



## 6

# IMPROVING BROWSING AND PRIVACY WITH THE SQUID PROXY



A proxy server acts as an intermediary between you and the internet. When you request a web page, the proxy receives the request and then forwards it on to the web server (if necessary). A proxy allows you to protect your privacy by obfuscating the metadata that is usually available to the services we interact with daily on the internet. Proxies also allow the administrator to block access to certain content, like social media or online gambling.

This chapter will show you how to install, configure, and use the *Squid proxy*, a solution that runs on most operating systems. With Squid, you'll be able to speed up access to websites, enhance your security, and allow or prevent access to specific domains or websites. **Chapter 7** covers another proxy solution, Pi-Hole, which offers the same benefits as Squid, but additionally blocks ads and prevents other tracking and privacy issues as well. Choosing the best proxy for your needs will depend on which you find easier to use and which provides you the best user experience.

## Why Use a Proxy?

Every time you visit a website, your computer makes a request to a web server that responds by sending you the information necessary to view the website. The communication between your browser and the server may expose your personal information (the browser you're using, your public IP address, and so on) through *metadata*. The metadata allows the web server to make guesses about you and your device, such as your location, what time of day it is where you are, and your browsing habits. For lots of reasons, you might want to keep this information private. Additionally, loading web pages and their content consumes bandwidth, so as more people use an internet connection, the connection can begin to slow down, negatively affecting everyone using it.

One great thing about proxies is that they *cache* any traffic that passes through them. This means that every time a web page is retrieved, the proxy will keep a local copy of that page. The next time someone tries to browse to that site, the proxy first checks its cache for a copy, and if it holds a copy, it presents that copy to the user rather than sending a request to the web server for a fresh copy of the web page. By default, Squid will keep a cached copy of a website for a set period before it no longer considers the cache "fresh" and will then retrieve the latest version of the page, whether or not the content has changed. This reduces the load on the network, the time it takes to load frequently visited sites, and the overall amount of bandwidth used, leading to a better experience for everyone involved.

Proxies also reduce the amount of *personally identifiable information* (PII) leaked to web servers. PII is any data or information that can be used to identify any specific individual (such as you). For example, a proxy can identify itself to a web server as any web browser. You might be using Google Chrome, but the proxy could present Firefox to the server instead. The proxy can also have a different public IP address to hide the one you're using if it's located somewhere other than

where you are (like in the cloud), obscuring your physical location and internet service provider.

Even though it isn't directly relevant to small network administrators, you might be interested to know that commercial organizations often rely on proxies (including Squid) for the benefits we've already discussed, as well as for content delivery, such as streaming audio and video. Content providers, such as Netflix and YouTube, strategically place proxy servers globally to keep local copies of content. This practice allows users of those services to access the content from a source closer to home, rather than all users accessing the content from a single location, which would be far less efficient and would result in poor performance in a lot of cases.

## **#21: Setting Up Squid**

The Squid web proxy provides all the benefits you just learned about: it reduces bandwidth, making surfing the web faster for users. It's also capable of anonymizing your personal information if configured correctly; information about your identity, such as where your web requests are coming from or the web browser you're using, can be stripped or changed before traffic is sent to the internet. Many enterprise-grade devices use Squid. While you could use many other proxy solutions, such as NGINX, Apache Traffic Server, or Forcepoint, Squid is free and open source, so it provides greater access to underlying configurations and data than a commercial solution might.

A wealth of information is available online about using Squid to protect and enhance your network. You can find more information on Squid proxy configuration in the Squid wiki at <https://wiki.squid-cache.org/SquidFaq/>.

This project will cover the initial installation and configuration of Squid, configuring clients in your network to use the proxy, testing the

proxy once configured, and performing some additional steps to allow or deny access to certain internet resources using the proxy.

## Configuring Squid

Create a base Ubuntu server following the steps in [Chapter 1](#). If you want to hide your location or prefer not to give away your internet service provider (in addition to preventing your metadata being recorded), create the proxy server in the cloud in a country different from your own. Otherwise, locate the proxy server inside your network. Don't forget to add your new server to your network map and asset list you created in previous chapters. Once you've done so, log in to the server via SSH as a standard, non-root user. To install the proxy, use the following command:

---

```
$ sudo apt install squid
```

---

The installation should complete in less than a minute. By default, you'll find the configuration file located at `/etc/squid/squid.conf`, the logfiles at `/var/log/squid/`, and the cache data (that is, cached website information) at `/var/spool/squid/`.

