C

Content Disarm and Reconstruction (CDR): How It Works

Content disarm and reconstruction parses an untrusted file, removes disallowed active content, and creates a new file containing permitted elements.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
4 min

CONTENTS

In this guide

01 How CDR works
02 CDR versus antivirus and sandboxing
03 Where CDR is useful
04 Limitations and tradeoffs
05 Deployment checklist
06 What to do when CDR blocks a file
07 Source

Quick answer: Content disarm and reconstruction (CDR) is a file-security technique that parses an untrusted document, removes or neutralizes content that policy does not allow, and builds a new file from permitted elements. Unlike antivirus, it does not need to identify a specific malware family before modifying the file.

01 How CDR works

A CDR system receives a file from email, a web upload, file transfer, collaboration service, or removable media. It identifies the real file type, parses the internal structure, and extracts supported content. It then removes or changes risky elements according to policy and writes a reconstructed output file. The output may keep the original format or be converted to a simpler format such as PDF.

Depending on product and policy, removed content can include macros, embedded executables, scripts, external links, OLE objects, active PDF features, hidden data, or unsupported nested files. Images and ordinary text may be preserved. “Sanitized” does not mean byte-for-byte identical: reconstruction intentionally changes the file.

02 CDR versus antivirus and sandboxing

- **Antivirus** looks for known malicious code, reputation, or suspicious behavior and may block or quarantine the whole file.
- **Sandboxing** opens or executes a file in an isolated environment to observe behavior. Malware may delay execution or recognize the analysis environment.
- **CDR** treats active or unsupported components as untrusted and rebuilds the file without them, even when no malicious signature exists.

These controls complement each other. Antivirus can block a known threat before processing; a sandbox can analyze uncertain behavior; CDR can provide a usable reduced-risk copy. None guarantees that every permitted element is safe.

03 Where CDR is useful

Common use cases include email attachments, public document-upload portals, cross-domain file transfer, remote-user downloads, and files entering sensitive networks. CDR is most valuable when users need the information in documents but do not need active content from untrusted senders.

Policies should vary by workflow. A finance team may need formulas but not macros; a print workflow may accept a flattened PDF; a software-development workflow cannot sanitize source archives without destroying their purpose.

04 Limitations and tradeoffs

- **Fidelity:** complex formatting, signatures, formulas, comments, accessibility tags, animations, or embedded media may change or disappear.
- **Unsupported formats:** a parser cannot safely reconstruct every proprietary or damaged file type.
- **Encryption:** a password-protected file cannot be inspected unless it is decrypted through an approved process.
- **Permitted-content flaws:** a vulnerability in a viewer or CDR parser can still create risk.
- **Non-file attacks:** CDR does not stop credential phishing in message text, malicious URLs, stolen accounts, or unsafe instructions.

05 Deployment checklist

1. Define accepted formats, file-size limits, nested-archive depth, and which active elements each business process needs.
2. Preserve the original in a restricted quarantine when legal, forensic, or business review requires it. Do not expose it to ordinary users.

3. Label reconstructed files and report which components were removed so users understand functional differences.
4. Provide a documented exception process for trusted senders or files. Exceptions should be narrow, approved, logged, and time-limited.
5. Test representative documents for visual fidelity, accessibility, digital signatures, formulas, and downstream automation.
6. Monitor parser failures, bypasses, latency, and requests for originals; keep CDR engines patched because they process hostile input.

06 What to do when CDR blocks a file

Verify the sender through a separate channel and determine which feature the workflow actually requires. Do not ask the sender to rename the extension, disable protection, or send the same file through a personal account. Security staff can inspect the original in an isolated environment or arrange a safer export, such as data-only CSV or a signed PDF.

07 Source

CISA describes CDR as detecting unapproved active content and facilitating its removal in its Trusted Internet Connections remote-user guidance.

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