HP WebInspect Tutorial

Introduction:

With the exponential increase in internet usage, companies around the world are now obsessed about having a web application of their own which would provide all the functionalities to their users with a single click. In this quest for providing the customers with one-click-done options, all the sensitive data is shifted on to a server which is then accessed by a web application. Web applications control valuable data as they have direct access to the backend database. With a simple well crafted malicious payload a hacker can now get all the information from database. So it's crucial that the web applications need to be secure enough to handle the attacks.

Securing Web applications:

It's now apparent that securing web applications is essential for the companies to be in business. The real question is how to achieve that. Below are some of the checks that are in place to ensure that security holes in the web application are identified:

1. **Threat Modeling** deals with identifying threats, attacks, vulnerabilities, and countermeasures for your application in the design phase.
2. **Security Code Reviews** comes into picture at the end of development phase. The entire code is inspected to find vulnerabilities.
3. **Manual Penetration Testing** is done after the application is deployed in some environment. The application is attacked and assessed for vulnerabilities.
4. **Automated Vulnerability Scanners** are the tools which aid Penetration testers by identifying the vulnerabilities present.

WebInspect is one of the most widely used automated vulnerability scanners in the market today. It helps us to identify vulnerabilities present in the web application by taking necessary input from us. For the rest of this article I will be focusing on using WebInspect to identify security vulnerabilities.

WebInspect:

WebInspect is a web application security scanning tool offered by HP. It helps the security professionals to assess the potential security flaws in the web application. WebInspect is basically a dynamic black box testing tool which detects the vulnerabilities by actually performing the attack. After initiating the scan on a web application, there are 'assessment agents' that work on different areas of the application. They report their results to 'security engine' which evaluates the results. It uses 'Audit engines' to attack the application and determine the vulnerabilities. At the end of the scan you can generate a report called 'Vulnerability Assessment Report' which would list the security issues in desired format. Using this

report client can fix the issues and then go for validation scanning to confirm the same. As with every other tool there are both advantages and disadvantages associated with using WebInspect.

Advantages:

1. Saves time when dealing with large enterprise applications
2. Simulates the attack, shows the results and presents you with a comprehensive view.
3. It is not dependent on the underlying language.

Disadvantages:

1. It's hard for any tool to find logical flaws, weak cryptographic storage, severity of the disclosed information etc.
2. It has a list of payloads that it uses on every web application. It does not use any wisdom in generating payloads depending on the type of application.
3. There could be false positives among the listed vulnerabilities.

Having said that, WebInspect scores high on many features and helps a great deal in providing scanning solutions.

Main Features in WebInspect 9.10:

WebInspect 9.10 is the latest version in use as of today. Below lines would throw an insight into various features that are available in WebInspect.

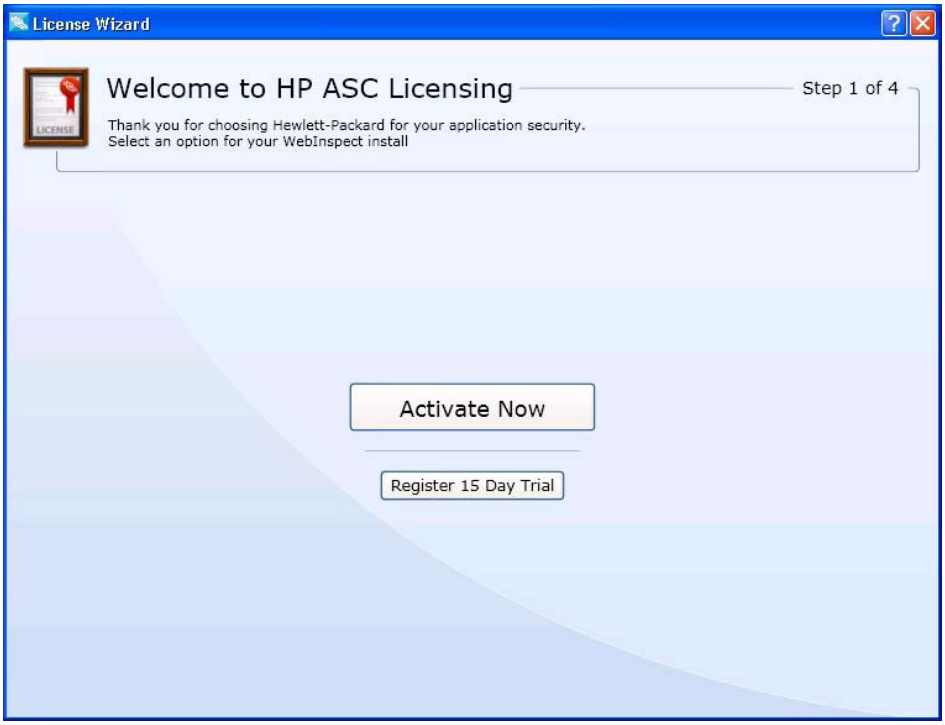
- Presents you with tree structure: By crawling the entire application WebInspect presents you with the hierarchical tree structure of the web application by listing all the available URLS.
- Customizable Views: While viewing the results of the scan WebInspect offers you sortable views as per your requirement.
- Scanning Policies: WebInspect gives you the freedom to edit and customize the scanning policies to suit your requirements and thus gives great flexibility.
- Manual Hacking Control: With this option you can actually simulate a true attack environment and see what's really happening for a particular attack.
- Report Generation: You can generate customizable reports by including desired sections and in desired format.
- Remediation: WebInspect would provide a summary and the necessary fixes required to fix the vulnerabilities detected during a particular scan.

- Web Services Scan: Web services usage is growing at a rapid pace. You can assess web service vulnerabilities by using WebInspect.
- Tools: There are lot many tools that come with WebInspect like web proxy, SQL Injector, web fuzzer, web macro recorder etc.

We will now move into the actual scanning part and will explore the tool and its features.

Installation Part:

Before you install WebInspect make sure that you have 2 GB RAM and Microsoft SQL Server installed. After installation, the first time you start WebInspect it will open the 'License Wizard' and prompt you to activate by entering the license key. If you don't have one you can go for a 15 day trial period for which activation token will be sent to your mail after giving the details.



Depending on the security policy selected, WebInspect will aggressively attack the web application which can affect the server. It sends many HTTP requests which results in increased traffic. So make sure that you keep these things in mind and accordingly conduct the scan.

THE TWO "C+A"s:

Two things that WebInspect will do for you: Crawl + Audit

Two things that you need to do for WebInspect: Configure + Analyze.

Crawl: Crawling is the process by which WebInspect will build the tree structure of the entire website by traversing every possible link on that site.

