Hacking BadStore.net

A hands-on approach to web application vulnerability discovery and exploitation

Welcome to BadStore.net - The most insecure store on the 'Net!

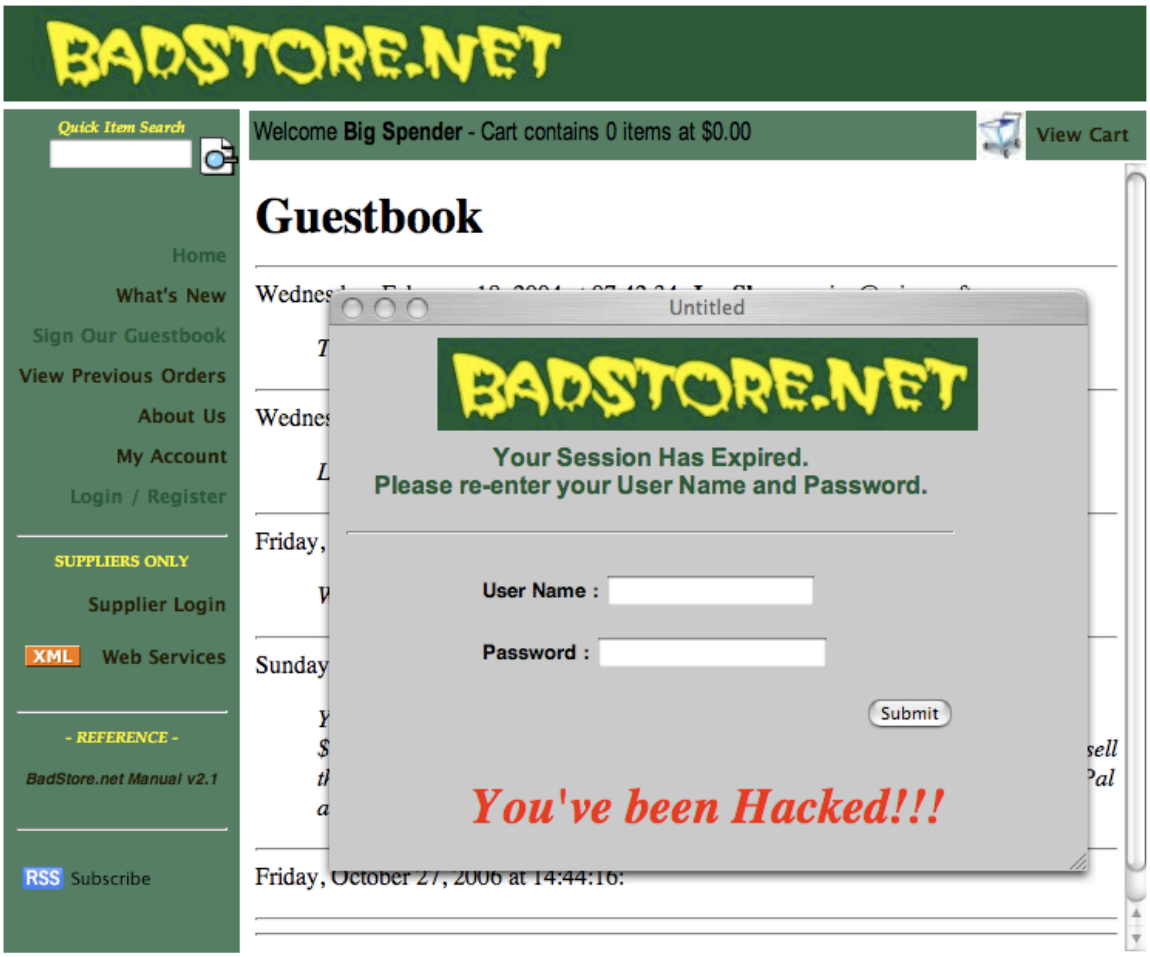

Table of Contents

WHAT IS BADSTORE.NET?

BadStore.net presents a typical three-tier web storefront application. This self-contained application was built from the ground up with typical security mistakes to serve as a platform for demonstration, security training, evaluation, and testing purposes.

