Cómo configurar Squid Proxy para conexiones privadas en Ubuntu 20.04

Introducción

Los servidores proxy son un tipo de aplicación de servidor que funciona como puerta de enlace entre un usuario final y un recurso de Internet. A través de un servidor proxy, un usuario final puede controlar y monitorear su tráfico web para una amplia variedad de propósitos, que incluyen privacidad, seguridad y almacenamiento en caché. Por ejemplo, puede usar un servidor proxy para realizar solicitudes web desde una dirección IP diferente a la suya. También puede usar un servidor proxy para investigar cómo se sirve la web de manera diferente de una jurisdicción a otra, o evitar algunos métodos de vigilancia o limitación del tráfico web.

Squid es un proxy HTTP estable, popular y de código abierto. En este tutorial, instalará y configurará Squid para proporcionar un proxy HTTP en un servidor Ubuntu 20.04.

requisitos previos

Para completar esta guía, necesitará:

- Un servidor Ubuntu 20.04 y un usuario no root con privilegios sudo. Puede obtener más información sobre cómo configurar un usuario con estos privilegios en nuestra guía Configuración inicial del servidor con Ubuntu 20.04 .

Utilizará el nombre de dominio **your_domain** en este tutorial, pero debe sustituirlo por su propio nombre de dominio o dirección IP.

Paso 1: Instalar Squid Proxy

Squid tiene muchos casos de uso más allá del enrutamiento del tráfico saliente de un usuario individual. En el contexto de las implementaciones de servidores a gran escala, se puede utilizar como un mecanismo de almacenamiento en caché distribuido, un equilibrador de carga u otro componente de una pila de enrutamiento. Sin embargo, algunos métodos de escalado horizontal del tráfico del servidor que normalmente habrían implicado un servidor proxy han sido superados en popularidad por los marcos de contenedores como Kubernetes, que distribuyen más componentes de una aplicación. Al mismo tiempo, el uso de servidores proxy para redirigir las solicitudes web como usuario individual se ha vuelto cada vez más popular para proteger su privacidad. Es útil tener esto en cuenta cuando se trabaja con servidores proxy de código abierto que pueden parecer tener muchas docenas de funciones en un modo de mantenimiento de menor prioridad. Los casos de uso de un proxy han cambiado con el tiempo,

Comience ejecutando los siguientes comandos como usuario no root para actualizar sus listados de paquetes e instalar Squid Proxy:

```
sudo apt update

sudo apt install squid

```

Squid configurará automáticamente un servicio en segundo plano y se iniciará después de la instalación. Puede comprobar que el servicio se está ejecutando correctamente:

```
systemctl status squid.service

```

```
Output

● squid.service - Squid Web Proxy Server

     Loaded: loaded (/lib/systemd/system/squid.service; enabled;
vendor preset: enabled)

     Active: active (running) since Wed 2021-12-15 21:45:15 UTC; 2min
11s ago
```

De forma predeterminada, Squid no permite que ningún cliente se conecte desde fuera de este servidor. Para habilitar eso, deberá realizar algunos cambios en su archivo de configuración, que se almacena en `/etc/squid/squid.conf`. Ábrelo en `nano` tu editor de texto favorito:

```
sudo nano /etc/squid/squid.conf

```

Tenga en cuenta que el archivo de configuración predeterminado de Squid es muy, muy largo y contiene una gran cantidad de opciones que se han deshabilitado temporalmente al colocar un `#`al comienzo de la línea en la que se encuentran, también llamado *comentario* . Lo más probable es que desee buscar en el archivo para encontrar las líneas que desea editar. En `nano`, esto se hace presionando `Ctrl+W`, ingresando su término de búsqueda, presionando `Enter` y luego presionando repetidamente `Alt+W` para encontrar la siguiente instancia de ese término si es necesario.

