Herramienta para la Detección de Vulnerabilidades basada en la Identificación de Servicios

David A. Franco, Jorge L. Perea y Luis C. Tovar

Universidad de Cartagena, Facultad de Ingeniería, Grupo de Investigación en Tecnologías de las Comunicaciones e Informática, GIMATICA, Avenida del Consulado, Calle 30, No. 48 – 152, Cartagena-Colombia. (e-mail: dfrancob@unicartagena.edu.co, jochyone@gmail.com, ltovarg@unicartagena.edu.co).

Recibido Ene. 03, 2013; Aceptado Mar. 11, 2013; Versión final recibida May. 10, 2013

Resumen

El objetivo principal de este trabajo fue diseñar un nuevo enfoque para la detección y evaluación de vulnerabilidades en equipos de red mediante la técnica de identificación de servicios. Este enfoque consiste en determinar los nombres y versiones de los servicios activos en un equipo de red para luego buscar las vulnerabilidades de seguridad de los mismos en la Base de Datos Nacional de Vulnerabilidades. Se desarrolló una herramienta computacional llamada UdeCEscaner y se escogieron varios laboratorios de pruebas. Los resultados permitieron determinar que el enfoque propuesto es efectivo para la identificación y evaluación de vulnerabilidades. Se concluye que la implementación del enfoque propuesto ayuda a los administradores de las tecnologías de la información y la comunicación en la mejor comprensión de los riesgos reales a los que están expuestos, facilitando la mitigación de vulnerabilidades de seguridad.

Palabras clave: vulnerabilidad de redes, servicios de red, seguridad, pruebas de penetración, identificación de servicios

Vulnerability Detection Tool using Banner Grabbing

Abstract

The main objective of this work was the design of a new approach for discovering and assessing vulnerabilities in network services through banner grabbing. This approach consists of identifying the services that are active in a network system to then find security vulnerabilities of such systems in the National Vulnerability Database. A tool called UdeCEscaner was implemented and some laboratories were selected for testing. The results show that this approach can effectively detect vulnerabilities in network services. It is concluded that the proposed approach helps managers of information and communication technology administrators to better understand the risks to which they are exposed to facilitate mitigation actions of security vulnerabilities.

Keywords: network vulnerabilities, network services, security, penetration test, banner grabbing

INTRODUCCIÓN

En los últimos años la interconexión de sistemas computacionales ha crecido enormemente y cada vez son más las ventajas que se obtienen de dicha interconexión. Sin embargo, el reciente incremento de ataques contra los sistemas computacionales y los servicios que estos ejecutan representan una seria amenaza para todo tipo de organizaciones. Las vulnerabilidades que se encuentran con más frecuencia generalmente brindan a los atacantes el control total del sistema o la denegación del acceso a sus servicios.

Diseñar enfoques efectivos para detectar estas vulnerabilidades es importante para los evaluadores de la seguridad de la información. Muchos enfoques han sido propuestos recientemente para la detección de vulnerabilidades de servicios activos en equipos de red. Algunos trabajos (Li et al., 2010; Duan et al., 2008; Corral et al., 2005) proponen novedosos enfoques para detección de vulnerabilidades en variedad de plataformas, sin embargo no utilizan sistemas de puntuación de vulnerabilidades, limitando así la información que un administrador de sistemas computacionales pueda obtener sobre el riesgo al que está expuesto (Lin et al., 2008). Otros trabajos (Dai et al., 2010; Watanabe et al., 2010; Cheng et al., 2010; Antunes y Vieira, 2011; Antunes y Fuentecilla, 2011; Dessiatnikoff et al., 2011) se basan en el envío de entradas especialmente elaboradas y el análisis de patrones de salida específicos para detectar vulnerabilidades en caso de que estas no concuerden con un patrón esperado, no obstante la efectividad del enfoque en mención se ve en ocasiones afectada por fallos en el proceso de generación de patrones, adicionalmente el enfoque en cuestión puede comprometer la disponibilidad del sistema computacional sometido a evaluación (Guo et al., 2005). A esto se le suma el problema de que muchas de las herramientas existentes en la actualidad requieren de conocimientos avanzados para su utilización, además de tener un costo elevado (Ke et al., 2009).

