

#####

HackWesTx - Fall 2024

Netboot.xyz

Bootear el sistema operativo desde la red en vez de un medio físico

Funcionamiento:

1. Preboot Execution Environment or (PXE):

PXE tecnología que permite que una computadora arranque desde la red antes de cargar un sistema operativo.

- Compatible con la mayoría de los sistemas operativos BIOS/UEFI
- Facilita la descarga de archivos de arranque desde un servidor en la red

2. Descarga del menú de netboot.xyz

Netboot ofrece un menú dinámico que te permite seleccionar varios sistemas operativos y herramientas de utilidad. Este menú se descarga desde un servidor en la red

3. Seleccionar sistema operativo

Después de arrancar el sistema puedes elegir entre sistema operativo que deseas instalar o iniciar en modo live (live iso)

4. Arranque del sistema operativo

el sistema utiliza esos archivos instalados temporalmente para iniciar el proceso de instalación o arranque. Por lo que no necesitas copia local para el sistema operativo

5. Elegir si instalar o iniciar live ISO

Beneficios

- Flexibilidad
- Actualizaciones constantes
- Menos dependencias

easyboot.xyz

myNetboot++ es un proyecto que apunta a hacer aun mas facil bootear desde la red, Principalmente nuestro objetivo es que el usuario pueda personalizar su distribucion invluso antes de haber booteado.

Posibilidades y Tiempos estimados:

- Despliegue automatizado de múltiples sistemas operativos (**~8 - ~9 horas**) ** [Ver más](#)
- Despliegue de sistemas personalizados con scripts preconfigurados (**~6 - ~8 horas**) 2. [Ver más](#)
- Probar distribuciones en vivo sin instalación (**~1 - ~2 horas**) [Ver más](#)
- Recuperación de sistemas con herramientas live **(~2 - ~3 horas) [Ver más](#)
- Automatización de instalaciones con preconfiguración (PXE + Kickstart/Preseed) (**~7 - ~9 horas**) [Ver más](#)
- Implementación de entornos efímeros **(~4 - ~5 horas) [Ver más](#)
- Instalación de contenedores directamente desde la red (**~6 - ~7 horas**) [Ver más#introduccion](#))
- Creación de un servidor PXE personalizado (**~10 - ~12 horas**) [Ver más](#)
- Instalación y mantenimiento de sistemas remotos (**~5 - ~6 horas**) [Ver más](#)
- Bootear imágenes ISO personalizadas desde la red (**~5 - ~7 horas**) [Ver más](#)
- Cargar entornos de virtualización ligera (**~5 - ~6 horas**) [Ver más](#)
- Soporte para instalar sistemas operativos no Linux (**~3 - ~4 horas**) [Ver más](#)

- Cargar entornos de pruebas de seguridad (pentesting) (**~2 - ~3 horas**) [Ver más](#)
- Monitoreo y análisis de red (**~4 - ~5 horas**) [Ver más](#)
- Implementación de entornos de desarrollo temporal (**~4 - ~5 horas**) [Ver más](#)

Documentacion

1. Despliegue automatizado de múltiples sistemas operativos

Configura un servidor PXE con Netboot.xyz para instalar diferentes sistemas operativos automáticamente en varias máquinas, sin intervención manual. (~8 - 9 horas)

Requerimientos

- **Hardware:**
 - Servidor dedicado o VM.'
 - Clientes PXE conectados a la red LAN.
- **Software:**
 - Servidor DHCP para asignar IPs.
 - Servidor TFTP para transferir archivos de arranque.
 - (Opcional) Servidor HTTP/FTP para alojar imágenes de sistemas operativos.
 - **Netboot.xyz** para gestionar la instalación de múltiples sistemas operativos.

Pasos de Implementación

1. Configuración del servidor TFTP

- Instalamos TFTP:

```
sudo apt install tftpd-hpa
```

Configuramos el archivo `/etc/default/tftpd-hpa` para especificar la carpeta de archivos:

```
TFTP_DIRECTORY="/srv/tftp"
```

Creamos la carpeta de archivos y le damos permisos:

```
sudo mkdir /srv/tftp && sudo chown -R tftp:tftp /srv/tftp
```

```
sudo apt install isc-dhcp-server
```

Configuramos el archivo `/etc/dhcp/dhcpd.conf` para señalar el servidor TFTP:

```
filename "pxelinux.0";  
next-server 192.168.1.10;
```

3. Descarga y Configuración de Netboot.xyz

Descargamos Netboot.xyz:

```
wget https://boot.netboot.xyz/ipxe/netboot.xyz.lkrn -P /srv/tftp
```

Configuramos PXE para arrancar Netboot.xyz.

Editamos el archivo `/srv/tftp/pxelinux.cfg/default` :

```
DEFAULT netboot.xyz  
KERNEL netboot.xyz.lkrn
```

4. Reiniciamos los servicios

Reiniciamos los servicios de DHCP y TFTP:

```
sudo systemctl restart isc-dhcp-server tftpd-hpa
```

Pruebas

pruebas aqui

2. Despliegue de sistemas personalizados con scripts preconfigurados

Requerimientos

- **Hardware:** Servidor o máquina virtual para alojar los scripts de instalación.
- **Software:**
 - **Kickstart** (CentOS/RedHat) o **Preseed** (Debian/Ubuntu) para automatizar la instalación.
 - Servidor HTTP/FTP para alojar los archivos de configuración.
 - Opcional: PXE si queremos combinarlo con la instalación remota sin intervención manual.