Comience navegando hasta la línea que contiene la frase `http_access deny all`. Debería ver un bloque de texto que explica las reglas de acceso predeterminadas de Squid:

/etc/calamar/calamar.conf

```
. . .

#

# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

#

include /etc/squid/conf.d/*

# Example rule allowing access from your local networks.

# Adapt localnet in the ACL section to list your (internal) IP
networks

# from where browsing should be allowed

#http_access allow localnet

http_access allow localhost



# And finally deny all other access to this proxy

http_access deny all

. . .
```

A partir de esto, puede ver el comportamiento actual: `localhost` está permitido; otras conexiones no lo son. Tenga en cuenta que estas reglas se analizan secuencialmente, por lo que es una buena idea mantener la `deny all` regla en la parte inferior de este bloque de configuración. Podría cambiar esa regla a `allow all`, permitiendo que cualquier persona se conecte a su servidor proxy, pero probablemente no quiera hacer eso. En su lugar, puede agregar una línea arriba `http_access allow localhost` que incluya su propia dirección IP, así:

/etc/calamar/calamar.conf

```
#

# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

#
```

```
include /etc/squid/conf.d/*

# Example rule allowing access from your local networks.

acl localnet src your_ip_address

# Adapt localnet in the ACL section to list your (internal) IP
networks

# from where browsing should be allowed

#http_access allow localnet

http_access allow localhost
```

- **acl** significa una **lista de control** de acceso, un **término** común para las políticas de permisos
- **localnet** en este caso es el nombre de su ACL.
- **src** es desde donde se originaría la solicitud bajo esta ACL, es decir, su dirección IP.

Si no conoce su dirección IP local, lo más rápido es ir a un sitio como Cuál es mi IP , que puede decirle desde dónde accedió. Después de hacer ese cambio, guarde y cierre el archivo. Si está usando `nano`, presione `Ctrl+X`, y luego cuando se le solicite, `Y` y luego `Enter`.

En este punto, puede reiniciar Squid y conectarse a él, pero hay más que puede hacer para asegurarlo primero.

Paso 2: asegurar el calamar

La mayoría de los servidores proxy y la mayoría de las aplicaciones del lado del cliente que se conectan a servidores proxy (por ejemplo, navegadores web) admiten varios métodos de autenticación. Estos pueden incluir claves compartidas o servidores de autenticación separados, pero más comúnmente implican pares regulares de nombre de usuario y contraseña. Squid le permite crear pares de nombre de usuario y contraseña utilizando la funcionalidad integrada de Linux, como un paso adicional o alternativo para restringir el acceso a su proxy por dirección IP. Para hacer eso, creará un archivo llamado `/etc/squid/passwords` y señalará la configuración de Squid.

Primero, deberá instalar algunas utilidades del proyecto Apache para tener acceso a un generador de contraseñas que le gusta a Squid.

```
1. sudo apt install apache2-utils
```

```
2.
```

Este paquete proporciona el `htpasswd` comando, que puede usar para generar una contraseña para un nuevo usuario de Squid. Los nombres de usuario de Squid no se superpondrán con los nombres de usuario del sistema de ninguna manera, por lo que puede usar el mismo nombre con el que inició sesión si lo desea. También se le pedirá que agregue una contraseña:

```
1. sudo htpasswd -c /etc/squid/passwords your_squid_username
2.
```

Esto almacenará su nombre de usuario junto con un hash de su nueva contraseña en `/etc/squid/passwords`, que Squid utilizará como fuente de autenticación. Puede `cat` ver el archivo después para ver cómo se ve:

```
1. sudo cat /etc/squid/passwords
2.
```

```
Output

sammy:$apr1$Dgl.Mtnd$vdqLYjBGdtoWA47w4q1Td.
```

Después de verificar que su nombre de usuario y contraseña hayan sido almacenados, puede actualizar la configuración de Squid para usar su nuevo `/etc/squid/passwords` archivo. Usando `nano` o su editor de texto favorito, vuelva a abrir el archivo de configuración de Squid y agregue las siguientes líneas resaltadas:

```
1. sudo nano /etc/squid/squid.conf
2.
```

/etc/calamar/calamar.conf

```
…

#

# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

#

include /etc/squid/conf.d/*
```

```
auth_param basic program /usr/lib/squid3/basic_ncsa_auth
/etc/squid/passwords

auth_param basic realm proxy

acl authenticated proxy_auth REQUIRED

# Example rule allowing access from your local networks.

acl localnet src your_ip_address

# Adapt localnet in the ACL section to list your (internal) IP
networks

# from where browsing should be allowed

#http_access allow localnet

http_access allow localhost

http_access allow authenticated

# And finally deny all other access to this proxy

http_access deny all

…
```

