SecureLayer7
Time and Again, Securing you

SAMPLE

APPLICATION SECURITY ASSESSMENT REPORT

DATE: 21 JAN 2017

CLASSIFICATION: CONFIDENTIAL

SecureLayer7
Time and Again, Securing you

Document Control

Item	Description
Document Title:	Vulnerability Assessment and Penetration Testing Report
Test Type	Black Box
Version No:	1.0
Status:	Final
Publish Date:	
Revision Date:	

Author(s)		
Name	***Functional Section, Department***	***Signature/Date***
	Senior Security Consultant	
Author(s)		
Name	***Functional Section, Department***	***Signature/Date***
	Security Engineer	
Approved by		
Name	***Functional Section, Department***	***Signature/Date***

SecureLayer7
Time and Again, Securing you

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

1.1 INTRODUCTION

The report presents the results of the Black-box vulnerability assessment and penetration testing conducted by SecureLayer7 for CLIENT.

The team from SecureLayer7 carried out the vulnerability assessment and penetration testing within a controlled environment and within timings as specified by CLIENT. The vulnerability assessment and penetration testing was conducted from SecureLayer7 Office. The assessment started on Start: 9th DEC, 2015 and ended on 19th DEC, 2015.

The team simulated and went through all the detailed documentation & live scenario from the point of view of an external attacker. The purpose of this assessment was to (i) Test the web application to identify technical vulnerabilities and discover whether a malicious user may leverage these flaws to compromise the security of CLIENT (ii) Provide recommendations for risk mitigation that may arise on successful exploitation of these vulnerabilities. The subsequent sections of this document provide statistics of the threat surface & vulnerabilities identified. The detailed technical findings section constitutes identified vulnerabilities with recommendations to mitigate security risks associated with the web applications.

Our security test results & findings provided in this report are valid for the period during which the assessment was carried out and is based on the information provided for the assessment. Projection of any conclusions based on our findings for future periods and web applications versions is subject to the risk that the validity of such conclusions may be altered because of changes made to the web applications or system. Furthermore, **the findings in this report reflect the conditions found during the assessment, and do not necessarily reflect current conditions.**

1.2 LIMITATIONS

The test was conducted with zero knowledge of application with valid test credentials given by CLIENT and from an external perspective. A time based approach was followed where the application was tested for vulnerabilities within a fixed time frame.

1.3 SCOPE

SecureLayer7 performed an application security test on the designated Domain under the scope covering:

S. No	URL /IP Addresses
1	https://CLIENT.corp.com/CLIENT/Default.aspx
2	https://www.AAAA.com/bbbbonline/LoginINetCLIENT.aspx?VectorIdRI=59cde347d7854183b986df1c1985ac83&

