

CVE-2026-98432 Kernel Flaw: Ubuntu, RHEL, Debian Status

CVE-2026-98432 is a CVSS 9.8 Linux kernel TCP flaw. Fix status for Ubuntu, RHEL, Debian, Amazon Linux and SUSE, and how to prove each server is patched.

AUTHOR

PatchMon Team

PUBLISHED

30 October 2026

CATEGORIES

Security, Patch Management

READ ONLINE

<https://patchmon.net/blog/cve-2026-98432-linux-kernel-tcp-use-after-free-patch>

Contents

1. [The Short Version](#)
2. [What CVE-2026-98432 Actually Is](#)
 - [What an attacker gets](#)
 - [The Anthropic exploit](#)
5. [Why the Score Is 9.8 on Ubuntu but 5.9 on Red Hat](#)
6. [Which Kernels Are Fixed Upstream](#)
7. [What Your Distro Is Shipping](#)
 - [Ubuntu](#)
 - [RHEL, Rocky Linux and AlmaLinux](#)
 - [Debian](#)
 - [Amazon Linux](#)
 - [SUSE and openSUSE](#)
 - [Oracle Linux](#)
14. [If You Cannot Patch Right Now](#)
15. [How to Tell Your Fleet Is Actually Patched](#)
16. [Frequently Asked Questions](#)
17. [Bottom Line](#)
18. [Sources and Further Reading](#)

Last updated: 30 October 2026. Distro status in this post was last checked against each vendor's own tracker at 00:10 UTC on 30 October 2026. Fixes are landing day by day, so we will keep the tables below current as Ubuntu and Red Hat ship.

A Linux kernel bug you can trigger with nothing more than TCP packets is rare. **CVE-2026-98432** is one. It is a use-after-free in the IPv4 TCP handshake code, reachable by an unauthenticated peer before any application sees the connection, and the kernel's own CVE team scores it **CVSS 9.8 Critical**. On 27 October, Anthropic said one of its research models had turned it into a working remote root exploit against servers with a multipath TCP listener.

The upstream fix has been in the stable kernels since early October. The problem is the gap between upstream and your servers: Debian, SUSE, Oracle Linux and parts of Amazon Linux have shipped fixed kernels, while **Ubuntu has not yet released one and Red Hat has deferred its fix**. This post sets out where every major distro stands, what you can do in the meantime, and how to prove each host is actually *running* the fixed kernel rather than just having it on disk.

The Short Version

If you only have two minutes:

What it is: A race condition use-after-free in `tcp_v4_syn_recv_sock()` and `ip_options_build()`, in the kernel's IPv4 TCP code. IPv4 only.

Severity: CVSS 3.1 **9.8** from the kernel CVE team and Ubuntu. SUSE and Amazon score it 8.1. Red Hat scores it 5.9. Same bug, different assumptions (explained below).

Who is affected: Every kernel from **Linux 4.4 onwards** until the fixed stable releases. That is effectively every supported distro.

What an attacker needs: Network access to any listening TCP port over IPv4. No credentials, no user interaction.

Fixed upstream in: 5.10.272, 5.15.223, 6.1.190, 6.6.159, 6.12.113, 6.18.56, 7.2.10 and 7.3.

Ubuntu: Not fixed yet. Fixes are in progress for 22.04, 24.04, 26.04 and 26.10.

RHEL, Rocky, AlmaLinux: Not fixed. Red Hat lists the fix as deferred.

Debian, SUSE, Oracle Linux UEK: Fixed kernels are available now.

What you do about it: Install the fixed kernel the moment your distro ships it, then **reboot**. A patched kernel on disk does nothing until the machine boots into it.

What CVE-2026-98432 Actually Is

The [official CVE record](https://www.cve.org/CVERecord?id=CVE-2026-98432) (<https://www.cve.org/CVERecord?id=CVE-2026-98432>), published by the Linux kernel CNA on 15 October 2026, describes the bug precisely. When a TCP connection completes its handshake, `tcp_v4_syn_recv_sock()` hands the IPv4 options saved from the client's SYN to the new child socket *without copying them*. Meanwhile, another CPU can be processing a retransmitted SYN for the same connection and building a SYNACK from those same options. If the child socket is torn down first, the options are freed with a plain `kfree()` while the SYNACK code is still reading them in `ip_options_build()`.

