Multi-factor authentication (MFA)

Introducción

SSH utiliza **contraseñas** para la autenticación de forma predeterminada y la mayoría de las recomendaciones para proteger los servicios de SSH recomiendan usar claves públicas/privadas. Sin embargo, una clave p/p sigue siendo solo un factor, aunque mucho más seguro. El terminal de su computadora enviará los datos a través de un túnel cifrado a la máquina remota por lo que las posibilidades de ataque son ínfimas. Pero así como un pirata informático puede adivinar una contraseña con fuerza bruta, puede también robar una clave SSH y luego, en cualquier caso, con ese único dato, un atacante puede obtener acceso a sus sistemas remotos. Esto es de gran importancia especialmente cuando utilizamos dispositivos portátiles que pueden ser comprometidos.

En este tutorial, configuraremos la autenticación multifactor para combatir eso. La autenticación multifactor (MFA) o la autenticación de dos factores (2FA) requiere más de un factor para autenticarse o iniciar sesión. Esto significa que un atacante tendría que comprometer varias cosas, como su computadora y su teléfono, para ingresar. Hay varios tipos de factores utilizados en la autenticación:

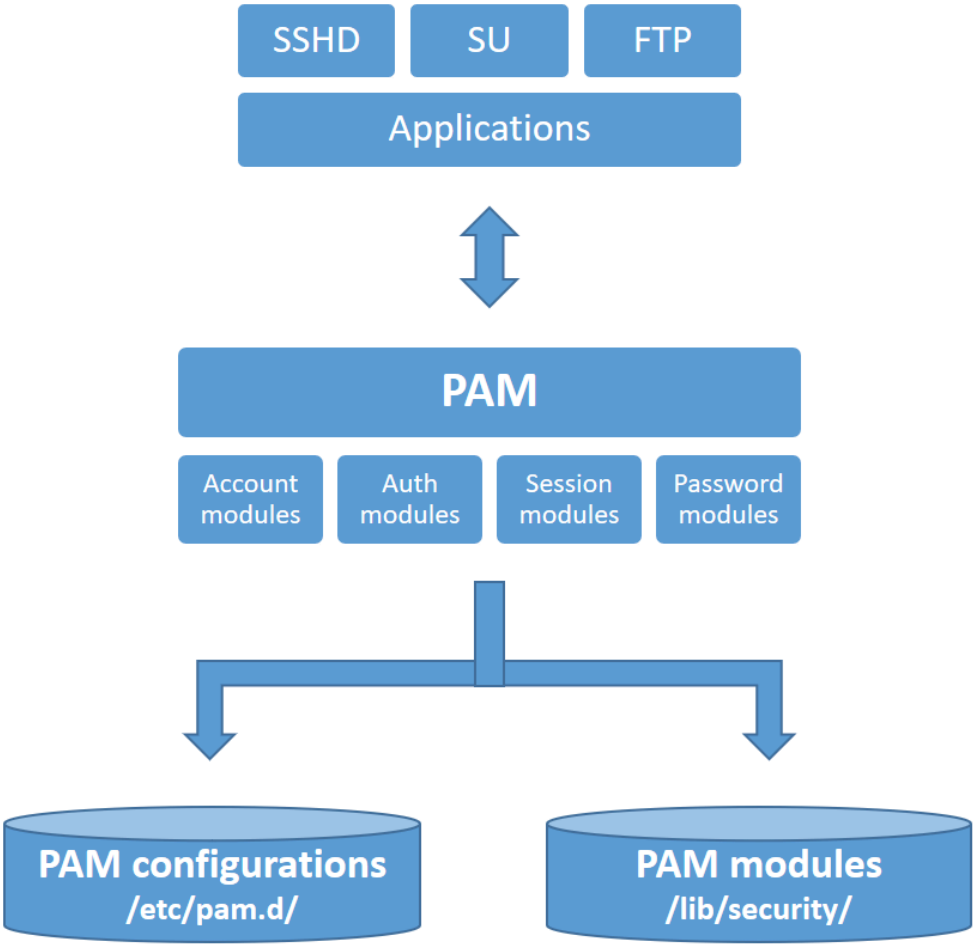
1. Algo que sepas , como una contraseña o una pregunta de seguridad
2. Algo que tengas , como una aplicación de autenticación o un token de seguridad
3. Algo que eres , como tu huella digital o tu voz.

