Actas de las XXII Jenui. Almería, 6-8 de julio 2016
ISBN: 978-84-16642-30-4
Páginas: 335-342

DSBOX: herramienta docente para el diseño y simulación de entornos de red virtualizados

Francisco J. Ribadas-Pena, Rubén Anido-Bande, Víctor M. Darriba-Bilbao
Departamento de Informática, Universidade de Vigo
Escola Superior de Enxeñería Informática
Edificio Politécnico, Campus As Lagoas, S/N. 32004 Ourense
ribadas@uvigo.es, rabande@esei.uvigo.es, darriba@uvigo.es

Resumen

En este trabajo se describe DSBOX, una herramienta gráfica para el diseño y simulación empleando virtualización de pequeñas redes de computadores. Se describe el contexto que ha dado origen a esta herramienta y se detalla su arquitectura y las optimizaciones que se han realizado con el fin de dotar a los alumnos de una herramienta operativa que les facilite la realización de prácticas relacionadas con la seguridad informática y la administración de sistemas. Se presenta también un repositorio de actividades prácticas que hacen uso de la herramienta desarrollada.

Abstract

This paper describes DSBOX, a graphical tool for the design and simulation of small computer networks, employing virtualization software. We describe the context where this tool was created and its architecture and the optimizations that have been made in order to provide our students with a tool giving support to teaching activities in computer security and systems management. Finally, a repository with practical exercises using this tool is also described.

Palabras clave

virtualización, recurso docente, simulación de redes, seguridad, administración de sistemas

1. Introducción

En multitud de contextos dentro de la docencia de Ingeniería Informática es habitual el uso de herramientas de virtualización, bien como un modo sencillo de distribuir entornos de trabajo preconfigurados listos para su uso o como un mecanismo para facilitar la realización de entregas por parte de los alumnos. También es frecuente el uso de virtualización en la docencia de materias relacionas con la administración de redes y sistemas operativos o en ámbitos como la docencia de seguridad informática. Es precisamente en estos contextos donde estas técnicas ofrecen todo su potencial docente al permitir contar con pequeños laboratorios de uso personal donde definir redes virtualizadas y experimentar con herramientas y configuraciones similares a las utilizadas en el mundo real.

En este trabajo se describe una herramienta docente denominada DSBOX que ofrece una interfaz multiplataforma para el diseño de redes virtualizadas y su posterior puesta en ejecución, haciendo uso, en su versión actual, de la plataforma de virtualización VirtualBox (http://www.virtualbox.org/). Desde el punto de vista técnico, se trata de una aplicación de escritorio desarrollada en Java donde el usuario puede especificar los equipos que conforman la red simulada y definir las conexiones entre ellos. Una vez definida la red, esta puede ser puesta en ejecución desplegando las respectivas máquinas virtuales huésped sobre el motor de virtualización de VirtualBox, que se encarga a su vez de simular los dispositivos de red que interconectan a los equipos simulados. La herramienta desarrollada permite incorporar cualquier imagen compatible con VirtualBox, tanto las que forman parte de la distribución por defecto de DSBOX, como cualquier otra imagen que se desee integrar en la redes diseñadas. El esquema empleado trata de sacar provecho de las funcionalidades que ofrece el motor de virtualización VirtualBox para simplificar la configuración de las máquinas virtuales por parte de los alumnos, empleando para ello los mecanismos de comunicación con las máquinas huésped que ofrece esta herramienta. También se hace uso de imágenes diferenciales para reducir los requisitos de almacenamiento y hacer posible la exportación e importación del estado de las simulaciones en curso.

Junto a la descripción de esta herramienta se incluye la presentación de un repositorio de actividades guiadas relacionadas con aspectos de seguridad informática y administración de sistemas. La herramienta DSBOX se ha desarrollado en el marco de la materia "Seguridad en Sistemas Informáticos" y este repositorio de actividades se ha alimentado con las prácticas de laboratorio propuestas en cursos precedentes, habiéndolas adaptado para emplear las imágenes base disponibles en la distribución por defecto de DSBOX y generado los correspondientes ficheros de configuración con las especificaciones de las redes y de las máquinas virtuales necesarias.

El trabajo presentado se organiza en tres grandes partes. En primer lugar se describe el contexto en el que ha surgido el desarrollo de esta herramienta, detallando las necesidades específicas derivadas de la docencia práctica en seguridad informática y administración de sistemas que una herramienta como DSBOX debe abordar, junto con una revisión de las herramientas similares actualmente existentes. La segunda parte del trabajo detalla el diseño de DSBOX y las estrategias empleadas para sacar provecho de las funcionalidades específicas ofrecidas por el entorno de virtualización empleado para hacer de DSBOX una herramienta práctica y productiva desde el punto de vista del alumno. Se describen también las funcionalidades y opciones que ofrece la versión actual de la herramienta. Finalmente, en la última parte del trabajo se presenta el repositorio de actividades prácticas disponible para su uso con esta herramienta de simulación, la estructura del mismo y la filosofía seguida en el diseño de este conjunto de actividades.

