Diseño de un Ambiente Simulado para Seguridad de la Información

Design of a Simulated Environment for Information Security

Desenho de um ambiente Simulado para Segurança da Informação

Julián Camilo Fonseca Romero[1]

[1]Grupo de Investigación MUISCA, Facultad de Ingeniería, Fundación Universitaria Juan de Castellanos, Tunja, Colombia.

fonsecamilo89@gmail.com

Recibido / Received: 24/08/2015 – Aceptado / Accepted: 06/11/2015

Resumen

El presente artículo muestra los resultados de un proyecto de investigación tecnológica, el cual describe el diseño y desarrollo de un simulador en el que se ejecuta un test de intrusión, cuyo fin es el que las personas obtengan, refuercen e incrementen sus conocimientos en seguridad informática de una forma rápida sin las complicaciones y limitaciones que tienen algunos informáticos (desde estudiantes hasta administradores de TI), los cuales por falta de experiencia podrían afectar sus sistemas a cargo. Se describe la infraestructura de red de datos del simulador y se establecen los requerimientos tecnológicos para que el simulador funcione sin problemas. Se describen el hardware y el software usados en el simulador. Se evalúa el comportamiento del simulador al realizar tres procedimientos muy comunes y aceptados en diversas metodologías de pruebas de intrusión, como son: enumeración, análisis de vulnerabilidades y explotación. A partir del desarrollo de estos procedimientos, se analiza la estabilidad del simulador y se verifican los conocimientos que adquieran las personas que realicen prácticas sobre el mismo. Se concluye que el simulador cumple con los objetivos propuestos y que este puede ser adaptado a otras actividades de seguridad informática, como análisis forense, implementación de sistemas de defensa y desarrollo de *malware* con fines académicos e investigativos.

Palabras clave: Hacking, informática, redes, simulador, capacitación, internet.

Abstract

This article shows the results of a technological research project, which describes the design and development of a simulator in which an intrusion test is executed, whose purpose is for people to gain, strengthen and expand their knowledge in information security in a fast way without the complications and limitations that some IT professionals have (from students to IT managers), where due to lack of experience could affect their systems in charge. The data network infrastructure of the simulator is described and technological requirements are set so the simulator runs smoothly. The hardware and software used in the simulator are described. In order to analyze the results and evaluate the simulator's performance, three procedures are performed. These procedures are used in various methodologies to perform a penetration test such as: enumeration, vulnerability analysis and exploitation. With the development of these procedures, the simu-

lator's stability is analyzed and the knowledge acquired by people performing practices on the simulator is verified. It is concluded that the simulator's goals are accomplished and the simulator can be adapted to various information security activities like forensics analysis, defense systems implementation and malware development with academic and investigation purposes.

Keywords: Hacking, informatics, networks, simulator, training, internet.

Resumo

Este artigo apresenta os resultados de um projeto de investigação tecnológica, que descreve a concepção e desenvolvimento de um simulador no qual se executa um teste de intrusão, cuja finalidade é que as pessoas obter, fortalecer e expandir seus conhecimentos em segurança informática duma maneira rápida sem as complicações e limitações que têm alguns profissionais de informática (desde estudantes a gerentes de TI), que devido à falta de experiência poderia afetar os sistemas que controlam. A Infraestrutura da rede de dados do simulador é descritos e exigências tecnológicas são definidas para que o simulador funcionar sem problemas. Além disso, o hardware e software utilizado no simulador são descritos. A fim de analisar os resultados e avaliar o desempenho do simulador, três procedimentos são realizados. Estes procedimentos são usados em várias metodologias para executar um teste de penetração, tais como: enumeração, análise de vulnerabilidade e explotação. Com o desenvolvimento destes procedimentos, a estabilidade do simulador é analisada e o conhecimento adquirido por pessoas que executam práticas no simulador é verificado. Conclui-se que os objetivos do simulador são conseguidos e o simulador pode ser adaptado a diversas atividades de segurança da informação como a análise forense, implementação de sistemas de defesa e de desenvolvimento de malware com fins académicos e de investigação.

