ataque expone la estructura de los directorios de la aplicación, el servidor web y el sistema operativo. Con este nivel de acceso a la infraestructura de la aplicación, un atacante puede:

- Enumerar el contenido de los archivos y directorios
- Acceder a páginas que en otras circunstancias requerirían autenticación.
- Obtener información de la aplicación.
- Descubrir ID’s de usuario y contraseñas
- Localizar código fuente.
- Ver información sensible como información de clientes.

Este ejemplo obtiene el archivo “/etc/passwd”[20] de un sistema UNIX/LINUX:

http://www.sitioejemplo.com/../../../../etc/passwd

Si bien, todas las amenazas y vulnerabilidades que se han mencionado antes pueden ser peligrosas por sí solas, cabe señalar que muchas de ellas pueden ser llevadas a cabo en conjunto para realizar ataques más sofisticados y de mayor alcance.

Existen maneras infinitas en las que un atacante puede utilizar y combinar todas estas amenazas con el fin de atacar un equipo o red. En general las etapas de un ataque se describen en la figura 2.3.

Como se observa, los ataques a determinados sistemas suelen ser planificados, es decir, el atacante en primera instancia busca un cierto objetivo, luego obtiene información general de la estructura de red, versiones de software, etc., y posteriormente utilizará una estrategia aplicando una o generalmente varias técnicas para vulnerar el sistema.

Probablemente en algún momento busque algo de anonimato para hacer sus intromisiones, por ejemplo utilizando una red pública, uno o varios sistemas intermediarios, o un sistema interno (es decir, desde dentro de la organización) desde dónde comprometer a otros sistemas internos. Luego, según se desee, se pueden limpiar los rastros o mantener el control del recurso obtenido mediante puertas traseras o rootkits y luego aprovechar el sistema comprometido para saltar a otro objetivo.

Es importante señalar que esta tendencia de ataques planificados y personalizados se encuentra en aumento debido a la creciente dificultad de encontrar una única manera de vulnerar un sistema. La figura 2.3 muestra las etapas típicas de un ataque.

[20] El contenido del archivo /etc/passwd determina quién puede acceder al sistema de manera legítima y qué se puede hacer una vez dentro del sistema. En él se tiene registradas las cuentas de usuarios, así como las claves de accesos y privilegios.

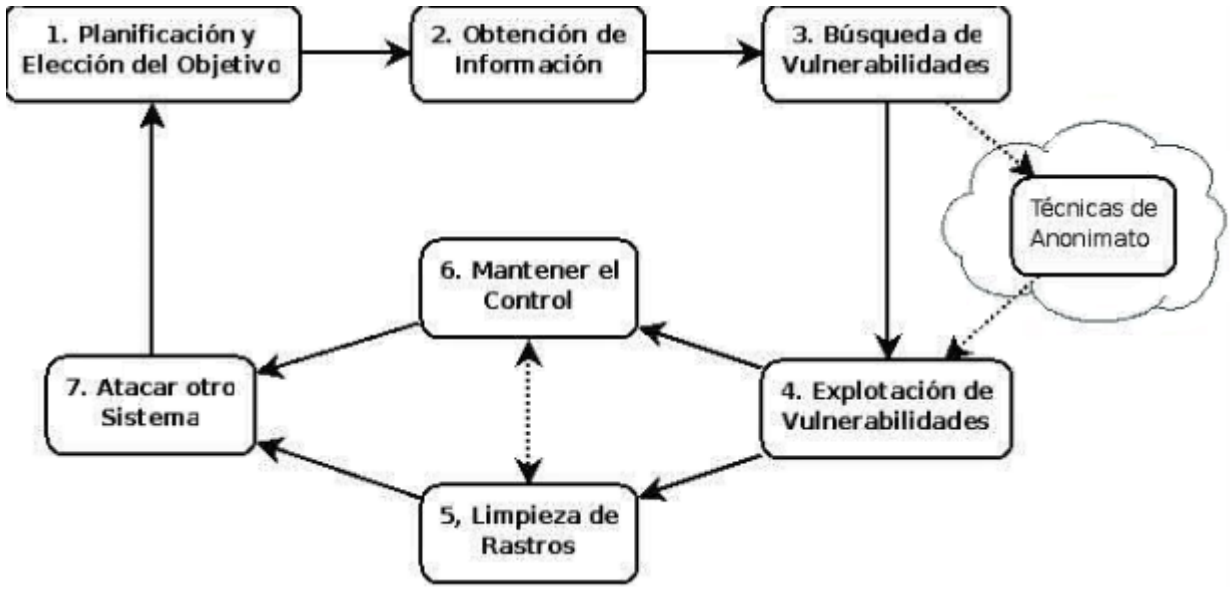

Figura 2.3 Etapas Típicas de un Ataque

Estas mismas etapas son utilizadas en la búsqueda de vulnerabilidades donde el ataque sirve para identificar los puntos más débiles en el sistema y a partir de esto tomar las medidas necesarias para su mitigación.

Las amenazas, vulnerabilidades y ataques antes mencionados son los más conocidos y los que suelen presentarse con mayor frecuencia, el siguiente capítulo describe algunas formas de prevención, detección y respuesta a todos ellos.

Capítulo 3.- Sistemas de Defensa

Ahora que se conocen las amenazas, vulnerabilidades y ataques que se presentan con mayor frecuencia en los incidentes de seguridad podemos hablar de los sistemas que en conjunto con los mecanismos de seguridad se utilizan para contrarrestarlos.

A continuación se mencionan los tipos de sistemas de seguridad que existen y después un listado de algunos de ellos.

3.1. Sistemas de Seguridad

Por muchos años la seguridad se basó en la prevención. Si se podía prevenir que alguien obtuviera acceso a los sistemas y redes se asumía que se había obtenido seguridad. Seguridad era entonces igual a prevención. Pero sin importar que tan sofisticado fuera el sistema de prevención alguien siempre encontraba la forma de evadirlos. Cuando esto ocurría el sistema quedaba desprotegido. Lo que se necesitaba eran múltiples técnicas de prevención y también tecnología que advirtiera cuando los sistemas de prevención habían fallado y diera una forma de solucionar el problema. Esto resultó en una modificación a la ecuación original con la adición de dos nuevos elementos, detección y respuesta. La ecuación queda de la siguiente forma:

$$Seguridad = Prevención + (Detección + Respuesta)$$

Donde cada elemento se obtiene de la siguiente manera:

- **Prevención:** Utilizando métodos de autenticación, identificación, control de acceso, transmisión segura, plataformas heterogéneas, sistemas honeypot
- [21], etc.
- **Detección:** A través de programas de auditoría encargados de realizar chequeos de integridad, proveer y presentar información al instante sobre el estado actual del sistema.
- **Respuesta:** Implementando métodos de backups y software de análisis forense (para detectar qué hizo el intruso y qué vulnerabilidad explotó).

Lo anterior es conocido como el modelo operacional de seguridad en cómputo. Cada técnica de seguridad que se utilice debe caer en al menos uno de los tres elementos de la ecuación como se observa en la figura 3.1.

[21] Se denomina Honeypot al software o conjunto de computadores cuya intención es atraer a atacantes, simulando ser sistemas vulnerables o débiles a los ataques.

Teniendo en cuenta este modelo operacional es posible implementar diferentes tipos de seguridad los cuales son descritos a continuación.

Figura 3.1 Modelo Operacional de Seguridad

3.2. Seguridad de Host (a nivel servidor)

Este tipo de seguridad toma una vista granular de la seguridad enfocándose en proteger cada equipo o dispositivo de manera individual en vez de proteger la red completa. Cuando este tipo de seguridad se utiliza, cada equipo se encarga de protegerse asimismo. Si una organización implementa este tipo de seguridad y no incluye seguridad en la red es muy probable que se presenten vulnerabilidades.

La seguridad a nivel servidor es importante y debe siempre ser atendida. Sin embargo, la seguridad no termina aquí, este es un proceso que se combina con la seguridad en la red.

3.3. Seguridad de Red

En sistemas pequeños, la seguridad a nivel servidor puede ser una opción, pero conforme los sistemas se van convirtiendo en redes, la seguridad debe incluirse en la red misma. En seguridad de red, se hace énfasis en controlar los accesos a equipos internos desde entidades externas. Este control puede ser a través de dispositivos como ruteadores, firewalls, hardware y software de autenticación, cifrado y un sistema de detección de intrusos (IDS)[22]

[22] IDS. Un sistema de detección de intrusos (o IDS de sus siglas en inglés Intrusion Detection System) es un programa usado para detectar accesos no autorizados a un computador o a una red. El funcionamiento de estas herramientas se basa en el análisis pormenorizado del tráfico de red, el cual al entrar al analizador es comparado con firmas de ataques conocidos o comportamientos sospechosos, como puede ser el escaneo de puertos, paquetes malformados, etc. El IDS no sólo analiza qué tipo de tráfico es, sino que también revisa el contenido y su comportamiento.

3.4. Mínimos Privilegios

Uno de los métodos que más se acerca a la seguridad es la de mínimos privilegios. Este concepto se aplica a ambientes de seguridad tanto de red como de equipos. Mínimos privilegios significa que un objeto (usuario, aplicación o proceso) debe tener únicamente los derechos necesarios para realizar una tarea sin permisos adicionales. Limitando los privilegios del objeto se limita la cantidad de daño que puede causar y así se limita el daño al que está expuesta la organización.

3.5. Seguridad en Capas

Un banco no protege el dinero que guardó utilizando únicamente una bóveda. Se tiene uno o más guardias de seguridad como primera defensa para vigilar actividades sospechosas. Seguramente tiene sistemas de monitoreo que vigilan las actividades realizadas dentro del banco, incluyendo empleados y usuarios. La bóveda se localiza al centro de las instalaciones rodeada de muros y cuartos antes de poder llegar a ella. Hay un control de accesos, el cual se asegura de que la persona que entre a la bóveda ha sido previamente autorizada. Y todos los sistemas están conectados a la estación de policía en caso de que un ladrón penetre de manera exitosa cualquiera de estas capas.

Las redes deben utilizar el mismo tipo de arquitectura de seguridad en capas. Es importante tener múltiples capas de seguridad. Estas capas pueden utilizar diferentes métodos como ruteadores, firewalls, segmentos de red, IDS's, cifrado, software de autenticación, seguridad física y control de tráfico. Las capas necesitan trabajar en conjunto y de manera coordinada para que una no impida el funcionamiento de la otra.

Las capas usualmente se representan empezando desde arriba con tipos más generales de protección y progresando hacia abajo a cada capa con un incremento granular en cada capa y acercándose cada vez más al recurso.

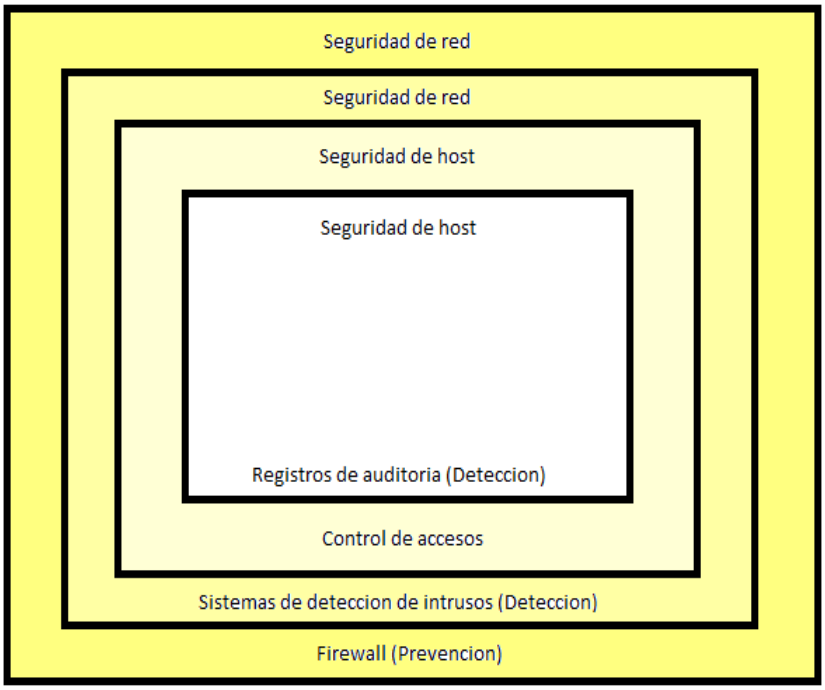

Figura 3.2 Seguridad en Capas

3.6. Controles de Accesos

El termino control de accesos se utiliza para describir diferentes esquemas de protección. Se utiliza a veces para hacer referencia a todas las características utilizadas para prevenir accesos no autorizados a equipos, sistemas y redes. En este sentido puede ser confundido con autenticación. De forma más específica, acceso es la habilidad de un individuo (persona, proceso o sistema) para interactuar con un objeto (archivo o dispositivo de hardware). En cambio la autenticación lidia con la verificación de la identidad del individuo. Para entender la diferencia, tomemos el ejemplo de una persona que intenta entrar a un equipo de cómputo, la autenticación es el proceso que el equipo utiliza para verificar que la persona es quien dice ser. El método más común para realizar esto es mediante un ID de usuario y una contraseña. Una vez que la identidad del individuo ha sido verificada, los controles de acceso regulan las acciones que el individuo puede realizar en el sistema. Sólo porque a una persona se le otorga acceso al sistema no quiere decir que puede tener accesos a toda la información contenida.

3.7. Mecanismos de Autenticación

Los controles de accesos definen las acciones que un usuario puede realizar o a que objetos puede tener acceso. Los controles asumen que la identidad del usuario ha sido verificada. El trabajo de los mecanismos de autenticación asegurarse que únicamente usuarios válidos son admitidos. Descrito de otra forma, la autenticación utiliza

mecanismos para probar que eres quien dices ser. Hay tres métodos generales utilizados en la autenticación. Para verificar tu identidad, puede proveer:

- Algo que sabes
- Algo que tienes
- Algo acerca de ti (algo que eres)

El mecanismo más común de autenticación es proveer algo que sólo tú, el usuario válido, sabe. Un ejemplo de esto es el usuario y contraseña. En teoría, esta información no debe ser compartida por lo tanto, únicamente tú conoces tu contraseña y es la forma en la que pruebas al sistema que eres quien dices ser.

Otro método de autenticación involucra el uso de algo que sólo los usuarios válidos poseen. Un ejemplo de esto sería una llave. Únicamente aquellos con la llave correcta serán capaces de abrir el cerrojo y obtener acceso ya sea a una casa, un auto, oficina o lo que sea que proteja el cerrojo. Un método similar puede ser utilizado para autenticar usuarios en un sistema de cómputo (mediante una llave electrónica, tarjeta inteligente o un dispositivo similar).

El tercer método de autenticación involucra algo acerca de ti. De manera física esto se utiliza por ejemplo, con las huellas digitales de las personas o su ADN, el cual puede ser utilizado para identificarlas. Este concepto puede ser utilizado para la autenticación en un sistema de cómputo. El área de la autenticación basada en algo acerca de ti o algo que eres es conocida como biometría[23].

Para terminar de señalar los tipos de seguridad, es importante hacer una mención especial a la seguridad física, la cual hace también referencia a las amenazas descritas en el capítulo dos como:

- Desastres naturales, incendios accidentales, humedad e inundaciones.
- Amenazas ocasionadas involuntariamente por personas.
- Acciones hostiles deliberadas como robo.

3.8. Seguridad Física

Consiste en todos los mecanismos utilizados para asegurarse que los accesos físicos a sistemas de cómputo están restringidos únicamente a usuarios autorizados. Cuando se piensa en seguridad física se deben considerar todos los puntos de acceso. No deben

[23] La biometría, o bio-identificación, es la práctica de la medición de las características físicas de una persona para verificar su identidad. Los sistemas biométricos más comunes son los de huella dactilares de la mano, algunos más avanzados son los de reconocimiento de voz y características en los ojos o la geometría de la totalidad de la cara.

ser examinados únicamente los puntos de acceso más obvios como puertas y ventanas, deben considerarse también las paredes, pisos y techos.

Para terminar este capítulo se señalan algunos ejemplos de mecanismos de prevención y detección que pueden ser utilizados en los tipos de seguridad antes mencionados así como una breve descripción de cada uno de ellos.

3.9. Algunos Mecanismos de Prevención y Detección

Ahora que se conocen los tipos de mecanismos de seguridad que existen se hace mención de algunos de los mecanismos de prevención y detección utilizados en diferentes sistemas.

- **Unicidad:** Consiste en incluir en los datos un número de secuencia, fecha/hora, número aleatorio, o alguna combinación de los anteriores para verificar la integridad de los mismos.
- **Control de Enrutamiento:** Permite enviar información por zonas clasificadas y a su vez permite solicitar y establecer rutas alternativas en caso de violaciones de seguridad.
- **Trafico de Relleno:** Consiste en enviar tráfico falso junto con los datos válidos para que el atacante no pueda diferenciar los reales de los falsos. Vinculado directamente con la esteganografía, es una forma de ocultar información en objetos que puedan llegar a pasar desapercibidos.
- **Cifrado:** Fundamental para garantizar la seguridad de la información. Consiste en aplicar un proceso de transformación a un texto claro mediante un cálculo matemático y así obtener un texto cifrado, inentendible para entidades no autorizadas.
- **Gestión de Claves:** Abarca la generación, distribución, almacenamiento, tiempo de vida, destrucción y aplicación de las claves de acuerdo a la política de seguridad aplicada.
- **Firewalls:** Los firewalls o cortafuegos son aplicaciones o equipos ubicados entre dos redes que establecen la política de acceso entre las partes.
- **Filtrado de Paquetes:** En este caso se realiza una lectura pormenorizada de la cabecera de cada paquete y en función a una serie de reglas se permiten el paso de los mismos o dicha trama es descartada. Los mismos pueden ser analizados verificando el protocolo utilizado, las direcciones de origen/destino, o los puertos de origen/destino.
- **Proxy de Aplicación:** Es un software encargado de eliminar las conexiones a servicios locales tales como FTP o Telnet y permiten únicamente la utilización de servicios en donde se encuentra un proxy.

- **Monitoreo de la Actividad:** Es indispensable para garantizar la seguridad del sistema, mantener monitoreada la actividad de las aplicaciones de seguridad y así poder detectar a tiempo los ataques a los que puede estar siendo sometido.

Hasta este punto ya se tiene un concepto de seguridad informática y se han mencionado las diferentes características que conllevan tanto sus servicios y mecanismos, así como las amenazas, vulnerabilidades y ataques a los que se enfrenta. Con los conceptos aprendidos es posible hablar de las “Pruebas de Penetración” descritas en el siguiente capítulo.

Capítulo 4.- Pruebas de Penetración (PenTest)

Este capítulo intenta dar de forma breve el concepto de pruebas de penetración así como su objetivo y los diferentes tipos de pruebas que existen. También se hace mención de las fases por las que atraviesan estas pruebas.

4.1. Concepto de Pruebas de Penetración

Las pruebas de penetración van más allá del escaneo de vulnerabilidades, a diferencia de este, el cual examina de manera individual los equipos, dispositivos de red o aplicaciones, las pruebas de penetración evalúan el modelo de seguridad de la red como un todo.

Las pruebas de penetración ayudan a los administradores de red, directores de TI e incluso a la parte ejecutiva de una empresa a conocer las dimensiones reales de un ataque real hacia la red.

Una prueba de penetración no sólo señalará las vulnerabilidades, también documentará cómo es que estas pueden ser explotadas y cómo muchas vulnerabilidades de menor importancia pueden ser aprovechadas por un atacante para comprometer un equipo o la red entera. Las pruebas de seguridad deben considerarse como una actividad que muestra los hoyos de seguridad en el modelo de seguridad de una organización.

Se debe señalar que una prueba de penetración se diferencia de un atacante únicamente por la ausencia de propósitos malintencionados. Esto quiere decir, que las pruebas que no son realizadas por profesionales pueden resultar en pérdidas de servicios y en una ruptura en la continuidad del negocio.

Cuando estas pruebas es realizan se debe tener siempre en claro el alcance de las pruebas, una descripción de lo que va a ser evaluado, dónde se llevarán a cabo las pruebas y la aprobación por escrito de la gerencia de la empresa. Debido a la naturaleza de las pruebas de penetración, al no obtener esta aprobación se puede caer en un delito informático a pesar de las buenas intenciones.

Es importante señalar que existen diferentes tipos de pruebas de penetración, basadas en ciertas características de las mismas, las cuales son descritas a continuación.

4.2. Tipos de Pruebas de Penetración

Las pruebas de penetración pueden ser clasificadas de acuerdo a la información que se tiene sobre el sistema que será probado o bien, según el lugar desde donde se realizan las pruebas. A continuación se mencionan los diferentes tipos de pruebas de penetración.

4.2.1. Prueba Externa

Una prueba de penetración externa es el método convencional para pruebas de penetración. La prueba se enfoca en los servidores, infraestructura y el software que esté relacionado al objetivo. Puede realizarse sin ningún conocimiento del sitio (Black box) o con información completa de la topología y ambiente (White box).

Este tipo de pruebas tendrán un análisis exhaustivo de la información pública disponible sobre el objetivo, una fase de enumeración donde los equipos objetivos serán identificados y analizados así como el comportamiento de los dispositivos de seguridad como dispositivos de filtrado de red. Las vulnerabilidades son entonces identificadas y verificadas y las consecuencias son evaluadas.

4.2.2. Prueba Interna

Las pruebas internas hacen uso de métodos similares a los de las pruebas externas y son consideradas como una vista más versátil de la seguridad. Las pruebas se realizan desde muchos puntos de acceso de la red incluyendo segmentos tanto físicos como lógicos.

Es fundamental tener en cuenta que a pesar de todo, la seguridad informática es un proceso que se lleva a cabo de manera continua y que las pruebas de penetración únicamente dan una vista instantánea de la postura de seguridad de una organización en ese momento.

4.2.3. Pruebas con Conocimiento Nulo (Black box)

Para poder simular un ataque real y minimizar los falsos positivos [24], los ejecutores de la prueba pueden optar por realizar una prueba de black-hat (sin información o asistencia del cliente) y mapear la red mientras enumeran los servicios, archivos de sistema compartidos y sistemas operativos de forma discreta. Además, se puede realizar wardialing[25] para detectar los módems activos y wardriving[26] para descubrir puntos de acceso vulnerables.

[24] Un falso positivo es un evento que se da como existente cuando realmente no existe, por ejemplo, decir que un sistema está infectado de virus cuando realmente está limpio.

[25] War dialing, es un método de escaneo automático de números telefónicos empleando un HmódemH para encontrar HcomputadorasH conectadas a estos. Este escaneo se realiza empleando programas llamados Hwar dialersH.

4.2.4. Pruebas con Conocimiento Parcial (Gray Box)

En ciertos casos, las organizaciones prefieren proveer información parcial como el servidor de nombre de dominio (DNS). Esto puede ahorrar tiempo y recursos en conocer el comportamiento de la organización.

Esta información puede incluir imágenes de los activos de la empresa y qué tan vulnerables se cree que son. También es posible que los ejecutores de la prueba interactúen con los administradores de la red.

4.2.5. Pruebas con Total Conocimiento (White box)

Si la organización necesita evaluar su seguridad contra un tipo específico de ataque o sobre un objetivo especifico, la información completa sobre él mismo puede ser entregada a las personas que realizarán las pruebas. La información entregada puede incluir documentos de la topología de la red, un inventario de activos, etc. Normalmente una organización opta por esto cuando requiere una completa auditoria de su seguridad.