2. Contexto

La herramienta descrita en este trabajo, DSBOX, surge como una evolución del esquema empleado por los autores en la docencia práctica de la materia "Seguridad en Sistemas Informáticos" [3], donde se empleaban redes de máquinas virtuales para proponer al alumnado entornos de trabajo cercanos a los reales. El uso de máquinas virtuales para la construcción de laboratorios y entornos de experimentación es una práctica habitual en el campo de la seguridad informática y de la administración de sistemas, tanto a nivel profesional como en el ámbito docente [4, 6]. La flexibilidad y las ventajas que aportan estas tecnologías en este contexto son bien conocidas y eliminan o atenúan muchos de los problemas típicos a la hora de afrontar las actividades prácticas en este tipo de materias (acceso con privilegios, uso de herramientas de administración, explotación de vulnerabilidades, etc).

Trabajar con máquinas virtuales permite experimentar en entornos muy similares a los reales, con la ventaja del aislamiento, la facilidad de recuperación ante errores o problemas y la posibilidad de replicar, comunicar y compartir los experimentos realizados simplemente mediante el intercambio de las imágenes utilizadas. Desde el punto de vista de la docencia, estos entornos virtualizados proporcionan numerosas ventajas [5] y existe una variada bibliografía [1, 2] sobre su empleo en la docencia de Sistemas Operativos, Redes de Computadoras o Seguridad Informática. Las máquinas virtuales son fáciles de crear y son muy seguras, ya que es posible aislarlas totalmente de la red de docencia y del exterior. La mayor parte de las herramientas de virtualización actuales contemplan la posibilidad de definir redes virtualizadas de complejidad arbitraria sobre las que podrán trabajar los alumnos. Adicionalmente, se consigue maximizar el aprovechamiento del tiempo de prácticas del alumno, ofreciéndole un entorno preconfigurado y adaptado específicamente a cada una de las actividades prácticas que se le propongan, reduciendo el tiempo y el trabajo extra necesario para su puesta en marcha.

En este contexto, con el desarrollo de DSBOX, se ha pretendido ofrecer al alumno, dentro de una única máquina (bien sea un equipo de un laboratorio compartido o bien su propio equipo personal), un entorno virtual a modo de "laboratorio personal". Se pretende que los alumnos trabajen con múltiples computadores conectados formando pequeñas redes, listos para ser usados, que contarán con una colección de herramientas preinstaladas y configuradas, simplificando al máximo las tareas de preparación y agilizando la puesta en marcha de las actividades prácticas propuestas.

2.1. Herramientas similares

En el uso docente de herramientas de virtualización nos encontramos, por una parte, con el empleo directo de soluciones convencionales como QEMU, KVM (Kernel Virtual Machine), Xen, VMware, VirtualBox, herramientas basadas en éstas como Vagrant o incluso el gestor de contenedores ligeros Docker. En esto caso es uso más habitual se da en la realización de pruebas o experimentos ocasionales o como una forma cómoda de distribuir entornos preconfigurados y/o gestionar entregables por parte de los alumnos.

Por otra parte, nos encontramos también con herramientas más sofisticadas que se abstraen del software de virtualización subyancente, ofreciendo mecanismos simplificados para la configuración del entorno virtualizado mediante documentos XML o incluso interfaces gráficas propias, como es el caso de DSBOX. Dentro de este tipo de herramientas tenemos ejemplos como Netkit (`http://wiki.netkit.org/`) [1], basado

[1] Existe un proyecto derivado más reciente, Netkit-NG, `https://netkit-ng.github.io/`