El trabajo presentado en este artículo propone un nuevo enfoque para la detección de vulnerabilidades basado en la identificación de servicios activos en equipos de red. Dada una o varias direcciones internet, nombre de equipo o localizador uniforme de recursos se realiza la identificación de servicios mediante la herramienta NMAP, la cual genera un archivo en formato XML que será procesado para obtener los nombres y versiones de los servicios encontrados activos. Luego se construye una sentencia SQL para buscar en una base de datos (previamente configurada con la información de la vulnerabilidades que han sido reportadas a la NVD) las vulnerabilidades asociadas a tales servicios. El resultado de la búsqueda se presenta luego al usuario en un reporte que contiene diferentes métricas acerca de cada vulnerabilidad, además de un identificador y una descripción. El enfoque descrito realiza un reducido número de interacciones con los servicios activos, de modo que evita el problema de la afectación de la disponibilidad del objetivo de evaluación. Por otro lado utiliza el sistema común de puntuación de vulnerabilidades CVSS para brindar a los administradores de sistemas computacionales información precisa y contextualizada sobre el riesgo real al que están expuestos (Maggi et al., 2008), adicionalmente provee identificadores y descripción CVE para facilitar la mitigación de las vulnerabilidades de seguridad.

Para evaluar la efectividad del enfoque planteado se realizó su implementación en una herramienta computacional llamada UdeCEscaner la cual se caracteriza por ser ágil y tener una interfaz amigable y de fácil uso. Posteriormente se llevó a cabo un estudio experimental en el cual se seleccionaron tres laboratorios de pruebas de penetración para ser sometidas a análisis. Por último los resultados encontrados por UdeCEscaner concernientes a la identificación y evaluación de vulnerabilidades se compararon con los obtenidos por las herramientas de detección de vulnerabilidades Nexpose y Nessus. Como resultado final se obtuvo que el enfoque propuesto sea efectivo para la identificación y evaluación de vulnerabilidades de seguridad.

TRABAJOS RELACIONADOS

La detección de vulnerabilidades ha sido abordada desde diferentes enfoques y ha sido dirigida hacia diversidad de plataformas (sistemas operativos, entornos Web, aplicaciones móviles, etc.). Li et al. (2010) proponen un proceso de evaluación de vulnerabilidades de seguridad en entornos de computación en la nube basado en análisis de riesgo y rendimiento. Otro trabajo (Corral et al., 2005) plantea un enfoque para la automatización de sistemas de detección de vulnerabilidades en redes inalámbricas basado en la metodología OSSTMM. De la misma manera se presenta un interesante esquema para la detección de vulnerabilidades en aplicaciones Web basado en análisis de características en formularios Web (Zhang et al., 2010), a la vez que soporta su enfoque en la construcción de una herramienta informática que lo implementa. Duan et al. (2008) presenta un enfoque que utiliza la técnica de identificación de servicios para la evaluación de vulnerabilidades. No obstante los anteriores trabajos no soportan el proceso de mitigación de vulnerabilidades al no basar su evaluación en base a métricas CVSS (Maggi et al., 2008). Austin y Williams (2011) presentan una comparación entre las técnicas de evaluación existentes. En forma similar otro trabajo establece parámetros para la construcción de herramientas software de evaluación de

vulnerabilidades (Liu et al., 2007). Estos enfoques no son soportados por herramientas software ni presentan mecanismos concretos para la identificación de vulnerabilidades.

Otro tipo de técnicas han sido desarrolladas para la detección de vulnerabilidades, Dai et al. (2010) presentan una metodología de detección de vulnerabilidades basado en Fuzzing, en la cual se modifica la configuración de un programa y se evalúa si ocurren violaciones de invariantes de seguridad para detectar vulnerabilidades, un análisis equivalente es presentando en (Zhang et al., 2012a). Un enfoque similar propone la detección de vulnerabilidades mediante la generación de patrones de evaluación en forma de entradas de datos (Watanabe et al., 2010), para evaluar la existencia de vulnerabilidades en base a las salidas de las aplicaciones, no obstante la evaluación de vulnerabilidades se limita a aplicaciones flash. Cheng et al. (2010) proponen un enfoque automatizado independiente de la plataforma para la detección de vulnerabilidades en programas binario basado en la manipulación de patrones de entrada y el análisis estático de código. En la misma dirección Zhang et. al (2012b) propone un método para detectar vulnerabilidades en aplicaciones software basado en una combinación de análisis estático y dinámico del código fuente.