Palavras-chave: Hacking, computação, redes, simulador, treinamento, Internet.

I. Introducción

Los estudiantes que no cuentan con un ambiente real para practicar sus conocimientos en seguridad informática, o los administradores de sistemas que carecen de estos conocimientos, se encuentran con el problema de no defender eficazmente sus sistemas a cargo. Los servidores, redes de datos y clientes terminan siendo vulnerables, permitiendo que algún delincuente informático explote estas vulnerabilidades y sus ataques terminen siendo exitosos, afectando así todos los sistemas en general en cualquier momento. Los jóvenes investigadores o desarrolladores están limitados a usar máquinas virtuales para realizar sus investigaciones académicas de *malware*, virus, troyanos, etc. Frente a estas circunstancias, se propone como objetivo diseñar y desarrollar un simulador de seguridad de la información.

Para llevar a cabo este objetivo, es necesario diseñar la topología de red del simulador, desarrollar el sistema simulado "vulnerable" con sus servicios y topología de red activos, definir el rol de los usuarios del simulador y, finalmente, ejecutar el test de intrusión en el simulador. ¿Mediante el uso de un ambiente simulado para seguridad de la información, los usuarios del mismo podrían llegar a tener la suficiente experiencia como para ejecutar un test de intrusión en sistemas reales así como también los investigadores considerarían lo suficientemente útil el simulador para realizar diferentes prácticas de seguridad informática?

Teniendo en cuenta lo anterior, se propone el diseño y desarrollo de un simulador en el que se pueda trabajar sin que exista la probabilidad de, accidentalmente, afectar o interrumpir servicios informáticos críticos, como podría suceder en un ambiente real. Un ambiente simulado en el que los estudiantes pongan a prueba sus conocimientos en seguridad informática, sin preocuparse de las consecuencias y sin limitaciones. Una oportunidad donde los usuarios obtengan experiencia sin preocuparse de los

riesgos que representa ejecutar un test de intrusión a un sistema.

Las actividades simuladas traen muchos beneficios a aquellas personas que no cuentan con la experiencia necesaria para realizar algún procedimiento que implique algún riesgo. En la fuerza aérea, por ejemplo, los pilotos deben cumplir con cierto número de horas de entrenamiento virtual de vuelo antes de tomar control en un avión real. En la NASA (Administración Nacional de Aeronáutica y Espacio), los astronautas entrenan sumergidos en una piscina para simular la falta de gravedad en el espacio. Los auditores de seguridad informática sin experiencia corren el riesgo de accidentalmente dejar sin servicios a la organización auditada en el intento de obtener evidencia de un ataque exitoso. Algunos estudiantes hacen prácticas en máquinas virtuales y recrean todo el proceso de desarrollo de un test de intrusión, pero la experiencia obtenida no es suficiente para realizar un procedimiento real.

El uso de un ambiente simulado significa la inexistencia de las consecuencias que se generarían a partir de la ejecución de un número ilimitado de procedimientos en los que se intente afectar la confidencialidad, integridad y disponibilidad de la información. Al mismo tiempo se obtiene experiencia y no se generarían problemas de lo que por aprender, probar o investigar podría iniciar una situación incontrolable en un ambiente real, como por ejemplo, dejar toda una línea de producción sin servicio, borrar copias de seguridad críticas o alterar la configuración de servidores.