Todas los tipos de pruebas de penetración pasan por las mismas fases para llevarlas a cabo, si bien los resultados varían dependiendo el sistema que se esté evaluando, las fases siempre son constantes.

4.3. Fases de una Prueba de Penetración

Ahora que se tiene un concepto más claro de lo que son las pruebas de penetración se presentan de manera individual cada una de las fases por las que atraviesa.

4.3.1. Fase de Pre-Ataque

Esta fase se enfoca en obtener la mayor información posible acerca del objetivo que será atacado. Puede ser invasiva o no invasiva dependiendo del tipo de reconocimiento que se realice.

4.3.1.1. Reconocimiento Pasivo

Comprende los intentos de un atacante de explorar o reconocer objetivos potenciales. Se resume en la obtención de información y puede incluir ingeniería social, atentados contra la seguridad física, etc. Suele hacerse de manera sigilosa y es la etapa en la que un atacante pasa más tiempo, incluso más que en el ataque en sí.

En el reconocimiento pasivo, la persona que realiza las pruebas recolectara toda la información que sea posible acerca de la compañía que está siendo evaluada. La mayor parte de la información que se obtiene es acerca de la topología de la red y los

[26] Wardriving es la acción de buscar Hredes inalámbricasH HWi-FiH empleando un vehículo en movimiento, utilizando una HcomputadoraH con antena Wi-Fi para detectar las redes

tipos de servicios que se utilizan dentro de ella. Esta información se utiliza provisionalmente para hacer un mapeado de la red y planificar una mejor coordinada estrategia de ataque.

El acceso a la información obtenida muchas veces es independiente de los recursos de la organización evaluada y puede ser accedida por cualquier persona. La información se obtiene con frecuencia en sistemas que no tienen relación con la empresa.

4.3.1.2. Reconocimiento Activo

El proceso de recolección de información va dirigido directamente a la organización. Aquí se explora el objetivo mediante escaneo de puertos, enumeración de archivos compartidos, cuentas de usuario, etc. Esto puede hacerse mediante el uso de herramientas que automatizan estas tareas como scanners y sniffers[27].

4.3.1.3. Resultados Esperados

En esta fase se puede obtener información como:

- Ubicación lógica y física de la organización
- Conexiones análogas como líneas telefónicas, fax, líneas dialup.
- Información de contactos ya sea en la red o en directorios telefónicos.
- Información acerca de otras organizaciones relacionadas a la organización evaluada.
- Cualquier información que sea potencialmente explotable. Esta información puede ser obtenida en chats, blogs, comunicados de prensa, foros, etc.

4.3.2. Fase de Ataque

Implica el compromiso del objetivo. Se explota una vulnerabilidad descubierta durante la fase de pre-ataque o se utilizan pequeños hoyos de seguridad como una política débil de seguridad para obtener acceso al sistema. El punto importante aquí es que un atacante necesita únicamente un punto de acceso mientras que la organización debe defender muchos de ellos. Una vez dentro, es posible escalar privilegios y lograr sus maliciosas intenciones.

4.3.2.1. Penetración del Perímetro

La ingeniería social es una actividad que se mantiene constante durante las pruebas de penetración y puede ser utilizada en cualquier momento de la prueba.

Las pruebas que pueden llevarse a cabo en este contexto incluyen (pero no se limitan únicamente a estas) hacer llamadas personalmente para obtener información sensible, correo electrónico, intentos para obtener detalles de autenticación legítima como

[27] Un sniffer es un programa para monitorear y analizar el tráfico en una red.

contraseñas y privilegios de acceso. La información obtenida aquí puede emplearse después en las pruebas en aplicaciones web.

Usualmente las pruebas de perímetro miden la habilidad del firewall de manejar la fragmentación: fragmentación de paquetes grandes, superposición de paquetes, desbordamiento de paquetes, etc. Los métodos utilizados para las pruebas de seguridad del perímetro incluyen:

- Evaluación de reportes de errores y gestión de errores con pruebas de ICMP
- Verificación de las listas de control de acceso con paquetes modificados
- Midiendo el umbral de Negación de servicio intentando conexiones persistentes de TCP, evaluando conexiones transitorias de TCP.
- Evaluando las reglas de filtrado de protocolos intentando hacer conexiones utilizando diferentes protocolos como SSH, FTP y Telnet.
- Evaluando la capacidad del IDS mediante contenido malicioso (como URLs malformadas).
- Examinando la respuesta del sistema de seguridad del perímetro a escaneos a los servidores web utilizando diferentes métodos como POST, DELETE y COPY.

4.3.2.2. Adquisición del Objetivo

Se refiere a todas las actividades que se realizaron para la recolección de toda la información posible acerca de una máquina o sistema en particular para ser utilizada en el proceso de explotación.

Ejemplo de estas actividades:

- Utilizar los resultados de los escaneos de red para obtener información que pueda llevar al compromiso del equipo o sistema.
- Correr escaneos de vulnerabilidades: Estos escaneos completan esta fase.

4.3.2.3. Escalación de Privilegios

Una vez que se ha llegado al objetivo, se intenta explotar el sistema y obtener acceso a recursos protegidos.

Actividades que pueden incluirse:

- Tomar ventaja de las políticas de seguridad para obtener información que conduzca a la escalación de privilegios.
- Uso de técnicas como fuerza bruta para alcanzar un estado con privilegios.
- Utilizar troyanos y analizadores de protocolos.

- Obtener acceso no autorizado a recursos, utilizar información obtenida a través de técnicas de ingeniería social.

4.3.2.4. Ejecutar, Implementar y Retirar

Se intentará ejecutar código arbitrario, archivos ocultos en el sistema comprometido y dejar al sistema sin notificación de alarmas. Entonces se intentarán actividades como:

- Ejecutar exploits[28] para tomar ventaja de las vulnerabilidades identificadas en el sistema.
- Utilizar técnicas como buffer overflow para introducir código arbitrario. Generar una consola de línea de comandos remota e intentar subir archivos y esconderlos en el sistema.
- Utilizar virus para explotar el sistema. Ser capaz de implementar un rootkit o un troyano que conduzcan al acceso de sistemas críticos también puede ser parte del proceso de pruebas.

Es común que un atacante remueva la evidencia de sus acciones borrando los archivos de registros. Las actividades de la fase de retiro incluyen la manipulación de estos registros para eliminar las actividades que se registraron.

4.3.3. Fase Post-Ataque

Esta fase es crítica en cualquier prueba de penetración y es responsabilidad de quien ejecuta la prueba regresar los sistemas a su estado previo a estas.

4.3.3.1. Actividades de la Fase Post-Ataque

Las actividades en esta fase incluyen:

- Eliminar los archivos subidos al sistema.
- Eliminar todas las entradas al registro.
- Revertir todas las manipulaciones hechas a archivos y configuraciones realizadas durante las pruebas.
- Revertir los cambios realizados a las configuraciones de usuarios.
- Eliminar las herramientas y exploits utilizadas.
- Documentación de los registros realizados en las pruebas.
- Analizar los resultados y presentarlos de forma organizada.

En el siguiente capítulo se hace una demostración de una prueba de penetración donde se muestran algunos de los ataques más comúnmente utilizados haciendo uso de las fases de las pruebas de penetración que se acaban de mencionar.

[28] Una forma definida de romper la seguridad de un sistema a través de vulnerabilidades conocidas es conocida como Exploit.

Capítulo 5.- Pruebas de Penetración (PenTest)

En este capítulo se hace una pequeña demostración de una prueba de penetración para tener más claro la forma en que se llevan a cabo. Se realizarán dos de las fases mencionadas en el capítulo anterior: Fase de Pre-Ataque y Fase de Ataque.

5.1. Ambiente de las Pruebas

Por motivos legales no es posible realizar una prueba a alguna empresa real sin su consentimiento, por lo tanto, se realizará sobre un laboratorio donde se tendrá una aplicación, la cual permite realizar algunos ataques explotando algunas vulnerabilidades. Las características de la prueba son las siguientes:

- 192.168.50.129 – IP de la aplicación sometida a las pruebas.
- La prueba es de tipo Gray-Box puesto que lo único que se sabe al iniciar es la IP del servidor de la aplicación.
- El primer objetivo es encontrar las vulnerabilidades en el servidor si es que existen.
- El segundo objetivo es obtener acceso a la aplicación contenida en el servidor. Una vez logrado el acceso a la aplicación, dentro de esta, se encuentran pequeñas aplicaciones las cuales se utilizarán para hacer una demostración de algunos ataques.

El desarrollo de las pruebas se hará mediante algunas herramientas que son comúnmente utilizadas en pruebas de penetración.

5.2. Herramientas Utilizadas

El siguiente listado contiene las herramientas utilizadas en la prueba de penetración para la fase de pre-ataque:

5.2.1. Nmap

Nmap es un programa de código abierto que sirve para efectuar rastreo de puertos escrito originalmente por Gordon Lyon (más conocido por su alias Fyodor Vaskovich). Se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir

servicios o servidores en una red informática. Nmap cuenta con las siguientes características:

- Descubrimiento de servidores: Identifica computadoras en una red, por ejemplo, listando aquellas que responden ping.
- Identificar puertos abiertos en una computadora objetivo.
- Determinar qué servicios está ejecutando la misma.
- Determinar qué sistema operativo y versión utiliza dicha computadora, (esta técnica es también conocida como fingerprinting).
- Obtiene algunas características del hardware de red de la máquina objeto de la prueba.

5.2.2. Acunetix Web Vulnerability Scanner

Es una herramienta de escaneo de vulnerabilidades sobre aplicaciones web el cual incluye características como:

- Pruebas de SQL Injection y Cross Site Scripting.
- Detección automática de servidor web.
- Editor de HTTP.
- Pruebas de seguridad sobre Ajax y Web 2.0.
- Analizador de contenidos web.

5.2.3. Nessus

Nessus. Es un programa de escaneo de vulnerabilidades en diversos sistemas operativos. Consiste en nessusd, el daemon Nessus, que realiza el escaneo en el sistema objetivo, y nessus, el cliente (basado en consola o gráfico) que muestra el avance y reporte de los escaneos. Desde consola nessus puede ser programado para hacer escaneos programados.

En operación normal, nessus comienza escaneando los puertos con nmap o con su propio escaner de puertos para buscar puertos abiertos y después intentar varios exploits para atacarlo. Las pruebas de vulnerabilidad, disponibles como una larga lista de plugins, son escritos en NASL (Nessus Attack Scripting Language, Lenguaje de Scripting de Ataque Nessus por sus siglas en inglés), un lenguaje scripting optimizado para interacciones personalizadas en redes.

5.2.4. Wireshark

Wireshark, antes conocido como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones para desarrollo de software y protocolos y como una herramienta didáctica para educación. Cuenta con todas las características estándar de un analizador de protocolos.

Contiene opciones de organización y filtrado de información. Así, permite ver todo el tráfico que pasa a través de una red (usualmente una red Ethernet, aunque es compatible con algunas otras) estableciendo la configuración en modo promiscuo.

Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada a través de los detalles y sumarios por cada paquete. Wireshark incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

5.2.5. Brutus

Brutus es una herramienta gratuita de crackeo de contraseñas en línea disponible para sistemas operativos Windows. Brutus puede ser utilizado en diferentes tipos de autenticación:

- HTTP (autenticación básica).
- HTTP (forma de HTML).
- POP3
- FTP
- SMB
- Telnet

5.2.6. Paros

Paros es una aplicación que funciona como un Proxy y que permite capturar las peticiones tanto HTTP como HTTPS para simplemente registrarlas a modo de debug o para poder modificarlas.

5.3. Desarrollo de las Pruebas

Lo primero, como se mencionó en el capítulo anterior, es la fase de pre-ataque donde se obtiene la mayor cantidad de información posible sobre el objetivo.

5.3.1. Recolección de Información

1. Como primer paso se utiliza la herramienta NMAP para descubrir los puertos y servicios que se encuentran activos en el servidor.

La siguiente tabla muestra el resultado del escaneo. Como se observa NMAP descubrió 6 puertos abiertos en el servidor corriendo diferentes servicios cada uno:

Puerto	Estado	Servicio	Versión
80/tcp	Open	http	Apache httpd 2.2.12 ((Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0)
135/tcp	Open	Msrpc	Microsoft
139/tcp	Open	netbios-ssn	
443/tcp	Open	ssl/http	Apache httpd 2.2.12 ((Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0)
445/tcp	Open	microsoft-ds	Microsoft Windows XP Microsoft-ds
3306/tcp	Open	Mysql	MySQL (unauthorrizated)

Tabla 5.1 Puertos y Servicios Obtenidos con NMAP

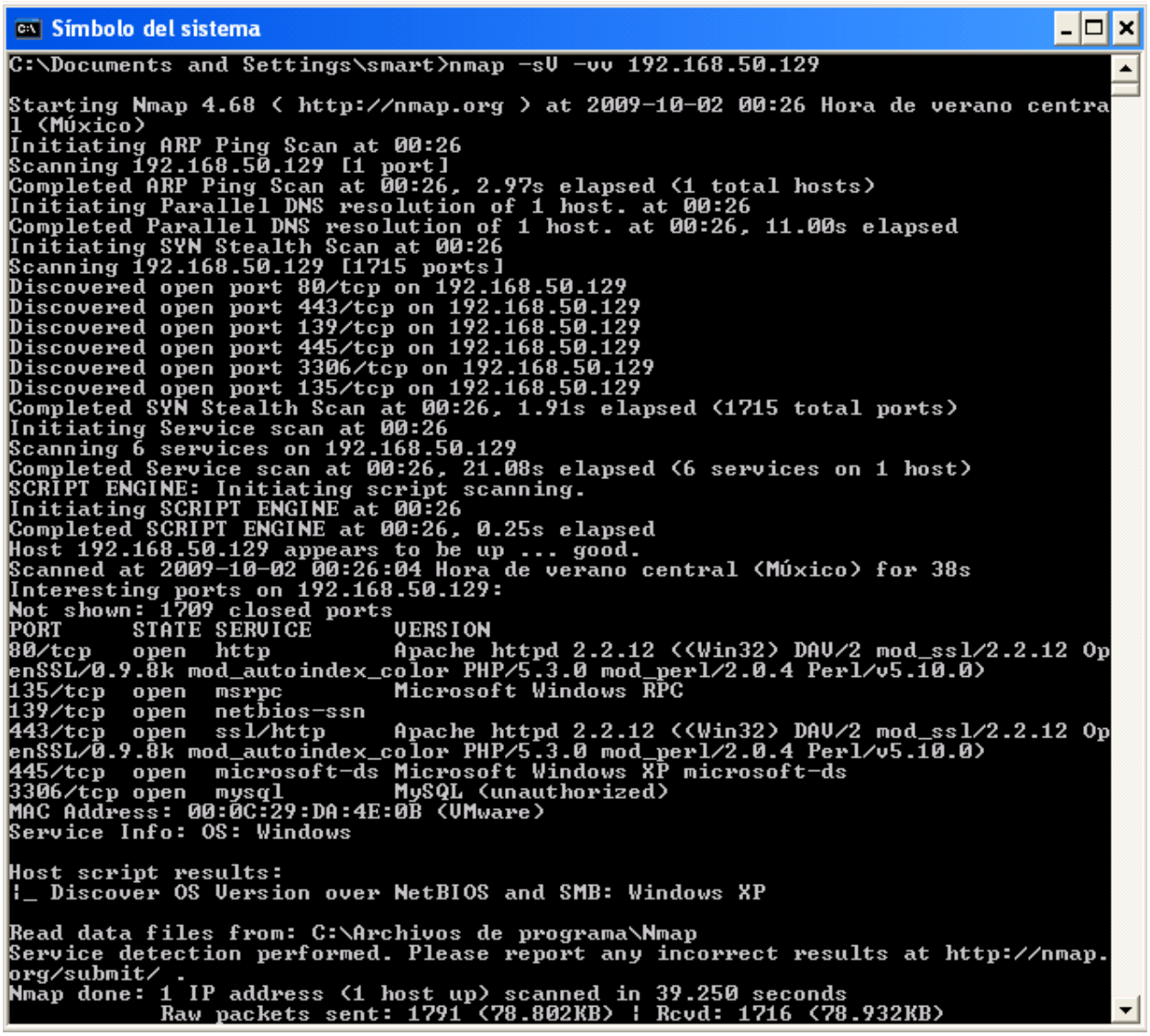

Figura 5.1 Escaneo con NMAP a IP 192.168.50.129

De lo anterior se obtiene lo siguiente:

- Se tienen abiertos los puertos 80 y 443, ambos con un servicio de Apache (servidor web).
- El sistema operativo del servidor es Windows XP.
- El servidor utiliza una base de datos MySQL.

Las figuras 5.2 y 5.3 corroboran que los puertos 80 y 443 son utilizados por un servidor web donde se ejecuta una aplicación de nombre DVWA.

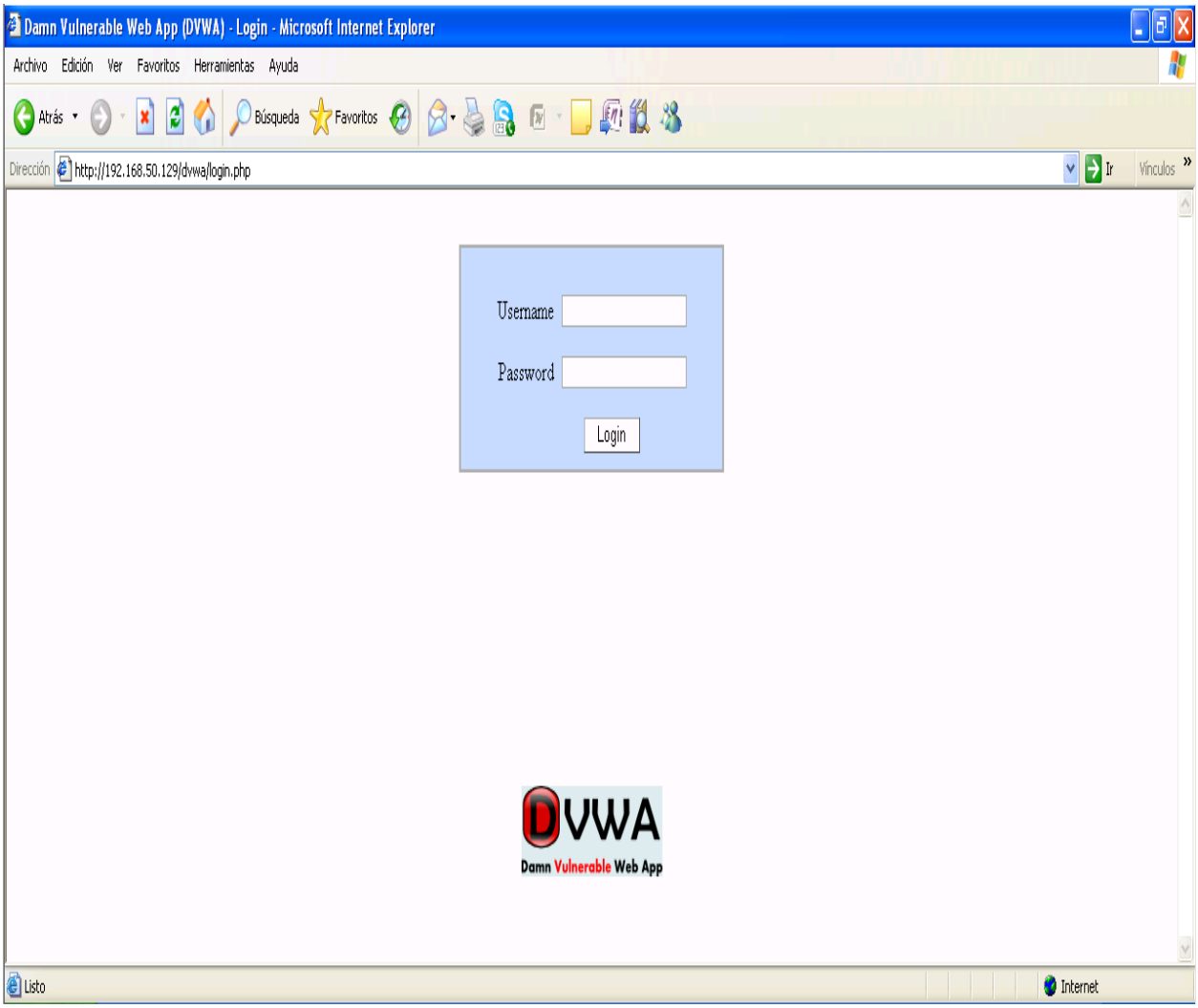

Figura 5.2 Acceso a la Aplicación por el Puerto 80

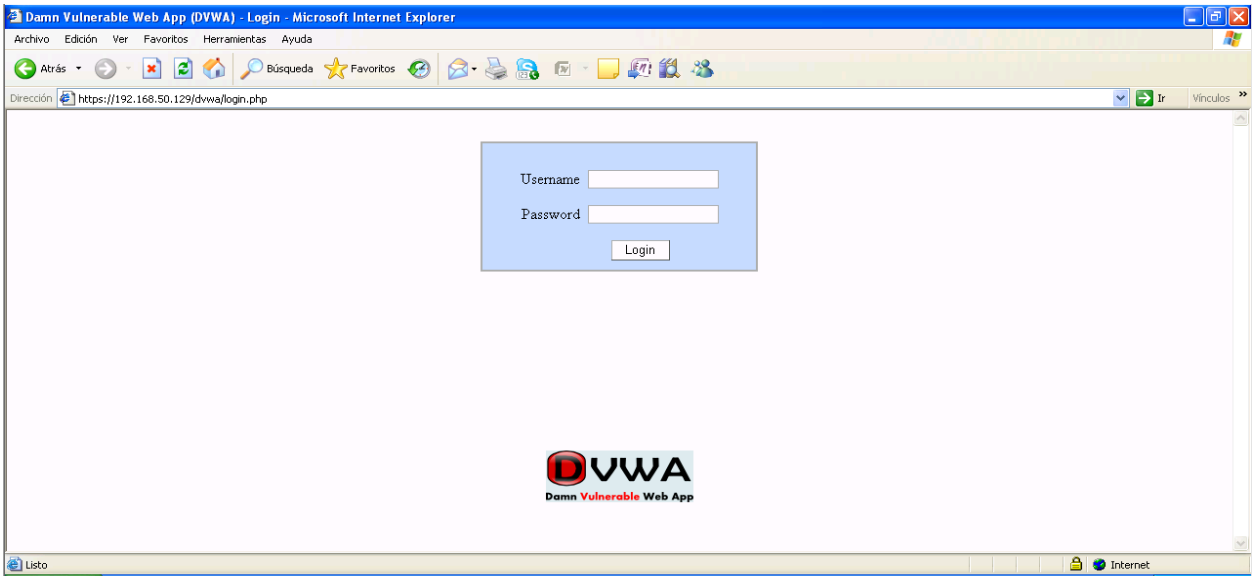

Figura 5.3 Acceso la Aplicación por el Puerto 443

2. Una vez que se sabe de la existencia de la aplicación web se procede a revisar las vulnerabilidades que esta puede tener. Para lo anterior se utiliza la herramienta Acunetix Web Vulnerability Scanner.

