Pen Test Report

Foundstone Hacme Bank

Table Of Contents

0. Document Information

Author:	Dinis Cruz	Status:	Published	Issue Date:	25/April/2006
Confidentiality Rating:		Public		Section	Owasp Pen Test
Approved By:		Dinis Cruz		Signed:	

0.1 Review Details

Version	Reviewed	Date	Detail
v1.0	Dinis Cruz	25/April/2006	

0.2 Revision History

Version	Author	Date	Detail
v1.0	Dinis Cruz	25/April/2006	

0.3 Related documents

Version	Author	Date	Detail
2004	Owasp		Owasp Top 10 Web Application Vulnerabilities

1.Introduction

1.1. Executive summary

The PenTest of the Hacme Bank IPoP identified a number of security issues that allows malicious anonymous attackers to gain complete control over the Hacme Bank database. It was also discovered that normal users are able to access confidencial data belonging to other users and to gain access to administrative interfaces.

Section 3 of this document identifies the vulnerabilities discovered and section 4 contains recommendation to resolve them

1.2. Criticality rating of overall test results

Critical	Serious security issues have been identified that result in disclosure of vulnerability information. Appropriate action should be scheduled as soon as possible.

1.3. Timeline and project scope

The tests where performed during normal business hours (GMT) over the period of 20 April 2005 till 24 April 2006 on the following subnets/servers:

- 127.0.0.1
- Localhost
- a.b.c.d.

1.4. Document sensitivity

The contents of this document are highly sensitive, and appropriate controls must be used when using, storing and transmitting this document.

Strong encryption should be used when storing and transmitting this document.

2. Scanned Hosts

2.1 All Hosts
The following hosts were scanned as part of this assessment:

- 127.0.0.1 (hacmebank)

2.2 Hosts without Findings
None of the scanned hosts were found to be free of security issues.

Origin:	PenTest Report	Title:	Foundstone Hacme Bank

3. Findings

3.1 127.0.0.1 (hacmebank)

ID	Layer	Category	Finding	Affected Item	Comment / Solution	Impact	Probability
123-0	Application	Input and Data Validation	Account Transfer validation for negative values is only performed at the client	• http://a.b.c.d/aspx/main.aspx?function=AccountTransfe	Use a proxy (or a browser tamper plugin) to inject a negative number in the Form http://a.b.c.d/aspx/main.aspx?function =AccountTransfer (this will transfer an amount TO the source account FROM the target account (i.e. the opposite of expected behavior)	Medium	Medium
123-1	Application	Input and Data Validation	Maximum number of login attempts is controlled by client-side cookie	• CookieLoginAttempts	Use a proxy (or a browser tamper plugin) to change the value of the CookieLoginAttempts (for example to 5000) • SW3.3 - Sensitive Data / Directory Indexes	Low	Medium
123-2	Application	Insecure Configuration	Directory Listing Enabled	• http://a.b.c.d/install/		Low	Low
123-3	Application	Access Control	Admin pages available to normal users	• http://a.b.c.d/aspx/Main.aspx?function=admin\Fetch_Web_Page • http://a.b.c.d/aspx/Main.aspxfunction=admin\Manage_Accounts • http://a.b.c.d/aspx/Main.aspxfunction=admin\Manage_Messages • http://a.b.c.d/aspx/Main.aspx?function=admin\Manage_Users • http://a.b.c.d/aspx/Main.aspx?function=admin\Sql_Query • http://a.b.c.d/aspx/Main.aspx?function=admin\Web_Services	After login, a normal user is able to access the affected admin pages.	Critical	Medium
123-4	Application	Access Control	Users are able to access account details belonging to other users	• http://a.b.c.d./aspx/Main.aspx?function =TransactionDetails&account_no=5204320422040001 • http://a.b.c.d./aspx/Main.aspx?function =TransactionDetails&account_no=5204320422040003	To test this vulnerability, Log-in as user jv and open the page http://a.b.c.d ./aspx/Main.aspx?function=TransactionDet ails&account_no=5204320422040001 . Then replace the 'account_no' GET value with 5204320422040003 (i.e. http://a.b.c.d./ aspx/Main.aspx?function=TransactionDetai ls&account_no=5204320422040003) and note that you are now accessing account details belonging to another user (in thiscase the user jm) • SW4.3 - Access Control / Session Management	Medium	Medium
123-5	Application	Access Control	Old Password requirement is not enforced in 'Change	• http://a.b.c.d./aspx/main.aspx?function=PasswordChange	To view this vulnerability follow thesesteps: Hijack user session (using for example a valid	Medium	Low