Open the `squid.conf` configuration file with a text editor to familiarize yourself with the settings:

---

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

---

Squid has many possible configurations, so it's easy to become overwhelmed. Notice, though, that many settings aren't active as they're commented out by default. Let's start by focusing on the active settings. You can explore other changes when your proxy server is functioning as you want it to.

Search by pressing CTRL-W; then type your search term and press ENTER to find the section marked `Recommended minimum configuration`:

```
--snip--
```

```
# Recommended minimum configuration:
```

```
#
```

```
# Example rule allowing access from your local networks.
```

```
# Adapt to list your (internal) IP networks from where browsing
```

```
# should be allowed
```

```
acl localnet src 0.0.0.1-0.255.255.255 # RFC 1122 "this" network  
(LAN)
```

```
acl localnet src 10.0.0.0/8          # RFC 1918 local private network  
(LAN)
```

```
--snip--
```

---

This section details the *access controls lists (ACLs)* that tell Squid which endpoints should have permission to access internet resources via the proxy server. An ACL is a list of ports, addresses, or resources that you've specifically allowed or banned from communication within the network.

An ACL consists of several elements. First is a unique name, such as `localnet`, that identifies a specific ACL. Each named ACL then contains an ACL type (such as `src`) followed by a value or list of values, such as IP addresses or port numbers. These values can be entered over multiple lines, and Squid will combine them into a single list.

Keywords like `src` indicate to Squid the direction in which the traffic is flowing; `src 10.0.0.0/8`, for example, indicates any traffic coming from an address in the `10.0.0.0/8` IP address range to any IP address in any range.

Comment out any lines that don't apply to your network. For example, if your internal IP addresses follow the *10.x.x.x* format, leave the relevant directive as is and comment out all other lines beginning with `acl localnet src` by adding a `#` at the start of each line:

---

```
--snip--
```

```
#acl localnet src 0.0.0.1-0.255.255.255 # RFC 1122 "this" network (LAN)
```

```
acl localnet src 10.0.0.0/8          # RFC 1918 local private network  
(LAN)
```

```
# acl localnet src 100.64.0.0/10     # RFC 6598 shared address space  
(CGN)
```

```
# acl localnet src 169.254.0.0/16    # RFC 3927 link-local machines
```

```
# acl localnet src 172.16.0.0/12     # RFC 1918 local private network  
(LAN)
```

```
# acl localnet src 192.168.0.0/16    # RFC 1918 local private network  
(LAN)
```

```
# acl localnet src fc00::/7          # RFC 4193 local private network  
range
```

```
# acl localnet src fe80::/10         # RFC 4291 link-local machines
```

```
--snip--
```

---

The second portion of the recommended minimum configuration section tells Squid which ports can send and receive traffic:

---

```
--snip--
```

```
acl SSL_ports port 443
```

```
acl Safe_ports port 80      # http

acl Safe_ports port 21      # ftp

acl Safe_ports port 443     # https

# acl Safe_ports port 70     # gopher

# acl Safe_ports port 210    # wais

acl Safe_ports port 1025-65535 # unregistered ports

# acl Safe_ports port 280    # http-mgmt

--snip--
```

---

Here, `SSL_ports` and `Safe_ports` are ACL names, and the `port` type tells Squid to interpret the number that follows as a port number used for communication by a specific service (see [Chapter 1](#)). The `acl SSL_ports port 443` line sets the port your proxy should use for secured, filtered tunnels, such as those used for HTTPS traffic. Directives containing the label `Safe_ports` determine the ports on which Squid should allow connections. If you don't need a certain protocol or port, comment it out to reduce your attack surface. To be prudent, you might keep only ports 80 and 443 and comment out the `acl Safe_ports port 1025-65535` line, thereby blocking ports from 1025 through 65535. However, doing so may cause some applications or services to malfunction if they require other ports. You can use Google and the website or manual for a given application to determine what other ports it might need to function correctly.

A little further in the configuration file, you'll find directives that enable these ACLs:

---

```
--snip--
```

# Recommended minimum Access Permission configuration:

#

# Deny requests to certain unsafe ports

http\_access deny !Safe\_ports

# Deny CONNECT to other than secure SSL ports

http\_access deny CONNECT !SSL\_ports

--snip--

---

The `http_access deny !Safe_ports` directive tells Squid to prohibit communication between all ports except those listed in the `Safe_ports` list. Likewise, the `http_access deny CONNECT !SSL_ports` line tells Squid to prohibit filtered tunnels on any port other than the one specified in `SSL_ports`.