Un factor común es una aplicación OATH-TOTP, como Google Authenticator. OATH-TOTP (contraseña de un solo uso basada en tiempo de autenticación abierta) es un protocolo abierto que genera una contraseña de un solo uso, comúnmente un número de seis dígitos que se recicla cada 30 segundos.

Este artículo explicará cómo habilitar la autenticación SSH utilizando una aplicación OATH-TOTP además de una clave SSH. Iniciar sesión en su servidor a través de SSH requerirá dos factores en dos dispositivos, lo que lo hará más seguro que una contraseña o una clave SSH por sí sola. Además, repasamos algunos casos de uso adicionales de MFA y algunos consejos y trucos útiles.

Paso 1: instalar PAM de Google

PAM, que significa *Módulo de autenticación conectable* , es una infraestructura de autenticación utilizada en sistemas Linux para autenticar a un usuario.



PAM permite el desarrollo de programas independientes del mecanismo de autenticación a utilizar. Ası es posible que un programa que aproveche las facilidades ofrecidas por PAM sea capaz de utilizar desde el sencillo /etc/passwd hasta dispositivos hardware —como lectores de huella digital—, pasando por servidores LDAP 3 o sistemas de gestión de bases de datos. Y, por supuesto, todo esto sin cambiar ni una sola línea de código.

Pero PAM va más allá todavía, permitiendo al administrador del sistema construir políticas diferentes de autenticación para cada servicio.

En resumen, podrían sintetizarse las ventajas más importantes de PAM en los siguientes puntos:

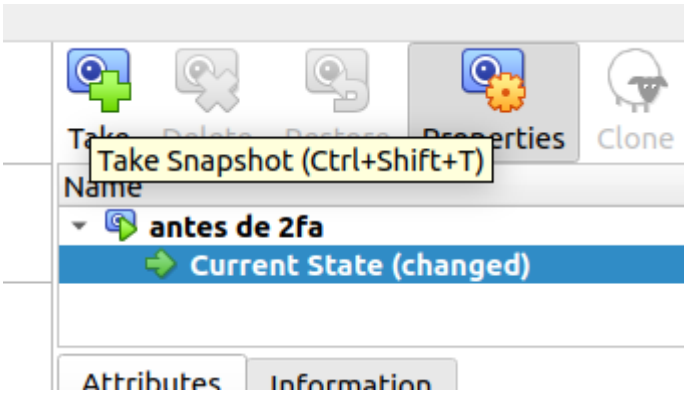
- Ofrece un esquema de autenticación común y centralizado.
- Permite a los desarrolladores abstraerse de las labores de autenticación.
- Facilita el mantenimiento de las aplicaciones.
- Ofrece flexibilidad y control tanto para el desarrollador como para el administrador de sistema

Debido a que Google creó una aplicación OATH-TOTP, también creó un PAM que genera TOTP y es totalmente compatible con cualquier aplicación OATH-TOTP, como Google Authenticator o Authy.

En este paso, instalaremos y configuraremos el PAM de Google.

Antes de continuar inicie la máquina virtual. Inicie sesión y reinicie la máquina

ATENCIÓN: Antes de comenzar realice un snapshot de su máquina virtual.



Recuerde siempre gestionar los cambios de los archivos de configuración que va a modificar.

Luego ingrese a la máquina virtual y actualice la caché del repositorios:

```
Unset
sudo apt-get update
```

Verifique que la actualización sea exitosa, en caso de que falle puede que se deba a un problema de acceso a internet. Verifique que la interfaz en modo bridge tenga acceso a internet.