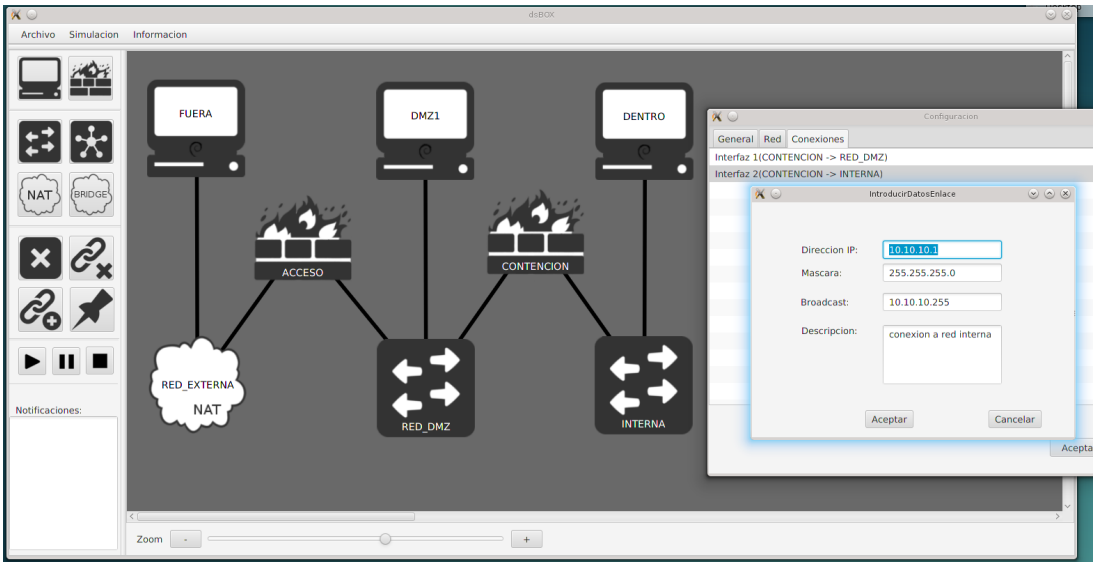

Figura 1: Visión general de DSBOX: escenario "*Uso de Shorewall: DMZ con doble firewall*" y configuración de conexiones en el cortafuegos *contención*.

en la plataforma de virtualización a nivel de sistema operativo *User Mode Linux* (UML); o como *Virtual Networks over LinuX* (VNX) (`http://web.dit.upm.es/vnxwiki/`), que a su vez es una evolución de VNUML(*Virtual Networks over User Mode Linux*), también basada en UML, y que en VNX se ha abstraído de la plataforma de virtualización concreta empleando el API de `libvirt` [2] e integrado el soporte para virtualización ligera mediante *Linux Containers* (LXC). En ambos casos, Netkit y VNX, se trata de herramientas con una clara orientación docente y los respectivos equipos de desarrollo mantienen sendos repositorios de "laboratorios" con diversos escenarios [3] [4].

Otras herramientas similares a estas serían el proyecto Marionnet (`http://www.marionnet.org/`), originado como una interfaz gráfica para Netkit, aunque sin actividad reciente; y el simulador *GNS3. Graphical Network Simulator* (`http://www.gns3.com/`) con un enfoque más generalista, que permite experimentar con dispositivos CISCO emulados mediante la herramienta Dynamips y que soporta la inclusión de máquinas virtuales basadas en VirtualBox dentro de las redes emuladas. Para más detalles, en [2] se presenta un estudio comparativo de algunas de estas herramientas y de otras similares respecto a su uso docente en la enseñanza de redes de computadoras.

[2] `http://libvirt.org/`

[3] Laboratorios Netkit: `http://wiki.netkit.org/index.php/Labs_Official`

[4] Laboratorios VNX: `http://web.dit.upm.es/vnxwiki/index.php/Allexamples`

3. Diseño y funcionalidades de DSBOX

Nuestra propuesta está a medio camino entre los dos grandes tipos de aproximaciones descritas en la sección anterior. En el caso de DSBOX se hace uso de una herramienta de virtualización convencional, VirtualBox, sobre la cual la interfaz gráfica desarrollada simplifica la definición de los escenarios a emplear, siendo posible, no obstante, el uso de las máquinas y redes virtuales definidas directamente desde VirtualBox, sin hacer uso de nuestra herramienta.

VirtualBox es una plataforma de virtualización completa para arquitecturas x86 y AMD64, distribuida en su versión básica bajo licencia GNU *General Public License* v2 y disponible para GNU/Linux, MS Windows, Mac OS y Solaris. Permite la ejecución de una amplia variedad de sistemas operativos en las máquinas virtuales huésped sin requerir privilegios de administrador y permitiendo, si así se requiere, aislamiento total respecto al equipo anfitrión y la red de docencia. Cuenta con una extensa documentación y una amplia comunidad de usuarios. Desde el punto de vista del diseño de DSBOX, las principales ventajas que aporta esta herramienta y que han llevado a su empleo son, por orden de importancia, las siguientes:

1. Soporte multiplataforma. Se trata de una herramienta disponible para los sistemas operativos y arquitecturas más usuales y que no impone restricciones especiales respecto al tipo de equipo anfitrión a utilizar. La principal limitación práctica es la cantidad de memoria RAM disponible.
2. Soporta la emulación de múltiples interfaces de

red en cada máquina virtual, que se pueden interconectar con las de otros huéspedes o con el anfitrión, formando redes virtualizadas.
Existen cuatro modos básicos: *(a)* modo *NAT*, que permite conexiones salientes desde los huéspedes empleando NAT (*Network Address Translation*) gestionada por el motor de VirtualBox; *(b)* modo *Bridged*, que permite el acceso directo del huésped a las redes donde resida el anfitrión, configurando la correspondiente interfaz de red en modo puente; *(c)* modo *Host Only*, que propociona al huésped una interfaz de red a través de la que puede comunicarse con el anfitrión; y *(d)* modo *Internal Network*, que proporciona un mecanismo para que distintas máquinas huésped compartan una red virtualizada común, aislada del exterior y del propio anfitrión.
En las simulaciones gestionadas por DSBOX, se utilizan fundamentalmente conexiones *Internal Network* para definir redes locales arbitrarias según las necesidades del escenario trabajado en cada actividad. Cada una de estas redes virtualizadas conforma un dominio de colisión, a modo de concentrador o *hub* virtual, donde se conectan las diferentes instancias de los equipos huésped.