That has been possible since commit `079096f103fa` ("tcp/dccp: install syn_recv requests into ehash table"), which shipped in Linux 4.4, because request sockets stopped being processed under the listener lock.

There is a second, narrower path through **request socket migration**, which only applies if you have set `net.ipv4.tcp_migrate_req=1` or loaded a `BPF_SK_REUSEPORT_SELECT_OR_MIGRATE` eBPF program.

The fix is small: free the options after an RCU grace period (`kfree_rcu()`), as other paths in the same code already did. **IPv6 is not affected** by the main bug, because `tcp_v6_syn_recv_sock()` copies the options. The IPv6 migration path has a similar issue limited to CALIPSO options, which the kernel developers say will be handled in a separate patch.

What an attacker gets

The kernel CNA's scoring rationale, published with the CVE, spells out all three impacts:

Information disclosure: the stale options are copied into the SYNACK sent back to the attacker, which can leak kernel heap memory over the network.

Memory corruption: reclaiming the freed object lets stale offsets steer writes in `ip_options_build()`, which "permits out-of-bounds kernel writes and potential control-flow corruption."

Denial of service: a reclaimed length can drive `skb_push()` past the packet headroom and hit `skb_under_panic()`, crashing the kernel.

The Anthropic exploit

On 27 October, [Anthropic published research](https://www.anthropic.com/research/discovering-vulnerabilities-with-claude) (<https://www.anthropic.com/research/discovering-vulnerabilities-with-claude>) describing this bug as a "remotely triggerable (zero-click)" kernel flaw, found and exploited by an unreleased internal Claude model. Their proof of concept targeted Linux servers with a **multipath TCP (MPTCP) listener**: it used the race to leak a kernel pointer and defeat KASLR, then overwrote a callback pointer to run a shell command as root. A lost race just fails that one handshake, so the attacker can keep retrying. In their test it took 48 minutes to leak the pointer and another five to reach command execution.

Anthropic has not published the exploit. As of 30 October, the CVE is not in [CISA's Known Exploited Vulnerabilities catalogue](https://www.cisa.gov/known-exploited-vulnerabilities-catalog) (<https://www.cisa.gov/known-exploited-vulnerabilities-catalog>), and we have found no vendor reporting exploitation in the wild. That is a reason to patch calmly, not a reason to wait.

Why the Score Is 9.8 on Ubuntu but 5.9 on Red Hat

Your scanner may show this CVE as Critical on one host and Moderate on another. The vendors are scoring the same bug with different assumptions:

Source	CVSS 3.1	Vector
Linux kernel CNA	9.8 Critical	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Ubuntu	9.8, priority High	Follows the CNA score
SUSE	8.1 High (CVSS 4.0: 9.2)	AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H
Amazon Linux	8.1 Important	AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H
Red Hat	5.9 Moderate (draft)	AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H

The two disagreements are:

Attack complexity. The CNA rates it Low, arguing that both sides of the race are attacker-induced and that retrying is cheap. SUSE, Amazon and Red Hat rate it High because winning a cross-CPU race is not trivial.

Impact. Red Hat treats denial of service as the practical worst case and scores confidentiality and integrity as None. Everyone else scores them High.

Anthropic's description of a working root exploit is hard to square with "availability only". Our advice is to treat this as remote code execution, whatever number your scanner prints.

Which Kernels Are Fixed Upstream

Per the [CVE record](https://www.cve.org/CVERecord?id=CVE-2026-98432) (<https://www.cve.org/CVERecord?id=CVE-2026-98432>), these stable releases contain the fix:

Stable branch	First fixed release
5.10	5.10.272
5.15	5.15.223
6.1	6.1.190
6.6	6.6.159
6.12	6.12.113
6.18	6.18.56
7.2	7.2.10
Mainline	7.3

This table only helps if you build your own kernels or run a distro that tracks upstream stable closely. Distribution kernels backport fixes onto their own version numbers. Ubuntu 24.04's **6.8.0** kernel, for example, is not an upstream stable branch at all, so a version comparison against this table tells you nothing about it. You need your distro's advisory.

What Your Distro Is Shipping

Checked against each vendor's own tracker on 30 October 2026.