Estas directivas adicionales le dicen a Squid que verifique en su nuevo `passwords` archivo los hashes de contraseña que se pueden analizar usando el `basic_ncsa_auth` mecanismo, y que requiera autenticación para acceder a su proxy. Puede revisar la documentación de Squid para obtener más información sobre este u otros métodos de autenticación. Después de eso, finalmente puede reiniciar Squid con sus cambios de configuración. Esto puede tardar un momento en completarse.

```
1. sudo systemctl restart squid.service
2.
```

Y no olvide abrir el puerto 3128 en su firewall si está usando `ufw`:

```
1. sudo ufw allow 3128
2.
```

En el siguiente paso, por fin te conectarás a tu proxy.

Paso 3: Conexión a través de Squid

Para demostrar su servidor Squid, utilizará un programa de línea de comandos llamado `curl`, que es popular para realizar diferentes tipos de solicitudes web. En general, si desea verificar si una conexión determinada debería funcionar en un navegador en circunstancias ideales, siempre debe probar primero con `curl`. Usará curl en su máquina **local** para hacer esto; está instalado de forma predeterminada en todos los entornos modernos de Windows, Mac y Linux, por lo que puede abrir cualquier shell local para ejecutar este comando:

```
1. curl -v -x
   http://your_squid_username:your_squid_password@your_server_ip:3128
   http://www.google.com/
2.
```

El `-x` argumento pasa un servidor proxy a curl y, en este caso, está utilizando el `http://`protocolo, especificando su nombre de usuario y contraseña para este servidor y luego conectándose a un sitio web conocido como `google.com`. Si el comando fue exitoso, debería ver el siguiente resultado:

```
Output

*   Trying 138.197.103.77...

* TCP_NODELAY set

* Connected to 138.197.103.77 (138.197.103.77) port 3128 (#0)

* Proxy auth using Basic with user 'sammy'

> GET http://www.google.com/ HTTP/1.1
```

También es posible acceder a `https://`sitios web con su proxy Squid sin realizar más cambios de configuración. Estos hacen uso de una directiva de proxy separada llamada `CONNECT` para preservar SSL entre el cliente y el servidor:

```
1. curl -v -x
   http://your_squid_username:your_squid_password@your_server_ip:3128
   https://www.google.com/
```

```
2.
```

```
Output

*   Trying 138.197.103.77...

* TCP_NODELAY set

* Connected to 138.197.103.77 (138.197.103.77) port 3128 (#0)

* allocate connect buffer!

* Establish HTTP proxy tunnel to www.google.com:443

* Proxy auth using Basic with user 'sammy'

> CONNECT www.google.com:443 HTTP/1.1

> Host: www.google.com:443

> Proxy-Authorization: Basic c2FtbXk6c2FtbXk=

> User-Agent: curl/7.55.1

> Proxy-Connection: Keep-Alive

>

< HTTP/1.1 200 Connection established

<

* Proxy replied OK to CONNECT request

* CONNECT phase completed!
```

Las credenciales que usó `curl` ahora deberían funcionar en cualquier otro lugar donde desee usar su nuevo servidor proxy.

Conclusión

En este tutorial, aprendió a implementar un popular punto final de API de código abierto para el tráfico de proxy con poca o ninguna sobrecarga. Muchas aplicaciones tienen soporte de proxy incorporado (a menudo a nivel del sistema

operativo) desde hace décadas, lo que hace que esta pila de proxy sea altamente reutilizable.

A continuación, es posible que desee aprender a implementar Dante , un proxy SOCKS que puede ejecutarse junto con Squid para transmitir diferentes tipos de tráfico web.

Debido a que uno de los casos de uso más comunes para los servidores proxy es el tráfico de proxy hacia y desde diferentes regiones globales, es posible que desee revisar cómo usar Ansible para automatizar las implementaciones de servidores a continuación, en caso de que desee duplicar esta configuración en otros centros de datos. .