Security and Privacy Considerations

Overview

This help section provides details about various precautions taken by gsutil to protect data security, as well as recommendations for how customers should safeguard security.

Transport Layer Security

gsutil performs all operations using transport-layer encryption (HTTPS), to protect against data leakage over shared network links. This is also important because gsutil uses "bearer tokens" for authentication (OAuth2) as well as for resumable upload identifiers, and such tokens must be protected from being eavesdropped and reused.

gsutil also supports the older HMAC style of authentication via the XML API (see gsutil endpoints (/storage/docs/request-endpoints#gsutil)). While HMAC authentication does not use bearer tokens (and thus is not subject to eavesdropping/replay attacks), it's still important to encrypt data traffic.

Prior to gsutil release 4.0 it was possible to use HTTP instead of HTTPS by setting the "is_secure" configuration parameter in the [Boto] section of the boto configuration file to False. However, starting with gsutil version 4.0 setting is_secure to False is disallowed. For more details about different credential options, see gsutil help creds (/storage/docs/gsutil/addlhelp/CredentialTypesSupportingVariousUseCases).

To add an extra layer of security, gsutil supports mutual TLS (mTLS) for the Cloud Storage JSON API. With mTLS, the client verifies the server certificate, and the server also verifies the client. To find out more about how to enable mTLS, see the install docs (/storage/docs/gsutil_install).

Local File Storage Security

gsutil takes a number of precautions to protect against security exploits in the files it stores locally:

- When the gsutil config (or gcloud init for Cloud SDK installs) command runs it sets file protection mode 600 ("-rw-------") on the .boto configuration file it generates, so only the user (or superuser) can read it. This is important because these files contain security-sensitive information, including credentials and proxy configuration.
- The gsutil config (or gcloud init for Cloud SDK installs) command also uses file protection mode 600 for the private key file stored locally when you create service account credentials.
- The default level of logging output from gsutil commands does not include security-sensitive information, such as OAuth2 tokens and proxy configuration information. (See the "RECOMMENDED USER PRECAUTIONS" section below if you increase the level of debug output, using the gsutil -D option.)

Note that protection modes are not supported on Windows, so if you use gsutil on Windows we recommend using an encrypted file system and strong account passwords.

Security-Sensitive Files Written Temporarily To Disk By Gsutil

gsutil buffers data in temporary files in several situations:

- While compressing data being uploaded via gsutil cp -z/-Z, gsutil buffers the data in temporary files with protection 600, which it deletes after the upload is complete (similarly for downloading files that were uploaded with gsutil cp -z/-Z or some other process that sets the Content-Encoding to "gzip"). However, if you kill the gsutil process while the upload is under way the partially written file will be left in place. See the "CHANGING TEMP DIRECTORIES" section in gsutil help cp (/storage/docs/gsutil/commands/cp) for details of where the temporary files are written and how to change the temp directory location.
- When performing a resumable upload gsutil stores the upload ID (which, as noted above, is a bearer token and thus should be safe-guarded) in a file under ~/.gsutil/tracker-files with protection 600, and deletes this file after the upload is complete. However, if the upload doesn't complete successfully the tracker file is left in place so the resumable upload can be re-attempted later. Over time it's possible to accumulate these tracker files from aborted upload attempts, though resumable upload IDs are only valid for 1 week, so

the security risk only exists for files less than that age. If you consider the risk of leaving aborted upload IDs in the tracker directory too high you could modify your upload scripts to delete the tracker files; or you could create a cron job to clear the tracker directory periodically.

- The gsutil rsync command stores temporary files (with protection 600) containing the names, sizes, and checksums of source and destination directories/buckets, which it deletes after the rsync is complete. However, if you kill the gsutil process while the rsync is under way the listing files will be left in place.

Note that gsutil deletes temporary files using the standard OS unlink system call, which does not perform data wiping (https://en.wikipedia.org/wiki/Data_erasure). Thus, the content of such temporary files can be recovered by a determined adversary.

Access Control Lists

Unless you specify a different ACL (e.g., via the gsutil cp -a option), by default objects written to a bucket use the default object ACL on that bucket. Unless you modify that ACL (e.g., via the gsutil defacl command), by default it will allow all project editors write access to the object and read/write access to the object's metadata and will allow all project viewers read access to the object.

The Cloud Storage access control system includes the ability to specify that objects are publicly readable. Make sure you intend for any objects you write with this permission to be public. Once "published", data on the Internet can be copied to many places, so it's effectively impossible to regain read control over an object written with this permission.

The Cloud Storage access control system includes the ability to specify that buckets are publicly writable. While configuring a bucket this way can be convenient for various purposes, we recommend against using this permission - it can be abused for distributing illegal content, viruses, and other malware, and the bucket owner is legally and financially responsible for the content stored in their buckets. If you need to make content available to customers who don't have Google accounts consider instead using signed URLs (see gsutil help signurl (/storage/docs/gsutil/commands/signurl)).

Software Integrity And Updates

gsutil is distributed as a standalone bundle via tar and zip files stored in the gs://pub bucket, as a PyPi module, and as part of the bundled Cloud SDK release. Each of these distribution methods takes a variety of security precautions to protect the integrity of the software. We strongly recommend against getting a copy of gsutil from any other sources (such as mirror sites).

Proxy Usage

gsutil supports access via proxies, such as Squid and a number of commercial products. A full description of their capabilities is beyond the scope of this documentation, but proxies can be configured to support many security-related functions, including virus scanning, Data Leakage Prevention, control over which certificates/CA's are trusted, content type filtering, and many more capabilities. Some of these features can slow or block legitimate gsutil behavior. For example, virus scanning depends on decrypting file content, which in turn requires that the proxy terminate the gsutil connection and establish a new connection - and in some cases proxies will rewrite content in ways that result in checksum validation errors and other problems.

For details on configuring proxies see the proxy help text in your .boto configuration file (generated by the gsutil config or gcloud init command).

Encryption At Rest

All Cloud Storage data are automatically stored in an encrypted state, but you can also provide your own encryption keys. For more information, see Cloud Storage Encryption (/storage/docs/encryption).

Data Privacy

Google will never ask you to share your credentials, password, or other security-sensitive information. Beware of potential phishing scams where someone attempts to impersonate Google and asks for such information.

Measurement Data

The gsutil perfdiag command collects a variety of performance-related measurements and details about your local system and network environment, for use in troubleshooting performance problems. None of this information will be sent to Google unless you choose to send it.

Recommended User Precautions

The first and foremost precaution is: Never share your credentials. Each user should have distinct credentials.

If you run gsutil -D (to generate debugging output) it will include OAuth2 refresh and access tokens in the output. Make sure to redact this information before sending this debug output to anyone during troubleshooting/tech support interactions.

If you run gsutil --trace-token (to send a trace directly to Google), sensitive information like OAuth2 tokens and the contents of any files accessed during the trace may be included in the content of the trace.

Customer-supplied encryption key information in the .boto configuration is security sensitive.

The proxy configuration information in the .boto configuration is security-sensitive, especially if your proxy setup requires user and password information. Even if your proxy setup doesn't require user and password, the host and port number for your proxy is often considered security-sensitive. Protect access to your .boto configuration file.

If you are using gsutil from a production environment (e.g., via a cron job running on a host in your data center), use service account credentials rather than individual user account credentials. These credentials were designed for such use and, for example, protect you from losing access when an employee leaves your company.

Last updated 2021-08-19 UTC.