3. Soporta la compartición de las imágenes de discos virtuales entre diferentes máquinas huésped y permite la creación de imágenes diferenciales. De este modo, es posible definir una única imagen base que será reutilizada por todas las máquinas virtuales que conforman un escenario dado, ahorrando espacio de disco y minimizando los tiempos de descarga de las imágenes utilizadas. Asimismo, es posible distribuir imágenes diferenciales que añadan modificaciones sobre las imágenes iniciales, permitiendo una actualización sencilla de los huéspedes utilizados.
4. Ofrece mecanismos de control de las máquinas virtuales muy potentes y flexibles, más allá de su interfaz gráfica. En concreto, la herramienta de línea de comandos `VBoxManage` permite configurar la totalidad de los parámetros relativos a las máquinas virtuales, así como controlar su ejecución, haciendo posible la definición y control de estas máquinas desde *shell scripts* fáciles de distribuir y ampliar.
   VirtualBox también expone sus funcionalidades a través de diversas APIs disponibles para distintos lenguajes de programación.
5. Permite la comunicación entre anfitrión y huéspedes mediante el intercambio de *Guest Properties*. Se trata de pares atributo-valor que son establecidos por la máquina anfitrión, bien desde la herramienta `VBoxManage` o desde alguna de las APIs ofrecidas por VirtualBox. Los valores vinculados a estos atributos que pueden ser recuperados desde las máquinas huésped que tengan instalado el software denominado VirtualBox *Guest Additions*, que proporciona una colección de drivers nativos, servicios y ejecutables, entre ellos el comando `VBoxControl` que permite consultar los valores de estas *Guest Properties*.
   En nuestro caso, este mecanismo se utilizará para inyectar en los equipos huésped los parámetros de configuración especificados por los usuarios en los correspondientes paneles de la interfaz de DSBOX. De este modo, es posible configurar, desde el exterior de las máquinas virtuales, aspectos como las conexiones de red, rutas por defecto, nombres y direcciones de máquinas conocidas, etc.

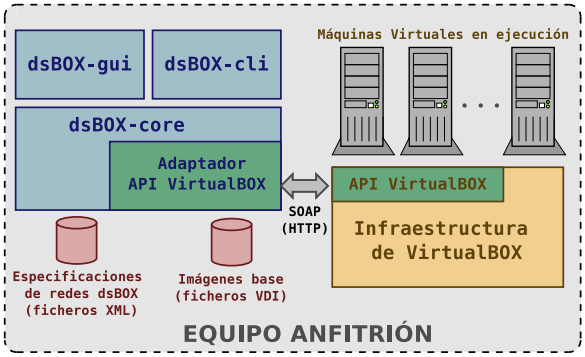

Figura 2: Arquitectura y componentes de DSBOX.

3.1. Interfaz gráfica DSBOX

El objetivo primordial de DSBOX es simplificar al máximo la creación de escenarios de red virtualizados por parte de los usuarios finales. Esencialmente, DSBOX proporciona un editor gráfico de redes donde se pueden vincular equipos, seleccionados a partir de una paleta con los tipos de equipos disponibles, con dispositivos de red, seleccionados de entre la paleta de tipos de dispositivos soportados. Adicionalmente, es posible configurar los parámetros de red de cada una de las conexiones presentes en los equipos de la red, junto con características relativas al modo en que VirtualBox asignará recursos a las máquinas huésped, como puede ser la cantidad de memoria RAM o el porcentaje de tiempo de CPU dedicado a cada máquina huésped.

Por último, desde la propia interfaz de DSBOX también es posible lanzar y controlar la ejecución de las máquinas virtuales que constituyen el escenario simulado, iniciándolas, suspendiendo su ejecución o finalizándola. En tiempo de simulación, a cada uno de los equipos le corresponderá una máquina virtual y los distintos dispositivos de red serán simulados configurando, de forma apropiada, las conexiones de red de estas máquinas, de acuerdo a las funcionalidades que ofrez-

ca la plataforma de virtualización externa empleada.

En la figura 1 se muestra un ejemplo del tipo de escenarios que es posible diseñar con DSBOX. En concreto, este caso se corresponde con la red utilizada en la actividad "*Uso de Shorewall: DMZ con doble firewall*" del repositorio de actividades prácticas que acompaña a DSBOX. Se muestra también uno de los paneles de configuración de las conexiones de red del cortafuegos "*contención*".

