Despliegue de escenarios de entrenamiento en seguridad utilizando técnicas de virtualización con clonado rápido

Ernesto Pérez E.[1], Paul F. Bernal B.[2]

[1] Docente de la FISEI, Universidad Técnica de Ambato, Avenida de los Chasquis y Río Payamino, Ambato, Ecuador, EC180103.

[2] Ingeniero especialista del CSIRT-CEDIA, Consorcio Ecuatoriano para el Desarrollo de Internet Avanzado, La Condamine 12-109 sector Subida del Vado, Cuenca, Ecuador, EC010150.

Autores para correspondencia: ernesto.perez@uta.edu.ec, paul.bernal@cedia.org.ec

Fecha de recepción: 21 de septiembre de 2014 - Fecha de aceptación: 17 de octubre de 2014

RESUMEN

La formación y preparación constante del personal de TI es una de las estrategias más efectivas para mejorar la calidad, estabilidad y seguridad de las redes y servicios asociados. En esta línea, el CEDIA ha venido implementando cursos y talleres de capacitación dirigidos a sus miembros y, dentro del CSIRT-CEDIA, se ha pensado en la posibilidad de optimizar los procesos asociados al despliegue de la infraestructura necesaria para proveer a los participantes de éstas capacitaciones, con el material personalizado adecuado, en las áreas de seguridad informática. Es así que se decidió usar técnicas de virtualización para aprovechar los recursos disponibles, pero aun cuando esto en sí no es una tendencia nueva, el uso de una copia completa del disco virtual para cada participante, no sólo resulta impráctico en cuestión de tiempo, sino también en cuanto al consumo de almacenamiento necesario. Este trabajo se orienta justamente a la optimización en los tiempos y consumos asociados a los procesos de replicación de un mismo equipo y disco virtuales para uso particularizado de varios participantes.

Palabras clave: KVM, virtualización, qcow2, backing storage, optimización, escenarios, seguridad, entrenamiento.

ABSTRACT

The training and constant preparation of the IT personnel is one of the most effective strategies to improve the quality, stability and security of networks and associated services. In this line, CEDIA has been implementing training courses and workshops for its members and within the CSIRT-CEDIA it has been considered the optimization possibility of the associated processes with the necessary infrastructure deployment to provide participants of these trainings, with appropriate personalized material in information security area. Thus we decided to use virtualization techniques to exploit the available resources, but even if by itself it isn't a new trend, the use of a complete copy of the virtual disk for each participant, is not only impractical in terms of time, but also in terms of the consumption of required storage. This work is aimed precisely to the consumption and time optimization associated with replication processes of the same machine and virtual disk for particular use of several participants.

Keywords: KVM, virtualization, qcow2, backing storage, optimization, scenarios, security, training.

1. INTRODUCCIÓN

En la vida cotidiana de los administradores de sistemas, se presentan eventos de diversos tipos como la preparación de un nuevo servicio, el despliegue de una cierta tecnología y el manejo de incidentes de seguridad, así como la experimentación e investigación de nuevas o más eficientes formas de realizar

ésas y otras tareas. Es frecuente que la preparación de los administradores de sistemas sobre dichas áreas, sea más bien teórica y exista poca experiencia práctica o entrenamiento de campo sobre las formas, técnicas y recursos que se pueden usar para atender dichos escenarios.

Esta realidad conlleva la necesidad de buscar formas de proveer escenarios de experimentación realistas para la adecuada preparación de los administradores. Sin embargo esto mismo se constituye en un problema a la hora de disponer de los recursos -especialmente de hardware y de tiempo- necesarios para abastecer con un escenario particular a cada estudiante, pues como se puede entender, no es viable compartir un mismo escenario porque se limitan las posibilidades de experimentación y aprendizaje.

Por otro lado, en el CSIRT de CEDIA pensamos que una de las más eficientes formas de implementar cambios es a través de la educación y preparación de los miembros de la comunidad de TI, pues así se podrán adquirir las destrezas y capacidades para atender oportuna y adecuadamente los diferentes eventos. Sobre esta base se han venido planificando múltiples capacitaciones orientadas a generar éstas experiencias prácticas en los participantes, usando escenarios virtuales que repliquen de una forma realista los diversos eventos de seguridad que resultan ser los más comunes (OWASP s.f.).