Distribution	Status	Fixed kernel or advisory
Ubuntu 26.10, 26.04 LTS, 24.04 LTS, 22.04 LTS	Vulnerable , fix in progress	Not yet released
Ubuntu 20.04, 18.04, 16.04	Vulnerable	None announced
RHEL 8, 9, 10 (incl. kernel-rt)	Fix deferred	None
OpenShift 4 (RHEL CoreOS)	Fix deferred	None
RHEL 7	Not affected	n/a
Debian 13 "trixie"	Fixed	<code>linux</code> 6.12.113-1 (DSA-6568-1)
Debian 12 "bookworm"	Fixed	<code>linux</code> 6.1.190-1 (DLA-4841-1), <code>linux-6.12</code> 6.12.113-1~deb12u1 (DLA-4850-1)
Amazon Linux 2023, <code>kernel6.18</code>	Fixed 24 Oct	ALAS2023-2026-3251
Amazon Linux 2023, <code>kernel6.12</code>	Fixed 29 Oct	ALAS2023-2026-3335
Amazon Linux 2023 default <code>kernel</code> ; Amazon Linux 2	Pending fix	None
SLES 16.0, openSUSE Leap 16.0, SL Micro 6.2	Fixed	<code>kernel-default</code> 6.12.0-160000.40.1
SLES 15 SP6-LTSS	Fixed	<code>kernel-default</code> 6.4.0-150600.23.143.1
SLES 15 SP5-LTSS, SLE Micro 5.5	Fixed	<code>kernel-default</code> 5.14.21-150500.55.195.1
SLES 15 SP4-LTSS, SLE Micro 5.3/5.4	Fixed	<code>kernel-default</code> 5.14.21-150400.24.246.1
SLES 15 SP7	In progress	None
Oracle Linux 7 to 10 (UEK)	Fixed 16 Oct	ELSA-2026-500397, ELSA-2026-500398, ELSA-2026-500399

Ubuntu

Ubuntu's CVE page (<https://ubuntu.com/security/CVE-2026-98432>) rates CVE-2026-98432 at priority **High** and lists every supported kernel as Vulnerable. Fixes are marked "work in progress" in these versions, which tells you what to look for once they are released:

Kernel package	26.10	26.04 LTS	24.04 LTS	22.04 LTS
<code>linux</code> (generic)	7.3.0-10.10	7.0.0-43.43	6.8.0-149.149	5.15.0-203.213
<code>linux-aws</code>	7.3.0-1004.4	7.0.0-1017.18	6.8.0-1069.74	5.15.0-1120.127
<code>linux-azure</code>	7.3.0-1004.4	7.0.0-1020.21	6.8.0-1073.82	5.15.0-1126.135
<code>linux-gcp</code>	7.3.0-1004.4	7.0.0-1017.19	in progress	5.15.0-1121.131
<code>linux-oracle</code>	7.3.0-1004.4	7.0.0-1016.18	6.8.0-1067.71	5.15.0-1117.123

These are not released packages yet. Until Ubuntu marks them Released, `apt upgrade` will not give you a fixed kernel. Ubuntu 20.04, 18.04 and 16.04 show plain Vulnerable with no fix version, so if you still run them under Ubuntu Pro or ESM, watch the same page.

RHEL, Rocky Linux and AlmaLinux

Red Hat (<https://access.redhat.com/security/cve/CVE-2026-98432>) lists RHEL 8, 9 and 10, including the real-time kernels and OpenShift's CoreOS nodes, as **Fix deferred**, and states that no mitigation meets its criteria. RHEL 7 is not affected. Rocky Linux and AlmaLinux rebuild RHEL, so check their errata once Red Hat ships rather than assuming the date. If you run an ELRepo or other mainline kernel on these systems, compare it with the upstream table above instead.

Debian

Debian's tracker (<https://security-tracker.debian.org/tracker/CVE-2026-98432>) shows fixed kernels in the security archive for trixie (6.12.113-1) and bookworm (6.1.190-1, plus 6.12.113-1~deb12u1 for the backported 6.12 kernel). The kernels in the main bookworm and trixie suites are still the vulnerable ones, so make sure security.debian.org is in your sources.