			Password' page		user's Session Cookie), open the page http://a.b.c.d./aspx/main .aspx?function=PasswordChange and change that user's password (without knowledge of that user's current password)		
123-6	Application	Authentication	Session Hijacking via ASP.NET_Session cookie	• User Session	Discover a valid ASP.NET_Session cookie, and hijack that account by changing the cookie on the browser or injecting it via a proxy	Medium	Medium
123-7	Application	Authentication	Admin site protected with weak cookie	• 'admin' cookie • http://a.b.c.d/aspx/main.aspx?function=AdminSection	Access to the admin site is controlled by a client side cookie called 'admin' (On login, this value is false, and set to true after successful Response to the Challenge posted here http://a.b.c.d/asp x/main.aspx?function=AdminSection). To a ccess the admin area, login as a normal user and change the value of the 'admin'cookie from false to true	High	Medium
123-8	Application	Authentication	WebServices are accessible by anonymous users:	• http://a.b.c.d/HacmeBank_V2_WS/WebServi ces/AccountManagement.asmx • http://a.b.c.d/HacmeBank_V2_WS/WebServi ces/UserManagement.asmx • http://a.b.c.d/HacmeBank_V2_WS/WebServi ces/UsersCommunity.asmx	Web Services pages can be directly accessed and invoked.	Medium	Low
123-9	Application	Authentication	ViewState replay vulnerability	• http://a.b.c.d./aspx/main.aspx?function=AccountTransfer	The source account on the Transfer Funds page (http://a.b.c.d./aspx/main.aspx? function=AccountTransfer) is controlledby ViewState. This means that the attacker cannot change this value by POST form injection, but means that if the attacker is able to grab a valid ViewState from another user (via Xss, cached copy of that page on a Hard Disk or by sniffing the traffic), it can replay that ViewState and make transfers from that account (to an external account).	Medium	Low
123-10	Application	Authorization	Web Services Session ID is not enforced	• http://a.b.c.d/HacmeBank_V2_WS/WebServi ces/AccountManagement.asmx • http://a.b.c.d/HacmeBank_V2_WS/WebServi ces/UserManagement.asmx • http://a.b.c.d/HacmeBank_V2_WS/WebServi ces/UsersCommunity.asmx	Invoke the web services directly without needing a valid SessionID • The correct resolution of this vulnerability is one where the Web Services are still publicly available but control to the exposed Web Services functionality is managed via the SessionID	High	Medium
123-11	Application	Input and Data Validation	Cross site Scripting (XSS)	• Account Transfer 'Comment': field http ://a.b.c.d/aspx/main.aspx?function=AccountTransfer • Request a Loan' 'Comment' field: http:/	It is possible to Insert XSS payloads on the 'Affected items' pages • SW1.3 - Parameter Manipulation / Cross	Medium	Medium

				/a.b.c.d/aspx/main.aspx?function=Loan • Post Message 'Subject' or 'Text' fields: http://a.b.c.d/aspx/main.aspx?function=PostMessageForm	Site Scripting		
123-12	Application	Parameter Manipulation	SQL Injection	• Login Page 'Username' or 'Password' fields: http://a.b.c.d/aspx/main.aspx?function=PostMessageForm • Transaction Details 'account_no' GET field: http://a.b.c.d/aspx/Main.aspx?function=TransactionDetails&account_no=5204320422040001 • Account Transfer 'Comment': field http://a.b.c.d/aspx/main.aspx?function=AccountTransfer • Request a Loan' 'Comment' field: http://a.b.c.d/aspx/main.aspx?function=Loan • Post Message 'Subject' or 'Text' fields: http://a.b.c.d/aspx/main.aspx?function=PostMessageForm	It is possible to Insert SQL Injection payloads on the 'Affected items' pages • SW1.1 - Parameter Manipulation / SQL Injection	Critical	High
123-13	Application	Exception Management	Detailed error messages sent to client	• Login Page 'Username' or 'Password' fields: http://a.b.c.d/aspx/main.aspx?function=PostMessageForm • Account Transfer 'Comment': field http://a.b.c.d/aspx/main.aspx?function=AccountTransfer • Request a Loan' 'Comment' field: http://a.b.c.d/aspx/main.aspx?function=Loan	To view the error messages Force SQL errors on the 'Affected Items' pages • SW2 - Exception Management	Medium	Medium
123-14	Service	Information Leak	'WAF redirect on attack detection' information leak		The normal WAF functionality of redirecting attacks detected to a custom errorpages, provides information to attackers that such type of defense (WAF) is inuse, and creates very dangerous 'False Positive' situations where valid user'sinput could be wrongly flagged - something that would severely affect the userexperience and business value (imaginea user filling a 4 page web form being redirected to the error page on the last page). • Dynamically 'normalize' potentially malicious input. For example, on a Form field vulnerable to SQL Injection, rewrite that field with only the allowed chars (for example letters and numbers) and flag an attack	Medium	Medium
123-15	Application	Sensitive Data	Asp.Net ViewState contains Challenge's Response	• ViewState	To view the Challenge's Response in clear text, follow these steps: 1) Decode the ViewState from the Admin Section login page (http://a.b.c.d/aspx/main.aspx?function=AdminSection), 2) discover the Ch	High	Low

					allenge's Response value in the decoded ViewState, and 3) use that value to login to the admin area (the Challenge's Response value is stored in a Asp.net control which is marked with 'visible=false' (but still stored in the ViewState))		
123-16	Application	Sensitive Data	Challenge's Response weak encryption	• System used to encrypt the Challenge's Response	It is possible to Brute force the Challenge's Response since it is calculated by XORing the Challenge against a simple number	High	Low

4. Recommendations

ID	Layer	Category	Solution
SW1.1	Service - Web Application	Parameter Manipulation / SQL Injection	Correctly validate user input before using it in database queries. Use stored procedures wherever possible queries. Do not use string concatenation to build SQL queries. Input that is to be used in SQL queries should be carefully vetted for the presence of SQL control characters.
SW1.3	Service - Web Application	Parameter Manipulation / Cross Site Scripting	All content to be displayed back to a user should be passed through escape functions, to escape any HTML/XML characters that may be embedded within that content.
SW2	Service - Web Application	Exception Management	Custom error messages should be created in such a way that they do not reveal any information and the web server should be configure to not send detailed error messages to clients. Do not send error messages to non-admin clients. Handle all errors in a consistent and non-informative way.
SW3.3	Service - Web Application	Sensitive Data / Directory Indexes	Directory indexes should always be disable on all directories accessible via a web server
SW4.3	Service - Web Application	Access Control / Session Management	Session management should be performed using an authentication system, and a session cookie. Before accessing elements within a site, it should be confirmed that a user has authorisation to access that element. Users should not be able to download the data of other users.