2. TRABAJOS RELACIONADOS

Fuertes & López (2009) se propone escenarios de emulación de sistemas en red que replican procesos en la realidad. Demostrando que es posible utilizar técnicas de virtualización para uso en escenarios de prueba y de entrenamiento. En este caso en particular se trata de demostrar que implementando ajustes adecuados al sistema de virtualización utilizado, se pueden lograr escenarios con comportamientos similares a los de la vida real. En este trabajo se analiza el subsistema de red en virtualización, haciéndose una mención al resto de subsistemas requeridos para virtualizar, como son: procesador, memoria, y almacenamiento.

Además se conocen las ventajas de la virtualización al ser usadas para desarrollar laboratorios de seguridad, las ventajas del aislamiento, consolidación y replicación de la virtualización (Fuertes *et al.*, 2009); evidenciando ventajas de efectividad en su costo de implementación, con un análisis sobre la acogida de esta tecnología desde varios aspectos como acceso remoto, aceptación por parte de profesores y estudiantes, manejo de la red, pruebas y validaciones. Pero no se hace énfasis en la tecnología de almacenamiento de estos escenarios.

Xen y KVM (citado en Wang *et al*., 2014) ya han sido estudiados y usados para describir ambientes de experimentación virtualizado, proponiendo el uso de un sistema denominado pVEE que permite la creación de ambientes de experimentación a través del cual profesores pueden generar máquinas virtuales (MV) con los contenidos necesarios a ser utilizados por estudiantes que así lo soliciten. Esta solución propone el uso de imágenes basadas en qcow2 (McLoughlin, 2008). Separándolas dentro de una MV en imágenes con los contenidos de las MV creadas por el profesor e imágenes con los contenidos del estudiante. Sin embargo no se justifica el porqué de la utilización de qcow2 para el sistema, aunque sí se describe el sistema de almacenamiento y su organización en templates. De acuerdo a la descripción del proceso de creación de las MV propuesto, se requiere de varias imágenes para ser utilizadas por la MV del estudiante, resultando un proceso de creación más bien complejo, siendo por lo mismo susceptible de simplificación y optimización.

Si bien, el uso de esquemas de virtualización para brindar entrenamiento no es nuevo, y como se ha visto, existen varios estudios previos con objetivos muy similares, éstos no necesariamente abordan directamente su implementación desde el punto de vista de la optimización en los procesos, recursos y procedimientos técnicos de despliegue usando virtualización y clonado rápido de discos, que es justamente a donde se orienta este trabajo, mismo que se concentra en aprovechar la tecnología de backing storage implementada por KVM en el formato de qcow2 con la finalidad de permitir un rápido despliegue de escenarios de pruebas con un ahorro apreciable de tiempo y en el uso de almacenamiento en los host de virtualización (HV) de estos escenarios de eventos de seguridad conteniendo problemas reales a solucionar por el estudiante.

El resto de este artículo se estructura de la siguiente forma: En la sección IV se plantean los escenarios de seguridad a implementar así como la documentación a usar el profesor. La sección V se describe el proceso de despliegue de los escenarios: configuración del HV, generación de los templates de escenarios, proceso de creación de las MV para los estudiantes, así como consideraciones de seguridad para su despliegue. La sección VI se discuten los resultados obtenidos. Y, finalmente, se brindan conclusiones y recomendaciones para el trabajo a futuro.

3. SOLUCIÓN PROPUESTA

El resultado final debe componerse de dos partes principales para constituirse en una solución completa y efectiva: la creación de los escenarios propiamente dichos y el material de apoyo respectivo que acompañe la labor del instructor.

3.1. Creación de los escenarios

La creación de escenarios prácticos se orienta a los problemas de seguridad más comunes de acuerdo a los informes de ShadowServer que recibimos sobre eventos ocurridos en nuestras redes o problemas de seguridad que detectan en estas (Shadowserver s.f.), también al Top Ten de Open Web Application Security Project (OWASP s.f.) así como a ataques de amplificación a través de UDP reportados por entidades de seguridad (US-CERT s.f.) con la finalidad de realizar actividades maliciosas. Estos escenarios son presentados de forma puntual en una máquina virtual que ha sido preparada con la respectiva vulnerabilidad disponible para su explotación y solución a través de su detección y análisis.