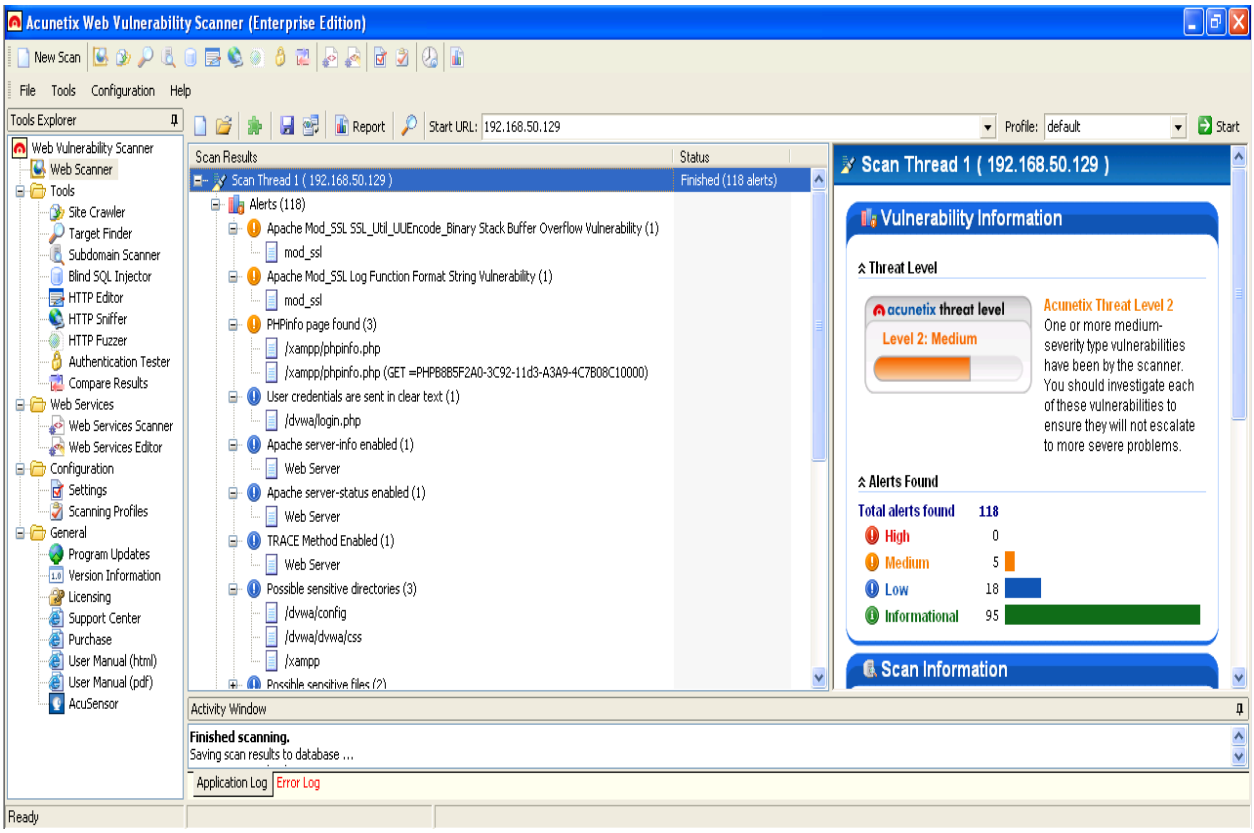

Figura 5.4 Escaneo de Vulnerabilidades con Acunetix Web Vulnerability Scanner Parte 1

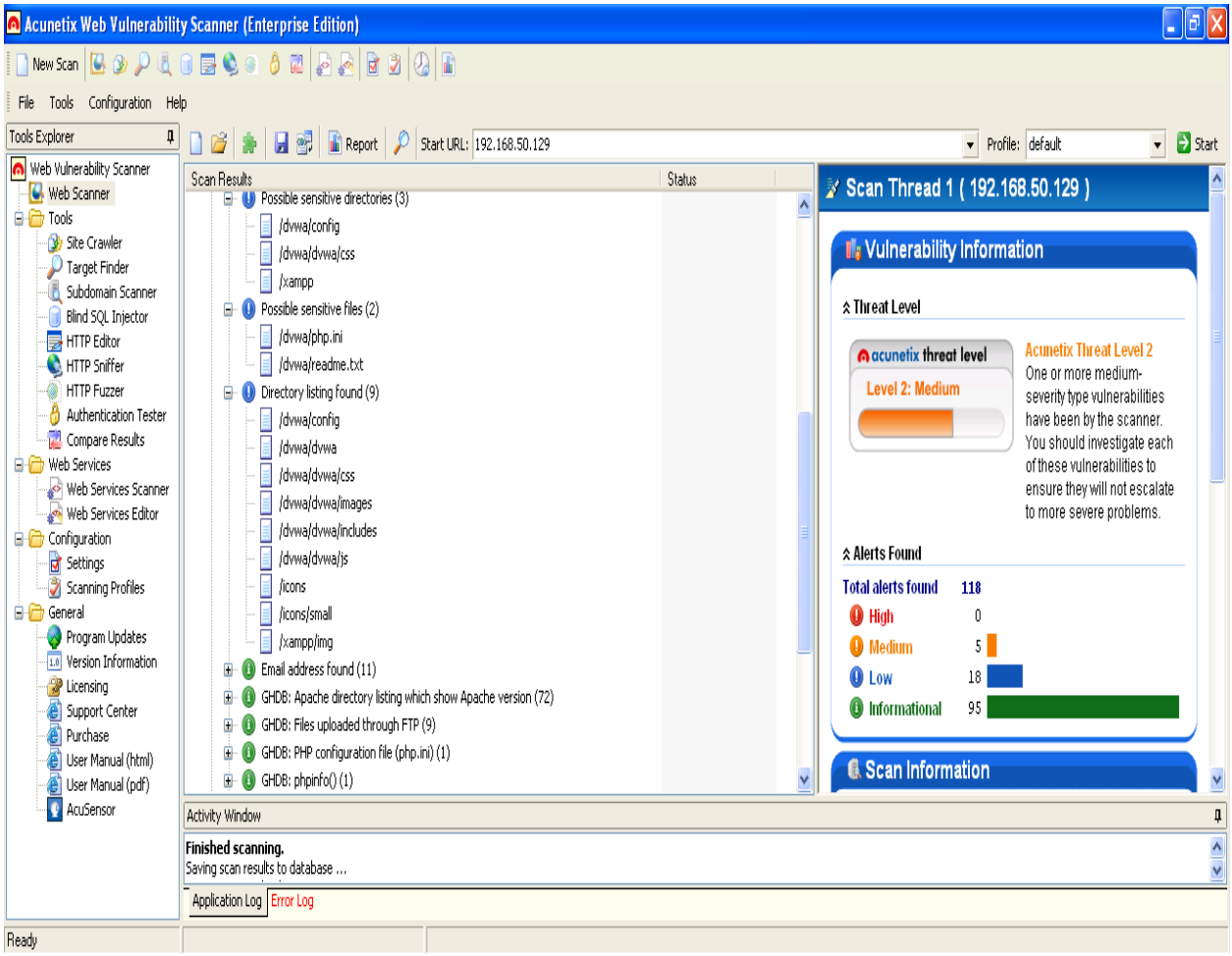

Figura 5.5 Escaneo de Vulnerabilidades con Acunetix Web Vulnerability Scanner Parte 2

Las figuras 5.4 y 5.5 muestran los resultados obtenidos por Acunetix Web Vulnerability Scanner donde los puntos más relevantes son los siguientes:

- Las credenciales de usuario son enviadas en texto claro, esto porque el servicio ofrecido en el puerto 80 no cuenta con cifrado de las comunicaciones.
- El servidor Apache utiliza una versión de SSL desactualizada.
- Se encuentro la página de PHPinfo. Esta página muestra información sobre el servidor de PHP, información del servidor, configuración, etc.
- Es posible listar directorios en el servidor. Esto puede revelar información de la configuración del servidor

La figura 5.6 muestra que es posible acceder a la página de PHPinfo donde se puede consultar información de la configuración del servidor. La figura 5.7 demuestra que es posible realizar un listado de directorios en la aplicación, aunque en este caso no se encontró información relevante.

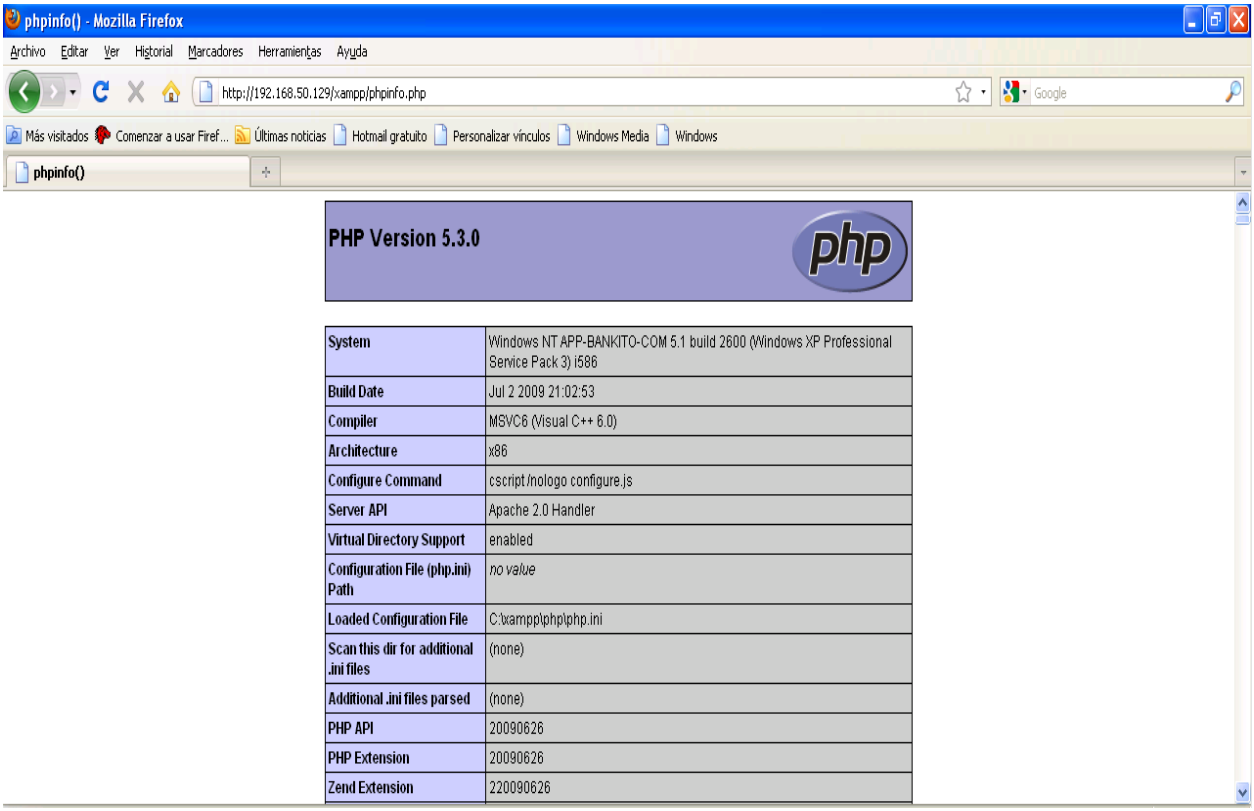

Figura 5.6 Página de PHPinfo

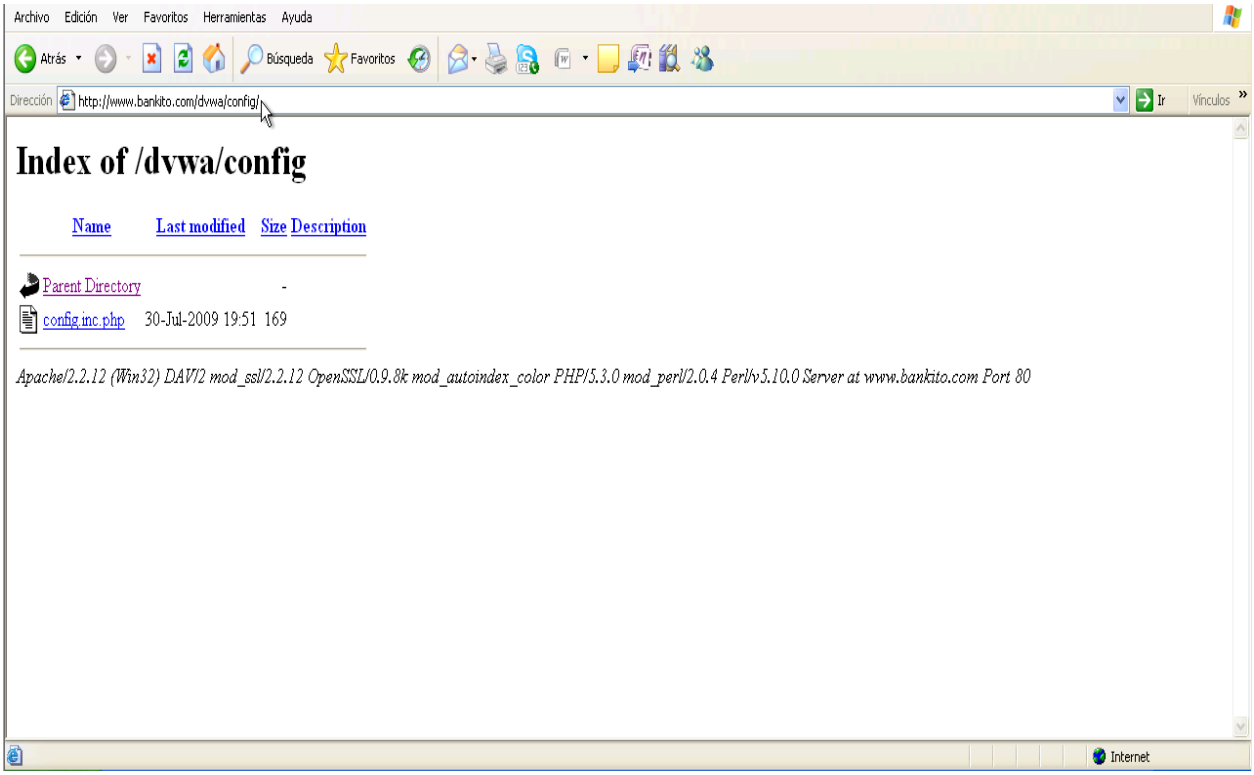

Figura 5.7 Listado de Directorios en la Aplicación

3. Ya se conocen algunas vulnerabilidades que presenta la aplicación que se está ejecutando en el servidor mediante el escaneo realizado, ahora con Nessus se hace un escaneo general en busca de vulnerabilidades no sólo de la aplicación sino del servidor como tal.

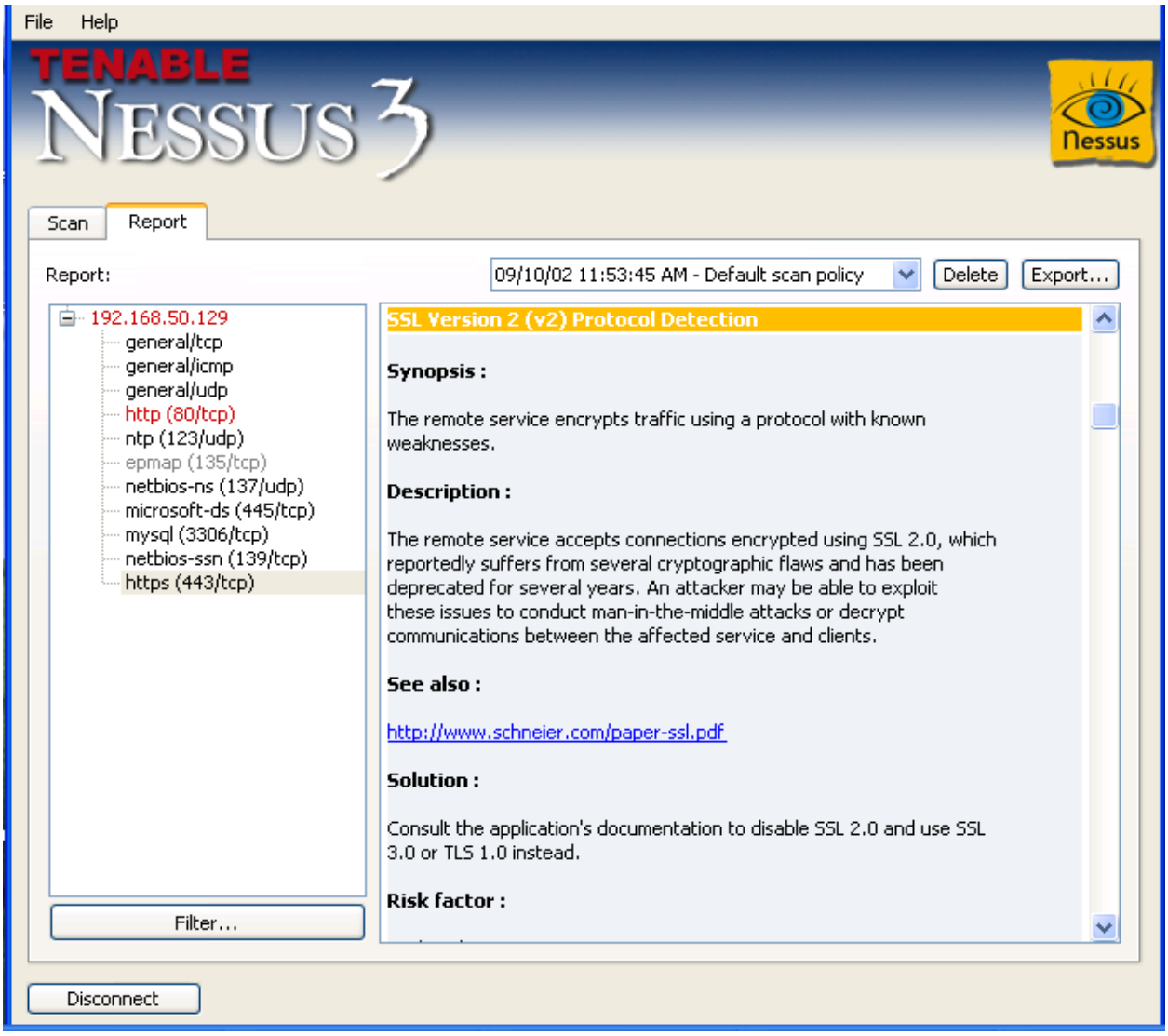

Figura 5.8 Resultados Nessus (SSL desactualizado)

Como se observa en la figuras 5.8, los resultados de Nessus no varían de los obtenidos anteriormente. El punto más destacado en Nessus es que la versión de SSL que utiliza en servidor esta desactualizada.

4. Ahora que se tiene información sobre el objetivo se procede a hacer un análisis de esta para definir cuál será la línea de ataque.

5.3.2. Análisis de la Información

De los 3 escaneos realizados se obtuvo la siguiente información:

- Se tiene una aplicación web que se ejecuta sobre un servidor web Apache. La aplicación puede ser accedida mediante los puertos TCP 80 y 443. Las credenciales de usuario se envían en texto claro cuando se accede desde el puerto 80.
- Se pueden listar los archivos del directorio donde se encuentra la aplicación, sin embargo, no se halló ninguna información útil.
- Se está utilizando una base de datos MySQL.
- El sistema operativo del servidor es Windows XP. No se encontró ninguna vulnerabilidad propia del servidor.

En base a los puntos anteriores se opta por atacar de forma directa la aplicación ya que no se encontró algún hoyo de seguridad en el servidor. Con esto termina la fase de pre-ataque y se empieza con la fase de ataque.

5.3.3. Ataques a la Aplicación

De acuerdo a la información obtenida en la fase de pre-ataque, la aplicación envía las credenciales de acceso de los usuarios en texto claro cuando esta es accedida por el puerto 80. Para comprobar esto se utiliza la herramienta WireShark.

Como se observa en la figura 5.9, después de realizar sniffing en la red, los paquetes capturados se filtran por el protocolo HTTP. Se puede observar que la IP 192.168.50.131 realizó una conexión a la aplicación, eso quiere decir, que las credenciales que envió para su autenticación están en texto claro. Las credenciales obtenidas son las siguientes:

- Usuario= 1337
- Contraseña= charley

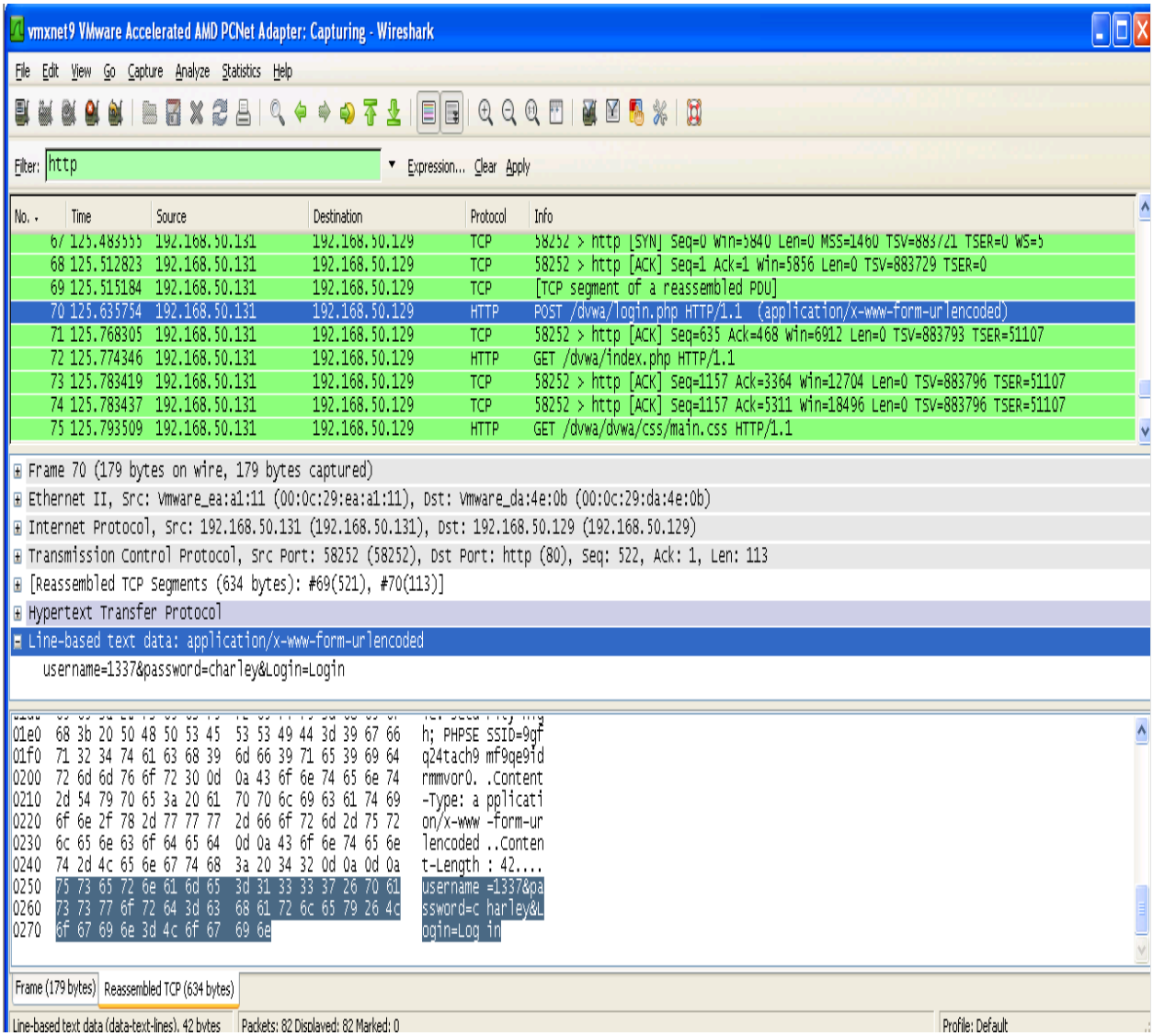

Figura 5.9 Obtención de Credenciales de Usuario con Wireshark

Cabe mencionar que para lograr capturar los paquetes se debe estar en el mismo segmento de red.