En cuanto a la arquitectura de esta herramienta, cuyo código fuente está disponible en `https://github.com/repossi/dsbox2/`, se trata de una aplicación de escritorio Java, desarrollada sobre el toolkit gráfico JavaFX 8. Se ha diseñado buscando la máxima modularidad, de modo que la interfaz gráfica para el diseño de redes sea independiente de los componentes encargados de interactuar con la plataforma de virtualización empleada para la creación, configuración y control de las máquinas virtuales que implementan la simulación. En la versión actual se emplea VirtualBox como plataforma de virtualización externa, aplicando todas las optimizaciones descritas en las secciones precedentes: compartición de imágenes base mediante imágenes diferenciales, uso de *Guest Properties* para la configuración interna de los huéspedes, etc. Como se muestra en la figura 2, la interacción con VirtualBox se lleva a cabo empleando la API de servicios web disponible para esta herramienta, en concreto la versión Java basada en JAX-WS (*Java API for XML Web Services*). De este modo, realizando invocaciones SOAP (*Simple Object Access Protocol*) a los métodos publicados mediante esta API es posible crear y configurar las máquinas utilizadas y controlar su ejecución.

Por defecto, el tipo de equipos soportados por la versión actual de DSBOX emplea una imagen base común distribuida con la propia herramienta. No obstante, es posible utilizar cualquier tipo de imagen soportado por VirtualBox. En estos casos no es posible la configuración automática de las conexiones de red, salvo que en dicha imagen se cuente con *scripts* específicos que procesen adecuadamente las *Guest Properties* descritas en el cuadro 1. En lo que respecta al tipo de dispositivos de red soportados, se corresponden con *switches* o *hubs* Ethernet simulados configurando convenientemente los equipos huésped para que empleen conexiones de tipo *Internal Network*. También se soporta la simulación de dos tipos de conexiones con el exterior, utilizando los modos de conexión *bridged* y *NAT* ofrecidos por VirtualBox.

Por último, toda la información relativa a las redes diseñadas y los datos necesarios para configurar y poner en marcha las máquinas virtuales que posteriormente las simularán se almacena en documentos XML (*eXtensible Markup Language*). Esta funcionalidad hace posible la distribución, intercambio y modificación de los entornos de red diseñados.

Imagen base (`base.vdi`)
Debian 8.x (Jessie)
Entorno gráfico ligero LXDE (deshabilitado por defecto)
VirtualBox *Guest Additions* 5.0.x
Servicios (desactivados por defecto):
servidor web (Apache 2)
servidores inetd: telnetd, ftpd, fingerd
servidor BD (mysql)
servidor smtp (postfix)
servidor pop3, imap (dovecot)
servidor ssh (openSSH)
servidor ldap (openLDAP)
Herramientas de seguridad:
analizador de protocolos WIRESHARK
escaner de vulnerabilidades: OPENVAS
escaner de puertos NMAP (interfaz ZENMAP)
metasploit framework
sistema de detección de intrusiones en red SNORT
sistema de detección de intrusiones en host SAGAN
cortafuegos: netfilter/iptables
interfaces iptables: Firestarter, Shorewall
consola eventos IDS SNORBY
openVPN, openssl, easy-rsa, tinyca
otras: netcat (`nc`), hping3, iptraf, netstat-nat, tcptraceroute
Aplicaciones web vulnerables:
OWASP WebGOAT (`http://webgoat.github.io/`)
Mutillidae (`http://sourceforge.net/projects/mutillidae/`)
Damn Vulnerable Web App (`http://www.dvwa.co.uk/`)
foro PHP vulnerable (desarrollo propio)
Wordpress 1.5.1.1
Software general:
clientes telnet, ftp y ssh
navegador web modo texto *Lynx*
navegador web *Mozilla Firefox*
editores: vi, nano, jed, leafpad
Otras herramientas:
balanceo de carga basado en proxies (HAProxy)
LinuxHA (heartbeat, pacemaker)
Tamaño de la imagen: aprox. 4,7 GB (2,1 GB comprimida)

Figura 3: Descripción de la imagen base común empleada en las actividades.

3.2. Estructura de los equipos y las redes virtualizadas

En lo que respecta al uso que se hace de las tecnologías de virtualización a la hora de plantear el "laboratorio personal" ofrecido por DSBOX, el esquema empleado es una evolución del descrito por los autores en [3]. Se ha pretendido optimizar las necesidades de almacenamiento mediante el empleo de imágenes de disco diferenciales creadas a partir de una imagen base común. También se ha perseguido simplificar el proceso de puesta en marcha de las simulaciones, consiguiendo reducir al mínimo el trabajo del alumnado previo a la realización de las actividades en sí, automatizando tareas como la configuración de conexiones de red, direcciones y rutas.