Amazon Linux

On [Amazon Linux 2023](https://explore.alas.aws.amazon.com/CVE-2026-98432.html) (<https://explore.alas.aws.amazon.com/CVE-2026-98432.html>), only the optional `kernel6.18` and `kernel6.12` packages are fixed so far. The default `kernel` package and every Amazon Linux 2 kernel are still Pending Fix.

SUSE and openSUSE

SUSE (<https://www.suse.com/security/cve/CVE-2026-98432.html>) has released fixed kernels for SLES 16.0, the 15 SP4 to SP6 LTSS streams and several SLE Micro releases. SLES 15 SP7 is still in progress, and SLE Micro 6.0 and 6.1 are listed as affected.

Oracle Linux

Oracle released fixed [Unbreakable Enterprise Kernel](https://linux.oracle.com/cve/CVE-2026-98432.html) (https://linux.oracle.com/cve/CVE-2026-98432.html) errata for Oracle Linux 7 to 10 on 16 October, before most other vendors. Hosts running the Red Hat Compatible Kernel instead of UEK should follow Oracle's errata for that kernel separately.

If You Cannot Patch Right Now

For most Ubuntu and RHEL-family hosts, "patch now" is not yet an option. Here is what you can do in the meantime. None of it replaces the kernel update.

Keep request migration off. The secondary path needs `tcp_migrate_req` enabled, and it is off by default. Confirm it:

```
sysctl net.ipv4.tcp_migrate_req # should print 0
```

This closes only the migration path, not the main bug.

Find your MPTCP listeners. The published exploit targeted servers with a multipath TCP listener, and you may have more than you think: since [Go 1.24](https://go.dev/doc/go1.24) (https://go.dev/doc/go1.24), `net.ListenConfig` "uses MPTCP by default on systems where it is supported (currently on Linux only)". To list MPTCP sockets on a host:

```
ss -Mlnp
```

Anthropic has not said whether servers without MPTCP are exploitable for root, and the bug itself is in the plain IPv4 TCP path. Treat removing MPTCP as reducing exposure, not closing it.

Filter IPv4 Record Route options at the edge. The trigger relies on a SYN carrying IPv4 Record Route or Timestamp options, which almost no legitimate traffic uses. With nftables you can drop Record Route:

```
nft add rule inet filter input ip option rr exists drop
```

(Adjust the table and chain names to your ruleset.) We tested that this rule is accepted by nftables 1.0.9 on Ubuntu 24.04. That version **cannot match the Timestamp option**, so this blocks only part of the trigger. No vendor has confirmed edge filtering as a mitigation. Treat it as defence in depth, and test it before you roll it out.

Watch for crashes. A failed or noisy exploitation attempt can oops the kernel. Look for the functions named in the CVE:

```
journalctl -k | grep -E 'ip_options_build|ip_build_and_send_pkt|skb_under_panic'
```

Prioritise by exposure. Internet-facing hosts with open TCP ports over IPv4 come first, then anything running Go services or MPTCP, then internal hosts.

How to Tell Your Fleet Is Actually Patched

Kernel CVEs have the same trap every time, as we saw with [Copy Fail](https://patchmon.net/blog/copy-fail-cve-2026-31431-linux-kernel-patch) (<https://patchmon.net/blog/copy-fail-cve-2026-31431-linux-kernel-patch>): **a fixed kernel package installed is not a fixed kernel running**. Three things go wrong:

The kernel updated and nobody rebooted. The new kernel sits in `/boot` while the old one keeps serving traffic.

A host runs a kernel flavour you did not check. An `-aws` or `-azure` kernel on Ubuntu, `kernel6.12` on Amazon Linux, UEK versus RHCK on Oracle Linux. Each has its own fix status.

The version string tells you nothing on its own. `6.8.0-149` only means something next to Ubuntu's advisory, and a backported RHEL kernel never matches an upstream version.

On a single host, check the running kernel and then confirm the fix in its changelog. Most distro kernel changelogs reference the CVE ID:

```
# What is actually running
uname -r

# Debian / Ubuntu
apt changelog linux-image-$(uname -r) 2>/dev/null | grep CVE-2026-98432
[ -f /var/run/reboot-required ] && echo "Reboot required"

# RHEL / Rocky / Alma / Oracle / Amazon Linux
rpm -q --changelog kernel-$(uname -r) | grep CVE-2026-98432
needs-restarting -r

# SUSE
rpm -q --changelog kernel-default | grep CVE-2026-98432
```

On one server that takes a minute. Across two hundred, with Ubuntu and Red Hat fixes still to land over the coming days, the real job is keeping one list current.

