MINISTERSTVO INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA A INFORMATIZÁCIE SLOVENSKEJ REPUBLIKY

CSIRT.SK

Pri úniku od lží: Od variant SQL k vzdialenému vykonávaniu kódu v OTRS

CVE-2026-48188

SQL injekcia spôsobená nesprávnou konfiguráciou, ktorá vedie k obídeniu overovania a vzdialenému vykonaniu kódu

Vytvoril: CSIRT.SK
Ministerstvo investícií, regionálneho rozvoja a informatizácie SR
Pribinova 25
811 09 Bratislava

Dátum vzniku: Jún 2026

TLP: Clear

Úvod

Moderné aplikácie sa často spoliehajú na mechanizmy sanitizácie údajov v databáze, o ktorých predpokladajú, že sú univerzálne. Čo sa však stane, keď databáza bez upozornenia zmení pravidlá?

V tomto článku sa zaoberáme nenápadnou, ale závažnou zraniteľnosťou v inštaláciách systému OTRS, ktoré využívajú MySQL alebo MariaDB. Ak je povolený určitý režim SQL (`NO_BACKSLASH_ESCAPES`), prestávajú platiť predpoklady týkajúce sa ošetrenia špeciálnych znakov v reťazcoch, čím sa inak bezpečné dopyty menia na základné prvky SQL injekcie.

Táto nezhoda medzi logikou aplikácie a správaním databázy môže v konečnom dôsledku viesť k **obídeniu overovania a vzdialenému vykonaniu kódu**.

Kľúčový problém

Jadrom tejto zraniteľnosti je nesúlad medzi spôsobom, akým aplikácia ošetruje špeciálne znaky vo vstupoch, a spôsobom, akým ich interpretuje databáza.

OTRS na bezpečné začlenenie užívateľských vstupov do dopytov SQL ošetruje špeciálne znaky použitím spätnej lomky. Pri bežnom správaní MySQL to funguje podľa očakávania.

Avšak pokiaľ je povolený režim SQL `NO_BACKSLASH_ESCAPES`, **spätná lomka stráca svoj význam**.

Príklad

```
``` sql
-- Aplikácia vyhodnotí, že toto je bezpečné
SELECT * FROM users WHERE login = 'admin\' OR 1=1 --';
```
```

Pri bežných podmienkach reťazec `\'` ošetruje znak úvodzovky.

Keď je však povolený `NO_BACKSLASH_ESCAPES`:

- Znak spätnej lomky sa považuje za bežný znak
- Znak úvodzovky nie je ošetrený
- Štruktúra dotazu sa naruší
- Vzniká možnosť injekcie

Kľúčový postreh

➢ Aplikácia sa domnieva, že ošetruje vstupné údaje. Databáza s tým nesúhlasí.

Hlavná príčina

Hlavnou príčinou nie je len nesprávne ošetrovanie znakov, ale spoliehanie sa na **predpokladanú sémantiku ošetrovania**, ktorá nie je zaručená vo všetkých konfiguráciách databázy.

Hlavná príčina v kóde

Problém vyplýva zo spôsobu, akým systém OTRS vykonáva manuálne ošetrovanie SQL vo svojej abstrakčnej vrstve databázy.

V implementácii ovládača MySQL sa užívateľské vstupy ošetrujú nahradením určitých znakov, vrátane jednoduchých úvodzoviek:

```perl
sub Quote {
    my ( $Self, $Text, $Type ) = @_;

    if ( defined ${$Text} ) {
        if ( $Self->{'DB::QuoteBack'} ) {
            ${$Text} =~ s/\\/$Self->{'DB::QuoteBack'}\\/g;
        }
        if ( $Self->{'DB::QuoteSingle'} ) {
            ${$Text} =~ s/'/$Self->{'DB::QuoteSingle'}'/g;
        }
        if ( $Self->{'DB::QuoteSemicolon'} ) {
            ${$Text} =~ s/;/$Self->{'DB::QuoteSemicolon'};/g;
        }
        if ( $Type && $Type eq 'Like' ) {
            if ( $Self->{'DB::QuoteUnderscoreStart'} || $Self->{'DB::QuoteUnderscoreEnd'} ) {
                ${$Text} =~ s/_/$Self->{'DB::QuoteUnderscoreStart'}_$Self->{'DB::QuoteUnderscoreEnd'}/g;
            }
        }
    }
    return $Text;
}
```

S konfiguráciou:

```perl
$Self->{'DB::QuoteSingle'} = '\\';
```

V dôsledku tohto sa jednoduché úvodzovky ošetria nasledovne: `' → \\'`.

