

Enseñanza y evaluación de administración de sistemas en un entorno virtualizado y controlado

Juan Piernas Cánovas
Departamento de Ingeniería y Tecnología de Computadores
Universidad de Murcia
30100 Murcia
piernas@ditec.um.es

Resumen

Describimos nuestra experiencia en el diseño e implantación de un entorno para la docencia en administración de sistemas Linux y la correspondiente evaluación. En el caso de la docencia, el entorno creado permite a un alumno arrancar al instante una máquina virtual «limpia» o continuar con la que estaba usando tras reiniciarla o apagarla. Para la evaluación, además de usar el sistema virtualizado de docencia, se ha construido un entorno controlado que permite que dos o más alumnos, en sus correspondientes turnos, usen un mismo ordenador para realizar su examen sin que haya posibilidad de copia entre ellos, ya sea directa o indirectamente a través de otros ordenadores. Este entorno controlado de evaluación se ha creado mediante una adecuada configuración de cada ordenador y sólo se activa durante los exámenes, permaneciendo inactivo el resto del tiempo para que los alumnos puedan usar los ordenadores con normalidad. Nuestra propuesta se complementa con una herramienta para la entrega de soluciones de examen que garantiza que las entregas son realizadas sólo por alumnos presentes físicamente.

Abstract

We describe our experience designing and implementing a convenient environment to teach Linux system administration and assess students' acquired knowledge. For teaching, our environment allows a student to immediately boot a "fresh" virtual machine or boot again the one she was using after rebooting or shutting it down. For assessment, we have built a system that allows two or more students in different shifts to use the same computer without possible cheats, either directly or indirectly through remote computers. Our evaluation system has been implemented by setting up computers in such a way that it is activated only during exams, remaining inactive the rest of the time for a regular use of computers. A tool to submit exam solu-

tions complements our proposal. This tool guarantees that submissions are only performed by students really taking the examination.

Palabras clave

Administración de sistemas, recursos para docencia y evaluación, virtualización, entorno controlado de exámenes.

1. Motivación

La asignatura Introducción a los Sistemas Operativos (ISO), que se imparte en el 2º curso del Grado en Ingeniería Informática (GII) de la Facultad de Informática de la Universidad de Murcia, introduce al alumno en los principales conceptos en los que se sustentan los sistemas operativos (procesos, ficheros, gestión de memoria, entrada/salida, etc.). Mientras que la parte teórica mantiene lo que podemos llamar un esquema «tradicional» para este tipo de asignaturas, la parte práctica refuerza los conceptos introducidos en teoría a través de la programación de guiones shell y la administración de sistemas Linux. La idea es que el alumno vea las implicaciones prácticas de lo aprendido en teoría y consiga la madurez de conceptos suficiente para poder llevar a cabo con éxito una implementación real de los mismos en asignaturas posteriores.

La docencia y la evaluación de la parte práctica de ISO se realizan en laboratorios con 24 puestos (también contamos con un laboratorio de 50 puestos para exámenes). Los ordenadores disponibles son bastante heterogéneos entre laboratorios en lo que a hardware se refiere, como nos muestra el cuadro 1, y algunos de ellos están bastante limitados (p.e., los ordenadores del modelo B apenas tienen 2 GiB de memoria RAM y un procesador Pentium E5700 lanzado hace más de 7 años). En todos los casos, la LAN es una red Ethernet a 100 Mbps. Los sistemas operativos instalados en cada

Elemento	Modelo A	Modelo B Modelo C	Modelo D
Placa base	ASUS H81M-K	ASUS P5G41T-M LX	ASUS H110M-D
Procesador	Pentium G3260 a 3,30 GHz	Pentium E5700 a 3,00 GHz	Pentium G4400 a 3,30 GHz
RAM	4 GiB 1600 MHz	2 GiB 800 MHz ó 4 GiB 800 MHz	8 GiB 2133 MHz
Disco duro	WD10EZEX-00M 1TB	ST3500413AS 500GB ó ST500DM002-1BC14 500GB	WD10EZRX-00H 1TB
Tarjeta gráfica	Integrada (Intel)	Intel GMA X4500	Intel HD Graphics 510

Cuadro 1: Principales características de los ordenadores de nuestros laboratorios de prácticas. En todos ellos se encuentran instalados Windows 7, Ubuntu 16.04 y Fedora 24.

ordenador son Windows 7, Ubuntu 16.04 y Fedora 24, siendo éste último el que usamos en prácticas.

La infraestructura necesaria para enseñar programación de guiones shell no presenta grandes problemas, ya que se puede realizar en los laboratorios disponibles usando Fedora y una cuenta de usuario regular (es decir, sin permisos de administrador). Esto hace que, por lo general, la actividad de un alumno durante una sesión de prácticas de guiones shell no interfiera con el trabajo de otros alumnos que posteriormente usen el mismo ordenador.

