MINISTERSTVO
INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA
A INFORMATIZÁCIE
SLOVENSKEJ REPUBLIKY

CSIRT.SK

Keď z administrátorských funkcií vznikne RCE: prípadová štúdia OTRS Package Design

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

Dátum vytvorenia: marec 2026

TLP: Clear

Úvod

Väčšina moderných webových aplikácií funguje na základe predpokladu, že nikomu nemožno úplne dôverovať, ani administrátorom. Niektoré architektonické rozhodnutia však administrátorom stále poskytujú možnosti, ktoré efektívne stierajú hranicu medzi *správou* a *vykonávaním ľubovoľného kódu*.

V tomto príspevku budem analyzovať rozhodnutie týkajúce sa dizajnu v tiketovacom nástroji OTRS: možnosť administrátorov inštalovať balíky rozšírení (`.opm`), ktoré môžu počas inštalácie vykonávať ľubovoľný kód Perl.

Nejedná sa o zraniteľnosť v klasickom zmysle. Neexistuje žiadne obídenie overovania, žiadna chyba umožňujúca injektovanie a žiadny únik zo sandboxu.

Je to však **nebezpečný dizajnový vzorec**, ktorý mení *prístup administrátora* na *vykonávanie príkazov na serveri*, pričom operátori si často neuvedomujú bezpečnostné dôsledky.

Príklad demonštrujúci toto správanie (PoC) si môžete pozrieť v mojom repozitári GitHub: [OTRS PoC Repository] (https://github.com/Habuon/exploits/tree/main/otrs)

Pozadie: OTRS Package Management

OTRS podporuje formát balíkov (`.opm`), ktoré slúžia na rozšírenie systému o nové funkcie. Tieto balíky môžu definovať logiku inštalácie pomocou blokov `<CodeInstall>`:

```xml
<CodeInstall><![CDATA[
   use strict;
   use warnings;
   # arbitrary Perl code
]]></CodeInstall>
```

Tento kód sa automaticky vykoná, keď správca nainštaluje balík prostredníctvom webového rozhrania.

Z hľadiska funkčnosti to dáva zmysel:

- Balíky musia modifikovať konfiguráciu
- Môžu potrebovať inštalovať súbory
- Môžu potrebovať interagovať so systémom

Z hľadiska bezpečnosti to však znamená:

- ➢ Inštalácia balíka = vykonanie ľubovoľného kódu na strane servera.

Detaily konceptu

Škodlivý balík môže do svojej inštalačnej rutiny vložiť vykonanie príkazov operačného systému:

```perl
my $content = `id`;
```

Tento príkaz sa vykonáva v kontexte používateľa spúšťajúceho službu OTRS.

S minimálnym kódom dokáže:

- Vykonávať ľubovoľné príkazy shellu
- Zapisovať výstup do koreňového adresára webu
- Vytvoriť perzistenciu
- Exfiltrovať tajomstvá
- Nasadzovať zadné vrátka

Ukážkový kód Python (PoC) automatizuje tento proces:

1. Prihlásením sa ako správca
2. Nahratím upraveného súboru `.opm`
3. Spustením inštalácie
4. Načítaním výstupu príkazu

Úplný ukážkový kód si môžete pozrieť v mojom repozitári GitHub tu: [OTRS PoC Repository] (https://github.com/Habuon/exploits/tree/main/otrs)

Nie sú potrebné žiadne reťazce exploitov. Žiadne poškodenie pamäte. Žiadna chyba deserializácie. Len zamýšľaná funkcia.

Prečo to nie je CVE

Toto správanie je:

- Autentifikované
- Zamýšľané
- Súčasťou dizajnu systému

Preto podľa väčšiny definícií nejde o zraniteľnosť.

„Nie je to zraniteľnosť“ však neznamená „bezpečný dizajn“.

Dopad na bezpečnosť

Tento dizajn spôsobuje zrútenie vrstiev privilégií:

Vrstva	Zamýšľané
Webový administrátor	Kontrola aplikácie
Systémový používateľ	Exekúcia na úrovni OS

Tým, že umožňujú vykonávanie ľubovoľného kódu v rámci rutiny inštalácie balíka, sa tieto dve vrstvy stávajú ekvivalentnými.

V praxi to znamená:

- Kompromitovanie administrátora OTRS = kompromitovanie servera.

Tým sa porušuje kľúčový bezpečnostný princíp:

Administrátorské oprávnenie ≠ vykonávanie ľubovoľného kódu.

Porovnanie s inými platformami

Moderné modulové systémy (WordPress, prehliadače, IDE) čoraz viac:

- Obmedzujú rozšírenia (sandboxing)
- Obmedzujú prístup k súborovému systému
- Zakazujú hooky na vykonávanie surového kódu
- Používajú deklaratívne kroky inštalácie

Model OTRS má bližšie k:

- „Spustite tento skript Perl ako súčasť inštalácie.“

Čo je výkonné, ale extrémne nebezpečné.

Získané poučenie

1. **Podpísané alebo dôveryhodné rozšírenia nie sú štandardne bezpečné** – dôvera by sa mala minimalizovať, nie maximalizovať.
2. **Administrátorské panely by nemali byť shellovými rozhraniami** – účet webového administrátora by nemal implicitne povoľovať vykonávanie príkazov operačného systému.
3. **Systémy rozšírení by mali byť deklaratívne, nie imperatívne** – inštalácia by mala popisovať, *čo* treba urobiť, nie *ako vykonávať kód*.
4. **Dopad po overení je dôležitý** – mnohé kompromitácie začínajú odcudzením prihlasovacích údajov.

Odporúčanie pre obranu

Ak prevádzkujete inštanciu OTRS:

• Prísne obmedzte administrátorské účty

• Kompromitáciu administrátorského účtu považujte za úplnú kompromitáciu servera

• Nevystavujte administrátorské rozhranie OTRS do verejného internetu

• Používajte izoláciu na úrovni operačného systému (kontajnery, chroot, SELinux, AppArmor)

• Monitorujte aktivitu inštalácie balíkov

• Zvážte použitie súborového systému len na čítanie pre webové procesy, kde je to možné

Záver

Táto prípadová štúdia sa netýka zraniteľnosti. Týka sa voľby dizajnu, ktorá vedie k nebezpečnému narušeniu bezpečnostných princípov.

Keď systémy rozšírení umožňujú vykonávanie ľubovoľného kódu počas inštalácie, správcovia už nie sú len správcami – stávajú sa používateľmi shellu.

V kontrolovaných prostrediach to môže byť prijateľné, ale malo by to byť explicitne dané a zrozumiteľné riziko.

Najnebezpečnejšie chyby nie sú niekedy vôbec chybami – sú to funkcie.