This is the problem [PatchMon](https://patchmon.net/features/patch-management) (<https://patchmon.net/features/patch-management>) is built for. Every host page shows the **running kernel and the installed kernel** side by side, and hosts that have installed a new kernel without booting into it get a **Reboot pending** flag. The agent reports every installed package and the updates available for it, including whether an update is a security update. When Ubuntu and Red Hat ship their fixed kernels, they show up as pending

security updates across the affected hosts; as each host updates, reboots and reports back, it drops off the list. That gives you one answer to "which servers are still running a vulnerable kernel?" instead of an afternoon of SSH, and a timestamped record for your auditor afterwards.

The screenshot shows the PatchMon interface for the **linux-image-amd64** package. The left sidebar contains navigation links for Dashboard, Assets (Hosts, Repos, Packages), Operations (Patching, Compliance, Reporting, Docker), and System (Automation, Settings, Links). The main content area shows the package details, including the latest version (6.1.190-1), update time (1 min ago), and a list of hosts with pending updates. The host **web-prod-01** is highlighted with a 'Security Update' status and a 'Required' reboot status.

HOST	CURRENT VERSION	STATUS	SOURCE REPO	LAST UPDATED	REBOOT REQUIRED	ACTIONS
web-prod-01	6.1.176-1	Security Update	—	1 min ago	Required	Patch

The **linux-image-amd64** package across the fleet: which hosts have the 6.1.190-1 security update waiting, and which need a reboot. Demo data.

The screenshot shows the PatchMon interface for the **Hosts** page. The left sidebar contains navigation links for Dashboard, Assets (Hosts, Repos, Packages), Operations (Patching, Compliance, Reporting, Docker), and System (Automation, Settings, Links). The main content area shows the hosts list, including the total number of hosts (3), the number of hosts needing updates (1), and the number of hosts needing reboots (1). The host **web-prod-01** is highlighted with a 'Required' reboot status and a 'Security Update' status.

Friendly Name	OS	OS Version	Connection	Reporting	Reboot	Updates	Security Updates
app-prod-02	Ubuntu	24.04 LTS (Noble Numbat)	WS	Reporting	No	0	0
db-prod-01	Rocky Linux	9.6 (Blue Onyx)	WS	Reporting	No	0	0
web-prod-01	Debian	12 (bookworm)	WS	Reporting	Required	6	3

The hosts list answers the fleet question directly: how many hosts need updates, and how many need a reboot. Demo data.

If you would like to see that against your own servers, have a look at the [PatchMon pricing page](https://patchmon.net/pricing) (<https://patchmon.net/pricing>) and start a PatchMon Cloud trial from there; it takes a few minutes to get the first agents reporting. If you would rather self-host, the [Community Edition](https://patchmon.net/open-source) (<https://patchmon.net/open-source>) is free and open source.

Frequently Asked Questions

Is Ubuntu 24.04 patched for CVE-2026-98432? Not as of 30 October 2026. Ubuntu lists the `linux` package for 24.04 as Vulnerable, with a fix in progress in 6.8.0-149.149. Check [Ubuntu's CVE page](https://ubuntu.com/security/CVE-2026-98432) (<https://ubuntu.com/security/CVE-2026-98432>) for the Released status.

Is Ubuntu 22.04 patched? Not yet. The fix is in progress in 5.15.0-203.213 for the generic kernel.

Does RHEL, Rocky Linux or AlmaLinux have a fix? No. Red Hat lists the fix for RHEL 8, 9 and 10 as deferred, and Rocky and AlmaLinux rebuild from RHEL. RHEL 7 is not affected.

Which Debian kernel fixes CVE-2026-98432? 6.12.113-1 on Debian 13 trixie, and 6.1.190-1 (or 6.12.113-1~deb12u1) on Debian 12 bookworm, from the security archive.

Is IPv6 affected? The main bug is IPv4 only. An IPv6 variant exists only on the migration path and only with CALIPSO options.

