**IBM Security AppScan**

**Introduction:**

**IBM Security AppScan**, previously known as **IBM Rational AppScan**, is a family of web security testing and monitoring tools from the [Rational Software](https://en.wikipedia.org/wiki/Rational_Software) division of [IBM](https://en.wikipedia.org/wiki/IBM). AppScan is intended to test Web applications for security vulnerabilities during the development process, when it is least expensive to fix such problems. The product learns the behavior of each application, whether an off-the-shelf application or internally developed, and develops a program intended to test all of its functions for both common and application-specific vulnerabilities.

IBM® Security AppScan® enhances web application security and mobile application security, improves application security program management and strengthens regulatory compliance. By scanning your web and mobile applications prior to deployment, AppScan enables you to identify security vulnerabilities and generate reports and fix recommendations.

 AppScan was integrated into the IBM rational software product line, adding tools aimed at helping developers to make security intrinsic to the application development lifecycle. To do so there are different edition for different web application and security management like source code, web application, web services, mobile security and many more. The Editions of appscan are as follows:

* AppScan Standard Edition
* AppScan Tester Edition
* AppScan Build Edition
* AppScan Enterprise Edition
* AppScan onDemand
* AppScan onDemand Production Site Monitoring
* AppScan Source Edition
* AppScan Reporting Console

AppScan is an application testing product, and can be used to scrutinize source code, websites, web applications and now mobile applications and services. It can come as a desktop product (AppScan Standard) or as a distributed environment with dedicated scanning servers (AppScan Enterprise).

**Working:**

## Installation:

To run Rational Appscan the system needs to have a minimum of 2 GB RAM. Also make sure to install .NET Framework and Adobe Flash player to execute flash content during scanning. Before we proceed further, it is worth noting that this automated scanner sends loads of data to the server while the scan is in progress. So it might delete files on the server, add new records or even bring the server down unintentionally. Thus, it is advisable to properly backup of all the data before you proceed with the scan.

Before you click the setup file, close any applications that are open. After clicking on the setup file, the installation wizard appears. If you have not installed .NET Framework version earlier, Appscan will now install the feature and asks for a restart. By following the wizard instructions the installation process can be completed pretty easily. If you are using a default license, you will be allowed to scan only the Appscan testing website. To scan your own site, you need to purchase one.

**Scanning:**

It functions in phases, firstly exploring the website as a user would, visiting pages and gathering information on parameters, cookies, links and pages.

**Explore:** In the explore stage Appscan tries to traverse through all the available links in the website and build a hierarchical structure. It sends requests and depending on the responses, it identifies the locations where there is a scope for vulnerability. For example, upon seeing a login page it would identify that there is a scope for authentication bypass through an sql injection. Note that it would only “identify” the test case but it would not perform any attack in this stage. In this way it sends several requests in this stage and builds the structure of the site while noting down the test cases.

**Test:**In the test stage, Appscan attacks the application to test for the vulnerabilities. The actual attack payloads are now unleashed to identify the security holes in the test cases that were built in the explore stage. It would also rank the severity of the risk.

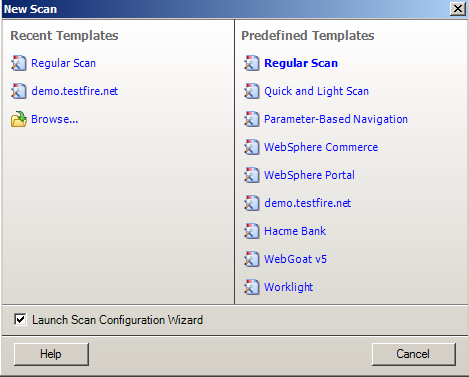
The test stage might reveal new links present in the site. So Appscan begins another round of scans after completing the explore and test stages and continues to do the same until there are no more links to be tested. Please note that the number of rounds of scanning is also configurable by the user in settings.

## Start a scan in Appscan :

To begin a scan, start Appscan and you’ll see the Welcomescreen.



Click on “Create New Scan” to start scanning a new web application.



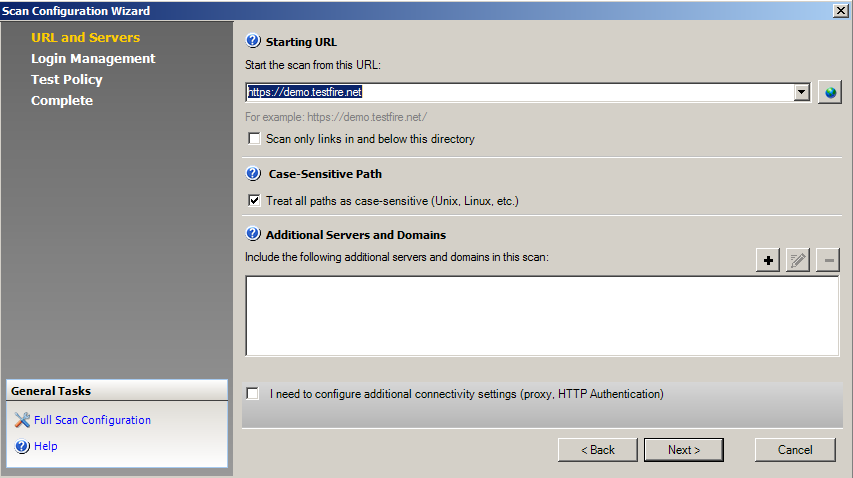
Select a scan template that suits your requirements. Templates consist of a scan configuration that is already defined. After selecting a template, the configuration wizard appears. It will ask you to select the type of scan. Select “AppScan” and click on next.

The scan configuration wizard is the core part of this tool. Using this we can let Appscan know what we are expecting from it. There are plenty of options available, and they have many choices among them.

**URL and Servers**

Starting URL: Under this feature specify the Starting URL of the scan. In most of the cases this would be the login page of the website. I have chosen <http://demo.testfire.net/> which is a demo site to test for web application vulnerabilities. If you want to limit the scan only to the links under this directory, select the check box.

Case Sensitive Path**:**  
If your server is case sensitive to URL’s, then select this option. Case-sensitivity of a server depends upon the underlying operating system. Linux/UNIX is case-sensitive, whereas Windows is not.



Additional Servers and Domains:

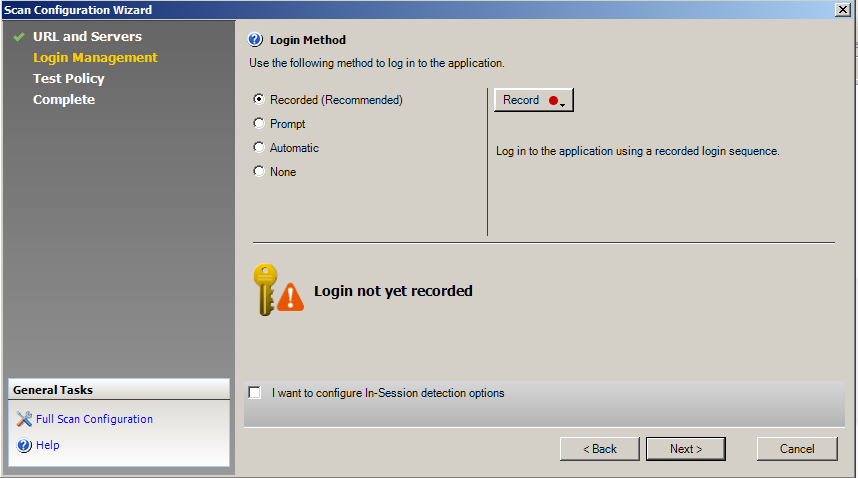
During the scan Appscan tries to crawl through all the links present in the site. When it discovers a link which is pointing to a different domain it will not attack the link unless it is specified under “Additional Servers and Domains” part. So by specifying a link under this tab, you are basically telling the Appscan that it’s OK to scan this link even though it’s not under the scan URL domain. Click on next button to proceed.

**Login Management:**

During the scan process, Appscan might accidentally hit a logout button or might hit a function that could logout the Appscan. So, to log in to the application we need to specify the process under this section.

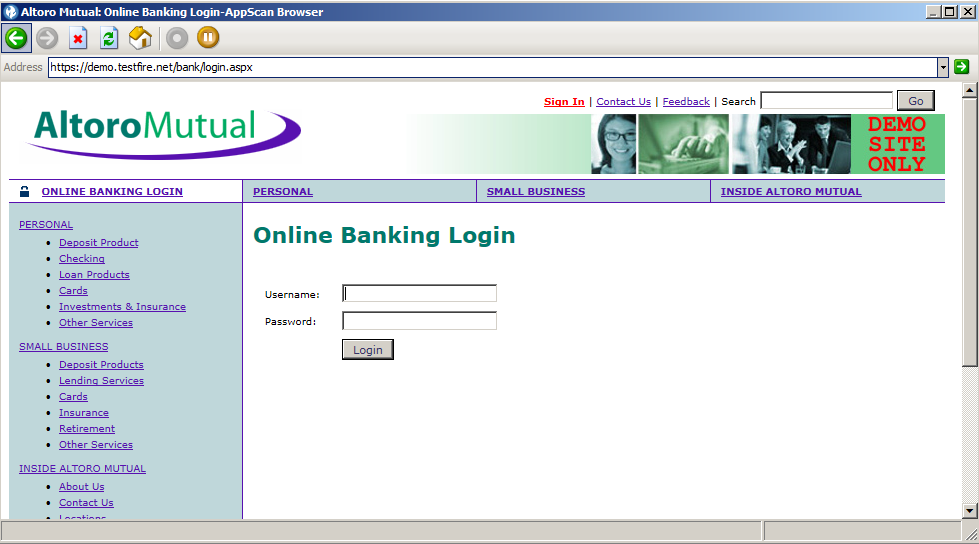
Recorded:

Upon selecting this option a new browser appears and tries to connect to the site specified as the start URL of this scan. You need to enter the credentials and log into the application. Once done, just close the browser. DO NOT click on the logout button as it defeats the whole purpose of going through this process. Also notice that new browser opened is not IE or Mozilla, but Appscan browser. You can change this browser option in Appscan under Tools à Options à Advanced, set the value of OpenIEBrowser to 0 -for the AppScan browser, 1 – for Internet Explorer, 2 – for Firefox, 3 – for Chrome. This is extremely useful in situations where the site behaves differently in each browser.



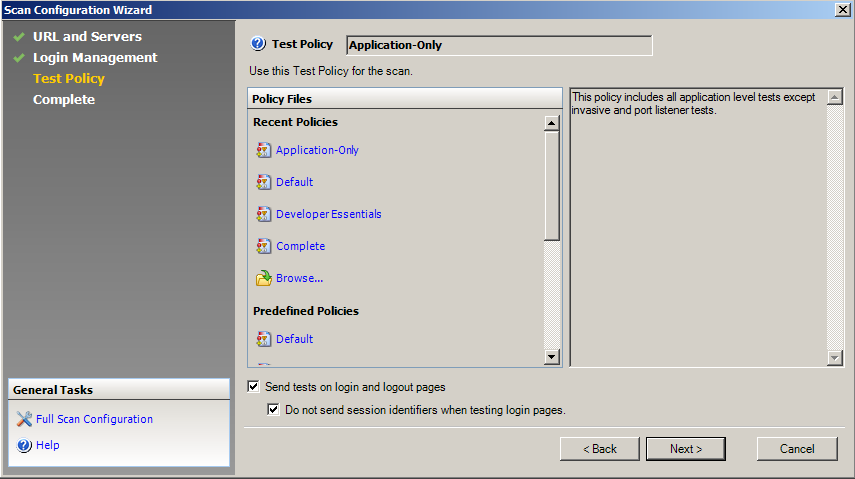
Prompt: Appscan prompts you to log into the application every time it logs out. Select this option only if you are planning to sit through the entire scan of your system. If your application implements CAPTCHA then this is one way to go ahead with the scan.

Automatic: Under this you can directly specify the user name and password that needs to be used to log into the application.

Click on next to continue.

**Test Policy:**

Under test policy you need to select the policy that most suits your requirements. The available policies are Default, Application-only, infrastructure-only, Invasive, Complete, the vital few, etc. out of which default policy is mostly used. If you do not want to send tests on login and logout pages, you can select that option here.



Click on next to continue.

**Complete:**

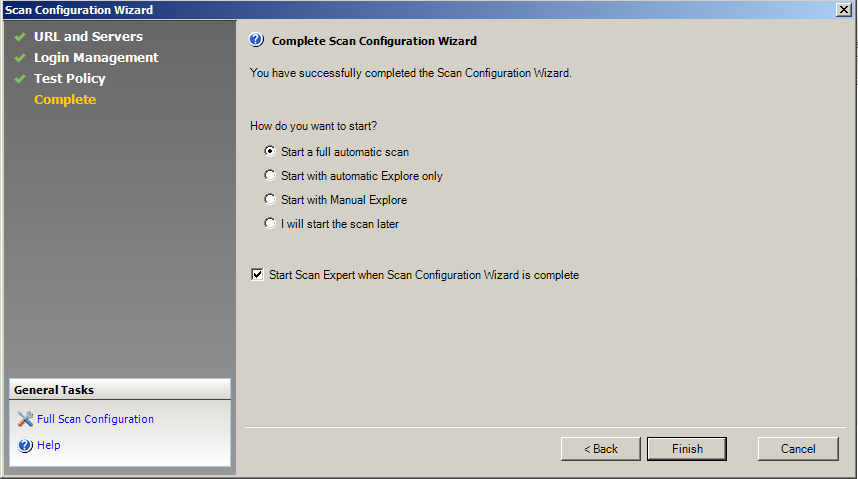
This is the final step in starting the scan. IBM Rational Appscan allows choosing the way you want to start the scan i.e. a full scan, explore only scan, etc.

Start a full automatic scan: With the configuration created earlier, Appscan would explore and then proceed to test stage as described earlier in this article.

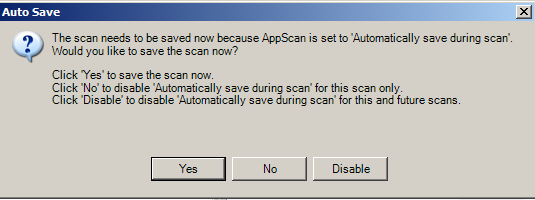
Start with automatic explore only: Appscan will only explore the application (i.e. crawl the application) but does not send any attacks.

Start with manual explore: A browser will be opened, and you can manually browse through the application.

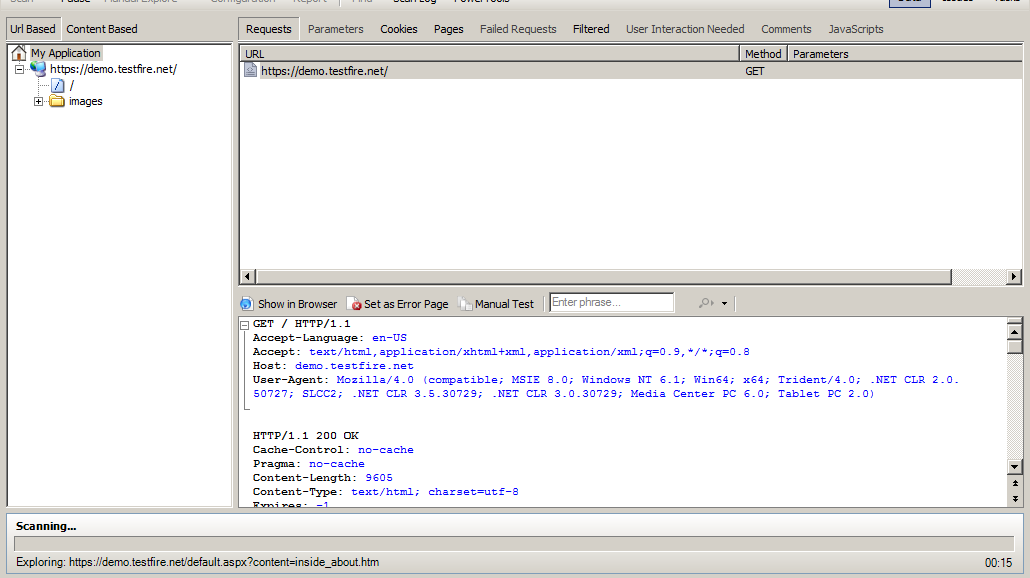
You can select the last option (i.e. I will start scan later) when you would like to make more changes to the scan configuration.



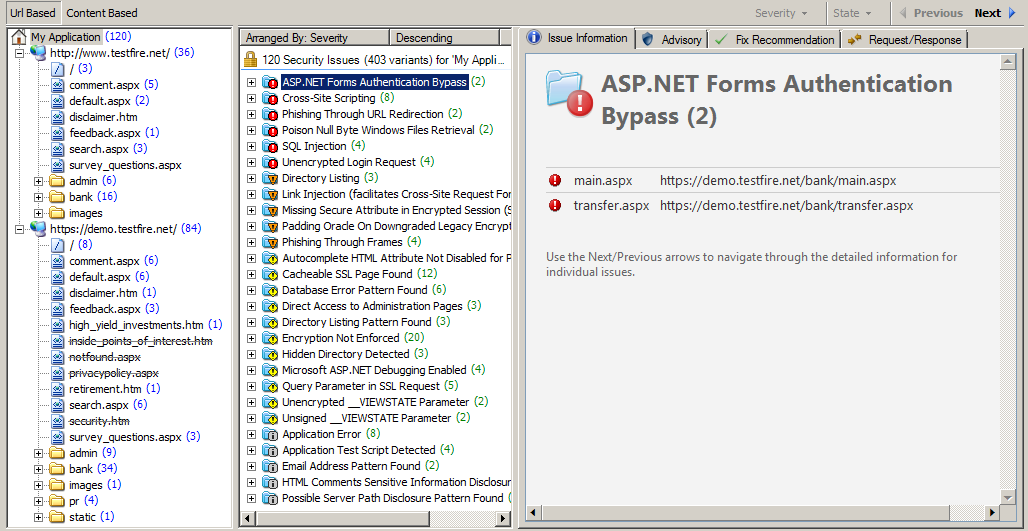
You can save your scan for future use.



And scan starts to scan the total website and explore all its components and pages.

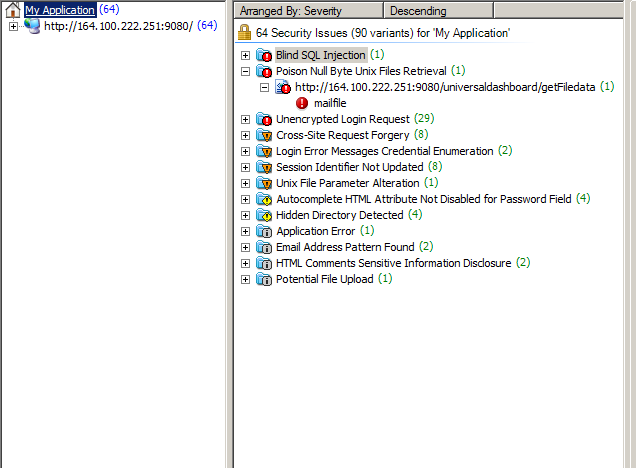


After the total scan completes you can view all the vulnerabilities to be sorted according to its criticality.



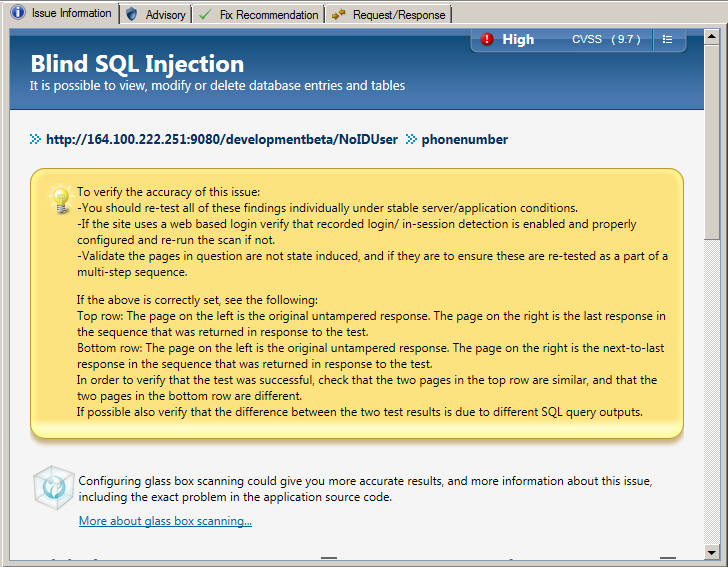
**Result:**

Once the first phase is completed, AppScan then begins its testing phase. This is where AppScan uses its extensive exploit database to test out potential exploits the target could be vulnerable to. After the scan is completed (which can take hours depending on what is being scanned), a compiled list of successful exploits is displayed.



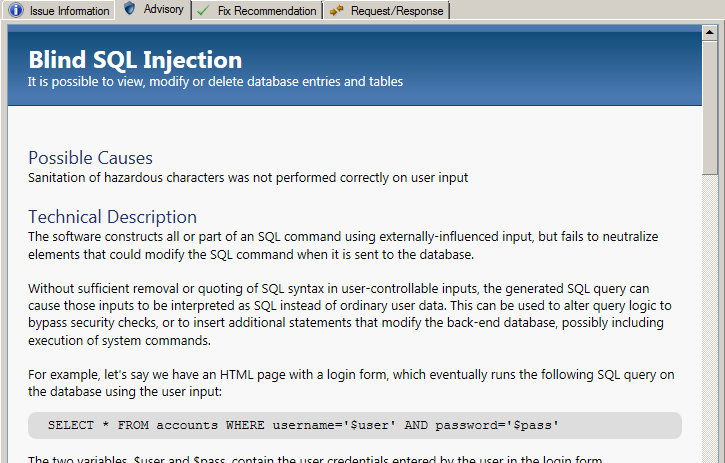
This is a typical response to a completed scan – AppScan lists all vulnerabilities that require remediation for the scanned site in a clear and concise list, which is by default, arranged by Severity. However, one of the determining features of AppScan is the additional information it supplies – which is divided amongst four tabs.

Firstly, the Issue information tab



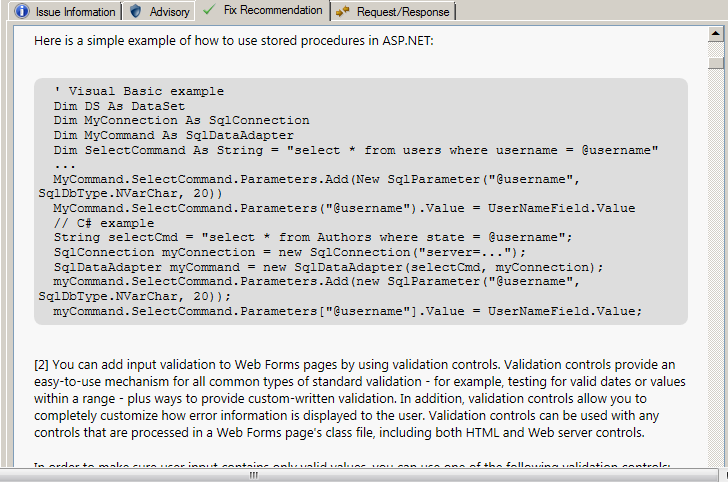
This tab provides a brief overview of the issue, displaying the original response and a short explanation of the exploit. A useful aspect of this tab is that it gives a rating and CVSS scoring, this gives anyone viewing the results a gauge as to how severe the exploit is.

The next tab is the advisory tab.



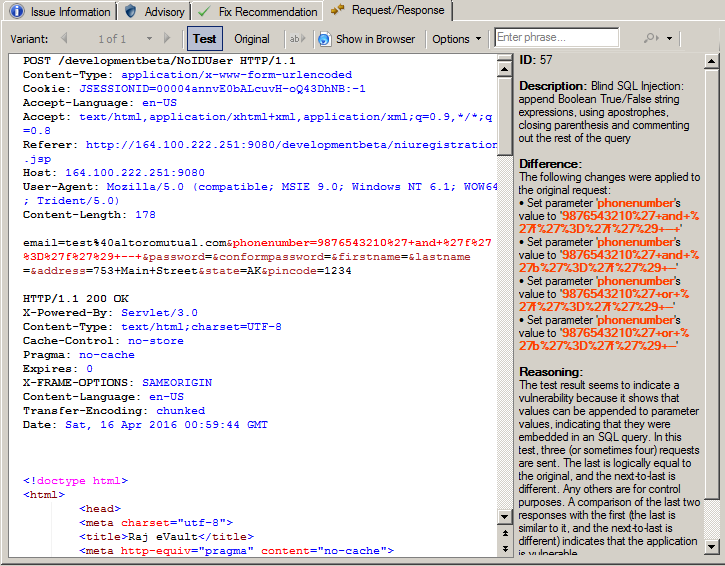
This tab provides a more detailed explanation of the exploit discovered. It provides: a technical description, the inclusion of which products were affected and –where possible- any references and relevant links. It is also noticible that in each tab there is a banner with a brief explanation of the exploit.      There is also a rational as to how the exploit was caused.

The next tab is the fix recommendations tab.



Rather self-explanatorily, the fix-recommendation tab provides recommendations to fix the exploit discovered. The level of information here varies depending on the discovered exploit. This tab is tailored towards a developer with an understanding of the product.

Finally, the Request/Response tab.



The request/response tab shows the code that AppScan sent to the application/site/server. The fields it includes are ID, description, difference, reasoning and –where possible- a CWE ID. This tab enables the developer to understand the code-level of the exploit, and provides additional information as to the functioning of the exploit executed.