De esta forma se han obtenido credenciales de acceso a la aplicación. Otra forma de conseguir esto es mediante ataques de fuerza bruta sobre la página de autenticación de la aplicación (figura 5.2). Este ataque se puede realizar con la herramienta Brutus (Figura 5.10), el ataque para este caso es realizado con diccionarios de palabras.

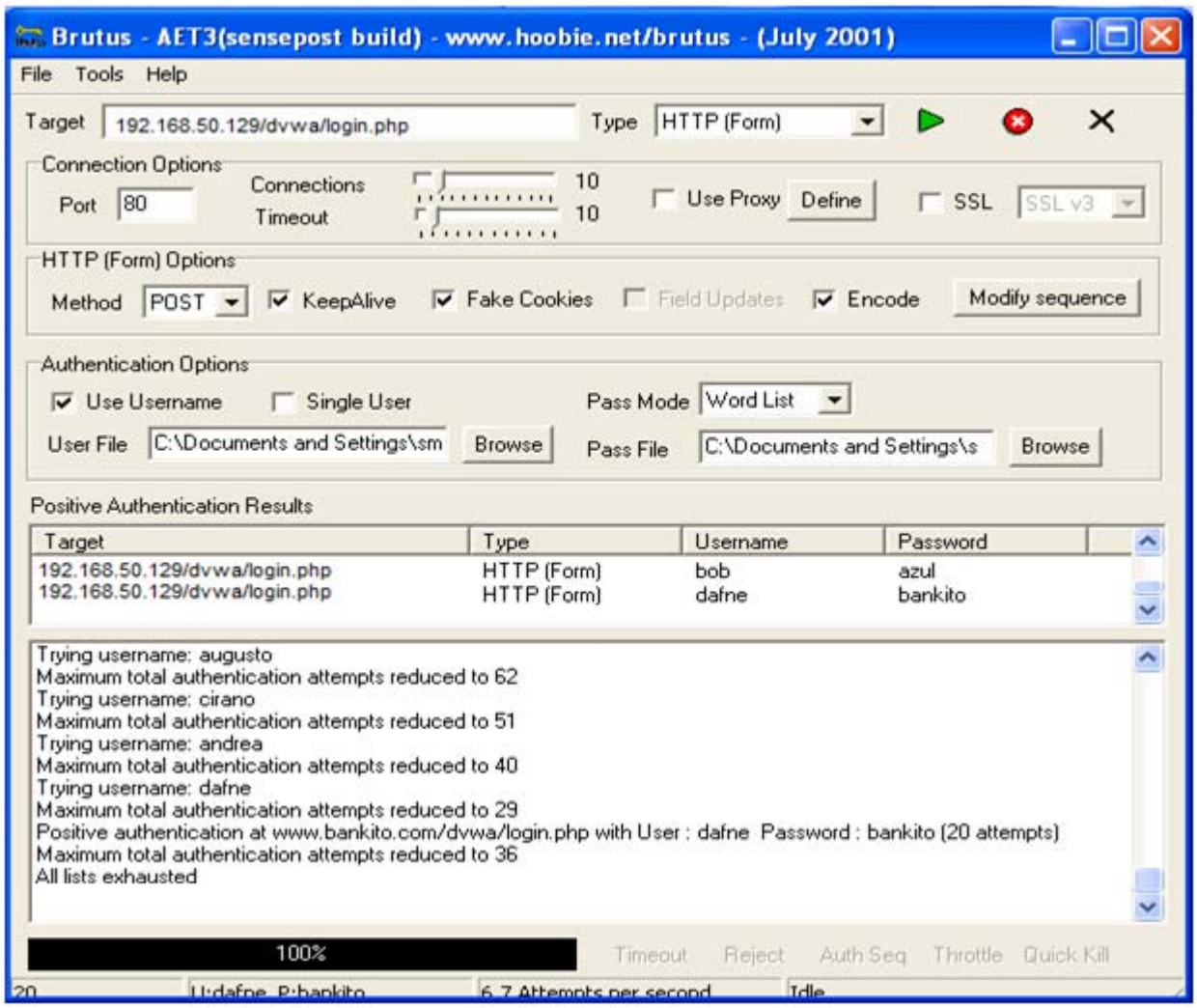

Figura 5.10 Ataque de Diccionario

Como se puede observar en la Figura 5.10 el ataque obtuvo 2 autenticaciones positivas con las siguientes credenciales:

Usuario	Contraseña
Dafne	bankito
bob	azul

Tabla 5.2 Resultados del Ataque de Fuerza Bruta.

Lo que se observa aquí es que tanto los nombres de usuario como sus contraseñas no son palabras complejas. Por tal motivo el ataque fue realizado con éxito mediante diccionarios de palabras comunes. A diferencia del usuario obtenido mediante Wireshark, el cual es un número en vez de una palabra y no se logró obtener por fuerza bruta porque los diccionarios utilizados eran únicamente de palabras.

Ahora se procede a confirmar que realmente sean válidos los accesos antes mencionados. La figura 5.11 muestra el acceso a la aplicación mediante la cuenta bob.

Figura 5.11 Acceso con usuario Bob

Se logró obtener usuarios válidos mediante sniffing de la red y por fuerza bruta y de esta manera entrar a la aplicación.

Ahora que se tiene acceso a la aplicación, se procede a una demostración de ataques de:

- Secuestro de Sesión
- Cross Site Scripting
- Ejecución de comandos
- SQL Injection

5.3.3.1. Ataque de Secuestro de Sesión

Para realizar este ataque es necesario utilizar un PROXY, en este caso Paros, una vez activado el proxy se accede a la aplicación y se atrapa la petición de autenticación hecha con las credenciales de acceso del usuario bob como se observa en la figura 5.12.

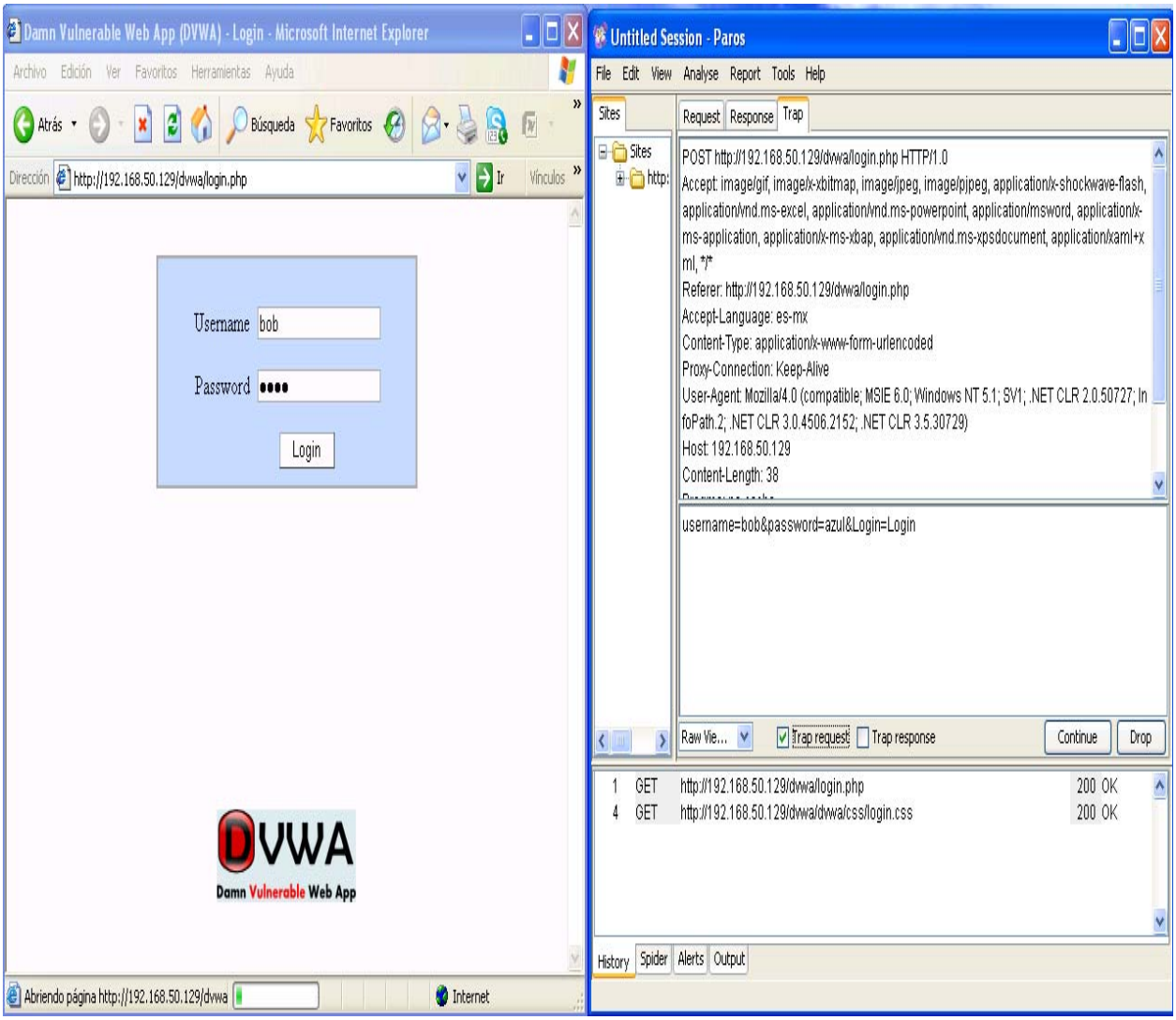

Figura 5.12 Proxy Paros para Atrapar Petición de Autenticación

Analizando la petición que se hace se puede observar que al realizar la autenticación se crea una cookie de sesión (PHPSESSIO) antes de otorgar el acceso (figura 5.13).

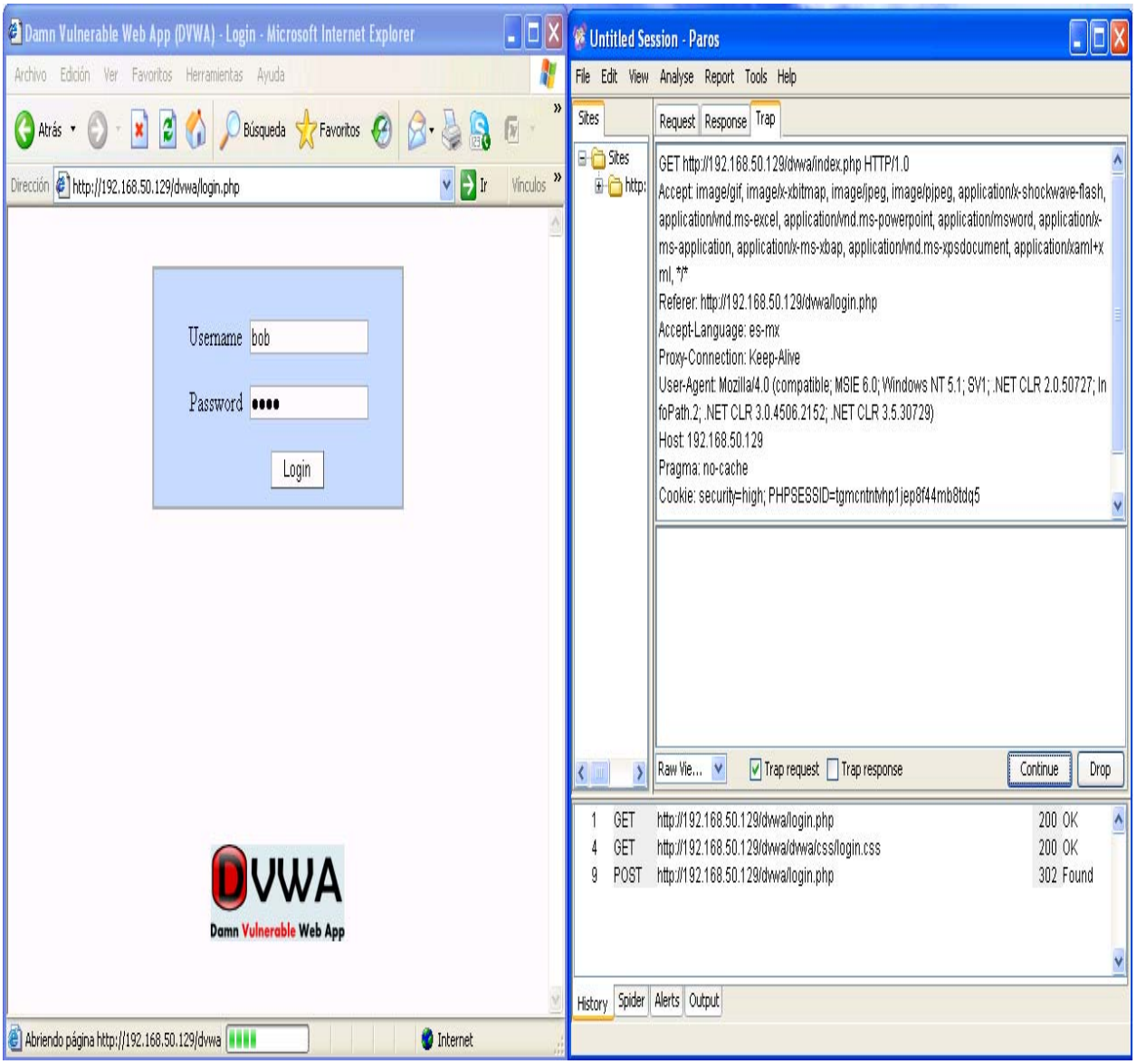

Figura 5.13 Generación de Cookie de Sesión

Toda la información que se genera será utilizada posteriormente en el ataque, por lo tanto es necesario tenerla presente.

Dejando pasar la petición con el proxy se observa cómo el acceso es otorgado (figura 5.14).

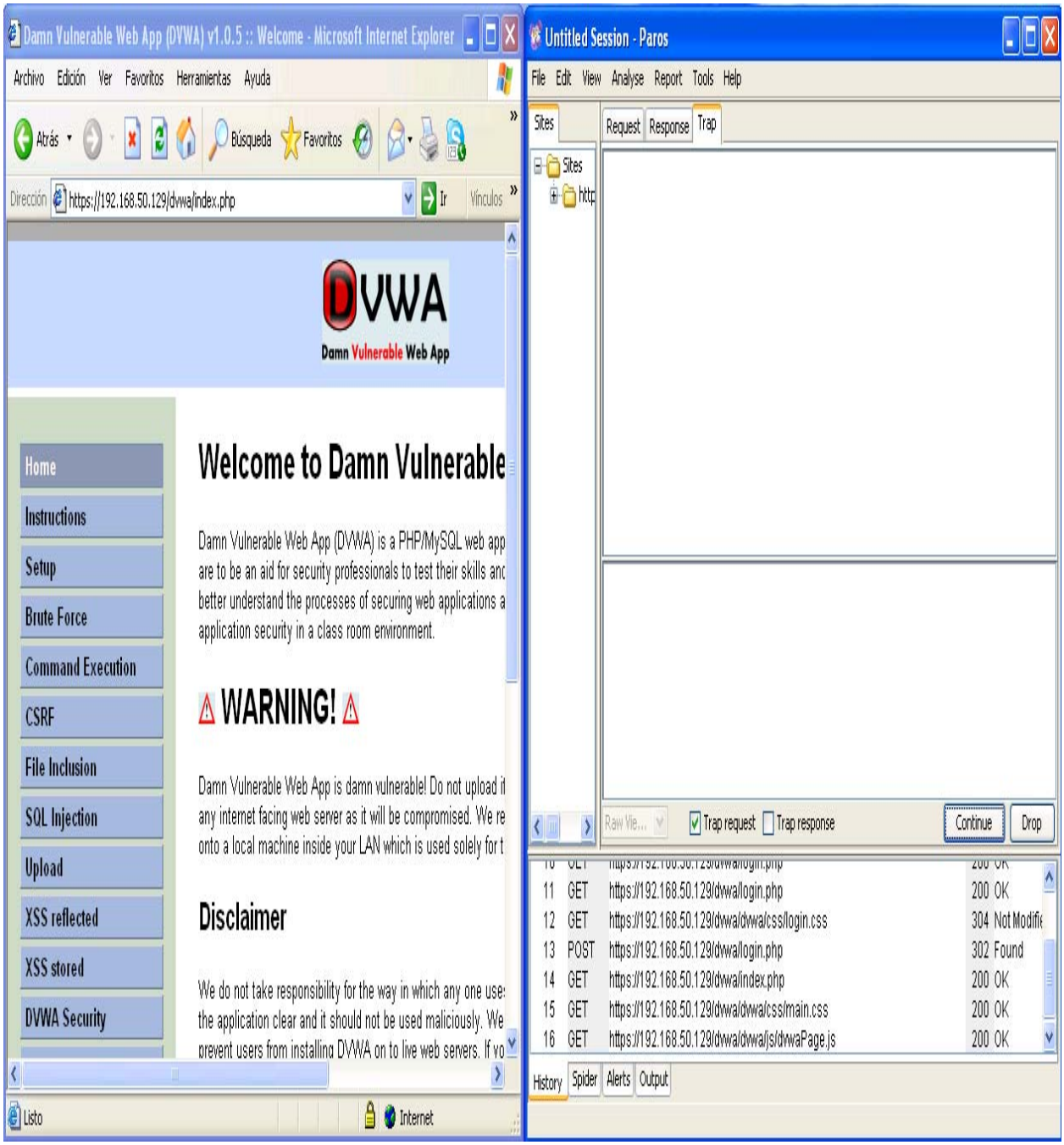

Figura 5.14 Acceso Otorgado

Hasta este punto toda la autenticación fue normal. Si se entra a algún link de la aplicación se puede observar que la sesión sigue siendo la misma como lo muestra la figura 5.15 donde las cabeceras indican que se hizo una petición de acceso a http://192.168.50.129/dvwa/instructions.php.

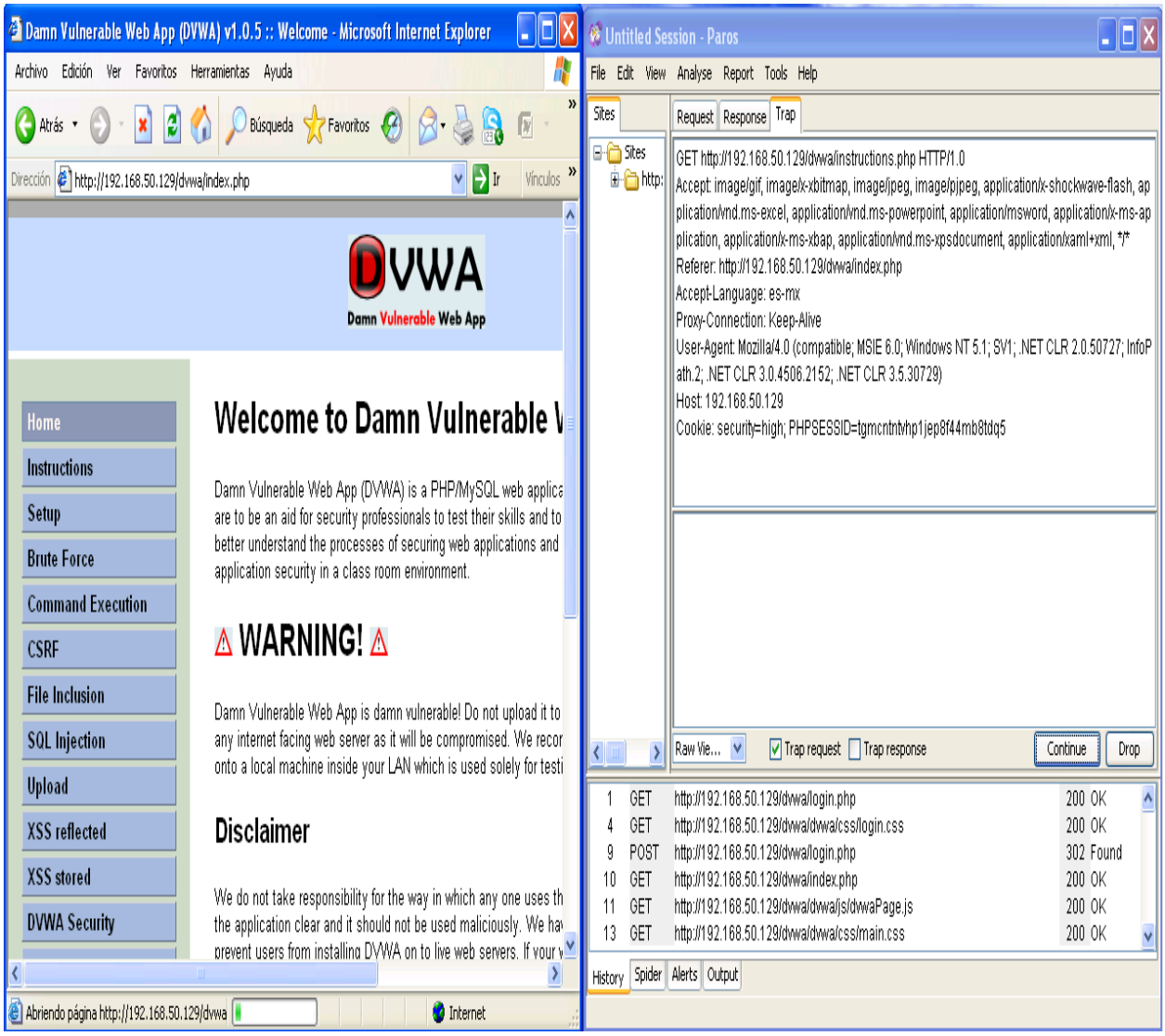

Figura 5.15 Misma Sesión en Diferentes Links de la Aplicación

Utilizando un browser y un proxy distinto, es decir en otra máquina, se puede utilizar la información que se generó cuando la aplicación otorgó el acceso al usuario bob (figura 5.13).

La figura 5.16 muestra la autenticación con credenciales no válidas (usuario = 123, contraseña = 123).

