Next

Rooted in Trust: Breaking HP Easy Start's macOS Privilege Boundaries (CVE-2026-12554, CVE-2026-12555, and CVE-2026-12556)

Author: Nir Yehoshua - September 2026

CVE-2026-12554, CVE-2026-12555, and CVE-2026-12556

Researcher: Nir Yehoshua, Cipher Security Labs **Publication date:** September 2026

Field	Value
Vendor	HP Inc.
Product	HP Easy Start for macOS (`com.hp.hp-easy-start`)
Affected	Versions prior to 2.16.7.260722
Fixed in	2.16.7.260722 and later
Vulnerable sample	2.16.0 (build 251010)
Advisory	HPSBPI04124
Case	PSR-2026-0072 / HP-PSRT-IR #6256

Executive summary. During a security review of HP Easy Start for macOS (v2.16.0, build 251010), Cipher Security Labs identified three vulnerabilities affecting the application's software-delivery pipeline, temporary-file handling under privilege, and transport-security posture. HP assigned **CVE-2026-12554**, **CVE-2026-12555**, and **CVE-2026-12556**, and remediated all three surfaces in **2.16.7.260722** under advisory HPSBPI04124. This article attributes each public CVE to laboratory evidence via the published CWE classifications, states what was proven in the lab, and separates a related `SWHelper` design observation that is *not* one of the three public CVEs.

The Three CVEs at a Glance

CVE	Attributed weakness	CWE	CVSS 4.0	Vector
CVE-2026-12554	Unmaintained OSPFTP download stack on the software-install path	CWE-1104	8.5 High	`CVSS:4.0/AV:L/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`
CVE-2026-12555	Predictable world-writable temp paths -> privileged write via Uninstaller	CWE-379	7.7 High	`CVSS:4.0/AV:N/AC:L/AT:P/PR:L/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`
CVE-2026-12556	Cleartext software-download transport via relaxed ATS and FTP fallback	CWE-319	7.7 High	`CVSS:4.0/AV:N/AC:L/AT:P/PR:L/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`

Overview

HP Easy Start is the path many users take to install printer software on macOS. On build **2.16.0**, three public CVEs attach to the same privileged workflow. Attributed by published CWE class: an unmaintained third-party download stack (**CVE-2026-12554**, CWE-1104), insecure temporary paths in the root Uninstaller (**CVE-2026-12555**, CWE-379), and cleartext software-download transport (**CVE-2026-12556**, CWE-319).

We validated remediation on HP Easy Start **2.16.7** (build **260722**) on 2026-07-29.

A fourth observation - use of deprecated `AuthorizationExecuteWithPrivileges` in `SWHelper` - is a risky privilege-boundary design with **conditional** exploitability. API use is confirmed; an end-to-end production package-swap TOCTOU was **not** demonstrated. It is **not** a public CVE in this disclosure.

> **Laboratory safety.** Run proofs only on systems you own or are authorized to test - preferably a snapshotted macOS VM. Do not point symlinks at production files. Do not MitM third-party users.

Threat Model

Local unprivileged user. Shares a Mac. Can create names in `/tmp` and `/private/tmp`. Cannot forge the administrator password, but can prepare the filesystem so an authorized elevation writes through attacker-chosen paths.

Network-positioned attacker. Shares a LAN segment or controls DNS in a lab. With ATS relaxed and FTP as a software-download fallback, this adversary can interfere with package acquisition. Full compromise via MitM is a *chain* that also depends on package validation - not a one-click outcome from the Info.plist alone.

Finding 1 - CVE-2026-12555: Privileged File Write/Corruption via Symlink Following

CVE-2026-12555 - Local - CWE-379 - CVSS 4.0 7.7

The HP Uninstaller component uses predictable file paths inside `/tmp` and `/private/tmp`, then opens those files while running with administrative privileges. Because symbolic links are not blocked before file access, a local attacker can redirect those writes to another file. This creates a **privileged file-write / file-corruption** primitive.

We deliberately avoid the phrase "arbitrary file write as root" here. The log writer appends application-generated log strings. Destination path control is attacker-chosen; content is not meaningfully attacker-controlled. The accurate label is **privileged file modification via predictable temporary paths**.