The next section of the configuration file relates to your local network as opposed to the internet:

---

--snip--

# 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
```

```
--snip--
```

---

Remove the `#` from the `http_access allow localnet` directive to enable the `localnet` settings you specified earlier, which allow endpoints on your local network to access the internet through your proxy. Finally, `http_access deny all` ensures the proxy denies all other traffic to keep it from affecting your internal network. By denying all traffic that isn't specifically allowed, you'll protect your network from unwanted traffic, which can include malware.

If you want to change the port on which Squid listens for requests, modify the following line in your configuration file:

---

```
--snip--
```

```
# Squid normally listens to port 3128
```

```
http_port 3128
```

```
--snip--
```

---

Your devices will use this port to connect to the proxy server so they can send requests, receive traffic, and generally browse the internet.

Once you've finished your edits, save and close the configuration file. Reload the updated Squid configuration using the following command so the changes take effect (be aware, though, that reloading the configuration can interrupt any open connections):

---

```
$ sudo systemctl reload squid
```

---

You can now make sure that Squid was able to start successfully and is running with the following command:

---

```
$ sudo systemctl status squid
```

squid.service - Squid Web Proxy Server

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

Active: active (running); 2min 5s ago

--snip--

---

A green dot before `squid.service` and a status of `active (running)` indicates Squid is running as expected. If Squid didn't start properly due to an error, you'll see a failed message with a red dot before `squid.service`:

---

```
$ sudo systemctl status squid
```

squid.service - Squid Web Proxy Server

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

Active: failed (Result: exit-code); 2min 5s ago

--snip--

---

Go back and check your configuration again or validate your configuration file using this command:

---

```
$ squid -k parse
```

2024/05/06 00:44:06 | Processing: acl denylist dstdomain .twitter.com

2024/05/06 00:44:06 | Processing: http\_deny denylist

```
2024/05/06 00:44:06 | /etc/squid/squid.conf:1406 unrecognized:
'http_deny'
```

```
2024/05/06 00:44:06 | Processing: anonymize_headers deny From
Referer Server
```

```
2024/05/06 00:44:06 | /etc/squid/squid.conf:1408 unrecognized:
'anonymize_headers'
```

```
2024/05/06 00:44:06 | Processing: anonymize_headers deny User-Agent
WWW-Authenticate
```

```
2024/05/06 00:44:06 | /etc/squid/squid.conf:1409 unrecognized:
'anonymize_headers'
```

```
2024/05/06 00:44:06 | Processing: http_access allow localnet
```

```
--snip--
```

---

This output shows what you'd see if you used the unrecognized directives `http_deny` and `anonymize_headers`. When you've resolved any errors with the configuration, start Squid with the `start` command:

---

```
$ sudo systemctl start squid
```

---

You've now finished the basic Squid proxy configuration.

## **Configuring Devices to Use Squid**

Next, you'll need to configure the proxy settings on each device that will use the proxy. We'll explain how to configure Windows, macOS, and Linux devices.

### Windows

1. On your Windows host, open the **Windows Settings** dialog.

2. In the Find a Setting box, search for *Proxy Settings*.
3. Turn on the **Use a Proxy Server** toggle in the Proxy window.
4. Enter your proxy server's IP address and port—for example, *192.168.1.50:3128*.
5. Be sure to tick the **Don't Use the Proxy Server for Local (Intranet) Addresses** checkbox.

## macOS

1. Open **System Preferences**.
2. Choose **Network** and select your wireless or Ethernet adapter.
3. Click **Advanced** ▶ **Proxies**.
4. Check the box for **Web Proxy (HTTP)**. Enter your proxy server's IP address and port number—for example, *192.168.1.50:3128*. Do this for each of the protocols listed, which you configured earlier in your */etc/squid/squid.conf* file.
5. Enter your local network into the Bypass Proxy Settings for these Hosts & Domains box.
6. Click **OK** and then **Apply**.

## Linux

1. On your Linux endpoint, open the **Settings** dialog.
2. Go to the **Network** ▶ **Network Proxy** settings.
3. Set the proxy to **Manual** and enter the HTTP Proxy IP address and port number—for example, *192.168.1.50:3128*.
4. Make sure to enter your local network in the Ignore Hosts box, and then close any open settings windows.

## Testing Squid

With both the Squid server and at least one of your devices configured, make sure the device is actually using the proxy and that the proxy functions as expected. On the Squid server, use the following command to view the Squid proxy logfile as it's populated:

```
$ sudo tail -f /var/log/squid/access.log
```

```
--snip--
```

```
1619747519.519  54 172.16.90.1 TCP_TUNNEL/200 39 CONNECT  
play.google.com:443 - HIER_DIRECT/172.217.25.174 -
```

```
1619747519.755  54 172.16.90.1 TCP_TUNNEL/200 39 CONNECT  
mail.google.com:443 - HIER_DIRECT/216.58.200.101 -
```

```
1619747519.776  55 172.16.90.1 TCP_TUNNEL/200 39 CONNECT  
mail.google.com:443 - HIER_DIRECT/216.58.200.101 -
```

```
1619747520.190 161 172.16.90.1 TCP_MISS/200 985 GET
```

```
--snip--
```

---

Your output will differ depending on the applications you're using in your network.

If you don't see any output (and your host is unable to browse the internet), update your iptables or other firewall rules using the steps in [Chapter 3](#) to allow traffic to and from the Squid proxy on port 3128 (or whichever port you configured Squid to listen on).

If you browse to Facebook from a host configured to use your proxy server while the `tail` command is running, you should see this request appear in the log as multiple requests to Facebook services:

---

```
--snip--
```

```
1584414232.470  3 192.168.1.51 NONE/503 0 CONNECT  
pixel.facebook.com:443 - HIER_NONE/- -
```

```
1584414237.647  0 192.168.1.51 NONE/503 0 CONNECT  
pixel.facebook.com:443 - HIER_NONE/- -
```

```
1584414242.652  0 192.168.1.51 NONE/503 0 CONNECT
```

```
pixel.facebook.com:443 - HIER_NONE/- -
```

```
1584414247.864 69023 192.168.1.51 TCP_TUNNEL/200 6426 CONNECT
```

```
static.xx.fbcdn.net:443 - HIER_DIRECT/157.240.8.23 -
```

```
1584414248.566  0 192.168.1.51 NONE/503 0 CONNECT
```

```
pixel.facebook.com:443 - HIER_NONE/- -
```

```
1584414254.535  0 192.168.1.51 NONE/503 0 CONNECT
```

```
pixel.facebook.com:443 - HIER_NONE/- -
```

```
--snip--
```

---

If not, try restarting the proxy server, your host, or both.

## **Blocking and Allowing Domains**

Now that your proxy works, you'll probably want to block (denylist) or allow (allowlist) some domains. For example, if you have children, you may want to prevent them from visiting distracting or inappropriate websites. To do this, open the *squid.conf* file in a text editor:

---

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

---

Now, find the comment that reads `INSERT YOUR OWN RULE(S) HERE`.

In that section, you can define rules (that is, ACLs) of your own. As mentioned, an ACL is made up of an ACL name, an ACL type such as `allow` or `deny`, and a list of elements, such as IP addresses or domains. Your configuration will consist of one or more of these rules, identifying what is and is not allowed through the proxy. (Previously, you enabled rules like `http_access allow localnet` and `http_access deny !Safe_ports` to use the ACLs from the recommended minimum configuration section.)

For example, to denylist Facebook, enter the following lines after the `include` directive:

---