En cambio, hasta el desarrollo del nuevo entorno de trabajo, la parte de administración de la asignatura ha provocado diversos problemas derivados del uso de la máquina real por parte de los alumnos como usuario `root`. Entre estos problemas, podemos destacar tres:

- Como los ordenadores se restauran a su estado original los viernes por la tarde de cada semana, los alumnos que tienen clases de práctica los primeros días de la siguiente semana se encuentran los ordenadores en perfecto estado, sin cambios. Sin embargo, los compañeros que tienen prácticas a lo largo de la semana se encuentran sistemas con numerosas modificaciones, lo que dificulta la realización de las tareas propuestas. El problema es especialmente serio para aquellos con sesiones de prácticas los jueves y viernes.
- Al tener acceso como administrador a la máquina real, un alumno puede realizar cambios que dejen el sistema en un estado tal que no pueda ser utilizado hasta una próxima restauración. Los cambios no tienen por qué afectar sólo a Fedora, sino que también pueden verse afectados el resto de sistemas operativos de la máquina.
- Cuando los alumnos preparan las prácticas en casa, no usan en muchos casos la misma distribución de Linux que hay en los laboratorios o ésta no está actualizada/configurada de la misma manera. Esto hace que sea difícil resolver algunas veces los problemas que nos plantean, ya que profesores y alumnos usamos entornos distintos.

Un problema común, que se da tanto en guiones shell como en administración, nos lo encontramos a la hora de realizar los exámenes. La programación de guiones shell se evalúa mediante la implementación de diversos guiones propuestos, mientras que los exámenes de administración consisten en la realización de varias tareas relacionadas con la gestión de usuarios, sistemas de ficheros, discos, etc. Tanto unos exámenes como otros se realizan delante de un ordenador. Si los alumnos usan las máquinas reales, sus soluciones y/o modificaciones quedarán en las mismas, no siendo posible entonces establecer varios turnos de examen para reducir tanto las necesidades de laboratorios como de personal de apoyo (con un único turno y unos 200 alumnos, necesitaríamos 8 laboratorios).

Como veremos en la sección 3, en el caso de los guiones shell es posible establecer una cuenta por turno, configurando el sistema adecuadamente para evitar que los alumnos compartan información a través de los directorios temporales del sistema o de sus propios directorios personales. Sin embargo, esta solución es del todo inadecuada para un examen de administración si hay un control absoluto de la máquina real por parte de los alumnos. La solución pasará por usar máquinas virtuales.

Otro problema de los exámenes realizados delante de un ordenador es la existencia de una conexión de red que podría utilizarse para transferir información entre ordenadores, obtener información de fuentes no autorizadas, etc. Una posible solución que ya se ha usado, aunque incómoda y propensa a errores, es desconectar el cable de red durante los exámenes, pero al final hay que restaurar la conexión en algún momento para que los alumnos puedan entregar los ficheros con sus soluciones (el uso de disquetes y memorias USB lo descartamos hace tiempo por su poca fiabilidad, frecuentes errores en las copias, tiempo necesario, etc.).

Finalmente, la propia entrega de los ficheros con las soluciones puede ser problemática. Si se usa un campus virtual o similar, un alumno puede subir su solución y la de otros compañeros conociendo sus creden-

ciales. Además, es posible realizar entregas no controladas desde fuera de los laboratorios. Para evitar estos casos, se hace necesario supervisar la entrega alumno por alumno, lo que demora bastante el proceso de entrega y tampoco está exento de errores. En la sección 3 veremos cómo evitar posibles copias y entregas no autorizadas de exámenes de una forma más conveniente.

2. Entorno para la docencia de administración de sistemas

En primer lugar, vamos a describir el entorno en el que se desarrollan las sesiones de prácticas de nuestra asignatura. Nos vamos a centrar en la parte de administración, ya que es la que presenta más problemas.

2.1. Máquinas virtuales para docencia

Cuando nos planteamos el uso de máquinas virtuales para la sesiones de prácticas de administración de sistemas Linux, teníamos claro que éstas debían reunir, al menos, las siguientes tres características:

1. Cada máquina virtual debía estar almacenada en la máquina real y no remotamente. De esta manera, el arranque de la máquina virtual se realizaría sin demoras y no estaríamos expuestos ni a las características de la red ni a posibles fallos de ésta.
2. De alguna forma, debía ser posible tener una copia maestra de la máquina virtual que los alumnos no pudieran modificar, pero, a la vez, permitir cambios durante el desarrollo de las prácticas. Estos cambios no debían perderse aunque la máquina virtual se apagara o reiniciara, y sólo debían desaparecer, para tener de nuevo una máquina virtual original, cuando así se deseara.
3. El rendimiento ofrecido por la máquina virtual debía ser aceptable, incluso en ordenadores con solo 2 GiB de RAM y procesadores relativamente antiguos. Además, debía incluir un entorno gráfico para poder navegar y acceder a la documentación de cada práctica y así poder trabajar usando únicamente la máquina virtual.