Audit: Auditing is the process of performing attacks to assess the vulnerabilities.

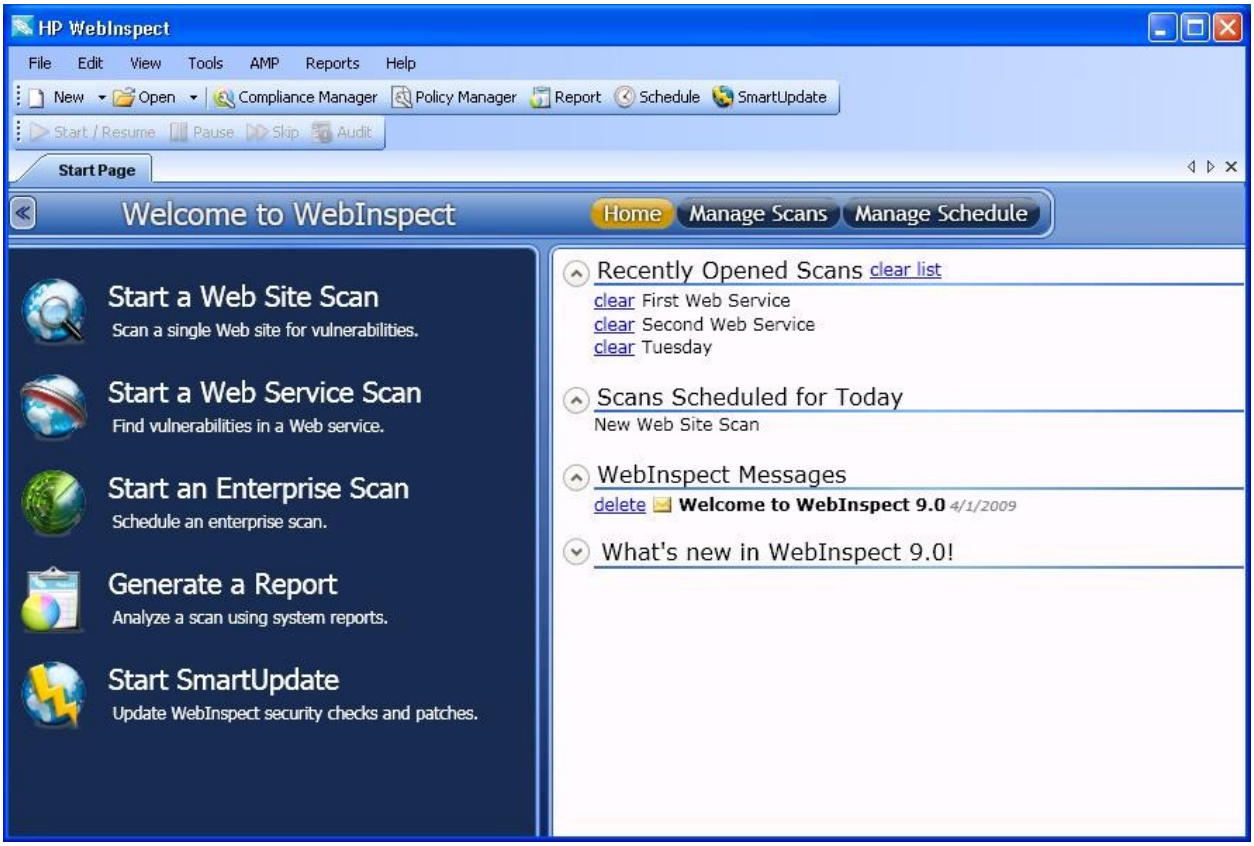
Crawl + Audit = Scan

Configure: You need to tell the WebInspect what you need from it. If you do not want it to hit a particular functionality in your site you have to tell it. If you want to find out only XSS & SQLi vulnerabilities you have to tell it. So configuring is basically letting the WebInspect know what you want and what you do not want.

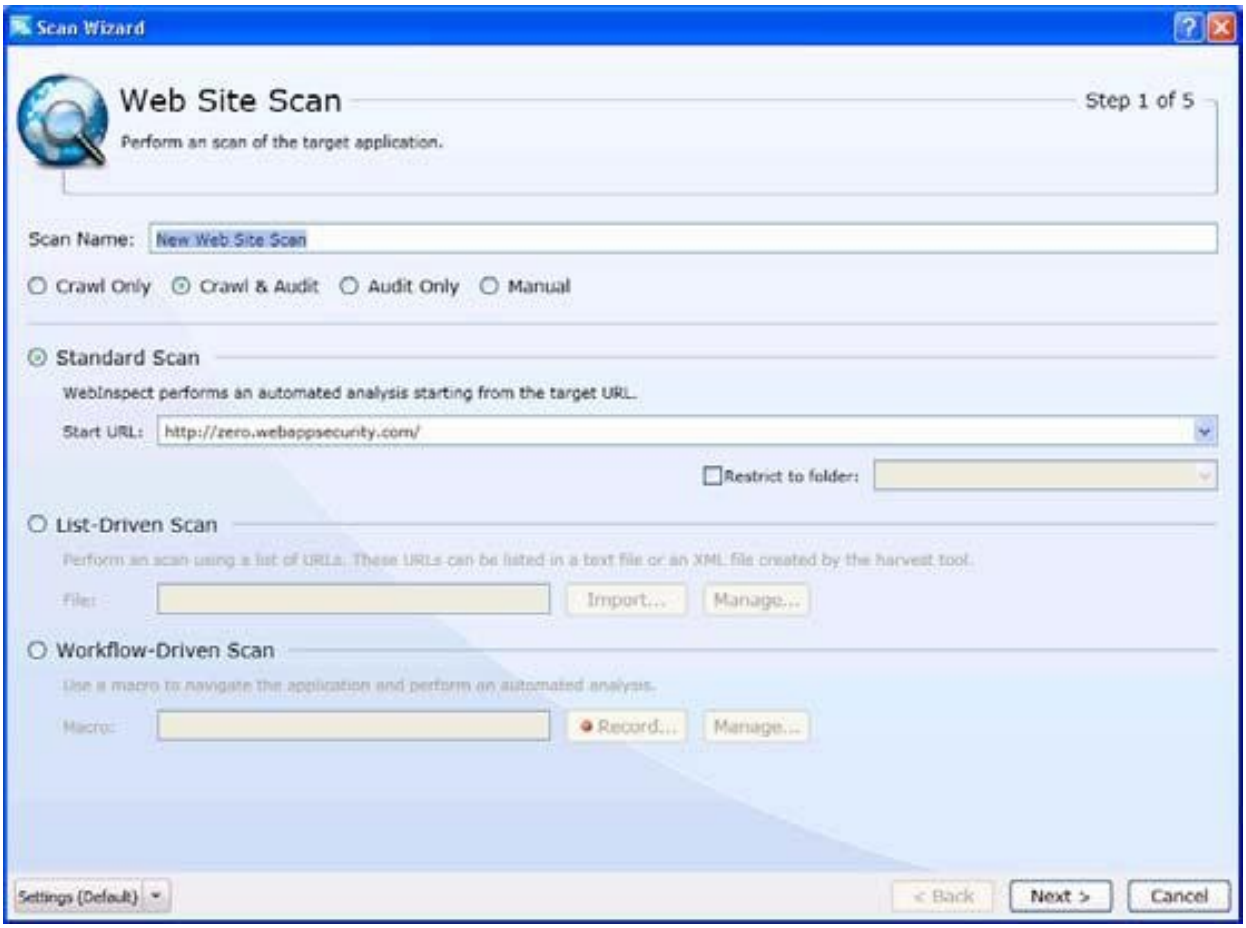
Analyze: Here you need to analyze the results presented by the WebInspect and eliminate the false positives.

