W

Wabbit (Rabbit) Malware: Resource Exhaustion, Fork Bombs, and Recovery

A wabbit or rabbit is a locally self-replicating program that exhausts processes, memory, CPU time, or storage. Learn how it differs from a virus and worm and how to recover safely.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
4 min

CONTENTS

In this guide

01 How a wabbit causes resource exhaustion

02 Wabbit vs virus vs worm vs fork bomb

03 Warning signs

04 How to stop a wabbit safely

05 Prevention for shared and multi-user systems

06 Frequently asked questions

A **wabbit**, also called a **rabbit**, is an informal term for code that repeatedly creates copies, files, or processes on one computer until it exhausts resources. The result can be extreme slowdown, an inability to start normal programs, a full disk, or a system crash. A fork bomb is the best-known process-based example.

Key distinction: a wabbit replicates locally to consume resources. It does not need to infect other files like a virus or spread across a network like a worm. Some references use “rabbit virus” informally, but wabbit is a behavior category rather than one named malware family.

01 How a wabbit causes resource exhaustion

The program starts a loop that creates more work faster than the operating system can reclaim it. Depending on its design, it may launch child processes, write repeated files, allocate memory, open handles, or create threads. Each new instance can repeat the same action, producing exponential growth.

Resource consumed	Possible symptom
Process table or threads	New applications cannot start and the interface stops responding
CPU time	Constant high utilization, heat, fan noise, and severe lag
Memory	Heavy paging, application failures, or an out-of-memory error
Disk space or inodes	Storage fills rapidly and applications cannot save files
File handles or other limits	Services fail even when disk and memory appear available

The behavior can be malicious, a prank, an unsafe demonstration, or a programming error. The effect may look the same, so determine which parent process started the growth and how it was launched.

02 Wabbit vs virus vs worm vs fork bomb

Term	Defining behavior

Wabbit or rabbit	Uncontrolled local self-replication or resource creation
Computer virus	Attaches to or modifies a host file and spreads when that host executes
Worm	Propagates between systems or through networks without needing a host file
Fork bomb	A process-replication technique that rapidly exhausts process and CPU limits

A fork bomb is a type of wabbit behavior, but not every wabbit relies on the operating system's fork mechanism. On Windows, similar exhaustion can be produced by repeatedly starting processes; other variants may create files instead.

03 Warning signs

- hundreds or thousands of similarly named processes appear quickly;
- the same executable or script continually recreates itself;
- free storage falls rapidly without normal data growth;
- CPU, memory, process count, or file handles spike at the same time;
- the machine becomes unresponsive immediately after a script or unknown program runs;
- normal applications fail to launch because a system limit has been reached.

A failing service, runaway legitimate application, disk log loop, or hardware problem can cause similar symptoms. Preserve the parent process, command line, file hash, and startup source when possible.

04 How to stop a wabbit safely

1. **Disconnect untrusted network access.** A classic wabbit is local, but an unknown program may have other capabilities or may have arrived from a shared location.
2. **Stop the parent, not only its children.** Terminating generated processes one by one is ineffective while the original loop continues.

3. **Use a trusted recovery path.** If the interface cannot respond, boot into Safe Mode or an approved recovery environment and prevent the source from launching.
4. **Preserve a sample and logs.** Record the file, script, process tree, scheduled task, service, or startup entry before removal.
5. **Remove persistence and generated data.** Delete only verified generated files; confirm paths carefully before a bulk cleanup.
6. **Run a full scan.** Determine whether the wabbit was the entire payload or one part of a larger infection.
7. **Check system integrity.** Repair files, databases, or applications damaged when storage filled or the machine crashed.

05 Prevention for shared and multi-user systems

- apply per-user process, memory, and storage quotas where the platform supports them;
- do not run unknown scripts or command snippets copied from chats or videos;
- limit execution from temporary and user-writable folders;
- monitor rapid process creation, disk growth, and repeated crash loops;
- use least privilege so one user cannot exhaust or damage every shared service.

06 Frequently asked questions

Does a wabbit steal data?

Resource exhaustion is its defining behavior, not data theft. An unknown sample could include additional functions, so investigate and scan rather than assuming it only creates copies.

Will restarting remove it?

A restart stops the active processes, but persistence can launch the source again. Find the scheduled task, service, startup item, script, or user action that starts it.

Is a wabbit a denial-of-service attack?

Its resource exhaustion denies availability on the local host and can disrupt a shared service. Unlike a classic network DDoS attack, it may require no incoming traffic at all.

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