Recently there were several reports on the following:
Security: CVE-2026-41940 - cPanel & WHM / WP2 Security Update 04/28/2026

Security: CVE-2026-41940 - cPanel & WHM / WP2 Security Update 04/28/2026

Devon Courtney
2 hours ago · Updated

Follow

Latest Article Changes:
04/29/26 02:46PM CST: Updated article's required actions and added detection script.
04/28/26 04:36PM CST: Updated article to include patched versions.
04/28/26 03:19PM CST: Updated article to reflect changes in mitigation steps
04/28/26 12:05PM CST: Initial article published.

Cause

An authentication bypass security issue has been identified in the cPanel software (including DNSOnly) affecting all versions after 11.40.

Resolution

We have pushed out a patch for the following cPanel & WHM versions:

- 11.86.0.41
- 11.110.0.97
- 11.118.0.63
- 11.126.0.54
- 11.130.0.19
- 11.132.0.29
- 11.136.0.5
- 11.134.0.20

We have pushed out a patch for the following WP Squared version:

- 136.1.7

Required Actions

1. Update the server to one of the above-listed versions immediately via the cPanel update script:

```
/scripts/upcp --force
```

2. Once the update has been completed, verify and confirm the cPanel build version being returned and perform a restart of the cPanel service (cpsrvd):

```
/usr/local/cpanel/cpanel -V
```

```
/scripts/restartsrv_cpsrvd
```

3. Please note that if you have disabled cPanel updates or pinned your cPanel update configuration to a specific version, then these will not auto-update. Please identify and update these servers manually as a priority. Information on how to customize cPanel's Update Preferences from the Command line can be found via the following support article:

How to customize cPanel's Update Preferences from the Command Line

4. In cases where you are not able to perform the above resolution, please apply one of the following mitigations:

- Block inbound traffic on ports 2083, 2087, 2095, and 2096 at the firewall. **Or,**
- Stop cpsrvd and cpdavd:

```
whmapi1 configureservice service=cpsrvd enabled=0 monitored=0 &&
```

We are currently working on finding paths to get a patch to versions not included above, especially for versions that have higher quantities of servers. In the meantime, it is highly recommended that you follow the instructions below until you can update to a supported version above that will continue to be updated as we release patches.

> ⚠ **Warning: If your server is not running a supported version of cPanel that is eligible for this update, it is highly recommended that you work toward updating your server as soon as possible, as it may also be affected.**

Detection Script

We are also providing the following detection script to look for indicators of compromise, and checks for sessions in the filesystem.