En otro sentido se presenta un enfoque para la detección de vulnerabilidades de inyección en servicios Web mediante la generación de un conjunto de peticiones válidas y otro de peticiones maliciosas (Antunes y Vieira, 2011), para determinar la presencia de vulnerabilidades mediante la monitorización del tráfico de la base de datos. De la misma manera Dessiatnikoff et al. (2011) plantean un algoritmo para la detección de vulnerabilidades en aplicaciones Web que se basa en la generación automática de entradas maliciosas para detectar vulnerabilidades basándose en los patrones de salida encontrados, este algoritmo es implementado en un escáner de vulnerabilidades elaborado por los investigadores. En la misma dirección Shar y Tan (2012) proponen un método para la detección de vulnerabilidades de tipo XSS en aplicaciones web, mediante análisis estático de código fuente. Así mismo Antunes y Fuentecilla (2011) presentan un enfoque basado en la inferencia de perfiles de comportamiento de la pila de protocolos TCP/IP de servidores de red, para analizar su estado luego de ejecutar múltiples casos de evaluación y determinar la presencia de vulnerabilidades si alguna salida viola el perfil de comportamiento. De otro lado Yeo (2013) propone el uso de pruebas de penetración para la mitigación de vulnerabilidades, dichas pruebas de penetración se basan en las técnicas descritas anteriormente. Finalmente en (Shahmehri et al., 2013) se propone un novedoso enfoque para la detección de vulnerabilidades basado en la localización de sus causas en las trazas de ejecución de los programas binarios.

Todos los trabajos mencionados se basan en envío de entradas especialmente elaboradas y el análisis de patrones de salida específicos para detectar vulnerabilidades en caso de que estas no concuerden con un patrón esperado, es decir, que dichos trabajos consideran cada perturbación del ambiente como una falla y el resultado de la misma será que la seguridad quedará comprometida. No obstante la efectividad de estos enfoques puede verse afectada, puesto que el hallazgo de vulnerabilidades depende altamente de las entradas que se generen, las cuales pueden no ocasionar el fallo de los objetivos de evaluación y por consiguiente no se detectarían algunas vulnerabilidades. Adicionalmente la generación y ejecución de estas entradas puede comprometer la disponibilidad del sistema computacional que está siendo evaluado, por eso Guo et al. (2005) proponen un enfoque para realizar este tipo de evaluación de forma que se minimice el impacto sobre el sistema computacional evaluado. El enfoque que se presenta en este artículo utiliza la técnica identificación de servicios para la detección de vulnerabilidades evitando con ello la pérdida de eficacia y afectación de la disponibilidad inherentes a las técnicas basadas en patrones de entrada/salida.

DEFINICIONES

Se presentan algunas definiciones y conceptos necesarios para situar la propuesta en su justo contexto. Por lo tanto en esta sección se definen términos tales como Base de datos Nacional de Vulnerabilidades (NVD), código de vulnerabilidades y amenazas comunes (CVE), Sistema común de puntuación de vulnerabilidades (CVSS), la Identificación de servicios y la prueba de penetración (penetration test o pentest).

NVD: Base de datos nacional de vulnerabilidades

NVD (National Institute of Standards and Technology, 2011) es un repositorio estandarizado del gobierno de los Estados Unidos en el cual se encuentra almacenada información acerca de la gestión de vulnerabilidades. Estos datos permiten la automatización de la gestión de vulnerabilidades y la toma de medidas de seguridad. NVD incluye bases de datos con listas de control de seguridad, fallos de seguridad relacionados con software, errores de configuración, nombres de productos y métricas de impacto. En la actualidad cada día se adiciona a la NVD un promedio de 12 nuevas vulnerabilidades. La información sobre estas vulnerabilidades puede ser accedida a través de diferentes fuentes de datos proporcionadas en formato XML cuya actualización es continua.

CVE

El código CVE (vulnerabilidades y amenazas comunes) es un identificador que se asigna a cada vulnerabilidad que se conoce públicamente con el fin de que pueda ser identificada de forma unívoca (The MITRE Corporation, 2011). Este código fue creado por la corporación MITRE y permite que los usuarios puedan conocer de forma objetiva las vulnerabilidades de un sistema computacional. Los identificadores CVE se presentan en el formato CVE-AÑO-NÚMERO y están acompañados de una breve descripción de la vulnerabilidad o amenaza y un grupo de referencias pertinentes.

