Z

Zip Bomb (Decompression Bomb): How It Works and How to Respond

A zip bomb is a small archive designed to expand into excessive data or nested files. Learn how decompression bombs work, warning signs, safe response, and server defenses.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
5 min

CONTENTS

In this guide

01 How a zip bomb works

02 Recursive and non-recursive zip bombs

03 Zip bomb warning signs

04 What to do if you receive a suspicious archive

05 If extraction has already started

06 How applications should defend against decompression bombs

07 Frequently asked questions

A **zip bomb**, also called a **decompression bomb** or zip of death, is an archive designed to consume excessive disk space, memory, CPU time, or scanning resources when software inspects or extracts it. A file that looks small while compressed can represent gigabytes or more of expanded data.

The goal is usually denial of service: freezing an archive tool, exhausting a server, delaying a security scanner, or increasing cloud-processing cost. The concept is not limited to ZIP; other compressed and container formats can create similar resource-exhaustion risks.

01 How a zip bomb works

Compression stores repeated data efficiently. A malicious archive deliberately maximizes the difference between compressed size and the work or storage required to process its contents.

Technique	What happens during processing
Extreme compression ratio	A very small archive expands into a huge amount of repetitive data.
Nested archives	Archives contain many more archives, multiplying files and work at each level.
Overlapping or specially structured entries	A parser may repeatedly process shared compressed data or misleading metadata.
Very large file count	Millions of small entries exhaust metadata, inode, memory, or scanning limits even if total bytes appear manageable.

A zip bomb does not have to exploit a software vulnerability. Correctly functioning software can still run out of resources if it trusts archive metadata or processes content without limits.

02 Recursive and non-recursive zip bombs

- **Recursive bombs** use layers of archives inside archives. They become dangerous when a tool automatically follows each layer without a depth or total-work limit.

- **Non-recursive bombs** can achieve an extreme expansion ratio in one archive by using the format efficiently or arranging entries to trigger excessive processing.

The well-known archive often called `42.zip` is a historical example of a highly nested zip bomb. Do not download or test suspected samples on a normal workstation or production service.

03 Zip bomb warning signs

- The reported uncompressed size is implausibly larger than the downloaded file.
- The archive contains many nested archives, repeated filenames, or an unusually large number of entries.
- An extraction or scanning process suddenly uses excessive CPU, memory, temporary storage, or disk space.
- The archive tool stops responding while the output directory grows rapidly.
- A security product reports a decompression bomb, archive bomb, oversized archive, or recursion-limit warning.
- A web upload causes timeouts, worker restarts, storage alerts, or unexpected processing cost.

Archive metadata can be deceptive. A safe processor should enforce measured limits while reading and extracting, not rely only on the size declared in the file header.

04 What to do if you receive a suspicious archive

1. **Do not preview or extract it.** File managers, email gateways, backup tools, and antivirus products may inspect archives automatically.
2. **Check the context.** Confirm the sender and whether an archive was expected through a separate trusted channel.
3. **Use an updated security tool.** Scan the file without disabling archive limits or security warnings.

4. **Keep it away from shared services.** Do not upload an unknown archive to a production application, shared drive, or automated pipeline for testing.
5. **Delete or quarantine it** if it has no legitimate business purpose. Preserve it only when an authorized security team needs evidence.

05 If extraction has already started

- Cancel the extraction or scanning job if the application still responds.
- Do not open the partially extracted files.
- Check free disk space, memory pressure, temporary directories, and whether other services were disrupted.
- Remove the extracted output only after confirming the exact directory. Avoid broad deletion commands or wildcards on an unfamiliar system.
- Restart a stuck application or isolated worker if needed; rebooting the whole system should not be the first response on a shared server.
- Run an updated scan because an archive can contain ordinary malware in addition to resource-exhaustion content.

06 How applications should defend against decompression bombs

- Limit compressed input size, total expanded bytes, compression ratio, file count, nesting depth, processing time, memory, and temporary storage.
- Apply limits cumulatively across nested archives, not separately to each child file.
- Stream processing where practical and stop as soon as a limit is crossed.
- Verify the actual path of every output entry and block path traversal, symbolic-link abuse, and writes outside the extraction directory.
- Process untrusted archives in an isolated, low-privilege worker with quotas and no access to sensitive data.

- Reject unsupported archive features and fail closed when metadata is inconsistent.
- Log the reason for rejection without repeatedly reprocessing the same file.

07 Frequently asked questions

Is a zip bomb a virus?

Not necessarily. A zip bomb is a resource-exhaustion technique, while a computer virus replicates by infecting other files or systems. An archive can contain both a decompression bomb and malicious executable content.

Can viewing an archive trigger a zip bomb?

It can. Some file managers, security scanners, indexing services, and upload pipelines automatically inspect archive contents. Safe software places limits on this background processing.

Will deleting the ZIP file restore disk space?

Deleting the original archive does not remove files already extracted to another directory or temporary storage. Stop the process, identify its output, and clean that exact location safely.

NEED MORE HELP?

We are here when the guide is not enough.

Visit the Gridinsoft Support Center for current product guidance or submit a support request with the details of your case.

Open this guide online at support.gridinsoft.com