Vulnerable code paths

Log file path

The log file path is statically defined:

```
$installerLogfile="/tmp/com.hp.uninstaller-log.txt"

def puts(msg)
    logMsg="#{currentDate} HP Uninstaller: #{msg}"
    File.open($installerLogfile,
        File::CREAT | File::APPEND | File::RDWR, 0666) do |f|
            f.puts "#{logMsg}"
    end
end
```

If `/tmp/com.hp.uninstaller-log.txt` is a symlink, the application appends log content to the symlink's target as root.

Lock file path

```
file = File.open("/private/tmp/com.hp.software.96EB4066-D8CA-4343-ACC7-4D3CCFF4FB01",
    File::RDWR|File::CREAT, 0666)
file.flock(File::LOCK_EX)
```

The UUID is identical across installations of this build - it is not unique per install. Combined with a world-writable directory, an attacker can plant a symlink there before the elevated process opens it.

Privileged execution confirmation

The HP Uninstaller invokes shell operations through an AppleScript administrative flow. The binary contains strings indicating use of `do shell script ... with administrator privileges`. After the user approves the prompt, the Ruby script performing the vulnerable `File.open()` operations runs as root.

Exploitation scenario

1. A local unprivileged user creates a symbolic link at one of the predictable HP temporary file paths.
2. The symbolic link points to a file chosen by the attacker (for research: a disposable proof file under `/tmp`).
3. A user later launches HP Uninstaller and authorizes the administrative prompt.
4. The uninstaller runs with elevated privileges, opens the attacker-prepared path, and follows the symlink.
5. Log data is written to the target file as root.

This is particularly reliable for the log file path because multiple log entries are appended during normal execution.

Impact

- Privileged file creation as root if the destination does not yet exist
- Privileged file corruption / append if the destination already exists
- Denial of service if configuration or service-state files are damaged
- Follow-on privilege escalation only if a suitable second-stage target exists (environment-specific; not automatic)

Mode `0666` is subject to umask (often ending as `0644`); that does not remove the integrity failure.

Remediation

- Use securely generated temporary files (`mkstemp()`, `mkdtemp()`, or platform-native secure temporary APIs) instead of static names.
- Open with `O_NOFOLLOW` and `O_EXCL` as appropriate - do *not* treat `lstat()`-then-`open()` as the primary defense; that is itself check-then-use.
- After open, validate with `fstat()` on the already-open file descriptor.
- Use restrictive permissions (e.g. `0600`) and avoid privileged logs in shared world-writable directories.
- Re-evaluate whether logging must run after elevation at all.

In 2.16.7 (260722), HP removed the vulnerable Uninstaller bundle from the Easy Start package we validated; the hardcoded log/lock strings were gone from that tree.

Finding 2 - CVE-2026-12554: Unmaintained OSPFTP Download Stack

CVE-2026-12554 - Download path - CWE-1104 - CVSS 4.0 8.5

HP's CNA record classifies this CVE as **CWE-1104**: Use of Unmaintained Third-Party Components. The software-download path embeds an `OSPFTP*` stack wired into component installation - not a decorative leftover.

Evidence

Static analysis of the main binary confirms classes and selectors including:

```
OSPFTPDownloadManager
OSPFTPDownloadOperation
OSPFTPFileDownloadTask
startDownloadingWithFallbackURLSchemes:toDestinationFolderURL:
makeFileDownloadWithSourceURL:destinationFolderURL:fallbackURLSchemes:
Supported fallback URL schemes %@
FtpURL
```

Why it matters

The fallback architecture means: if the primary download scheme fails, the application can fall back to FTP for the same software component. FTP transmits data in cleartext with no encryption or integrity protection. That interacts

directly with CVE-2026-12556 (cleartext policy) and with the privileged installer path that later consumes downloaded packages.

> We do **not** claim that every install hits FTP on the happy path. We claim that an unmaintained FTP download component remained live on the software-delivery surface of a privileged installer.

Impact

- Expands the cleartext substitution surface for package bytes
- Increases maintenance and supply-chain risk inside a privileged download path
- Enables conditions under which a network adversary can interfere with acquisition when primary schemes fail

Remediation

- Remove the unmaintained OSPFTP stack from the software-download path.
- Use HTTPS-only acquisition with mandatory certificate validation.
- Bind integrity checks to the downloaded bytes (hash / signature) before privileged install.

