E

Executable File: Types, Extensions, and Safety Checks

Learn what makes a file executable on Windows, macOS, and Linux, why extensions can mislead, and how to inspect a downloaded program before running it.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
3 min

CONTENTS

In this guide

01 Are executable files malicious code?
02 Common executable formats
03 Extensions do not determine safety
04 Executable files on Windows
05 Executable permissions on macOS and Linux
06 How to check a file before running it
07 Why a clean scan is not proof of safety
08 What to do after running a suspicious file
09 Safer software execution

An **executable file** contains instructions or a script that an operating system or interpreter can run. Applications, installers, command-line tools, libraries, scripts, and system components may all participate in code execution.

Executable does not mean malicious. It means the file can cause actions rather than serving only as passive data. The source, signature, expected behavior, permissions, and environment determine the risk.

01 Are executable files malicious code?

No. Operating systems and legitimate applications rely on executable files, and most are not malicious. The risk is that an executable can perform actions with the permissions of the user or service that launches it. Treat an unexpected executable as untrusted until its source, publisher, signature, and purpose have been verified.

02 Common executable formats

- Windows commonly uses Portable Executable (PE) files such as `.exe`, `.dll`, `.sys`, and `.scr`.
- macOS applications use Mach-O binaries, app bundles, packages, and scripts.
- Linux and other Unix-like systems commonly use ELF binaries and executable scripts.
- Scripts such as `.ps1`, `.js`, `.vbs`, `.sh`, and `.py` run through an interpreter.
- Installers and archives can contain multiple executable components.

03 Extensions do not determine safety

Changing a filename extension does not convert the underlying format. Attackers use double extensions, Unicode look-alike characters, misleading icons, and hidden extensions. A file named `invoice.pdf.exe` may appear to be a PDF when Windows hides known extensions.

Enable display of file extensions and inspect the complete name, but do not rely on the extension alone. File headers and trusted analysis tools provide stronger identification.

04 Executable files on Windows

Windows PE files contain headers describing architecture, sections, imports, and entry points. Authenticode can verify that a file was signed and has not changed since signing. A valid signature identifies the signing publisher; it does not guarantee that the program is safe, necessary, or free of vulnerabilities.

05 Executable permissions on macOS and Linux

Unix-like systems use permission bits and file headers rather than depending only on extensions. A script may run when it has execute permission and a valid interpreter line. macOS also applies code-signing, notarization, quarantine, and Gatekeeper checks. Do not remove quarantine or bypass a warning without independently verifying the software.

06 How to check a file before running it

1. Download software from the developer’s official site or an approved repository.
2. Verify the exact filename, publisher, version, and expected size.
3. Check the digital signature and compare a published cryptographic hash when available.
4. Scan with updated security tools.
5. Be suspicious of installers delivered through ads, unsolicited messages, cracks, or fake update pages.
6. For uncertain business files, use an authorized isolated analysis service or security team.

07 Why a clean scan is not proof of safety

New, targeted, packed, or environment-aware malware may not be detected immediately. A clean result is one signal. Source reputation, signature, observed behavior, age, prevalence, and business need also matter. Do not upload confidential files to public scanning services without permission because samples may be shared.

08 What to do after running a suspicious file

Disconnect the device when appropriate, preserve the file and its hash, and record what happened. Use a clean device to protect important accounts if credentials may be exposed. Run approved scans and review processes, startup entries, scheduled tasks, services, and outbound connections. Rebuild from a known-good source if privileged compromise cannot be ruled out.

09 Safer software execution

Use standard user accounts, application allowlisting where practical, current operating-system protections, and a sandbox or disposable virtual machine for authorized testing. Keep memory protections such as DEP enabled. Isolation reduces risk but does not authorize running unlicensed or malicious software.

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