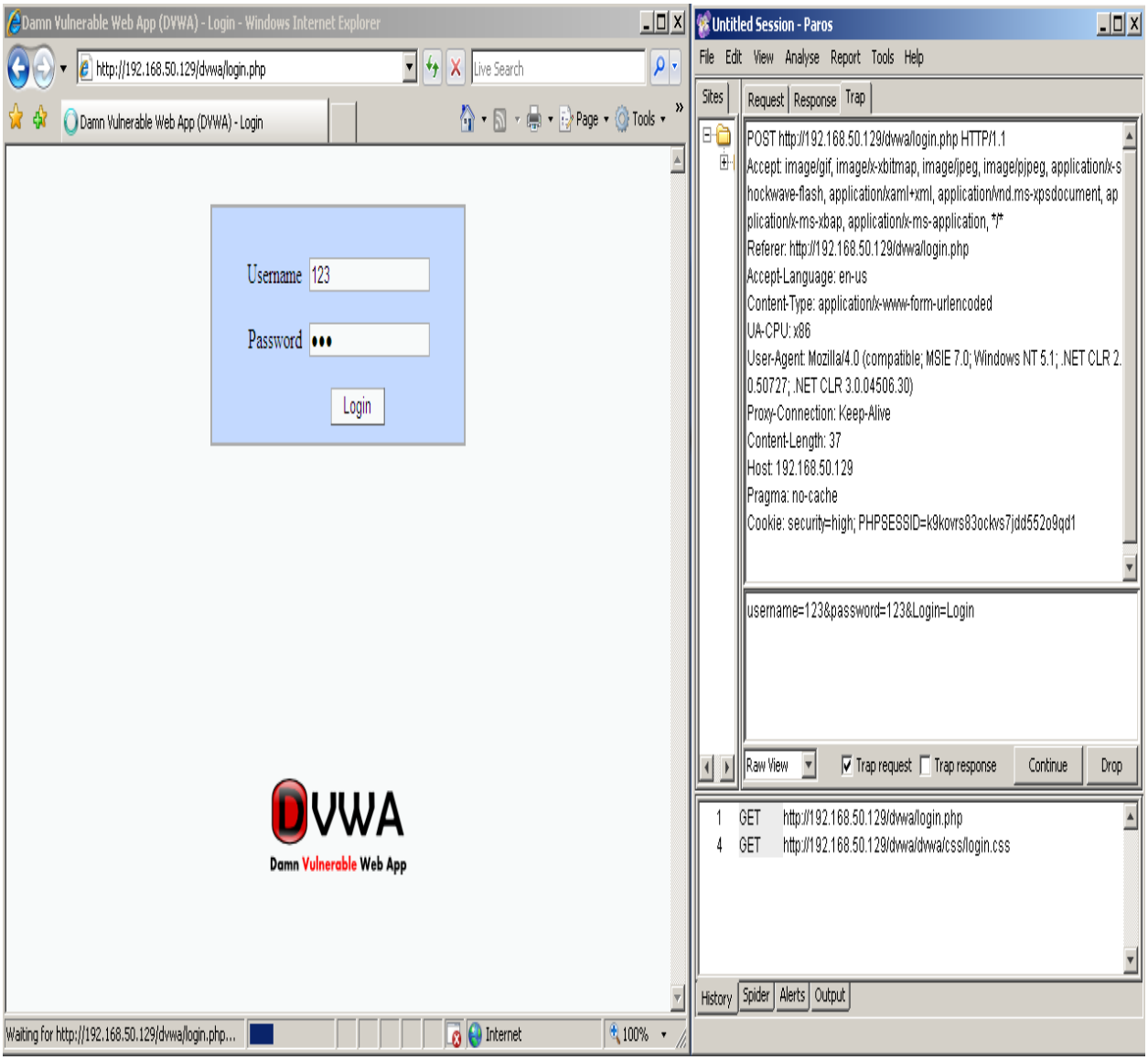

Figura 5.16 Autenticación con Credenciales Inválidas

Como se puede observar la sesión que se genera es distinta a la que se muestra en la figura 5.13. Si colocamos las cabeceras que obtuvimos en la figura 5.13 en vez de las que se generan ahora y eliminamos los datos de usuario, la aplicación aceptará la autenticación como válida ya que la información de sesión que se le envía son de un usuario válido y de este modo se logra acceso a la aplicación, realizando un secuestro de sesión, como se muestra en la figura 5.17. Es importante señalar que el secuestro de sesión funcionará hasta el momento en el que el usuario bob salga de la aplicación.

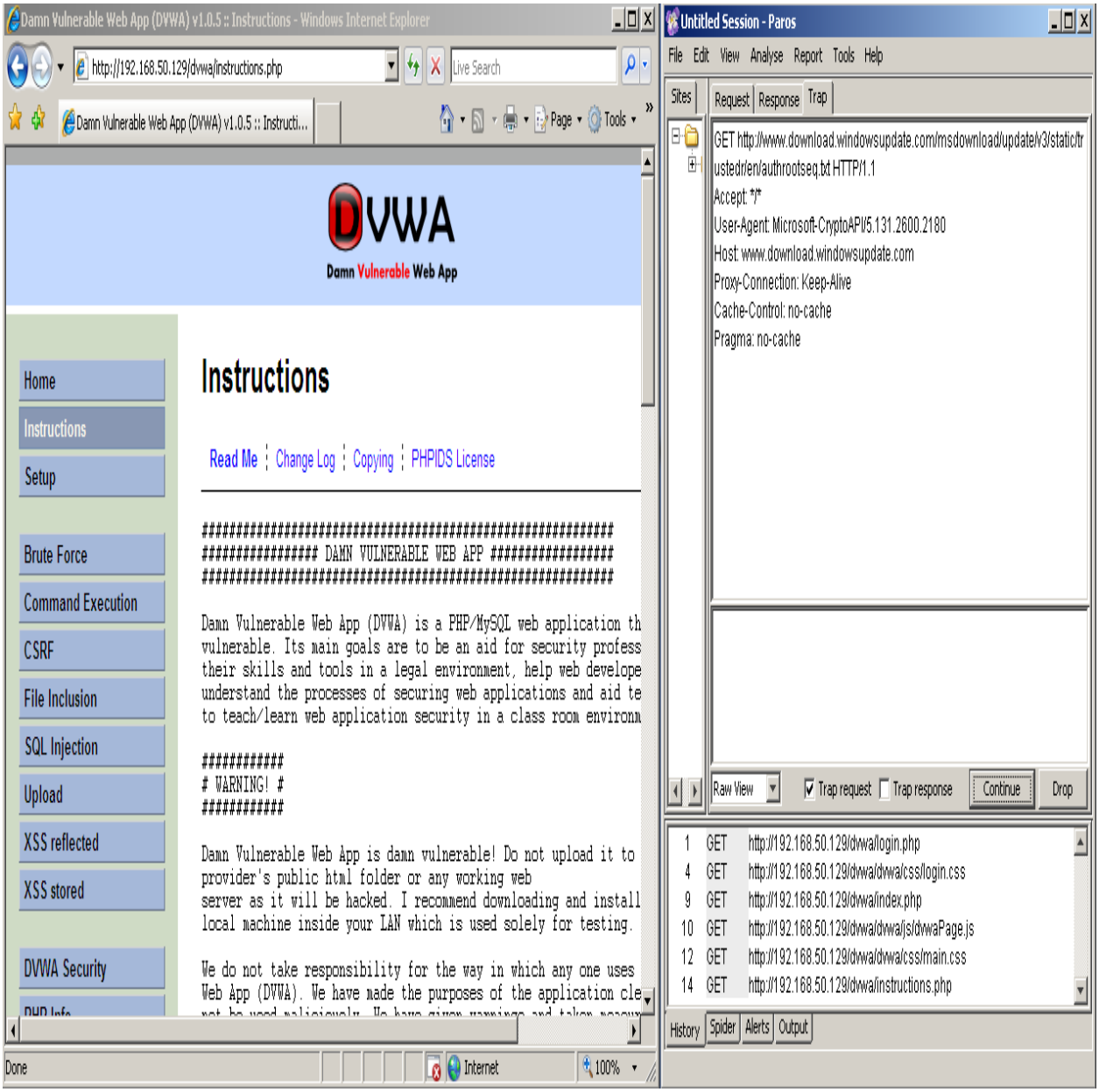

Figura 5.17 Secuestro de Sesión

5.3.3.2. Ataque de Cross Site Scripting (XSS)

El siguiente ataque es de Cross Site Scripting (XSS). Para este caso se utiliza un link de la aplicación donde se tiene una entrada de datos que el usuario otorga (figura 5.18).

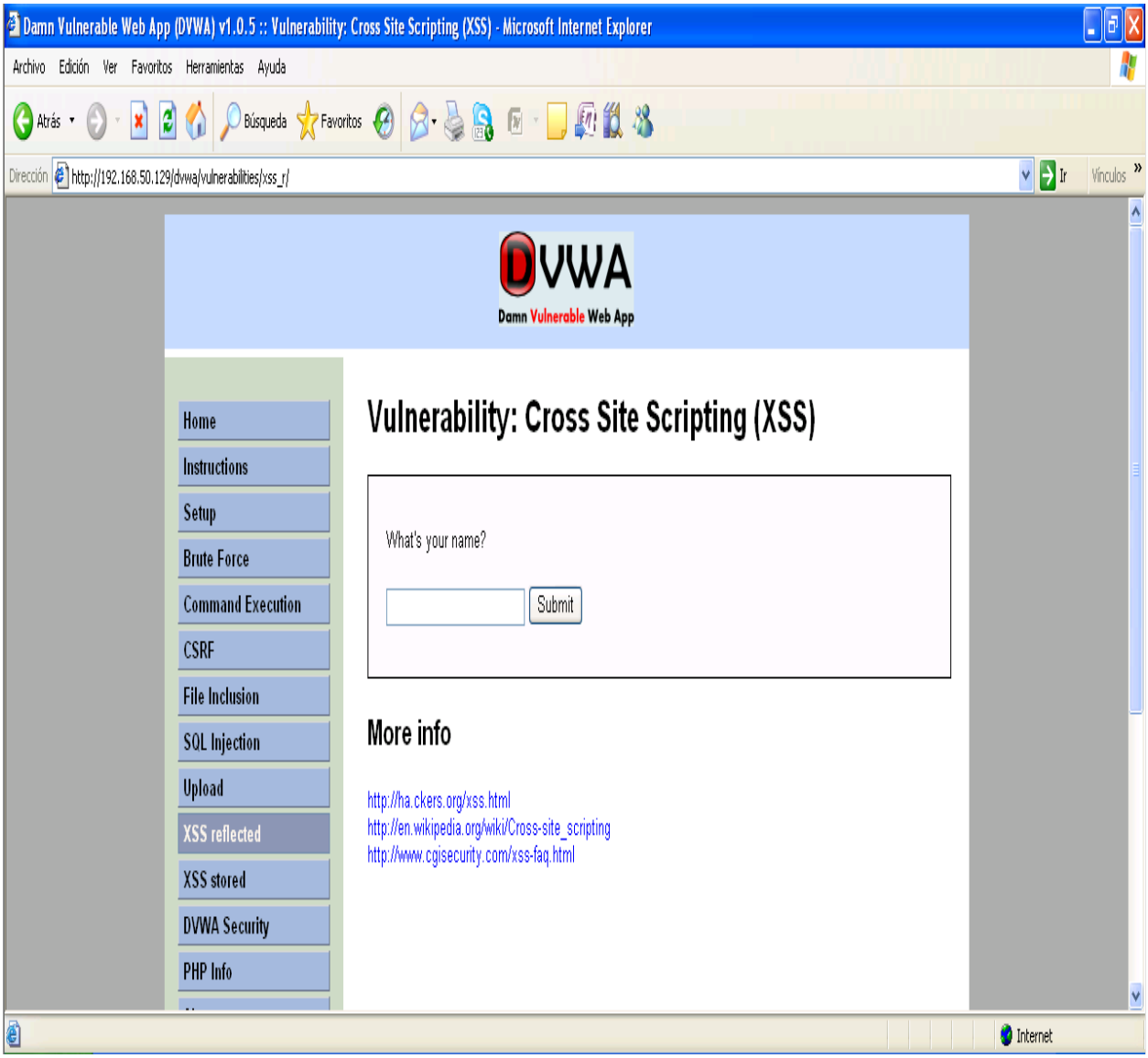

Figura 5.18 Aplicación para XSS

Como se observa en la figura 5.19 el funcionamiento de la aplicación es el siguiente: El usuario coloca un dato (en este caso su nombre) y la aplicación lo imprime en pantalla junto con la palabra Hello.

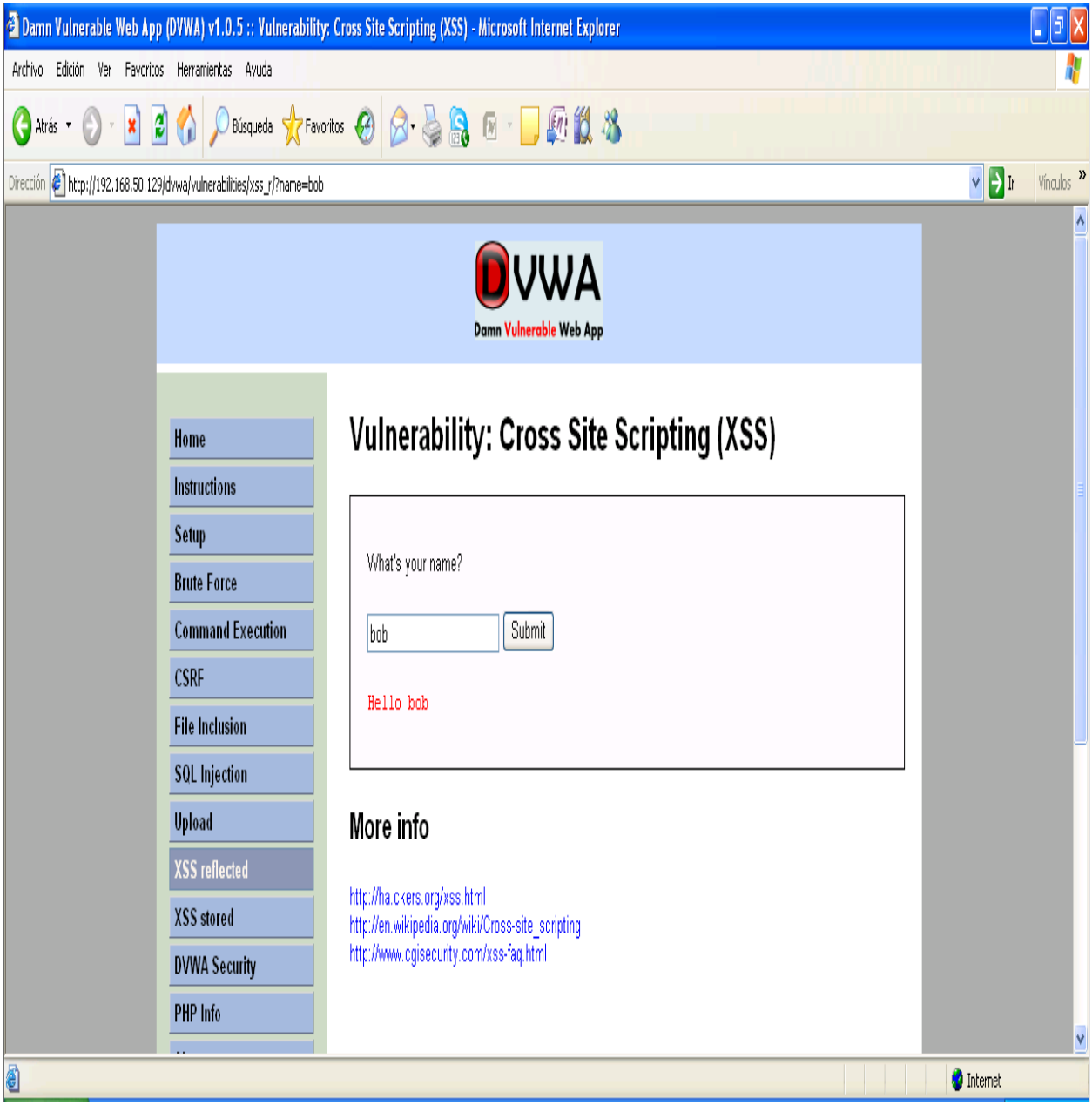

Figura 5.19 Funcionamiento de Aplicación para XSS

Como se observa en la figura 5.19 una vez que la aplicación ha sido ejecutada, la URL muestra la variable que utilizó en su código para pedir los datos (name) junto con el texto que el usuario colocó. Esta variable se puede utilizar para inyectar código JavaScript y con esto conseguir XSS.

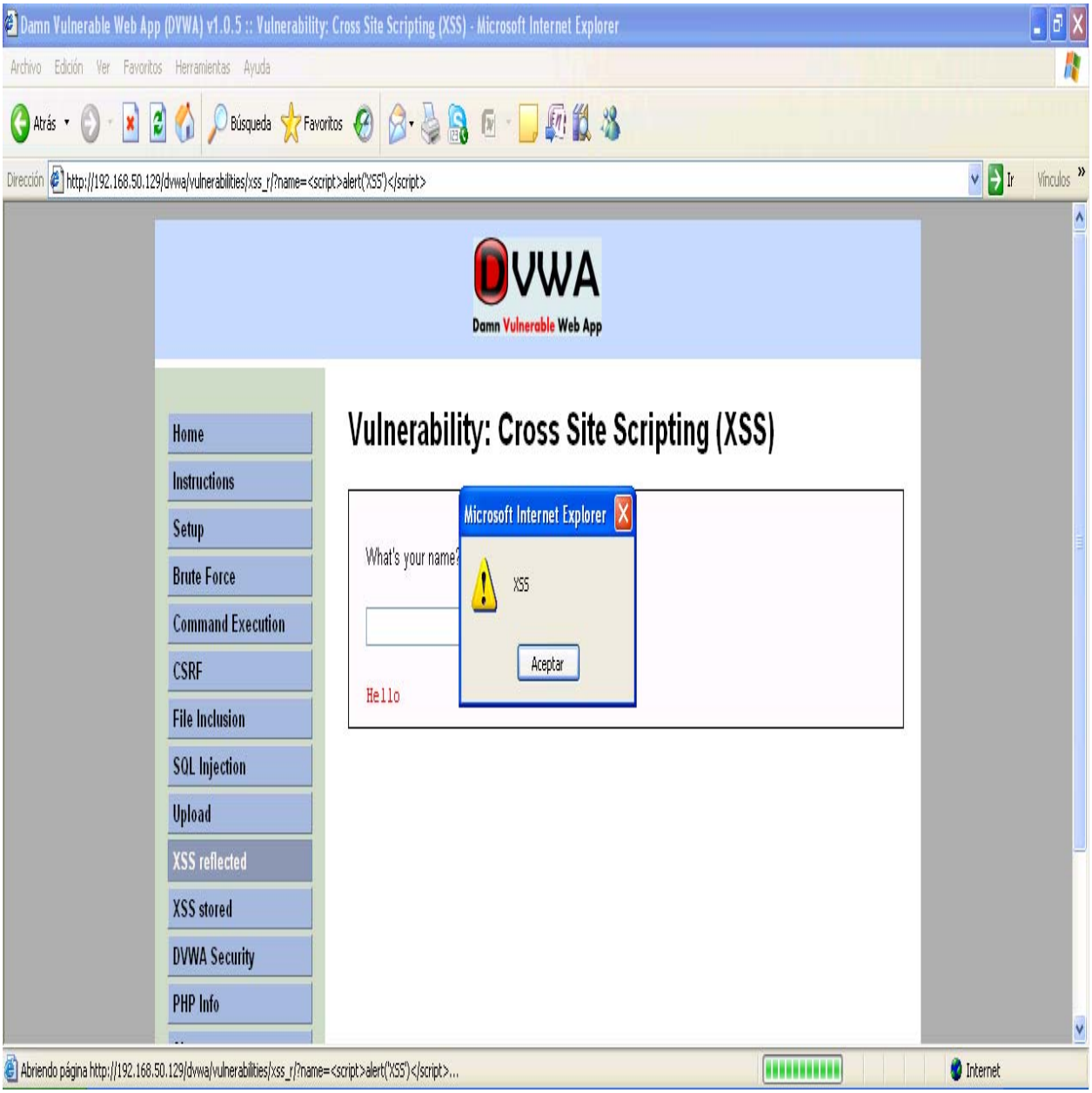

Figura 5.20 Ejecución de XSS

Como se puede observar en la figura 5.20, la información que la variable nombre tenía (bob) fue cambiada por código JavaScript (<script>alert (‘XSS´) </script>), la aplicación no validó los datos de entrada y acepto el código JavaScript que en este caso muestra un “pop-up” con las siglas XSS.

La siguiente demostración se refiere a ejecución de comandos.

5.3.3.3. Ataque de Ejecución de Comandos

Muchos scripts de PHP son vulnerables a este tipo de amenazas. Para este ejemplo se utiliza otro link de la aplicación. En la figura 5.21 se puede observar que la aplicación pide una IP para realizar un PING.

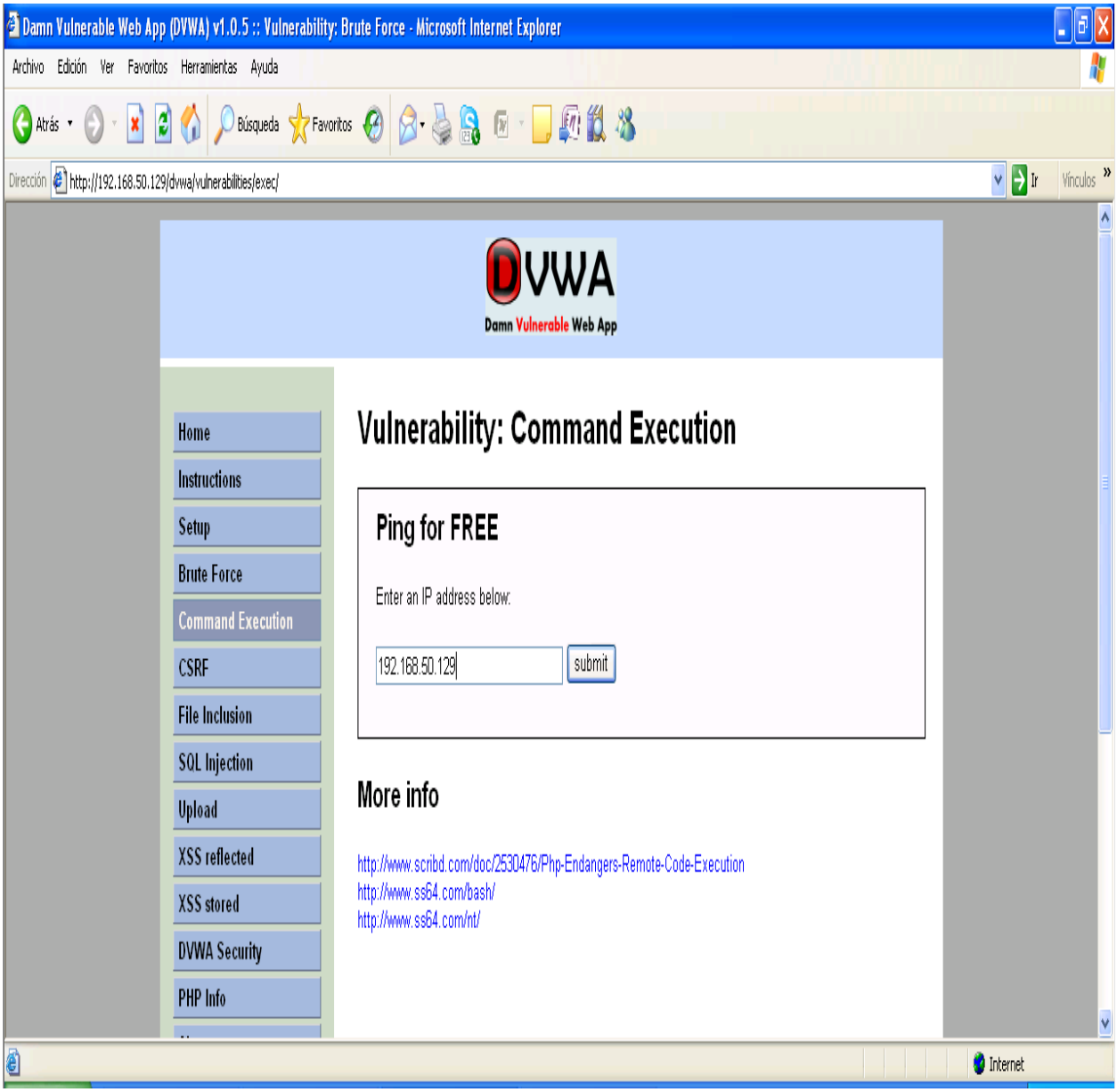

Figura 5.21 Ejecución de Comandos

La manera en que esta aplicación trabaja esta descrita en la figura 5.22. Se puede observar que manda llamar una función (Shell_exec) la cual ejecuta sobre línea de comandos la sentencia “ping –c 3” y le agrega el valor que tiene la variable “$target” que es la que obtiene la IP que el usuario otorga. Por lo tanto la sentencia final quedará de esta forma “ping –c 3 x.x.x.x (IP otorgada).