A continuación, instale el PAM:

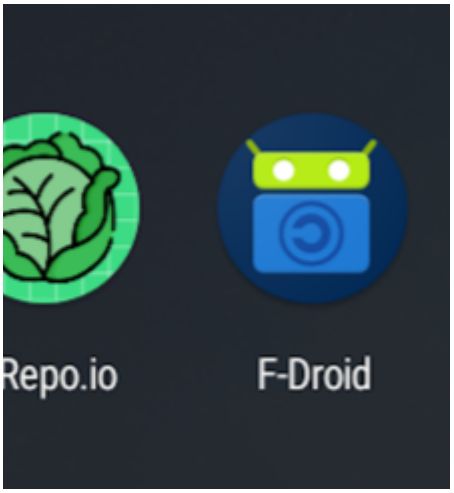
```
Unset
sudo apt-get install libpam-google-authenticator
```

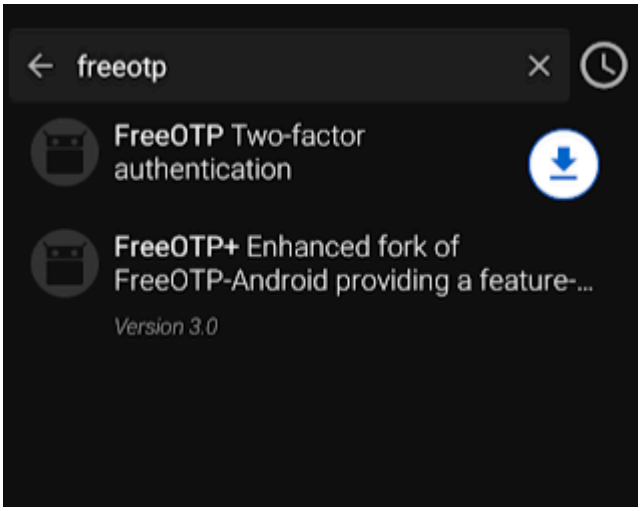
Con el PAM instalado, usaremos una aplicación auxiliar que viene con el PAM para generar una clave TOTP para el usuario que necesita un segundo factor. Esta clave se genera usuario por usuario, no en todo el sistema. Esto significa que cada usuario que quiera utilizar una aplicación de autenticación TOTP deberá iniciar sesión y ejecutar la aplicación auxiliar para obtener su clave; no puede simplemente ejecutarlo una vez para habilitarlo para todos (pero hay algunos consejos al final de este tutorial para configurar o requerir MFA para muchos usuarios).

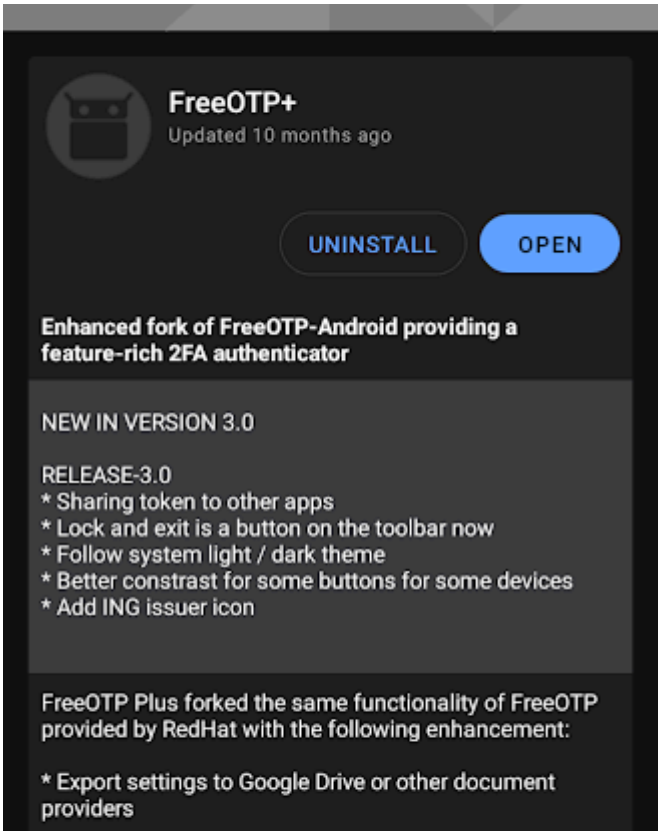
Para ello en sus teléfonos android instalen la aplicación FREEOTP+