Teniendo en cuenta todo lo anterior, se analizaron tres posibles tecnologías de virtualización disponibles en sistemas Linux: Xen, QEMU (con hipervisor KVM) y VirtualBox. De ellas, se eligió QEMU+KVM.

El problema de Xen es que es una solución «invasiva», pues requiere de la instalación de un hipervisor que pueda tomar el control de la máquina antes siquiera de la carga de cualquier sistema operativo. Además, como otro objetivo es que los alumnos puedan usar en sus propios ordenadores la misma máquina virtual de

los laboratorios, esta solución no parece muy adecuada para alumnos de 2º curso con, generalmente, escasa experiencia en la administración de sistemas Linux.

VirtualBox tiene varias características que la hacen, a priori, una posibilidad atractiva, como son su interfaz gráfica, que facilita la gestión y ejecución de máquinas virtuales, y su carácter multiplataforma. La interfaz gráfica, no obstante, supone un consumo de recursos que puede mermar el rendimiento de aquellos ordenadores con poca RAM, aunque este problema se puede paliar usando la interfaz de línea de órdenes mediante la orden `vboxmanage`. VirtualBox, sin embargo, presenta dos problemas que nos han hecho descartar esta solución. El primero es que su rendimiento no suele ser el mejor cuando se comparaba con el de Xen y QEMU+KVM¹. El segundo es que, aunque VirtualBox permite usar instantáneas para capturar el estado de una máquina virtual en un cierto instante y recuperar ese estado posteriormente, el entorno deja de manejar correctamente las instantáneas cuando se eliminan permisos de escritura y/o se cambia la propiedad de ciertos ficheros para que sólo `root` pueda modificarlos.

Por su parte, QEMU, al igual que VirtualBox, es multiplataforma (con aceleración KVM en Linux y HAXM en Windows) y dispone (al menos en Linux) de aplicaciones gráficas, como `virt-manager`, que facilitan también el uso de las máquinas virtuales. Sin embargo, a diferencia de VirtualBox, QEMU ofrece un mejor rendimiento y, lo que es más importante, permite, de una forma muy sencilla, tener una copia maestra de la máquina virtual que sólo el superusuario puede modificar, pero sobre la que los usuarios regulares pueden trabajar, haciendo cambios de cualquier tipo, con la posibilidad de conservar esos cambios o eliminarlos en cualquier momento. Esto se consigue haciendo que el fichero que actúa como disco de la máquina virtual realmente almacene los cambios que se hacen respecto a una *backing file*, que no es otra cosa que la copia maestra de la máquina virtual. De esta manera, mientras el *backing file* puede pertenecer al administrador, en un directorio y con unos permisos que impidan su modificación directa y eliminación por parte de los usuarios regulares, el fichero que guarda los cambios puede pertenecer al usuario que utiliza la máquina virtual, siendo posible, además, que cada usuario guarde sus propios cambios. De esta forma, la misma copia maestra de la máquina virtual puede ser usada por varios usuarios a la vez, como veremos en el apartado 3.2.

Nótese que VirtualBox dispone de una solución similar a los *backing files* que es la «clonación enlazada» (*linked clone*). Sin embargo, como esta solución crea una instantánea en la máquina virtual original, la descartamos por los problemas que hemos descrito antes.

¹<https://www.phoronix.com/scan.php?page=article&item=ubuntu-1510-virt&num=1>

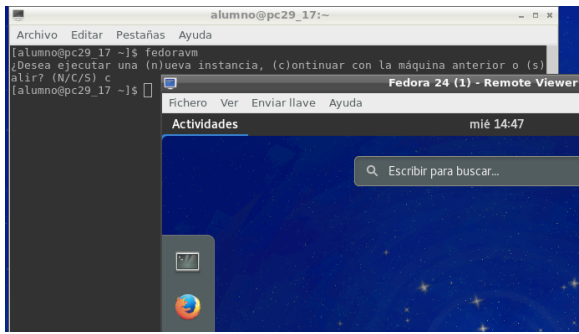


Figura 1: Captura parcial de escritorio en el que se muestra la máquina virtual, con escritorio GNOME, lanzada por `fedoravm`.

2.2. La orden `fedoravm`

Para facilitar el uso de las máquinas virtuales sin necesidad de usar una aplicación gráfica, se ha implementado la orden `fedoravm` (un guión shell de Bash) que se encarga de todos los detalles de bajo nivel a la hora de lanzar una de estas máquinas. El cuadro 2 resume las principales características de una máquina virtual creada por esta orden².

