Blog

VULNERABILITY RESEARCH

Spektion Discovers an Unquoted Path Vulnerability in Punto Switcher

Punto Switcher's signed binary looks clean to file scanners. The flaw is an unquoted RunDll32.exe call it makes at launch that lets a local attacker run arbitrary code, caught by Spektion Research at runtime.

David Westcott
June 17, 2026

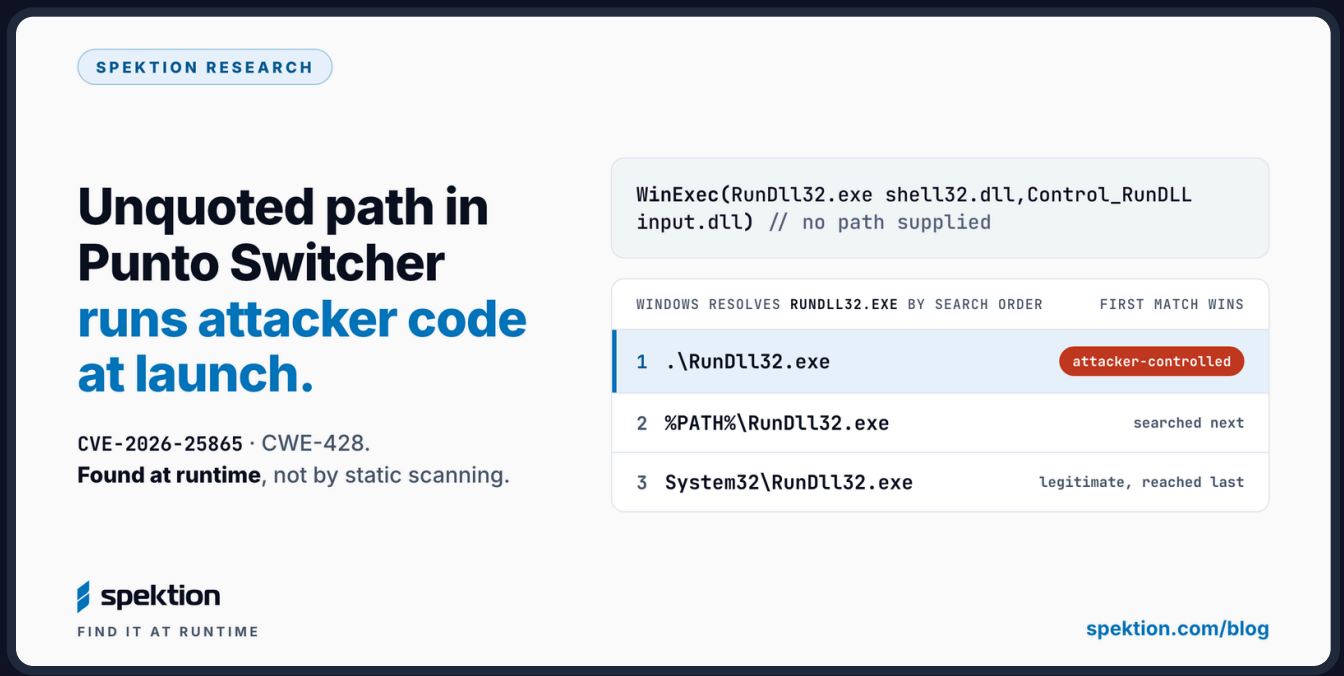

Table of contents

Spektion Research has discovered an unquoted path vulnerability (CVE-2026-25865) in Punto Switcher, an automatic keyboard layout switcher application for Windows. Punto Switcher is a Russian language/keyboard layout switching utility that Spektion has observed in use at large enterprise customers. By exploiting an unquoted executable path in Punto Switcher's application launch feature, an attacker with local filesystem access can execute arbitrary code with the privileges of the running user.

Status: *CVE-2026-25865 is currently in RESERVED status and has not yet been published. The CVE number is included here for reference. Once VulnCheck publishes its advisory and a record is available on cve.org, Spektion will update this post, including the Disclosure Timeline below, with those references.*

This post is part of an ongoing series from the Spektion Research Team documenting vulnerabilities discovered through Spektion's runtime analysis.

Key Takeaways

01 Punto Switcher version 4.5.0.583 and all earlier versions are vulnerable to arbitrary code execution via an unquoted executable path related to the product launcher (punto.exe).