Figura 5.22 Código para Ejecución de Comando PING

En la figura anterior se observa que no existe una restricción en cuanto a los datos que la variable $target recibe. Por lo tanto es posible añadir datos extras, como por ejemplo, “&& DIR” como se observa en la figura 5.23, esto hará que la sentencia final ejecutada por la función Shell_exec sea la siguiente “ping –c 3 x.x.x.x && DIR”.

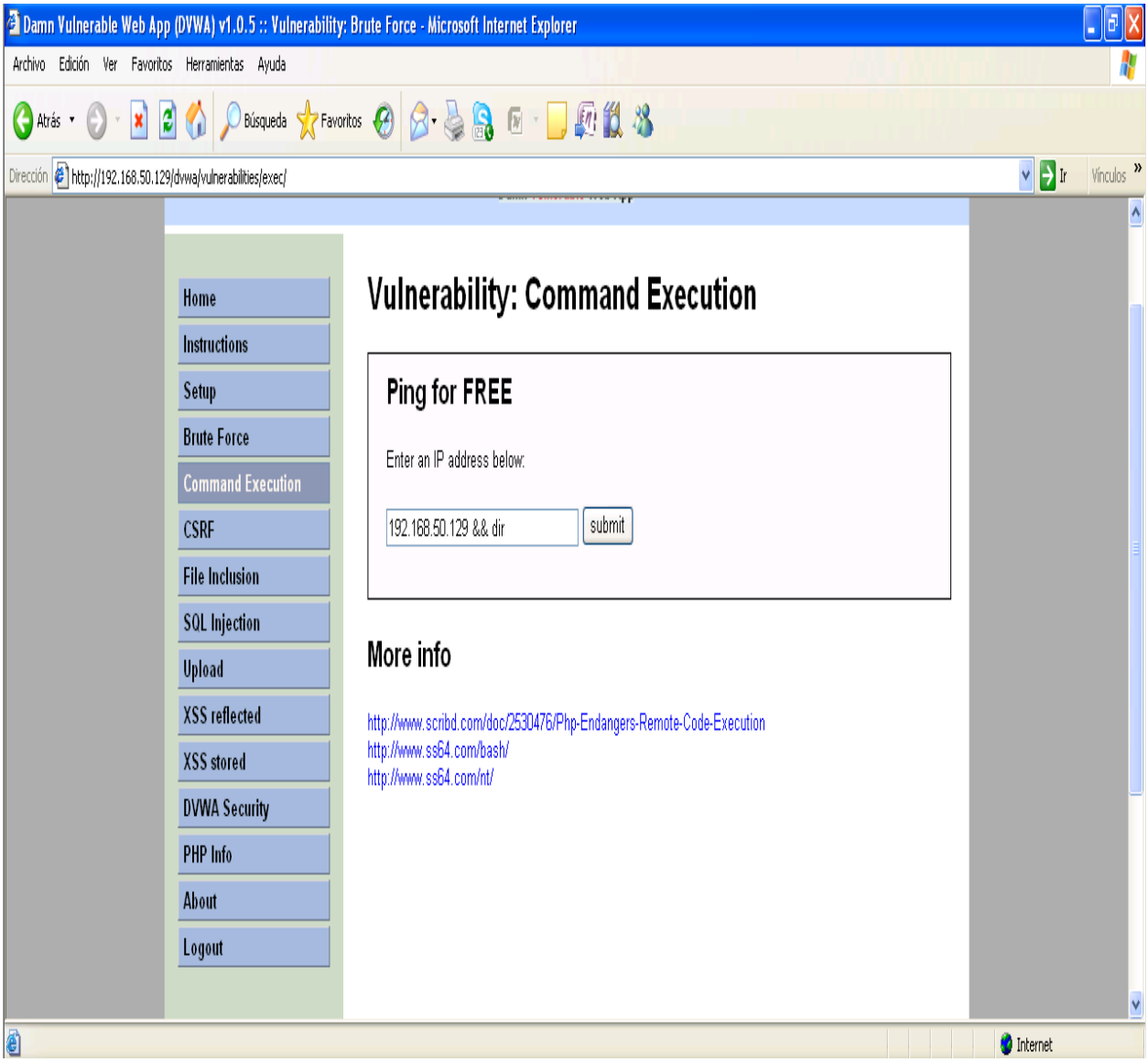

Figura 5.23 Datos Inyectados

Lo anterior hace que se realice un ping a la IP otorgada y además se hará un DIR al directorio donde se encuentra aplicación como se observa en la figura 5.24.

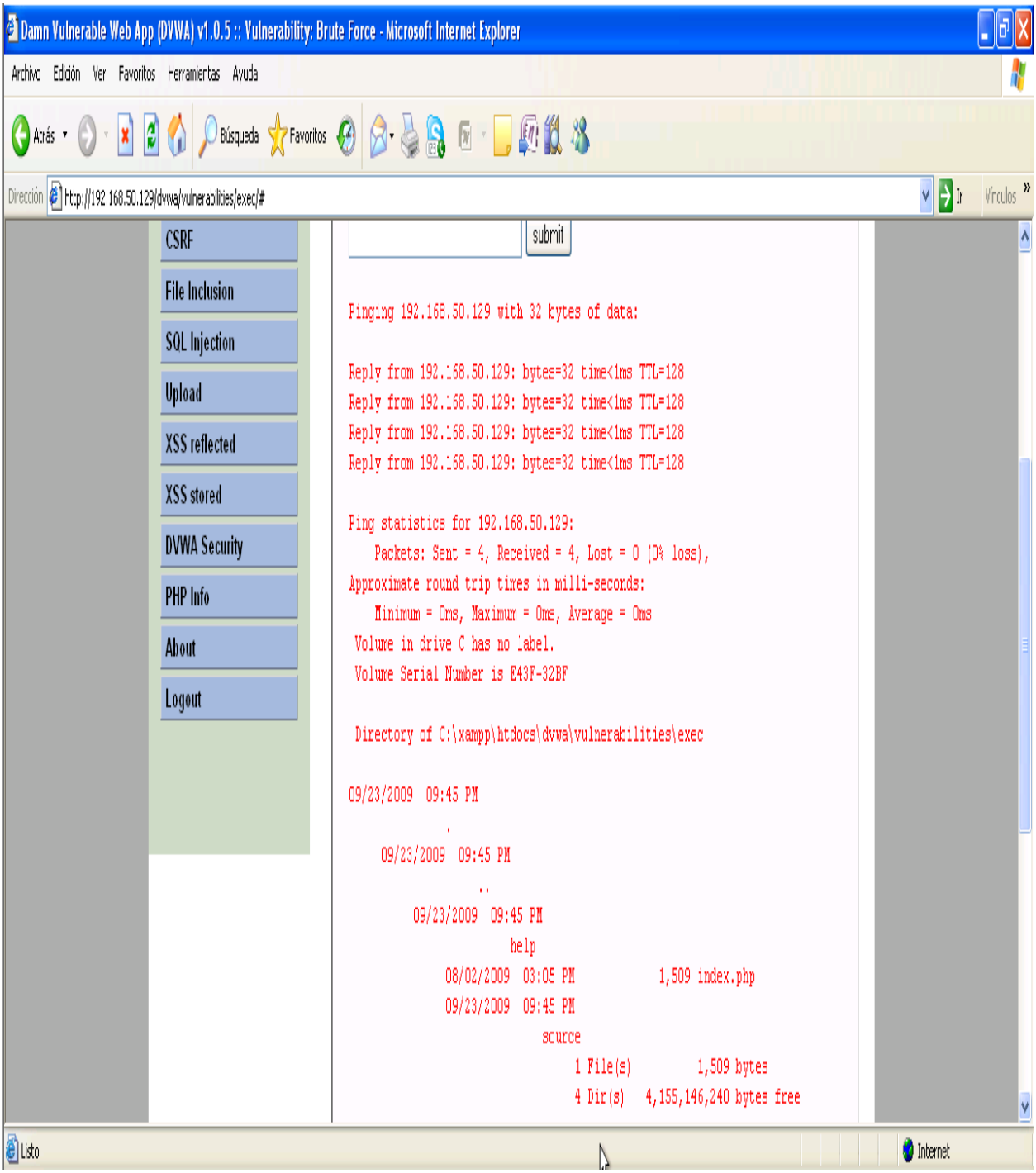

Figura 5.24 Resultado de la Inyección de Datos

Por último se realiza una demostración de un ataque de SQL Injection en el cual el objetivo es obtener los hashes de las contraseñas de los usuarios.

5.3.3.4. *Ataque de SQL Injection*

La figura 5.25 muestra el funcionamiento de la aplicación que se utiliza para este ejemplo. Como se puede observar la aplicación pide el ID de usuario y regresa en pantalla el mismo ID, el nombre y el apellido del usuario.

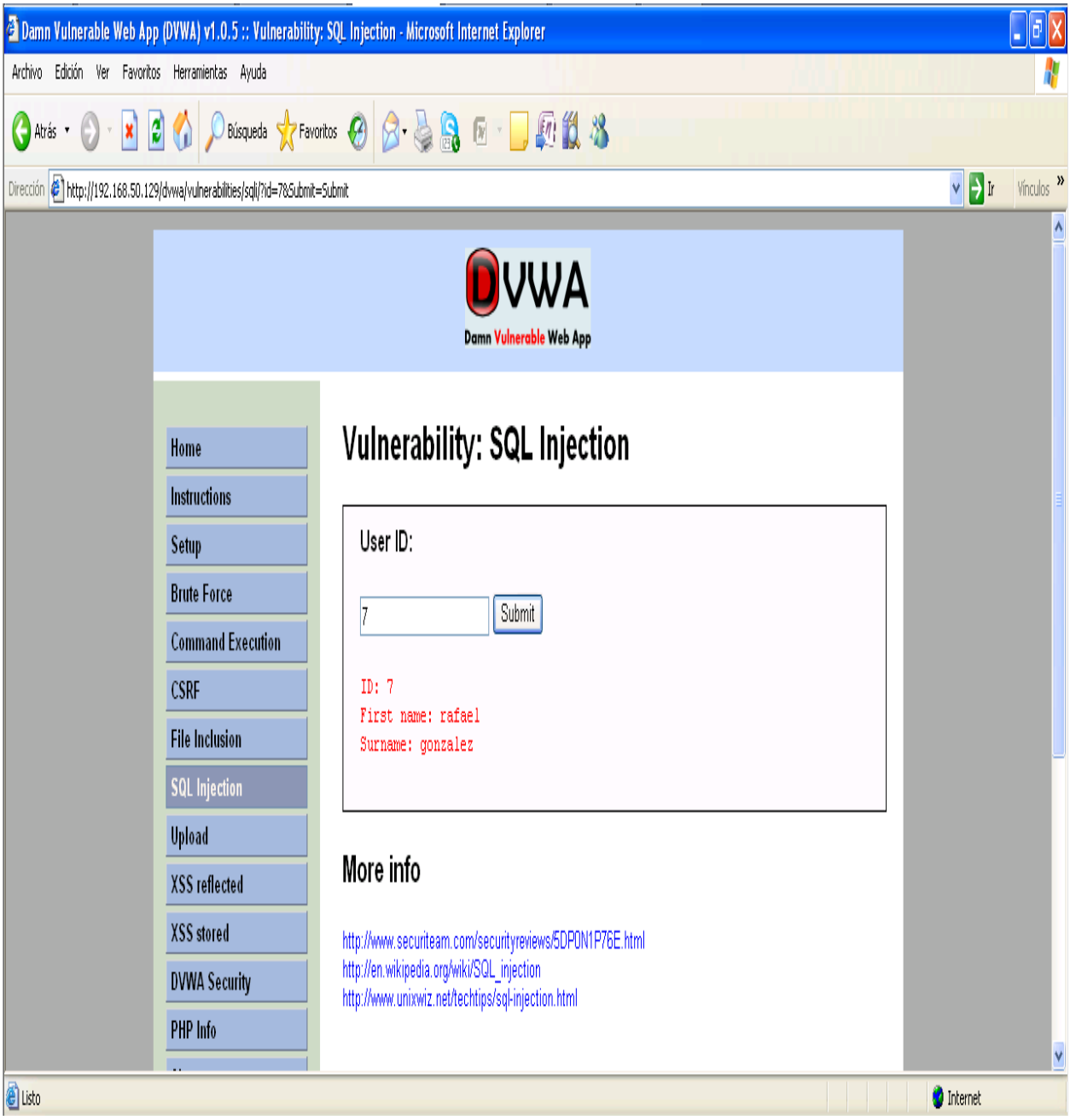

Figura 5.25 Funcionamiento de Aplicación para SQL Injection

EL primer paso es verificar que la aplicación valide los datos de entrada que el usuario otorga. Se coloca una comilla como parámetro de entrada como se observa en la figura 5.26.

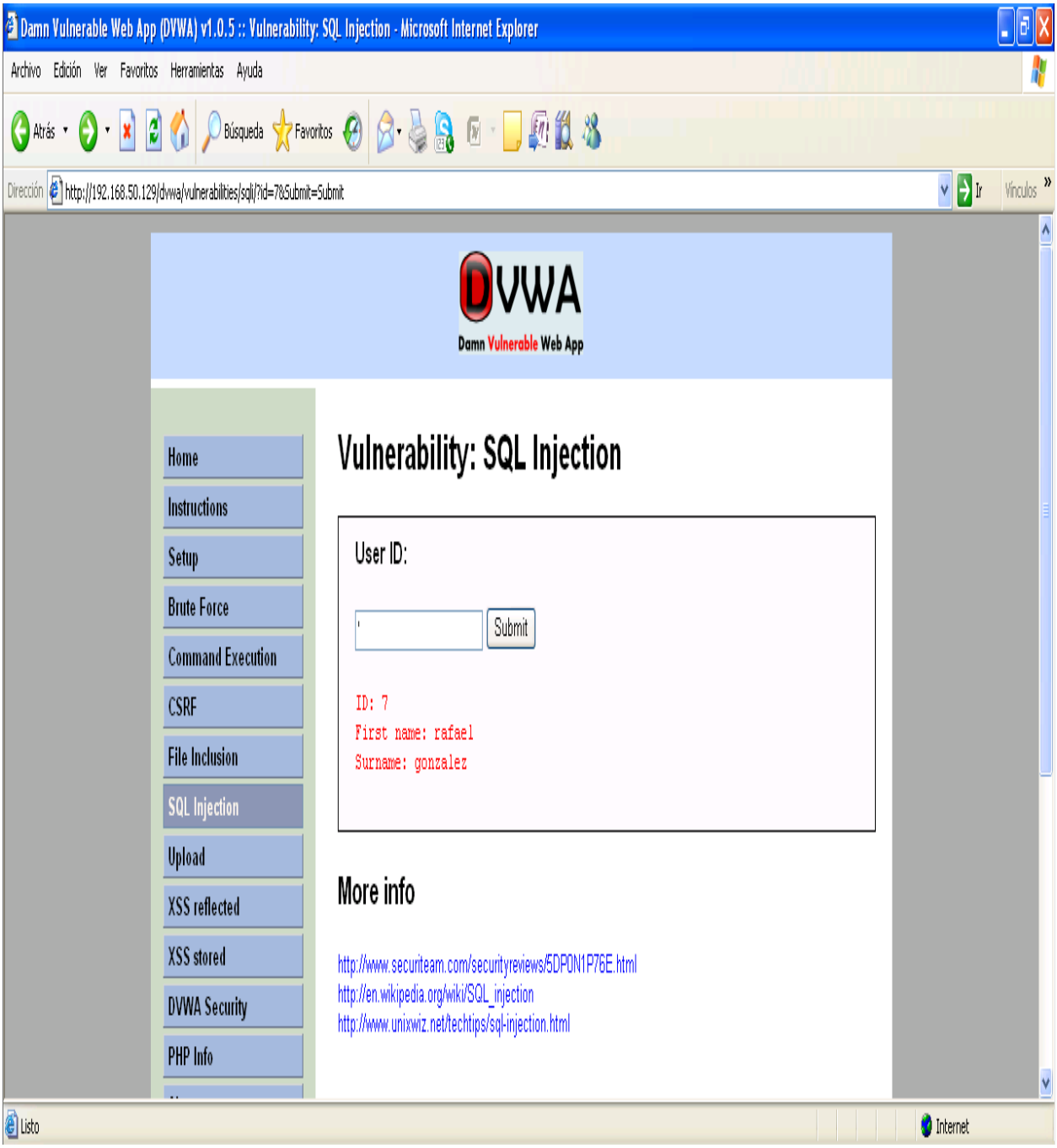

Figura 5.26 Inyección de Caracteres Especiales

Como se observa en la figura 5.27 la aplicación no valida el dato y lo acepta, sin embargo, manda un mensaje advirtiendo que existe un error en la sintaxis de la sentencia de SQL que se utiliza para realizar el query que nos devuelve la información solicitada. Este tipo de errores suele ocurrir cuando no se cierran de manera adecuada las sentencias, lo que puede ocurrir en este caso, es que el dato que el usuario otorga es tomado por la aplicación y asignado a una variable dentro de la query que se utiliza.

Por ejemplo:

SELECT nombre, apellido FROM usuarios where ID='7';

Donde 7 es el valor otorgado por el usuario y se encuentra dentro de comillas por ser un valor numérico, por lo tanto al dar una comilla como parámetro la sentencia puede quedar de la siguiente forma:

SELECT nombre, apellido FROM usuarios where ID='';

Provocando que se genere el error de sintaxis debido a la falta de la comilla que cierre la primera.

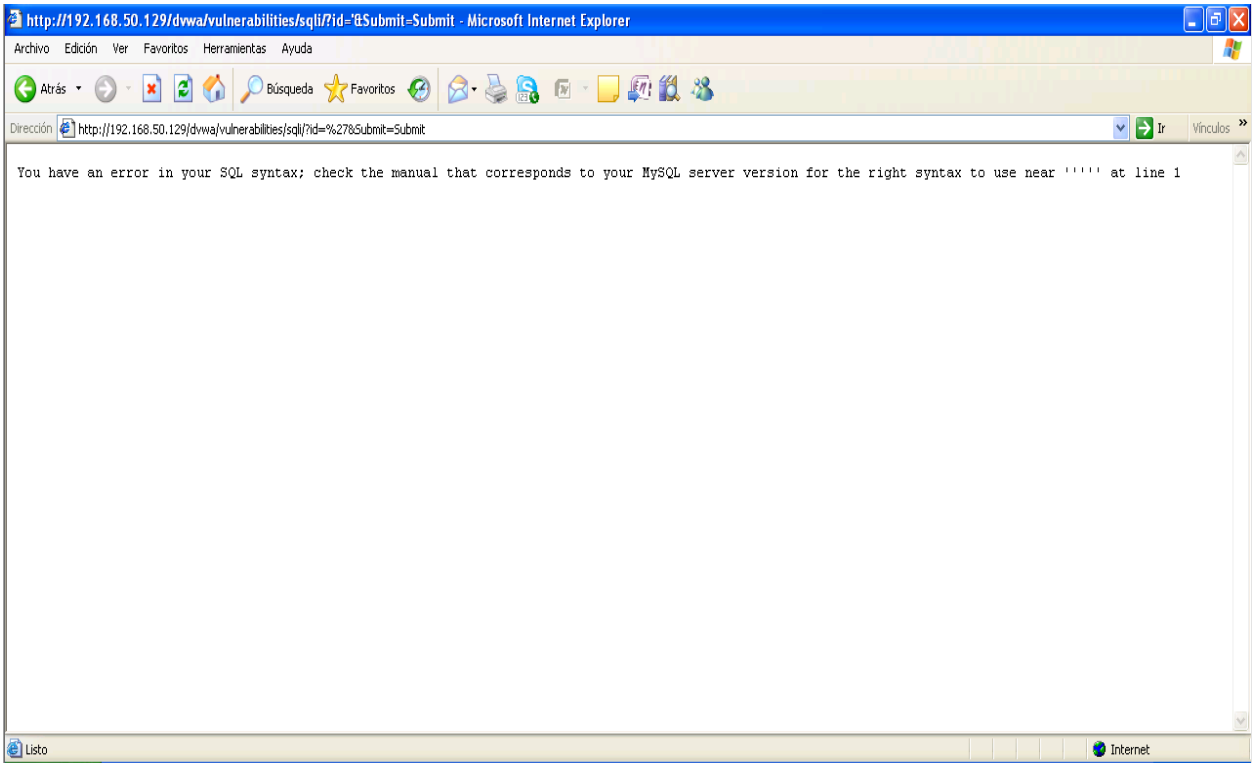

Figura 5.27 La Aplicación no Valida los Datos

Para explotar esta falla en la validación de los datos, se ingresa los siguientes datos (figura 5.28):

' or '1'='1

Siguiendo el ejemplo anterior la sentencia queda de la siguiente forma:

*SELECT nombre, apellido FROM usuarios WHERE ID= '' **or '1'='1** ';*

Como se puede observar la sentencia cierra la última comilla colocada por lo tanto no existirá error en la sintaxis. En este caso la sentencia pide el nombre y el apellido de la

tabla usuarios donde el ID sea igual a ningún valor o 1 igual 1, puesto que 1=1 siempre será válido, en teoría la información devuelta deben ser todos los usuarios. En la figura 5.29 se observa la información devuelta al hacer la inserción.

