

**Security Testing Techniques**

**A complete guide of Web and Desktop Application Security testing techniques**

4/7/2017

MagikMinds Software Services Pvt. Ltd.

Surendra Madugula

**Contents**

[**Security Testing Definition**: 3](#_Toc479350755)

[**Desktop and Web Security Testing**: 3](#_Toc479350756)

[**Security Testing Techniques**: 3](#_Toc479350757)

[1) Access to Application: 3](#_Toc479350758)

[2) Data Protection: 3](#_Toc479350759)

[3. Brute-Force Attack: 4](#_Toc479350760)

[4. SQL Injection and XSS (cross site scripting): 4](#_Toc479350761)

[5. Service Access Points (Sealed and Secure Open): 5](#_Toc479350762)

# ****Security Testing Definition:****

**“Security means that authorized access is granted to protected data and unauthorized access is restricted”.** So, it has two major aspects; first is protection of data and second one is access to that data. Moreover, whether the application is desktop or web based, security revolves around the two aspects.

# ****Desktop and Web Security Testing:****

A desktop application should be secure not only regarding its access but also with respect to organization and storage of its data. Similarly, a web application demands even more security with respect to its access, along with data protection. Web developer should make the application immune to SQL Injections, Brute Force Attacks and XSS (cross site scripting). Similarly, if the web application facilitates remote access points then these must be secure too. Moreover, keep in mind that Brute Force Attack is not only related to web applications, desktop software is also vulnerable to this.

# Security Testing Techniques:

### **1) Access to Application:**

Whether it is a desktop application of website, access security is implemented by **‘Roles and Rights Management’.** It is often done implicitly while covering functionality, e.g.in a Hospital Management System a receptionist is least concerned about the laboratory tests as his job is to just register the patients and schedule their appointments with doctors. So, all the menus, forms and screen related to lab tests will not be available to the Role of ‘Receptionist’. Hence, the proper implementation of roles and rights will guarantee the security of access.

**How to Test:** To test this, thorough testing of all roles and rights should be performed. Tester should create several user accounts with different as well multiple roles. Then he should use the application with the help of these accounts and should verify that every role has access to its own modules, screens, forms and menus only. If tester finds any conflict, he should log a security issue with complete confidence.

### **2) Data Protection:**

There are further three aspects of data security. First one is that **a user can view or utilize only the data which he is supposed to use**. This is also ensured by roles and rights e.g. a TSR (tele sales representative) of a company can view the data of available stock, but cannot see how much raw material was purchased for production.

So, testing of this aspect is already explained above. The second aspect of data protection is related to **how that data is stored in the DB**. All the sensitive data must be encrypted to make it secure. Encryption should be strong especially for sensitive data like passwords of user accounts, credit card numbers or other business critical information. Third and last aspect is **extension of this second aspect**. Proper security measures must be adopted when flow of sensitive or business critical data occurs. Whether this data floats between different modules of same application, or is transmitted to different applications it must be encrypted to make it safe.

**How to Test Data Protection:** The tester should query the database for ‘passwords’ of user account, billing information of clients, other business critical and sensitive data and should verify that all such data is saved in encrypted form in the DB. Similarly (s)he must verify that between different forms or screens, data is transmitted after proper encryption. Moreover, tester should ensure that the encrypted data is properly decrypted at the destination. Special attention should be paid on different ‘submit’ actions. The tester must verify that when the information is being transmitted between client and server, it is not displayed in the address bar of web browser in understandable format. If any of these verifications fail, the application has security flaw.

### **3. Brute-Force Attack:**

Brute Force Attack is mostly done by some software tools. The concept is that using a valid user ID, s**oftware attempts to guess the associated password by trying to login again and again.** A simple example of security against such attack is account suspension for a short period of time as all the mailing applications like ‘Yahoo’ and ‘Hotmail’ do. If, a specific number of consecutive attempts (mostly 3) fail to login successfully, then that account is blocked for some time (30 minutes to 24 hrs.).

**How to test Brute-Force Attack:** The tester must verify that some mechanism of account suspension is available and is working accurately. (S)He must attempt to login with invalid user IDs and Passwords alternatively to make sure that software application blocks the accounts that continuously attempt login with invalid information. If the application is doing so, it is secure against brute-force attack. Otherwise, this security vulnerability must be reported by the tester.

**The above three security aspects should be considered for both web and desktop applications while, the following points are related with web based applications only.**

### **4. SQL Injection and XSS (cross site scripting):**

Conceptually speaking, the theme of both these hacking attempts is similar, so these are discussed together. In this approach, **malicious script is used by the hackers to manipulate a website**. There are several ways to immune against such attempts. For all input fields of the website, field lengths should be defined small enough to restrict input of any script e.g. Last Name should have field length 30 instead of 255. There may be some input fields where large data input is necessary, for such fields proper validation of input should be performed prior to saving that data in the application. Moreover, in such fields any html tags or script tag input must be prohibited. To provoke XSS attacks, the application should discard script redirects from unknown or untrusted applications.

**How to test SQL Injection and XSS:** Tester must ensure that maximum lengths of all input fields are defined and implemented. (S)He should also ensure that defined length of input fields does not accommodate any script input as well as tag input. Both these can be easily tested e.g. if 20 is the maximum length specified for ‘Name’ field; and input string “<p>thequickbrownfoxjumpsoverthelazydog” can verify both these constraints. It should also be verified by the tester that application does not support anonymous access methods. In case any of these vulnerabilities exists, the application is in danger.

### **5. Service Access Points (Sealed and Secure Open):**

Today, businesses depend and collaborate with each other, same holds good for applications especially websites. In such case, both the collaborators should define and publish some access points for each other. So far, the scenario seems quite simple and straightforward but, for some web based product like stock trading, things are not so simple and easy. When there is enormous number of target audience, the access points should be open enough to facilitate all users, accommodating enough to fulfill all users’ requests and secure enough to cope with any security-trial.

In some cases, these **access points can be sealed for unwanted applications or people**. This depends upon the business domain of application and its users, e.g. a custom web based Office Management System may recognize its users based on IP Addresses and denies establishing a connection with all other systems (applications) that do not lie in the range of valid IPs for that application.

Tester must ensure that all the**inter-network and intra-network access** to the application is from trusted applications, machines (IPs) and users. To verify that a direct access point is secure enough, tester must try to access it from different machines having both trusted and untrusted IP addresses. Different sort of real-time transactions should be tried in a bulk to have a good confidence of application’s performance.  By doing so, the capacity of access points of the application will also be observed clearly.

Tester must ensure that the application entertains all the communication requests from trusted IPs and applications only while all the other request is rejected. Similarly, if the application has some open access point, then tester should ensure that it allows (if required) uploading of data by users in secure way. By this secure way I mean, the file size limit, file type restriction and scanning of uploaded file for viruses or other security threats. This is all how a tester can verify the security of an application with respect to its access points.