02 This is an instance of CWE-428 (Unquoted Search Path or Element), a weakness class that recurs across production software because the flaw only manifests when the software executes.

03 The vulnerability was discovered through Spektion's runtime analysis, not static scanning, because the flaw only manifests when the software runs.

04 An attacker with local filesystem access can hijack Punto Switcher's process launch and execute arbitrary code with the privileges of the running user.

05 The vendor (Yandex) did not respond to multiple notifications related to this vulnerability over a 120-day period.

Background

Punto Switcher was originally developed by Sergey Moskalyov in 2007 and acquired by Yandex in 2008. It is primarily used as an automatic keyboard layout switching utility designed for native Russian speakers. The application

focuses on automatic layout correction, hotkeys, exclusions, diary, and auto-replacement features.

When the application (punto.exe) launches, Punto Switcher uses `WinExec` to call an unqualified `RunDll32.exe` to open the Windows Text Services / Input / keyboard language settings Control Panel applet (input.dll). "Unqualified" here means the application does not tell Windows where to find the file it needs to run. This is where the vulnerability lives. The weakness is an instance of CWE-428 (Unquoted Search Path or Element), and it matters beyond this one vendor: any application that launches a process without a fully qualified path is exposed to the same hijack.

Discovery

Spektion's runtime analysis identified this vulnerability on an affected endpoint. Spektion monitors software behavior during execution by observing system calls, process creation events, and API activity in real time, rather than scanning files at rest.

When Punto Switcher launched on an end-user computer, Spektion observed punto.exe issue a process creation call that resolved an executable without a fully qualified path, and flagged it. Spektion's Research Team then confirmed and reproduced the issue.

A file-based or static scanner inspects what is written on disk: the binary, its version, and its known CVEs. By that measure, Punto Switcher looks clean. The installed executable is legitimate and signed, and nothing in the file is malformed. The flaw is in the command Punto Switcher hands to Windows at launch and how Windows resolves it, and that command does not exist until the process runs. A scan of the file at rest has nothing to catch. Runtime analysis catches it by observing how the software behaves while it runs.

Root Cause Analysis

The vulnerability is caused by Punto Switcher calling `WinExec` without a fully qualified executable path. An example excerpt from Spektion's telemetry is below:

```
"process_path": "C:\Program Files (x86)\Yandex\Punto Switcher\punto.exe",
"module_path": "C:\Program Files (x86)\Yandex\Punto Switcher\punto.exe",
"call": "WinExec(RunDll32.exe shell32.dll,Control_RunDLL input.dll)"
```

When punto.exe opens the Windows keyboard language settings, it asks Windows to run `RunDll32.exe` and pass it `shell32.dll,Control_RunDLL input.dll`. The problem is the first token. `RunDll32.exe` is named with no directory in front of it. The application never tells Windows that it means the trusted system binary at `C:\Windows\System32\RunDll32.exe`.

When an executable is named without a full path, Windows does not assume System32. It walks a fixed search order and runs the first match it finds.

Under `WinExec`, that order checks the current working directory, then the directories on the PATH, before it reaches the System32 directory where the legitimate `RunDll32.exe` lives. Whichever `RunDll32.exe` Windows encounters first is the one that runs, inside the Punto Switcher process and with that user's privileges.

That gap is the vulnerability. An attacker who can write a file named `RunDll32.exe` into any directory Windows searches before System32, for example the process's current working directory at launch or a directory on the PATH, gets their executable run in place of the system binary. The attacker does not need to modify punto.exe or tamper with any signed file. They only need to place one file where the search order will find it first.

The sequence is:

1. Punto Switcher launches and calls `WinExec("RunDll32.exe ...")` with no path on the executable name.
2. Windows resolves `RunDll32.exe` by search order, checking directories that precede System32 first.
3. An attacker-placed `RunDll32.exe` in one of those earlier directories is found before the legitimate system copy.
4. The attacker's binary executes with the full security context of the Punto Switcher process and its user.

Figure 1: How an unqualified RunDll32.exe call lets an attacker-placed executable run in place of the legitimate system binary.

Microsoft's `CreateProcess` documentation advises developers to supply a fully qualified executable path for exactly this reason. The pattern is not unique to Punto Switcher. Spektion has observed it across dozens of applications. For a broader look, see Spektion's research: Unquoted Paths: The Decades-Old Flaw Still Enabling Hidden Code Execution.