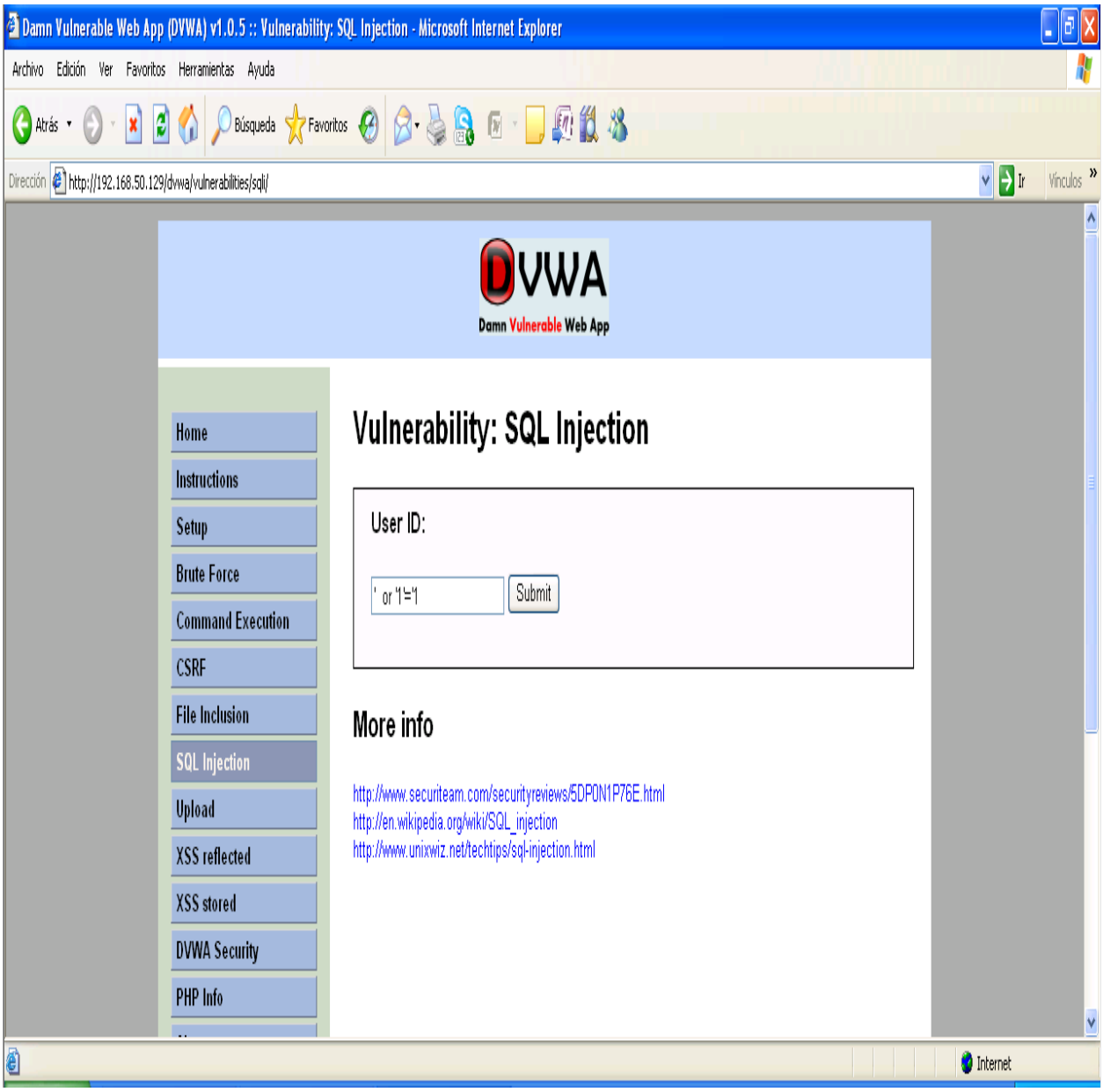

Figura 5.28 Inyección de SQL

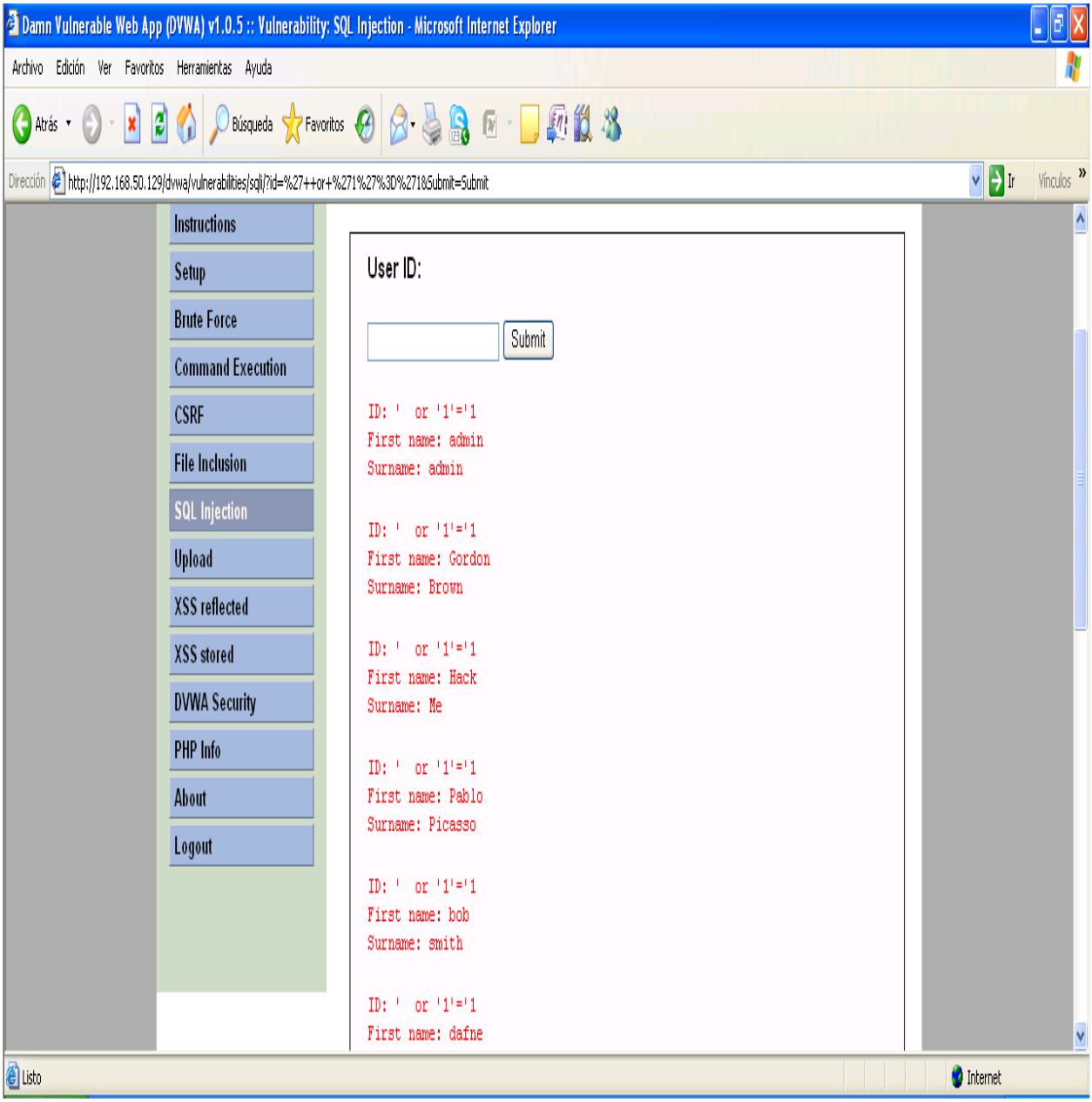

Figura 5.29 Obtención de Usuarios

Tal como se mencionó la aplicación tomó como válido 1=1 y devolvió el nombre y apellido de todos los usuarios. Entonces se sabe que mientras la sintaxis de la sentencia de SQL quede sin errores la aplicación aceptará cualquier parámetro que se le otorgue.

Para que lo anterior sea posible a los datos que se otorguen debe terminar con una variable y la omisión de la última comilla figura (5.30), por ejemplo:

‘ or ‘1’=’1’ UNION SELECT password FROM password WHERE password!=’1

Donde además de los datos utilizados anteriormente se utiliza anexa otra sentencia mediante la función UNION, en donde intencionalmente se pide la contraseña de la tabla password donde la contraseña sea diferente a 1. El resultado se observa en la figura 5.31.

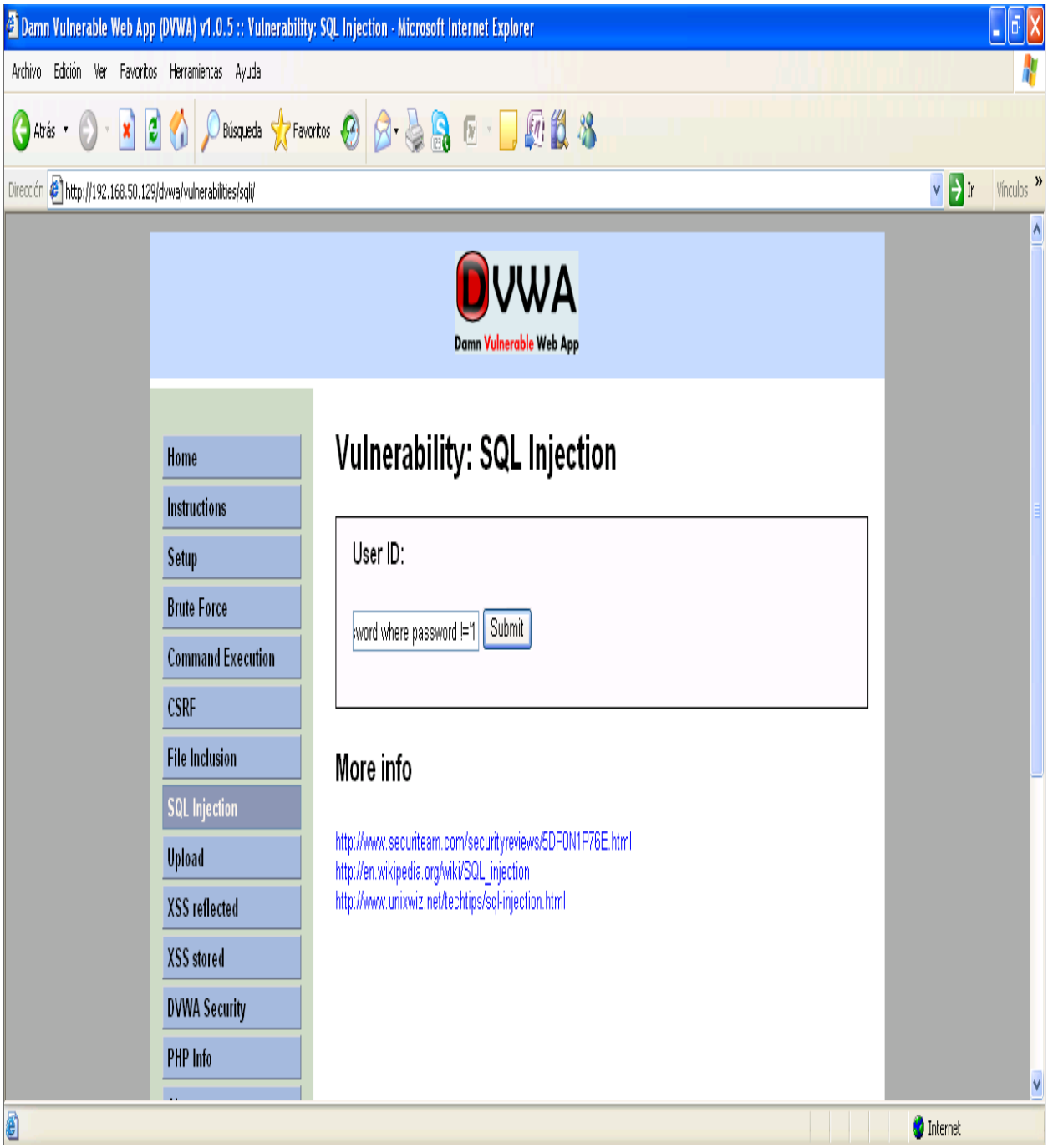

Figura 5.30 Segunda inyección de SQL

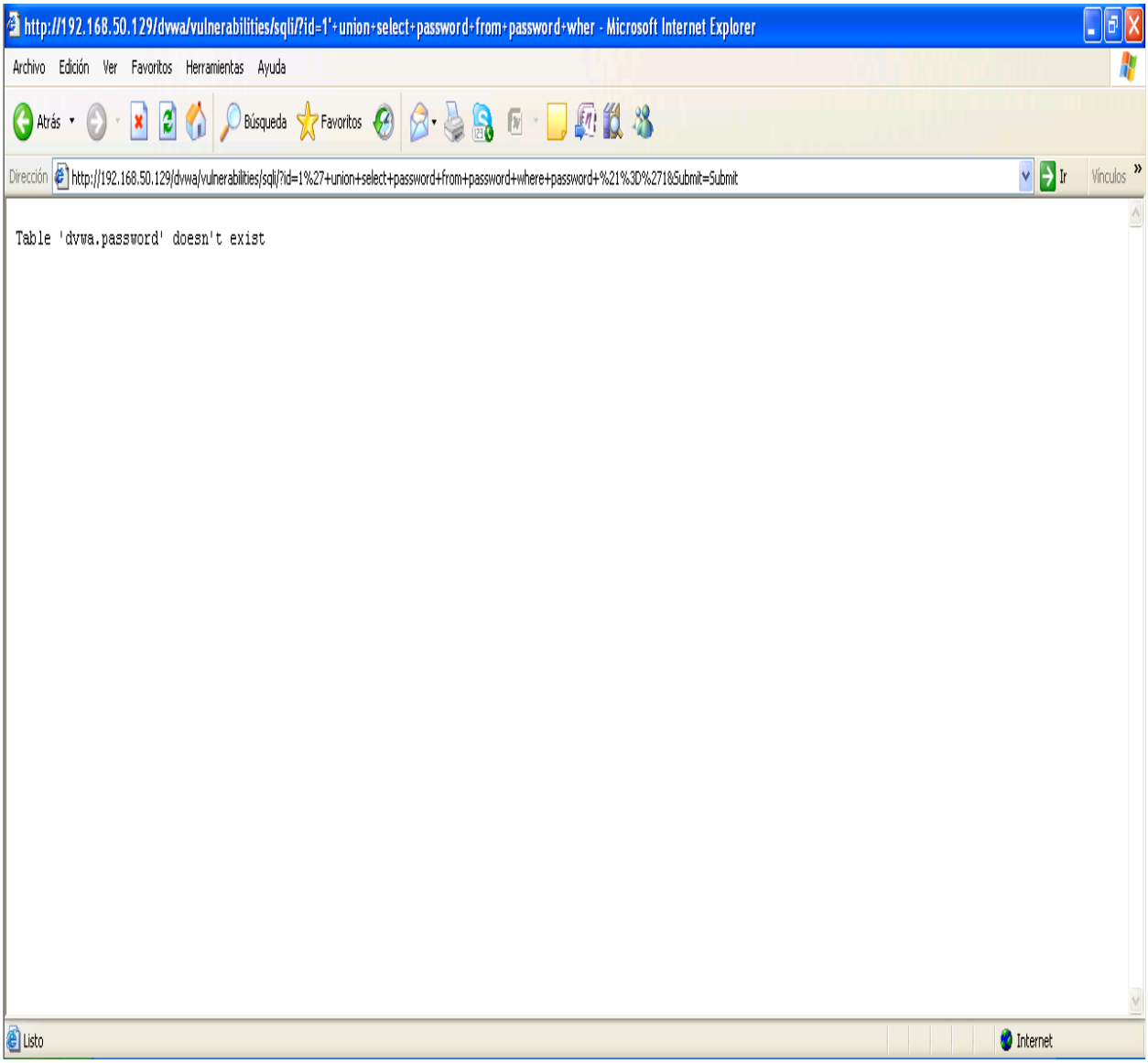

Figura 5.31 Obtención de Nombre de la Tabla

Como se observa en la figura 5.31 la sentencia no tuvo errores de sintaxis, sin embargo, devuelve un error donde indica que la tabla password no existe en la base de datos dvwa. Por lo tanto se ha conseguido el nombre de la base de datos (dvwa). Lo siguiente sería un proceso de fuerza bruta para obtener el nombre de la tabla y el nombre de las columnas. Asumiendo que se realiza este proceso y se llega a la instancia en que los siguientes datos son aceptados:

‘ or ‘1’=’1’ UNION SELECT password FROM users WHERE password!=’1

Donde se concluye que el nombre de la tabla es users y que la columna password existe, sin embargo, el error que ocurre ahora es referente a la función UNION,

advirtiendo que se está utilizando un número diferente de columnas para realizar la función (figura 5.32).

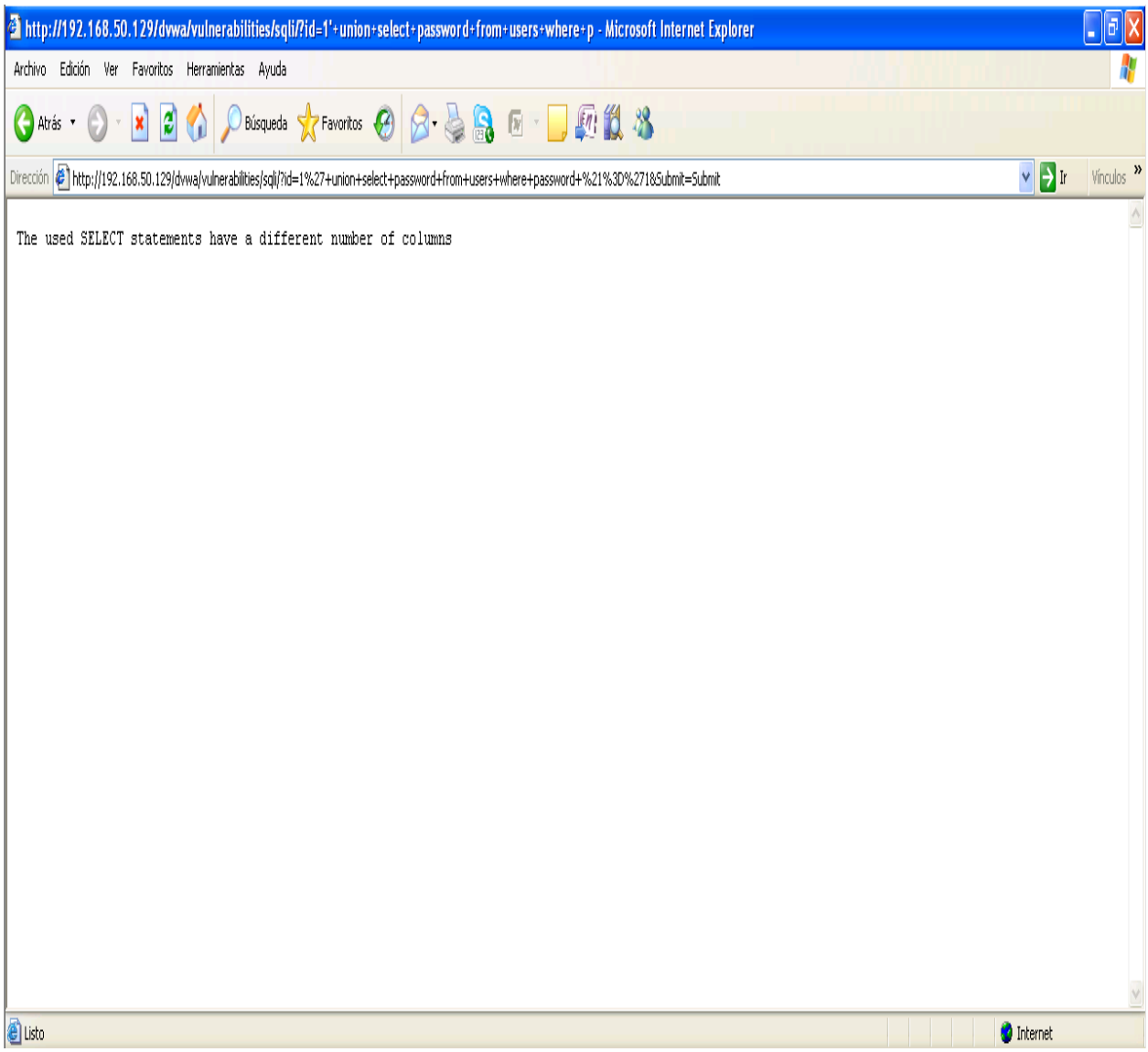

Figura 5.32 Obtención de Query Utilizado

En la sentencia que se utiliza únicamente se pide una columna, por lo tanto el error nos indica que hacen falta más. Entonces se utiliza lo siguiente:

‘ or ‘1’=’1’ UNION SELECT password, 1 FROM users WHERE password!=’1

Donde 1 se utiliza para hacer referencia a una columna dentro de la tabla que aunque no exista sirve para que la función no marque error.

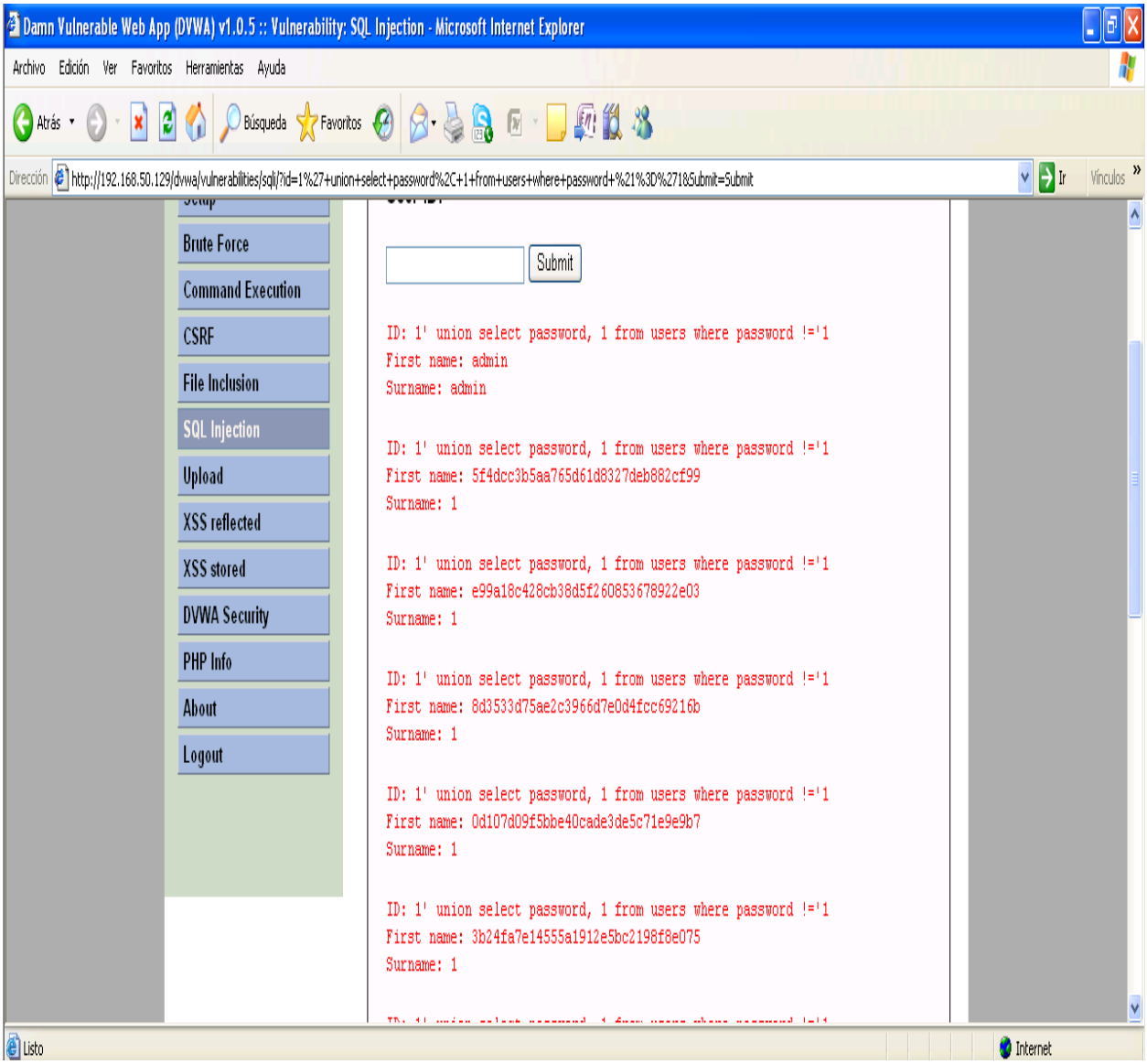

Figura 5.33 Obtención de Hashes

Como se puede observar en la figura 5.33 la función tomó los valores como válidos y devolvió el contenido de la columna password, donde los datos se encuentran cifrados. Este no es problema porque sólo es cuestión de tiempo para descifrarlos sin embargo de poco nos sirven si no se tienen los nombres de los usuarios. Como ya se sabe el nombre de la tabla, que la columna password existe y que se necesitan 2 columnas para que la función UNION funcione, se procede a realizar fuerza bruta sobre la columna faltante con el fin de encontrar en nombre de la columna que contiene los nombres de los usuarios. Omitiendo el proceso de fuerza bruta y llegando a la instancia en que se prueba 'user' como nombre de la columna, los datos quedarían de esta forma:

' or '1'='1' UNION SELECT user, password FROM users WHERE password!='1

El resultado obtenido se muestra en la figura 5.34 donde se observa que la aplicación devuelve el nombre del usuario y contraseña (cifrado). Con esto, lo único que falta es romper los hashes de la contraseña. Para esto pueden utilizarse diferentes herramientas de crackeo de contraseñas o incluso aplicaciones que se encuentran en internet (figura 5.35).