`fedoravm` también maneja los dos escenarios básicos de uso. En concreto, al ejecutar la orden, nos pregunta si queremos continuar con la máquina virtual con la que estábamos trabajando (si así era) o si queremos iniciar una máquina virtual «limpia», tal cual se encuentra en la copia maestra (ver figura 1). `fedoravm` maneja un tercer y último escenario que se da cuando el usuario cierra el visor de la máquina virtual. En este caso, si se ejecuta de nuevo `fedoravm`, detecta que hay una máquina virtual en ejecución y nos da la posibilidad de volvernos a conectar a ella o terminarla.

2.3. Ajustes para mejorar el rendimiento

Para mejorar el rendimiento de la solución propuesta, han sido necesarios algunos cambios, principalmente, la desactivación de servicios que son innecesarios en nuestro entorno (impresión, cortafuegos y virtualización dentro de la propia máquina virtual, etc.). Especialmente interesante ha sido el servicio de actualización de software `packagekitd`. La desactivación de este servicio es importante de cara al rendimiento. Si no se desactiva tanto en la máquina real como en la virtual, puede ocurrir que las dos empiecen a la vez a comprobar la existencia de actualizaciones de paquetes (normalmente, cuando las dos obtengan acceso a Internet al mismo tiempo). Esto provocará un aumento en el uso del procesador, una gran actividad de disco y

²La orden `fedoravm`, la máquina virtual que usamos en prácticas y unas breves instrucciones de instalación están disponibles en <http://ditec.um.es/iso/Fedora24Lab.tgz>.

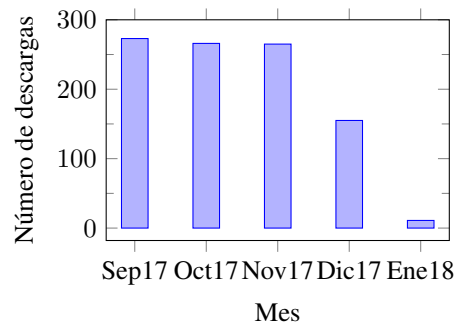


Figura 2: Descargas de la máquina virtual durante el primer cuatrimestre del curso 2017-2018.

un consumo importante de memoria RAM que ralentizarán el sistema, especialmente en máquinas con poca memoria. Debemos tener en cuenta que la simple desactivación de este servicio no es suficiente por sí sola, ya que puede ser activado por otros componentes, como el entorno gráfico utilizado. La solución, por tanto, pasa por enmascararlo en `systemd`.

En el caso de la máquina real, además de la desactivación de servicios mencionada, el cambio más destacable ha sido el uso del entorno gráfico LXDE en lugar de GNOME en aquellos ordenadores con 2 GiB de RAM. En estos ordenadores, GNOME deja menos de 800 MiB libres una vez en ejecución. LXDE deja, al menos, 1300 MiB libres, lo que permite fijar sin problemas la cantidad de memoria RAM de la máquina virtual en 1280 MiB. En el resto de laboratorios, donde los ordenadores tienen 4 u 8 GiB de RAM, se sigue usando GNOME y se le da a la máquina virtual la mitad de la RAM disponible en la máquina real.

Se podría mejorar todavía más el rendimiento de la máquina virtual mediante el uso de *drivers* paravirtualizados en el huésped, y que en QEMU+KVM se conocen como *Virtio*. El uso o no de estos *drivers* depende del trabajo que se quiera hacer dentro de la máquina virtual. En nuestro caso, por ejemplo, nos planteamos usar *Virtio* para los cuatro discos que se simulan en la máquina virtual. El problema es que, entonces, ya no es posible mostrar el funcionamiento de los planificadores de disco que proporciona Linux, ni es posible trabajar con prioridades de E/S. En el caso de otros dispositivos, como tarjetas de red, el rendimiento obtenido es más que suficiente. Todo ello hace que, por ahora, no usemos *Virtio* para nuestra máquina virtual.

2.4. Resultados obtenidos

Según una encuesta sobre el uso de la máquina virtual, a la que respondieron 30 de 85 posibles alumnos, la mayoría ha encontrado fácil su uso (63,3%), la ha usado tanto en casa como en los laboratorios de la Facultad (60,0%) y la ha instalado una vez al menos

Elemento	Características
Sistema operativo	Fedora 24 con escritorio GNOME.
Procesador	Como el del anfitrión, con dos procesadores y un núcleo por procesador.
RAM	Desde 1280 MiB hasta 4096 MiB, según laboratorio.
Discos simulados	Uno de 10 GiB, que contiene el sistema, y tres de pruebas de 1 GiB cada uno.
USB	USB 3 y hasta dos dispositivos.
Gráficos	VMWare SVGA. La máquina virtual toma automáticamente la resolución de la máquina real cuando se pasa a pantalla completa.
Red	RTL8139 y salida mediante NAT.
Copiar & Pegar	Entre huésped y anfitrión, en cualquier sentido.
Transferencias de ficheros	A través de red o mediante el protocolo SPICE y WebDAV.