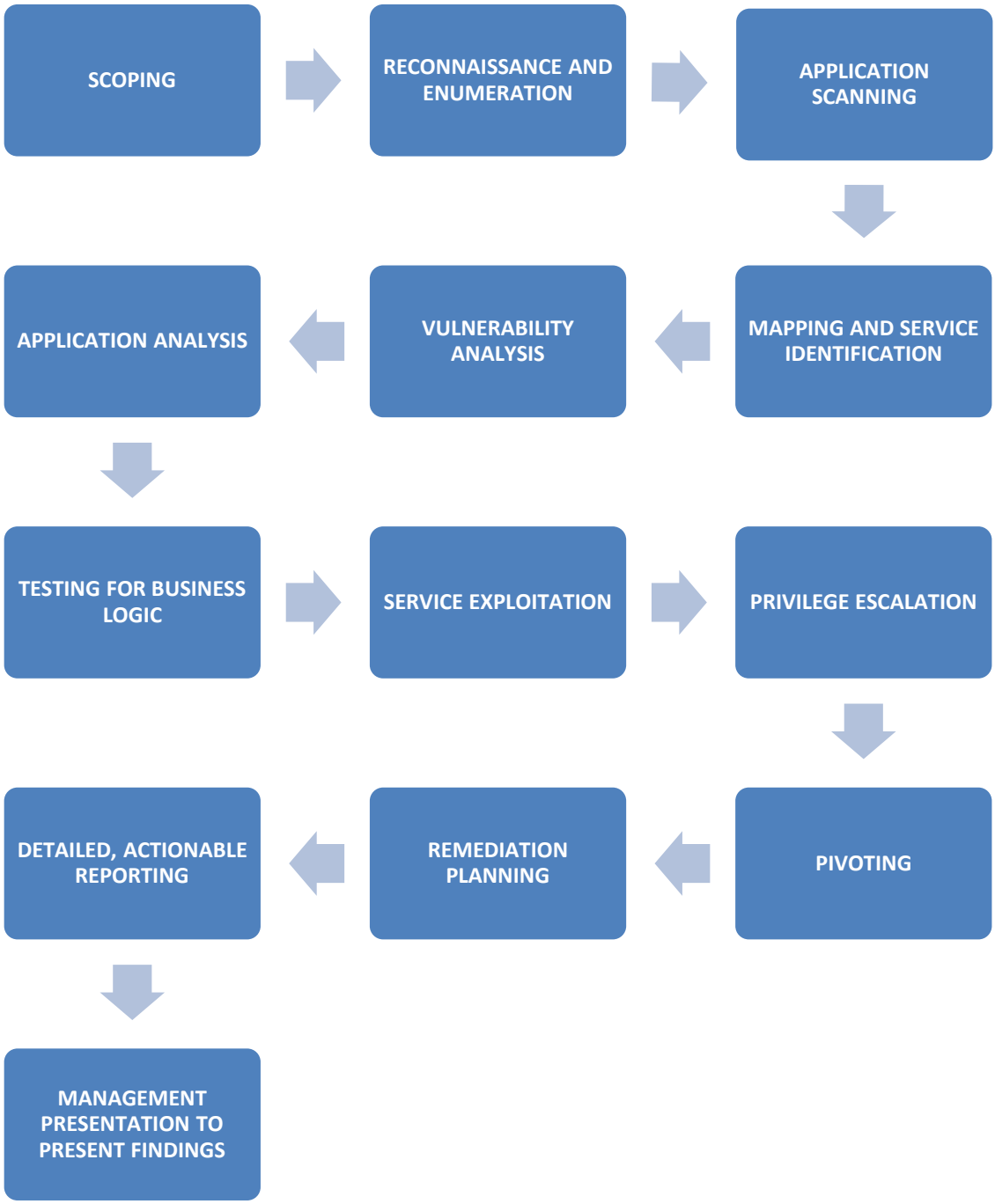
1.4 VULNERABILITY ASSESSMENT AND PENETRATION TESTING METHODOLOGY

Global standards and frameworks define the most efficient and effective methods to perform security assessments. They also differentiate and suggest assessment methodologies and processes based on the business requirements and the system architecture. Our security assessment methodology and process uses the following reliable open standards and frameworks:

1. Open Web Application Security Project (OWASP)
2. Web Application Security Consortium (WASC)

Below we have provided the chronicle for CLIENT Bank website security assessment methodology. During the time of assessment, the testing team will consider all latest vulnerabilities and security threats that were disclosed in the past 12 months & also look for any possible zero day disclosure(s).

The SecureLayer7 Vulnerability Assessments and Penetration tests (VAPTs) are scaled to meet the needs of your business. For a sophisticated website, you can choose the comprehensive, all components VAPT. However, below is the general overview of our methodology approach as follows:


SCOPING
RECONNAISSANCE AND ENUMERATION
APPLICATION SCANNING
MAPPING AND SERVICE IDENTIFICATION
VULNERABILITY ANALYSIS
APPLICATION ANALYSIS
TESTING FOR BUSINESS LOGIC
SERVICE EXPLOITATION
PRIVILEGE ESCALATION
PIVOTING
REMEDIATION PLANNING
DETAILED, ACTIONABLE REPORTING
MANAGEMENT PRESENTATION TO PRESENT FINDINGS

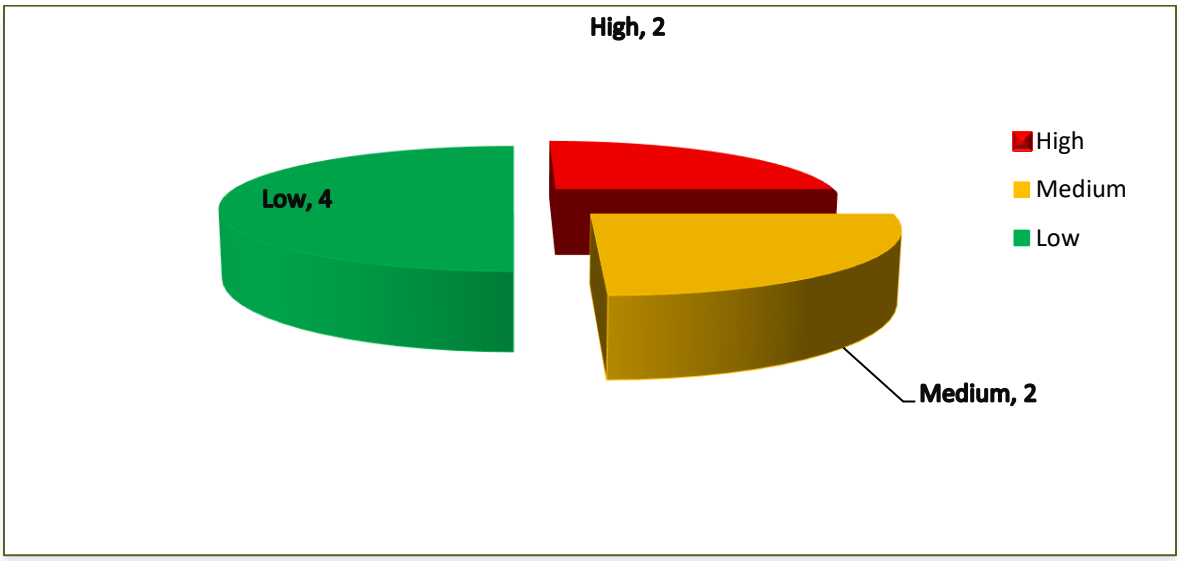
1.5 RISK CATEGORIES

The vulnerabilities have been classified into 3 categories based on the impact and ease of exploitation