Starting a scan:

To begin a scan, start WebInspect and click File-->New. As you can see in the below picture 'Scan Wizard' opens and you can select the type of scan you want to conduct. So select 'Website scan' (Both web service and Enterprise scan will be discussed in another post). In the scan wizard on the right hand side you can see the recently opened scans and the scans that are in schedule. You can schedule a scan to begin at a particular time.



Upon selecting the Website scan you will be taken to the below window where you need to enter the scan name. Select crawl and audit button and select the type of scan.



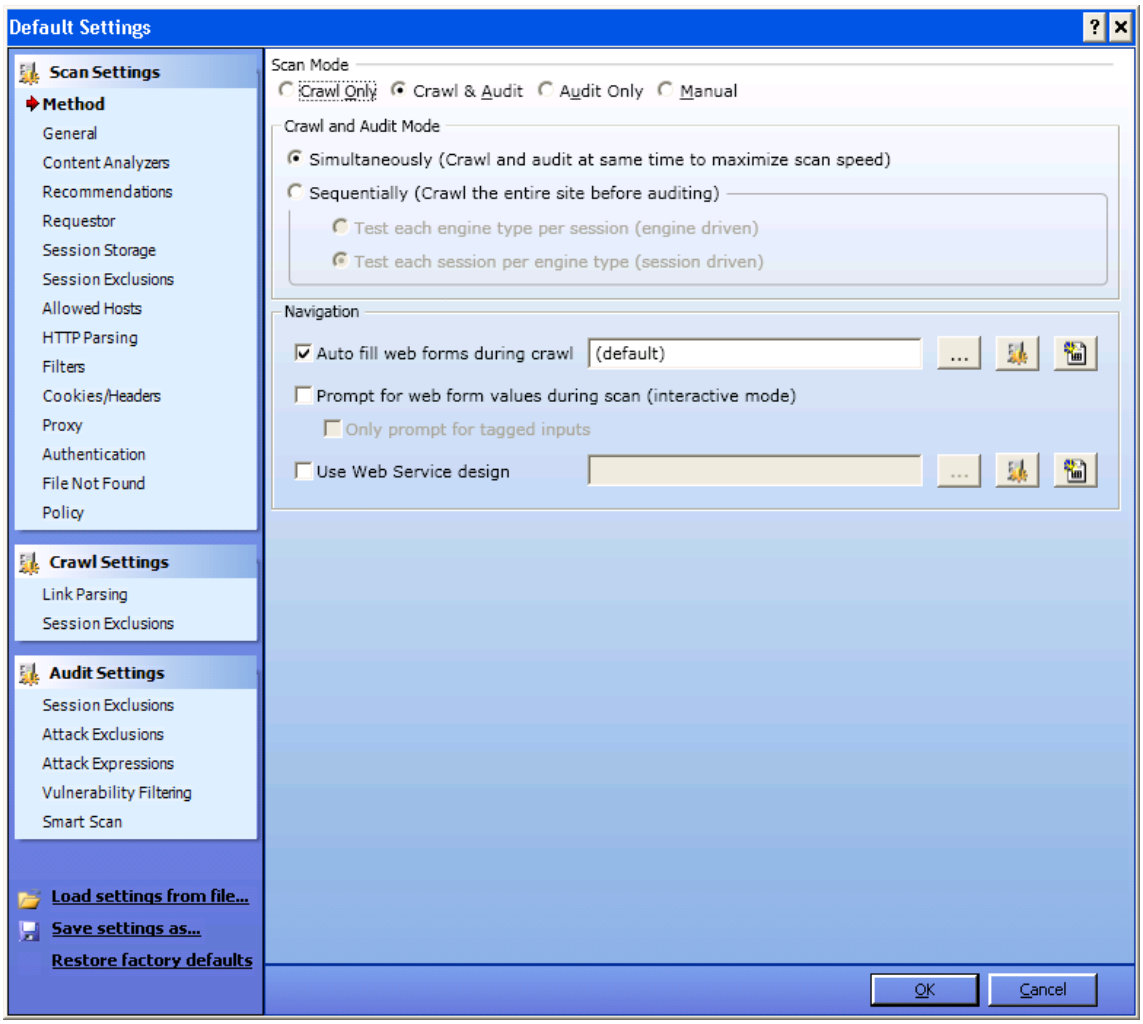
Standard scan: Used in most of the cases. It is normal way to start a scan.

List Driven scan: Allows you to specify the list of URLs that are to be scanned. Only those URLs will be scanned.

Workflow Driven scan: This is used to scan only a part of your site not the entire site. The part that needs to be scanned can be specified by a workflow macro which we will be looking into soon.

Manual scan: Allows you to manually specify the links that are to be scanned by browsing through them in the step mode.

In the bottom left hand side there is a button ‘Settings (Default)’ which is the heart of WebInspect. Using this we configure the scan and tell WebInspect what we want from it. Click on Settings (Default) and ‘Default Settings’ window will open.



There are many options and sub options present in this category. I will try to cover as many as possible and the left over ones are something which are easy to understand. Under default settings, as you can see on the left hand side of the below picture we have Scan settings, Crawl settings and Audit settings.

Scan settings:

Method

Based on your input in the previous window, scan mode will be shown here automatically as 'crawl and audit'. As seen earlier to conduct a scan, WebInspect has to crawl and audit.