Cuadro 2: Principales características de la máquina virtual usada en las prácticas de nuestra asignatura.

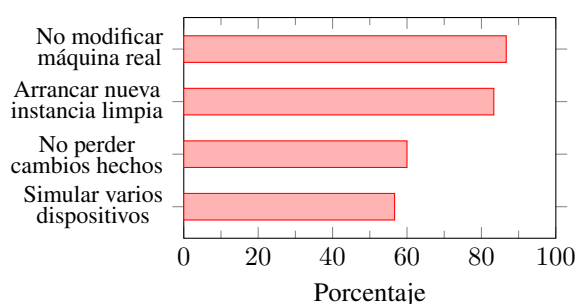


Figura 3: Características del entorno de virtualización mejor valoradas por los alumnos.

(86,7 %). Esto último concuerda con el gran número de descargas de la máquina virtual durante el primer cuatrimestre del curso 2017-2018 (ver figura 2). También se les ha preguntado por las características del entorno de virtualización proporcionado, y las que mejor valoran son las que aparecen en la figura 3.

En cuanto a los problemas encontrados por los alumnos, el principal (53,3 %) ha estado en la instalación de la máquina virtual en sus equipos. Aunque hay un fichero con instrucciones y se comentan los principales problemas que pueden surgir y cómo solucionarlos, creemos que el que la instalación se tenga que hacer ejecutando varias órdenes en una consola, que es una interfaz que apenas conocen, ha sido la principal dificultad. Otro problema añadido parece haber sido el que la instalación se tenga que hacer necesariamente en una distribución de Linux. Posiblemente, la implementación de un programa que guíe al alumno en el proceso de instalación solucione dichos problemas.

3. Entorno para la evaluación de administración de sistemas

La realización de exámenes usando ordenadores se debe hacer en un entorno controlado para, entre otras

cosas, evitar copias entre alumnos, como ya hemos comentado. Además, teniendo en cuenta que los ordenadores empleados se utilizan también para docencia de un gran número de asignaturas, es necesario que el entorno de exámenes sólo se active en ciertos momentos y permanezca inactivo el resto del tiempo. Los siguientes apartados describen el trabajo realizado para construir nuestro entorno de evaluación³.

3.1. Cuentas de usuario

Para poder establecer dos turnos de examen, los ordenadores de laboratorio disponen en Fedora 24 de dos cuentas llamadas «examen1» y «examen2». Estas cuentas están bloqueadas y ocultas por defecto, y sólo se activan para realizar los exámenes. Por contra, la cuenta «alumno», usada habitualmente durante las sesiones de prácticas, se bloquea durante los exámenes.

Los alumnos de un turno sólo pueden usar la cuenta correspondiente a su turno. Este control temporal se realiza con el módulo `time` del sistema de autenticación PAM, cuyo fichero de configuración se modifica convenientemente (ver apartado 3.3).

También, para evitar que algún alumno del primer turno cambie los permisos de su directorio personal para proporcionar acceso a su contenido al alumno del segundo turno que use el mismo ordenador, o para evitar que deje ficheros con permiso de lectura para todos en directorios públicos, como `/tmp` o `/var/tmp`, durante el examen hay una tarea que cada 5 minutos corrige los permisos para que estos accesos no se den.

3.2. Máquina virtual para exámenes

La máquina virtual para los exámenes es la misma que la utilizada en las sesiones de prácticas. Como los cambios que realiza un usuario en la máquina virtual

³Los ficheros usados para crear el entorno de evaluación se encuentran en http://ditec.um.es/iso/entorno_examenes.tgz.

se guardan en ficheros de su propiedad, en su propio directorio personal, estos cambios no serán visibles a otros usuarios del mismo ordenador.