Sobre esta base se identificaron e implementaron los problemas más comunes presentados y se desarrollaron equipos virtuales para cada uno de los siguientes escenarios:

- *Defacements de sitios web debido a fallas en el manejador de contenidos:* Consiste de 3 escenarios del manejador de contenidos que han sido desfigurados utilizando claves débiles: template-joomla-weakpassword.qcow2 de 287 MB, template-joomla-admin25.qcow2 de 108 MB y template-joomla-admin3.qcow2 de 295 MB.
- *Open Resolvers:* Se generan dos escenarios: servidor tipo bind (named) template-openresolver-bind.qcow2 de 8.5 MB y otro servidor tipo dnsmasq template-openresolver-dnsmasq.qcow2 de 5.0 MB, los cuales pueden ser utilizados como práctica para detectar y corregir ataques de amplificación de UDP en este caso utilizando Open Resolvers.
- *Open Proxies:* Consiste en un escenario que aprovecha fallas en la configuración de un proxy squid template-openproxy-squid.qcow2 de 5.7 MB y otro escenario aprovechando el servidor web de Apache configurado de forma inapropiada template-openproxy-apache.qcow2 de 6.5 MB. Ambos le permiten al atacante navegar haciendo uso de estos servidores con la finalidad de tratar de ocultar el origen de su actividad.
- *Phishing:* Escenario real de phishing a PayPal, este escenario es una página web con una simulación del sitio de PayPal con la finalidad de capturar información que sea digitada a través de este sitio pescador.tgz de 283 KB.
- *Envío de SPAM a través de servicio de correo:* Servidor de sendmail con saslauthd y dovecot template-sendmail-spam.qcow2 de 280 MB. Este escenario contiene un usuario con una clave débil que es aprovechado por un atacante para realizar envíos de correos no deseados.
- *Open NTP:* Servidor de tiempo de red template-ntp.qcow2 de 6 MB. Configurado con opciones que permiten realizar ataques utilizando técnicas de amplificación de UDP, en este caso a través del protocolo NTP.

3.2. Entregables a instructores

Complementando el material de cada curso, además de la plantilla de la MV que consiste en un archivo qcow2 con el escenario de ejemplo listo para ser desplegado a los participantes, el instructor dispone de un documento de apoyo al profesor con indicaciones para la instalación del escenario,

cuestionarios para el participante y sugerencias para apoyar al estudiante en la solución del problema (Pérez s.f.).

4. DESPLIEGUE DE ESCENARIOS

Los escenarios fueron desarrollados utilizando Kernel-based Virtual Machine (KVM) (Shah s.f.) como plataforma de virtualización. Se aprovecha la característica de QEMU de utilizar archivos base (backing storage) para un rápido despliegue (Novich s.f.).

Todos los clientes acceden en modo de sólo lectura al mismo disco base y los discos de los clientes son utilizados solamente con la finalidad de guardar los cambios que se requieran en la MV propia de cada usuario. La MV de un participante buscará primero un archivo en el disco de él y si no lo encuentra, lo buscará en el disco base.

De esta forma se logra un ahorro en el espacio de almacenamiento y en el tiempo de despliegue, pues pueden instalarse grandes cantidades de MV en pocos minutos y con bajo consumo de disco.

Una generalización del proceso puede ser observada en la Fig.1 donde el disco base tiene 3 archivos (A, B y C), a partir de él se crean los discos particulares de dos escenarios. El escenario 1 usa el archivo A del disco base y tiene sus propios archivos B y C. Dentro del escenario 1, están los discos de dos participantes que heredan el archivo A del disco base y el archivo B del escenario 1, manejando cada uno su propia versión de C, almacenada de forma particular.