Pasos de Implementación

1. Creación del Script Preseed (para Debian/Ubuntu):

- Generamos un archivo `preseed.cfg` que contendrá las respuestas para la instalación automática:

```
d-i locale string en_US
d-i keyboard-configuration/xkb-keymap select us
d-i netcfg/get_hostname string myhostname
d-i passwd/root-password password mypassword
d-i partman-auto/method string regular
d-i grub-installer/only_debian boolean true
```

- Guardamos este archivo en un servidor HTTP o en una ruta accesible por PXE.

2. Creación del Script Kickstart (para CentOS/RedHat)

- Creamos un archivo `ks.cfg` con configuraciones específicas:

```
lang en_US.UTF-8
keyboard us
network --bootproto=dhcp
rootpw --iscrypted $6$.....
firewall --disabled
part / --fstype ext4 --size=1 --grow
reboot
```

- Al igual que con Preseed, subimos este archivo a un servidor accesible.

3. Integración con Instalaciones PXE o ISO:

- Si usamos PXE o queremos modificar una ISO personalizada, añadimos una referencia al script de configuración durante el arranque:

```
append initrd=initrd.gz auto
preseed/url=http://<servidor>/preseed.cfg
```

- Para Kickstart:

```
append initrd=initrd.img ks=http://<servidor>/ks.cfg
```

4. Ajustes Personalizados:

- En ambos casos, los archivos de configuración permiten personalizar detalles como la selección de paquetes, la configuración de la red, usuarios predeterminados, y particiones del disco, adaptando el sistema operativo al entorno deseado.

3. Probar Distribuciones en Vivo sin Instalación mediante PXE

Requerimientos

- **Hardware:**
 - Servidor que actúe como PXE, preferiblemente con buena capacidad de red.
 - Clientes PXE (máquinas que bootearemos desde la red).
- **Software:**
 - **Netboot.xyz:** Permite bootear diferentes distribuciones de Linux directamente en modo live.
 - **Servidor TFTP y DHCP:** Ya configurados para el arranque PXE.

Pasos de Implementación

1. Configuración de Netboot.xyz:

- Como ya tenemos Netboot.xyz configurado, este paso consiste en usar su menú para elegir la distribución Linux que queremos probar.

- En el cliente, seleccionamos la opción **Linux Distributions** en el menú de Netboot.xyz.
- Elegimos la distribución deseada, como **Ubuntu**, **Fedora**, **Debian**, entre otras, y seleccionamos la opción "Live" si está disponible.

2. Boot en Modo Live:

- Netboot.xyz descargará la imagen live del sistema operativo seleccionado directamente desde internet o desde un servidor HTTP local (si está configurado).
- La máquina cliente arrancará en modo "live" sin necesidad de instalación, permitiéndonos probar la distribución.
- Esto funciona sin escribir en el disco duro, lo que permite cambiar de distribución de forma rápida.

3. Acceso a las Distribuciones:

- Probamos las diferentes características de la distribución, como su entorno de escritorio, aplicaciones preinstaladas, y compatibilidad con el hardware.
- Ya que estamos en modo live, no se requiere intervención adicional ni configuración específica para este tipo de pruebas.

Recuperación de Sistemas con Herramientas Live usando Netboot.xyz

Requerimientos

- **Hardware:**

- Servidor PXE previamente configurado.
- Máquina cliente afectada que necesita recuperación.
- **Software:**
 - **Netboot.xyz:** Ya configurado para proporcionar acceso a distribuciones live y herramientas de recuperación.
 - Herramientas de recuperación como **Rescuezilla** o **SystemRescue**, accesibles desde Netboot.xyz.

Pasos de Implementación

1. Acceso a Netboot.xyz en el Cliente Afectado:

- En la máquina afectada, iniciamos desde PXE usando el servidor previamente configurado.
- En el menú de Netboot.xyz, seleccionamos la opción **Utilities**.

2. Selección de Herramienta de Recuperación:

- Elegimos una herramienta de recuperación, como **Rescuezilla** o **SystemRescue**, desde el menú.
- Si necesitamos reparar particiones o recuperar datos de un disco dañado, seleccionamos **Rescuezilla** o una herramienta similar.
- Estas herramientas se cargarán en modo live, sin necesidad de instalación.

3. Ejecutar Tareas de Recuperación:

- Una vez que la herramienta live esté funcionando, usamos sus utilidades para realizar tareas de recuperación:
 - **Rescuezilla:** Permite hacer copias de seguridad, clonar discos y recuperar datos.
 - **SystemRescue:** Ofrece herramientas como `fsck` para reparar sistemas de archivos, `parted` para manejar particiones y `testdisk` para recuperar particiones dañadas.

- También podemos montar discos, revisar particiones corruptas y copiar archivos críticos.

4. Reparación de Sistemas:

- Ejecutamos diagnósticos y reparamos errores de particiones, sistemas de archivos dañados o recuperamos datos críticos. Si es necesario, copiamos los datos recuperados a un dispositivo externo.