Save the following as `ioc_checksessions_files.sh`:

```
#!/bin/bash
# Scan for compromised session files

SESSIONS_DIR="/var/cpanel/sessions"
COMPROMISED=0

echo "[*] Scanning session files for injection indicators..."

for session_file in "$SESSIONS_DIR"/raw/*; do
    [ -f "$session_file" ] || continue
    session_name=$(basename "$session_file")

    # Check if this session is/was pre-auth
```

```
preauth_file="$SESSIONS_DIR/preauth/$session_name"

# IOC 0: Session has both token_denied AND cp_security_token and
#
# token_denied is set by do_token_denied() in cpsrvd when a requ
# supplies an incorrect security token. cp_security_token is the
# attacker-injected token value. This combination indicates:
#
#   1. Attacker injected a cp_security_token via newline payload
#   2. Attacker attempted to use the injected token
#   3. cpsrvd recorded the token mismatch (token_denied counter)
#      during the exploitation window before the session was
#      fully promoted
#
# In a legitimate session:
#   - token_denied is only present after a user-initiated
#     security token failure (rare, typically from expired bookm
#   - It would never co-exist with a badpass origin AND an
#     attacker-controlled cp_security_token
#
# This IOC catches BOTH successful and failed exploitation attem
if grep -q '^token_denied=' "$session_file" && \
   grep -q '^cp_security_token=' "$session_file"; then

    # Extract values for triage context
    token_val=$(grep '^cp_security_token=' "$session_file" | hea
    denied_val=$(grep '^token_denied=' "$session_file" | head -1
    origin=$(grep '^origin_as_string=' "$session_file" | head -1
    used=$(grep -a "$token_val" /usr/local/cpanel/logs/access_lo
    external_auth=$(grep '^successful_external_auth_with_timesta

    # High confidence if origin is badpass (session was pre-auth
    if grep -q '^origin_as_string=.*method=badpass' "$session_fi
            if [ -z "$external_auth" ] && [ -z "$used" ]; then
                    echo "Found possible injected session file:
                    echo "  - No sign of usage"
            else
                echo "[!] CRITICAL: Exploitation artifact - toke
```

```
                    echo "    - cp_security_token=$token_val"
                    echo "    - token_denied=$denied_val"
                    echo "    - origin=$origin"
                    echo "    - Verdict: Session was pre-auth (badpa
                    echo "    - USED:  $used"
                    COMPROMISED=1
                fi
            # Medium confidence but still suspicious for any session
            else
                echo "[!] WARNING: Suspicious session with token_denied
                echo "    - cp_security_token=$token_val"
                echo "    - token_denied=$denied_val"
                echo "    - origin=$origin"
                echo "    - Review manually: may be legitimate token exp
            fi
        fi

        # IOC 1: Pre-auth session with authenticated attributes
        if [ -f "$preauth_file" ]; then
            if grep -qE '^successful_external_auth_with_timestamp=' "$se
                echo "[!] CRITICAL: Injected session detected: $session_
                echo "    - Contains 'successful_external_auth_with_time
                COMPROMISED=1
            fi
        fi

        # IOC 2: Any session with tfa_verified but no valid origin
        if grep -q '^tfa_verified=1' "$session_file" && \
           ! grep -q '^origin_as_string=.*method=handle_form_login' "$se
           ! grep -q '^origin_as_string=.*method=create_user_session' "$
           ! grep -q '^origin_as_string=.*method=handle_auth_transfer' '
            echo "[!] WARNING: Session with tfa_verified but suspicious
            COMPROMISED=1
        fi

        # IOC 3: Password field containing newlines (corrupted session f
        if grep -qP '^pass=.*\n.' "$session_file" 2>/dev/null; then
            echo "[!] CRITICAL: Multi-line pass value detected: $sessior
```

```
        COMPROMISED=1
    fi
done

if [ "$COMPROMISED" -eq 0 ]; then
    echo ""
    echo "[+] No indicators of compromise found."
else
    echo ""
    echo "[!] INDICATORS OF COMPROMISE DETECTED - IMMEDIATE ACTION R
    echo "    1. Purge all affected sessions"
    echo "    2. Force password reset for root and all WHM users"
    echo "    3. Audit /var/log/wtmp and WHM access logs for unautho
    echo "    4. Check for persistence mechanisms (cron, SSH keys, b
fi
```

Run this as the following:

```
# /bin/bash ./ioc_checksessions_files.sh
```

Example output originating from IP `100.96.3.23` :

```
# /bin/bash ./ioc_checksessions_files.sh
[*] Scanning session files for injection indicators...
[!] CRITICAL: Exploitation artifact - token_denied with injected cp_
   - cp_security_token=/cpsess04396539398
   - token_denied=1
   - origin=address=100.96.3.23,app=whostmgrd,method=badpass
   - Verdict: Session was pre-auth (badpass origin) with attacker-ir
[!] WARNING: Session with tfa_verified but suspicious origin: /var/c

[!] INDICATORS OF COMPROMISE DETECTED - IMMEDIATE ACTION REQUIRED
   1. Purge all affected sessions
   2. Force password reset for root and all WHM users
   3. Audit /var/log/wtmp and WHM access logs for unauthorized acces
```

```
   4. Check for persistence mechanisms (cron, SSH keys, backdoors)
File: ioc_checksessions_files.sh
```

Was this article helpful?

Yes No

21 out of 32 found this helpful

Have more questions?

Submit a Request

Return to top

Related articles

MySQL8.4 upgraded to MySQL9.7 during nightly updates.

How to Enable or Disable Service (Proxy) Subdomains

Why is my cPanel AutoSSL (Powered by Sectigo) request failing for some domains?

AutoSSL excludes www for new accounts

LFD is stopped and does not start automatically after cPanel version upgrades

Comments

0 comments

Article is closed for comments.

cPanel