CVSS: Sistema común de puntuación de vulnerabilidades

CVSS (Mell et al., 2007) es un sistema de puntuación de vulnerabilidades diseñado con el fin de proporcionar un método abierto y estandarizado para la clasificación de vulnerabilidades en tecnologías de la información. Con esto ayuda a las organizaciones a priorizar y coordinar una respuesta concertada para la mitigación de vulnerabilidades de TIC. Adicionalmente CVSS provee a profesionales en seguridad informática, ejecutivos y usuarios finales un lenguaje común para discutir la severidad de las vulnerabilidades de seguridad.

CVSS utiliza tres tipos de métricas: Base, temporal, ambiental. La primera representa la característica fundamental de la vulnerabilidad que no varía con el tiempo ni el ambiente; la segunda representa las características que cambian con el tiempo pero no varían entre ambientes de trabajo; la tercera representa las características de las vulnerabilidades que solo son relevantes para un entorno específico. Las tres métricas de CVSS están en un rango de 0 a 10 y se utilizan para determinar la severidad de las vulnerabilidades.

Identificación de servicios

Es una técnica de enumeración utilizada para obtener información acerca de un sistema computacional dentro de una red y de los servicios que se ejecutan en sus puertos. Esta técnica suele ser utilizada por administradores para determinar los sistemas y servicios en sus redes; sin embargo los atacantes informáticos la utilizan para conocer las versiones de los servicios activos en los equipos de red (Franco et al., 2012) y determinar si existen exploits para estos (McClure et al., 2005).

Laboratorio de pruebas de penetración

Se conoce como prueba de penetración (penetration test o pentest) a la valoración de la seguridad de un sistema computacional en la que un evaluador lleva a cabo ataques del mundo real para identificar métodos para eludir los mecanismos de seguridad de una aplicación, sistema o red. Dicha evaluación generalmente implica la ejecución de ataques reales contra sistemas reales mediante las herramientas y técnicas utilizadas comúnmente por los atacantes informáticos. La mayoría de las pruebas de penetración involucran la búsqueda de vulnerabilidades en uno o más sistemas computacionales que puedan ser utilizadas para ganar acceso a los mismos (Scarfone et al., 2008). Debido a que una prueba de penetración implica la ejecución de ataques reales, aquellos interesados en aprender esta técnica no disponían de plataformas para ponerla en práctica de forma segura y legal. Por lo que la única forma de aprender dicha técnica es mediante la creación de laboratorios de pruebas de penetración, los cuales consisten en plataformas informáticas (Sistemas operativos, aplicaciones Web, entre otras) configuradas con problemas de seguridad para que los interesados puedan poner en práctica sus habilidades para llevar a cabo una prueba de penetración (Wilhelm, 2009).

LA HERRAMIENTA UdeCEscaner

En esta sección se presenta el marco de trabajo para la herramienta UdeCEscaner. Este consiste principalmente de 4 módulos: XML_DBMS, Escáner, Construcción de consultas y Reportes. La herramienta UdeCEscaner es comparada con Nessus versión 5.0.0 y Nexpose. El marco de trabajo se presenta en la figura 1, en la cual las líneas sólidas representan la dirección del flujo de datos y los números entre paréntesis representan el orden de ejecución del mismo. En las siguientes subsecciones se describe cada uno de los módulos.

Fig 1: Marco de trabajo de la herramienta UdeCEscaner.

XML_DBMS

La información acerca de las vulnerabilidades que se encuentran reportadas en la NVD es accesible a través de archivos en formato XML. Estos han sido habilitados para facilitar la automatización del proceso de identificación y evaluación de vulnerabilidades. El módulo XML_DBMS toma estos archivos como entrada y realiza un volcado de los mismos en un sistema gestor de bases de datos (DBMS) para que posteriormente el módulo Construcción de consultas pueda acceder a esta información mediante el lenguaje de consulta estructurado (SQL).

Escáner

Dada una dirección de internet, nombre de servidor o localizador uniforme de recursos este módulo identifica los puertos que se encuentran abiertos en el objetivo especificado, posteriormente mediante la identificación de servicios determina el nombre y versión de los servicios que se ejecutan en cada puerto. Esta información es generada en formato XML y posteriormente almacenada en un archivo por la herramienta NMAP. Luego de esto se realiza el volcado del archivo generado en estructuras de datos. Estas contendrán la siguiente información: dirección IP de los equipos escaneados, número de los puertos encontrados y la versión y nombre de los servicios enumerados.

Construcción de consultas

Este módulo accede a las estructuras mencionadas arriba, para obtener la versión y el nombre de los servicios enumerados en cada equipo. Con esta información se construyen sentencias SQL que serán utilizadas para determinar las vulnerabilidades asociadas a los servicios encontrados en los diferentes equipos.

Reportes

Mediante las sentencias SQL creadas por el módulo anterior se realizan consultas al DBMS que contiene información detallada de las casi 50.000 vulnerabilidades contenidas en la NVD. Dichas consultas se construyen de forma que sea posible determinar si la versión y el nombre de cada servicio enumerado está asociada a una o varias vulnerabilidades. Con base en el resultado de la consulta SQL se presenta al usuario de la aplicación un reporte de las vulnerabilidades encontradas en cada equipo, dicho reporte contiene principalmente: el identificador de la vulnerabilidad en formato CVE, el nombre y versión del producto, la complejidad y vector de acceso, el impacto de la vulnerabilidad sobre la confidencialidad, integridad y disponibilidad del servicio y finalmente una breve descripción de la vulnerabilidad. Cabe precisar que las métricas citadas arriba corresponden a métricas del CVSS.

Diseño del Experimento

Se propone un enfoque para la detección de vulnerabilidades y lo soporta en la herramienta informática llamada UdeCEscaner. Con el fin de determinar la efectividad de dicho enfoque se utilizó la herramienta en cuestión para encontrar las vulnerabilidades de tres laboratorios de pruebas de penetración, luego se utilizaron dos de las herramientas más utilizadas en la actualidad, con el objetivo de identificar vulnerabilidades sobre las mismas tres plataformas de prueba, finalmente se estableció un comparativo entre los resultados obtenidos por las herramientas comerciales y los encontrados con la herramienta que soporta el enfoque propuesto (UdeCEscaner). Apoyándose en la comparación realizada se determinó la efectividad del enfoque planteado. Los laboratorios de pruebas de penetración utilizados fueron los siguientes:

De-ICE 1.100: Hace parte de un grupo de siete plataformas de prueba desarrollados por Heorot.net, esta consiste de una distribución de Linux llamada Slax en la cual se ejecutan diferentes servicios de red. Esta plataforma cuenta con diversas vulnerabilidades que pueden ser aprovechadas para obtener acceso a la cuenta de usuario root.

Metasploitable: Creado por el equipo de Metasploit, es al igual que la anterior una distribución de Linux (específicamente Ubuntu) en la cual se ejecutan varios servicios de red vulnerables, por medio de la explotación de dichos servicios se puede obtener de diversas formas la cuenta de súper usuario root.

Kioptrix nivel 1: Creada por un grupo de expertos en seguridad informática, hace parte de un conjunto de cuatro laboratorios de pruebas de penetración que al igual que en los casos anteriores ejecutan un conjunto de servicios de red de los que puede sacarse provecho para obtener acceso a la cuenta de usuario con mayor privilegio en el sistema. Kioptrix nivel 1 está basada en una distribución Red Hat de Linux.

RESULTADOS Y DISCUSIÓN

Los resultados han sido divididos en dos grupos, en primer lugar se presentan aquellos resultados relacionados exclusivamente con la detección de vulnerabilidades haciendo énfasis en el análisis de vulnerabilidades de severidad alta; luego se dan a conocer aquellos relacionados con la comparación realizada entre las características de cada herramienta de detección.

Las figuras 2 a 4 presentan el primer grupo de resultados. En cada una se presentan los resultados obtenidos por las 3 herramientas de detección de vulnerabilidades para cada laboratorio de pruebas de penetración en particular. Todas estas figuras contienen una gráfica de dos dimensiones en las que el eje Y representa el número de vulnerabilidades encontradas en la evaluación y el eje X el nivel de severidad determinado por cada herramienta. Es importante precisar en este punto que se ha establecido un mapeo entre el nivel de seguridad arrojado por cada herramienta y una escala definida por los autores con el fin de facilitar la comparación entre las herramientas, este se presenta en la siguiente tabla.

Tabla 1. Mapeo entre las escalas de severidad utilizadas por las herramientas de evaluación y la escala definida por los autores.

UdeCEscaner	Nexpose	Nessus	Nueva escala
Alto	Critico	Critico, Alto	Alto
Medio	Severo	Medio	Medio
Bajo	Moderado	Bajo, Info	Bajo

La tabla anterior indica entre otras cosas que la severidad Alta para UdeCEscaner, equivale a la Crítica para Nexpose, así como también a la Crítica y Alta para Nessus y que para efectos de las gráficas presentadas en lo que sigue, se tomarán como nivel Alto. La figura 2 presenta los resultados obtenidos para la plataforma Kioptrix nivel 1.

Fig 2: Resultados para el laboratorio de pruebas de penetración Kioptrix versión 1.

Como se puede apreciar en la figura 2, la herramienta que detectó más vulnerabilidades fue Nessus con un total de 85, seguido por Nexpose con 80 y UdeCEscaner con 12. En este caso la herramienta UdeCEscaner presenta un bajo número de vulnerabilidades encontradas con relación a las otras dos herramientas. Con relación al tipo de vulnerabilidad de severidad alta detectada por las herramientas, se halló que Nessus reportó la mayoría en los servicios OpenSSH y Apache, con fallas a nivel de desbordamiento de memoria y ejecución de código remota respectivamente; mientras que UdeCEscaner reportó este tipo de vulnerabilidades en el servicio OpenSSH relacionada con la denegación de servicios. Por último NexPose mostró resultados similares a los de Nessus, no advirtiendo la presencia de vulnerabilidades de denegación de servicios a diferencia de UdeCEscaner.

Fig 3: Resultados para el laboratorio de pruebas de penetración De-ICE 1.100.

En la figura 3 se presentan los resultados para la plataforma De-ICE 1.100, allí se aprecia que la diferencia entre el número de vulnerabilidades detectadas por UdeCEscaner (24) y Nessus (43) se ha reducido con relación a lo obtenido para la plataforma Kioptrix. Por otro lado Nexpose fue la herramienta que más hallazgos realizó con un total de 99, la mayoría de los cuales fue de nivel medio. Con relación al tipo de vulnerabilidad de severidad alta detectadas, Nessus solo encontró este tipo de vulnerabilidades en el servicio relacionado con el servidor apache, las cuales están relacionadas con desbordamiento de memoria y ejecución remota de código; mientras que UdeCEscaner detectó vulnerabilidades relacionadas con denegación de servicios en el proceso relacionado con OpenSSH; por otro lado NexPose halló fallas relacionadas con los dos servicios mencionados anteriormente, pero no encontró el problema de la denegación de servicios reportada por UdeCEscaner. Los resultados relacionados con la plataforma Metasploitable se presentan en la figura 4.

En esta ocasión se ve como nuevamente Nexpose tiene la delantera con 126 vulnerabilidades detectadas, mientras que Nessus con 67 y UdeCEscaner con 44 lo siguen de lejos. Se recalca además que UdeCEscaner realizó más hallazgos de severidad media que Nessus. Con relación al tipo de vulnerabilidades de severidad alta se obtuvo que Nessus las reportó en su mayoría en el servicio SAMBA, UdeCEscaner en el servicio BIND y Nexpose en Apache. Finalmente con el objeto de proveer un análisis más completo se realizó una comparación entre algunas características de las herramientas de detección utilizadas. La tabla 2 ilustra dicho paralelo.

Fig 4: Resultados para el laboratorio de pruebas de penetración Metasploitable.

Tabla 2: Comparación entre las características de las herramientas de detección de vulnerabilidades.

Herramienta/Parámetro	UdeCEscaner	Nexpose	Nessus
Facilidad de uso	Alta	Alta	Alta
Requerimientos del sistema	Bajo	Alto	Medio
Facilidad de instalación	Simple	Compleja	Media
Interfaz amigable	Alto	Alto	Alto
Complejidad	Baja	Alto	Alto
Nº de vulnerabilidades detectadas	80	305	195
Legibilidad del reporte final	Alta	Alta	Alta
Contenido del reporte	Medio	Alta	Medio
Velocidad de detección	Alta	Baja	Medio