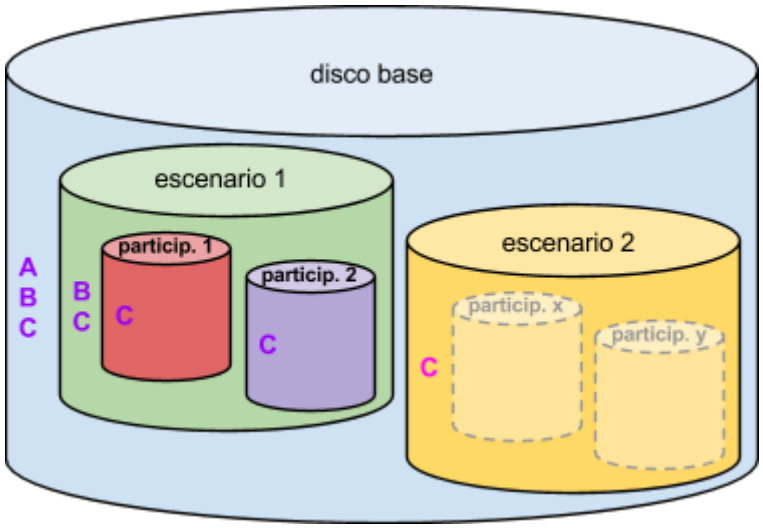

Figura 1. Ejemplo de herencia de archivos desde el disco base.

El proceso de búsqueda del archivo A en la MV del participante 1 sería:

1. Buscar archivo A en el disco participante 1, no lo encuentra entonces,
2. Buscar archivo A en el disco escenario 1, no lo encuentra entonces,
3. Buscar archivo A en el disco base, lo encuentra.

Como se puede observar los discos de los participantes solamente almacenan los cambios que ellos generan, utilizando los discos base y del escenario para leer la información que no modificaron. Lo mismo ocurre con los participantes de los otros escenarios creados durante el trabajo.

4.1. Configuración del host de virtualización

El Sistema Operativo (SO) utilizado como HV es CentOS-6 de 64 bits (CentOS s.f.), el hardware sobre el cual corre tiene 8 GB de RAM y un disco duro SATA de al menos 500 GB, que evidencia la

baja demanda de recursos necesarios para utilizar los escenarios. Al instalar el SO se escogió el grupo de paquetes "Virtualización" que provee de las herramientas necesarias para utilizar KVM.

El disco físico del HV, fue particionado de forma tal que hubiera espacio suficiente para alojar las MV en formato qcow2 (McLoughlin s.f.) usando un esquema de particionamiento simplificado:

- Partición /boot de 200 MB,
- Partición SWAP de 1 GB y,
- Partición / que abarca el resto del disco del HV.

Para el almacenamiento de las imágenes de los discos virtuales, se utilizó el directorio sugerido por defecto en CentOS: /var/lib/libvirt/images.

Antes de ejecutar cualquier operación adicional, el servidor se actualizó con el comando yum update para eliminar fallas de seguridad conocidas y se limitó el acceso de root por SSH únicamente a usuarios que utilicen clave pública-privada, añadiendo la directiva PermitRootLogin without-password en el archivo /etc/ssh/sshd_config. Para realizar las clonaciones, igualmente con yum se instaló el paquete libguestfs-tools-c.

Se creó una interfaz de red de tipo puente (bridge) llamada br0, asociada a la tarjeta de red que conecta al HV con la LAN. Cada MV tiene una interfaz de red asociada a este bridge, de forma tal que pueda acceder a la red donde está conectado el servidor y al resto de MV de ser necesario.

Se ha realizado el mismo proceso utilizando Fedora Linux-20 (Fedora s.f.) de 64 bits para ello es necesario cambiar las variables emulator y machine por las diferencias de versiones existentes entre libvirt/KVM de CentOS-6 y Fedora-20. Por ello el archivo clonar.sh (Pérez, Script de despliegue de máquinas virtuales s.f.) referido más adelante, ha sido preparado de forma tal que puede ser utilizado tanto para CentOS-6 como para Fedora-20.

4.2. Generación de disco base

El disco base contendrá una instalación mínima de un sistema CentOS-6 actualizada. Los escenarios a instalarse, harán uso de este sistema base mínimo. Con el fin de simplificar la creación de éste disco base, se usó virt-manager (VirtManager s.f.) tal cual se puede ver en la Fig. 2:

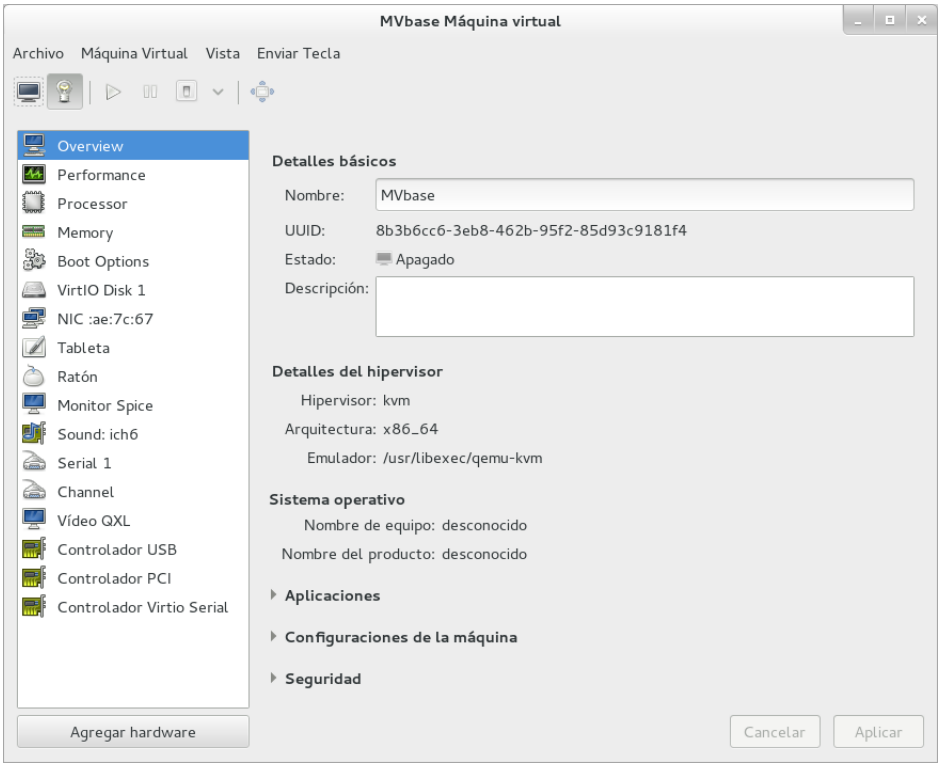

Figura 2. Creación de la MV base con virt-manager**.**

Una vez instalado, se actualizó el SO y se eliminó los históricos con la finalidad de que el participante no perciba los cambios que se pudieron hacer durante el proceso de preparación del escenario. Adicionalmente, con la finalidad de que al clonarse el disco, la red de las MV no llegue a fallar por diferencias entre la dirección MAC de la imagen y de la MV, se eliminó el archivo /etc/udev/rules.d/70-persistent-net.rules y se retiró también el parámetro HWADDR= del archivo /etc/sysconfig/network-scripts/ifcfg-eth0.

Se apagó el firewall con la finalidad de que no interfiriera con los posibles servicios que corran los escenarios. En caso de ser requerido por un escenario en particular, el firewall puede levantarse.

El disco base generado del proceso template-base.qcow2, puede usarse para la creación de los escenarios subsecuentes y la preparación de los escenarios de entrenamiento.

4.3. Generación de escenarios

Cada plantilla de escenario template-*.qcow2 aprovecha los contenidos del disco base y en los discos de las máquinas escenario se implementan los sistemas preparados con las fallas a estudiar y -de ser requerido- algunas pistas falsas que puedan aumentar el realismo o complejidad del escenario.

El proceso para la generación de los escenarios, inicia con la creación de la plantilla de escenario como hija del disco base:

```
# qemu-img create -b discoBase.qcow2 -f qcow2 discoEscenario.qcow2
```

Luego se elimina del discoEscenario el archivo de reglas de los dispositivos de red de udev:

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules
```

Al igual que el parámetro HWADDR= del archivo de configuración de la interfaz de red:

```
# sed -i '/HWADDR=/d' /etc/sysconfig/network-scripts/ifcfg-eth0
```

Y finalmente se eliminan los históricos de comandos previos ejecutados:

```
# export HISTSIZE=0;history -c
```

4.4. Generación de MV para participantes

Cada máquina de participante estará basada en un disco de un escenario apropiado escogido y se inicia con la creación de la MV utilizando como disco base al escenario relativo al taller/curso a realizarse:

```
# qemu-img create -b discoEscenario.qcow2 -f qcow2 discoParticipante.qcow2
```

Luego hay que abrir el disco de la MV del participante para cambiar la IP y de ser necesario el nombre de la máquina:

```
# guestmount -a discoParticipante.qcow2 -i /mnt
# rm -f /etc/udev/rules.d/70-persistent-net.rules
# sed -i '/HWADDR=/d' /etc/sysconfig/network-scripts/ifcfg-eth0
# vi /mnt/etc/sysconfig/network
```

Ahora hay que definir la MV del alumno dentro del sistema KVM, creando primero el archivo XML respectivo y finalmente arrancar la máquina del Alumno y probar conectividad:

```
# vi /tmp/mvParticipante.xml
# virsh define /tmp/mvParticipante.xml
# rm -f /tmp/mvParticipante.xml
# virsh create mvParticipante
```

Toda esta labor puede ser viable para un número reducido de MV, pero se complica y se vuelve cada vez más propensa a fallos, a medida que el número crece. Por ello se desarrolló un script que automatiza el proceso y que permite un despliegue ágil y simplificado de MV para varios usuarios. El script clona.sh (Pérez s.f.) es el que se encarga del proceso y se divide en dos secciones claramente demarcadas: una sección de configuración y una de generación de la MV.

En la sección de configuración de establecen los parámetros necesarios para la creación de las MV:

- *templatename:* Nombre del escenario que se utilizará para generar las máquinas de los alumnos.
- *clasec:* El prefijo de clase C, o los primeros tres octetos de la dirección IP, que se utilizará como base para generar las IP de las MV de los estudiantes.
- *ipinicio:* El número del cuarto octeto de la IP con que comienzan a generarse las IP para las máquinas de los participantes.
- *ipfin:* El número del cuarto octeto de la IP con que finaliza la generación de las IP para las máquinas de los participantes. La cantidad de máquinas a generarse se deduce de: ipfin-ipinicio+1.
- *mascara:* La máscara de red a configurar en cada máquina del participante.
- *gw:* La dirección IP de la puerta de enlace a configurar en cada máquina del participante.
- *imagesdir:* Esta variable contiene el directorio donde se almacenan las imágenes del disco base, el escenario y hacia donde se crearán los discos de los participantes.

La sección de generación de MV es más compleja y se ocupa de generar propiamente las MV de los participantes siguiendo a breves rasgos el siguiente flujo:

- Determinar si el HV está corriendo bajo Fedora o CentOS. En dependencia del SO se inicializan los valores emulator y machine que difieren en cada caso.
- Verificar que el administrador haya instalado el paquete libguestfs-tools-c de lo contrario finaliza la ejecución emitiendo un mensaje al administrador para que lo instale.
- Verificar que el escenario, definido por la variable templatename, esté presente en el directorio definido en la variable imagesdir, de lo contrario falla emitiendo un mensaje al administrador para que pueda corregir el inconveniente.
- Para cada una de las IP, desde ipinicio hasta ipfin, realiza un ciclo que se ocupa de crear la máquina del participante (qemu-img create), montar esta imagen en el directorio /mnt, sobreescribir el archivo de configuración de la interfaz de red de la máquina, así como asignarle un nombre a la máquina. Luego se elimina el archivo de configuración de udev que define la interfaz eth0, mismo que se regenerará automáticamente al arrancar la máquina. Finalmente se desmonta el disco de la máquina del participante de /mnt.
- Crear en /tmp/ un archivo con extensión .xml que definirá las propiedades de la máquina del participante como: nombre, cantidad de RAM, cantidad de procesadores, UUID, nombre del disco, dirección MAC; y se asocia la tarjeta de red virtual al bridge br0.
- Finalmente, a través del comando virsh-define, se crea la MV utilizando este archivo XML que luego es borrado pues queda ya registrado en /etc/libvirt/qemu/.

4.5. Consideraciones de seguridad para las MV

Se sugiere que para la configuración de estos escenarios se tenga en cuenta y/o se realice siempre los siguientes consideraciones:

- Las MV de los estudiantes contienen sistemas con una finalidad didáctica que han sido deliberadamente debilitados, ya sea con claves simples o sistemas anticuados y con fallas conocidas y por lo tanto no se sugiere en lo absoluto que estos sistemas estén expuestos a Internet.
- La implementación de este sistema, está recomendado sólo dentro de una red local, utilizando IPv4 privadas y los estudiantes deben conectarse a ellas, ya sea desde la misma LAN o a través de una red privada virtual (VPN).
- Una vez se haya probado el escenario, las MV deben ser apagadas y removidas del servidor para evitar que puedan ser utilizadas como punta de lanza para ataques a terceras personas.
- Bajo ninguna circunstancia deben utilizarse estas máquinas como base para implementar sistemas públicos o de producción.

5. RESULTADOS OBTENIDOS

Hemos evidenciado mejoras en dos aspectos fundamentales y claramente definidos que coinciden con los objetivos planteados inicialmente para el proyecto, como son el ahorro de espacio de almacenamiento y el ahorro en el tiempo de despliegue de los escenarios.

5.1. Ahorro de espacio

En la Fig. 3 podemos observar que el disco base: template-base.qcow2 consume 1.2G. El Escenario openresolver: template-openresolver-bind.qcow2 consume 8.5 MB y 10 MV de alumnos basadas en el Escenario: template-openresolver-bind-e*.qcow2 con 960KB de consumo cada una.