Vulnerability Rating	Level of Severity
High	**Impact**: Vulnerability noted on the affected IT asset can be exploited to obtain remote privileged or unprivileged access, and cause severe impact to system operations. **Ease of Exploit**: Exploit techniques are well known. The techniques can be easily obtained and executed by unskilled attackers. The circumstances under which the attack may occur are very common.
Medium	**Impact**: Vulnerability noted on the affected IT asset can be exploited to obtain limited user privileges or network level access. **Ease of Exploit**: Exploit techniques are fairly well known. Techniques can be easily obtained and executed by persons with general computer security knowledge. The circumstances under which the attack may be successful are common.
LOW	**Impact**: Vulnerability noted on the affected IT asset provides little or no chance for exploitation. **Ease of Exploit**: Exploit techniques are not widely known. Techniques are difficult to obtain and execute, and requires detailed computer security knowledge and experience. The circumstances under which the attack may be successful are rare.

1.6 SUMMARY OF FINDINGS

The below table shows number of risk rating findings:



S. No.	Finding	Risk Rating	Finding ID
1	Arbitrary file upload vulnerability in account registration form	High	CLIENT-EXT -ID-01
2	Cross site scripting(xss) attack in twitter share link	High	CLIENT-EXT -ID-02
3	Improper session management in online banking	Medium	CLIENT-EXT -ID-03
4	Customer Id bypass from bnline banking	Medium	CLIENT-EXT -ID-04
5	Information discloser in account creation form	Low	CLIENT-EXT -ID-05
6	Information discloser in resume upload form	Low	CLIENT-EXT -ID-06
7	Missing anti-clickjacking X-Frame-Option in header	Low	CLIENT-EXT-ID-07
8	ASPX and server information getting disclosed on all pages	Low	CLIENT-EXT-ID-08

1.7 SUMMARY OF RECOMMENDATIONS

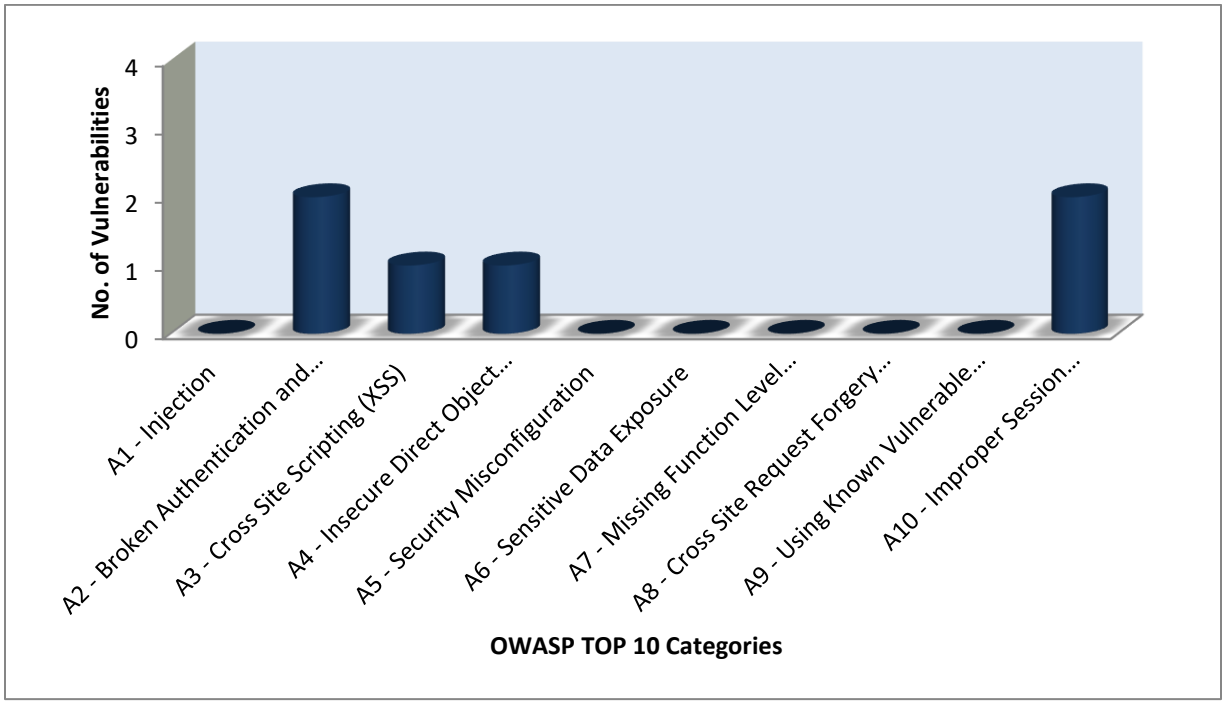
The recommendations to close these vulnerabilities are:

- Cross-site scripting vulnerability has been observed it is highly recommended to follow secure coding practices by implementing respective parsers according to the type of request input and validating services used.
- Arbitrary file upload vulnerability found it is strongly recommended to mitigate this vulnerability and detailed recommendation given in "Detailed Findings and Proof of Finding" section.
- The recommended debug option should be by default disabled in configuration file.
- Since there is no proper validation for html striping function while passing any argument by the user, therefore, filtering mechanism and various other encoding techniques are recommended to be used for proper validation of the function.
- To implement clickjacking protection, you need to add the X-Frame-Options HTTP Response header to any page that you want to protect from being clickjacked via framebusting.

1.8 BENCHMARKING WITH OWASP TOP 10

Our application security tests are modeled along the methodologies specified by the Open Web Applications Security Project (OWASP). OWASP has rated the Top Ten Vulnerabilities found in web applications worldwide.

The below chart depicts the number of discovered vulnerabilities mapped to OWASP TOP 10 vulnerabilities:



2. Detailed Findings And Proof of Findings

2.1 Arbitrary File Upload Vulnerability in Account Registration Form

Finding ID	Description
CLIENT-EXT-ID-01	During the test we found that the image upload in account registration form is vulnerable to Arbitrary file upload vulnerability. AccountInfo.aspx contains functionality to handle image upload. A remote attacker could use this functionality to upload malicious executable files on the system.
CVSS Score	**Risk Rating**
7.0 (High)	**High**
Affected Resource	
https://www.website.com/userimage/poc@securelayer7.net.aspx	
Impact/Consequence	
An attacker can exploit this vulnerability to upload arbitrary code and execute it in the context of the web server process. This may facilitate unauthorized access or other attacks are also possible.	
Recommendation	

And some special recommendations for the developers and webmasters:

- Never accept a filename and its extension directly without having a white-list filter.
- It is necessary to have a list of only permitted extensions on the web application. And, file extension can be selected from the list. For instance, it can be a “select case” syntax (in case of having VBScript) to choose the file extension in regard to the real file extension.
- All the control characters and Unicode ones should be removed from the filenames and their extensions without any exception. Also, the special characters such as “;”, “:”, “>”, “<”, “/” ,”\”, additional “.”, “*”, “%”, “$”, and so on should be discarded as well. If it is applicable and there is no need to have Unicode characters, it is highly recommended to only accept Alpha-Numeric characters and only 1 dot as an input for the file name and the extension; in which the file name and also the extension should not be empty at all (regular expression: [a-zA-Z0-9]{1,200}\.[a-zA-Z0-9]{1,10}).

Tool used	References
Burp Suite	https://www.owasp.org/index.php/Unrestricted_File_Upload

Proof of Vulnerability

To reproduce Arbitrary File Upload Vulnerability in the CLIENT bank application, please find the video where we have explained the detailed procedure of reproducing Arbitrary File Upload Vulnerability. In the given figure 1.0