Problemas Comunes

- **Compatibilidad de Hardware:** A veces, las herramientas de recuperación no reconocen correctamente el hardware de la máquina afectada. En este caso, probamos con otras herramientas como **SystemRescue**.
- **Errores graves en el disco:** En caso de daños irreparables en el disco, es posible que no se pueda recuperar toda la información, por lo que hacemos copias de seguridad de los datos accesibles lo antes posible

5. Automatización de instalaciones con preconfiguración (PXE + Kickstart/Preseed)

Requerimientos

- **Hardware:**
 - Un servidor PXE configurado.
 - Máquinas clientes en las que automatizaremos la instalación.
- **Software:**
 - **PXE:** Para iniciar las máquinas cliente a través de la red.

- **Kickstart** (CentOS/RedHat) o **Preseed** (Debian/Ubuntu) para automatizar la instalación con configuraciones predefinidas.
- **Servidor HTTP o TFTP**: Para almacenar y acceder a los scripts de preconfiguración.

Pasos de Implementación

1. Configuración del Servidor PXE:

- Ya habíamos configurado el servidor PXE para arrancar distribuciones desde la red usando Netboot.xyz. Ahora lo extendimos para incluir scripts de instalación automática usando Kickstart o Preseed.
- Aseguramos que el servidor TFTP y DHCP ya estaban activos y funcionando, como se hizo en pasos anteriores.

2. Creación del Archivo Preseed (para Debian/Ubuntu):

- Preparamos el archivo `preseed.cfg` que contiene todas las respuestas predefinidas para la instalación automática:

```
d-i debian-installer/language string en
d-i console-setup/ask_detect boolean false
d-i time/zone string UTC
d-i passwd/root-password password mypassword
d-i partman-auto/method string lvm
```

- Colocamos este archivo en un servidor HTTP o TFTP accesible por el servidor PXE.

3. Creación del Archivo Kickstart (para CentOS/RedHat):

- Creamos el archivo `ks.cfg` para CentOS/RedHat, definiendo configuraciones como particiones, paquetes y usuarios:

```
install
url --url="http://mirror.centos.org/centos/7/os/x86_64/"
lang en_US.UTF-8
keyboard us
network --bootproto=dhcp
rootpw --plaintext mypassword
reboot
```

- Subimos el archivo `ks.cfg` a un servidor HTTP o TFTP para que sea accesible por las máquinas clientes durante la instalación.

4. Configuración del Servidor PXE para Usar Kickstart/Preseed:

- Modificamos el archivo de configuración PXE para hacer referencia a los scripts Kickstart o Preseed:

- Para Preseed:

```
append initrd=initrd.gz auto
preseed/url=http://<servidor>/preseed.cfg
```

- Para Kickstart:

```
append initrd=initrd.img ks=http://<servidor>/ks.cfg
```

5. Automatización Completa de la Instalación:

- Las máquinas cliente arrancan desde PXE, y el servidor PXE proporciona el script Preseed o Kickstart. Este script se encarga de ejecutar la instalación del sistema operativo de manera totalmente desatendida, configurando automáticamente las particiones, paquetes y ajustes de red.
- El sistema se instalará sin intervención manual siguiendo las configuraciones predefinidas.

Problemas Comunes

- **Acceso a los scripts:** Nos aseguramos de que los archivos `preseed.cfg` o `ks.cfg` estén correctamente accesibles desde la red (HTTP o TFTP).
 - **Errores en los scripts:** Revisamos que no haya errores en los archivos de preconfiguración, ya que un error en el script puede interrumpir la instalación.
-

6. Implementación de entornos efímeros

Bootear sistemas operativos que viven en RAM y desaparecen al apagar la máquina, ideal para entornos temporales o de prueba. (~4 - 5 horas)

Requerimientos

- Hardware
 - Un servidor PXE configurado.
 - Clientes que van a bootear desde PXE.
- Software
 - **Netboot.xyz:** Para iniciar los sistemas operativos directamente en RAM.
 - Distribuciones Linux que soporten ejecución desde RAM, como **Tiny Core Linux** o **Porteus**.
 - Servidor TFTP y DHCP configurados para el arranque PXE.

Pasos de Implementación

1. Configuración del Servidor PXE**:

- Como el servidor PXE ya está configurado con Netboot.xyz, utilizamos la opción de bootear sistemas desde la red en modo live.

- Accedemos al menú de **Netboot.xyz** en los clientes y seleccionamos una distribución que soporte la ejecución desde RAM, como **Tiny Core Linux** o **Porteus**.

2. Selección de Distribución para Ejecutar en RAM**:

- En el menú de Netboot.xyz, seleccionamos una distribución ligera que pueda ejecutarse completamente desde la RAM.
- Estas distribuciones cargan todo el sistema operativo en la memoria del cliente, lo que permite un entorno temporal que no escribe en el disco duro.

3. Arranque del Sistema Efímero**:

- Una vez seleccionada la distribución, la máquina cliente descarga la imagen en RAM y arranca directamente desde allí.
- El sistema vive completamente en la RAM del cliente. Una vez que la máquina se apaga o reinicia, todos los cambios se pierden, lo que lo hace ideal para pruebas o entornos temporales.

4. Uso de Entornos Temporales**:

- Durante la sesión en RAM, podemos utilizar el sistema operativo con todas sus capacidades: instalar paquetes, probar configuraciones, y ejecutar scripts sin afectar el disco duro de la máquina.
- Ideal para pruebas rápidas o configuraciones temporales donde no es necesario guardar el estado del sistema.