Simultaneously vs Sequential: If it crawls and audits simultaneously it's called 'Simultaneously' mode. If it crawls the entire site and then audits one by one it's called 'Sequential' mode. So you can select the option you prefer. If your site content changes before the crawl gets completed the go for

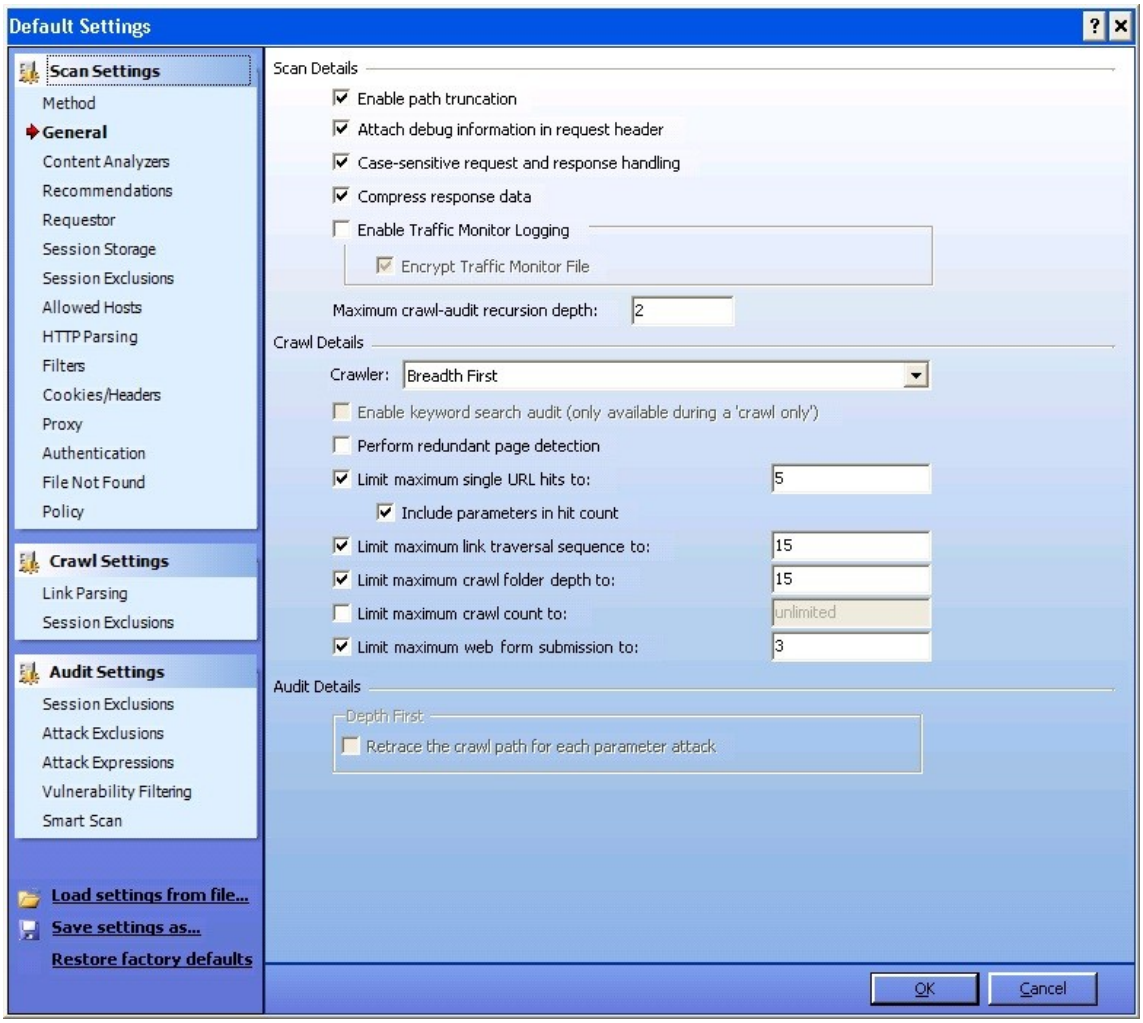
'Simultaneously' mode. If you select 'Sequential' you need to select the order in which crawl and audit have to take place.

Test each engine type per session: WebInspect audits all sessions using the first audit engine, then audits all sessions using the second audit engine etc

Test each session per engine type: WebInspect runs all audit engines against the first session, then runs all audit engines against the second session, continuing in sequence until all sessions are audited.

During the scan WebInspect will encounter pages where input is required to move to the next page. If you want WebInspect to auto fill those forms you can select auto fill option under navigation. If you want WebInspect to ask you for values then select 'prompt for values option'. (But be present during the scan or the scan wouldn't proceed without the input). Now click on 'general' tab.

<u>General</u>:



Enable Path Truncation: If you want WebInspect to look for 'Path Truncation Attacks' (requesting directories without filenames) select it.

Attach debug information in request header: WebInspect will include a header 'Memo' in the HTTP request which can be used for debugging purpose.

Case sensitive request response handling: If the server you are hitting is case sensitive then select this option

Compress response data: WebInspect will save you some space by storing the response in compressed manner in its database.

Enable Traffic Monitor Logging: Each and every request and response will be logged and you can view it later under 'Traffic Monitor' option while analyzing.

Max crawl audit recursion depth: If vulnerability is found in one page WebInspect crawls and follows the link. If that link points to another then recursion depth is one. If that link points to another then recursion depth is 2. Default value is 2 and maximum value is 1000.