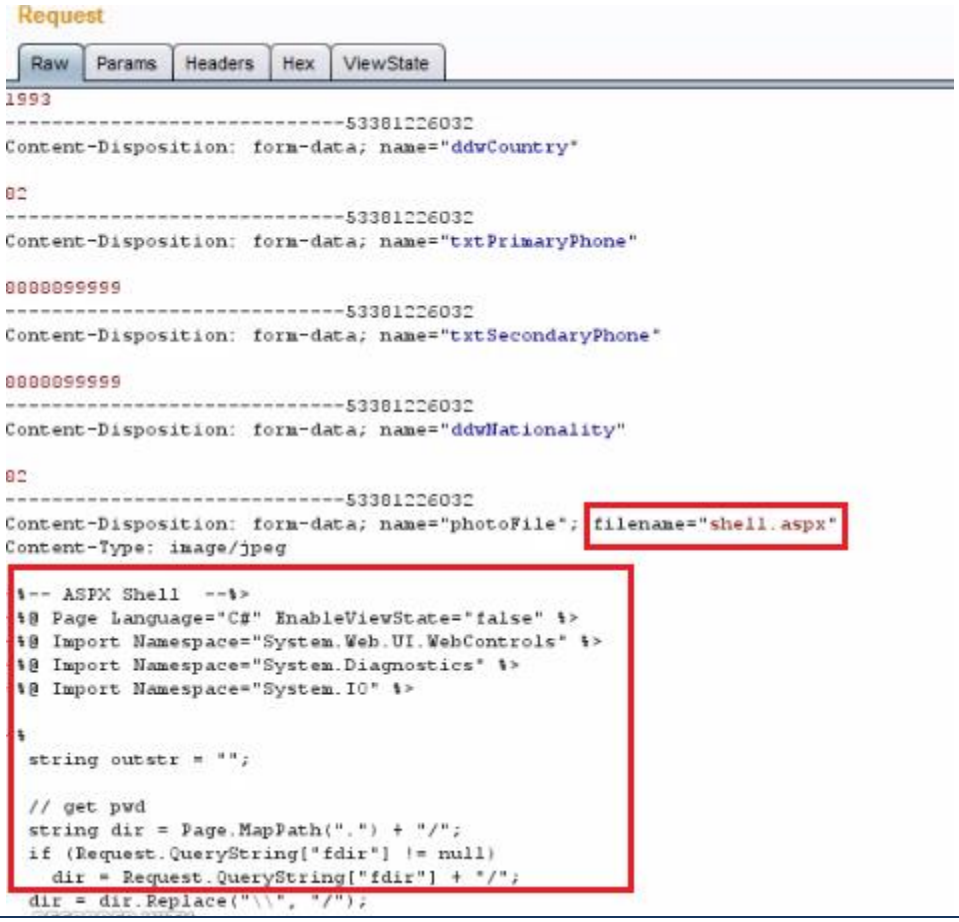


Figure 1.0 – Arbitrary File upload

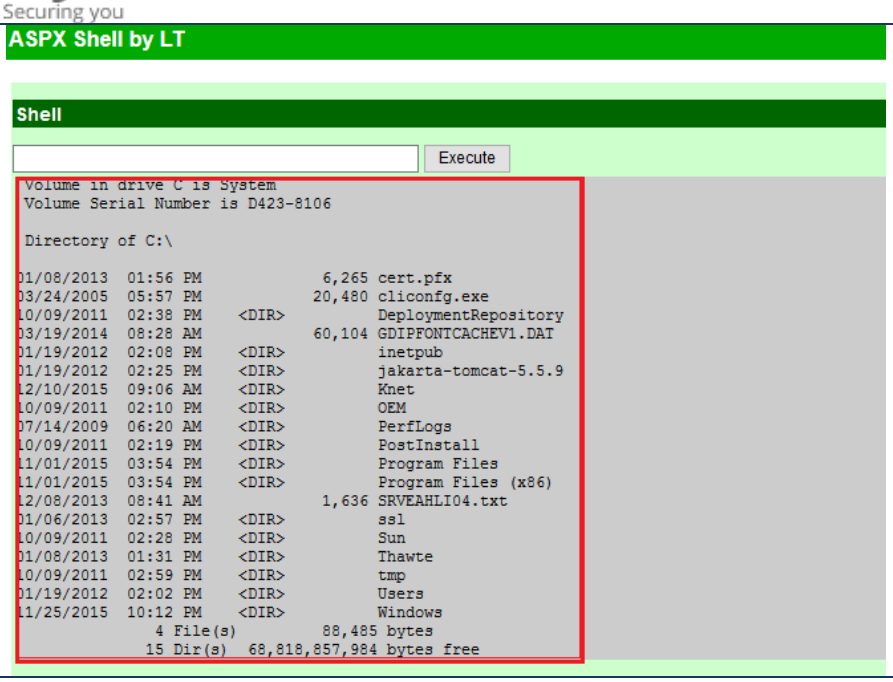


Figure – Server Compromised

2.2 CROSS SITE SCRIPTING (XSS) ATTACK IN TWITTER SHARE LINK

Finding ID	Description
CLIENT-EXT-ID-02	During the test we found that the many pages via share link field are vulnerable to XSS. Cross-site scripting is a very common injection type web application vulnerability. By exploiting an XSS vulnerability, a malicious hacker can inject malicious client-side script in a website which is executed by the victims. Typically, cross-site scripting attacks are used to bypass access controls and to impersonate users.
CVSS Score	**Risk Rating**
7.0 (High)	**High**

Affected Resource

https://CLIENT.website.com/AccountInfo.aspx?id=1?%27%20onmouseover=alert%281%29;%20%27

Impact/Consequence

By exploiting a Cross-site scripting vulnerability the attacker can hijack a logged in user's session.

This means that the malicious hacker can change the logged in user's password and invalidate the session of the victim while the hacker maintains access. As seen from the XSS example in this article, if a web application is vulnerable to cross-site scripting and the administrator's session is hijacked, the malicious hacker exploiting the vulnerability will have full admin privileges on that web application.

Recommendation

The simplest and the easiest form of XSS protection would be to pass all external data through a filter which will remove dangerous keywords, such as the infamous <SCRIPT> tag, JavaScript commands, CSS styles and other dangerous HTML markup (such as those that contain event handlers.)

When performing Escaping you are effectively telling the browser that the data you are sending should be treated as data and should not be interpreted in any other way. If an attacker manages to put a script on your page, the victim will not be affected because the browser will not execute the script if it is properly escaped.

Escaping has been used to construct this article. I have managed to bring many scripts into your browser, but none of these scripts has executed! The technique used to do that is called, escaping, or as the W3C calls it "Character Escaping".

In HTML you can escape dangerous characters by using the &# sequence followed by its character code.
An escaped < character looks like this: &#60. The > character is escaped like this: &#62.