Es importante resaltar algunos aspectos sobre los datos presentados en la tabla 2, en primer lugar se anota que la herramienta Nexpose a pesar de tener la mayor tasa de detección de vulnerabilidades tiene requerimientos de instalación demasiado altos (demanda por ejemplo 8Gb de memoria RAM para su instalación) en contraste con UdeCEscaner que requiere una capacidad de procesamiento y de memoria mínimos. Por otro lado el tiempo que tarda la instalación de Nexpose puede exceder las dos horas dependiendo de la potencia computacional del equipo en el cual se realiza la instalación, mientras que UdeCEscaner no necesita instalación. Adicionalmente las herramientas Nexpose y Nessus presentan una alta complejidad que si bien contribuye al hallazgo de un gran número de vulnerabilidades ocasiona el consumo excesivo de recursos computacionales, lo cual no se presenta en UdeCEscaner. De otro lado las tres herramientas presentan una facilidad de uso alta, la cual se evaluó teniendo en cuenta la velocidad de respuesta en el acceso a las diferentes vistas y la simplicidad de la navegación entre las mismas. Por último se resalta que el tiempo promedio aproximado de detección de vulnerabilidades de Nexpose y Nessus fue de 15 y 5 minutos respectivamente, mientras que UdeCEscaner presentó un tiempo promedio de detección de 2 minutos.

Con base en lo dicho anteriormente y en lo presentado en las gráficas anteriores, se puede sostener que el enfoque planteado en este artículo permitió crear una herramienta de detección de vulnerabilidades con una tasa razonable de detección, menos requisitos de ejecución, con una instalación más rápida y sencilla, y un tiempo de detección de vulnerabilidades mucho menor que las dos herramientas de detección más utilizadas en la actualidad. Adicionalmente los resultados presentados permiten conocer que la técnica de identificación de servicio es útil para detectar vulnerabilidades en servicios activos en equipos de red. Se anota por último que UdeCEscaner presenta los resultados del escaneo de vulnerabilidades en un puntaje

de riesgo CVSS real, preciso y contextualizado para ayudar a los administradores a entender el riesgo al que están expuestos. Adicionalmente cada vulnerabilidad es presentada con una breve descripción y un identificador CVE con miras a facilitar su mitigación. Se destaca además que UdeCEscaner puede evaluar las vulnerabilidades de una o varias plataformas al tiempo, dejando al usuario final solo la tarea de ingresar la dirección IP o nombre de dominio.

CONCLUSIONES

De los resultados y el análisis presentado se pueden enunciar las siguientes conclusiones sobre el enfoque propuesto y la herramienta que lo soporta: 1) La implementación del enfoque de detección de vulnerabilidades planteado es adecuada para identificar vulnerabilidades de forma fácil, mediante una interfaz amigable, con pocos requisitos ejecución, con una instalación sencilla y de forma rápida; 2) La herramienta construida ayuda a los administradores de sistemas computacionales a entender el riesgo al que están expuesto y facilita la mitigación de las vulnerabilidades; 3) La aplicación que implementa el enfoque propuesto, es capaz de detectar vulnerabilidades que otras herramientas pueden omitir; 4) Mediante la técnica de identificación de servicios se pueden detectar vulnerabilidades en los servicios activos en equipos de red pertenecientes a diferentes plataformas.

REFERENCIAS

- Antunes, J. y Fuentecilla, N, *Using Behavioral Profiles to Detect Software Flaws in Network Servers*, 2011 IEEE 22nd International Symposium on Software Reliability Engineering, 1-10 (2011).
- Antunes, N. y Vieira, M, *Enhancing Penetration Testing with Attack Signatures and Interface Monitoring for the Detection of Injection Vulnerabilities in Web Services*, 2011 IEEE International Conference on Services Computing, 104-111 (2011).
- Austin, A. y Williams, L, *One Technique is Not Enough: A Comparison of Vulnerability Discovery Techniques*, 2011 International Symposium on Empirical Software Engineering and Measurement, 97-106 (2011).
- Cheng, S., Wang, J., Wang, J., Yang, J. y Jiang, F, *PDVDS: A Pattern-Driven Software Vulnerability Detection System*, 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 536-541 (2010).
- Corral, G., Cadenas, X., Zaballos, A. y Cadenas, M, *A Distributed Vulnerability Detection System for WLANs*, In First International Conference on Wireless Internet - WICON 2005, 86-93 (2005).
- Dai, H., Murphy, C. y Kaiser, G, *Configuration Fuzzing for Software Vulnerability Detection*, 2010 International Conference on Availability, Reliability and Security, 525-530 (2010).
- Dessiatnikoff, A., Akrouf, R., Alata, E., Kaâniche, M. y Nicomette, V, *A Clustering Approach for Web Vulnerabilities Detection*, 2011 IEEE 17th Pacific Rim International Symposium on Dependable Computing, 194-203 (2011).
- Duan, B., Zhang, Y. y Gu, D, *An Easy-to-Deploy Penetration Testing Platform*, 2008 The 9th International Conference for Young Computer Scientists, 2314-2318 (2008).
- Franco, D., Perea, J. y Puella, P, *Metodología para la Detección de Vulnerabilidades en Redes de Datos*, Información Tecnológica, 23(3), 113-120 (2012).
- Guo, F., Yu, Y. y Chiueh, T, *Automated and Safe Vulnerability Assessment*, 21st Annual Computer Security Applications Conference (ACSAC'05), 150-159 (2005).
- Ke, J., Yang, C. y Ahn, T, *Using w3af to achieve automated penetration testing by live DVD/live USB*, 2009 International Conference on Hybrid Information Technology, New York, USA (2009).
- Li, H., Liang, P., Yang, J. y Chen, S, *Analysis on Cloud-Based Security Vulnerability Assessment*, 2010 IEEE 7th International Conference on e-Business Engineering, 490-494 (2010).
- Lin, C., Chen, C. y Lai, C, *A Study and Implementation of Vulnerability Assessment and Misconfiguration Detection*, 2008 IEEE Asia-Pacific Services Computing Conference, 1252-1257 (2008).