Impact

An attacker with local filesystem write access, whether via a low-privileged account on the machine, a compromised file share, or a secondary vulnerability that allows file placement, can drop a malicious executable at the appropriate path location. The next time any user on the machine opens Punto Switcher, the attacker's payload executes with that user's full privileges.

If the Punto Switcher user is a local administrator or domain administrator, this results in full administrative code execution on the machine. From there, an attacker can dump credentials, establish persistent remote access, or execute additional arbitrary code.

The preconditions are realistic in exactly the environments where Punto Switcher is deployed. Shared workstations, terminal servers, and other multi-user machines common across large enterprises give a low-privileged user the foothold needed to place a file in a directory that a higher-privileged user's session will later resolve. There is no phishing, no remote exploit, and no user interaction beyond someone opening an application they already use every day.

CVSS scores this as a local-access issue, and that base vector can understate the practical risk. On a machine where an administrator runs Punto Switcher, a low-privileged foothold converts directly into administrative code execution, and the blast radius extends to everything that account can reach across the domain. The base score describes the vector; runtime context is what reveals whether a given endpoint is exposed and worth a remediation team's finite capacity.

- **CVSS v4.0 Score:** 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
- **CVSS v3.1 Score:** 7.8 (High), CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
- **CWE:** CWE-428, Unquoted Search Path or Element
- **ATT&CK TID:** T1574.009, Hijack Execution Flow: Path Interception by Unquoted Path

Affected Versions

- Punto Switcher version 4.5.0.583 and all earlier versions.

Remediation

At the time of this writing, the vendor has not acknowledged this vulnerability despite multiple attempts to notify them. As such, they have not released a fix. Since a remediation is not available from the vendor, Spektion recommends the following measures to reduce exposure:

- Audit the directories that Windows would search when Punto Switcher launches `RunDll32.exe` (particularly drive roots, the application's working directory, and Program Files parent directories) and verify that no unexpected executables exist at those locations.

- Restrict filesystem write permissions on shared machines so that low-privileged users cannot write to drive roots or intermediate path directories.
- Avoid running Punto Switcher as an elevated or administrative user for routine tasks where possible.

These mitigations reduce the attack surface but do not eliminate the vulnerability.

Researcher Credit

This vulnerability was discovered by the Spektion Research Team while running Spektion and is disclosed in accordance with Spektion's coordinated vulnerability disclosure policy. Spektion thanks its partners at VulnCheck for managing the disclosure process as its CNA.

Disclosure Timeline

- **February 9, 2026:** VulnCheck engages Yandex to initiate a CVD case under a 120-day disclosure deadline. Yandex acknowledges the notification and passes it to its internal team, noting it will reach out if interested.
- **February 18, 2026:** VulnCheck follows up after no response. Yandex reemphasizes that its internal team will reach out directly if interested.
- **June 8, 2026:** VulnCheck provides Yandex a draft of the CVE record for comments.
- **June 17, 2026:** CVE-2026-25865 published by VulnCheck.

Closing Thoughts

The unquoted path vulnerability class has been known for decades, appears in Microsoft's own API documentation as an anti-pattern, and still ships in production software today. Runtime behavior is a different attack surface than static analysis or file-based scanning can see, and this finding sits in that gap: the file is legitimate and signed, and the danger is in a call the software makes at launch. With no vendor patch available, runtime visibility is what surfaces the exposure and points to a mitigation.

RELATED CONTENT

VULNERABILITY RESEARCH

Spektion Discovers Local Privilege Escalation Vulnerability in MobaXterm That Enables Arbitrary Code Execution

Spektion Research discovered and disclosed a privilege escalation vulnerability in MobaXterm — caught at runtime, where static tools don't look.

Read article →

VULNERABILITY RESEARCH

Unquoted Paths: The Decades-Old Flaw Still Enabling Hidden Code Execution

Unquoted paths (CWE-428) remain a hidden threat in today's software. See how runtime visibility exposes what legacy vulnerability tools overlook.

Read article →

VULNERABILITY PRIORITIZATION

Same CVE, Different Risk: Why Runtime Context Changes Everything

A deep dive into how runtime evidence transforms vulnerability prioritization.

Read article →