Tool used	References

Burp Suite	http://code.tutsplus.com/tutorials/preventing-xss-in-aspnet--cms-21801

Proof of Vulnerability

In the below image attacker can execute malicious code from URL address bar. It can be show below figure 2.0

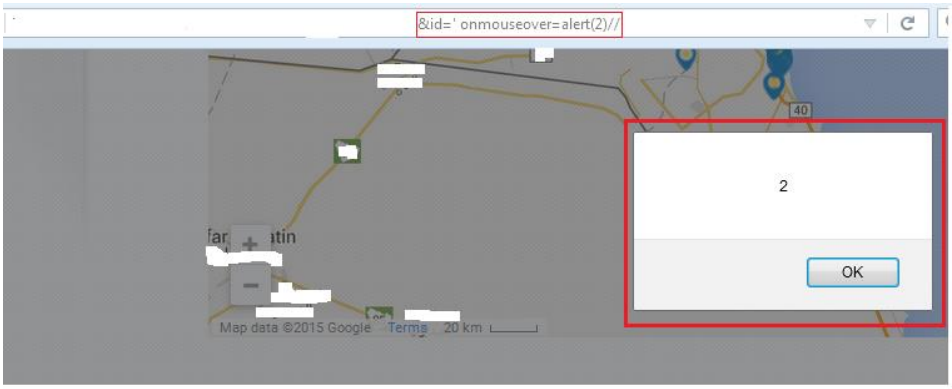


Figure 2.0 – Cross Site Scripting in twitter share link

2.3 INFORMATION DISCLOSER IN ACCOUNT CREATION FORM

Finding ID	Description
CLIENT-EXT-ID-05	Sensitive information is viewed in plain text in account.aspx file. The sensitive information should not be show in any page.
CVSS Score	**Risk Rating**
3.0 (Low)	**Low**
Affected Resource	
https://www.corp.com/CLIENT/Account.aspx	
Impact/Consequence	
It's bad from information leak point of view, So if attackers find a code that takes too long to execute this attack may turn into a denial of service.	
Recommendation	
Try to apply better methods and technique for sanitisation, filters and handling errors. A better solution is to turn off this validation, use validation only where needed. In earlier versions of ASP.NET, adding validateRequest="false" to the page directive in Webforms would turn the validation off for a page.	
Tool used	**References**
Burp Suite	http://code.tutsplus.com/tutorials/preventing-xss-in-aspnet--cms-21801
Proof of Vulnerability	
In the below image attacker is trying to feed malicious code and server discloses the arbitrary information, it's shown in below the figure 5.0	