```
--snip--
```

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

```
acl denylist dstdomain .facebook.com
```

```
http_access deny CONNECT denylist
```

```
--snip--
```

---

The `acl` directive at the beginning of the line tells Squid to treat what follows as a list of items to either allow or deny. Next, `denylist` is the unique name of the list; choose any name you'd like, so long as it consists of alphanumeric characters. The `dstdomain` directive indicates that what follows is a list of destination domains. The period at the start of a domain indicates to Squid that it should denylist the entire domain, including subdomains. For example, [www.facebook.com](http://www.facebook.com) is a top-level domain name that might have a subdomain of *campus.facebook.com* or *hertz.facebook.com*. If you omit the leading period, Squid will block only the parent domain ([facebook.com](http://facebook.com)). Finally, the `http_access` directive with `deny` and `CONNECT` parameters tells the proxy to forbid connections to the domains or URLs specified in the ACL.

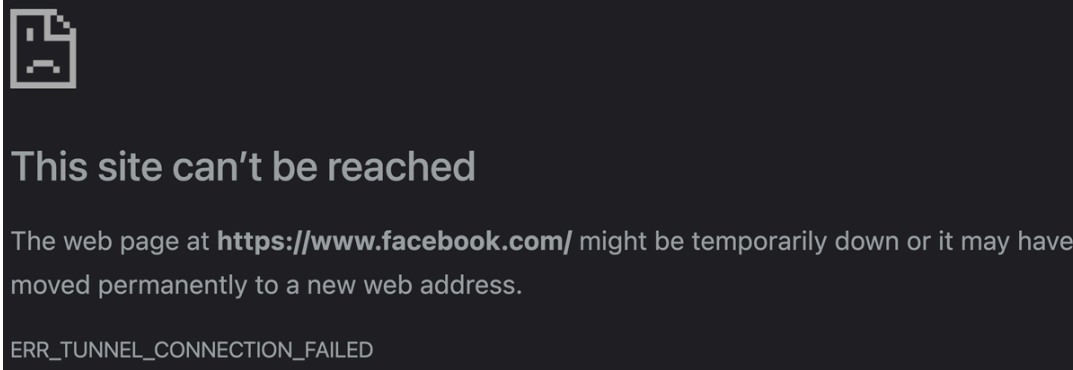
Save the configuration file and reload Squid to make the change take effect:

---

```
$ sudo systemctl reload squid
```

---

Now, try browsing to [www.facebook.com](http://www.facebook.com) from a host configured to use the proxy server. You should see an error page as in [Figure 6-1](#).



*Figure 6-1: Web browser error caused by Squid*

To allow access to Facebook again, either delete or comment out the lines you added, save the configuration file, and reload Squid.

You can repeat the process for additional domains by adding them to the same denylist ACL:

---

```
acl denylist dstdomain .facebook.com .twitter.com .linkedin.com
```

---

Alternatively, you could create separate ACLs for each website or for groups or categories of websites as you desire.

Allowlisting works in pretty much the same way; any domains that are added to the allowlist will be allowed, but only for users who are authenticated to the proxy:

---

```
--snip--
```

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

```
acl allowlist dstdomain .facebook.com .twitter.com .linkedin.com
```

```
http_access allow CONNECT allowlist
```

```
--snip--
```

---



If you add new ACL rules, be aware of where they are located in relation to each other in the configuration file. Squid will interpret the ACL rules in the order they appear, much like a firewall. If there's a `deny all` rule at the beginning of the list of ACL rules, Squid will interpret this rule first and then ignore any further rules in the file. That means you should put any custom rules before the following lines:

---

```
--snip--
```

```
# And finally deny all other access to this proxy
```

```
http_access deny all
```

```
--snip--
```

---

## **Protecting Personal Information with Squid**

Squid is highly configurable and allows you as the administrator to set how much information about your users and their devices you want exposed to the internet. By default, there is no anonymization of the traffic that passes from a client device through the proxy to the internet.

To prevent anyone outside your network knowing where your traffic is coming from (that is, the server information or from which website or resource you may have been referred, like Amazon or a blog), use the `request_header_access` directive to deny this information:

---

```
--snip--
```

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

```
request_header_access From deny all
```

```
request_header_access Referer deny all
```

```
request_header_access Server deny all
```

```
--snip--
```

---

To further anonymize your traffic, it may be wise to also deny the `User-Agent`, `WWW-Authenticate`, and `Link` header values, which may leak additional information about your browser or browsing activity:

---

```
--snip--
```

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

```
request_header_access From deny all
```

```
request_header_access Referer deny all
```

```
request_header_access Server deny all
```

```
request_header_access User-Agent deny all
```

```
request_header_access WWW-Authenticate deny all
```

```
request_header_access Link deny all
```

```
--snip--
```

---

Anonymizing your traffic with these options will limit the amount of PII you're sending over the internet, making you more difficult to track and protecting, to some extent, your browsing history and habits.

**NOTE** *Some websites and services use user agents to determine how to display content to users, so be mindful that by removing the header information, you may experience content differently.*

## Disabling Caching for Specific Sites

There may be some websites that you don't want Squid to cache, as you always want to retrieve the latest version from the web server rather than the cached version from your proxy. This is achieved by denying caching of that site or sites:

---