In 2.16.7 (260722), `OSPFTPDownloadManager` / `OSPFTPFileDownloadTask` and literal `ftp://` software-download markers were removed from the validated build; software download appeared to use HTTPS endpoints instead.

Finding 3 – CVE-2026-12556: Cleartext Software-Download Transport via Relaxed ATS and FTP Fallback

CVE-2026-12556 – Network – CWE-319 (Cleartext Transmission of Sensitive Information) – CVSS 4.0 7.7

HP Easy Start v2.16.0 globally relaxes macOS App Transport Security (ATS) by setting `NSAllowsArbitraryLoads=true`, and combines that policy with cleartext FTP as a software-download fallback (via the stack in CVE-2026-12554).

Complete ATS relaxation

```
"NSAppTransportSecurity" => {
    "NSAllowsArbitraryLoads" => true
```

```
  }
```

With this flag set, ATS permits unsecured HTTP loads that would otherwise be blocked. Separately, the embedded OSPFTP download stack provides a cleartext FTP fallback, extending the application's unencrypted software-delivery surface. Hardcoded informational `http://` URLs in the binary reinforce the cleartext posture.

Clarification - what this is not

`NSAllowsArbitraryLoads=true` is **not**, by itself, a certificate-validation bypass. Certificate policy is a separate concern: the binary also exposes `_validateCertFlag` / `_validateCertificate` toggles (supporting **CWE-295** evidence). That observation is **not** a fourth public CVE.

We did not claim universal bypass of HP package signature verification, and we did not claim every install uses FTP on the happy path. Combined with the FTP fallback, a network adversary who can induce primary failure gains a cleartext substitution opportunity against package bytes the GUI may later hand to the privileged installer path.

Impact

- Cleartext HTTP permitted by relaxed ATS policy, with a separate FTP fallback in the software-download stack
- Network-positioned interference with software-component download under fallback conditions
- Integrity risk to the software delivery pipeline when combined with weak or absent package binding

Remediation

- Remove `NSAllowsArbitraryLoads`; use per-domain ATS exceptions only where absolutely necessary.
- HTTPS-only for software downloads; delete FTP from the software path.
- Make certificate validation mandatory, not a toggleable configuration.

In 2.16.7 (260722), `NSAllowsArbitraryLoads` was tightened (no longer blanket `true` for arbitrary loads), and the FTP software download path was removed as

described under CVE-2026-12554.

Design Note - Deprecated Authorization API in SWHelper (Not a Public CVE)

Design note - not CVE-2026-12554/55/56. Related privilege-boundary observation (reported to HP; not a public CVE).

HP Easy Start 2.16.0 includes a privileged helper, `SWHelper`, that installs packages as root. Instead of Apple's modern helper model, it relies on the deprecated `AuthorizationExecuteWithPrivileges` API.

What is confirmed

Static analysis of `Contents/MacOS/SWHelper` shows:

- Imports: `AuthorizationCreate`, `AuthorizationCopyRights`, `AuthorizationFree`, `dlsym`
- Strings: `AuthorizationExecuteWithPrivileges`, `system.privilege.admin`, `STPrivilegedTask`, `/usr/sbin/installer`, `/usr/sbin/pkgutil --check-signature`, `OSPPackageSignatureChecker`
- Parent validation via `spctl -a -t exec` (executable check of the calling app - not package install-type validation)
- Shared predictable path string: `/private/tmp/com.hp.software.96EB4066-D8CA-4343-ACC7-4D3CCFF4FB01`

Data-flow status (analysis gate)

Apple deprecated `AuthorizationExecuteWithPrivileges` and explicitly warns that the API poses a security concern because it can execute arbitrary tools with root privileges. Separately, privilege-boundary designs that validate and later execute objects through mutable filesystem paths can introduce TOCTOU exposure. In SWHelper, the legacy API and path-oriented workflow therefore warrant security scrutiny, although an end-to-end production race was not demonstrated.