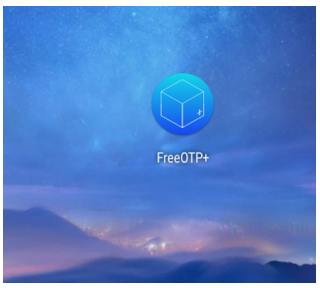
La pueden encontrar en las tiendas “oficiales” de aplicaciones y en fdroid

https://f-droid.org/es/packages/org.liberty.android.freeotpplus/









NOTA:Se sugiere además habilitar la autenticación en la aplicación. Para ello acceda al menú de la parte superior derecha y seleccione la opción correspondiente.

Ejecute la aplicación de inicialización:

```
Unset
google-authenticator
```

Después de ejecutar el comando, la aplicación le hará algunas preguntas. El primero pregunta si los tokens de autenticación deberían basarse en el tiempo:

```
Unset
Do you want authentication tokens to be time-based (y/n) y
```

Este PAM permite tokens basados en el tiempo o secuenciales. El uso *de tokens basados en secuencias* significa que el código comienza en un punto determinado y luego lo incrementa después de cada uso. El uso *de tokens basados en el tiempo* significa que el código cambia después de un período de tiempo determinado. Nos quedaremos con el tiempo porque eso es lo que anticipan aplicaciones como Google Authenticator, así que responda si.

Después de responder esta pregunta, aparecerán muchos resultados, incluido un código QR grande.

Utilice su aplicación de autenticación en su teléfono para escanear el código QR o escriba manualmente la clave secreta. Si el código QR es demasiado grande para escanearlo, puede usar la URL que se encuentra encima del código QR para obtener una versión más pequeña.

Una vez agregado, verás el nombre del servidor en tu aplicación, cuando lo presiones se generará un código de seis dígitos que cambia cada 30 segundos en tu aplicación.

Debes ingresar el primer código en la consola y presionar enter.

```
Unset

Enter code from app (-1 to skip): 551796
```

Nota : asegúrese de registrar la clave secreta, el código de verificación y los códigos de recuperación en un lugar seguro, como un administrador de contraseñas. Los códigos de recuperación son la única forma de recuperar el acceso si, por ejemplo, pierdes el acceso a tu aplicación TOTP.

Las preguntas restantes informan al PAM sobre cómo funcionar. Los repasamos uno por uno:

```
Unset
Do you want me to update your "~/.google_authenticator" file (y/n) y
```

Esto escribe la clave y las opciones en el archivo `.google_authenticator`. Si dice que no, el programa se cierra y no se escribe nada, lo que significa que el autenticador no funcionará:

```
Unset
Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it
increases your chances to notice or even prevent man-in-the-middle
attacks (y/n) y
```

Al responder sí aquí, evita un ataque de repetición al hacer que cada código caduque inmediatamente después de su uso. Esto evita que un atacante capture un código que acaba de usar e inicie sesión con él.

```
Unset
By default, a new token is generated every 30 seconds by the mobile
app.
In order to compensate for possible time-skew between the client and
the server, we allow an extra token before and after the current
time. This allows for a time skew of up to 30 seconds between the
authentication server and client. Suppose you experience problems
with poor time synchronization. In that case, you can increase the
window from its default size of 3 permitted codes (one previous
code, the current code, the next code) to 17 permitted codes (the
```

```
eight previous codes, the current code, and the eight next codes).
This will permit a time skew of up to 4 minutes between client and
server.
Do you want to do so? (y/n) n
```

Responder sí aquí permite hasta 17 códigos válidos en una ventana móvil de cuatro minutos. Al responder que no, lo limita a 3 códigos válidos en un período continuo de 1:30 minutos. A menos que encuentre problemas con la ventana de 1:30 minutos, responder no es la opción más segura. Puede cambiar esta configuración más adelante en el archivo `.google_authenticator` almacenado en la raíz de su directorio de inicio:

```
Unset
If the computer that you are logging into isn't hardened against
brute-force login attempts, you can enable rate-limiting for the
authentication module.
By default, this limits attackers to no more than three login
attempts every 30s.
Do you want to enable rate-limiting (y/n) y
```

La limitación de velocidad significa que un atacante remoto solo puede intentar una cierta cantidad de conjeturas antes de verse obligado a esperar un tiempo antes de poder volver a intentarlo. Si no ha configurado previamente la limitación de velocidad directamente en SSH, hacerlo ahora es una excelente técnica de refuerzo.

