ORIGINAL RESEARCH CVE-2026-53804

CVE-2026-53804 — OTRS PGP Configuration Command Injection

Authenticated OS command injection in OTRS Community Edition, Znuny and OTOBO through PGP executable and option settings that are interpolated into a Perl backtick command.

Published by h00die-gr3y · 2026-08-18

Overview

CVE-2026-53804 is an authenticated OS command injection vulnerability in the PGP configuration path used by OTRS Community Edition and the related Znuny and OTOBO forks.

An administrator can control the configured PGP executable and command-line options. Those values are assembled into a command string in `Kernel/System/Crypt/PGP.pm` and executed through Perl backticks while the application determines the installed GnuPG version. Because the executable and option values are not safely separated from the shell command, shell syntax supplied through the configuration can be executed by the application.

This page is the canonical technical publication for CVE-2026-53804.

Affected PGP configuration path

The vulnerable settings are exposed under the administrative PGP system configuration:

Skip to content

Admin → System Configuration → Core → Crypt → PGP

The corresponding configuration routes are:

- **OTRS Community Edition:** `/otrs/index.pl?Action=AdminSystemConfigurationGroup;RootNavigation=Core::Crypt::PGP`
- **Znuny:** `/znuny/index.pl?Action=AdminSystemConfigurationGroup;RootNavigation=Core::Crypt::PGP`
- **OTOBO:** `/otobo/index.pl?Action=AdminSystemConfigurationGroup;RootNavigation=Core::Crypt::PGP`

The relevant settings are:

- `PGP` — enables the PGP subsystem;
- `PGP::Bin` — selects the PGP executable;
- `PGP::Options` — supplies arguments that are appended to the executable command.

The affected record covers all versions of OTRS Community Edition, Znuny and OTOBO and does not list a patched version.

Exploitation flow

Exploitation requires an authenticated administrator who can change and deploy system configuration.

1. Sign in with administrative privileges.
2. Open **Core → Crypt → PGP** in System Configuration.
3. Enable `PGP`.
4. Modify `PGP::Bin`, `PGP::Options`, or both so that shell syntax reaches the generated command string.
5. Deploy the selected configuration changes.
6. Trigger a workflow that initializes the PGP subsystem. Confirmed trigger paths include creating a new email ticket, opening an existing ticket, and opening the ticket status view.
7. During PGP initialization, `_Init()` constructs `{GPGBin}` and executes it through Perl backticks to obtain the GnuPG version. The injected shell syntax executes in the application

Skip to content

There are two practical injection paths: replacing `PGP::Bin` with a shell executable, or leaving the normal GnuPG binary in place and injecting shell syntax through `PGP::Options`.

Controlled validation

Use only in an isolated test environment.

One low-impact validation replaces the PGP executable with `/usr/bin/bash` and uses a marker command as the option value:

```
PGP::Bin     = /usr/bin/bash
PGP::Options = -c 'id > /tmp/cve-2026-53804'
```

When the PGP subsystem initializes, the generated command is functionally equivalent to:

```
LC_MESSAGES=POSIX /usr/bin/bash -c 'id > /tmp/cve-2026-53804' --version
```

The presence of `/tmp/cve-2026-53804` confirms operating-system command execution without requiring a remote shell.

The second path leaves `PGP::Bin` at `/usr/bin/gpg` and places shell control characters in `PGP::Options`. For example, a controlled marker can be appended after otherwise valid GnuPG options so the shell executes it before the trailing version argument is reached.

For installations using the default service home directories, adjust PGP paths for the deployed fork (`/opt/otrs`, `/opt/znuny`, or `/opt/otobo`).

Root cause in `Kernel/System/Crypt/PGP.pm`

The vulnerable behavior is in `_Init()` in `Kernel/System/Crypt/PGP.pm`.

The implementation loads `PGP::Bin` and `PGP::Options` directly from configuration:

Skip to content

```
$Self->{GPGBin}  = $ConfigObject->Get('PGP::Bin')     || '/usr/bin/gpg';
$Self->{Options} = $ConfigObject->Get('PGP::Options') || '--batch --no-tty --yes';
```

On non-Windows systems those values are concatenated into a single shell command string:

```
$Self->{GPGBin} = "LC_MESSAGES=POSIX $Self->{GPGBin} $Self->{Options}";
```

The resulting string is then executed through Perl backticks:

```
my $VersionString = '';

eval {
    $VersionString = `$Self->{GPGBin} --version`;
};
```

The security boundary fails because configuration-controlled executable and option values become part of a shell command. No shell-safe argument separation is applied before execution, so metacharacters can alter the command that the shell interprets.

This is an instance of **CWE-77: Improper Neutralization of Special Elements used in a Command (‘Command Injection’)**.

Impact

Successful exploitation gives an authenticated administrator arbitrary operating-system command execution in the context of the OTRS, Znuny or OTOBO application process.

The vulnerability therefore crosses the application-to-host boundary. An attacker who already controls an administrative application account can execute commands with the permissions of the service account, access data available to that account, modify application or host state, and use the resulting host-level execution as a pivot for further compromise.

The finding is recorded as **high severity**. No CVSS score or vector is assigned in the technical

Skip to content

blication.

Relationship to CVE-2017-16921

The same PGP command-construction pattern was previously associated with CVE-2017-16921. CVE-2026-53804 records that the unsafe behavior remains present in the OTRS-derived products covered by this research.

Mitigation status

No fixed version is listed at publication time, and the recorded mitigation status is “No fix yet.”

Until patched releases or project-specific remediation are available, defensive measures should focus on reducing exposure to the vulnerable configuration path:

- restrict System Configuration access to trusted administrators only;
- disable PGP where it is not required;
- monitor changes to `PGP::Bin` and `PGP::Options`;
- keep `PGP::Bin` pinned to an expected executable path and reject unexpected option values;
- monitor the application service account for unusual child processes, shell execution and outbound network activity.

The code-level fix should avoid executing a shell command assembled from configuration strings. The PGP executable and its arguments should be validated and passed to a process API as separate arguments rather than interpolated into a backtick command.

CVE and disclosure

- **CVE:** CVE-2026-53804
- **Publication:** h00die-gr3y Original Research
- **Publication date:** 2026-08-18
- **Finder:** h00die-gr3y
- **Weakness:** CWE-77
- **status:** No fix yet

Skip to content

References

- CVE-2026-53804
- CVE-2017-16921

Credits

Discovery and finder credit: **h00die-gr3y**.

RESEARCH METADATA

RESEARCH TYPE
Original research

VULNERABILITY
OS command injection, Remote code execution

AUTHENTICATION
Authenticated

PRIVILEGES
Administrator

PLATFORM
Web application, Linux / Unix

VENDOR
OTRS / Znuny / OTOBO

PRODUCT
OTRS Community Edition, Znuny, OTOBO

CWE
CWE-77

PUBLISHED
2026-08-18

PUBLICATION

AFFECTED VERSIONS

- OTRS Community Edition — all versions
- Znuny — all versions
- OTOBO — all versions

Skip to content

h00die-gr3y

DISCLOSURE

STATUS
Published

DISCLOSURE
2026-08-18

CREDIT
h00die-gr3y — Finder

PUBLICATION PROVENANCE

First published by h00die-gr3y as part of this research site.