```
--snip--
```

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

```
acl deny_cache dstdomain .facebook.com
```

```
no_cache deny deny_cache
```

```
--snip--
```

---

Remember to add an ACL entry for each website you want to prevent Squid from creating and keeping a cached copy.

## Squid Proxy Reports

You may have noticed that the Squid logs can be difficult to read and take some getting used to. Third-party solutions are available that make activity reporting and reviewing logs easier. One of the simpler solutions is *Squid Analysis Report Generator (SARG)*. SARG is a web-based report generator and viewer that allows you to view your Squid logs in a browser window, rather than from the terminal.

On your Squid server, install SARG:

---

```
$ sudo apt install sarg
```

---

The SARG report files will be accessed via a web browser, so you also need to install a web server. Install Apache:

---

```
$ sudo apt install apache2
```

---

Next, open the SARG configuration file that should be located at */etc/sarg/sarg.conf*:

---

```
$ sudo nano /etc/sarg/sarg.conf
```

---

Find the line that starts with `access_log`, which specifies the Squid access log location:

---

```
--snip--
```

```
access_log /var/log/squid/access.log
```

```
--snip--
```

---

Then, close the file and use the `find` command to make sure it matches the actual location of the logfile:

---

```
$ sudo find / -name access.log
```

---

```
/var/log/squid/access.log
```

---

Open the file in a text editor and find the output directory tag (the line that starts with `output_dir`), comment out the line containing `/var/lib/sarg`, and replace it with a line that sets the directory to the Apache web location */var/www/html/squid-reports/*:

---

```
--snip--
```

```
# output_dir /var/lib/sarg
```

```
output_dir /var/www/html/squid-reports/
```

```
--snip--
```

---

Save and close the file. Feel free to peruse the other settings if you would like.

To generate a SARG report, run the following command on your Squid server:

---

```
$ sudo sarg -x
```

---

In your web browser, navigate to the reports location on your proxy server: `http://<proxy_ip_address>/squid-reports`. You should see a basic website, as shown in [Figure 6-2](#).



FILE/PERIOD	CREATION DATE	USERS	BYTES	AVERAGE
<a href="#">2021Oct31-2021Oct31</a>	Sun 31 Oct 2021 07:43:27 PM PDT	1	8,68M	8,68M

Generated by [sarg-2.4.0 Jan-16-2020](#) on Oct/31/2021 19:43

*Figure 6-2: SARG reports summary*

Click the relevant report on the page displayed, and you should see information about each user connection through the proxy, how much data was transferred for each connection, how long the connection lasted, and a timestamp indicating when the connection was established, as shown in [Figure 6-3](#).



### Squid User Access Reports

Period: 2021 Oct 31


Sort: bytes, reverse

**Top users**

[Top sites](#)

[Sites & Users](#)

[Denied accesses](#)

NUM	USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT	ELAPSED TIME	MILLISEC	%TIME
1	 172.16.90.1	225	8,68M	100.00%	100.00% 0.00%	08:02:00	28,920,882	100.00%
<b>TOTAL</b>		<b>225</b>	<b>8,68M</b>		<b>100.00% 0.00%</b>	<b>08:02:00</b>	<b>28,920,882</b>	
<b>AVERAGE</b>		<b>225</b>	<b>8,68M</b>			<b>08:02:00</b>	<b>28,920,882</b>	

Generated by [sarg-2.4.0 Jan-16-2020](#) on Oct/31/2021 19:43

*Figure 6-3: SARG report output*

The report shows the users, or hosts, that have used the proxy; the level of traffic they have sent and received (represented as bytes); and various other useful things about the use of the proxy. There are also links included for subreports, such as the top sites accessed via the proxy; the sites and users report, which lists the sites accessed and a list of the users or hosts that accessed each; and any cache or website access that was denied by the proxy based on the rules and configuration you provided.

Try using your new proxy server for a few weeks to see if it helps your bandwidth usage and browsing speed. Once you're comfortable, you could investigate and begin experimenting with the proxy's more advanced features, such as preventing users from downloading large files (this might be advisable if your internet service provider has data caps and charges for bandwidth).

## Summary

Using a proxy server such as Squid offers you a great deal of control over what's allowed in and out of your network. You'll be able to control the PII exposed from your endpoints, such as the web browser

you're using, to improve your network's online privacy. A proxy server also provides a better overall browsing experience.