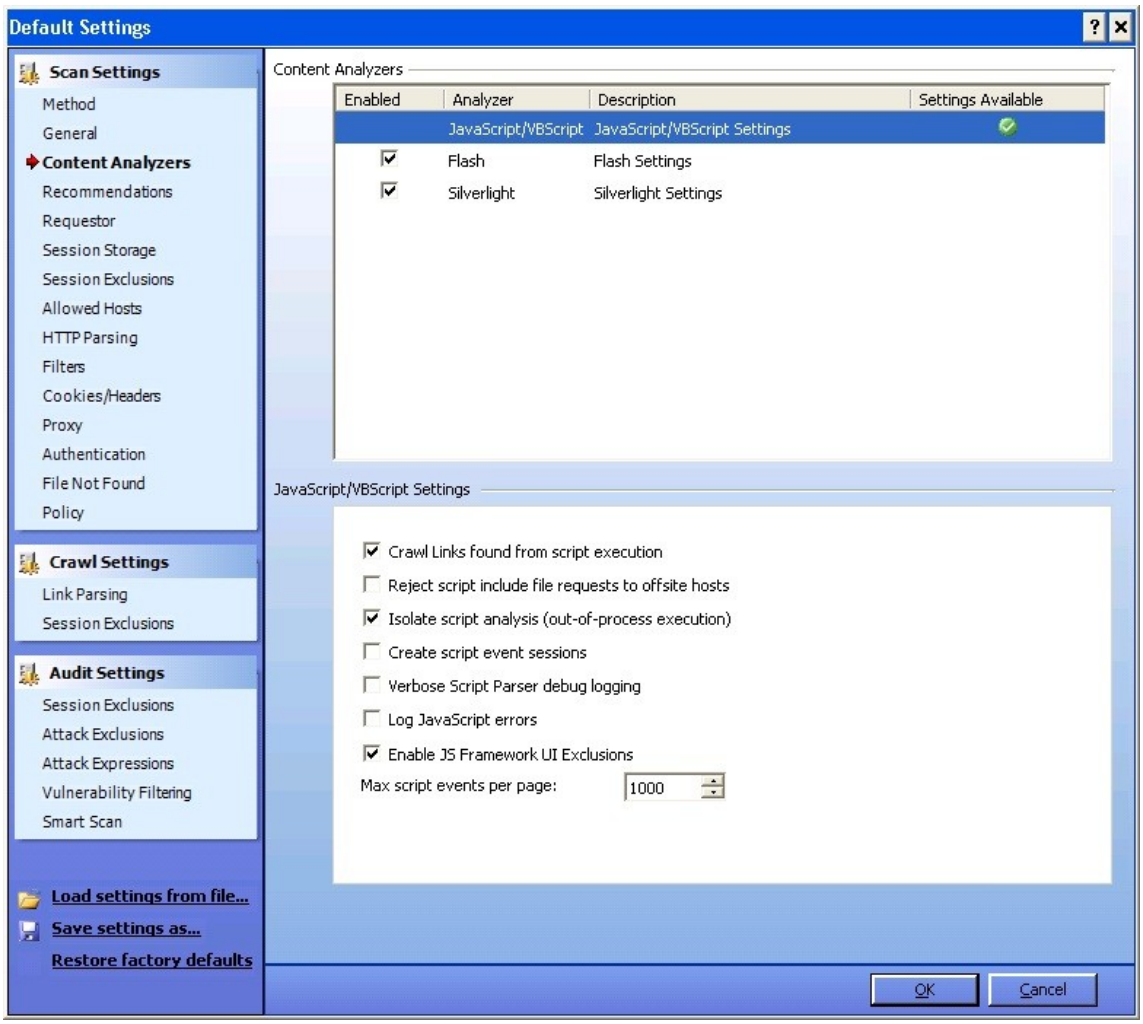
Depth first vs Breadth first: If your web application follows ordering of requests (for ex in online shopping cart, the user is required to visit shopping cart page before accessing check out page) then select depth first or else you can go with breadth first.

Limit maximum single URL hits to: This is regarding the number of times a page can be hit by WebInspect. This is important because sometimes depending on the architecture of your site WebInspect might enter into an endless loop. So in such situation this option comes to your rescue.

The functionality of other options present under this tab is easily guessable by their name. Click on the next tab 'Content Analyzer'.

Content Analyzer:

This deals with the settings regarding the content that has to be scanned.



Flash: Select this if you want WebInspect to analyze flash files.

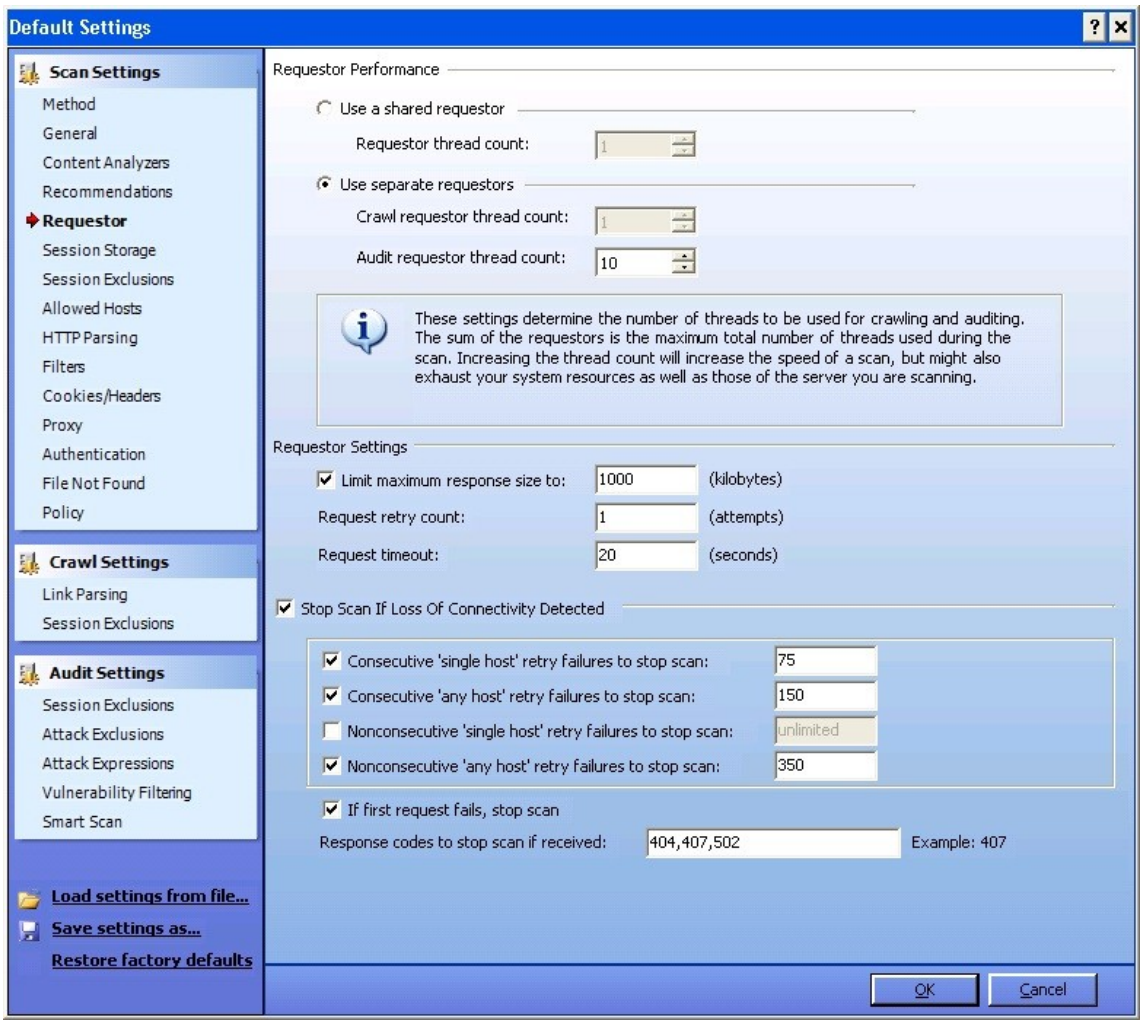
Java Script/VB Script: This is enabled by default. Click on this and you will find other options under this using which you can reject scripts that include requests to offsite hosts, log JavaScript errors etc.

Recommendations:

If this option is enabled, at the end of the scan WebInspect will present you the list of recommendations to perform the scan better the next time.

Requestor:

Requestor deals with HTTP requests and responses.



Requestor Performance: Shared Requestor vs Separate Requestor:

With shared requestor, crawler and auditor use common requestor while scanning a site and they use the same state. With separate requestor, both crawler and auditor use separate requestors. If maintaining state is not an issue then you can go with shared requestor. Alternately, separate requestors would result in much faster scans.

Requestor Settings:

Limit max response size to: You can specify the maximum response size that can be accepted from the server. However this does not apply to flash files.

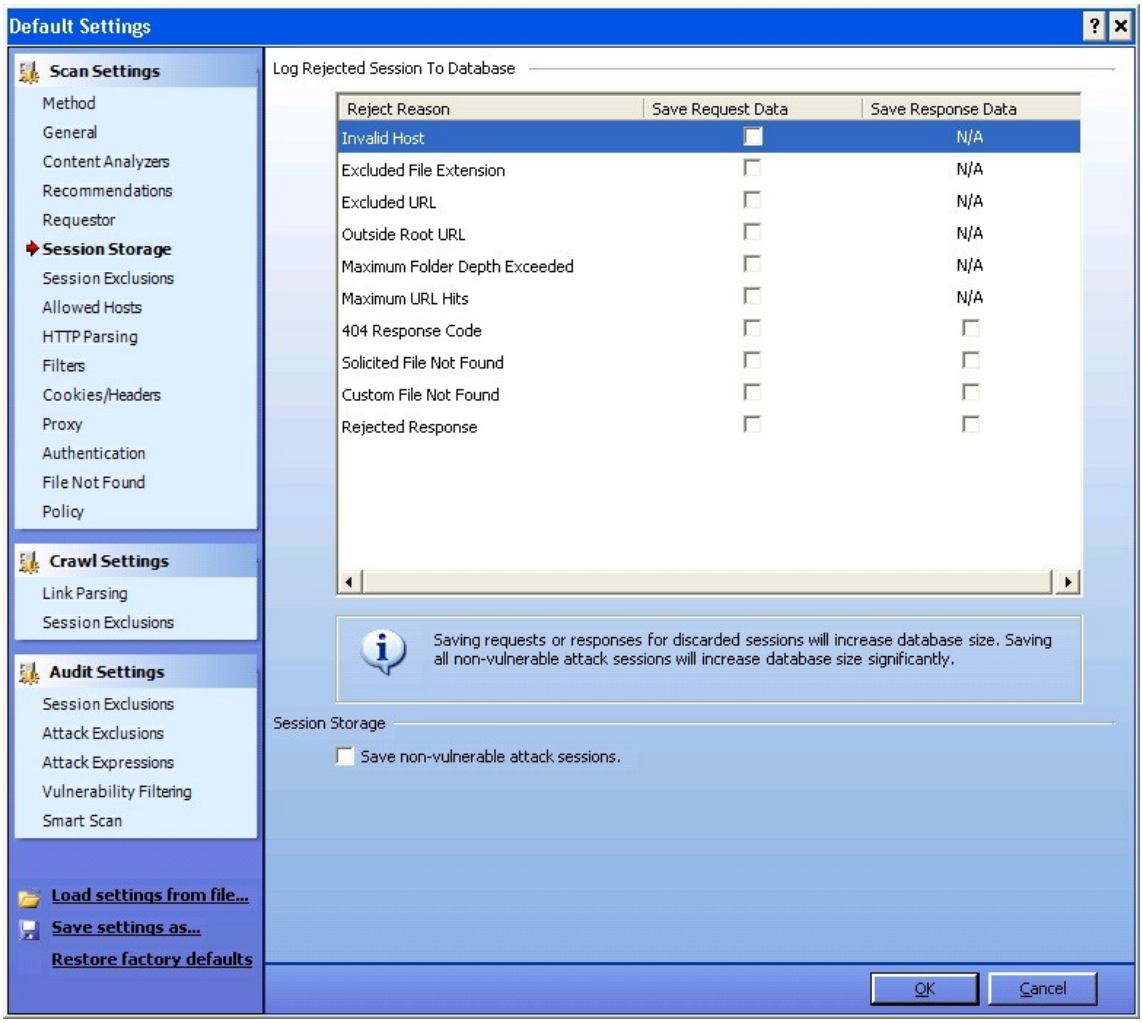
Request retry count: You can specify how many times WebInspect can resend a HTTP request after it receives a failed response.

Request timeout: You can specify how long WebInspect can wait for HTTP response.

Stop scan if loss of connectivity detected: During the scan WebInspect encounters variety of situations like server is not responding etc. So you can specify some conditions under this section which would instruct Webinspect to stop the scan if those conditions are detected.

Session Storage:

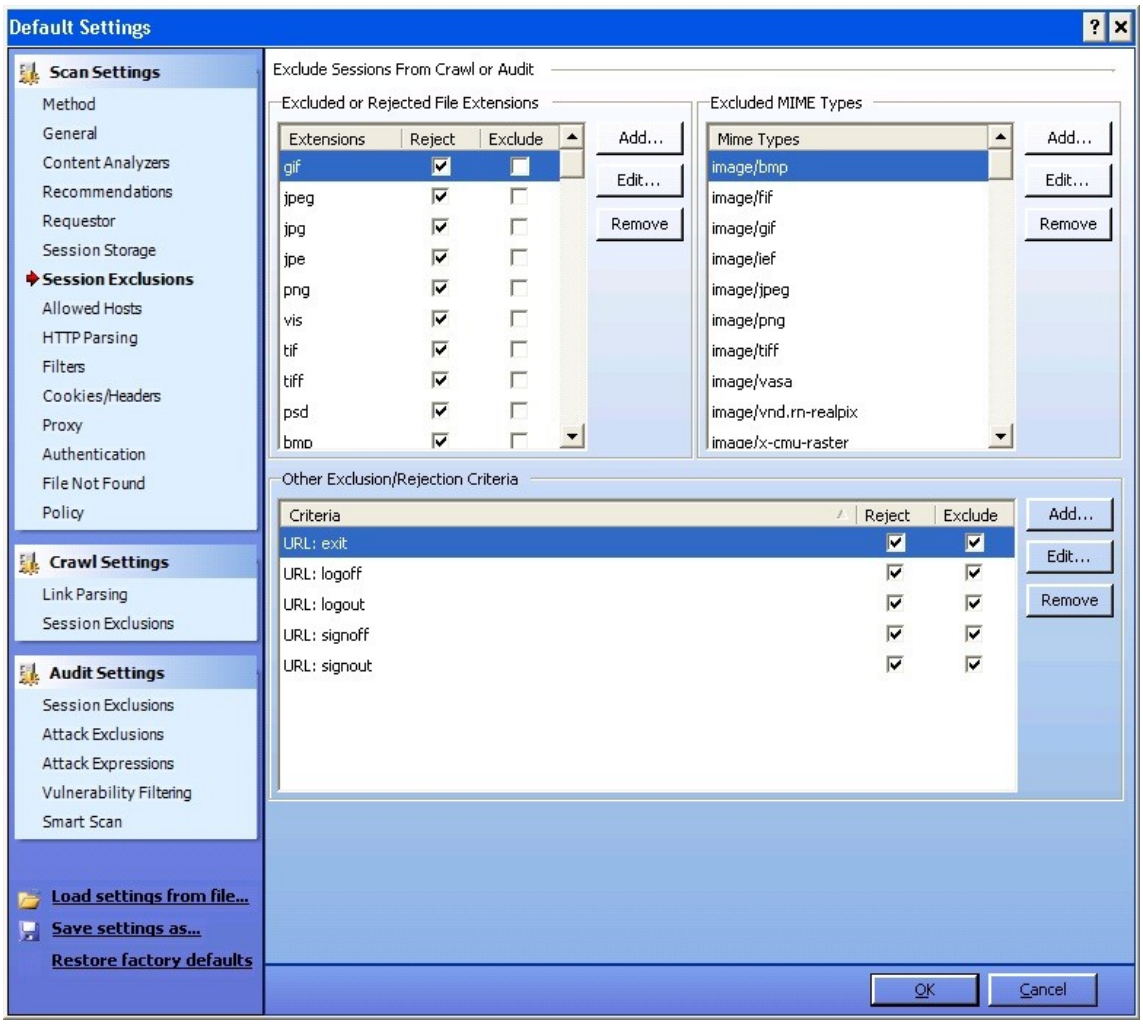
As you can see from the below picture, you can log the rejected sessions. You can save the request data and response data as applicable to them by enabling the below options.



Session Exclusions:

Excluded or rejected file extensions:

Using this Using this you can exclude or reject certain file extensions from either crawl or audit or both. If you are rejecting a file, WebInspect will not request the file at all. If you are excluding a file WebInspect will request but it will not attack them during the audit phase.



Similarly you can also specify MIME types and URL types that need to be rejected. For example during a scan you don't want to hit the logout button. Here if the url goes something like www.ex.com/abc.jsp?logout=true then by specify the criteria above you can avoid this url hitting.

Allowed Hosts:

You might be using multiple domains for your website. So you can add those domains under this section so that they will be allowed during the crawl and audit. During the scan if WebInspect encounters other domains it will not scan them. So you can specify those domains under this section so that they will be included.

HTTP Parsing:

If your application uses URL rewriting or any post data techniques to maintain the state in the application then you need to identify the parameters that maintain the state. For example, PHP uses PHPSESSID and jsp uses jsessionid.

Filers:

Websites handle very sensitive data like credit card numbers, SSN numbers etc which are not supposed to be viewed by anyone including the Pentester. So with this option you can search and replace those values so that the data cannot be viewed by any person. You can filter HTTP request content and also HTTP response content.

Cookies/Headers:

Here you can include 'referrer' and 'host' in HTTP header requests and also you can add custom headers and custom cookies to the requests sent by the WebInspect.

Proxy:

Under this tab you need to specify the proxy setting in case you are using one. If you are not using any proxy you can select 'direct connection'. If you are using a PAC (Proxy Automatic Configuration) file you can specify that or you can select to import the proxy settings from your browser.

Authentication:

This section is important as it deals with the authentication part of your application. During a scan WebInspect might encounter a situation where it has to authenticate before proceeding to next page. In such situations, depending on the details you provide in this tab it will handle the situation. From web application point of view, passwords and digital signatures are most used forms of authentication. You can specify if your scan requires any of the following:

- Network Authentication
- Client Certificates
- Client Certificates for tools

Login Macro & Startup Macro:

Macro is used to reply or playback the sequence of steps that you have recorded. So you need to record the authentication steps required to login to the application. In case the WebInspect unknowingly hits the logout button or any other button that logs it out of the web application, it can use the macro to relogin into the application. You can record a macro using the Web Macro Recorder tool of WebInspect, store it and later under this section you can browser for it and upload the same. Let's see the difference between login macro and startup macro.

Login Macro: If the authentication part contains simple login page containing username and password, you can use this so that WebInspect can use to login back into the application.

Startup Macro: If you scan is targeting a particular part of application or if you cannot determine the logout signature of the application you can record a startup macro.

File not found:

Select this option to find file not found response from server and also you can specify which responses should not be treated as file not found response.

Policy:

Depending on the policy selected WebInspect will scan for the vulnerabilities. A policy details the kind of vulnerabilities that WebInspect has to look for. For example, if OWASP TOP 10 policy is selected, WebInspect will look only for the owasp top 10 vulnerabilities. WebInspect by default includes some standard policies like OWASP 2010 etc. You can also create your own custom policy by stating the vulnerabilities you want.

Crawl Settings:

As discussed earlier crawler is something which traverses through the hierarchical structure of the site and constructs the tree structure. So here you can find the options which instruct the WebInspect about how to crawl that content.

Link Parsing:

Hyperlinks are usually defined by either HTML or JavaScript. But there might be some protocols which use different way of specifying hyperlinks. To accommodate this you can use custom links feature under this.

Session Exclusions:

You can specify the areas that need to be excluded from the crawl here.

Audit Settings:

Under this you have options which control the way in which audit will be conducted.

Session Exclusions:

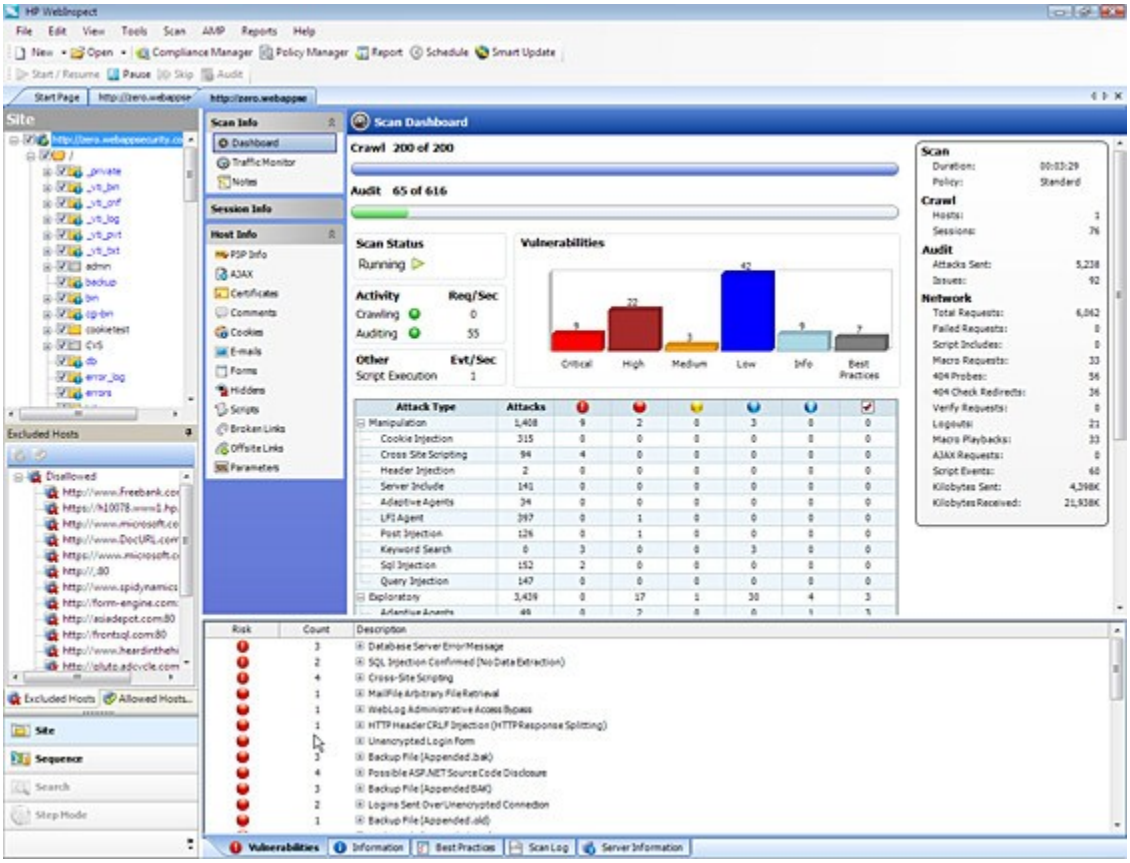
You can specify the areas that need to be excluded from the audit here.

Attack Exclusions:

You can manually enter the Parameters, cookies and headers which need to be excluded from the audit.

With this we are done with the configuring part. Click on next button in every window from here on and finally click on scan. WebInspect will now start scanning for vulnerabilities and will present you with the

issues and we are left with analyzing part now. After the scan gets completed, WebInspect will present you with the below screen.



This screen can be divided into 3 panes: Navigation Pane, Information Pane and Summary Pane.

<u>Navigation Pane:</u>

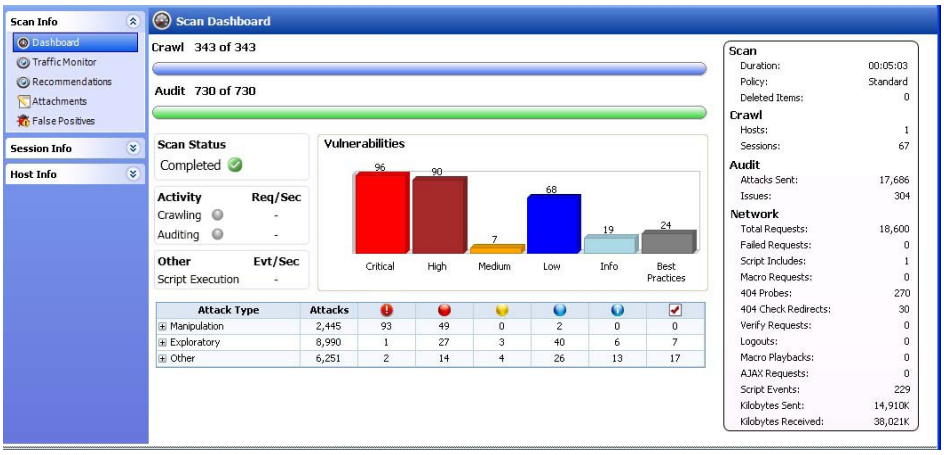
There are 4 views under this:

- *Site view*: This view shows the hierarchical structure of the website and also highlights those sessions in which vulnerability is found.
- *Sequence view*: Shows the order in which WebInspect traversed the tree structure i.e. the order in which WebInspect hit the URLs.
- *Search view*: Helps you to search for HTTP message components. For example you can search for all the sessions where cookies are set.
- *Step view*: After the scan completion, if you find that a particular URL is missing using step mode you can manually browse to that page and include it in the scan.

<u>Information Pane:</u>

This contains ‘scaninfo’ which contains information about the scan, ‘sessioninfo’ which contains information specific to selected session and ‘hostinfo’ which gives details about host.

‘Dashboard’ is an important link present under the scaninfo tab which presents you with the comprehensive view of the details of the scan. It is the summary of the scan results as shown below.



In the session info you can see the vulnerability type, HTTP request, HTTP response, browser view and many other options. You can explore by clicking on each one of them. ‘Hostinfo’ tab doesn’t contain much valuable information but you can find details about P3P info, certificates, cookies etc if you want to.

Summary Pane:

This is at the bottom of the window where you can access vulnerability information quickly by accessing one by one. Note that by clicking on vulnerability in summary pane corresponding session is automatically selected in Navigation pane. Then you can click on web browser under ‘sessionview’ to view it in browser or you can click http request to see the request headers etc. This is where you start analyzing them to eliminate the false positives. If you are satisfied that a particular finding reported by WebInspect is not a vulnerability right click on that and ‘ignore vulnerability’. If you wish to bundle them as false positives you can do the same. You can change the severity of the reported vulnerability too. You can also find server information and scan log information under this section. By proceeding in this manner we will be left with some vulnerabilities which have to be reported.

Reporting:

To generate a report select Report-->Generate Report and include the parameters that you want to and WebInspect also provides a description and fix for the identified vulnerabilities. You can generate report in desired format. This is the ‘Vulnerability Assessment Report’ generated by WebInspect.

Thus WebInspect stands out to be a wonderful tool for automating the vulnerability assessment of web applications.

About Me:

Rohit T is an Information Security Professional with 3 years of experience in Penetration testing & Vulnerability assessments of web applications.

Rohit's blog is located at http://webappsecure.blogspot.in/

Email: rorot33@gmail.com