Prečo to nefunguje

Tento prístup predpokladá, že databáza bude interpretovať `\'` ako ošetrený znak úvodzovky.

Avšak keď MySQL/MariaDB beží s `NO_BACKSLASH_ESCAPES`:

- Znaky spätnej lomky sa považujú za bežné znaky
- `\'` už nie je platná ošetrovacia sekvencia
- Úvodzovka ukončuje reťazec tak, ako je

V dôsledku toho sa logika ošetrenia špeciálnych znakov stáva neúčinnou, čo umožňuje injekciu.

➢ Aplikácia vynucuje ošetrenie špeciálnych znakov. Databáza ich však bez upozornenia deaktivuje.

Poznámka

Hoci uvedený príklad kódu odkazuje na OTRS Community Edition, rovnaká logika ošetrovania špeciálnych znakov a predpoklady sa nachádzajú aj v oficiálnej kódovej základni OTRS, čo bolo potvrdené v rámci koordinovaného zverejnenia.

Prečo je to „podprahové“

Nejedná sa o typickú SQL injekciu spôsobenú absentujúcou sanitizáciou.

Je to:

- **Zraniteľnosť, ktorá vzniká konfiguráciou**
- Spúšťaná **režimom databázy SQL**
- Spôsobená **neplatnými predpokladmi o správaní pri ošetrovaní špeciálnych znakov**

To ju robí obzvlášť nebezpečným, pretože:

- Kód aplikácie sa môže javiť ako správny
- Pri bezpečnostných kontrolách môže zostať nepovšimnutá
- Prejavuje sa len v určitých podmienkach

- Bezpečnostné predpoklady nezlyhávajú nápadne — zlyhávajú nenápadne, keď sa zmení prostredie.

Reťazec útoku

Hoci už samotná SQL-injekcia predstavuje vážny problém, skutočný dopad spočíva v tom, ako ju možno prepojiť s existujúcimi funkciami v systéme OTRS.

SQL injekcia

↓

Obídenie prihlasovania

↓

Administrátorský prístup

↓

Inštalácia balíkov

↓

Vzdialené vykonávanie kódu

Vzťah k predchádzajúcemu výskumu

V predchádzajúcom výskume som skúmal, ako je možné zneužiť administratívne funkcie systému OTRS na vzdialené vykonanie kódu:

- When Admin Features Become RCE: A Case Study in OTRS Package Design
  - https://habuon.github.io/2026/02/23/When-Admin-Features-Become-RCE-A-Case-Study-in-OTRS-Package-Design.html
  - https://csirt.sk/wp-content/uploads/2026/03/OTRS_SK.pdf

Tento výskum vychádzal z predpokladu, že útočník už mal administratívny prístup.

Tento nový objav zásadným spôsobom mení model hrozby:

- Umožňuje prístup k administratívnym funkciám **bez overenia identity**.

Overenie koncepcie (prehľad)

Cieľ

Cieľom overenia koncepcie je preukázať, že zmena režimu SQL zneplatňuje predpoklady týkajúce sa ošetrovania špeciálnych znakov a umožňuje obísť overovanie identity.

Prostredie

- OTRS s MySQL alebo MariaDB
- Režim SQL: `` `NO_BACKSLASH_ESCAPES` ``

Hlavná myšlienka

Táto injekcia využíva únik z reťazca v úvodzovkách, čo umožňuje zneplatnené ošetrenie pomocou spätnej lomky.

Príklad vstupu:

```
admin\' OR 1=1 --
```

Ukážka (koncepčná)

Zadaním špeciálne upraveného prihlasovacieho údaja sa výsledný SQL dopyt vyhodnotí ako pravdivý, čím sa obídu overovacie kontroly.

Nie sú potrebné žiadne chyby ani varovania v aplikácii — toto správanie závisí výlučne od interpretácie databázy.

Reprodukovanie problému

! Nasledujúce kroky sú určené výlučne na testovanie v kontrolovanom prostredí.

1. Nastavenie prostredia

Tento problém je možné reprodukovať pomocou štandardného nasadenia OTRS Community Edition v prostredí Docker s backendom MySQL.

2. Zmena režimu SQL

Akonáhle je prostredie spustené, upravte režim SQL v MySQL tak, aby obsahoval `` `NO_BACKSLASH_ESCAPES` ``:

```bash
docker exec -it <mysql_container_id> mysql -uroot -p -e "SET GLOBAL sql_mode =
CONCAT(@@sql_mode, ',NO_BACKSLASH_ESCAPES');"
```

V predvolenom nastavení Dockeru je heslo používateľa root zvyčajne:

```
changeme
```

3. Vyvolanie zraniteľnosti

Po zmene režimu SQL prestáva logika ošetrovania špeciálnych znakov aplikácie fungovať v dôsledku zmeny spôsobu interpretácie spätných lomiek.