```
[root@hc6pe images]# ll -h template-{base,openresolver-bind}*
-rw-r--r-- 1 qemu qemu 1.2G Jan 14  2014 template-base.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:24 template-openresolver-bind-e112.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:24 template-openresolver-bind-e113.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:25 template-openresolver-bind-e114.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:25 template-openresolver-bind-e115.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:26 template-openresolver-bind-e116.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:26 template-openresolver-bind-e117.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:27 template-openresolver-bind-e118.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:27 template-openresolver-bind-e119.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:27 template-openresolver-bind-e120.qcow2
-rw-r--r-- 1 root root 960K Aug 20 23:28 template-openresolver-bind-e121.qcow2
-rw-r--r-- 1 qemu qemu 8.5M Jan 14  2014 template-openresolver-bind.qcow2
```

Figura 3. Ahorro de espacio con disco base.

El consumo de 8.5 MB del escenario openresolver guarda sólo los cambios que ella tiene en relación al disco base, sucediendo lo mismo con las MV de los alumnos en relación a la del escenario openresolver.

5.2. Rapidez en el despliegue

Para verificar la velocidad de despliegue, utilizamos el comando time aplicado al comando qemu-img usados dentro del script clona.sh, lo que nos permitió crear una máquina respaldada por un disco base, proceso que tomó poco menos de 2 décimas de segundo.

```
[root@hc6pe images]# time qemu-img create -b template-openresolver-bind.qcow2 -f
 qcow2 template-openresolver-bind-e212.qcow2
Formatting 'template-openresolver-bind-e212.qcow2', fmt=qcow2 size=8589934592 ba
cking_file='template-openresolver-bind.qcow2' encryption=off cluster_size=65536
lazy_refcounts=off

real    0m0.179s
user    0m0.032s
sys     0m0.024s
```

Figura 4. Tiempo de ejecución utilizando una imagen respaldada por disco.

A diferencia de crear una máquina clonada de forma tradicional, utilizando el comando virt-clone, que tomó prácticamente 3 minutos y medio.

```
[root@hc6pe ~]# time virt-clone --original template-openresolver-bind-e112 --aut
o-clone
Allocating 'template-openresolver-bind-e112-clone.qcow2'    | 8.0 GB  03:27

Clone 'template-openresolver-bind-e112-clone' created successfully.