Nota : Una vez que finalice esta configuración, si desea hacer una copia de seguridad de su clave secreta, puede copiar `~/.google-authenticator` a una ubicación confiable. Desde allí, puede implementarlo en sistemas adicionales o volver a implementarlo después de una copia de seguridad.

Ahora que el PAM de Google está instalado y configurado, el siguiente paso es configurar SSH para usar su clave TOTP. Necesitaremos informar a SSH sobre el PAM y luego configurar SSH para usarlo.

Paso 2: Configurar OpenSSH para usar MFA/2FA

Debido a que realizaremos cambios de SSH a través de SSH, es importante que nunca cierres tu conexión SSH inicial. En su lugar, abra una segunda sesión SSH para realizar pruebas. Esto es para evitar quedarse fuera de su servidor si hubo un error en su configuración SSH. Una vez que todo funcione, podrá cerrar cualquier sesión de forma segura. Otra precaución de seguridad es crear una copia de seguridad de los archivos del

sistema que edita, de modo que si algo sale mal, pueda volver al archivo original y comenzar de nuevo con una configuración limpia.

Para comenzar, haga una copia de seguridad del archivo de configuración:

```
Unset
sudo cp /etc/pam.d/sshd /etc/pam.d/sshd.back
```

Ahora abra el archivo usando editor de texto favorito:

```
Unset
sudo nano /etc/pam.d/sshd
```

Agregue la siguiente línea al final del archivo:

/etc/pam.d/sshd

```
Unset
. . .
# Standard Un*x password updating.
@include common-password
auth required pam_google_authenticator.so nullok
auth required pam_permit.so
```

La palabra `nullok` al final de la última línea le dice al PAM que este método de autenticación es opcional. Esto permite a los usuarios sin un token OATH-TOTP iniciar sesión simplemente usando su clave SSH. Una vez que todos los usuarios tengan un token OATH-TOTP, puede eliminar `nullok` de esta línea para que MFA sea obligatorio. La segunda línea `pam_permit.so` es necesaria para permitir la autenticación de usuarios que no utilizan un token MFA para iniciar sesión. Al iniciar sesión, cada método necesita un SUCCESS para permitir la autenticación. Si un usuario no utiliza la herramienta de autenticación MFA, utilizara la opción `nullok` devuelve un IGNOR para pasar a la autenticación del teclado interactivo. `pam_permit.so` luego devuelve SUCCESS y permite que continúe la autenticación.

Guarde y cierre el archivo.

A continuación, configuraremos SSH para admitir este tipo de autenticación.

Primero haga una copia de seguridad del archivo:

```
Unset
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.back
```

Ahora abra el archivo de configuración SSH para editarlo:

```
Unset
sudo nano /etc/ssh/sshd_config
```

Busque `ChallengeResponseAuthentication` y establezca su valor en `yes`:

/etc/ssh/sshd_config

```
Unset
. . .
# Change to yes to enable challenge-response passwords (beware
issues with
# some PAM modules and threads)
ChallengeResponseAuthentication yes
. . .
```

Guarde y cierre el archivo, luego reinicie SSH para recargar los archivos de configuración. Reiniciar el servicio no cerrará nuestras conexiones abiertas actuales, lo que significa que no correrá el riesgo de bloquearse con este comando:

```
Unset
sudo systemctl restart sshd.service
```

Para comprobar que todo funciona hasta ahora, abra OTRA terminal e intente iniciar sesión a través de SSH. Es muy importante que mantenga abierta su sesión SSH actual y la pruebe con una sesión adicional, o se bloqueará en algún momento y necesitará usar la consola web para volver a ingresar.