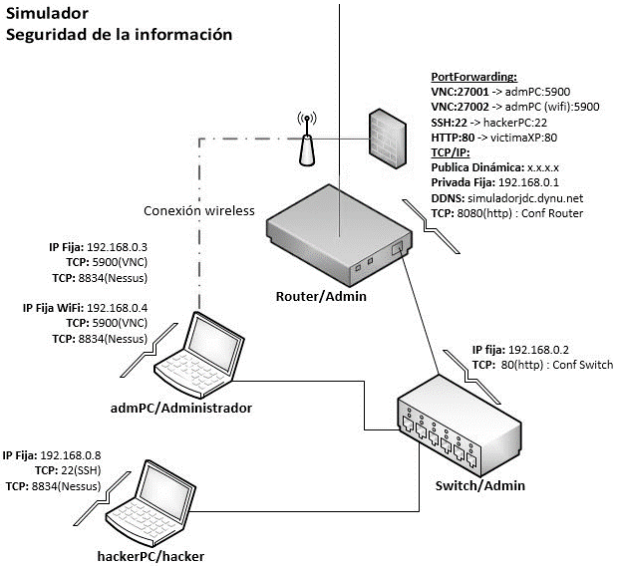
El proyecto se limita a ejecutar un test de intrusión donde se comprometa la seguridad de un sistema simulado intencionalmente vulnerable, donde los servicios están limitados a un número mínimo de clientes o estaciones de trabajo que no superen dos o tres personas. Sin embargo, el proyecto puede ser adaptado a otro tipo de actividades relacionadas con seguridad informática. Por ejemplo, se podría simular un ataque de denegación de servicio en tiempo real, para que el usuario del simulador se encargue de neutralizarlo. Se puede adaptar el ambiente simulado para recrear una escena de un ataque exitoso, y así realizar un análisis forense real para obtener evidencia en un caso de delito informático. Se puede adaptar el ambiente simulado para probar el nivel de daño de un malware, efectividad de un troyano o analizar sistemas actuales para descubrir nuevas vulnerabilidades de día cero que no hayan sido registradas en ninguna base de datos, o que no existan parches de seguridad para las mismas.

II. Metodología

A continuación, se muestran las etapas que se desarrollaron para la investigación.

A. Diseño y Construcción de Topología Física de Red de Datos

Se tiene un módem conectado a internet. Este módem es configurado para hacer redireccionamiento de puertos TCP/IP [1]. De esta manera, los usuarios en la red WAN pueden conectarse al simulador con solo saber su dirección IPv4 pública. La inclusión de un *switch* garantiza estabilidad, ya que en este se pueden activar o desactivar máquinas en el simulador a través de la apertura o cierre de sus interfaces Ethernet. La Fig. 1 muestra la topología de red física del simulador. En esta, se muestra los dispositivos de red, clientes y servicios que componen el simulador, y cómo se comunican entre sí.



Fuente: El Autor, 2015.

Fig. 1. Topología de red física del simulador.

Con los dispositivos de red y clientes conectados, se procede a configurar los servicios de los mismos para que sean vulnerables.

B. Desarrollo de Sistema Vulnerable con Servicios y Topología de Red Activos

Los servicios en el simulador van asociados a puertos TCP/IP. Los servicios trabajan en sistemas operativos, los cuales también están intencionalmente vulnerables. Por ejemplo, el sistema operativo en el que se ejecuta el test de intrusión es Windows XP SP2. Este sistema dejó de recibir actualizaciones de seguridad hace años, lo cual lo hace bastante vulnerable. Estos servicios recaen en una red LAN y algunos de estos tendrán acceso desde la red WAN, por ejemplo, el servicio HTTP sirve una web vulnerable para ejecutar diversos ataques como inyección SQL, XSS, entre otros. La Fig. 2 muestra los servicios presentes en el simulador para su funcionamiento. El servicio HTTP está asociado al puerto 80, el cual trabaja en una máquina víctima con dirección IPv4 192. 168. 0. 6. Con los servicios vulnerables configurados, se procede a definir los roles de los usuarios que interactúan en el simulador.

Port Forwarding

Internal			External							
IP Address	Start Port	End Port	IP Address	Start Port	End Port	Prot	Description	Enabled		Remove All
192.168.0.6	80	80	0.0.0.0	80	80	TCP	HTTP Server	Yes	Edit	Remove
192.168.0.3	5900	5900	0.0.0.0	27001	27001	BOTH	VNC	Yes	Edit	Remove
192.168.0.4	5900	5900	0.0.0.0	27002	27002	BOTH	VNC-Wireless	Yes	Edit	Remove
192.168.0.8	22	22	0.0.0.0	22	22	BOTH	SSH	Yes	Edit	Remove

Fuente: El Autor, 2015.

Fig. 2. Re-direccionamiento de puertos TCP/IP.

C. Definición de Roles de Usuarios en el Simulador

Básicamente hay dos máquinas. Una máquina administradora y una máquina para el usuario de nombre hackerPC. La máquina administradora es usada por el administrador del sistema. El rol del administrador del simulador es activar o desactivar el sistema, garantizar la estabilidad del mismo y guiar al usuario en capacitación. El rol del usuario es el de una persona en capacitación. El administrador tiene acceso a la máquina real, a los servicios vulnerables y a la configuración del *switch* y del módem para redireccionamiento de puertos. El usuario en capacitación, identificado en el sistema con nombre "hacker", solo tiene acceso a la máquina hackerPC. Esta máquina está configurada exclusivamente con las herramientas de hacking necesarias para que el usuario en capacitación lleve a cabo un test de intrusión exitoso. Finalmente, con el simulador activo y listo para ser usado por los usuarios: administrador y hacker, se procede a realizar un test de intrusión en el mismo para medir su estabilidad y comportamiento.

D. Test de Intrusión y Análisis de Resultados

Con el simulador activo, los usuarios realizan tres procedimientos muy comunes en la ejecución de un test de intrusión. Estos procedimientos son seleccionados a partir del modelo DragonJar [2]. A partir del modelo, los procedimientos a ejecutar son: enumeración, análisis de vulnerabilidades y explotación. El simulador es evaluado a partir de la adquisición y cantidad de conocimientos que el usuario obtenga al realizar los tres procedimientos, y también es evaluado midiendo su estabilidad y comportamiento al estar sometido constantemente a las diferentes actividades que el usuario en capacitación y el administrador del sistema realicen sobre el mismo. Después de finalizado el test de intrusión, se analizan los resultados de las actividades realizadas desde los aspectos fundamentales de la

seguridad, como son la confidencialidad, integridad y disponibilidad de la información.

III. Resultados

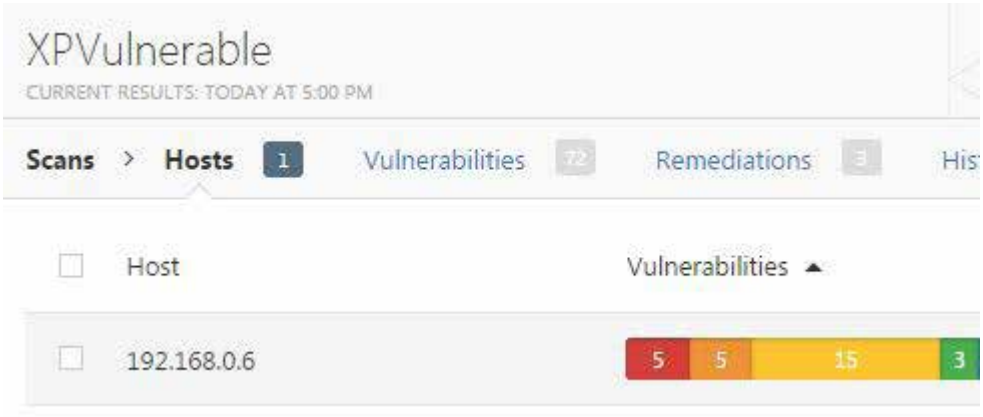
Los resultados dan a conocer el comportamiento del simulador a través de la ejecución de ataques a máquinas víctimas intencionalmente vulnerables, para que los usuarios practiquen y obtengan conocimientos en seguridad informática en un sistema controlado por un administrador. Los resultados se describen a partir de tres fases comunes en el desarrollo de un test de intrusión, como son enumeración, análisis de vulnerabilidades y explotación.