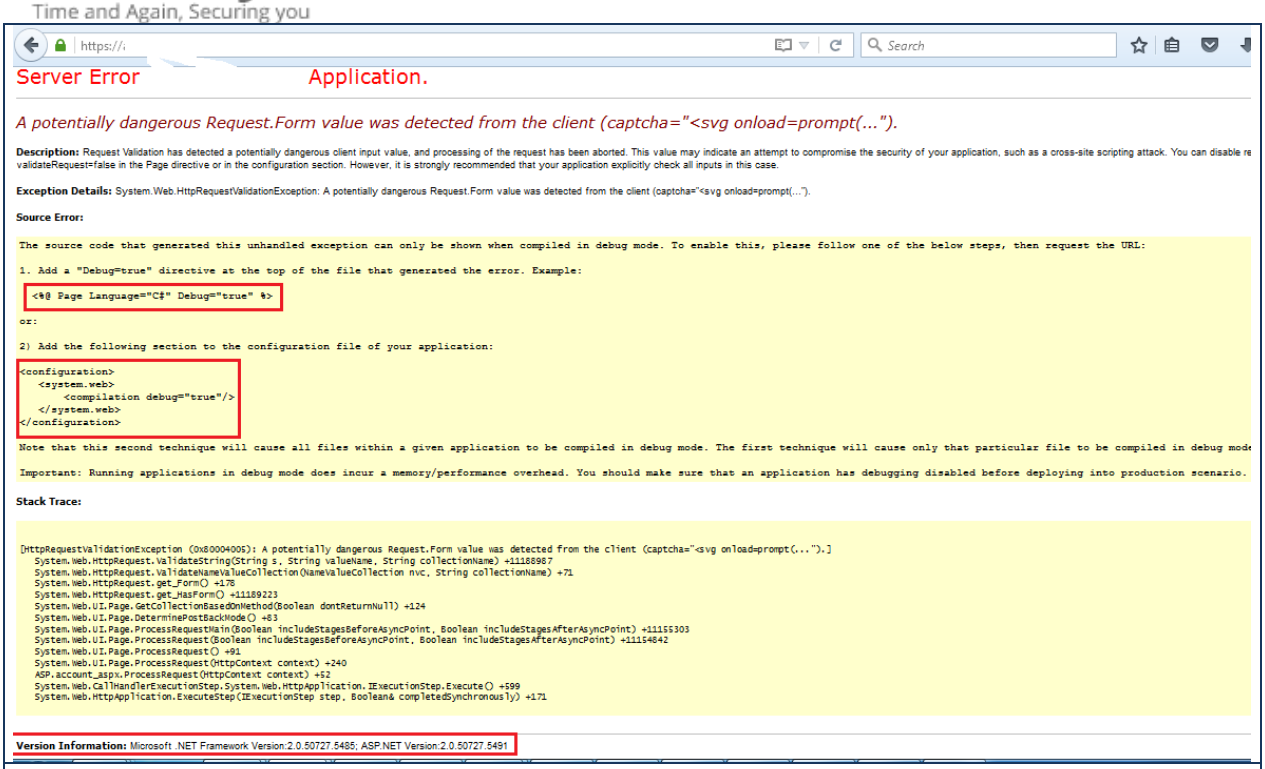


Figure 5.1 Debugging mode enable in Account.aspx

2.4 ASPX And Server Information Getting Disclosed on All Pages

Finding ID	Description
CLIENT-EXT-ID-08	Sensitive information is viewed in response code in online banking page. During the test, we found that in online banking page was vulnerable to information disclosure.
CVSS Score	**Risk Rating**
2.0 (Low)	**Low**
Affected Resource	
https://www.corp.com/corponline/GenericErrorINetCLIENT.aspx	
Impact/Consequence	
Attacker can use this data in further phases of penetration testing.	
Recommendation	
Use custom header for the server responses which should not contain any sensitive information about server.	
Tool used	**References**
Burp Suite	http://code.tutsplus.com/tutorials/preventing-xss-in-aspnet--cms-21801
Proof of Vulnerability	
In the below figure we can see the response code disclose the server information as well as version of server being used. The headers provided by the servers should not include any server information.	

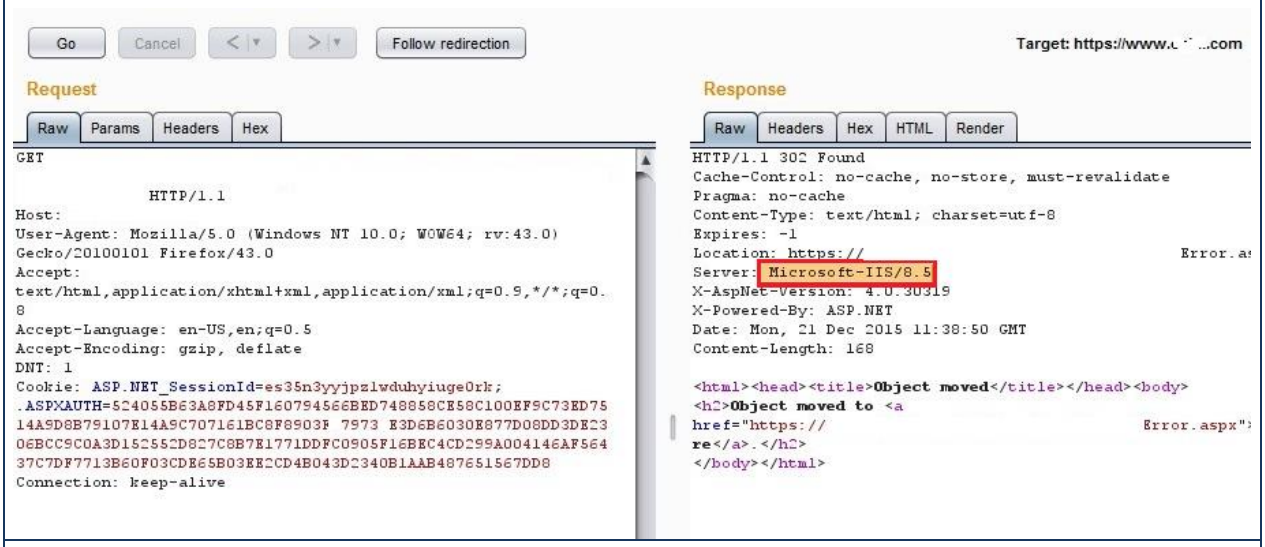


Figure 8.0 – Information disclose in online banking form

3. General Comments and Security Advice

3.1 Implement Centralized Filtering Method Potential

Another recommendation is to use a centralized filtering method for all values that can be tampered with by adversaries and actors other than the potentially affected user. This holds for the user-name, the conversation name as well as values being sent by a potentially rogue communication server instance. The instances might be tampered with by motivated attackers. Therefore, a centralized filter tool which would continuously ensure proper output filtering for any incoming byte-string is urged, as it would assist in keeping Application and any upcoming versions of the tool as safe and secure as possible.

3.2 Source Code Audit

Source Code Audit Review is a process to identify source code errors and find the un-sanitized portions in the source code of the application which can be a threat for the application security over the web, and can compromise the security of the application.

Our innovative methodology to audit source code for an application provides a comprehensive framework to identify the flaws and security issues inside the working source code of the application. In our source code audit methodology we don't rely only upon the automated tools for the source code audits. We do automate as well as manual source code review to cover all the problematic areas of the source code. "We at SecureLayer7 ensure the thorough auditing and reviewing of the source code of the application according to the defined standard".