- Liu, S., Holt, L. y Cheng, B, *A Practical Vulnerability Assessment Program*, IT Professional, 9(6), 36-42 (2007).
- Maggi, P., Pozza, D. y Sisto, R, *Vulnerability Modelling for the Analysis of Network Attacks*, 2008 Third International Conference on Dependability of Computer Systems DepCoS-RELCOMEX, 15-22 (2008).
- McClure, S., Scambray, J. y Kurtz, G, *Hacking exposed: network security secrets and solutions*, 3° edición, McGraw-Hill Osborne Media, New York, USA (2005).
- Mell, P., Scarfone, K., y Romanosky, S, *CVSS:A Complete Guide to the Common Vulnerability Scoring Systems Version 2.0*, FIRST: Forum of Incident Response and Security Teams, Junio (2007).
- National Institute of Standards and Technology, *National Vulnerability Database Version 2.2*, <http://nvd.nist.gov/home.cfm>. Acceso: 5 de Marzo (2011).
- Scarfone, K., Souppaya, M., Cody, A. y Orebaugh, A, *Technical Guide to Information Security Testing and Assessment*, Technical Report Spec. Publ. 800-11, U.S. Department of Commerce, National Institute of Standards and Technology (2008).
- Shahmehri, N., Mammar, A., Montes de Oca, E., Byers, D., Cavalli, A., Ardi, S., y Jimenez, W., *An advanced approach for modeling and detecting software vulnerabilities*. Information and Software Technology. 54(9),997-1013 (2012).
- Shar, L. K., y Tan, H. B., *Automated removal of cross site scripting vulnerabilities in web applications*, Information and Software Technology, 54(5), 467-478 (2012).
- The MITRE Corporation, *Common Vulnerability and Exposures, About CVE Identifiers*, <http://cve.mitre.org/cve/cve.html>. Acceso: 5 de Marzo (2012).
- Wilhelm, T., *Professional Penetration Testing: Creating and Operating a Formal Hacking Lab*, Syngress Publishing, 101-119 (2009).
- Watanabe, T., Cheng, Z., Kansen, M. y Hisada, M, *A New Security Testing Method for Detecting Flash Vulnerabilities by Generating Test Patterns*, 2010 13th International Conference on Network-Based Information Systems, 469-474 (2010).
- Yeo, J., Using penetration testing to enhance your company's security. Computer Fraud & Security, 2013(4), 17-20 (2013).
- Zhang, L., Gu, Q., Peng, S., Chen, X., Zhao, H. y Chen, D, *D-WAV: A Web Application Vulnerabilities Detection Tool Using Characteristics of Web Forms*, 2010 Fifth International Conference on Software Engineering Advances, 501-507 (2010).
- Zhang, D., Liu, D., Lei, Y., Kung, D., Csallner, C., Nystrom, N., y Wang, W., *SimFuzz: Test case similarity directed deep fuzzing*, Journal of Systems and Software, 85(1), 102-111 (2012a).
- Zhang, R., Huang, S., Qi, Z., y Guan, H., *Static program analysis assisted dynamic taint tracking for software vulnerability discovery*. Computers & Mathematics with Applications, 63(2), 469-480 (2012b).