

## Abstract

Securing and operating today's complex systems is challenging and demanding. Mission and operational requirements to deliver services and applications swiftly and securely have never been greater. This white paper stresses the need for effective security testing. Testing serves several purposes. No matter how well a given system may have been developed, the nature of today's complex systems with large volumes of code, complex internal interactions, interoperability with uncertain external components, unknown interdependencies coupled with vendor cost and schedule pressures, means that exploitable flaws will always be present or surface over time. Accordingly, security testing must fill the gap between the state of the art in system development and actual operation of these systems. Organizations that have an organized, systematic, comprehensive, on-going, and priority driven security testing regimen are in a much better position to make prudent investments to enhance the security posture of their systems.

## The Problem

The Internet has brought about many changes in the way organizations and individuals conduct business, and it would be difficult to operate effectively without the added efficiency and communications brought about by the Internet. At the same time, the Internet brought about problems as the result of intruder attacks, both manual and automated, which can cost many organizations excessive amounts of money in damages and lost efficiency. Thus, organizations need to find methods for achieving their mission goals in using the Internet and at the same time keeping their Internet sites secure from attack. Computer systems today are more powerful and more reliable than in the past; however they are also more difficult to manage.

## Understanding Security Testing

Organizations need to find methods for achieving their mission goals in using the Internet and at the same time keeping their Internet sites secure from attack. System administration is a complex task, and increasingly it requires that system administration personnel receive specialized training. Security testing is perhaps the most conclusive determinant of whether a system is configured and continues to be configured to the correct security controls and policy. Security testing, if made part of standard system and network administration, can be highly cost-effective in preventing incidents and uncovering unknown vulnerabilities. The types of security testing that can be performed are:

- Web Application Security – HTTP
- Directory Traversal
- Canonical Traversal
- Encoding Schemes – URL, Unicode, HTML, Base64
- Cross Site Scripting
- Injection Flaws
- Malicious File Execution
- Insecure Direct Object Reference
- Cross Site Request Forgery
- Information Leakage and Improper Error Handling
- Broken Authentication and Session Management
- Insecure Communication
- Failure to Restrict URL Access

### *Web Application Security*

In today's day and age, the major communication platform being used is the Internet, through which sensitive information like employee details, patient medical records, credit card information etc is regularly accessed. Most of this information flows through ports 80 and 443. With the increasing flow of information and the corresponding increase in application layer level attacks, it has become even more important to ensure application layer level security exists.

### *Directory Traversal*

It is also popularly known as the 'dot dot slash' method which enables directory traversal to access restricted files by intelligently crafting input requests. Testing is performed which helps to determine whether restricted files are accessed or not on the web server by manipulating URLs. We also aim to determine if access to restricted files allows a combination of browsing, reading and/or execution.

### *Canonical Traversal*

Canonical traversal is the use of various equivalent names which resolve to a single, standard name. Testing for canonical traversal includes use of abbreviated names, as well as aliases to determine if restricted paths in addition to sub paths are accessible or not. Further testing also focuses on path traversal sequences and special input characters which have the ability to by underlying file extension checks i.e., %00

### **URL Encoding Scheme**

Encoding is the process of transforming information from one format into another. Depending on the nature of encoding, it could either be test based or character based. Depending on the inclusive range for URL encoding, 0x20 - 0x7e, testing will also include special character encoding and the specific use of % sign, followed by two digit ACSII code.

### **Unicode Encoding Scheme**

Testing will aim to ensure 16-bit encoding, along with the UTF-8 variable length encoding standard. It is imperative to test for special characters which in this case are represented with the use of %u, followed by Unicode character point expressed in hexadecimal.

### **HTML Encoding Scheme**

HTML encoding is tested for security by verifying the appropriate use of & sign, followed by the ASCII code in hexadecimal form.

### **Base64 Encoding Scheme**

Testing for Base64 encoding aims to verify if only printable ASCII characters have been used, especially for HTTP authentication and email attachments for SMTP communication. Popular use of base64 encoding includes cookies. The use of = sign at the end of a string is an indication of base64 encoding at work.

### **Cross Site Scripting**

HTML integrity is verified as part of Cross Site Scripting testing to ensure that malicious script is not executed to steal session tokens or compromise web pages. Scripts are mostly found in search or Error pages, and any input data can be echoed i.e. query, post, get etc. As majority of browser technology is used for cross site scripting attack, it becomes essential to check popular technologies as well i.e., Flash, I Frame etc.

### **Injection Flaws**

During injection flaw testing, it becomes essential to verify whether database data is modified or accessed. Moreover, whether sensitive data on the server can be accessed by executing server side commands along with whether authentication is bypassed also form essential parts of injection flaw testing.

### **Malicious File Execution**

Testing for malicious file execution validates whether any part of the application causes the user to execute/download/create a file on the server which can seriously compromise server security and data integrity. Not only do such events enable complete server takeover, but potentially result in site defacement and cross site scripting options to be exploited as well.

### **Insecure Direct Object Reference**

Insecure direct object reference testing targets the possibility of a

resource name controlled by a user input, which can have severe implications i.e., allowing access to sensitive resources, information leakage or potentially allow future hacking attempts as well.

### **Cross Site Request Forgery**

The purpose of testing for cross site request forgery is to determine if the user is tricked into sending an unwitting, request to another site, using the user session and/or network access. Not only does this compromise local resources, it also has the potential of adversely affecting the external network as well.

### **Information Leakage and Improper Error Handling**

Such testing determines the possible causes behind information, particularly sensitive information, becoming available via errors, or other methods (accidental or intentional). Implications of not testing for such security flaws can lead to internal logic i.e., source code, SQL syntax etc to be exposed, and thus facilitating future hacks.

### **Broken Authentication and Session Management**

Testing for broken authentication and session management aims to verify if session tokens are guarded and validated is to ensure that such tokens are not planted by hackers by utilizing cross site scripting or cross site forgery attack methods.

### **Insecure Communication**

Testing for transmission of sensitive data over unencrypted channels is vital to ensure the integrity of the data is maintained. It is therefore imperative to verify if the data has been stolen or manipulated internally or externally as well to check whether strong encryption keys have been used.

### **Failure to Restrict URL Access**

Testing if resources that should only be available to authorized users can forcefully be accessed by browsing them is important to test to determine if application or web site functionality can be compromised by intelligently browsing to specific website/URL addresses which should in fact be restricted. This also has the potential of providing administrative access to a hacker which then enables them to make application or web page-wide changes to allow greater and more granular control i.e., horizontal and vertical access.

### **Insecure Cryptographic Storage**

Testing to determine if sensitive resources have only been secured with weak or no cryptographic protection is essential which reflects a lack of emphasis and safeguard on keys. Implication of such storage means that session tokens can be stolen, and sensitive data is accessible through database access via SQL injection.

## Solution

Security testing aims to address inherent flaws and weaknesses in applications to ensure that the final product is robust, and can safeguard sensitive data without compromising data confidentiality and integrity. Depending on testing time frames, more specific, or thoroughly comprehensive testing can be performed to determine the robustness of an application in the environment within which it will eventually be deployed.

## Conclusion

Security testing has gained unparalleled significance over the years, and will continue to rise up the popularity chart given the frequent advancements that we have come to notice and appreciate in the world of technology. Moreover, the importance of data integrity and confidentiality have begun to play a significant role, as have the client demands increased in terms of having access to secure software which delivers without compromising sensitive information. Our aim is to perform thorough security testing to highlight shortcomings and weaknesses in applications to enable developers to build more reliable, and dependable applications in future.