5. Apagado y Desaparición del Sistema**:

- Cuando terminamos la sesión, simplemente apagamos o reiniciamos la máquina cliente. Como todo el sistema estaba en la RAM, no se guarda nada y el entorno desaparece, volviendo a su estado original.

Problemas Comunes

- **Limitación de RAM:** Algunas máquinas pueden no tener suficiente RAM para cargar sistemas operativos más grandes. En este caso, usamos distribuciones más ligeras como Tiny Core.
 - **Pérdida de datos no deseada:** Al ser entornos efímeros, cualquier cambio o dato se perderá al apagar. Para mantener datos importantes, es recomendable utilizar dispositivos de almacenamiento externos si es necesario.
-

7. Instalación de contenedores directamente desde la red

Prepara un entorno mínimo que cargue contenedores como Docker o Podman desde la red, sin un sistema operativo completo. (~6 - 7 horas)

Requerimientos

- Hardware
 - Servidor PXE configurado.
 - Máquinas cliente que van a ejecutar contenedores desde la red.
- Software
 - **Netboot.xyz:** Para bootear los clientes desde la red.
 - **Docker** o **Podman:** Para ejecutar contenedores.
 - Distribuciones ligeras que soporten contenedores, como **Alpine Linux** o **RancherOS**.
 - Servidor TFTP y DHCP configurados para el arranque PXE.

Pasos de Implementación

1. Configuración del Servidor PXE:

- Ya teníamos el servidor PXE funcionando con Netboot.xyz, así que aprovechamos este entorno para bootear una distribución mínima desde la red que sea capaz de ejecutar contenedores, como **Alpine Linux** o **RancherOS**.
- En el cliente, seleccionamos la distribución adecuada desde el menú de **Netboot.xyz**.

2. Arranque de la Distribución Mínima:

- Seleccionamos **Alpine Linux** o **RancherOS** en el menú de Netboot.xyz, ambas opciones son muy ligeras y están diseñadas para funcionar bien con contenedores.
- La máquina cliente descarga la imagen mínima en RAM, arrancando sin un sistema operativo completo, pero con las herramientas necesarias para correr contenedores.

3. Instalación de Docker o Podman:

- Después del arranque, instalamos Docker o Podman directamente desde el repositorio oficial de la distribución:
 - Instalación de Docker en Alpine

```
apk add docker
rc-update add docker boot
service docker start
```

- Instalación de Podman en Alpine

```
apk add podman
```

- En RancherOS, Docker ya está preinstalado, por lo que solo lo arrancamos:

```
sudo system-docker start
```

4. Cargar Contenedores desde la Red:

- Con Docker o Podman ya instalados, procedemos a ejecutar contenedores desde la red.
- Ejemplo para Docker:

```
docker run -d --name webserver -p 80:80 nginx
```

- Ejemplo para Podman:

```
podman run -d --name webserver -p 80:80 nginx
```

- Podemos cargar y ejecutar cualquier contenedor disponible en Docker Hub u otros repositorios.

5. Uso de Contenedores en un Entorno Mínimo**:

- Las máquinas cliente ahora están ejecutando contenedores directamente en un entorno mínimo sin la sobrecarga de un sistema operativo completo.
- Esto es ideal para pruebas o despliegues rápidos sin necesidad de configurar un sistema operativo tradicional.

Problemas Comunes

- **Acceso a Internet:** Aseguramos que las máquinas cliente tengan acceso a la red para descargar imágenes de contenedores.
 - **Limitaciones de RAM:** Si la máquina cliente tiene poca RAM, puede que no sea capaz de manejar múltiples contenedores pesados.
 - **Compatibilidad de Contenedores:** Algunas distribuciones ligeras pueden no soportar todos los contenedores disponibles. En este caso, optamos por contenedores más ligeros o adaptamos los scripts de arranque.
-

8. Creación de un servidor PXE personalizado

Configura tu propio servidor PXE con Netboot.xyz para tener un control total sobre qué distribuciones y herramientas están disponibles para bootear. (~10 - 12 horas)

Requerimientos

- Hardware
 - Un servidor dedicado o máquina virtual para actuar como servidor PXE.
 - Clientes que van a bootear desde la red.
- Software
 - **Netboot.xyz**: Permite personalizar las distribuciones y herramientas de arranque.
 - **Servidor DHCP y TFTP**: Para gestionar la asignación de IPs y la transferencia de archivos de arranque.
 - **Servidor HTTP o FTP**: Opcional, para alojar imágenes personalizadas de distribuciones y herramientas.