En la distribución por defecto de DSBOX se dispone

CLAVE ATRIBUTO	TIPO VALOR	DESCRIPCIÓN
`/DSBOX/num_interfaces`	núm. entero	Número de dispositivos de red conectados a la MV huésped
`/DSBOX/{interface}/tipo`	cadena: `dhcp` ó `static`	Asignación de direcciones IP para `{interface}`: modo automático con DHCP ó modo manual (configuración estática)
`/DSBOX/{interface}/address`	dirección IP	Dirección IP a asignar `{interface}` en modo estático
`/DSBOX/{interface}/netmask`	máscara de red (formato `nnn.nnn.nnn.nnn`)	Máscara de red de `{interface}` en modo estático
`/DSBOX/{interface}/broadcast`	dirección IP	Dirección IP de broadcast de `{interface}` en modo estático
`/DSBOX/default_gateway`	dir. IP o nombre completo	Dirección IP o nombre de la máquina que hace el papel de puerta de enlace (ruta por defecto de la MV)
`/DSBOX/default_nameserver`	dirs. IP separadas por `","`	Direcciones IP de los servidores de nombres a utilizar por la MV
`/DSBOX/host_name`	cadena de texto	Nombre completo de la máquina virtual en su respectiva red
`/DSBOX/etc_hosts_dump`	pares `nombre:dirección` separados por `","`	Pares (nombre máquina, dir. IP) a incluir en el fichero `/etc/hosts` del huésped (permite que las MVs sean accesibles por nombre sin necesidad de un servidor DNS propio en la red virtualizada)

Cuadro 1: Parámetros configurables en las máquinas virtuales mediante *Guest Properties*.

de una imagen base de un sistema GNU/Linux que hace uso de la distribución Debian 8.0, y que cuenta una serie de herramientas preinstaladas, junto con un conjunto de servidores típicos (gestor de base de datos, servidor HTTP, etc) inicialmente deshabilitados. Los detalles de dicha imagen base se presentan en la figura 3. Esta imagen es la que se utilizará para lanzar la mayoría de los huéspedes que formarán parte de las redes virtuales diseñadas con DSBOX, para, de este modo sacar provecho del soporte de imágenes diferenciales de VirtualBox y minimizar las necesidades de almacenamiento.

Además del control de la simulación desde la interfaz DSBOX, para cada entorno diseñado con esta herramienta, el usuario puede generar un *shell script* de puesta en marcha, tanto para GNU/Linux como para MS Windows, que permite automatizar su puesta en funcionamiento desde fuera de la interfaz gráfica. Estos *scritps* hacen uso de las funcionalidades de la herramienta `VBoxManage` y replican las funcionalidades de la interfaz gráfica DSBOX, encargándose de crear y definir los parámetros de las respectivas máquinas huésped, configurándolas para utilizar réplicas diferenciales de la imagen base descrita anteriormente. La otra gran responsabilidad de estos *scripts* es definir las conexiones de red de los huéspedes en modo *Internal Network* de modo que se pueda realizar la interconexión entre huéspedes para conformar las redes virtualizadas definidas. Adicionalmente, estos *scripts* también pueden configurar los parámetros de red propios de cada huésped, como direcciones IP, máscaras de red, rutas, fichero `/etc/hosts` inicial, etc.

Para permitir la configuración automática de estos parámetros de red en los equipos huésped, tanto desde DSBOX como desde los *scripts* de puesta en marcha, se hace uso de las *Guest Properties* ofrecidas por VirtualBox. Las imágenes base que vayan a soportar la configuración automática de la red deben de incluir un *script* de arranque específico (`dsbox_network_autconfigure.pl`) y deberán de contar con una instalación funcional de las *Guest Additions* de VirtualBox. Este *script* de autoconfiguración se ejecuta durante el arranque de las máquinas virtuales y consulta los valores de las *Guest Properties* descritas en el cuadro 1 mediante el comando `VBoxControl` para, en el caso de que éstas estuvieran establecidas, invocar los correspondientes comandos de configuración de redes de la máquina huésped.

4. Repositorio de actividades

Como complemento a la herramienta de diseño de entornos virtualizados DSBOX, se ha publicado un repositorio web donde se recoge la colección de actividades y ejercicios basados en máquinas virtuales desarrollados a lo largo de estos años de actividad docente en la materia "Seguridad en Sistemas Informáticos". En este repositorio, además de la imagen base utilizada, se incluye para cada actividad el documento XML con la definición DSBOX del entorno de red utilizado, los respectivos *scripts* de puesta en marcha para GNU/Linux y MS Windows, junto con un fichero en formato *Markdown* con el código fuente del enunciado y las instrucciones proporcionadas a los alumnos para realizar cada actividad concreta.

Este repositorio está publicado en `https://github.com/repossi/repossi/` y se organiza en 4 secciones, que se corresponden aproximadamente con la estructura seguida en las actividades prácticas de las asignaturas de seguridad impartidas.