A. Enumeración

En esta fase, el usuario hacker identifica todos los dispositivos presentes en el sistema. Esto se hace a partir de la identificación del direccionamiento IPv4 de cada uno de los dispositivos conectados en la red. Para la enumeración, se usa la herramienta NMAP [3]. Esta herramienta es utilizada por muchos informáticos en el proceso de enumeración. Antes de usar esta herramienta, es necesario saber en qué red se encuentra el usuario hacker, por tanto a partir de los resultados del comando ifconfig se usa la herramienta NMAP, la cual también puede identificar los servicios activos presentes en las máquinas del simulador. Estos servicios están asociados a puertos TCP/IP, ya sea que estén abiertos, filtrados o cerrados. Los resultados que arroja NMAP dependen de la configuración del administrador sobre el sistema y dependen también de los conocimientos del usuario hacker en el simulador. En este procedimiento, el simulador se comporta establemente. Esto se debe a que NMAP funciona en modo consola y, por ende, no es mucha la carga de datos a transmitir en internet. Es necesario tener en cuenta que, el procesamiento lo tienen las máquinas locales y que esto no influye en el comportamiento de la red, si el procesamiento se aplicara en máquinas remotas, este sería aún más lento, ya que no solo se tendrían que enviar los resultados de dicho procesamientos sino también se tendrían que enviar datos relacionados con la ejecución del procesamiento en sí mismo. Los resultados son correctos y el usuario está listo para seguir con el procedimiento de análisis de vulnerabilidades.

B. Análisis de Vulnerabilidades

Después de identificar los objetivos, se analizan sus vulnerabilidades a partir de sus servicios. Para esta fase, es necesario usar la herramienta NESSUS [4]. Esta herramienta habilita un puerto en la máquina del usuario para poder realizar peticiones de escaneo de vulnerabilidades en una base de datos que es actualizada regularmente. El usuario trabaja con la herramienta a través de cualquier navegador web. El link de acceso a la herramienta Nessus es https://localhost:8843. Los resultados en esta fase también son exitosos, ya que se desarrolla un escaneo de vulnerabilidades sin inconvenientes. En este trabajo se ejecutó un análisis contra la máquina vulnerable XP, sin embargo existe otra máquina tipo LINUX también vulnerable de nombre metasploitable [5], a la cual también se le puede ejecutar el análisis y los resultados también son efectivos. Nessus se caracteriza por tener un control fácil de creación de políticas donde se activan o desactivan *plugins* dependiendo del objetivo a analizar. La Fig. 3 muestra un ejemplo de resultados arrojados por Nessus cuando se ejecuta un análisis a la máquina víctima XPVulnerable. Nessus demuestra ser una herramienta muy útil, ya que en este caso arroja cinco vulnerabilidades críticas, cinco vulnerabilidades altas, quince vulnerabilidades medias y tres vulnerabilidades bajas.



Fuente: El Autor, 2015.

Fig. 3. Resultados arrojados por Nessus.

