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: Successfully compromised
Attacking Machine: Kali Linux
Target Host: `bad.store`

🔍 1. Recon (Information Gathering)

Method	Result
nmap -sS -sV -T4 -Pn bad.store	Only port 80/tcp open → Apache HTTP Server 1.3.20
Web interface	Accessible at http://bad.store → BadStore Webshop
dirb http://bad.store	Discovered cgi-bin/badstore.cgi, supplier, accounts, admin
Manual analysis	Search field and feedback form identified; visible SQL error messages
Documentation (Manual v1.2)	Mentions test account big@spender.com : money, various vulnerabilities listed

✅ **Confirmed:** Access to web application, vulnerable CGI scripts, functional frontend.

🧩 Summary

Phase	Status	Notes
Recon	✅ completed	Service detection & web path enumeration
Exploitation	✅ active	Login, file upload, SQLi confirmed
Privilege Escalation	⏳ pending	Shell upload path currently being verified
Post-Exploration	🔍 started	Account extraction & shell deployment preparation

🚀 Recommended Next Steps

1. **Upload and locate the webshell**

2. **Test command execution via `cmd=`**

3. **Stabilize shell & gain system access**

4. **Privilege escalation (e.g. SUID binaries, `sudo -l`, PATH hijack)**

exploitation

exploitation
lineup exploitation

💥 2. Exploitation

✅ A. **Login Vulnerability**

• **Test account:** `big@spender.com` with password `money`

• Login successful → Role: **"Supplier"**

• Access to protected area:
`http://bad.store/cgi-bin/badstore.cgi?action=supplierportal`

✅ B. **Upload Function in Supplier Portal**

◇ Function: Upload of price lists (`.txt`) → no MIME type check observed

◇ Form accepts arbitrary filenames (freely configurable)

◇ Possibility to upload a `.php` webshell (e.g., `shell.php`)

◇ Upload path not yet verified → **Next step**

🧨 C. **SQL Injection Indicators**

◇ In `searchquery` parameter → SQL error message with incorrect syntax

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

◇ Response includes SQL error on line 207 in `badstore.cgi`

◇ **(not yet exploited, but confirmed)**

🧪 D. **Extracted Account Data**

◇ File: `/supplier/accounts` publicly accessible

◇ Content: Base64-encoded entries

◇ Successfully decoded: `joeuser`, `janeuser`, `kbookout` with passwords and IPs

privilege escalation

privilege escalation
lineup privelege escalation

3. Privilege Escalation

Status: `n.a.`
No **root or system shell** obtained yet.
→ Privilege escalation could possibly be achieved via:

- **Command injection**
- **Webshell execution**
- **LFI (Local File Inclusion)**

…to be tested in the next phase.

post-exploration

post-exploration
lineup post-exploration

🧾 4. Post-Exploration

Item	Status
User accounts	✅ successfully decoded (/supplier/accounts)
Shell upload path	❓ not yet verified
Session handling	Cookies can be saved, but no role escalation observed
Admin panel access	Not possible via ?action=admin without higher privileges
Manual field manipulation	No effect (e.g., role=admin)