ACTIVE EXPLOITATION IN THE WILD · ESTIMATED 2+ BILLION AFFECTED DEVICES · NO PATCH AVAILABLE



Terminal2Shell

CVE ID pending · also tracked as "TermGeddon"

A critical flaw present in every operating system since 1971 allows anyone who opens a terminal to obtain a fully interactive shell. Instantly. With no authentication beyond the one they already did.

11.0	0 ms	55 yrs	1 click
CVSS score (out of 10)	Time to exploit	Window of exposure	Attack complexity

What is Terminal2Shell?

Terminal2Shell is a design-level vulnerability in terminal emulators. When a terminal is launched, it immediately spawns a **shell process** and connects it directly to the user's keyboard. From there, the attacker can issue arbitrary commands with the full privileges of the logged-in user.

Researchers have confirmed that the shell will execute *any* command typed into it, including commands that delete files, install software, and modify system configuration. In some cases, the shell has been observed executing commands **the user typed themselves**.

Proof of concept

We are releasing a working proof of concept below. It is fully weaponized. Click inside and start typing. Try `help`.

```
victim@localhost: ~ (pwned)

Last login: Wed Nov 11 2026 on ttys000
WARNING: Terminal2Shell exploitation in progress.
An interactive shell has been obtained. Type 'help'.
victim@localhost:~$
```

The page you are reading now has a shell in it. Close the tab if you are concerned.

Exploitation steps

1. Open a terminal.
2. ~~There is no step 2.~~

Exploitation requires local access, a working keyboard, and the ability to locate the Terminal application. Threat intelligence indicates all three are widely available to attackers.

Affected systems

Product	Vector	Status
macOS (all versions)	Terminal.app, iTerm2, Ghostty, Warp	VULNERABLE
Linux (all distributions)	gnome-terminal, Konsole, xterm, Ctrl+Alt+F2	VULNERABLE
Windows	cmd.exe, PowerShell, Windows Terminal, WSL	VULNERABLE
FreeBSD / OpenBSD	Literally the default experience	VULNERABLE
DEC VT100 (1978)	Physical serial cable	VULNERABLE
Your router	SSH, telnet, the little console port on the back	VULNERABLE
VS Code	Ctrl+` (one keystroke!)	VULNERABLE
Microsoft Paint	None known	Not affected

Indicators of compromise

If you see any of the following characters followed by a blinking cursor, your system is actively compromised:

`$` `%` `>` `❯` `#`

A `#` prompt indicates the attacker has obtained **root**. Disconnect from the network, then from society.

Mitigations

No vendor patch is available. Until one is released, we recommend the following compensating controls:

Do not open terminals

This is the most effective mitigation. Note that many developers report opening terminals "for work," which security teams should treat as a red flag.

Close terminals immediately after opening them

Partial mitigation only. Reopening the terminal reintroduces the vulnerability.

Type slower

Reduces commands per minute and therefore attack throughput.

Remove the keyboard

Effective, but the on-screen keyboard is a known bypass.

`Set SHELL=/bin/false`

Fully remediates the issue. Also fully remediates your ability to use the computer.

Disclosure timeline

- **1971** Vulnerable code ships in Unix v1 as the Thompson shell.
- **1979** Bourne shell released. Vulnerability preserved with full backwards compatibility.
- **1989** Bash released. Name confirms the attack is intentional.
- **2009** zsh adoption grows. Now the shell is also pretty, which researchers believe lures more victims.
- **2026** Researchers notify all vendors simultaneously.
- **2026** Vendors respond: "That is what a terminal is."
- **2026** Researchers register a domain, design a logo, and contact the press. Coordinated disclosure considered complete.

FAQ

▸ Am I affected?

▸ Is this being exploited in the wild?

▸ Can the attacker escalate to root?

▸ Why does it have a logo?

▸ Is this related to the other *2Shell vulnerabilities?

▸ Should I panic?

Stay vigilant. Close your terminals.