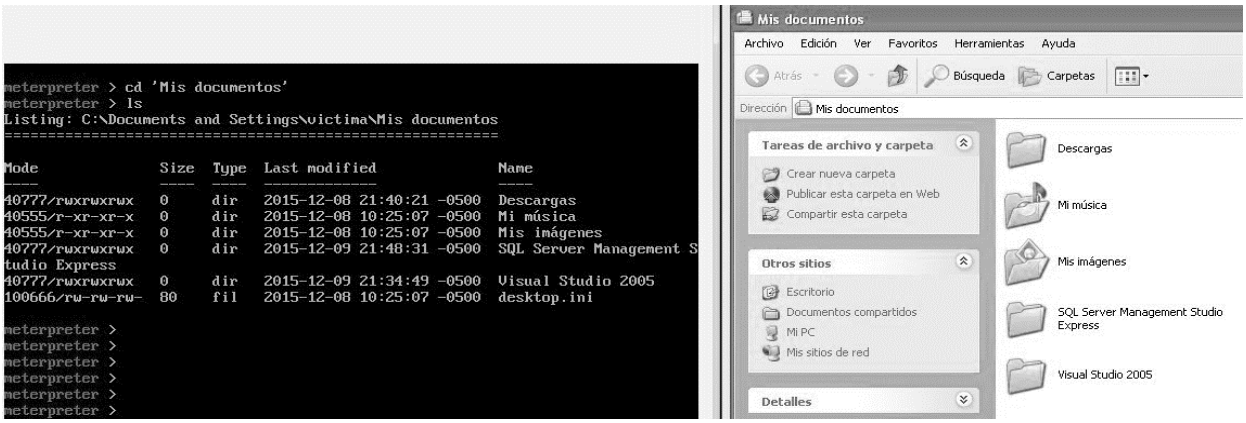
C. Explotación

Con el análisis de vulnerabilidades terminado, ahora se usa el [6] framework Metasploit para la fase de explotación. Para iniciar Metasploit es recomendable, pero no obligatorio, tener corriendo la base de datos postgresql en la máquina hackerPC. El puerto TCP abierto de postgresql es 5432. La consola de metasploit inicia con el comando msfconsole. Después de iniciar Metasploit, es necesario seleccionar el exploit reportado en la fase de análisis de vulnerabilidades. En este caso, el exploit es MS08-067. Se ajustan otras opciones como el código payload, el cual establece lo que se quiere hacer al explotar esta vulnerabilidad. La Fig. 4 muestra el exploit configurado para iniciar el ataque. Se ejecuta el código exploit, el cual es exitoso, y finalmente se tiene una consola remota de la víctima sin que esta se dé cuenta. La Fig. 5 muestra los resultados de ejecutar el exploit sin inconvenientes.

```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.0.6      yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LPORT     4444             yes       The listen port
   RHOST     192.168.0.6      no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Fuente: El Autor, 2015

Fig. 4. Configuración de Metasploit.



Fuente: El Autor, 2015.

Fig. 5. Vista de la máquina HackerPC y pantalla de la máquina Víctima.

D. Seguridad de la Información

Con el test de intrusión finalizado, el usuario detalla que se han violado tres aspectos de seguridad, como lo son: la confidencialidad, integridad y disponibilidad de la información de un sistema. Se viola la confidencialidad cuando el usuario ingresa a la máquina víctima sin autorización. El usuario hacker remotamente navegó entre directorios y obtuvo información confidencial alojada en el sistema. El usuario también obtuvo imágenes del escritorio remoto en el sistema. Se violó la integridad del sistema, ya que el usuario cargó y modificó documentos. También tuvo acceso al registro del sistema operativo y modificó algunos valores. El usuario cargó al sistema un *malware* que se inicia automáticamente con el sistema operativo y que deja una puerta trasera a través de la apertura no autorizada de un puerto TCP/IP. El acceso a esta puerta trasera, sirve una consola remota de la máquina víctima sin necesidad de explotar alguna vulnerabilidad. Por último, la disponibilidad del sistema también resulta afectada, ya que el usuario hacker apaga la máquina víctima sin autorización y, por consiguiente, los servicios alojados en esta máquina también son desactivados, como por ejemplo el servicio http que servía una página web en el sistema. Esta página web contiene una aplicación, la cual maneja un sistema de información de tipo de procesamiento de transacciones [7]. La aplicación web está conectada a una base de datos Microsoft SQL Server, y por medio de una interfaz se recopila, almacena y altera la información relacionada con el acceso de usuarios víctimas. En la Fig. 6 se muestra la aplicación web intencionalmente vulnerable del sistema.

Fuente: El Autor, 2015.

Fig. 6. Aplicación Web Vulnerable.

IV. Discusión

A nivel general, los resultados en el comportamiento del simulador demuestran estabilidad y continuidad. Se analizan los resultados de las tres fases que se usan frecuentemente en un test de intrusión. Al realizar ataques reales hacia máquinas víctimas, la obtención de buenos resultados radica en si es posible desarrollar el ataque y lograr que el usuario hacker obtenga y refuerce sus conocimientos sobre seguridad informática y, específicamente, obtenga y refuerce sus conocimientos de cómo realizar un test de intrusión. Por tanto, la curva de aprendizaje del usuario depende de lo siguiente: el tiempo que le tome aprender manipular el simulador y el nivel de experticia que obtenga en manipular las herramientas para desarrollar un test de intrusión. Se analiza el nivel de conocimientos de los usuarios antes y después de usar el simulador, por tanto se selecciona una persona con conocimientos limitados en seguridad informática.