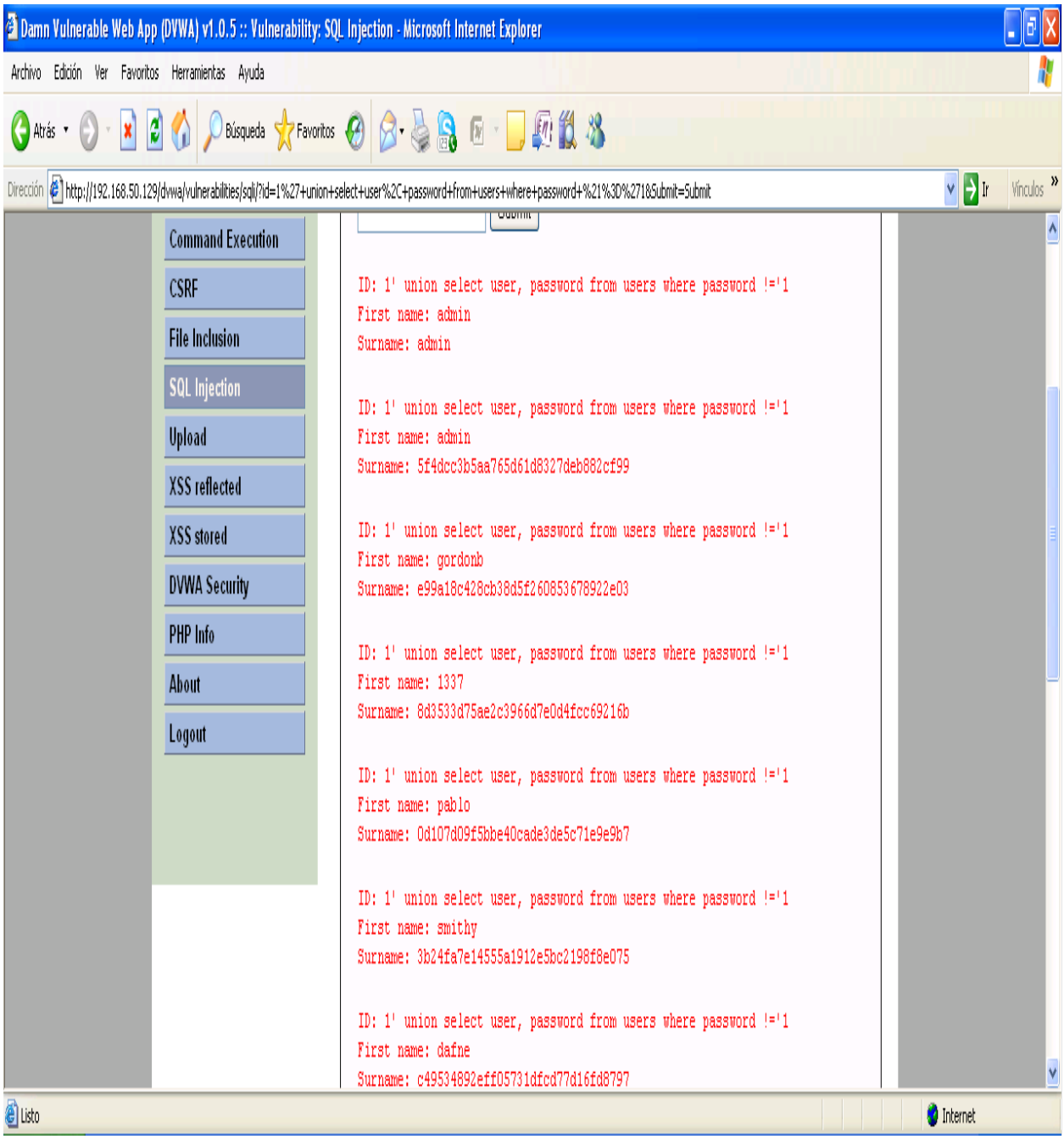

Figura 5.34 Obtención de Usuarios y sus Respectivos Hashes

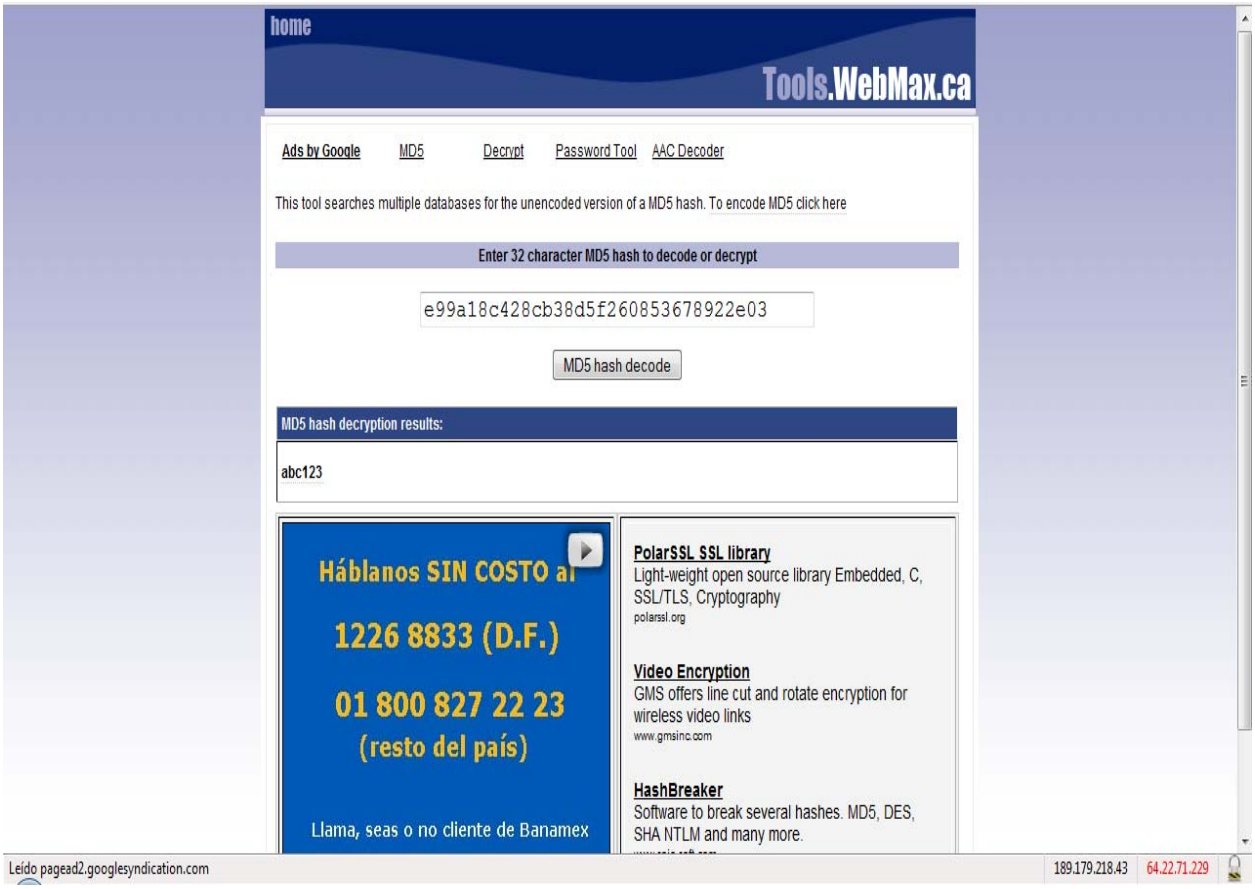

Figura 5.35 Contraseña de Usuario Gordonb

Como se observa en la figura 5.35 se pudo romper el hash de la contraseña del usuario gordonb. Esto se hace con cada uno de los hashes y de esta manera se obtienen las credenciales de todos los usuarios de la aplicación.

5.4. Resultados

Como se observó en las pruebas realizadas el lograr el acceso a la aplicación y realizar los ataques presentados fue relativamente sencillo y se obtuvo lo siguiente:

- Se obtuvieron credenciales de acceso debido a que no se obliga a la aplicación a manejar las comunicaciones mediante un canal cifrado.
- Fue posible obtener credenciales de acceso mediante un ataque por fuerza bruta exitoso debido a que no se tiene complejidad en las contraseñas.
- Fue posible el secuestro de sesión debido a que las sesiones son estáticas.
- Se realizó con éxito ataques de inyección de comandos, XSS y SQL injection debido a que no se cuenta con una validación adecuada de los parámetros de entrada que recibe la aplicación.

Si bien las pruebas fueron realizadas sobre un laboratorio y con una aplicación diseñada para hacer ejercicios de ataques de seguridad informática, es importante señalar que en la mayoría de los casos muchas de las empresas presentan todas estas vulnerabilidades e incluso más de las presentadas en esta prueba. Muchas veces estas vulnerabilidades se presentan por desatenciones en el aseguramiento de los equipos o como se observo en la figura 5.22 por mal diseño de las aplicaciones.

Con todo lo descrito a lo largo de este reporte se puede observar que la seguridad informática es un punto importante para cualquier empresa por diversas razones por lo tanto no puede ser ignorada.

Capítulo 6.- Conclusiones y Recomendaciones Generales

En los tiempos modernos, los computadoras se integran cada vez más a las actividades diarias que realizamos, facilitándonos tareas que antes eran lentas y repetitivas, haciendo el trabajo más fácil, mejorando no solamente la calidad de nuestros resultados, sino nuestro nivel de vida.

Estas bondades que otorgan estas máquinas traen adicionalmente ciertos riesgos y es responsabilidad nuestra, no solamente como usuarios sino como beneficiarios de las virtudes de estas máquinas, ser parte integral de lo que es llamado Seguridad Informática.

Hoy en día la seguridad informática significa más que mantener a extraños fuera de la red y es importante no sólo para las organizaciones sino para uno mismo. Como se pudo observar en la prueba de penetración realizada en el capítulo 5, irrumpir en un sistema es relativamente sencillo.

Por lo tanto, es necesario generar una cultura de seguridad informática en la gente y no sólo a nivel usuario sino también administrativo o de lo contrario no importa que tan avanzada sea la tecnología de seguridad que se posea, el factor humano siempre será la debilidad más grande. La Seguridad Informática empieza en cada uno de nosotros.

Es importante entender que la responsabilidad de un sistema seguro no cae únicamente sobre el sistema operativo en el cual se apoya sino que debe considerarse una posición compartida, junto con el administrador del sistema y los usuarios finales. El administrador es quien mantiene la potestad de decidir qué medidas se deben tomar para alcanzar un nivel de seguridad óptimo y también quien debe garantizar la optimización del sistema día a día para que, con el pasar del tiempo, no se convierta en un sistema obsoleto. El papel de los usuarios no es menos importante, ya que deberían estar educados para que asuman su papel como elemento activo en el sostenimiento de las políticas de seguridad definidas. En cuanto a dichas medidas de seguridad, nunca se llegará el momento en que esté todo escrito sobre el tema sino que se convertirá en un camino que crece paralelamente con el avance tecnológico de las comunicaciones y el manejo de la información.

Queda claro entonces que no hay una receta única y especial de cómo hacer un sistema seguro, sino que la seguridad es una forma de vida que parte desde el uso

consciente de herramientas y costumbres y esto no se logra de un momento a otro, la seguridad informática es un proceso constante y laborioso.

Al igual que en la vida real seguimos unas pautas de seguridad básica, debemos hacer lo propio con la información que manejamos especialmente cuando se hace uso de Internet. Normalmente, esto no se hace por pereza o por exceso de confianza.

Algunas recomendaciones básicas para aumentar la seguridad del equipo de cómputo son las siguientes:

- Configurar contraseñas/passwords seguros en todas las cuentas. No está demás que se aclare hasta el cansancio no utilizar contraseñas con palabras de uso frecuente como fechas de cumpleaños, nombres propios o palabras muy cortas. Conviene utilizar contraseñas largas, de por lo menos ocho caracteres, con letras mayúsculas y minúsculas y en lo posible intercalar algún número.
- Confidencialidad en las contraseñas: Esta es otra característica que hay que tener en cuenta, no divulgar la contraseña ni tenerla escrita en un lugar a la vista. Reemplazarla cada tiempo determinado y no utilizar la misma contraseña en diferentes sistemas (inicio de sesión, correo electrónico, banca en línea).
- Restricción por usuario de ejecución de programas: Una de las mejores implementaciones de seguridad es considerar el uso de políticas de grupo en la red local, desde la cual se puede restringir el acceso a instalación y ejecución de aplicaciones por grupos de usuarios.
- Restricción de booteo de los equipos: Para que arranquen sólo desde el disco duro adecuado (un sistema que permita el arranque desde otra unidad ofrece una forma extremadamente sencilla de acceso a los datos del mismo, sobrepasando todas las restricciones de seguridad locales).
- Utilización de claves BIOS: Para mantener a los usuarios alejados de estas mismas.
- Mensajes cifrados: Una costumbre poco utilizada pero de muy fácil implementación es el uso de llaves públicas y privadas para garantizar la comunicación entre de dos partes de forma única y segura como GnuPG y PGP.
- Sustituir el uso de protocolos que envíen la información en texto por aquellos que cuenten con cifrado de comunicación.
- Cerrar o eliminar servicios que no sean estrictamente necesarios.
- Utilizar herramientas de monitoreo de equipos de red y sus servicios. Si es posible coordinado con avisos por email para un aviso instantáneo al administrador del sistema.
- Mantener todos los equipos con las actualizaciones y parches de seguridad más recientes.
- Hacer un chequeo periódico de los registros del sistema.

- Organizar los usuarios en grupos. Si están bien organizados, y deshabilitadas cuentas que no se utilizan es más difícil tener acceso a los equipos informáticos.
- Configurar los puertos en los routers y firewall. Aprovechando el uso de los puertos permitidos y denegados, o aplicaciones con conexión a internet permitida o no.
- Implementar backups (respaldos). No hay nada peor que perder todos los archivos en los que se invierte tiempo, por eso lo mejor es hacer copias de seguridad cada cierto tiempo y comprobar que funcionan correctamente.
- Instalar un antivirus en el equipo y programarlo para que realice revisiones periódicas. Verificar también periódicamente que está activo (muchos virus detienen los programas antivirus y dejan al equipo indefenso frente a otros ataques). Además, cada día aparecen virus nuevos y para poder protegerse de ellos, el antivirus necesita conocer la “firma”, es decir, las características de esos virus. Actualizar el antivirus, bien manualmente bien de forma programada, frecuentemente y si fuera posible, a diario.
- Asegurarse que todo el software instalado en el equipo proviene de una fuente conocida y segura. No instalar copias de software pirata. Además de transgredir la Ley, pueden contener virus, spyware o archivos de sistema incompatibles con los del equipo, lo cual provocará inestabilidad. Tampoco se debe confiar en los archivos gratuitos que se descargan de sitios web desconocidos, ya que son una potencial vía de propagación de virus. En cualquier caso, se debe analizar con el antivirus cualquier archivo que se descargue de una página web.
- Desconfiar de lo desconocido. Esto es lo más importante de todo, de nada sirve todo lo anterior si se ejecutar cualquier programa que se nos presente o vamos a darle la contraseña a cualquiera que la pida.

Aunado a esto es importante conocer todas las normas, estándares y buenas prácticas de seguridad informática e implementar aquellas que funcionen mejor para lo que se quiere obtener. Por lo tanto es indispensable primero identificar las necesidades que la empresa o uno mismo requiera en cuanto al manejo de la información se refiere y de acuerdo a esto implementar los mecanismos de seguridad necesarios y no aquellos que puedan repercutir en el desempeño de la operación. Por lo tanto cabe señalar que cuando se busca seguridad en la información es necesario concentrarse en hacer que el sistema a proteger sea fiable de manera costo-efectiva y no que sea seguro al cien por ciento, ya que nada lo es.

Este reporte ha visto de manera general lo que implica la seguridad informática, sin embargo, para poder tener una visión más amplia de esta es necesario conocer como mínimo los siguientes estándares:

ISO 17799. Este estándar define la seguridad de la información y menciona algunas razones por las cuales es necesario proteger la información, e igual de importante, menciona que puntos de inicio se pueden considerar para implementar la seguridad de la información

ISO 7498-2. Éste estándar proporciona una descripción general de los servicios de seguridad y sus respectivos mecanismos que pueden ser implementados, también describe donde se pueden usar dichos servicios y mecanismos

ISO 27001. Esta norma no está orientada a aspectos técnicos sino a aspectos organizativos, es decir, tiene por objetivo organizar la seguridad de la información, debido a ello propone acciones de establecimiento, implementación, operación, monitorización, revisión, mantenimiento y mejora del Sistema Administrativo de Seguridad Informática (ISMS por sus siglas en ingles).

Anexos

La siguiente tabla hace referencia a los servicios con los que cuenta Windows XP. La tabla muestra el estado recomendado para cada uno de estos servicios con el fin de mantener el equipo lo más seguro posible asumiendo que este es utilizado para funciones básicas.

SERVICE NAME	RECOMMENDED STATE
Alerter	Disabled
Application Layer Gateway	Automating if using ICS; disable if not
Application Management	Disable unless you participate in an active directory domain
Automatic Updates Services	Requieres Cryptographic to be running. Automatic if you don´t wish to use windows update manually.
Background intelligent Transfer Service	Disabled
Clipbook	Disabled
COM+Event System	Disabled
COM+System Application	Disabled
Computer browser	Disabled
Cryptographic Services	Automatic
DHCP clients	Automatic if require; disabled if not.
Distributed link tracking client	Disabled
Distributed transaction coordinator	Disabled
DNS client	Automatic
Error reporting Service	Disabled
Event log	Automatic
Fax Service	Disabled; or don´t install from distribution media
Telephony	Disabled unless required
FTP Publishing service	Disabled; or don´t install from distribution media
Help and support	Automatic
Human interface device access	Disabled
IIS admin	Disabled; or don´t install from distribution media
IMAPI CD-Burning COM Service	Automatic
Indexing Service	Disabled
Internet connection firewall an internet connection sharing	Automatic if sharing connection, disable if not required
IPSEC services	Disabled
Logical disk manager	Manual
Logical disk manager administrative Service	Manual
Message queuing	Disabled; or don´t install from distribution media
Message queuing triggers	Disabled; or don´t install from distribution media
Messenger	Disabled
MS software shadow copy provider	Enabled
NetMeeting remote desktop sharing	Disabled
Network connections	Automatic
Network DDE	Disabled
Network DDE DSDM	Disabled
Network location awareness	Disable unless running ICS or ICE
NTLM security support provider	Automatic
Performance logs and alerts	Disabled
Plug & Play	A

Portable media serial number	Disabled
Print spooler	Automatic
Protected storage	Disabled
QoS RSVP	Disable unless required by your network administrator
Remote access auto	Disabled
Remote procedure call locator	Disabled
Remote registry service	Disabled
Removable storage	Disabled
RIP listener	Disabled; or don´t install from distribution media
Routing and remote access	Disabled; or don´t install from distribution media
Secondary logon	Automatic
Security accounts manager	Automatic
Simple TCP/IP services	Disabled; or don´t install from distribution media
SNMP service	Disabled; or don´t install from distribution media
System restore service	Disabled
TCP/IP NetBIOS helper service	Disabled unless sharing is enabled
Task scheduler	Disabled unless absolutely required
Telnet	Disabled; or don´t install from distribution media
Terminal services	Disabled; or don´t install from distribution media
Upload manager	Disabled
Volume shadow copy	Disabled
Windows audio	Automatic
Windows installer	Manual
Windows management instrumentation (WMI)	Automatic
Windows management instrumentation driver extension	Manual
Windows time	Automatic
Wireless zero configuration	Disabled
WMI performance adapter	Disabled
Workstation	Automatic
World wide web publishing services	Disabled; or don´t install from distribution media

Tabla Anexo. Servicios Recomendados

Referencias

Libros

I. CHARLES P. PFLEEGER, SHARI LAWRENCE PFLEEGER **"Security in Computing",** New Jersey, USA: Prentice Hall, Third Edition, 2003

II. VICENTE ACEITUNO CANAL **"Seguridad de la información",** Madrid, España: Editorial Limusa, Primera Edición, 2004

III. ERIC A. FISCH, GREGORY WHITE, **"Secure computers and networks: Analysis Design and Implementation",** USA: CRC, First Edition, 2000

IV. BRUCE SCHNEIER **"Secrets & Lies: Digital Security in a Networked World",** Indianapolis, USA: Editorial Wiley, First Edition, 2004

V. MARIA GRAZIA FUGINI, CARLO BELLETTINI **"Information Security: Policies and Actions in Modern Integrated Systems"** Hersbey, USA: Idea Group Publishing, First Edition, 2004

VI. DEBORAH RUSSELL , G.T. GANGEMI SR. **"Computer Security Basics",** USA: O'REILLY, First Edition, 1991

VII. S.M. BHASKAR, S.I. AHSON **"Information Security: A Practical Approach",** Atlanta, USA: Alpha Science, First Edition 2008

VIII. JONATHAN HASSELL **"Hardening Windows"** New York, USA: Apress, Second Edition, 2006

IX. DAFYDD STUTTARD, MARCUS PINTO **"The Web Application Hacker's Handbook: Discovering and exploiting Security Flaws",** USA: Wiley, First Edition, 2007

X. ARTHUR CONKLIN, GREGORY WHITE, CHUCK COTHREN, DWAYNE WILLIAMS, ROGER L. DAVIS "**Principles of Computer Security"**, USA: McGrawHill, Second Edition, 2004

Apoyo Didáctico

I. EC-COUNCIL OFFICIAL CURRICULUM
"**Ethical Hacking & Countermeasures "**
Version 5, Volume 1 – 4
Del curso de Certificación Ethical Hacking
EC-COUNCIL 2006

Documentos

I. OWASP Foundation
“OWASP Testing Guide V3.0”
OWASP Foundation 2008
Ultima actualización

Internet

- http://www.seguridadinformatica.es/
- http://www.hispasec.com/
- http://www.segu-info.com.ar/
- http://www.shellsec.net/
- http://www.hackhispano.com/
- http://www.inegi.gob.mx/inegi/contenidos/espanol/ciberhabitat/museo/cerquita/redes/seguridad/intro.htm
- http://9lessons.blogspot.com/2008/12/sql-injection.html
- http://www.taringa.net/posts/info/2253257/Esteganograf%C3%ADa-vs_-Criptograf%C3%ADa.html
- http://www.scribd.com/doc/18997468/Seguridad-Informatica-Tecnicas-Comunes-de-Ataque-a-Sistemas-Unix-o-Derivados
- http://euitio178.ccu.uniovi.es/wiki/images/2/22/XSS_Ataque_reflejado.gif
- https://www.ccn-cert.cni.es/publico/serieCCN-STIC401/es/n/non_repudiation.htm