Do I need to reboot? Yes, unless a livepatch for this specific CVE has been applied. Check with your livepatch provider whether one is available.

Is it being exploited in the wild? No vendor or CERT has reported in-the-wild exploitation, and it is not in CISA KEV as of 30 October 2026. A working exploit has been described by Anthropic but not published.

Bottom Line

CVE-2026-98432 is a remotely reachable kernel memory-corruption bug in code every Linux server runs, with a credible root exploit already described. The fix exists upstream and on Debian, SUSE and Oracle Linux. Ubuntu and Red Hat users are still waiting.

What to do today:

Inventory your kernels now: distro, flavour and running version on every host.

Patch and reboot wherever a fix exists: Debian, SUSE, Oracle UEK and the Amazon Linux 2023 `kernel6.x` packages.

For Ubuntu and RHEL-family hosts, reduce exposure while you wait: keep `tcp_migrate_req` at 0, find MPTCP listeners, and consider edge filtering of IPv4 options.

Install and reboot the moment Ubuntu and Red Hat release, then verify the *running* kernel, not the installed one.

The teams that get through events like this calmly are the ones who can answer "which hosts are still running the vulnerable kernel?" in minutes. If you want to see how PatchMon does that, the [pricing page](https://patchmon.net/pricing) (<https://patchmon.net/pricing>) is the place to start a trial. Either way, get those reboots scheduled.

Sources and Further Reading

[CVE.org: CVE-2026-98432](https://www.cve.org/CVERecord?id=CVE-2026-98432) (<https://www.cve.org/CVERecord?id=CVE-2026-98432>)

[Upstream fix commit e3f33b0a1d89](https://git.kernel.org/stable/c/e3f33b0a1d89d738e2b913dab519226eaf76e15b)

(<https://git.kernel.org/stable/c/e3f33b0a1d89d738e2b913dab519226eaf76e15b>)

[Ubuntu: CVE-2026-98432](https://ubuntu.com/security/CVE-2026-98432) (<https://ubuntu.com/security/CVE-2026-98432>)

[Red Hat: CVE-2026-98432](https://access.redhat.com/security/cve/CVE-2026-98432) (<https://access.redhat.com/security/cve/CVE-2026-98432>)

[Debian security tracker: CVE-2026-98432](https://security-tracker.debian.org/tracker/CVE-2026-98432) (<https://security-tracker.debian.org/tracker/CVE-2026-98432>)

[Amazon Linux: CVE-2026-98432](https://explore.alas.aws.amazon.com/CVE-2026-98432.html) (<https://explore.alas.aws.amazon.com/CVE-2026-98432.html>)

[SUSE: CVE-2026-98432](https://www.suse.com/security/cve/CVE-2026-98432.html) (<https://www.suse.com/security/cve/CVE-2026-98432.html>)

[Oracle Linux: CVE-2026-98432](https://linux.oracle.com/cve/CVE-2026-98432.html) (<https://linux.oracle.com/cve/CVE-2026-98432.html>)

[Anthropic: Discovering vulnerabilities with Claude](https://www.anthropic.com/research/discovering-vulnerabilities-with-claude) (<https://www.anthropic.com/research/discovering-vulnerabilities-with-claude>)

[Go 1.24 release notes \(MPTCP default\)](https://go.dev/doc/go1.24) (<https://go.dev/doc/go1.24>)

[Linux kernel docs: IP sysctl](https://docs.kernel.org/networking/ip-sysctl.html) (<https://docs.kernel.org/networking/ip-sysctl.html>)

[Linux kernel docs: MPTCP](https://docs.kernel.org/networking/mptcp.html) (<https://docs.kernel.org/networking/mptcp.html>)

[CISA Known Exploited Vulnerabilities catalogue](https://www.cisa.gov/known-exploited-vulnerabilities-catalog) (<https://www.cisa.gov/known-exploited-vulnerabilities-catalog>)

The open source Linux patch management platform

PatchMon gives sysadmins one dashboard for patching across Linux and FreeBSD fleets, with Windows supported alongside. Run it as a managed SaaS on PatchMon Cloud (per-host billing, 14-day trial, no fleet minimum) or self-host the AGPLv3 Community Edition on your own infrastructure.

[Start a trial: patchmon.net/pricing](https://patchmon.net/pricing)