3.3. Activación del modo examen

La activación de lo que podemos llamar *modo examen* de los ordenadores es una tarea importante pues, además de dejar los ordenadores en un estado controlado, debe implementarse de tal manera que sólo se active cuando sea necesario. En nuestro caso, esta tarea la realizan dos unidades de servicio de systemd, llamadas `alumno.service` y `examen.service`. La primera se ejecuta antes que la segunda y desactiva el modo examen, bloqueando las cuentas «examen1» y «examen2», desbloqueando la cuenta «alumno» y desactivando la tarea periódica que corrige los permisos. Por su parte, la unidad `examen.service` se activa tras las unidades `alumno.service` y `chrony-wait.service` y ejecuta un guión shell que realiza lo siguiente:

1. Dentro de un bucle infinito, intenta primero descargar un fichero predeterminado de un servidor web y, si falla, intenta hacer lo mismo usando un segundo servidor web. Si consigue obtener el fichero de cualquiera de los servidores, sale del bucle y continúa. Si no, permanece en el bucle por tiempo indefinido.
2. El fichero descargado en el paso anterior es un guión shell que contiene dos variables: una con la fecha y hora de inicio del examen y otra con la fecha y hora de fin. Este guión shell se ejecuta con la orden `source` de `bash`, por lo que las variables que en él se definan pasan a ser variables del guión shell que ejecuta nuestra unidad `examen.service`. El que este fichero se trate como un guión shell añade una gran flexibilidad al entorno (ver apartado 3.4).
3. Se comprueba si estamos en el periodo de realización del examen. En este punto, es importante que el ordenador tenga la hora correcta. Nos aseguramos de esto haciendo que `examen.service` se ejecute tras la unidad `chrony-wait.service`. Por lo tanto, el servicio `chronyd` debe estar correctamente configurado. Si se determina que no se está realizando un examen, se termina. En otro caso, se continúa.
4. Se entra en otro bucle infinito, similar al del primer paso, para descargar el fichero `time.conf` que se usará para la configuración del módulo `time` de PAM. Una vez descargado, el fichero se copiará al directorio `/etc/security`.
5. Finalmente, se bloquea la cuenta «alumno», se desbloquean «examen1» y «examen2» y se activa la tarea periódica para corregir los permisos.

Como acabamos de comentar, al arrancar un ordenador se ejecuta el servicio `alumno.service`, que desactiva el modo examen, y, a continuación, se ejecuta el servicio `examen.service`, que lo vuelve a activar si es el caso. Puede parecer absurdo ejecutar un servicio que desactiva el modo examen para ejecutar a continuación otro servicio que lo puede activar, en lugar de tener un único servicio que active o no el modo examen. El problema de tener un único servicio es que si, tras realizar un examen, un ordenador de los usados se arranca y no consigue obtener la hora por red (por ejemplo, por un cable de red desconectado), el ordenador quedará en modo examen, por lo que no podrá ser utilizado de forma habitual. Por eso, es mejor desactivar siempre el modo examen al arrancar y activarlo sólo si se tiene la certeza de que hay que hacerlo.

Un último aspecto a tener en cuenta es la interacción con SELinux. Si no se establecen los contextos adecuados, ni las unidades, ni los guiones shell que en ellas se indican, se ejecutarán.

3.4. Cortafuegos

Activar un cortafuegos en cada máquina real es fundamental si se quiere evitar el uso de canales de comunicación no autorizados. Por ejemplo, en un primer momento, se establecieron cortafuegos a nivel de laboratorio, por lo que sólo se podía acceder a nuestros servidores durante los exámenes. El problema fue que algunos alumnos usaban la orden `ncat` para establecer un pequeño *chat* entre ellos. Se pensó en desinstalar el paquete de dicha orden, pero el software de virtualización dependía de él. La solución, por tanto, fue configurar el cortafuegos de cada máquina real para permitir conexiones de salida sólo hacia nuestros servidores.

El tener un cortafuegos configurado así también beneficia a los alumnos, ya que no es necesario eliminarles la conexión de red completamente y pueden ir realizando entregas parciales de sus soluciones durante el examen, a modo de copias de seguridad, por si hubiera algún problema con la máquina real o virtual.

En nuestro caso, el cortafuegos lo configura el guión shell que se descarga al arrancar un ordenador (ver apartado 3.3). Nos gustaría destacar que el uso de este guión nos permitió configurar el cortafuegos la primera vez de forma centralizada a pocos días de un examen, sin tener que modificar la configuración de todos los ordenadores de los laboratorios. Esto nos hizo ver la importancia de tener este tipo de mecanismos para solucionar problemas inesperados o de última hora.

3.5. La orden uploader

La última pieza importante de nuestro entorno de evaluación es un guión shell llamado `uploader`, que se encuentra instalado en la máquina virtual. Nada más

arrancar ésta e iniciar sesión en la misma, los alumnos ejecutan `uploader` para descargar el material que van a tener disponible, incluyendo el propio enunciado del examen. La descarga también incluye una lista con los nombres válidos de los ficheros en los que se deben dejar las soluciones de los distintos ejercicios.