Pasos de Implementación

1. Instalación del Servidor DHCP y TFTP**:

- Instalamos los servicios necesarios para que el servidor actúe como PXE.

```
sudo apt install isc-dhcp-server tftpd-hpa
```

- Configuramos el archivo de DHCP

```
/etc/dhcp/dhcpd.conf
```

para asignar IPs a las máquinas clientes y especificar el servidor TFTP:

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.100 192.168.1.200;  
    option routers 192.168.1.1;  
    filename "pxelinux.0";  
    next-server 192.168.1.10;  
}
```

- Creamos y configuramos la carpeta del servidor TFTP:

```
Copy code  
sudo mkdir -p /srv/tftp  
sudo chown -R tftp:tftp /srv/tftp
```

2. Descarga de Netboot.xyz e Integración con PXE**:

- Descargamos la imagen de Netboot.xyz y la colocamos en el directorio TFTP:

```
wget https://boot.netboot.xyz/ipxe/netboot.xyz.lkrn -P /srv/tftp
```

- Creamos el archivo de configuración PXE para que las máquinas cliente puedan arrancar desde Netboot.xyz:

```
DEFAULT netboot.xyz  
LABEL netboot.xyz  
    KERNEL netboot.xyz.lkrn  
    APPEND initrd=
```

- Reiniciamos los servicios de TFTP y DHCP:

```
sudo systemctl restart isc-dhcp-server tftpd-hpa
```

3. Personalización de las Distribuciones y Herramientas Disponibles**:

- Para tener control total sobre las distribuciones y herramientas que estarán disponibles, editamos el archivo de configuración de **Netboot.xyz**. Esto nos permite agregar o eliminar distribuciones según nuestras necesidades.
- Si queremos ofrecer imágenes personalizadas o distribuciones que no estén en Netboot.xyz, las descargamos y las alojamos en un servidor HTTP/FTP local. Luego, agregamos referencias a esas imágenes en el menú de arranque.

Ejemplo de una entrada personalizada en el menú de arranque:

```
LABEL custom_ubuntu
    KERNEL http://<servidor>/images/ubuntu/vmlinuz
    INITRD http://<servidor>/images/ubuntu/initrd.img
    APPEND root=/dev/ram0 ramdisk_size=1500000 ip=dhcp
```

4. Integración de Herramientas Personalizadas**:

- Agregamos herramientas de recuperación o diagnósticos específicas, como **Clonezilla**, **GParted**, o imágenes ISO personalizadas, en el menú de Netboot.xyz.
- Colocamos las herramientas en un servidor accesible (HTTP o FTP) y referenciamos las rutas en el archivo de configuración PXE:

```
LABEL clonezilla
    KERNEL http://<servidor>/tools/clonezilla/live/vmlinuz
    INITRD http://<servidor>/tools/clonezilla/live/initrd.img
    APPEND boot=live union=overlay
```

5. Pruebas de Arranque y Validación**:

- Probamos el arranque desde PXE en una máquina cliente para asegurarnos de que las distribuciones personalizadas y herramientas funcionan correctamente.
- Verificamos que las imágenes carguen desde el servidor TFTP o HTTP y que las configuraciones sean correctas para cada distribución.
- Validamos que podemos cambiar, agregar o eliminar distribuciones según lo necesitemos sin problemas.

6. Documentación y Mantenimiento:

- Documentamos todas las configuraciones realizadas y creamos una lista de las distribuciones y herramientas disponibles en el menú de arranque.
- Mantenemos el servidor actualizado y listo para agregar nuevas distribuciones o herramientas según sea necesario.

Problemas Comunes

- **Errores en el archivo de configuración PXE:** Verificamos cuidadosamente los caminos y nombres de archivos, ya que un pequeño error en la configuración puede causar que las distribuciones no arranquen correctamente.
 - **Compatibilidad de hardware:** Algunos clientes pueden no soportar todas las distribuciones. En este caso, ajustamos las configuraciones de arranque o cambiamos a distribuciones más ligeras.
 - **Lentitud en la carga de imágenes:** Si las imágenes se están descargando desde internet, la carga puede ser lenta. En este caso, usamos un servidor HTTP local para acelerar el proceso.
-

9. Instalación y mantenimiento de sistemas remotos

Gestiona la instalación y mantenimiento de servidores remotos usando Netboot.xyz para resolver problemas de sistemas sin intervención física. (~5 - 6 horas)

Requerimientos

- Hardware
 - Servidor PXE configurado en una ubicación central.

- Servidores o máquinas remotas conectadas a la red y configuradas para arrancar desde PXE.
- Software
 - **Netboot.xyz**: Para realizar instalaciones y mantenimientos de sistemas operativos de forma remota.
 - **Kickstart** (CentOS/RedHat) o **Preseed** (Debian/Ubuntu) para automatizar instalaciones sin intervención física.
 - **SSH** y herramientas de administración remota para el mantenimiento posterior a la instalación.

Pasos de Implementación

1. Configuración del Servidor PXE para Sistemas Remotos**:

- Ya habíamos configurado el servidor PXE con Netboot.xyz, lo que nos permite usarlo para gestionar instalaciones de servidores remotos.
- Configuramos las máquinas remotas para que puedan arrancar desde PXE accediendo al servidor central. Esto puede implicar ajustar las opciones de red o BIOS para habilitar el arranque PXE.

2. ####Automatización de la Instalación Remota:

- Preparamos scripts de preconfiguración como **Kickstart** o **Preseed** para eliminar la intervención manual durante la instalación de los servidores remotos.
- Para CentOS/RedHat, creamos un archivo

```
ks.cfg
```

con las configuraciones necesarias:

```
install
url --url="http://<servidor>/centos/7/os/x86_64/"
lang en_US.UTF-8
keyboard us
network --bootproto=dhcp
rootpw --plaintext mypassword
reboot
```

- Para Debian/Ubuntu, configuramos el archivo

```
preseed.cfg
```

con opciones similares:

```
d-i debian-installer/locale string en_US
d-i netcfg/get_hostname string remote-server
d-i passwd/root-password password mypassword
d-i partman-auto/method string regular
```

- Estos archivos de configuración se colocan en un servidor HTTP o TFTP accesible por las máquinas remotas.

3. Arranque y Despliegue en los Servidores Remotos:

- Las máquinas remotas arrancan desde PXE y automáticamente comienzan la instalación del sistema operativo utilizando los archivos de configuración **Kickstart** o **Preseed**.
- La instalación ocurre sin intervención física, y una vez completada, el sistema se reinicia y queda listo para ser administrado de forma remota.

4. Mantenimiento de los Sistemas Remotos:

- Después de la instalación, accedemos a los servidores remotos a través de **SSH** para realizar cualquier ajuste adicional o tareas de mantenimiento.
- Utilizamos herramientas como **Ansible** o **Bash scripts** para automatizar el mantenimiento de los sistemas, como actualizaciones, monitoreo y backup.
- En caso de fallos críticos o sistemas dañados, volvemos a utilizar **Netboot.xyz** para reiniciar el servidor en modo live y ejecutar herramientas de recuperación remota como **Rescuezilla** o **SystemRescue**.

5. Reinstalación o Recuperación Remota**:

- Si necesitamos reinstalar o recuperar un servidor remoto, Netboot.xyz nos permite reiniciar el servidor afectado desde la red, iniciar una distribución live, y reparar el sistema o reinstalarlo sin la necesidad de intervención física.
- Usamos herramientas de diagnóstico y recuperación, como **fsck**, **parted**, y **dd**, para resolver problemas de disco o sistema de archivos.

Problemas Comunes

- **Conectividad de Red:** Es esencial que las máquinas remotas tengan acceso constante a la red y que no existan problemas de latencia o pérdida de conexión durante el proceso de instalación.
 - **Fallo en los Scripts de Preconfiguración:** Un error en los archivos **Kickstart** o **Preseed** puede interrumpir la instalación. Verificamos bien todos los parámetros antes de implementarlos.
 - **Acceso a los Servidores:** En algunos casos, puede ser necesario tener acceso a un sistema de administración remota fuera de banda (como iLO o IPMI) para reiniciar los servidores o ajustar configuraciones de BIOS.
-

10. Bootear imágenes ISO personalizadas desde la red

Modifica Netboot.xyz para agregar tus propias imágenes ISO personalizadas y bootearlas desde el menú.

Requerimientos

- Hardware
 - Un servidor PXE configurado con Netboot.xyz.
 - Clientes capaces de bootear desde PXE.
- Software
 - **Netboot.xyz**: Ya configurado para arranque por red.
 - Imágenes ISO personalizadas que desees bootear.
 - **Servidor HTTP o FTP** para alojar las imágenes ISO personalizadas.

Pasos de Implementación

1. Configuración del Servidor PXE:

- Como ya tenemos un servidor PXE con **Netboot.xyz** configurado, nos enfocamos en modificarlo para soportar el booteo de imágenes ISO personalizadas.
- Verificamos que los servicios **TFTP**, **DHCP** y **HTTP/FTP** estén funcionando correctamente para gestionar las solicitudes de arranque de las máquinas clientes.

2. Subida de las Imágenes ISO Personalizadas:

- Subimos las imágenes ISO personalizadas a un servidor

HTTP

o

FTP

accesible por los clientes PXE. Este servidor puede estar en la misma máquina que el servidor PXE o en otra máquina de la red.

```
sudo mkdir -p /var/www/html/isos  
sudo cp /ruta/a/mi_imagen.iso /var/www/html/isos/
```

3. Modificación del Menú de Netboot.xyz**:

- Editamos el archivo de configuración PXE para incluir nuestras imágenes ISO personalizadas en el menú de arranque.
- Modificamos el archivo de configuración PXE

```
/srv/tftp/pxelinux.cfg/default
```

o el equivalente según la configuración:

```
LABEL custom_iso  
MENU LABEL Boot Custom ISO  
KERNEL memdisk  
INITRD http://<servidor>/isos/mi_imagen.iso  
APPEND iso raw
```

- Esta entrada le indica al cliente que cargue la imagen ISO personalizada desde el servidor HTTP o FTP usando la herramienta **memdisk**, que permite bootear imágenes ISO directamente desde la red.

4. Prueba del Arranque de la ISO Personalizada:

- Desde un cliente PXE, arrancamos la máquina y accedemos al menú de **Netboot.xyz**.

- Seleccionamos la opción **Boot Custom ISO** que configuramos en el archivo de menú y verificamos que la imagen personalizada cargue correctamente.
- La imagen ISO se descargará desde el servidor HTTP/FTP y se ejecutará en el cliente sin necesidad de instalación en disco.

5. Personalización Adicional:

- Si queremos ofrecer varias imágenes ISO personalizadas, simplemente repetimos el proceso para cada ISO y agregamos más entradas al archivo de configuración PXE.
- Podemos incluir distribuciones personalizadas de Linux, herramientas de diagnóstico o cualquier otra imagen que queramos cargar remotamente.

Ejemplo de varias imágenes ISO en el menú:

```
LABEL custom_ubuntu
    MENU LABEL Boot Custom Ubuntu ISO
    KERNEL memdisk
    INITRD http://<servidor>/isos/ubuntu_custom.iso
    APPEND iso raw