El simulador funciona establemente con una ligera, pero no permanente demora al momento de ser usado por el usuario hacker. Esto se debe a que la red de datos del simulador posee una banda ancha con salida a internet de 5Mbps. El consumo de banda ancha es considerable, ya que cuando se usa el simulador, están siendo cargados y descargados a internet los datos equivalentes a un escritorio remoto junto con la operatividad de unas máquinas virtuales, y al mismo tiempo son cargados y descargados lo equivalente en datos a un servicio SSH de una consola remota con servicio XORG [8] para un aplicativo web donde se realizan análisis de vulnerabilidades. Pese a una limitada demora, es posible realizar las tres fases del test de intrusión en un equipo remoto siempre y cuando la banda ancha supere 5Mbps y no haya ninguna clase de restricción en el equipo remoto cliente. Se discute la posibilidad de buscar más eficiencia en el simulador, para esto se establece los siguiente: “entre menos interfaz gráfica sea usada, más rápido será el comportamiento del simulador”. De cualquier forma, al final esto no es un impedimento para las actividades que se realizan, ya que muchas de las herramientas usadas se trabajan sin interfaz gráfica, como el caso de NMAP o Metasploit.

En este trabajo, el usuario hacker solo usa interfaz gráfica para trabajar en un navegador web y trabajar en una consola gráfica. Sin embargo, el programa [9] screen, al igual que una consola gráfica, permite dividir la pantalla de la consola en diferentes partes sin necesidad de usar una interfaz gráfica. El dividir la pantalla en varias partes es indispensable para que el usuario hacker pueda tener una mejor comprensión de su trabajo en el simulador. El uso del programa screen sin interfaz gráfica ahorra consumo de banda ancha y, por ende, el comportamiento del simulador es más rápido. Se hicieron pruebas de aprendizaje con distintos usuarios, cuyos conocimientos en seguridad informática eran limitados y se obtuvieron buenos resultados. Los usuarios aprendieron a usar NMAP para los procesos de enumeración y, al mismo tiempo, aplicaron sus conocimientos en redes de datos con el protocolo TCP/IP. Los usuarios aprendieron sobre puertos TCP/UDP abiertos relacionados con servicios, los cuales dependiendo de su versión pueden llegar a ser vulnerables. Por medio de fase Análisis de vulnerabilidades, los usuarios aprendieron sobre servicios web locales y clientes web. Esta fase les permitió conocer las diferentes metodologías y bases de datos sobre vulnerabilidades encontradas que pueden ser explotadas con mucha facilidad. Aprendieron sobre la magnitud que representaría no aplicar parches de seguridad en las aplicaciones, así como también el riesgo que representa trabajar sobre entornos desactualizados y, peor aún, que no tengan soporte. Finalmente, en la fase de explotación, los usuarios se dieron cuenta que al usar el framework Metasploit es posible atacar fácilmente una víctima, ya sea obteniendo una consola remota sin autorización o simplemente causar un ataque de denegación de servicio. Los usuarios, por medio de la práctica, se dan cuenta de la diferencia entre un código exploit y un código payload. Todo lo anterior sin correr riesgos de llegar a causar una interrupción en los servicios de la organización, o causar una pérdida irrecuperable de información crítica.