4. CONCLUSION & ROADMAP

It is a well-known fact that no organization can claim 100% security due to ever changing threat and vulnerability scenarios. New vulnerabilities surface on a daily basis and both seasoned and casual hackers can exploit these vulnerabilities to cause serious harm to the organization. We conclude the Client's target system security posture as below:

IP Address/URL	Overall Security Posture	Comments
`https://sub.corp.com/path/Default.aspx`	**High**	During our analysis the applications were found to be High at risk to external attacks due to the lack of applicable security best practices.
`https://www.corp.com/corponline/LoginINetCLIENT.aspx?VectorIdRI=59cde347d7854183b986df1c1985ac83&`	**Medium**	During our analysis the applications were found to be Medium at risk to external attacks due to the lack of applicable security best practices.

The vulnerabilities identified when exploited by an adversary may cause:

- Disclosure of confidential information
- Reputational damage
- Service disruption

We recommend that CLIENT create a detailed plan for closure of the gaps found during this penetration test as soon as possible. The closure plan should address the highly severe vulnerabilities followed by the medium and low level vulnerabilities. The closure plan should be tested before making any changes to the production environment. Some of the recommendations for the closure plan are:

- **Harden** and **update** all the servers
- **Test** the critical devices on a periodic basis either quarterly or half yearly
- **Tighten** the application security
- **Enforce** corporate security policy on all assets and services
- **Develop** and render an awareness programme to keep employees up to date on information security and its implications

We also recommend that CLIENT adopt a programme based approach which ensures that all new vulnerabilities are identified in a timely manner and closed before they are exploited by an attacker. Thus the security posture of external facing systems can be maintained at a high level. This will assist in securing the information, IT infrastructure and image of CLIENT.

We thank CLIENT for giving us this opportunity and we assure full assistance in securing information and IT infrastructure in the long run.

5. ANNEXURE

5.1 OPEN WEB APPLICATION SECURITY PROJECT FRAMEWORK (OWASP)

The Open Web Application Security Project (OWASP) is a worldwide not-for-profit charitable organization. The OWASP Web Application Penetration Testing method is based on the black box approach. The tester knows nothing or very little information about the application to be tested. The test is divided into 2 phases:

- Passive mode: where the tester tries to understand the application's logic, and plays with the application. Tools can be used for information gathering, for example, an HTTP proxy to observe all the HTTP requests and responses. At the end of this phase, the tester should understand all the access points (gates) of the application.
- Active mode: In this phase, the tester begins to test using the following set of active tests in 9 sub-categories for a total of 66 controls:
    - Configuration Management Testing
    - Business Logic Testing
    - Authentication Testing
    - Session Management Testing
    - Authorization testing
    - Data Validation Testing
    - Denial of Service Testing
    - Web Services Testing
    - Ajax Testing

5.2 THE OPEN SOURCE SECURITY TESTING METHODOLOGY MANUAL (OSSTMM)

Open Source Security Testing Methodology Manual (OSSTMM) was written by Pete Herzog, and is being distributed by Institute for Security and Open Methodologies (ISECOM). It gives emphasis on getting business value. It gives broad description of categories of testing, and it includes step-by-step process description and information on penetration testing tools. OSSTM covers Competitive Intelligence Review, Internet Security (port scanning, firewalls, etc.), Communication Security, Physical Security, Wireless Security, etc.

5.3 NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST)

The United States National Institute of Standards and Technology (NIST) have released a document called as Technical Guide to Information Security Testing and Assessment which addresses and covers network penetration testing methodologies at a high level. This technical guideline has been prepared for use by federal agencies. It may be used by nongovernmental organizations for the evaluation of their security posture.

5.4 CVE - COMMON VULNERABILITIES AND EXPOSURE

Common Vulnerabilities and Exposures is a dictionary of common names (i.e., CVE Identifiers) for publicly known information security vulnerabilities. CVE's common identifiers make it easier to share data across separate network security databases and tools, and provide a baseline for evaluating the coverage of an organization's security tools. CVE is now the industry standard for vulnerability and exposure names. The process of creating a CVE Identifier begins with the discovery of a potential security vulnerability. The information is then assigned a CVE Identifier by a CVE Numbering Authority (CNA) and posted on the CVE List on the CVE Web site by the CVE Editor.

6. CONCLUSION

This analysis is based on the technologies and known threats as of the date of this report. SecureLayer7 recommends that all recommendations suggested in this document be performed in order to ensure the overall security of the systems and applications. Specifically, the following action should be taken:

- Setting up input validation mechanism in server side
- Password should be alpha-numeric character
- Maintaining the secret question
- Web application server hardening
- Implement the session management
- During the testing we found that there is no validation at server side
- Use custom header for the server responses.
- Send the proper X-Frame-Options HTTP response header.

SecureLayer7 suggests CLIENT to implement the recommendations in this document with respect to affected application. A review should be conducted post implementation of recommended counter measures to ensure that the vulnerabilities are fixed or closed.

END OF THE REPORT