- **Seguridad en redes.** Incluye ejemplos de técnicas y herramientas de seguridad en redes, organizados en dos subsecciones: *(a) Protocolos seguros*, donde se revisan los problemas del uso de protocolos no cifrados y se presenta el funcionamiento interno de protocolos como

SEGURIDAD EN REDES: SEGURIDAD PERIMETRAL	DIFICULTAD	HERRAMIENTAS/TECNOLOGÍAS
Uso de Shorewall: DMZ con doble firewall	media	shorewall, nmap
Uso de Shorewall: DMZ con firewall de 3 interfaces	alta	shorewall, nmap
Túneles con openVPN	media	openvpn, openssl, tinyca, shorewall
DMZ con firewall de 3 interfaces empleando NETFILTER/iptables	media	iptables, nmap
SNORT: sistema de detección de intrusiones en red	media	snort, barnyard2, snorby, scapy
SAGAN: sistema de detección de intrusiones en host	media	snort, barnyard2, snorby, syslog

SEGURIDAD EN REDES: PROTOCOLOS SEGUROS	DIFICULTAD	HERRAMIENTAS/TECNOLOGÍAS
Análisis de protocolos y escaneo de puertos	baja	wireshark, nmap
Man in the middle sobre SSL en redes locales	media	ettercap, sslsniff, sslstrip

DESARROLLO SEGURO	DIFICULTAD	HERRAMIENTAS/TECNOLOGÍAS
Aplicaciones web inseguras: SQLi y XSS	baja	wordpress, webgoat, mutillidae, dvwa
Securización de aplicaciones web con mod-security	media	mod-security, wordpress, mutillidae, dvwa

ADMINISTRACIÓN DE SISTEMAS	DIFICULTAD	HERRAMIENTAS/TECNOLOGÍAS
Autenticación centralizada en GNU/Linux: OpenLDAP y PAM	alta	openldap, pam, phpldapadmin, gosa
Controlador de dominio AD sobre MS Windows Server 2008	baja	windows server, active directory, ldap
Balanceo de carga con HAproxy	baja	haproxy
Alta disponiblidad con LinuxHA (heartbeat y pacemaker)	media	linuxHA, heartbeat, pacemaker

TESTS DE INTRUSIÓN	DIFICULTAD	HERRAMIENTAS/TECNOLOGÍAS
Análisis de vulnerabilidades: openVAS	baja	nmap, openvas
Explotación de vulnerabilidades: Metasploit sobre Metasploitable2	alta	nmap, metasploit, armitage, metasploitable

Cuadro 2: Listado de actividades disponibles y herramientas empleadas.

TLS/SSL, y *(b) Protección perimetral*, con actividades relacionadas con el uso de cortafuegos y las distintas topologías disponibles para definir zonas desmilitarizadas, ejemplos de uso de herramientas de detección de intrusos, así como soluciones para la construcción de redes privadas virtuales.

- **Desarrollo seguro.** Con ejemplos de vulnerabilidades típicas en aplicaciones web. Se hace uso de aplicaciones educativas que cuentan intencionadamente con vulnerabilidades conocidas, como inyección SQL o *Cross Site Scripting*. También se presentan alternativas para mitigar estos problemas mediante el uso de "cortafuegos de nivel de aplicación" como `mod-security`.
- **Administración de sistemas.** Se revisan cuestiones relativas a la administración segura de sistemas, fundamentalmente las relacionadas con la autenticación centralizada. Se incluyen también ejercicios complementarios relacionados con balanceo de carga y con soluciones de alta disponibilidad, que si bien van más allá de lo que abarca típicamente un curso sobre seguridad, sirven para mostrar la versatilidad de nuestra propuesta al aplicarla en otros campos relacionados.
- **Tests de intrusión.** Se presentan ejemplos de uso de herramientas empleadas típicamente para dar soporte a tareas habituales en los tests de intrusión, como detectores/analizadores de vulnerabilidades o frameworks de explotación de vulnerabilidades como Metasploit.

En el cuadro 2 se detallan las actividades actualmente disponibles dentro de cada una de estas secciones, junto con una indicación de las herramientas concretas trabajadas en cada una de ellas.

Se ha intentado seguir un esquema uniforme a la hora de describir cada actividad. Así, la documentación de las actividades que actualmente forman parte del repositorio responde a la siguiente estructura:

1. Descripción general de la actividad.
   - Objetivos (y vinculación con contenidos teóricos).
   - Escenario: redes, equipos, herramientas.
   - Recursos: información de las herramientas, manuales, tutoriales externos, etc.
2. Desarrollo guiado.
   - Documento XML con la especificación DSBOX del entorno a utilizar.
   - Scripts de puesta en marcha para entornos GNU/Linux y MS Windows.
   - Pasos a seguir comentados.
3. Actividades autónomas.
   - Pruebas/experimentos a realizar.
   - Cuestionarios.
4. Descripción de entregables requeridos (opcional).

