**Security testing**

1) **Information Gathering-**

[4.2.1 Conduct Search Engine Discovery and Reconnaissance for Information Leakage (OTG-INFO-001)](https://www.owasp.org/index.php/Conduct_search_engine_discovery/reconnaissance_for_information_leakage_(OTG-INFO-001))

[4.2.2 Fingerprint Web Server (OTG-INFO-002)](https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002))

[4.2.3 Review Webserver Metafiles for Information Leakage (OTG-INFO-003)](https://www.owasp.org/index.php/Review_Webserver_Metafiles_for_Information_Leakage_(OTG-INFO-003))

[4.2.4 Enumerate Applications on Webserver (OTG-INFO-004)](https://www.owasp.org/index.php/Enumerate_Applications_on_Webserver_(OTG-INFO-004))

[4.2.5 Review Webpage Comments and Metadata for Information Leakage (OTG-INFO-005)](https://www.owasp.org/index.php/Review_webpage_comments_and_metadata_for_information_leakage_(OTG-INFO-005))

[4.2.6 Identify application entry points (OTG-INFO-006)](https://www.owasp.org/index.php/Identify_application_entry_points_(OTG-INFO-006))

[4.2.7 Map execution paths through application (OTG-INFO-007)](https://www.owasp.org/index.php/Map_execution_paths_through_application_(OTG-INFO-007))

[4.2.8 Fingerprint Web Application Framework (OTG-INFO-008)](https://www.owasp.org/index.php/Fingerprint_Web_Application_Framework_(OTG-INFO-008))

[4.2.9 Fingerprint Web Application (OTG-INFO-009)](https://www.owasp.org/index.php/Fingerprint_Web_Application_(OTG-INFO-009))

[4.2.10 Map Application Architecture (OTG-INFO-010)](https://www.owasp.org/index.php/Map_Application_Architecture_(OTG-INFO-010))

Conduct search engine discovery

Use a search engine to search for:

* Network diagrams and configurations
* Archived posts and emails by administrators and other key staff
* Log on procedures and username formats
* Usernames and passwords
* Error message content
* Development, test, UAT and staging versions of the website

Use different search engines as all have different algorithms to pull results.

Google Hacking Database is list of useful search queries for Google.

## **Tools**

[4] FoundStone SiteDigger - <http://www.mcafee.com/uk/downloads/free-tools/sitedigger.aspx>   
[5] Google Hacker - <http://yehg.net/lab/pr0js/files.php/googlehacker.zip>  
[6] Bishop Fox's Google Hacking Diggity Project - <http://www.bishopfox.com/resources/tools/google-hacking-diggity/>   
[7] PunkSPIDER - <http://punkspider.hyperiongray.com/>

**2) Testing for Configuration and Deployment management-**

[4.3.1 Test Network/Infrastructure Configuration (OTG-CONFIG-001)](https://www.owasp.org/index.php/Test_Network/Infrastructure_Configuration_(OTG-CONFIG-001))

[4.3.2 Test Application Platform Configuration (OTG-CONFIG-002)](https://www.owasp.org/index.php/Test_Application_Platform_Configuration_(OTG-CONFIG-002))

[4.3.3 Test File Extensions Handling for Sensitive Information (OTG-CONFIG-003)](https://www.owasp.org/index.php/Test_File_Extensions_Handling_for_Sensitive_Information_(OTG-CONFIG-003))

[4.3.4 Review Old, Backup and Unreferenced Files for Sensitive Information (OTG-CONFIG-004)](https://www.owasp.org/index.php/Review_Old,_Backup_and_Unreferenced_Files_for_Sensitive_Information_(OTG-CONFIG-004))

[4.3.5 Enumerate Infrastructure and Application Admin Interfaces (OTG-CONFIG-005)](https://www.owasp.org/index.php/Enumerate_Infrastructure_and_Application_Admin_Interfaces_(OTG-CONFIG-005))

[4.3.6 Test HTTP Methods (OTG-CONFIG-006)](https://www.owasp.org/index.php/Test_HTTP_Methods_(OTG-CONFIG-006))

[4.3.7 Test HTTP Strict Transport Security (OTG-CONFIG-007)](https://www.owasp.org/index.php/Test_HTTP_Strict_Transport_Security_(OTG-CONFIG-007))

[4.3.8 Test RIA cross domain policy (OTG-CONFIG-008)](https://www.owasp.org/index.php/Test_RIA_cross_domain_policy_(OTG-CONFIG-008))

Test File extensions- files having different extensions may help in understanding web server behaviour. These are indicative of technologies, languages or plugins that are used by web servers or application servers.

For example, a “.pl” extension is usually associated with server-side Perl support.

The following file extensions should never be returned by a web server, since they are related to files which may contain sensitive information or to files for which there is no reason to be served.

* .asa
* .inc

To identify files having a given extensions a mix of techniques can be employed. THese techniques can include Vulnerability Scanners, spidering and mirroring tools, manually inspecting the application (this overcomes limitations in automatic spidering), querying search engines (see [Testing: Spidering and googling](https://www.owasp.org/index.php/Testing:_Spidering_and_googling)).

# Test Application Platform Configuration -

#### Sample and known files and directories

#### Comment review

It is very common, and even recommended, for programmers to include detailed comments on their source code in order to allow for other programmers to better understand why a given decision was taken in coding a given function. However, comments included inline in HTML code might reveal internal information that should not be available to an attacker. Sometimes, even source code is commented out since a functionality is no longer required, but this comment is leaked out to the HTML pages returned to the users unintentionally.

Comment review can be done through an analysis of the web server static and dynamic content and through file searches.

Logging is an important asset of the security of an application architecture, since it can be used to detect flaws in applications. the main intention of the application logs is to produce debugging output that could be used by the programmer to analyze a particular error. server logs might contain sensitive information.

# Test HTTP Methods

the Hypertext Transfer Protocol (HTTP) allows several methods that are used to access information provided by a web server.

* HEAD
* GET
* POST
* **PUT-** This method allows a client to upload new files on the web server. An attacker can exploit it by uploading malicious files
* **DELETE-** This method allows a client to delete a file on the web server.
* **TRACE-** This method simply echoes back to the client whatever string has been sent to the server,
* OPTIONS
* **CONNECT-** This method could allow a client to use the web server as a proxy.

Some of these methods(highlighted) can potentially pose a security risk for a web application.

How to test-

The OPTIONS method represents a request for information about the communication options available on the request/response chain identified by the Request-URI". OPTIONS provides a list of the methods that are supported by the web server