Nota: Si creó previamente una clave SSH y la está usando, notará que no tuvo que escribir su contraseña de usuario ni el código de verificación MFA. Esto se debe a que una clave SSH anula todas las demás opciones de autenticación de forma predeterminada. De lo contrario, debería haber recibido un mensaje de contraseña y código de verificación.

A continuación, para habilitar una clave SSH como un factor y el código de verificación como segundo, debemos indicarle a SSH qué factores usar y evitar que la clave SSH anule todos los demás tipos.

Paso 3: Hacer que SSH sea consciente de MFA

MFA todavía no funciona si está utilizando una clave SSH. Para que SSH reconozca MFA, vuelva a abrir el `sshd` archivo de configuración:

```
Unset
sudo nano /etc/ssh/sshd_config
```

Agregue la siguiente línea al final del archivo. Esto le dice a SSH qué métodos de autenticación son necesarios. Le decimos a SSH que los usuarios necesitan una clave SSH y una contraseña o un código de verificación (o los tres):

/etc/ssh/sshd_config

```
Unset
. . .
AuthenticationMethods publickey,keyboard-interactive
```

Guarde y cierre el archivo.

A continuación, abra nuevamente el archivo de configuración PAM :

```
Unset
sudo nano /etc/pam.d/sshd
```

Busque la línea `@include common-auth` y coméntela agregando un `#`carácter como primer carácter de la línea. Esto le indica a PAM que no solicite una contraseña:

/etc/pam.d/sshd

```
Unset
. . .
# Standard Un*x authentication.
#@include common-auth
. . .
```

Guarde y cierre el archivo, luego reinicie SSH:

```
Unset
sudo systemctl restart sshd.service
```

Ahora intente iniciar sesión en el servidor nuevamente con una sesión/ventana de terminal diferente. A diferencia de la última vez, SSH debería solicitar su código de verificación. Ingrese y completará el inicio de sesión. Aunque no hay indicios de que se haya utilizado su clave SSH, su intento de inicio de sesión utilizó dos factores. Si desea verificar esto, puede agregar `-v` (para obtener más detalles) después del comando SSH.

El `-v` interruptor producirá una salida como esta:

```
Unset
ssh osboxes@192.168.4.71 -v


     Example SSH output\

. . .
debug1: Authentications that can continue: publickey
debug1: Next authentication method: publickey
debug1: Offering RSA public key: /Users/sammy/.ssh/id_rsa
debug1: Server accepts key: pkalg rsa-sha2-512 blen 279
Authenticated with partial success.
debug1: Authentications that can continue:
password,keyboard-interactive
debug1: Next authentication method: keyboard-interactive
Verification code:
```

Hacia el final del resultado, verá dónde SSH usa su clave SSH y luego solicita el código de verificación. Ahora se puede iniciar sesión a través de SSH con una clave SSH y una contraseña de un solo uso. Si desea aplicar los tres tipos de autenticación, puede seguir el siguiente paso.

Felicitaciones, ha agregado con éxito un segundo factor al iniciar sesión de forma remota en su servidor a través de SSH. Si esto es lo que desea (usar su clave SSH y un token TOTP para habilitar MFA para SSH (para la mayoría de las personas, esta es la configuración óptima), entonces ya está.

Paso 4: Agregar un tercer factor (opcional)

En el Paso 3, enumeramos los tipos de autenticación aprobados en el `sshd_config` archivo:

1. `publickey`(Clave SSH)
2. `password publickey`(contraseña)
3. `keyboard-interactive`(código de verificación)

Aunque enumeramos tres factores diferentes, las opciones que hemos elegido hasta ahora solo permiten una clave SSH y el código de verificación. Si desea tener los tres factores (clave SSH, contraseña y código de verificación), un cambio rápido los habilitará.