5. Uso docente y conclusiones

Si bien DSBOX aún no ha sido puesto en uso a gran escala en la docencia, en la figura 4 se presentan los resultados de las encuestas realizadas a los alumnos que durante los cursos 2013/14 y 2014/15 han utilizado

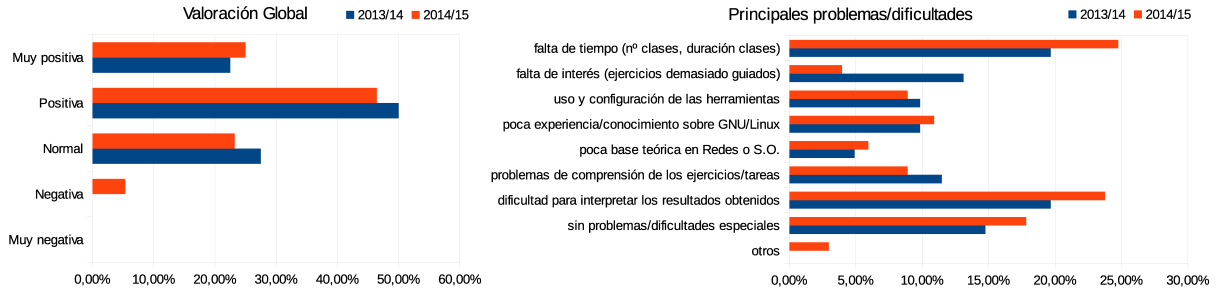

Figura 4: Evaluación global por parte del alumnado y principales problemas o dificultades.

la misma aproximación basada en máquinas virtuales en la que se basa DSBOX para llevar a cabo parte de las actividades prácticas de la materia obligatoria de 6 créditos ECTS "Seguridad en Sistemas Informáticos". Esta materia se ubica en el último curso del Grado en Ingeniería Informática impartido en nuestro centro, la Escola Superior de Enxeñería Informática, ubicadada en el Campus de Ourense de la Universidade de Vigo . En ambas ediciones del curso sólo fueron utilizadas parte de las actividades incluidas en el repositorio descrito anteriormente, sin llegar a emplearse la herramienta DSBOX, que no estaba disponible en ese momento.

Los resultados en cuanto a la valoración global del uso de entornos virtualizados son positivos, así como la percepción de los alumnos respeto a su facilidad de uso, a la adecuación de las actividades propuestas con respecto a los contenidos de la materia y al nivel de dificultad en el desarrollo de los ejercicios concretos que les fueron asignados. Nuestro objetivo es que con la introducción de DSBOX se simplifique aún más la realización de estas actividades prácticas, dado que añade funcionalidades demandadas por los alumnos como la posibilidad de exportar e importar en otros equipos las simulaciones en curso, de modo que permitirá trasladar el trabajo iniciado en el laboratorio a sus equipos personales y viceversa.

Por último, en cuanto a las futuras líneas de trabajo, el paso inmediato es integrar completamente el uso de DSBOX en la docencia práctica y explorar mejores formas de usar este tipo de actividades guiadas para fomentar el interés de los alumnos, sacando provecho de la facilidad de diseño de entornos virtualizados que nos ofrece DSBOX para proponer tareas más autónomas que permitan profundizar en el aprendizaje y asimilación de los conceptos y herramientas estudiadas. En cuanto al desarrollo futuro de DSBOX, la línea a seguir se orienta a mejorar las funcionalidades relativas al control de la ejecución de las simulaciones, la inclusión de asistentes que simplifiquen incorporación de nuevas imágenes base y nuevos tipos de equipos a la paleta de componentes del editor de redes o incorporar el uso de otros motores de virtualización, bien de forma nativa o mediante APIs independientes como `libvirt`.

Referencias

[1] Juan Antonio Gil Martínez-Abarca, Adolfo Albaladejo Blázquez, Francisco Maciá Pérez, Francisco José Mora Gimeno, Segismundo Ferrairó Pons. Entorno de red virtual para la realización de prácticas realistas de administración de sistemas operativos y redes de computadores. En *Actas de las XI Jornadas sobre la Enseñanza Universitaria de la Informática*, Jenui 2005, pp. 349 – 356. Madrid, 2005.

[2] Fernando Pereñíguez-García, Antonio Ruiz Martínez, Francisco J. Ros, Rafael Marín López y Pedro M. Ruíz Martínez. VNUML vs GNS3 en el desarrollo de laboratorios de redes virtuales. En *Actas de las XVIII Jornadas sobre la Enseñanza universitaria de la Informática*, Jenui 2012, pp.263 – 270. Ciudad Real, 2012.

[3] F.J. Ribadas-Pena, F.M Barcala-Rodríguez, V.M Darriba-Bilbao, J.Otero-Pombo. Diseño de un entorno virtualizado para la docencia práctica de Seguridad en Sistemas de Información. En *Actas de las XIV Jornadas sobre la Enseñanza Universitaria de la Informática*, Jenui 2008, pp. 325 – 332. Granada, 2008

[4] J. Son, C. Irrechukwu, P. Fitzgibbons. A Comparison of Virtual Lab Solutions for Online Cyber Security Education. *Communications of the IIMA*, Vol. 12(4), 2012

[5] Adam Vollrath, Steven Jenkins. Using virtual machines for teaching system administration. *Journal of Computing on Small Colleges (JCSC)*, Vol. 20(2), pp. 287–292, 2004.

[6] C. Willems, C. Meinel. Practical Network Security Teaching in an Online Virtual Laboratory. *Proc. of 2011 International Conference on Security and Management*, pp. 65–71, 2011.