La segunda vez que se ejecute `uploader` se omitirá la descarga de material y se intentará hacer una primera entrega del examen, siempre que el contenido del directorio con soluciones sea correcto. Esto será así si el directorio sólo contiene ficheros regulares no vacíos que puedan ser leídos y cuyos nombres se encuentren en la lista de nombres válidos. En caso de error, se da un mensaje adecuado y se termina.

Si el directorio de soluciones es correcto, lo último que hace `uploader` es pedir al alumno sus datos personales y subir su solución. Estos datos sólo se piden la primera vez que se hace una entrega. En las siguientes entregas, bastará con confirmarlos (también se puede corregir algún dato si es incorrecto).

Un aspecto fundamental de `uploader` es que, al subir una solución, genera lo que llamamos un *código de examen*, que no es más que los 5 primeros caracteres del resumen MD5 de varios ficheros. Los alumnos deben escribir este código en un papel, junto con su nombre, al finalizar el examen.

El código de examen es como una firma digital que garantiza que la solución que un alumno dice haber subido es realmente suya. Como hemos comentado en la sección 1, si se usa un campus virtual o similar para entregar las soluciones, un alumno podría subir su solución y la de otros compañeros conociendo sus credenciales. Con nuestra solución, aunque un alumno suba una solución con los datos personales de otro, todavía tendría que pasarle el código generado para que el compañero lo escribiera en su hoja de examen. Además, nuestros servidores registran desde qué IP se ha subido cada examen, por lo que podemos detectar si las entregas de dos alumnos distintos se han hecho desde la misma IP o, incluso, si las entregas de un mismo alumno se han hecho desde dos IPs distintas.

Nuestra propuesta también evita las entregas no controladas desde fuera de los laboratorios, algo que se podría dar si se usara un campo virtual, como también hemos comentado. Esto es así porque el cortafuegos configurado en nuestros servidores impide accesos desde fuera de los laboratorios durante un examen y porque, de alguna manera, habría que hacer llegar el código del examen al alumno que se está examinando.

3.6. Servidores

Para proporcionar al guión shell ejecutado por `examen.service` (ver apartado 3.3) el fichero que le dice si debe pasar o no a modo examen, se usa un ordenador en el que se ha activado un servidor web y del

Elemento	Características
Sistema Operativo	Fedora 27
Placa base	ASUS P6T Deluxe V2
Procesador	Intel Core i7 920 a 2,67 GHz
RAM	12 GiB ECC 1067 MHz
Disco SSD	Kingston SH103S3 120 GB

Cuadro 3: Características de nuestro servidor.

que se pueden descargar ése y otros ficheros necesarios sin autenticarse de ninguna manera.

Por otro lado, las transferencias entre `uploader` y este ordenador se hacen mediante la orden `scp` y dos cuentas de usuario: una para descargar el material, en la que sólo se tiene permiso de lectura, y otra para subir las soluciones (parciales o finales) de los exámenes, y en la que sólo se tiene permiso de escritura. Para evitar que dos entregas, del mismo o de alumnos diferentes, colisionen, se han asignado nombres convenientes a los subdirectorios de esta segunda cuenta donde quedarán las soluciones. Tanto para una cuenta como para la otra, se ha usado `rssh` para restringir el tipo de conexiones permitidas (sólo `scp` en nuestro caso) y configurar máscaras de permisos.

Como hemos explicado, el servidor lleva un registro para conocer desde qué IP se ha descargado el material del examen o se ha subido una solución. Para conseguir este registro, ha sido necesario modificar el código fuente de `rssh` para que muestre qué proceso crea cada directorio de entrega. Esta información, junto con el resto de información proporcionada por otros servicios del sistema operativo, como `audit`, nos permite asociar inequívocamente una entrega con una IP.

En cuanto a las características hardware del servidor, nuestra experiencia nos dice que un ordenador relativamente moderno es más que suficiente para atender a los 100 alumnos que puede haber a la vez haciendo un examen. El cuadro 3 muestra las principales características de nuestro servidor. Tenemos, además, un segundo servidor por si hubiera problemas con el principal.

3.7. Tareas post-examen

Una vez finalizado el examen, quedan por hacer las siguientes tareas:

- Verificar que los códigos de examen se corresponden con el contenido de los ficheros subidos.
- Comprobar que cada alumno realiza sus entregas desde una única IP y que ésta no se repite con ninguna otra IP usada en el mismo turno.
- Enviar a cada alumno un correo con los ficheros de su última solución subida.

Todas estas tareas se realizan automáticamente usando diversos guiones shell.

4. Trabajos relacionados

El uso de máquinas virtuales o de tecnologías similares para docencia no es algo nuevo, como prueban los numerosos trabajos que han aparecido en las JENUI durante los últimos años [1, 2, 3, 5, 6, 7]. Un inconveniente de la mayoría de estos trabajos es que suelen delegar la gestión de la máquina virtual en los alumnos. Nuestra propuesta, en cambio, se centra en facilitar al máximo el uso de esta tecnología, permitiendo a un alumno, de una forma sencilla, tanto continuar con la máquina virtual con la que estaba trabajando como arrancar al instante una máquina virtual nueva.

También hay entornos donde las máquinas virtuales se lanzan automáticamente, como en algunos cursos ofrecidos a través de la plataforma NetAcad⁴, o con un simple clic de ratón, como ocurre en los servicios de máquinas virtuales ofrecidos por muchas universidades. Incluso se ha propuesto el uso de recursos computacionales en la nube, gestionados por docentes, para actividades educativas [4]. Las ventajas de estas soluciones son su facilidad de uso por parte del alumno y su disponibilidad desde cualquier lugar, pero tienen como inconvenientes el depender de una red de banda ancha y el hecho de que el estado de la máquina virtual se pierda cuando se apaga ésta o se deja de usar.

Ninguno de los trabajos analizados muestra la construcción o el uso de un entorno de evaluación similar al nuestro en el que se empleen máquinas virtuales. La disponibilidad de tales entornos parece ser una necesidad creciente ya que, en facultades como la nuestra, cada vez más asignaturas realizan pruebas de evaluación en las que se requiere el uso de un ordenador.

5. Conclusiones y trabajos futuros

Hemos presentado nuestra experiencia diseñando e implementando un entorno para la docencia y evaluación de administración de sistemas usando máquinas virtuales. Para docencia, nuestra propuesta permite a un alumno arrancar al instante una máquina virtual nueva o continuar con la que estaba usando de forma sencilla. Para la evaluación, el sistema creado permite la realización de exámenes en un entorno controlado que presenta varias características interesantes como: entregas parciales, verificación de la presencia del alumno, posibilidad de establecer turnos, etc. De manera general, el uso de máquinas virtuales nos ha permitido establecer una plataforma común de prácticas útil tanto para profesores como para alumnos.

Por sus características, creemos que nuestra solución también es aplicable en otras asignaturas, requieran o no acceso como administrador, usen o no Linux como sistema operativo.

Finalmente, tenemos mejoras pendientes, como el no permitir el acceso a dispositivos USB durante los exámenes, que quedan como trabajos futuros.

Agradecimientos

Mi más sincero agradecimiento a Javi Cuenca y Manolo Acacio por sus comentarios y sugerencias, gracias a los cuales ha sido posible mejorar el entorno creado.

Referencias

- [1] Juan A. Gil Martínez-Abarca, Adolfo Albaladejo Blázquez, Francisco Maciá Pérez, Francisco J. Mora Gimeno y Segismundo Ferrairó Pons. Entorno de red virtual para la realización de prácticas realistas de Administración de Sistemas Operativos y Redes de Computadores. En *XI Jornadas sobre Enseñanza Universitaria de la Informática*, págs. 349–355, julio 2005.
- [2] Pedro P. Gómez Martín. Máquinas virtuales en las clases de informática. En *XII Jornadas sobre Enseñanza Universitaria de la Informática*, págs. 417–424, julio 2006.
- [3] Germán Moltó y Oscar Sapena. Entorno virtualizado de aprendizaje para facilitar el desarrollo de destrezas de programación. En *XIX Jornadas sobre Enseñanza Universitaria de la Informática*, págs. 327–334, julio 2013.
- [4] Germán Moltó, Damian Segrelles y Miguel Caballer. Gestión de recursos computacionales en el Cloud para actividades educativas. En *XX Jornadas sobre Enseñanza Universitaria de la Informática*, págs. 99–106, julio 2014.
- [5] Francisco J. Ribadas Pena, Rubén Anido-Bande y Víctor M. Darriba-Bilbao. DSBOX: herramienta docente para el diseño y simulación de entornos de red virtualizados. En *XXII Jornadas sobre Enseñanza Universitaria de la Informática*, págs. 335–342, julio 2016.
- [6] Francisco J. Ribadas Pena, Francisco M. Barcala Rodríguez, Víctor M. Darriba Bilbao y Juan Otero Pombo. Diseño de un entorno virtualizado para la docencia práctica de Seguridad en Sistemas de Información. En *XIV Jornadas sobre Enseñanza Universitaria de la Informática*, págs. 325–332, julio 2008.
- [7] Antonio Ruiz Martínez, Rafael Marín López, Fernando Pereñíguez García, Pedro M. Ruiz Martínez y Antonio F. Gómez Skarmeta. Experiencia con la herramienta de virtualización VNUML para la enseñanza de redes de computadores. En *XVI Jornadas sobre Enseñanza Universitaria de la Informática*, págs. 161–168, julio 2010.

⁴<https://www.netacad.com/>