real    3m28.973s
user    0m0.715s
sys     0m0.237s
```

Figura 5. Tiempo de ejecución utilizando el comando virt-clone.

Se pudo determinar así que el proceso de uso de disco base resulta ser unas 120 veces más rápido que el proceso utilizando la clonación tradicional.

Aunque no se analizó el impacto que los discos base tienen sobre el desempeño de una MV, se sabe que no hay una diferencia de tiempos apreciables entre el formato RAW y el nuevo qcow2 (Wolf s.f.). Adicionalmente no se encontró necesaria esta medición en vista de que los escenarios no son equipos de producción y estos estarán funcionando mayormente durante breves periodos de tiempo, solamente mientras el estudiante se encuentre analizando la información contenida en ellos.

Estos escenarios prácticos para entrenamiento han sido desarrollados por el CSIRT CEDIA, y sus resultados fueron probados en un taller práctico dictado durante el congreso TICEC 2013 (Pérez s.f.).

6. CONCLUSIONES Y RECOMENDACIONES DE TRABAJO FUTURO

El uso de técnicas de virtualización de clonado rápido aporta con la optimización de recursos a la hora de establecer procesos de aprendizaje de ciberseguridad, utilizando tecnologías de virtualización en Software Libre.

Si bien este trabajo se ha enfocado en el despliegue de escenarios orientados a la seguridad, las mismas técnicas y procedimientos pueden ser usados, con poca o ninguna adaptación, para el despliegue de escenarios orientados a cualquier otra área de formación.

Es necesario continuar realizando escenarios con eventos de seguridad adicionales como son: OpenSIP Server, servicio de FTP con clave débil, servicio de SSH con clave débil, Cross-Site Scripting, Cross Site Request Forgery, Inyección de SQL y otros ataques de amplificación de UDP a través de servicios como: netbios, snmpv2, chargen, qotd y ssdp.

Se deberá extender el script de clonación para soportar CentOS-7 y futuras versiones de Fedora, así como para simplificar aún más los procedimientos, encapsulando la complejidad y permitiendo que puedan ser usados por instructores con menores conocimientos en las técnicas y tecnologías usadas.

Estudiar el posible impacto que puede tener el uso de discos base en el desempeño de los talleres, entregará información que pueda ser usada en la determinación de infraestructuras de producción para grupos mucho más grandes de participantes con miras a la masificación y constancia de las capacitaciones.

AGRADECIMIENTOS

Agradecemos la disposición de CEDIA por permitirnos desarrollar la investigación usando su infraestructura y ofrecernos las facilidades necesarias para desplegar los escenarios.

REFERENCIAS

CentOS. *CentOS Project.* s.f. Disiponible en *https://www.centos.org/*.

Fedora. Fedora Project: Freedom, Friends, Features, First. s.f. Disponible en *https://fedoraproject.org/*.

Fuertes, W., J.E. López de Vergara, F. Meneses, 2009. *Educational platform using virtualization technologies: Teaching-learning applications and research uses cases*. Proc. II ACE Seminar: Knowledge Construction in Online Collaborative Communities, 6 pp.

Fuertes, W., J.E. López de Vergara, 2009. *An emulation of VoD services using virtual network environments.* Workshops der Wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen (WowKiVS 2009). Electronic Communications of the EASST, 17, 14 pp.

McLoughlin, M., 2008. The QCOW2 Image Format. s.f. Disponible en *https://people.gnome.org/~markmc/qcow-image-format.html.*

Novich, L. RedHat, Using qemu-img. s.f. Disponible en *https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Virtualization/sect-Virtualization-Tips_and_tricks-Using_qemu_img.html.*

OWASP. OWASP Top Ten Project. s.f. Disponible en *https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.*

Pérez, E. Documento de apoyo al profesor. s.f. Disponible en *http://csirt.cedia.org.ec/?attachment_id=274.*

—. Script de despliegue de máquinas virtuales. s.f. Disponible en *http://csirt.cedia.org.ec/?attachment_id=242.*

—. Taller práctico, 2013. *Reaccionando a incidentes informáticos*. I Encuentro de Tecnologías de Información y Comunicación de las Universidades del Ecuador. Quito, Ecuador.

Shadowserver. Services/Reports. s.f. Disponible en *http://www.shadowserver.org/wiki/pmwiki.php?n=Services/Reports.*

Shah, A. Kernel Based Virtual Machine. s.f. Disponible en *http://www.linux-kvm.org/page/Main_Page.*

US-CERT. Alert (TA14-017A) UDP-based Amplification Attacks. s.f. Disponible en *https://www.us-cert.gov/ncas/alerts/TA14-017A.*

VirtManager. Manage virtual machines with virt-manager. s.f. Disponible en *http://virt-manager.org/*.

Wang, X., X. Hong, T. Li, B. Gao, Z. Wa, 2014. *Pervasive Computing and the Networked World.* Springer International Publishing.

Wolf, K. QCOW2 performance, Linux-KVM. s.f. Disponible en *http://www.linux-kvm.org/page/Qcow2*.