V. Conclusiones

Se concluye que el simulador se comporta de forma estable, mientras el acceso remoto sea constante y no presente restricciones por parte del pro-

veedor de servicios. Las aplicaciones de consola como NMAP o Metasploit no consumen recursos gráficos y, por ende, la carga y descarga de datos en internet no es lo suficiente como para percibir lentitud en el simulador. De todas formas, se nota cierta eficiencia en lugares cuya banda ancha supere los 5Mbps. El simulador se ajusta a todas las actividades que realice el usuario hacker, sin que su estabilidad se vea comprometida por el tiempo que el usuario hacker lo requiera. Por el contrario, el usuario administrador maneja un escritorio remoto el cual le obliga a lidiar con mucho tráfico en internet. Esto al final no es problema, ya que el administrador del sistema tiene un rol pasivo, es decir solo hará uso del simulador cuando su presencia sea requerida, como por ejemplo en configuraciones del sistema o activación y desactivación de interfaces, la otra parte del tiempo el administrador solo monitorea las actividades que el usuario hacker desarrollará constantemente y que se ven reflejadas en las máquinas víctimas.

El simulador se diseña pensando en ser adaptable, por tanto en futuras investigaciones sobre el mismo, se propone el uso de otras herramientas para hacer un test de intrusión y analizar su comportamiento en el simulador. Pese a que el desarrollo de la investigación se basó en simular una prueba de intrusión, el simulador puede ser adaptado a cualquier otra actividad relacionada con seguridad de la información. El proyecto permite ser adaptado para practicar actividades relacionadas con análisis forense en una máquina que haya sido víctima de un ataque exitoso. El proyecto también permite ser adaptado para ejecutar una guía, paso a paso, de cómo configurar un *firewall* y después ponerlo a prueba. Se puede probar el impacto que generaría un *malware* en un sistema. Futuros proyectos relacionados con seguridad de la información que necesitas en un ambiente controlado, pueden hacer uso del simulador aprovechando las ventajas descritas en el presente artículo.

Referencias

[1] B. A. Forouzan, Transmisión de datos y redes de comunicaciones, 2nd ed. España: McGraw-Hill, 2002, pp. 41-55.

[2] J. A. Restrepo, Introducción al Pentesting – Metodología DragonJar, 2015. [Online]. Available: http://www.dragonjar.education/file/112725

[3] L. Gordon, Nmap ("Network Mapper") Security Scanner, 2015. [Online]. Available: https://nmap.org/book/man.html

[4] Tenable Network Security, Nessus Vulnerability Assessment Solution, 2015. [Online]. Available: http: //static.tenable.com/documentation/nessus_6.4_user_guide.pdf

[5] A. Matthews, T. Giakouminakis, and C. Loder, RAPID7Community, Metasploitable 2 Exploitability Guide, 2013. [Online]. Available: https://community.rapid7.com/docs/DOC-1875

[6] A. Matthews, T. Giakouminakis, and C. Loder, RAPID7 Community, Metasploit Community User Guide, Release 4.9, 2014. [Online]. Available: https://community.rapid7.com/servlet/JiveServlet/downloadBody/1563-102-14-6126/community-user-guide.pdf

[7] E. Borja, Sistemas de Información, 2014. [Online]. Available: http://edwinborjalopez.blogspot.com.co/2014/06/blog-post.html

[8] D. Dawes, and L. Glenn, Xorg – X11R6 X server, 2015. [Online]. Available: ftp://www.x.org/pub/X11R6. 8. 2/doc/Xorg.1.html

[9] Free Software Fundation, Inc, Screen, the virtual terminal manager, 2003. [Online]. Available: https://www.gnu.org/software/screen/manual/screen.pdf