LABEL custom_centos
    MENU LABEL Boot Custom CentOS ISO
    KERNEL memdisk
    INITRD http://<servidor>/isos/centos_custom.iso
    APPEND iso raw
```

6. Validación del Funcionamiento:

- Probamos el booteo de cada imagen ISO personalizada en diferentes clientes para asegurarnos de que todas las imágenes carguen correctamente y sin problemas.
- Verificamos la integridad y funcionalidad de las imágenes ISO en el entorno live o instalación, según sea el propósito de cada imagen.

Problemas Comunes

- **Errores de Red:** Aseguramos que la red sea lo suficientemente rápida para cargar las imágenes ISO de manera eficiente. Si la red es lenta, la descarga de la ISO puede tardar demasiado.
 - **Tamaño de las Imágenes ISO:** Si las imágenes ISO son muy grandes, podemos encontrar limitaciones de memoria en los clientes. En estos casos, usamos distribuciones ligeras o ajustamos el tamaño de la ISO.
 - **Errores en la Configuración de PXE:** Un error en las rutas de los archivos o en las opciones de arranque puede causar fallos al cargar la ISO. Revisamos cuidadosamente las configuraciones para evitar problemas.
-

11. Cargar entornos de virtualización ligera

Configura un sistema que cargue herramientas de virtualización como QEMU o KVM desde la red, sin necesidad de un sistema operativo anfitrión completo. (~5 - 6 horas)

12. Soporte para instalar sistemas operativos no Linux

Bootear instaladores de sistemas como Windows o FreeDOS desde Netboot.xyz, facilitando la instalación de estos sistemas a través de la red. (~3 - 4 horas)

13. Cargar entornos de pruebas de seguridad (pentesting)

Inicia distribuciones de pentesting como Kali Linux o Parrot OS directamente desde la red para auditorías de seguridad rápidas y sin instalación local. (~2 - 3 horas)

14. Monitoreo y análisis de red

Cargar herramientas como Ntopng o Wireshark para monitorear el tráfico de red y diagnosticar problemas desde un entorno ligero. (~4 - 5 horas)

Requerimientos

- Hardware
 - Un servidor PXE configurado.
 - Máquinas cliente con soporte para virtualización de hardware (VT-x o AMD-V habilitado en la BIOS).
- Software
 - **Netboot.xyz**: Para iniciar desde la red y cargar entornos de virtualización ligera.
 - Herramientas de virtualización ligera como **QEMU** o **KVM**.
 - **Alpine Linux**, **Tiny Core Linux**, o **RancherOS**, que son distribuciones ligeras compatibles con virtualización.

Pasos de Implementación

1. Configuración del Servidor PXE**:

- Como ya tenemos un servidor PXE con Netboot.xyz configurado, lo utilizamos para cargar un entorno ligero que soporte virtualización sin un sistema operativo anfitrión completo.
- Nos aseguramos de que los servicios de **TFTP**, **DHCP**, y **HTTP/FTP** estén activos y correctamente configurados.

2. Elección de una Distribución Ligera con Soporte de Virtualización**:

- Elegimos una distribución ligera, como **Alpine Linux** o **RancherOS**, que sea compatible con herramientas de virtualización como **QEMU** o **KVM**.
- Descargamos la distribución elegida y subimos la imagen al servidor HTTP/FTP accesible:

```
wget https://dl-cdn.alpinelinux.org/alpine/v3.14/releases/x86_64/alpine-standard-3.14.2-x86_64.iso -P /var/www/html/isos
```

3. Modificación del Menú de Netboot.xyz**:

- Editamos el archivo de configuración PXE para agregar una opción que cargue la distribución ligera desde la red:

```
LABEL alpine_virtualization
  MENU LABEL Boot Alpine Linux with QEMU/KVM
  KERNEL http://<servidor>/isos/alpine-standard-3.14.2-x86_64.iso
  INITRD initrd.img
  APPEND root=/dev/ram0 ramdisk_size=1500000 ip=dhcp
```

- Esta configuración permite bootear Alpine Linux desde el servidor para usar QEMU o KVM sin necesidad de un sistema operativo anfitrión completo.

4. Instalación y Configuración de QEMU o KVM**:

- Una vez que la máquina cliente ha arrancado desde la red con Alpine Linux

QEMU

KVM

para gestionar entornos virtualizados:

- Instalación de QEMU

```
apk add qemu-system-x86_64 qemu-img
```

- Instalación de KVM

```
apk add qemu-kvm  
modprobe kvm_intel
```

- Esto nos permite crear y gestionar máquinas virtuales desde un entorno ligero sin un sistema operativo anfitrión completo.

5. Ejecución de Máquinas Virtuales en el Cliente:

- Con QEMU o KVM instalado, podemos ejecutar máquinas virtuales directamente en la máquina cliente desde el entorno ligero.
- Ejemplo de cómo crear y ejecutar una máquina virtual con QEMU:

```
qemu-system-x86_64 -hda /ruta/disco_virtual.img -m 1024 -boot d -  
cdrom /ruta/imagen.iso
```

- Esta configuración permite a los clientes cargar y ejecutar entornos virtualizados sin la necesidad de un sistema operativo anfitrión completo.

6. Mantenimiento y Gestión de las Máquinas Virtuales**:

- Las máquinas virtuales creadas pueden gestionarse y mantenerse usando las herramientas estándar de QEMU o KVM, y el entorno se puede volver a cargar fácilmente desde la red en caso de que necesitemos restablecer o actualizar los entornos.

Problemas Comunes

- **Soporte de Hardware para Virtualización:** Aseguramos que las máquinas cliente tengan habilitado el soporte de virtualización de hardware en la BIOS (VT-x o AMD-V).
- **Limitaciones de RAM:** Las máquinas cliente deben tener suficiente memoria para ejecutar tanto el entorno ligero como las máquinas virtuales.
- **Acceso a las Imágenes ISO:** Las imágenes de los sistemas operativos a instalar en las máquinas virtuales deben estar accesibles en el cliente o en un servidor de red.

Pruebas y Validación

- Probamos el arranque de **Alpine Linux** o **RancherOS** en varias máquinas cliente para asegurarnos de que el entorno ligero se cargue correctamente y las herramientas de virtualización estén operativas.
- Validamos la creación y ejecución de máquinas virtuales con **QEMU** y **KVM** desde el entorno ligero, verificando el rendimiento y la estabilidad de las VM.

Tiempo Estimado

- Este proceso de configuración y prueba toma entre 5 y 6 horas, dependiendo del número de clientes y la complejidad de las máquinas virtuales que queramos ejecutar.

15. Implementación de entornos de desarrollo temporal

Configura entornos mínimos de desarrollo que se carguen desde la red, proporcionando rápidamente herramientas de desarrollo sin configuración adicional. (~4 - 5 horas)

Requerimientos

- Hardware
 - Servidor PXE previamente configurado.
 - Máquinas cliente que cargarán los entornos de desarrollo desde la red.
- Software
 - **Netboot.xyz**: Para gestionar el arranque de las máquinas cliente desde la red.
 - Distribuciones ligeras como **Alpine Linux** o **Tiny Core Linux**, con herramientas de desarrollo preinstaladas o listas para instalar.
 - Servidor HTTP o FTP para alojar scripts de configuración o imágenes ISO personalizadas, si es necesario.

Pasos de Implementación

1. Configuración del Servidor PXE:

- Utilizamos el servidor PXE ya configurado con Netboot.xyz para arrancar distribuciones ligeras que actúan como entornos de desarrollo temporales.
- Nos aseguramos de que los servicios **TFTP**, **DHCP**, y **HTTP/FTP** estén funcionando correctamente y accesibles por las máquinas cliente.

2. Selección de una Distribución Ligera:

- Elegimos una distribución ligera como **Alpine Linux** o **Tiny Core Linux** que pueda ser fácilmente configurada para funcionar como un entorno de desarrollo temporal.
- Si es necesario, creamos una imagen ISO personalizada que incluya las herramientas de desarrollo que queremos cargar, como **GCC**, **Python**, **Git**, y **Vim**.

3. Modificación del Menú de Netboot.xyz:

- Editamos el archivo de configuración PXE para agregar una entrada que cargue el entorno de desarrollo temporal:

```
LABEL alpine_dev
  MENU LABEL Boot Alpine Linux for Development
  KERNEL http://<servidor>/isos/alpine-standard.iso
  INITRD initrd.img
  APPEND root=/dev/ram0 ramdisk_size=1500000 ip=dhcp
```

- También podemos incluir scripts que instalen automáticamente las herramientas de desarrollo al arrancar el sistema, lo que permite tener un entorno listo sin intervención adicional.

4. Carga del Entorno de Desarrollo en el Cliente:

- Las máquinas cliente arrancan desde PXE y cargan la distribución ligera que hemos configurado.
- Una vez en el entorno de desarrollo temporal, instalamos rápidamente las herramientas necesarias:
 - Instalación de herramientas en Alpine Linux

```
apk add gcc g++ make python3 git vim
```

- Tiny Core Linux

también ofrece la posibilidad de instalar herramientas de desarrollo mediante su gestor de paquetes:

```
tce-load -wi compiletc python3 git vim
```

5. Personalización del Entorno de Desarrollo:

- Podemos incluir scripts adicionales para configurar el entorno de desarrollo según las necesidades del proyecto. Por ejemplo, podemos cargar automáticamente repositorios de Git, configurar entornos virtuales de Python, o preparar compiladores específicos.

6. Uso y Mantenimiento del Entorno:

- El entorno de desarrollo temporal permite a los desarrolladores trabajar sin necesidad de instalar ni configurar nada en los discos locales de las máquinas cliente.
- Una vez que se termina el trabajo, el entorno se reinicia o apaga, y no se guarda ninguna configuración, lo que es ideal para sesiones temporales de desarrollo o pruebas.

Problemas Comunes

- **Limitaciones de RAM:** Dado que estos entornos viven en RAM, las máquinas cliente deben tener suficiente memoria para soportar tanto el entorno como las herramientas de desarrollo.
- **Acceso a Internet:** Aseguramos que las máquinas cliente tengan acceso a Internet o a los repositorios locales de software para descargar e instalar las herramientas de desarrollo necesarias.
- **Persistencia de Datos:** Como estos entornos son efímeros, cualquier dato importante debe guardarse en almacenamiento externo o en la red para evitar pérdidas al reiniciar.