

# *VulnHub - badstore-1.2.3*

## **badstore 1.2.3**

### **Welcome to badstore.net**

<https://www.vulnhub.com/entry/badstore-123,41/>

Welcome to Badstore.net

Badstore.net is dedicated to helping you understand how hackers prey on Web application vulnerabilities, and to showing you how to reduce your exposure. Our Badstore demonstration software is designed to show you common hacking techniques.

Source: <http://www.badstore.net/>

v1.0 – Original version for 2004 RSA Show

v1.1 – Added:

- More supported NICs.
- Referrer checking for Supplier Upload.
- badstore.old in /cgi-bin/
- Select icons added to the /icons/ directory.

v1.2 – Version presented at CSI 2004

Added:

- ◇ Full implementation of MySQL.
- ◇ JavaScript Redirect in index.html.
- ◇ JavaScript validation of a couple key fields.
- ◇ My Account services, password reset and recovery.
- ◇ Numerous cosmetic updates.
- ◇ 'Scanbot Killer' directory structure to detect scanners.
- ◇ favicon.ico.

- ◇ Reset files and databases to original state without reboot.
- ◇ Dynamic dates and times in databases.
- ◇ Additional attack possibilities.

Source: BadStore\_Manual.pdf

**recon**

**recon**

**lineup recon**

## VulnHub: Badstore 1.2.3 – Writeup

**Status:** Erfolgreich kompromittiert

**Angriffsmaschine:** Kali Linux


**Zielhost:** `bad.store`

### 1. Recon (Informationsgewinnung)

Methode	Ergebnis
<code>nmap -sS -sV -T4 -Pn bad.store</code>	Nur Port 80/tcp offen → Apache HTTP Server 1.3.20
Weboberfläche	Erreichbar unter <code>http://bad.store</code> → BadStore Webshop
<code>dirb http://bad.store</code>	Entdeckte <code>cgi-bin/badstore.cgi</code> , <code>supplier</code> , <code>accounts</code> , <code>admin</code>
Manuelle Analyse	Suchfeld und Feedback-Formular identifiziert; sichtbare SQL-Fehlermeldungen
Dokumentation (Manual v1.2)	Test-Account <code>big@spender.com</code> : money erwähnt, diverse Schwachstellen gelistet

✅ **Gesichert:** Zugang zur Webanwendung, verwundbare CGI-Skripte, funktionierende Benutzeroberfläche.

### Zusammenfassung

Phase	Status	Hinweise / Aktionen
Recon	✅ abgeschlossen	Webserver erkannt, CGI-Skripte, Upload-Form entdeckt
Exploitation	✅ aktiv	Login + Upload möglich, SQLi und Base64-Leak entdeckt
Privilege Escalation	🕒 ausstehend	Shell aus Upload steht bevor
Post-Exploration	 begonnen	Benutzerinfos extrahiert, Shell-Ausführung vorbereitet

## Nächste empfohlene Schritte

1. **Webshell hochladen und auffinden**
2. **Command Execution über `cmd=` testen**
3. **TTY-Stabilisierung & System-Shell erzeugen**
4. **Privilege Escalation (SUID-Binaries, `sudo -l`, PATH-Hijack)**

Wenn du willst, bereite ich dir gleich eine passende Reverse Shell (`bash`, `nc`, oder `php`) + automatisierten Upload/Scan vor. Weiter? 🕶

# exploitation

## exploitation

### lineup exploitation

## 💣 2. Exploitation (Ausnutzung von Schwachstellen)

### ✅ A. Login-Schwachstelle

• **Testaccount:** `big@spender.com` mit Passwort `money`

• Login erfolgreich → Rolle: **"Supplier"**

• Zugang zu geschütztem Bereich:

`http://bad.store/cgi-bin/badstore.cgi?action=supplierportal`

### ✅ B. Upload-Funktion im Supplier-Portal

◇ Funktion: Upload von Preislisten (`.txt`) → kein MIME-Type-Check festgestellt

◇ Formular akzeptiert beliebige Dateinamen (manipulierbar)

◇ Möglichkeit zum Hochladen einer `.php`-Webshell (z. B. `shell.php`)

◇ Uploadzielpfad noch nicht sicher ermittelt → **Nächster Schritt**

### 💣 C. SQL Injection Hinweise

◇ Im `searchquery`-Parameter → SQL-Fehlermeldung bei falscher Syntax

◇ Beispiel: `searchquery=test') OR 1=1 --`

◇ Rückmeldung mit SQL-Fehler auf Zeile 207 in `badstore.cgi`

◇ **(noch nicht ausgenutzt, aber validiert)**

### 🔍 D. Accountdaten ausgelesen

- ◇ Datei: `/supplier/accounts` öffentlich erreichbar
- ◇ Inhalt: Base64-codierte Datensätze
- ◇ Erfolgreich dekodiert: `joeuser`, `janeuser`, `kbookout` mit Passwörtern & IPs

# *privilege escalation*

## **privilege escalation**

### **lineup privelege escalation**



## **3. Privilege Escalation**

**Status:** n.a.

Es liegt derzeit **keine Root- oder System-Shell** vor.

→ Eskalation auf Betriebssystemebene kann ggf. durch:

◇ **Command Injection**

◇ **Upload & Ausführung der Shell**

◇ **LFI (Local File Inclusion)**

...im weiteren Verlauf getestet werden.

# *post-exploration*

## **post-exploration**

### **lineup post-exploration**



#### **4. Post-Exploration**

Punkt	Status
Benutzeraccounts	✓ erfolgreich dekodiert (/supplier/accounts)
Uploadpfad der Shell	? noch nicht sicher verifiziert
Session-Handling	Cookies speicherbar, aber noch keine Rolle-Eskalation
Admin Panel Zugriff	über ?action=admin nicht möglich ohne höhere Rechte
Manuelle Feldmanipulation	ohne Wirkung (z. B. role=admin)