For a production LPE claim we required a concrete path:

1. Where the `.pkg` is staged
2. What `OSPPackageSignatureChecker` / `pkgutil --check-signature` actually verifies
3. When authorization rights are obtained

4. Which filesystem object `/usr/sbin/installer` finally opens
5. Whether an unprivileged writer can mutate that object in the window

Under that criterion, a practical end-to-end package-swap race against the *production* SWHelper staging path was **not demonstrated** in our retest package (PSR follow-up). A laboratory harness can demonstrate the *class* of check-then-use race against a synthetic tool path; that is not evidence that SWHelper's production path is raceable end-to-end.

> **Publish rule applied:** treat this as *risky / deprecated design with conditional exploitability*, not as a counted public CVE and not as a proven local root exploit.

Remediation (modern)

Do not migrate to `SMJobBless` as the long-term target - Apple marks that API deprecated. Prefer:

- `SMAppService`-managed privileged LaunchDaemon / helper
- Authenticated XPC with code-signing requirements for clients
- Explicit authorization of privileged operations
- Package integrity via open file descriptor and/or cryptographic hash - not mutable paths alone

In 2.16.7 (260722), HP removed `SWHelper` and introduced `com.hp.easystart.helper` using `SMAppService` + `NSXPCConnection`, with client restriction to `com.hp.hp-easy-start` / Team ID `6HB5Y2QTA3`, and an install API that verifies SHA-256 on an open fd before invoking `/usr/sbin/installer`. Static evidence from the remediating build (LaunchDaemon `com.hp.easystart.helper`, the XPC code-signing requirement, and the SHA-256-on-fd install path) is shown below; `AuthorizationExecuteWithPrivileges` and `SWHelper` are absent from that build.

```
installPackageAtPath:withSHA256Hash:reply:
[helper] STEP 2: Opening fd (O_RDONLY|O_CLOEXEC)...
[helper] STEP 3: Computing SHA256 of fd=%d ...
[helper] STEP 3 OK: SHA256 verified
[helper] STEP 5: Launching /usr/sbin/installer -verboseR -target / -pkg %@
anchor apple generic and identifier "com.hp.hp-easy-start"
  and certificate leaf[subject.OU] = "6HB5Y2QTA3"
```

The deprecated `AuthorizationExecuteWithPrivileges` design was removed. The remaining path-based handoff to `/usr/sbin/installer` could still benefit from stronger validation-to-use binding (for example copy-to-root-owned staging or an fd-bound install path).

Disclosure Timeline

Date	Event
2026-04-02	Reported to HP PSRT (PSR-2026-0072 / HP-PSRT-IR #6256)
2026-07-02	CVE-2026-12554, CVE-2026-12555, and CVE-2026-12556 assigned
2026-07-29	HP provided remediating build 2.16.7 (260722); CSL verified the fixes
2026-08-24	Vendor bulletin HPSBPI04124 published

References

- HP Security Bulletin HPSBPI04124 - HP Easy Start for macOS, fixed in 2.16.7.260722+.
- CVE records: CVE-2026-12554 (CWE-1104), CVE-2026-12555 (CWE-379), CVE-2026-12556 (CWE-319).
- Apple Developer Documentation: App Transport Security; AuthorizationExecuteWithPrivileges (Deprecated); SMAppService; SMJobBless (Deprecated).
- https://ciphersecuritylabs.com

Technical Appendix

Artifact details for independent verification of the analyzed **2.16.0** (build 251010) sample and the remediating **2.16.7** (build 260722) build.

Artifact	SHA-256
Vulnerable main binary (2.16.0 / 251010)	`66a8ca13d3c3fdd96a9ce00c250f0203543c962385759e193fbe53a740de6402`
Vulnerable distribution zip (2.16.0 / 251010)	`11b4e7e6f8d6fc2b3ecd4eb9d40b56d259203ed7f48a12b51a85ca68be4d44f7`

Fixed main binary (2.16.7 / 260722)	`f7cce862c6d3200b3f776aec210db1991a8ea9a7528e11ecc7db81f7baa4b9c4`
Fixed privileged helper (2.16.7 / 260722)	`9273b4da8cf2c9edc83b4fb9e943928b48d267dab014919a6a1be618fcc4354a`
Fixed PoC-build zip (2.16.7 / 260722)	`e1f923f6975db41704574e468c5d5e192aa838653b5a0cde19fa73fbe72d2fd9`