The BadStore.net application is delivered as a bootable application server image. The image runs the Trinux operating system, Apache web server, a CGI (Common Gateway Interface) application, and full MySQL interaction with multiple database tables. This architecture is commonly known as LAMP (Linux, Apache, MySQL, Perl), and presents a real application environment that uses real coding methods. Rather than being a simulation, BadStore.net operates in the same way as many commercial websites, albeit with a high concentration of application security vulnerabilities.

To run the BadStore.net application, simply boot the BadStore.net CD or image in a suitable host machine. Optionally, BadStore.net can be used under a virtual environment, such as VMWare. More details about the environment are found in subsequent sections.

After boot, BadStore.net acts as a network-accessable server that clients may interact with using a Web browser. This educational playground exists for you to break. And, best of all, when you reboot, everything is back to where you started! There's no need to rebuild after successful "hacks" screw everything up. So get out your browser and start enjoying the world of web application security.

BadStore.net is currently available in English and Japanese language versions and was released under the terms of the GNU General Public License.

WHERE TO OBTAIN BADSTORE.NET

The current version of BadStore.net can be downloaded from the appropriate links on the http://www.badstore.net site.

BadStore.net will exist as in ISO image that can be downloaded and burned to CD or run from your favorite vitual machine. Adobe Acrobat pdf files contain this manual and images for the CD labels.

THE PURPOSE OF BADSTORE.NET?

Many information security professionals and organizations have never ″seen″ the real impact of application security vulnerabilities.

BadStore.net illustrates the common vulnerabilities present in many applications exposed to Intranets, Extranets, and the Internet. By allowing application security students and instructors to demonstrate vulnerabilities, attacks and their associated potential business impacts, participants will better understand the threats and how to avoid them. In this way, BadStore.net assists with security awareness, vulnerability discovery, security training, and security testing.

VULNERABILITIES PRESENTED IN BADSTORE.NET

BadStore.net deliberately contains the following security vulnerabilities:

- Input Validation Attacks, including Cross Site Scripting (XSS) and SQL Injection
- Denial of Service Attacks, including Buffer Overflow and Application DoS
- Session-based Attacks, including Cookie Poisoning, Parameter/Form Tampering
- Directory Traversal/Forceful Browsing and Command Injection
- Information Disclosure, including Cookie Snooping and Error Message Interception
- Information Manipulation, including Log Tampering
- JavaScript Validation Bypass, AJAX-introduced Automation, and Dynamic RSS
- XML Web Services Attacks

... and more!

All of these lead to the total ability to “0wn” the application, including the web server, SQL databases, application logic, operating system, and the sensitive data they "protect". Just like a real app... ;-)

UPDATES AND ENHANCEMENT REQUESTS TO BADSTORE.NET

BadStore.net will be periodically updated to introduce new functionality and more bugs! Information on the most current version of BadStore.net can be found at

http://www.badstore.net

To submit an enhancement request to BadStore.net, send an email to: kurt_roemer@yahoo.com with the subject "BadStore.net Enhancement Request" and an explanation of what you'd like to see and why you feel it would be particularly useful. Enhancement Requests for technical aspects of the system, usability, and documentation are welcome!

PROJECT TEAM AND CREDITS

The following individuals and organizations have been instrumental in delivering BadStore.net:

Ryan Barnett, Stefan Drege, Masaaki Futagi, Deral Heiland, Paul Rice, Hirofumi Teragawa, Citrix Systems, NetContinuum, and the Web Application Security Consortium (WASC) are all valued contributors to this project. BadStore.net was conceived, developed, and is maintained by Kurt Roemer.

INSTALLATION OF BADSTORE.NET

BadStore.net boots from image or CD-ROM and runs as a Linux/Apache server. There is no installation necessary, and nothing is copied to the hard drive of the PC. Please note, however, that vulnerabilities in BadStore.net would allow an attacker to access the hard drive on the host (server) PC. It is highly recommended that BadStore.net only be used in non-production environments (see the Disclaimer for more information).

BadStore.net also runs well under virtual environments, including Q on the MacIntosh, QEMU under Linux, and the free VMWare Player on Windows platforms.

Once the BadStore.net application server has been booted, add an entry to the local 'hosts' file on the client and go to:

http://www.badstore.net/

or, if JavaScript support is unavailable in the browser (you won't get far, though):

http://www.badstore.net/cgi-bin/badstore.cgi

without a DNS name assignment or 'hosts' entry (this will also be problematic):

http://serveripaddress/cgi-bin/badstore.cgi

Hosts files are /etc/hosts on *nix platforms and typically the following on Windows systems:

```
C:\Windows\System32\drivers\etc\hosts
```

System Requirements for BadStore.net

BadStore.net is intended to be run as a client/server system. The BadStore.net CD is booted in the designated server system, and a client system with a browser is used to access the BadStore.net application over a network.

The following are system requirements:

- A Personal Computer to run as the server. The tested minimum is a Pentium w/ 128MB RAM
- A CD-ROM/DVD or compatible drive on a PC configured to boot from CD, or appropriately configured virtualization software
- A supported Network Adapter for the BadStore.net server (not an issue with virtualization software)
- A network to connect the BadStore.net server to the client or an Ethernet crossover cable (not required with virtualization software, but a loopback plug may be necessary)
- A suitable client system with network adapter and browser
- Cookies enabled in the client browser
- JavaScript support enabled in the client browser

Network Configuration

Due to the highly vulnerable nature of the BadStore.net application, the safest way to play with the program is via a private network. Use a Cross-Over Ethernet cable between the client and the BadStore.net server and do not connect either system to any other network. Further containment within a virtual environment will further contain BadStore.net.

The BadStore.net server attempts to boot and assign an IP address via DHCP. If you do not have a DHCP server available, BadStore.net will come up without an IP address assignment for your Ethernet adapter. Use ifconfig to assign an address, as follows:

Example: *(To assign an address of 10.10.100.52 on a Class-C (/24) subnet)*

```
ifconfig eth0 up 10.10.100.52 netmask 255.255.255.0 broadcast 10.10.100.255
```

For a list of supported Ethernet adapters, see the Trinux documentation at:

http://trinux.sourceforge.net/network.html

or use the adapter support inherent in your favorite virtualization software.

Browser Compatibility

With the AJAX and CSS updates to BadStore.net, browser compatibility has become an issue. The dynamic screen content updates through JavaScript just don't work consistently in all browsers. YMMV.

Internet Explorer v7 also requires that *Native XMLHTTP Support* is disabled for AJAX functionality to work in BadStore.net. This can be disabled in IE7 through *Tools/Internet Options/Advanced* - uncheck the *Native XMLHTTP Support* box.

SUPPORT

There is no additional installation support or any general support of any kind for BadStore.net.

DISCLAIMER

This section explains important considerations for the use of BadStore.net.

Important Disclaimer: *No Lifeguard On Duty! - Surf at your own risk!*

BadStore.net has been developed to illustrate the common vulnerabilities present in many applications exposed to Intranets, Extranets, and the Internet. As such, the BadStore.net application platform contains dangerous vulnerabilities that expose the application and environment to attack.

BadStore.net should only be used in a lab or test environment, and must never be installed on a production system. You have been warned!

This site has been developed using common HTML, CGI(PERL), AJAX, and JavaScript coding techniques. Any similarity to an existing free or commercial application is purely coincidental. All images utilized are believed to be in the public domain or are used in a satirical context. There is no implied warranty for any use of this application.

DEMONSTRATING THE THREATS IN BADSTORE.NET

Purpose

The demonstration of Web Application attacks has proven to be a powerful tool to increase understanding for application owners, developers, network /security administrators, and organizational management. Many have not seen the actual attacks that are openly discussed in industry publications and seminars, and we often need to see the damaging effects of these attacks to truly understand and combat the threat.

BadStore.net is a web application designed to illustrate and demonstrate Web Application threats. Using BadStore.net, one can demonstrate business issues common to application platforms and illustrate common security vulnerabilities present in applications. Many of these attacks are "blended attacks" and combine several techniques to produce an exploit.

BadStore.net is a real application – not a simulated environment. BadStore.net uses many of the technologies present in applications, including CGI, SQL-based database calls, AJAX, RSS, and the Apache web platform.

BadStore.net is not intended to be the complete implementation of on on-line store or eCommerce environment. It contains key elements to illustrate security vulnerabilities and attack techniques, but does not contain store elements such as a payment gateway. This is a safety feature to prevent users of BadStore.net from doing stupid things.

This document explains the primary layout and workflow of BadStore.net, the vulnerabilities exposed throughout the applications, and how to demonstrate associated web application attacks.

Note: This document is not a cookbook that will give you the exact steps to conduct specific attacks against BadStore.net! You are strongly encouraged to discover and exploit the vulnerabilities without using the hints. Homework and formulating your own attack plan are required: Try, fail (or maybe get lucky!), try again - learn. Enjoy! ;-)

Interesting components of BadStore include:

- A full eCommerce web application running on an Apache web server
- A full SQL database for inventory, user management, and cart management
- A guestbook with a vulnerable flat-file database
- Robots.txt with path disclosure
- Backup and administrative directories, extra and old CGI files
- A "hidden" administrative portal
- An overly open httpd.conf file, bad directory permissions and symbolic links

- Helpful comments which ad troubleshooting – and attacks
- Web 2.0 technologies, including AJAX, RSS, and XML Web Services

System Configuration for a Demonstration Environment

1. Turn off any personal firewall
2. Turn off any special browser security settings and clear the browser cache (reset security and cookie/privacy features to default for demo, ensure pop-ups aren't blocked)
3. Assign an IP address to the browser's workstation
4. Boot the BadStore.net application server
5. Assign an IP address to the BadStore application server
6. Add an entry for www.badstore.net in the client's hosts file
7. Using your browser, go to http://www.badstore.net and verify proper browser operation
8. Go to View/Text Size and select an appropriate large text size for the browser
9. Set your browser's default start page to this address to make it easier for you to return to it during the demo
10. *If you mess up the databases beyond repair, either reboot or run initdbs.cgi from the server command line or from within the browser.*

Yet Another Disclaimer and Reminder

The demonstration materials presented below illustrate several web application attack techniques – the same techniques that would be used by a hacker. These techniques are intended to be used only in an offline environment while disconnected from all organizational network and application resources. Remember that testing the security of any system should only be performed with the express written permission of the application's owner. (I'm not an attorney – consult your attorney or General Counsel with any questions regarding the legal use of these attacks against organizational systems). The use of these attacks can break applications – try them at your own risk!

Acknowledging The WASC Threat Classification Project

To present the vulnerabilities present in BadStore.net in an organized manner, we will use the format of the Web Application Security Consortium Threat Classification v1.0: http://www.webappsec.org/projects/threat

Legend: When you see the Magic Rabbit, read on for helpful hints:

Hint:
- *Read these if you want a hint at where a vulnerability exists or how a simple example of the vulnerability may be exploited.*

1 Authentication

Look at the areas in BadStore.net that require authentication. Most just submit HTTP Basic credentials and submit these via insecure means.

1.1 Brute Force

A Brute Force attack is an automated process of trial and error used to guess a person's username, password, credit-card number or cryptographic key.

Hint:
- *Brute force access of accounts can easily be accomplished through Brutus or other similar tools.*
- *It really shouldn't take you more than a handful of attempts to guess the admin password! Take a look at the Guestbook for valid account names.*
- *In Login/Register, try to guess passwords.*
- *Reverse Brute Forcing is also possible.*

1.2 Insufficient Authentication

Insufficient Authentication occurs when a web site permits an attacker to access sensitive content or functionality without having to properly authenticate.

Hint:
- *Find the secret 'admin' portal!*
- *Check out the information presented for the 'new and cool' xml web services functionality. Manipulate the 'action=' parameter in the URL bar to read 'action=admin' and see what kind of mischief you can get into.*
- *Steal the SSOid cookie and become another user without knowing authentication credentials.*
- *View all the 'helpful' information presented as part of the XML Web Services functionality that should require authentication.*
- *Some of the additional directories presented (look at robots.txt for hints) should also require authentication.*

1.3 Weak Password Recovery Validation

Weak Password Recovery Validation is when a web site permits an attacker to illegally obtain, change or recover another user's password.

Hint:
- Password recovery is accomplished via a simple question: What is your favorite color?
- Go find a valid userid and guess away
- The Guestbook contains a few valid user IDs - try to guess their favorite color. If you guess correctly, the password is both immediately displayed (and is a consistent default password).

2 Authorization

Easily gain access to information that should require much stronger authorization.

2.1 Credential/Session Prediction

Credential/Session Prediction is a method of hijacking or impersonating a web site user.

Hint:
- Credentials are weakly encoded via Base64 in an SSOID cookie and passwords are simply MD5 hashed without a salt. Session IDs are time-based and very predictable.
- Login without knowing a valid password. Try basic SQL Injection.
- Easily predict a valid user's favorite color and recover their password.

2.2 Insufficient Authorization

Insufficient Authorization is when a web site permits access to sensitive content or functionality that should require increased access control restrictions.

Hint:
- View Previous Orders does not require authentication, and displays credit card information. It's rather trivial to view the orders of another user.
- Manipulate the session cookie. Using the proxy, change the "U" to an "A" as you create a new account (you saw that hidden parameter while you were profiling the application, didn't you? ;-). Go find the secret admin portal…

2.3 Insufficient Session Expiration

Insufficient Session Expiration is when a web site permits an attacker to reuse old session credentials or session IDs for authorization.

Hint:
- No Session Expiry allows you to use the browser's back button and access valid session information that should have required re-authentication - especially if an admin had just logged in from the same browser.

2.4 Session Fixation

Session Fixation is an attack technique that forces a user's session ID to an explicit value.

Hint:
- coming soon!

3 Client-side Attacks

The possibility for client-side attacks in BadStore.net are only limited by your imagination... ;-)

3.1 Content Spoofing

Content Spoofing is an attack technique used to trick a user into believing that certain content appearing on a web site is legitimate and not from an external source.

Hint:
- Use XSS to place content on the page and generate popup boxes for Phishing attacks,
- The Search integration into RSS can be used to inject content for all that subscribe to the RSS feed.
- Easily impersonate other users and post comments to the Guestbook.

3.2 Cross-site Scripting (XSS)

Cross-site Scripting (XSS) is an attack technique that forces a web site to echo attacker-supplied executable code, which loads in a user's browser. Cross Site Scripting, abbreviated as XSS (not CSS, which would be confused with Cascading Style Sheets) is not a traditional web application side attack. An XSS attack is an attack against the clients that use an organization's web site.

Hint:

- XSS can be used as a test to pop up an alert box (with the supplied message) in the browser, and is also commonly used to print the user's cookie, possibly containing session information and secret account credentials.
- Exploit an XSS attack in the "Sign Our Guestbook" screen. Enter the following as separate comments:

```
<script>alert('This is an XSS attack')</script>

<script>alert(document.cookie)</script>

<script>print(document.cookie)</script>
```

Note that with the print command, we were able to control an action from the browser directly - what else can the browser do or have access to?

Phishing attacks (enticing users to do something stupid by making it appear legitimate) is often carried out through an XSS attack.

Hint:

- As a good example of a Phishing attack using an XSS vector, copy and paste the following script into the comments in the Guestbook:

```
<script type="text/javascript">myOtherWindow = open("", "secondWindow", "width=425,
height=275, left=300, top=275");if (myOtherWindow != null);{var otherWindowDefinition =
'<html><head><meta http-equiv="content-type" content="text/html;charset=iso-8859-
1"><title>Session Expiry<\/title><\/head><body bgcolor="silver"><Center><img
src="http://www.badstore.net/images/BadStore.jpg"></Center><table width="400"
height="200"><tr align="center"><td width="375" height="40"><font size="3" color="004b2c"
face="Arial"><b>Your Session Has Expired.<br>Please re-enter your User Name and
Password.<\/b><\/font><\/td><\/tr><tr><td><hr><\/td><\/tr><tr align="center"><td><font size="2"
face="Helvetica"><b>User Name : <\/b><\/font><input type="text" name="user_id"
size="18"><\/td><\/tr><tr align="center"><td><font size="2" face="Helveti-
ca"><b>  Password  : <\/b><\/font><input type="password" name="password"
size="20"><\/td><\/tr><tr><td align="middle"><INPUT TYPE="button" align="right" NAME="Sen-
dIt" VALUE="Submit" onClick="alert(\'User= \'+user_id.value+\' & Pass= \'+password.value+\' &
CookieInfo= \'+document.cookie+\'      Sent to Hackersite.com
\');self.close();"><\/td><\/tr><\/table><\/body><\/html>';myOtherWindow.document.write(otherWin
dowDefinition);}</script>
```

Of course, these are simple examples. A truly dangerous script could have emailed or otherwise transferred private credentials to a remote attacker. Using active content (JavaScript, ActiveX, or Flash) to control remote browsers is very dangerous, and attack automation via AJAX makes XSS interactive – what if a malicious script was injected where an administrator would run it?

4 Command Execution

Running commands allows bypass of application controls and often allows direct access to the operating environment.

4.1 Buffer Overflow

Buffer Overflow exploits are attacks that alter the flow of an application by overwriting parts of memory.

Hint:
- Try a credit card number that is REALLY long and watch the application stop responding.
- Use a proxy to modify the CardID cookie and submit a really long credit card number.

4.2 Format String Attack

Format String Attacks alter the flow of an application by using string formatting library features to access other memory space.

Hint:
- Format strings are used to provide a $x.yy view of prices in BadStore.net.
- Try a format string attack as full name, email, or password.

4.3 LDAP Injection

LDAP Injection is an attack technique used to exploit web sites that construct LDAP statements from user-supplied input.

Hint:
- coming soon!

4.4 OS Commanding

OS Commanding is an attack technique used to exploit web sites by executing Operating System commands through manipulation of application input.

Hint:
- Try to view files and run command from within View Pricing File under Supplier Login.

4.5 SQL Injection

SQL Injection is an attack technique used to exploit web sites that construct SQL statements from user-supplied input.

Hint:
- A SQL Injection attack attempts to to directly obtain or manipulate data in the database. First, we want to dump the contents of the product database. Then we'll go further…
- Go to the "Quick Item Search" area and type in a single quote " ' ", otherwise known as a tick mark (also commonly represented in the world of URL encoding at %27). Let's see if this is susceptible to SQL Injection. Note the helpful display of the SQL statement! This helps to show what SQL looks like, and what it's doing behind the scenes.
- To test for SQL Injection, we will use the most common technique. Go back to the Search input box and type: ' OR 1=1
  The ' (tick mark) character is used to delineate variables in a SQL query, and 1=1 always evaluates as True. So, what we're doing in this simple example is telling the web application to evaluate our request as valid, potentially bypassing application security controls.
- Notice that you received the same "Mismatched single quote" error. Try to match up the quotes. Type the following: 1001' OR 1=1 OR '1002
  Were you able to dump the entire contents of the product database, including test items?

There are many ways to conduct testing of SQL Injection vulnerabilities which bypass all application controls and go directly for the data in the databases. Notice that SQL injection can happen through the URL, through hidden form fields, and through search and query functions. Additionally, user credentials can be stored and verified in a SQL database.

Hint:
- See if BadStore.net does stores and uses credentials from a SQL database.
- Go to the Supplier Login screen and enter in a tick mark for the email address.
- Now try: john' OR 1=1 OR 'mary in the email box. If this works, you don't even need to enter a password.

To prevent SQL Injection, you need a solution that completely validates user input.

4.6 SSI Injection

SSI Injection (Server-side Include) is a server-side exploit technique that allows an attacker to send code into a web application, which will later be executed locally by the web server.

Hint:
- coming soon!

4.7 XPath Injection

XPath Injection is an attack technique used to exploit web sites that construct XPath queries from user-supplied input.

Hint:
- coming soon!
- The XPath framework is already used to build a couple databases.

5 Information Disclosure

Information can be leaked through helpful hints, troubleshooting routines, common mistakes, and error messages.

5.1 Directory Indexing

Automatic directory listing/indexing is a web server function that lists all of the files within a requested directory if the normal base file is not present.

Hint:
- Take a look at robots.txt. How does this differ from what an application scanner told you existed? See whether the "excluded" directories exist.
- Also look at /images/, etc.

5.2 Information Leakage

Information Leakage is when a web site reveals sensitive data, such as developer comments or error messages, which may aid an attacker in exploiting the system.

Hint:
- Banner Grabbing is the ability to glean helpful information for an attack from headers presented by the web server and a basic form of information leakage. Perform banner grabbing by bringing up a command prompt in your OS and typing:

```
telnet www.badstore.net 80 <Enter>
HEAD / HTTP/1.0 <Enter> <Enter>
```

- *Note that the commands entered in telnet don't always echo to the screen.*
- Scroll up and show all the information presented, then try GET instead of HEAD to obtain all page information.

Source Code is often a rich source of information to an attacker and discloses company secrets and enumerates vulnerabilities.

Hint:
- Show the rich comments often embedded in source code:
- Go to the Search screen and enter an item number
- Go to view/source in the browser
- Scroll through the html - do you see anything interesting?
- *What could this be? (<!-- Search code developed by Bobby Jones - summer intern, 1996 --><!-- Comment the $sql line out after troubleshooting is done -->)*

Source Code Comments are important to the developers, but should be stripped off before the page is presented to the web user.

Hint:
- The robots.txt file is used to manage web crawlers that index the site. Web crawlers are usually the familiar search engines, such as Google, Yahoo!, and Altavista. The contents of robots.txt tells the crawler which directories to index, and which directories to avoid. Often, the directories to avoid include areas that would be interesting to an attacker, and the proliferation of internal search appliances has increased the threat vector.
- Go to http://www.badstore.net/robots.txt to display the following:

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

User-agent: badstore_webcrawler
Disallow:

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

5.3 Path Traversal

The Path Traversal attack technique forces access to files, directories, and commands that potentially reside outside the web document root directory.

Hint:
- In Supplier Portal, see if the file upload functionality can be used to place files in other directories.
- Also, see if View Pricing File allows you to see files in other directories.
- SQL Injection can be used to view any arbitrary file on the server.

5.4 Predictable Resource Location

Predictable Resource Location is an attack technique used to uncover hidden web site content and functionality.

Hint:
- Take a look at robots.txt for directories that might exist outside of the application environment (http://www.badstore.net/robots.txt).
- After viewing several of the action= parameters in the URL bar, what other actions do you suppose might be enabled for more privileged users?
- Try handcrafted action= parameters, including test and admin.
- Look for old versions of files (badstore.old?).
- See if a test application exists to quickly show if the application is available (test.cgi?).

6 Logical Attacks

6.1 Abuse of Functionality

Abuse of Functionality is an attack technique that uses a web site's own

features and functionality to consume, defraud, or circumvents access controls mechanisms.

Hint:
- Exploit Supplier Upload

6.2 Denial of Service

Denial of Service (DoS) is an attack technique with the intent of preventing a web site from serving normal user activity.

Hint:
- Exploit DoS on Apache server, AppDoS on authentication, cookie overflow, database requests through AJAX

6.3 Insufficient Anti-automation

Insufficient Anti-automation is when a web site permits an attacker to automate a process that should only be performed manually.

Hint:
- Exploit Automated vulnerability scanning. Scanbot
-

6.4 Insufficient Process Validation

Insufficient Process Validation is when a web site permits an attacker to bypass or circumvent the intended flow control of an application.

Hint:
- Exploit Cart items, cost, and authentication of new users through Cookie Tampering and Poisoning.

APPENDICES

Appendix A - BadStore.net Change Log

v1.0 – Original version for 2004 RSA Show

v1.1 – Enhancement Requests Added:

- More supported NIC's
- Referrer checking for Supplier Upload
- badstore.old in /cgi-bin/
- Minor cosmetic updates.

v1.2 – Version presented at CSI 2004. Added:

- Full implementation of MySQL
- JavaScript Redirect in index.html and JavaScript validation of a couple key fields
- My Account services, password reset and recovery
- Numerous cosmetic updates, favicon.ico
- 'Scanbot Killer" directory structure to detect scanners
- Reset files and databases to original state without reboot (initdbs.cgi)
- Dynamic dates and times in databases
- Additional attack possibilities

v2.0 – Web Services Edition. Added:

- XML Web Services through SOAP, SOA functionality
- Interaction with JAVA-based MegaSupplier.net site in a supply chain relationship
- Ability to reset databases through initdbs.cgi

v2.1 – Web 2.0 Edition. Added:

- Capabilities for SSL, AJAX, XPath, CSS, RSS, and updated DBI
- Filesystem expansion for more stuff, requiring 128MB RAM
- LUHN checkum for credit cards, eval{, and RegEx for more fields

Appendix B - License

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.
The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed

(in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients′ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.
It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and ″any later version″, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.