Abra el archivo de configuración PAM de `sshd` :

```
Unset
sudo nano /etc/pam.d/sshd
```

Localice la línea que comentó anteriormente `#@include common-auth` y descomente la línea eliminando el `#`carácter. Guarde y cierre el archivo. Ahora, una vez más, reinicie SSH:

```
Unset
sudo systemctl restart sshd.service
```

Al habilitar la opción `@include common-auth`, PAM ahora solicitará una contraseña además de verificar una clave SSH y solicitar un código de verificación, que ya habíamos trabajado anteriormente. Ahora podemos usar algo que sabemos (contraseña) y dos tipos diferentes de cosas que tenemos (clave SSH y código de verificación) en dos canales diferentes (su computadora para la clave SSH y su teléfono para el token TOTP).

Paso 5: Recuperar el acceso a Google MFA (opcional)

Al igual que con cualquier sistema que refuerce y proteja, usted se vuelve responsable de administrar esa seguridad. En este caso, eso significa no perder su clave SSH o su clave secreta TOTP y asegurarse de tener acceso a su aplicación TOTP. Sin embargo, a veces suceden cosas y puedes perder el control de las claves o aplicaciones que necesitas para ingresar.

Perder su clave secreta TOTP

Si pierde su clave secreta TOTP, puede dividir el proceso de recuperación en un par de pasos. El primero es volver a ingresar sin conocer el código de verificación y el segundo es encontrar la clave secreta o regenerar para el inicio de sesión normal de MFA. Esto puede suceder a menudo si obtienes un teléfono nuevo y no transfieres tus secretos a una nueva aplicación de autenticación.

Para ingresar después de perder la clave secreta TOTP en un DigitalOcean Droplet, puede usar la consola virtual desde su tablero para iniciar sesión con su nombre de usuario y contraseña. Esto funciona porque solo protegemos su cuenta de usuario con MFA para conexiones SSH. Las conexiones que no son ssh, como el inicio de sesión en la consola, no utilizan el módulo PAM de Google Authenticator.

Si está en un sistema que no es Droplet, tiene dos opciones para recuperar el acceso:

1. Acceso de consola (local/no ssh) al sistema (normalmente físico o mediante algo como iDrac)
2. Tener un usuario diferente que no tenga MFA habilitado

La segunda opción es la menos segura, ya que el objetivo de usar MFA es reforzar todas las conexiones SSH, pero es a prueba de fallos si pierde el acceso a su aplicación de autenticación MFA.

Una vez que haya iniciado sesión, hay dos formas de ayudar a obtener el secreto TOTP:

1. Recuperar la clave existente
2. Generar una nueva clave

En el directorio de inicio de cada usuario, la clave secreta y la configuración de Google Authenticator se guardan en un `~/.google-authenticator`archivo. La primera línea de este archivo es una clave secreta. Una forma rápida de obtener la clave es ejecutar el siguiente comando, que muestra la primera línea del `google-authenticator`archivo (es decir, la clave secreta). Luego, toma esa clave secreta y escríbela manualmente en una aplicación TOTP:

Unset

```
head -n 1 /home/sammy/.google_authenticator
```

Una vez que haya recuperado su clave existente, puede escribirla manualmente en su aplicación de autenticación o completar los detalles relevantes en la URL a continuación y hacer que Google genere un código QR para que lo escanee. Deberá agregar su nombre de usuario, nombre de host, la clave secreta del `.google-authenticator` archivo y luego cualquier nombre que elija para 'entry-name-in-auth-app' para identificar fácilmente esta clave frente a un token TOTP diferente:

Unset

```
https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/username@hostname%3Fsecret%3D16-char-secret%26issuer%3Dentry-name-in-auth-app
```

Si hay un motivo para no utilizar la clave existente (por ejemplo, no poder compartir fácilmente la clave secreta con el usuario afectado de forma segura), puede eliminar el

`~/.google-authenticator` archivo directamente. Esto permitirá que el usuario inicie sesión nuevamente usando un solo factor, suponiendo que no haya aplicado MFA al eliminar la `opción nullok`. Luego pueden ejecutarse `google-authenticator` para generar una nueva clave.