V tomto momente je možné obísť logiku overovania pomocou špeciálne upraveného vstupu, ktorý narúša očakávanú štruktúru SQL dopytu.

4. Skript na overenie koncepcie (PoC)

Na demonštrovanie obídenia overovania je k dispozícii sprievodný skript v jazyku Python, ktorý umožňuje reprodukovateľné vykonanie tohto postupu.

Kompletný PoC vrátane nastavenia a automatizácie je k dispozícii v repozitári:

https://github.com/Habuon/CVE-2026-48188

Prečo je tento PoC dôležitý

Mnohí vývojári po otestovaní tohto kódu dospeli by k záveru, že je bezpečný.

Tento PoC dokazuje, že správnosť závisí od konfigurácie databázy — čo je často mimo priamej kontroly aplikácie.

Preto je tento problém v reálnych prostrediach obzvlášť nebezpečný.

Zasiahnuté konfigurácie

Tento problém sa vyskytuje za nasledujúcich podmienok:

- OTRS s MySQL alebo MariaDB
- Režim SQL zahŕňa `NO_BACKSLASH_ESCAPES`

Dopad

V zasiahnutých prostrediach môže útočník:

- Vykonávať SQL injekciu
- Obchádzať prihlasovanie
- Získať administrátorský prístup
- Vykonávať ľubovoľný kód v systéme

V skratke:

- Vzdialený neautentifikovaný útočník môže potenciálne plne kompromitovať systém.

Aspekty detekcie

Detegovanie tohto problému môže byť náročné:

- Autentifikačné logy môžu zobrazovať iba úspešné prihlásenia
- Chyby SQL sa nemusia generovať
- Škodlivé pokusy o prihlásenie sa môžu javiť ako bežné pokusy o prihlásenie

Odporúča sa monitorovať nezvyčajné správanie pri overovaní.

Lekcie obrany

Táto zraniteľnosť poukazuje na niekoľko dôležitých ponaučení pre bezpečný vývoj:

1. Nikdy sa nespoliehajte na samotné ošetrenie špeciálnych znakov

Mechanizmy ošetrovania sú krehké a závisia od prostredia.

Namiesto toho používajte parametrizované dopyty (prepared statements).

2. Považujte konfiguráciu za súčasť útočnej plochy

Bezpečnosť nie je len o kóde.

Nastavenia databázy, prepínače počas behu programu a rozdiely v prostredí môžu spôsobiť vznik zraniteľností.

3. Testujte v rôznych režimoch SQL

Aplikácie by sa mali testovať v rôznych konfiguráciách databázy, vrátane:

- NO_BACKSLASH_ESCAPES

- Režimy Strict
- Režimy Compatibility

4. Overte bezpečnostné predpoklady

Ak vaša aplikácia predpokladá určité správanie:

- Zdokumentujte to
- Vynúťte to
- Alebo úplne odstráňte túto závislosť

Zverejnenie informácií

Táto chyba bola zodpovedne nahlásená bezpečnostnému tímu OTRS Product Security Team prostredníctvom procesu koordinovaného oznamovania.

Časová línia

- **2026-03-06** Zraniteľnosť zistená v priebehu bezpečnostného výskumu týkajúceho sa overovania v systéme OTRS a správania pri práci s databázou.
- **2026-03-18** Problém bol súkromne nahlásený bezpečnostnému tímu OTRS Security Team spolu s technickými podrobnosťami a informáciami o tom, ako sa dá problém reprodukovať.
- **2026-04-10** Dodávateľ potvrdil existenciu tejto chyby a overil jej dopad v rámci zasiahnutých konfigurácií MySQL/MariaDB.
- **2026-05-21** Chybe bol priradený identifikátor **CVE-2026-48188**.
- **2026-06-01** Koordinované zverejnenie informácií a vydanie oznámenia.

Hodnotenie dodávateľa

CVSS v4.0

Critical — **9.1**

CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N/AU:Y/R:U/V:D/RE:L/U:Amber

CVSS v3.1

Critical — **9.1**

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

Záver

Táto zraniteľnosť nám pripomína, že bezpečnosť nespočíva len v písaní správneho kódu — ide o zabezpečenie toho, aby vaše predpoklady platili v každom prostredí.

Nebola to len chyba v kóde. Bola to chyba v predpokladoch.

Keď tieto predpoklady zlyhajú, aj dobre napísané aplikácie sa môžu stať zraniteľnými neočakávanými spôsobmi.