Perder acceso a la aplicación TOTP

Si necesita iniciar sesión en su servidor pero no tiene acceso a su aplicación TOTP para obtener su código de verificación, aún puede iniciar sesión usando los códigos de recuperación que se mostraron cuando creó su clave secreta por primera vez, y son los últimos cinco. líneas del `.google-authenticator` archivo. Tenga en cuenta que estos códigos de recuperación son de un solo uso. Sin embargo, para que esto funcione, debes tener tus códigos de recuperación disponibles cuando no tengas acceso a la aplicación TOTP.

Paso 6: Cambiar la configuración de autenticación (opcional)

Si desea cambiar su configuración de MFA después de la configuración inicial, en lugar de generar una nueva configuración con la configuración actualizada, puede simplemente editar el `~/.google-authenticator` archivo. Las opciones de este archivo aparecen de la siguiente manera:

Diseño del autenticador de google

```
Unset
<secret key>
<options>
<recovery codes>
```

Las opciones configuradas en este archivo tienen una línea en la sección de opciones; Si respondió "no" a una opción particular durante la configuración inicial, el programa excluye la correspondiente.

Aquí hay algunos cambios que puede realizar en este archivo:

- Para habilitar códigos secuenciales en lugar de códigos basados en tiempo, cambie la línea `" TOTP_AUTHa " HOTP_COUNTER 1`.
- Para permitir múltiples usos de un solo código, elimine la línea `" DISALLOW_REUSE`.
- Para extender la ventana de vencimiento del código a 4 minutos, agregue la línea `" WINDOW_SIZE 17`.
- Para deshabilitar múltiples inicios de sesión fallidos (limitación de velocidad), elimine la línea `" RATE_LIMIT 3 30`.

- Para cambiar el umbral de limitación de velocidad, busque la línea `" RATE_LIMIT 3 30`y ajuste los números. En `3`el original indica el número de intentos durante un período y `30`el tiempo en segundos.
- Para desactivar el uso de códigos de recuperación, elimine los cinco códigos de ocho dígitos que se encuentran en la parte inferior del archivo.

Paso 7: Evitar MFA para algunas cuentas (opcional)

Puede haber una situación en la que un solo usuario o algunas cuentas de servicio (es decir, cuentas utilizadas por aplicaciones, no por humanos) necesitan acceso SSH sin MFA habilitado. Por ejemplo, es posible que algunas aplicaciones que utilizan SSH, como algunos clientes FTP, no admiten MFA. Si una aplicación no tiene una forma de solicitar el código de verificación, la solicitud puede bloquearse hasta que se agote el tiempo de espera de la conexión SSH.

Para controlar qué factores usar para un usuario, puede editar el archivo /etc/pam.d/sshd.

Para permitir MFA para algunas cuentas y SSH solo para otras, asegúrese de que las siguientes configuraciones `/etc/pam.d/sshd` estén activas:

/etc/pam.d/sshd

```
Unset
# PAM configuration for the Secure Shell service

# Standard Un*x authentication.
#@include common-auth

. . .

# Standard Un*x password updating.
@include common-password
auth required pam_google_authenticator.so nullok
```

Aquí `@include common-auth` está comentado porque las contraseñas deben estar deshabilitadas. No puede forzar MFA si algunas cuentas tienen MFA deshabilitado, así que deje la `nullok` opción en la última línea.

Después de establecer esta configuración, ejecútela `google-authenticator` como cualquier usuario que necesite MFA y no la ejecute para usuarios que solo usarán claves SSH.

Paso 8: Automatización de la configuración con Gestión de configuración (opcional)

Muchos administradores de sistemas utilizan herramientas de gestión de configuración , como Puppet, Chef o Ansible, para gestionar sus sistemas. Puede utilizar un sistema como este para instalar y configurar una clave secreta cada vez que un nuevo usuario crea una cuenta.

`google-authenticator` admite modificadores de línea de comandos para configurar todas las opciones en un solo comando no interactivo. Para ver todas las opciones, puedes escribir `google-authenticator --help`. A continuación se muestra el comando que configuraría todo como se describe en el Paso 1: