Acunetix Web Vulnerability Scanner

Manual

V6.5

By Acunetix Ltd.

Acunetix Ltd.

http://www.acunetix.com

E-mail: info@acunetix.com

Document version 6.5

Last updated 27th July 2009.

Contents

1. Introduction to Acunetix Web Vulnerability Scanner

Why You Need To Secure Your Web Applications

Website security is possibly today's most overlooked aspect of securing the enterprise and should be a priority in any organization.

Increasingly, hackers are concentrating their efforts on web-based applications – shopping carts, forms, login pages, dynamic content, etc. Accessible 24/7 from anywhere in the world, insecure web applications provide easy access to backend corporate databases and also allow hackers to perform illegal activities using the attacked sites. A victim's website can be used to launch criminal activities such as hosting phishing sites or to transfer illicit content, while abusing the website's bandwidth and making its owner liable for these unlawful acts.

Hackers already have a wide repertoire of attacks that they regularly launch against organizations including SQL Injection, Cross Site Scripting, Directory Traversal Attacks, Parameter Manipulation (e.g., URL, Cookie, HTTP headers, web Forms), Authentication Attacks, Directory Enumeration and other exploits. Moreover, the hacker community is very close-knit; newly discovered web application intrusions are posted on a number of forums and websites known only to members of that exclusive group. These are called Zero Day exploits. Postings are updated daily and are used to propagate and facilitate further hacking.

Web applications – shopping carts, forms, login pages, dynamic content, and other bespoke applications – are designed to allow your website visitors to retrieve and submit dynamic content including varying levels of personal and sensitive data.

If these web applications are not secure, then your entire database of sensitive information is at serious risk. A Gartner Group study reveals that 75% of cyber attacks are done at the web application level.

Why does this happen?

- Websites and related web applications must be available 24 hours a day, 7 days a week to provide the required service to customers, employees, suppliers and other stakeholders.
- Firewalls and SSL provide no protection against web application hacking, simply because access to the website has to be made public.
- Web applications often have direct access to backend data such as customer databases and, hence, control valuable data and are much more difficult to secure.
- Corporate web applications have large amounts of bandwidth available. Since bandwidth is expensive, for a hacker to transfer huge amounts of illegal content, they revert to steal bandwidth from others.
- Most web applications are custom-made and, therefore, involve a lesser degree of testing than off-the-shelf software. Consequently, custom applications are more susceptible to attack.

Various high-profile hacking attacks have proven that web application security remains the most critical. If your web applications are compromised, hackers will have complete access to your backend data even though your firewall is configured correctly and your operating system and applications are patched repeatedly.

Network security defense provides no protection against web application attacks since these are launched on port 80 (default for websites) which has to remain open to allow regular operation of the business.

For the most comprehensive security strategy, it is therefore imperative that you regularly and consistently audit your web applications for exploitable vulnerabilities.

The need for automated web application security scanning

Manual vulnerability auditing of all your web applications is complex and time-consuming. It also demands a high-level of expertise and the ability to keep track of considerable volumes of code and of all the latest tricks of the hacker's 'trade'.

Automated vulnerability scanning allows you to focus on the more challenging issue of securing your web applications from any exploitable vulnerability that jeopardizes your data.

The Acunetix Web Vulnerability Scanner

The Acunetix Web Vulnerability Scanner (WVS) broadens the scope of vulnerability scanning by introducing highly advanced heuristic and rigorous technologies designed to tackle the complexities of today's web-based environments.

WVS is an automated web application security testing tool that audits your web applications by checking for vulnerabilities to SQL Injections, Cross site scripting and other exploitable hacking vulnerabilities. In general, the product scans any website or web application that is accessible via a web browser and that respects HTTP/HTTPS rules.

Besides automatically scanning for exploitable vulnerabilities, WVS offers a strong and unique solution for analyzing off-the-shelf and custom web applications including those relying on JavaScript (e.g., AJAX applications).

The Acunetix WVS is suitable for any small, medium sized and large organizations with intranets, extranets, and websites aimed at exchanging and/or delivering information with/to customers, vendors, employees and other stakeholders.

How WVS Works

Acunetix WVS has a vast array of automated features and manual tools and, in general, the automated scan works in the following manner:

1. The crawler crawls the entire website by following all the links on the site and in the robots.txt file and sitemap.xml (if available). WVS will then map out the website structure and display detailed information about every file. If Acunetix **AcuSensor Technology** is enabled, the sensor will retrieve a listing of all the files present in the web application directory and adds the files not found by the crawler to the crawler output. Such files usually are not discovered by the crawler as they are not accessible from the web server, or not linked through the website or even hidden application files, such as web.config.

2. After the crawling process, WVS automatically launches a series of vulnerability attacks on each page found, in essence emulating a

hacker. Also, WVS analyzes each page for places where it can input data, and subsequently attempts all the different input combinations. This is the Automated Scan Stage. If the **AcuSensor Technology** is enabled, a series of vulnerability checks which cannot be done when using a typical black box application scanner are launched against the website. For **AcuSensor Technology** details please refer to page 43.

3. During the scan process, a port scan is run against the web server hosting the website (the port scanner can be switched off from the configuration settings). If open ports are found, Acunetix WVS will perform complex network security checks against the service running on that port.

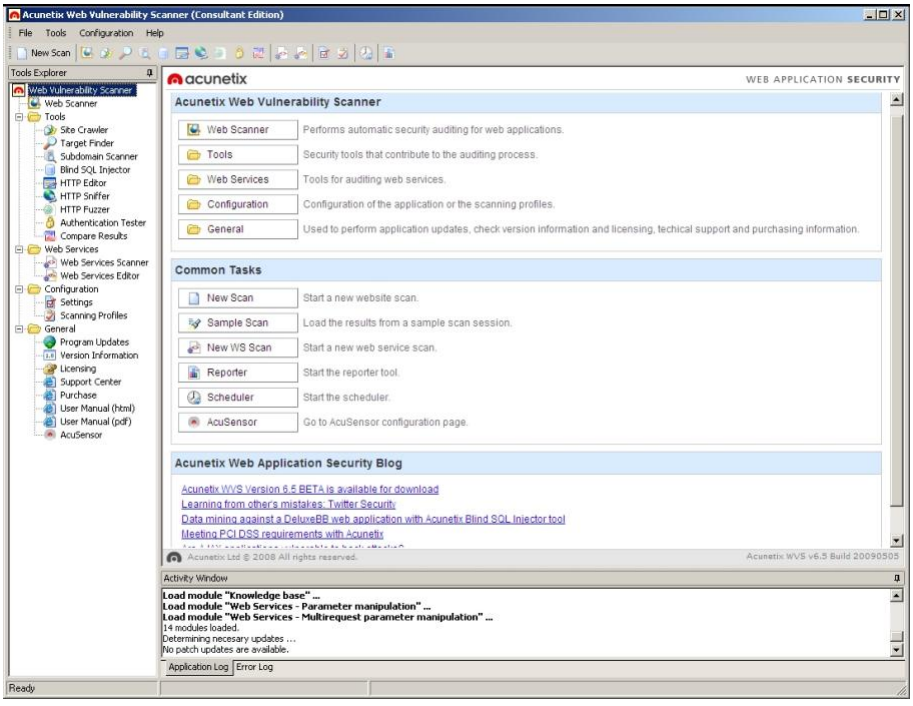
4. As vulnerabilities are found, Acunetix WVS reports these in the 'Alerts' node. Each alert contains information about the vulnerability such as POST variable name, affected item, http response of server and more. If **AcuSensor Technology** is used details such as source code line, stack trace, SQL query which lead to the vulnerability are listed. Recommendations on how to fix the vulnerability are also listed.

5. If open ports are found, they will be reported in the 'Knowledge Base' node. The list of open ports contains information such as the banner returned from the port and if a security test failed.

6. After a scan has been completed, it may be saved to file for later analysis and for comparison to previous scans. Using Acunetix reporter tool a professional report can be created summarizing the scan.

Acunetix WVS Program Overview

The following pages briefly explain the main WVS tools and features:



Screenshot 1 - Acunetix Web Vulnerability Scanner

Web Scanner

The Web Scanner is the most important component – it launches the security audit of a website. The scan consists of two phases:

1. Crawling – This discovery phase will automatically analyze the website and build a site structure.

2. Scanning – A vulnerability scan consists of a series of attacks launched against the crawled site structure, in effect, emulating a hacker.

The results of a scan are displayed in an Alert Node tree with details on all the vulnerabilities found within the website.

AcuSensor Technology

Acunetix AcuSensor Technology is a new security technology that allows you to identify more vulnerabilities than a traditional Web Application Scanner, whilst generating less false positives. In addition it indicates exactly where in your code the vulnerability is. The increased accuracy is achieved by combining black box scanning techniques with dynamic code analyzes while the source code is executed.

Port Scanner and Network Alerts

The Port Scanner and network alerts allow you to perform a port scan against the web server where the scanned website is running. When open ports are found, Acunetix WVS will perform complex network level security checks against the network service running on that port, such as DNS Open recursion tests, badly configured proxy server tests, weak SNMP community strings and many other network level security checks.

Target Finder

The Target Finder is a port scanner that allows you to locate open web server ports (port 80, 443) within a given range of IP addresses. If a web server is found to be running, the scanner will also display the response header of the server and the web server software. The port numbers to scan for are configurable.

Subdomain Scanner

The Subdomain scanner allows fast and easy identification of active Sub domains in a DNS zone using various techniques and guessing of common sub domain names. The Subdomain Scanner can be configured to use the target's DNS server, or one specified by the user for flexibility.

Blind SQL Injector

Ideal for penetration testers, the Blind SQL injector is an automated database data extraction tool where you can make manual tests to test further a website for SQL injections. The tool is also be able to enumerate databases, tables, dump data and also read specific files on the file system of the web server if a SQL injection is found.

HTTP Editor

The HTTP Editor allows you to create custom HTTP requests and debug HTTP requests and responses. It also includes an encoding and decoding tool to encode / decode text and URL's to MD5 hashes, UTF-7 formats and many other formats.

HTTP Sniffer

The HTTP Sniffer acts as a proxy and allows you to capture, examine and modify HTTP traffic between an HTTP client and a web server. You can also enable, add or edit traps to trap traffic before it is sent to the web server or back to the web client. This tool is useful to:

- Analyze how Session IDs are stored and how inputs are sent to the server.
- Alter any HTTP request being sent back to the server before it gets sent.
- Navigate through parts of the website which cannot be crawled automatically because, for example, of certain JavaScript code and import results to the scanner.

To use this tool, all http requests must pass through WVS thus the software must be set as the proxy server for your browser. You can read more about HTTP Sniffer and it's configuration on page 69 of this manual.

HTTP Fuzzer

The HTTP Fuzzer allows sophisticated testing for buffer overflows and input validation. With this tool you can easily create input rules for Acunetix WVS to test.

An example would be the following URL:

http://testphp.acunetix.com/listproducts.php?cat=1

Using the HTTP Fuzzer you can create a rule which would automatically replace the last part of the URL - '1' - with numbers between 1 and 999. Only valid results will be reported. This degree of automation allows you to quickly test the results of a 1000 queries while significantly reducing the amount of manual input.

Authentication Tester

With the Authentication Tester you can perform a dictionary attack on login pages which use HTTP (NTLM v1 and NTLM v2) or HTML form authentication. This tool uses two predefined text files (dictionaries) which contain a list of common usernames and passwords. These text files may be modified to include your own combinations.

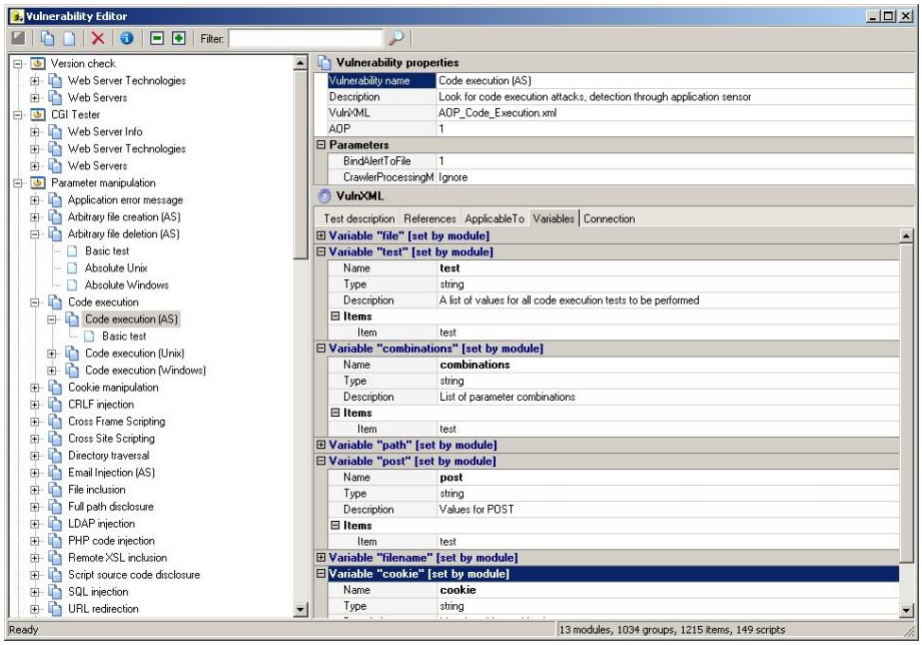
Web Services Scanner

The Web Services Scanner allows you to scan in an automated way for vulnerabilities in Web Services.

Web Services Editor

The Web Services Editor allows you to import an online or local WSDL for custom editing and execution of various web service operations over different port types for an in depth analyses of WSDL requests and responses. The editor also features syntax highlighting for all languages to easily edit SOAP headers and customize your own manual attacks.

Vulnerability Editor



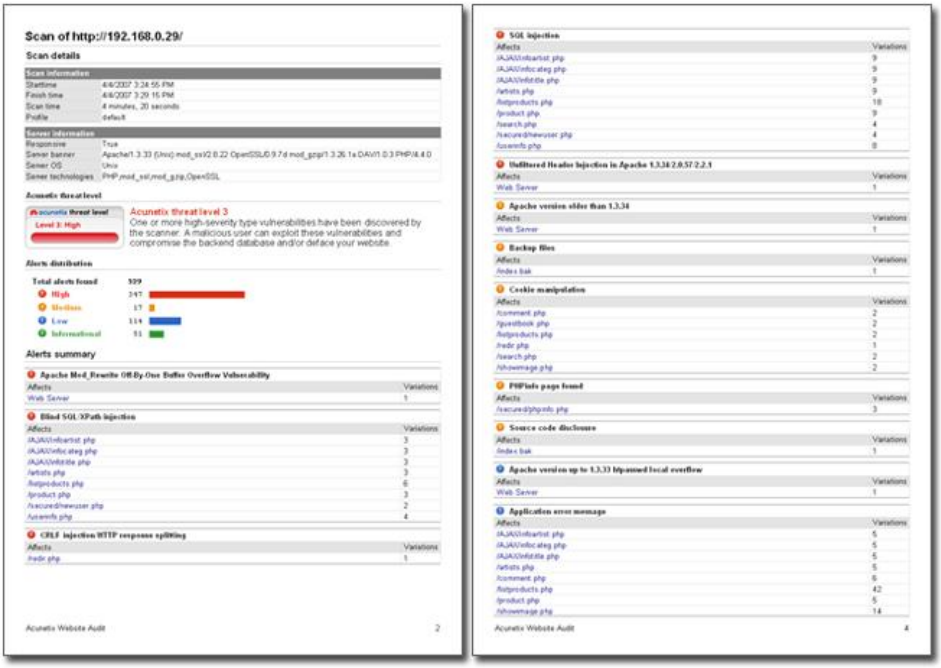
Screenshot 2 – The Vulnerability Editor

The Vulnerability Editor allows you to create custom security checks.

You will also notice changes and additions to the Vulnerability Editor as updates to the Acunetix WVS are installed. For more information on updating the Acunetix WVS please refer to page 31 of this manual.

Reporter

The Reporter allows you to generate reports of scan results in a printable format. Various report templates are available, including summary, detailed reports and also compliance reporting. The Consultant Version of the WVS allows further customization of the report headers.



Screenshot 3 - Typical WVS Report including Chart of alerts

What's new in Acunetix WVS Version 6.5

New features

- File upload forms vulnerability checks
- New Login Sequence recorder, supporting much more authentication forms and web technologies.
- Session Auto Recognition module; during crawling, if the session is invalidated or logged out, the scanner will automatically replay login sequence without the need for manual intervention.
- Actions drop down menu; for each selected node, the actions drop down menu is activated showing all possible functions to improve usability.
- CAPTCHA and single sign on support.
- New reporter supporting multi user scenario.

Major Improvements

- Improved cookie management and session handling to support modern dynamic websites.
- Port Scanner and Network Alerts results will appear in a separate node in the results tree.
- Users can import their settings from Version 6 to Version 6.5.
- Added Blind SQL Injection timing test using MySQL's sleep and MS SQL's waitfor functions.

License Scheme

Acunetix Web Vulnerability Scanner (WVS) is available in 3 versions: Small Business, Enterprise and Consultant.

Perpetual or Time Based Licenses

Acunetix WVS is sold as a one-year or perpetual license. The 1 year license expires 1 year from the date of activation. The perpetual license never expires.

The Enterprise and Consultant versions are available as both a one-year and perpetual license. The Small Business version is available as a perpetual license only.

Without the maintenance agreement, free support and upgrades are included for the first month from purchase.

To extend the period of support to one or more years, a maintenance agreement should be purchased. This entitles you to free upgrades and support for the duration of the agreement.

Small Business Version 1 Site/Server

The Small Business Version license allows you to install one copy of Acunetix WVS on one computer, and scan one nominated site; this site must be owned by yourself (or your company) and not by third parties. In the case of companies, you must obtain proper authorization to scan the website. Acunetix Small Business version will leave a trail in the log files of the scanned server and scanning of third party sites is prohibited with this license.

To scan multiple websites you would require the Enterprise unlimited license.

To install copies on several computers, you require purchasing the necessary individual licenses.

Enterprise Version Unlimited Sites/Servers

The Enterprise version license allows you to install one copy of Acunetix WVS on one computer, and scan an unlimited number of sites or servers. The sites or servers must be owned by yourself (or your company) and not by third parties. In the case of companies, you must obtain proper authorization to scan the website. Acunetix Enterprise version will leave a trail in the log files of the scanned server and scanning of third party sites is prohibited with this license.

To install copies on several computers, you are required to purchase the necessary individual licenses.

Consultant Version

The Consultant version license allows you to install one copy of Acunetix on one computer, and scan an unlimited number of sites or servers including 3rd party, provided that you have obtained permission from the respective site owners. This is the correct version to use if you are a consultant who provides web security testing services, or an ISP. The consultant edition also includes the capability of modifying the reports to include your own company logo. Furthermore this version does not leave any trail in the log files of the scanned server.

To install copies on several computers, you are required to purchase the necessary individual licenses.

Upgrading from an Evaluation to a Purchased Version

If you decide to purchase Acunetix WVS, you will need to upgrade the evaluation version to the purchased version. You will receive a new download location to obtain the unlocked and full version.

After download, simply launch the setup file. Setup will ask whether it can remove the evaluation version and install the full version. Any settings you have already made will be retained.

You will be able to enter the License key you received, after which you will install the full version and scan your website.

Extending a Purchased Version

If you have already installed the full version, but only want to extend the license key, you can enter your new license key in the 'Licensing' node under the 'General' section. Right-click on the General/Licensing Node, select 'License Product' and enter your new license key.

Limitations of Evaluation Version

The evaluation version of WVS, which is downloadable from the Acunetix main website, is practically identical to the full version in functionality and in the set of tools that it presents – with the following limitations:

- Websites will be scanned only for Cross Site Scripting (XSS) vulnerabilities: only the Acunetix test websites will be scanned for all types of vulnerabilities.
- Only the default report can be generated and it cannot be printed or exported.
- Scan Results cannot be saved.

Purchasing Acunetix WVS

To purchase any of these licenses please visit:

http://www.acunetix.com/ordering/ and contact one of the Channel Partners in your area. If there are no Channel Partners in your country, you may place your order online from http://www.acunetix.com/ordering/pricing.htm

Pricing is available at http://www.acunetix.com/ordering/pricing.htm

2. Installing Acunetix WVS

System Minimum Requirements

- Microsoft Windows XP Professional or Home Edition, Windows 2000, Windows Server 2003 and Windows Vista.
- 1 GB of RAM.
- 200 MB of available hard-disk space.
- Microsoft Internet Explorer 6 (or higher).
- Microsoft SQL Server / Access support – if reporting database is enabled (optional).

Installation Procedure

1. Download the latest version of Acunetix Web Vulnerability Scanner from the download location you were given when a license was purchased. Double click on the webvulnscan6.exe file to launch Acunetix WVS installation wizard and click 'Next'.

2. You will be asked to review and approve the License agreement and to enter your Name, Company Name and License key. If you are evaluating the product, leave the license key edit box blank.

3. Select the folder location where you want to install Acunetix Web Vulnerability Scanner, choose whether to install or not the Acunetix Firefox extension and also choose whether a program shortcut icon is to be created on the desktop.

4. Click Install to start the installation. Setup will now copy all files and install the necessary Windows Service. Click 'Finish' when ready and the Acunetix WVS main window will launch, unless the option is un-ticked.

Note: If using the evaluation version, you will only be able to scan one of the Acunetix test websites:

http://testphp.acunetix.com	-	A test website with PHP technology
http://testasp.acunetix.com	-	A test website with ASP technology
http://testaspnet.acunetix.com	-	A test website with ASP.NET technology

Furthermore, you will not be able to save the scan results.

Upgrade Procedure

Note: Acunetix Version 6.5 will replace your current Acunetix Version 6 install.

1. Double click on webvulnscan65.exe file to launch Acunetix WVS installation wizard. The installer automatically detects any previous builds of the same installed version and will display a dialog which gives you a choice if to continue or not.

2. Click on 'Yes' to proceed with the upgrade.

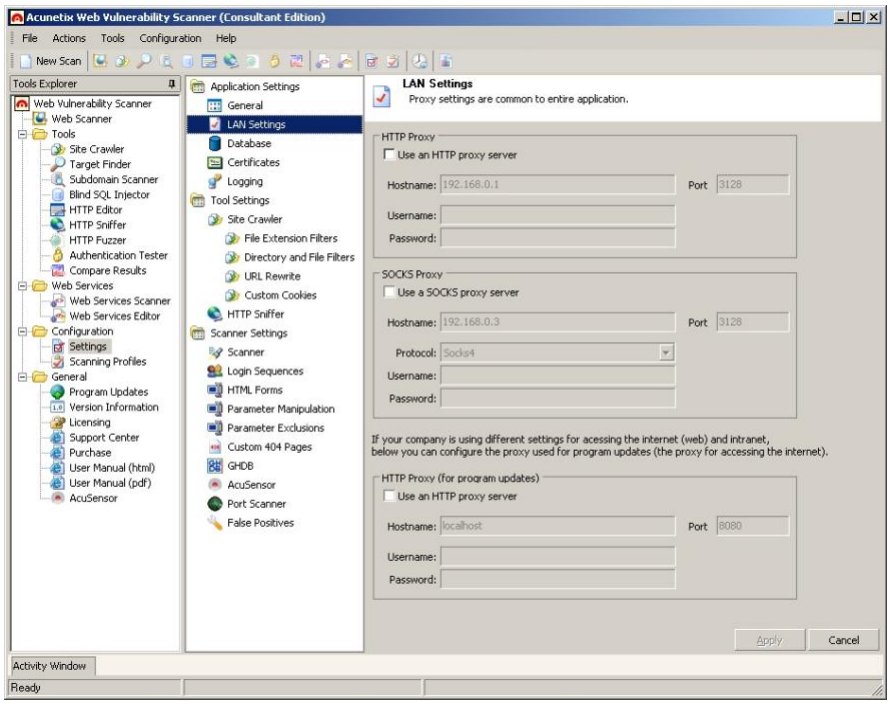
3. At this point the uninstaller is launched and it will verify again that you want to uninstall the previous version of Acunetix WVS. Click on 'Yes' to proceed with the upgrade.

4. To keep your past scan results and use them in the new version of Acunetix WVS, select 'No' when asked to remove the current database.

5. The rest of the installation procedure will be identical to a new installation from here onwards.

6. After the installation is finished, run Acunetix WVS. The application will present a dialog to upgrade any previous settings from the previous build or version that was installed. Click on 'Yes' to restore any previous configurations to the new version or build just installed.

Configuring a HTTP Proxy or SOCKS proxy Server



Screenshot 4 - LAN HTTP Proxy Settings

If your machine is sitting behind a proxy server, you need to set the Proxy server settings in Acunetix WVS.

From the Tools Explorer Panel select 'Settings' In the Configuration Section and then select 'LAN Settings' node under the Application Settings section to access the HTTP Proxy and SOCKS proxy settings page as shown in screenshot 4.

You can setup the Acunetix Web Vulnerability Scanner to use both technologies concurrently.

HTTP Proxy Settings

- **Use an HTTP proxy server** - Tick the check box to configure Acunetix WVS to use a HTTP proxy server.
- **Hostname and Port** - Hostname (or IP address) and port number of the HTTP proxy server.
- **Username and Password** - Credentials used to access the proxy. If no authentication is required, leave these options empty.

SOCKS Proxy Settings

- **Use a SOCKS proxy server** - Tick the check box to configure Acunetix WVS to use a SOCKS proxy server.

- **Hostname and Port** - Hostname (or IP address) and port number for the SOCKS proxy server.
- **Protocol** - Select which SOCKS protocol to use. Both Socks v4 or v5 protocols are supported by Acunetix WVS.
- **Username and Password** - The credentials used to access this proxy. If no authentication is required, leave these options empty.

HTTP Proxy Settings (For program updates)

If you are using different Proxy servers for accessing the internet (web) and intranet (internal network), in this section you can configure the Proxy server used for Acunetix WVS program updates, i.e. specify the Proxy server used for accessing the internet.

3. Getting Started: Scanning Your Website

Starting a Scan

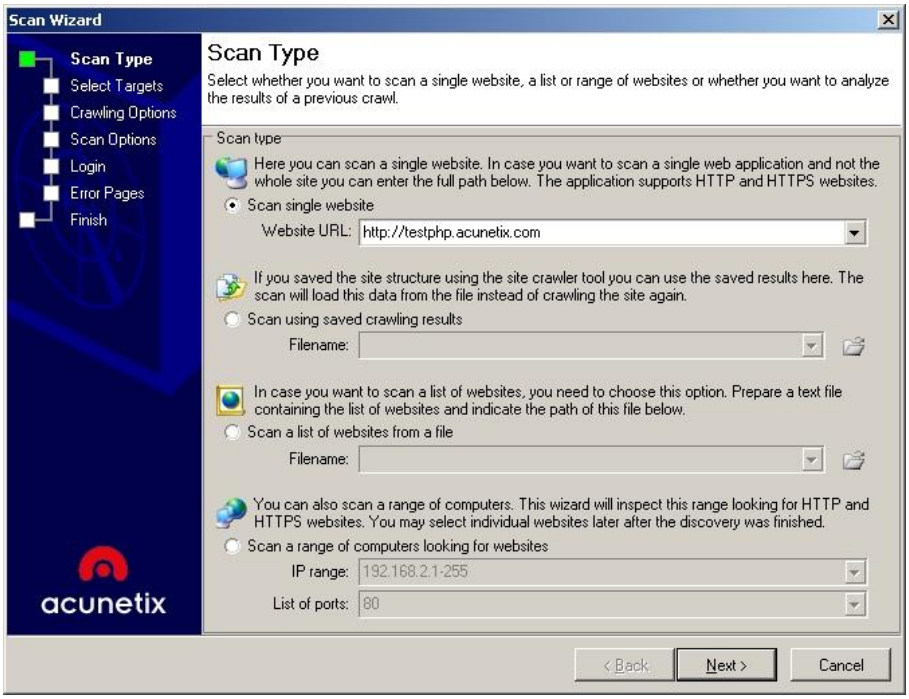
The Scan Wizard allows you to quickly set-up an automated crawl and scan of your website. An automated scan provides a comprehensive and deep understanding of the level website security by simply reviewing the individual alerts returned.

This chapter explains the process of launching a security audit of your website through the Scan wizard.

NOTE: DO NOT SCAN A WEBSITE WITHOUT PROPER AUTHORISATION! The web server logs will show the scans and any attacks made by Acunetix WVS. If you are not the sole administrator of the website please make sure to warn other administrators before performing a scan. Some scans might cause a website to crash requiring a restart of the website.

Step 1: Select Target(s) to Scan

1. Click on 'File > New > New Website Scan' to start the Scan Wizard or click on 'New Scan' button on the top right hand of the Acunetix WVS user interface.



Screenshot 5 – Scan Wizard Select Scan Type

2. Specify the target or targets to be scanned. The scan target options are:

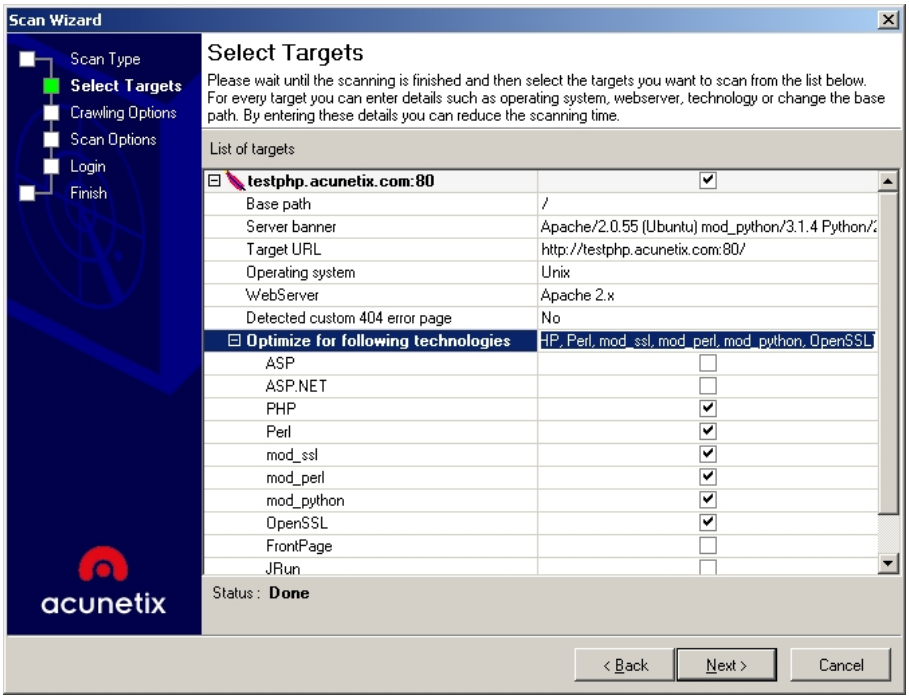
- **Scan single website** - Scans a single website. Enter a URL, e.g. http://testphp.acunetix.com, https://.testaspnet.acunetix.com.
- **Scan using saved crawling results** - If you previously performed a crawl on a website and saved the results, you can analyze these results

directly without having to crawl the site again. Specify the 'Saved crawler results' file by clicking on the folder button.

- **Scan List of Websites** - Scans a list of target websites specified in a plain text file (one target per line). Every target in the file is to be specified in the format <URL> or <URL:port> if the web server is listening on a non default port. The maximum number of websites Acunetix WVS can scan at 1 time is between 20 and 30 sites; depending on the size of the websites.
- **Scan Range of Computers** - This will scan a specific range of IP's (e.g. 192.168.0.10-192.168.0.200) for target sites which are open on the specified ports (Default 80, 81 and 443).

3. Click 'Next' to continue.

Step 2: Confirm Targets and Technologies Detected



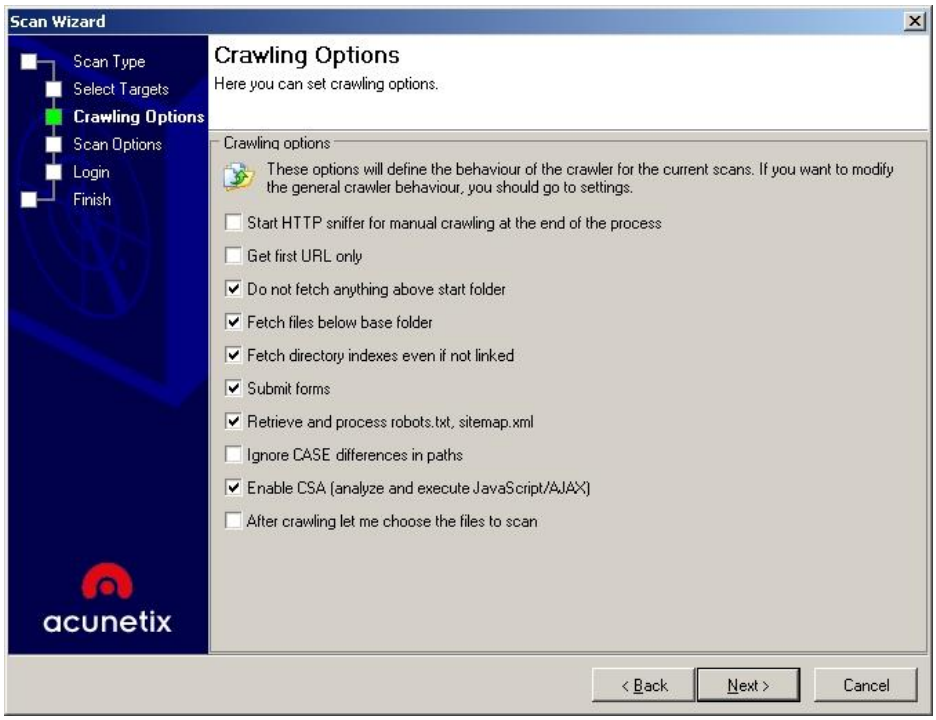
Screenshot 6 – Scan Wizard Selecting Targets and Technologies

Acunetix WVS will automatically probe the target website(s) for basic details such as operating system, web server, web server technologies and whether a custom error page is used (For more details on Custom Error Pages refer to page 26 of this manual).

The web vulnerability scanner will optimize the scan for the selected technologies and use these details to reduce the number of tests performed which are not applicable (e.g. Acunetix WVS will not probe IIS tests on a UNIX system). This will reduce scanning time.

Click on the relevant field and change the settings from the provided check boxes if you would like to add or remove scans for specific technologies.

Step 3: Specify Crawler Options



Screenshot 7 – Scan Wizard Crawling Options

In this dialog you can configure the crawling options.

Crawling Options

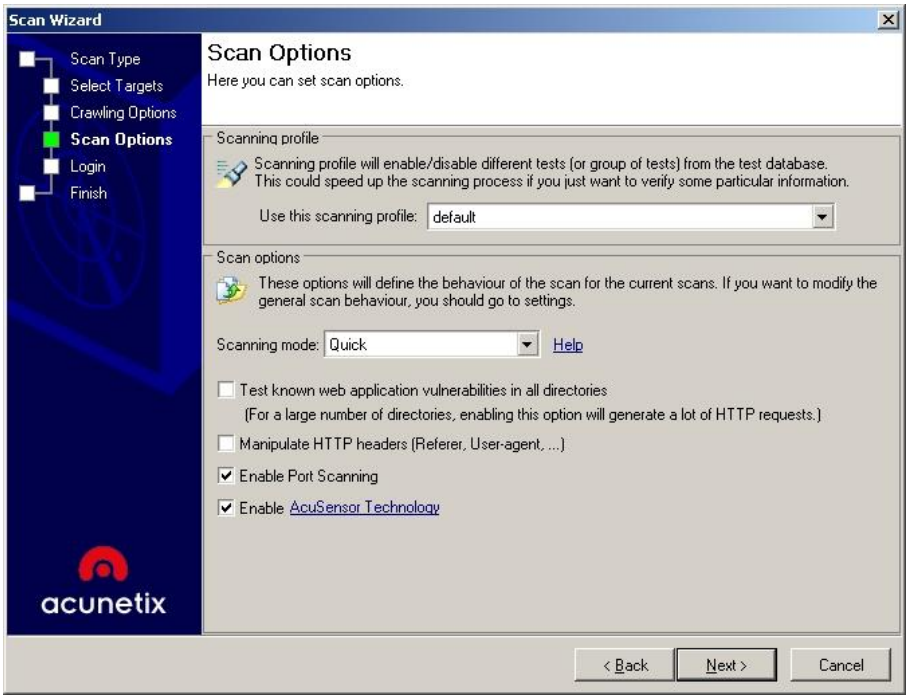
The Crawler traverses the entire website and identifies its structure. The following crawling options may be configured:

- Start HTTP Sniffer for manual crawling at the end of the scan process.
- Get first URL only.
- Do not fetch anything above start folder.
- Fetch files below base folder.
- Fetch directory indexes even if not linked.
- Submit forms.
- Retrieve and process robots.txt, sitemap.xml.
- Case insensitive paths.
- Analyze JavaScript.
- After crawling let me choose the files to scan

If the scan is from a saved crawl result, these options will be grayed out.

Note: Using default crawling options allows you to scan your websites without any problems. If you would like to tweak the crawling options, please refer to Configuring the Crawler section on page 51 in Chapter 10.

Step 4: Specify Scanning Profile Options and Mode



Screenshot 8 – Scan Profile and Mode Options

In this dialog you can configure the scanning profile and scan options, including the options for the scanning mode.

Scanning Profile

The **Scanning Profile** will determine which tests are to be carried out against the target site. For example, if you only want to test your website(s) for SQL injection, select the profile sql_injection and no additional tests would be performed.

Refer to the 'Scanning Profiles' section on page 41 for more information on how to customize existing profiles and create new scanning profiles.

Scan Options

From this section you can select the **Scanning Mode** which will be used during the scan. The scanning mode options are the following:

- **Quick** - In this mode the scanner will test for just the first value of every parameter.
- **Heuristic** - In this mode the scanner will try to automatically figure out for which parameters to test all values and for which not to test all values.
- **Extensive** - In this mode the scanner will test all possible combinations for all parameters on the website. In some cases, this can generate a huge number of requests and should be used with caution.

The other options which you can select from this step of the wizard are:

- **Test known web application vulnerabilities on every directory** - If this option is selected, the scanner will test for known web application vulnerabilities on every directory instead of the default directory only. This option will generate a lot of HTTP traffic and will extend the scanning time if the website being scanned is very large.

- **Manipulate HTTP headers** - With this option selected, the scanner will try to manipulate the HTTP headers which might be used by server side technologies.
- **Enable Port Scanning** - Tick this option to run the port scanner against the web server during a website scan. For more details about the Port Scanner refer to page 41 'Configuring the Port Scanner'.
- **Enable AcuSensor Technology** - Tick this option to enable AcuSensor Technology during the scan. Note that the AcuSensor client has to be installed on the web server which is being scanned. For more details about the AcuSensor Technology refer to page 43.

Note: If the scan is being launched from saved scan results, in the Enable AcuSensor Technology section you can specify to use sensor data from crawling results without revalidation, or to not use sensor data from crawling results or to revalidate sensor data from the drop down menu.

Step 5: Configure Login for Password Protected Areas

Acunetix supports 2 types of Login pages:

- **HTTP Authentication** - This type of authentication is handled by the web server, where the user is prompted with a password dialog.
- **Forms Authentication** - This type of authentication is handled via a web form. The credentials are sent back to the server for validation by a custom script.

Screenshot 9 - Login Details Options

Scanning a HTTP Password protected area:

Tick the box 'Authenticate with this user name and password combination' and enter the username and password.

Scanning a Forms Password protected area:

1. Click on 'Record new login sequence', browse to the web forms login page, enter a username and password and authenticate. Click on 'End login sequence' button.

2. Once authenticated, you also need to identify the logout link so the crawler will not crawl that link and logs out the user session. Click on 'Select Restricted Links' and click on the link which logs out the user session. Click again on 'Select Restricted Links' to terminate the session recording.

3. Click on exit and Save the Login Sequence. You can reuse the login sequence during future scans. Login sequences can be edited from the Tools Explorer by selecting 'Configuration > Scanner Settings > Login sequences' node in the Settings Interface.

Note: The Login Sequence Recorder can also be used to configure the crawler to crawl a web application in a pre-defined manner, such as a shopping cart. To configure the crawler to crawl a web application in a pre-defined manner, crawl the web application in the second step of the wizard 'Record Login Actions', and do not configure 'In-session' details in the fourth step of the wizard. For further information about creating and editing Login Sequences or pre-defined crawling, refer to page 34.

Step 6: Configure Custom 404 Error Pages

A 404 error page is the page which appears when a requested page is not found. In many cases, rather than displaying the standard error 404, many websites show a page formatted according to the look and feel of the website to inform the user that the page requested does not exist. Custom 404 error pages do not necessarily represent a server 404 error (Page not found), and therefore Acunetix WVS must be able to automatically identify these pages to detect the difference between a non existing URL and a valid web page.

The scan wizard will automatically try to detect whether the site uses custom error pages. If it does, WVS will display the custom error page and will automatically attempt to locate the unique identifier of such an error page; in this case Error 404: Page Not Found. If it does not detect custom 404 error pages but the site uses them, then they have to be configured manually.

Note: Typically, most of the websites return 404 errors when a requested URL is not found. If you need to configure a Custom 404 Error page, refer to page 39 in this manual, Custom 404 Pages settings.

Step 7: Select the Files and directories to Scan

If the option 'After crawling let me choose the files to scan' was ticked in the crawling options, a window with the site structure will open up, from which a selection of files to scan and ones to ignore can be made after the crawl process is finished.

Step 8: Completing the scan

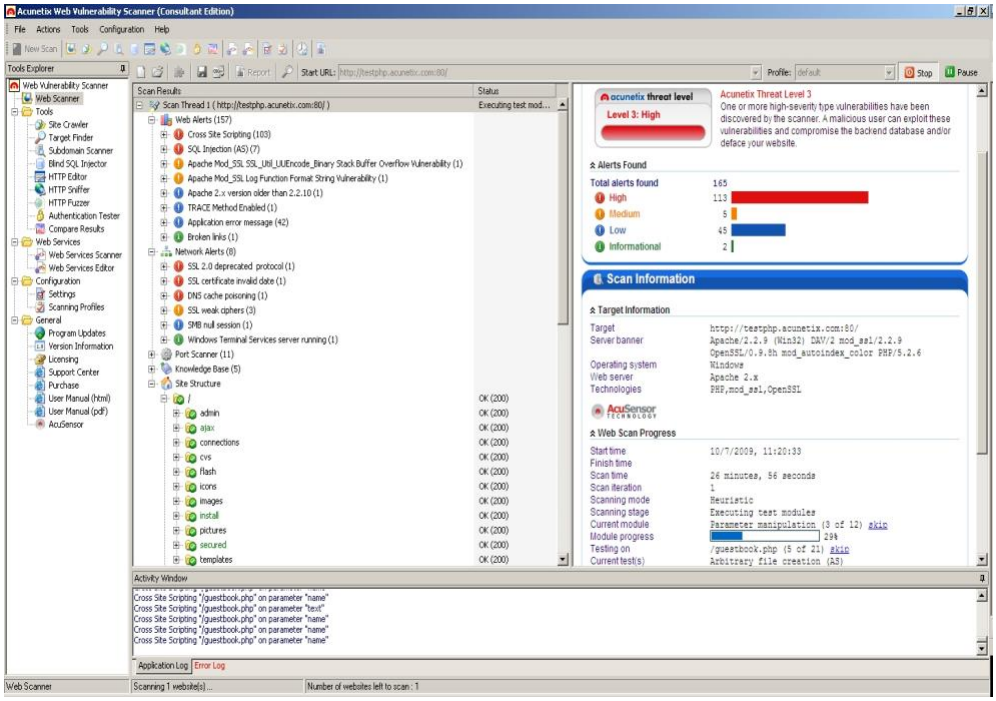
If you want to save the scan results to a database, enable 'Save scan results to the database for report generation'. Click on the 'Finish' button to start the scan.

Now click 'Finish' to start the scan. Depending on the size of the website a scan may take several hours!

4. Analyzing the Scan Results

Introduction

During the scan, a list of alerts which are found while scanning the website starts being populated. The list of alerts can be seen in the Alerts node in the 'Scan Results' window. A node 'Site Structure' is also populated with a list of files and folders discovered in the website.



Screenshot 10 - Scan Result and Information window

Web Alerts node

The Web Alerts node displays all vulnerabilities found on the target web application. Web Alerts are sorted into four severity levels:

Severity HIGH	**High Risk Alert Level 3** – Vulnerabilities categorized as the most dangerous, which put a site at maximum risk for hacking and data theft.
Severity MEDIUM	**Medium Risk Alert Level 2** – Vulnerabilities caused by server miss-configuration and site-coding flaws, which facilitate server disruption and directory intrusion.
Severity LOW	**Low Risk Alert Level 1** – Vulnerabilities derived from lack of encryption for data traffic, or directory path disclosures.
Severity INFO	**Informational Alert** – Sites which are susceptible to revealing information through GHDB search strings, or email addresses disclosure.

The number of vulnerabilities detected is displayed in brackets () next to the alert categories. If (AS) is displayed next to a vulnerability group, it means

that this vulnerability was reported from AcuSensor Technology. By clicking on an alert category node more information will be shown:

- **Vulnerability description** - A description of the current vulnerability.
- **Affected items** - The list of files which are vulnerable to the reported vulnerability.
- **The impact of this vulnerability** - What impact can this vulnerability have on the website or web server if exploited.
- **Attack details** - Details about the parameters and variables use to test for this vulnerability. E.g. For a Cross Site Scripting alert, the name of the effected variable and the string it was set to will be displayed. In this node, the HTTP headers sent to the web server and the response sent back from the web server are also displayed. The attack can be inspected and re-launched manually by clicking on 'Launch the attack with HTTP Editor". For more information, please refer to the 'HTTP Editor' chapter.
- **How to fix this vulnerability** - This section provides a recommendation on how the problem can be fixed.
- **Detailed information** - This section provides extensive detailed information about the vulnerability in question.
- **Web references** - A list of web links from where more information could be gathered about the current vulnerability and how to fix it.

Note: A vulnerability can be marked as 'False Positive' by clicking on 'Mark this alert as False Positive' under Attack Details in the vulnerability description. A vulnerability can be removed from the False Positive list by navigating to 'Configuration > Settings' node in the Tools Explorer and 'Scanner Settings > False Positives' node.

Network Alerts Node

The Network Alerts node displays all vulnerabilities found in scanned network services, such as DNS, FTP and SSH servers. These security checks can be switched off by un-ticking the option 'Enable Port Scanning' in the scan wizard. These alerts are generated when the web server is port scanned and advanced security checks are launched against the network services discovered. Network alerts are sorted into four severity levels, like alerts in Web Alerts node (refer to the paragraph above for reference).

The number of vulnerabilities detected is displayed in brackets () next to the alert categories. By clicking on an alert category node more information will be shown:

- **Vulnerability description** - A description of the current vulnerability.
- **Affected items** - The network service which is vulnerable to the reported vulnerability.
- **The impact of this vulnerability** - What impact can this vulnerability have on the website or web server if exploited.
- **How to fix this vulnerability** - This section provides a recommendation on how the problem can be fixed.
- **Detailed information** - This section provides detailed information about the vulnerability in question.
- **Web references** - A list of web links from where more information could be gathered about the current vulnerability and how to fix it.

Port Scanner Node

The Port Scanner node displays all the open ports found on the scanned server. Port Scanning of the target server can be switched off by un-ticking the option 'Enable Port Scanning' in the scan wizard. By clicking on a reported open port, the port banner is displayed.

Knowledge Base Node

The knowledge base node displays the following items:

- List of open TCP ports found on the server, including the port banner.
- List of Network Services running on the web server and their response.
- List of files with inputs found on the website. It also lists how many inputs each file has.
- List of links to external hosts found on the website. E.g. testphp.acunetix.com is being scanned and a link to www.acunetix.com is found on the website.
- List of uncommon HTTP responses and the HTTP requests that generated such response, such as Server Internal Error – HTTP 500.

Site Structure Node

The Site Structure Node displays the layout of the target site including all files and directories discovered during the crawling process. For every item retrieved more detailed information is available in the right information pane.

Summary information for a selected file or directory includes:

- **Info** - Generic information about the selected file such as file name, page title, path, URL etc.
- **Referrers** - A list of other files on the website from were Acunetix crawled to the selected file.
- **HTTP Headers** - The HTTP request sent to the web server to retrieve the selected file and the response headers received from the web server when requesting the selected file.
- **Inputs** - A list of variables present in the selected file that accept input. A list of possible input values is also shown.
- **View Source** - The source HTML sent to the scanner when accessing the selected file.
- **View Page** - The page is displayed as it is shown in a web browser.
- **HTML Structure Analysis** - Specific HTML structural information such as IMG, LINK, OBJECT tags found in the HTML of the selected file.
- **Alerts** - Alerts found on the selected file.

Grouping of Test Variants

When more than a single instance of the same vulnerability is detected on any page, the scanner will group the variants of each exploit according to the parameter which was tested. This makes it easy to understand how many total exploits were detected, and also how many files were found to be vulnerable.

This organization of vulnerability data makes it easier to keep track of vulnerable pages and what vulnerabilities need to be fixed. Vulnerability data

can also be presented in a report with this system of grouping, by selecting the Vulnerability Report template in the reporting application.

Saving a Scan Result

When a scan is completed you can save the scan results to an external file for analysis and comparison at a later stage. The saved file will contain all the scans from the current session including alert information and site structure.

To save the scan results go to 'File' and 'Save Scan Results'.

To load the scan results go to 'File' and 'Load Scan Results'.

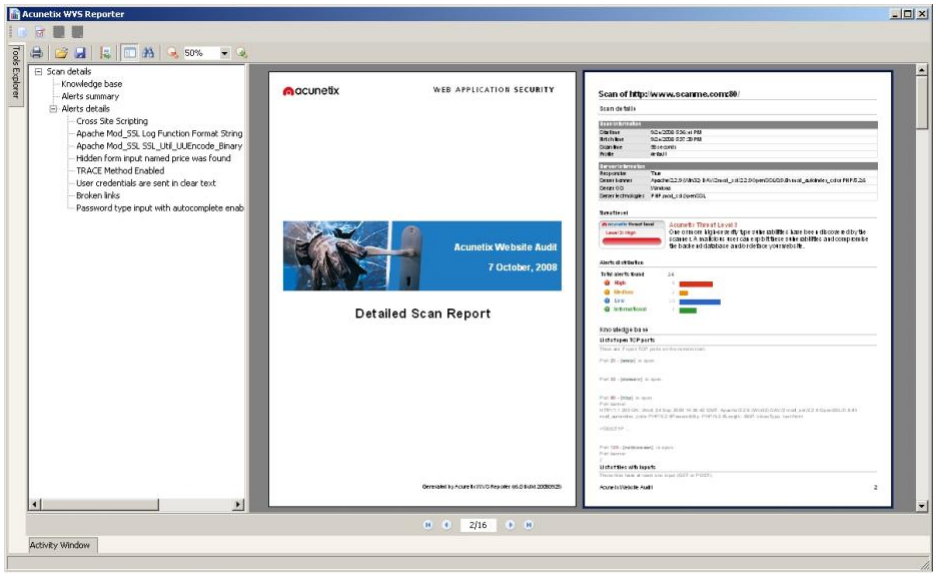
If the option 'Save scan results to database for report generation' is not ticked in the wizard, where by default it is ticked, no reporting details are saved from the scan. So to generate a report from a saved scan one has to import the scan details to the reporting database. To do so, right click 'Web Scanner' node and select 'Import scan results to database'.

Generating a Report from the Scan Results

To generate a report, click on the  **Report** button on the toolbar at the top. This will start the Acunetix WVS Reporter.

From here reports such as Developer Report, Executive Report, Compliance Report and many others can be generated through a wizard. Once the report is generated, it can be exported to various different formats, such as PDF, HTML and others supported formats.

More information on how to configure the default report, which is generated when clicking on the Report button, can be found on page 90 of this manual.



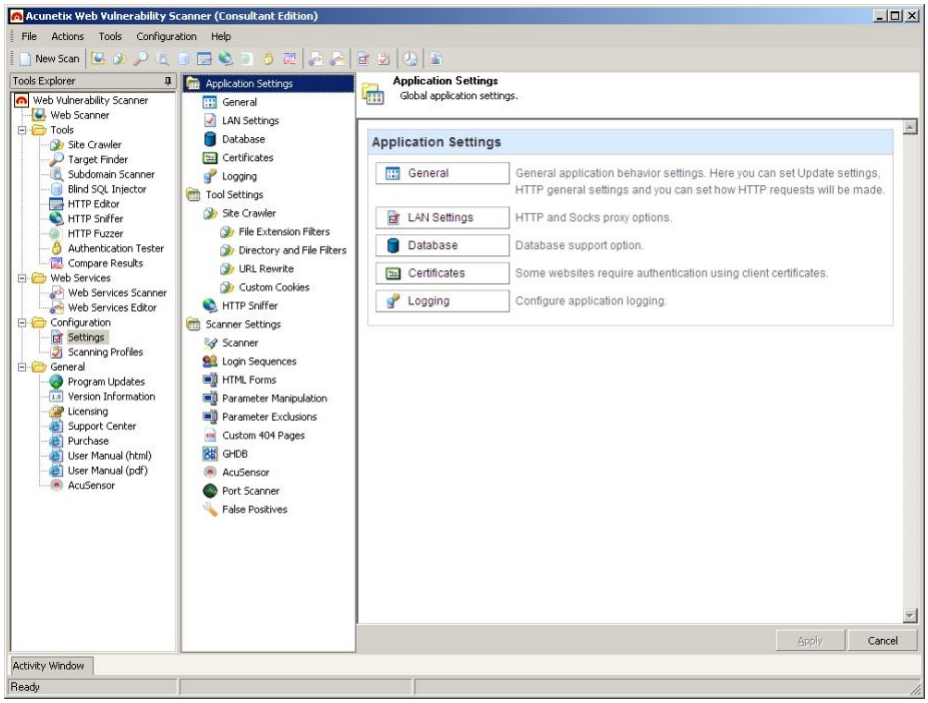
Screenshot 11 – Default Generated Report from Scan Results

Note: To generate a report, a database must be configured (either MDB or SQL). This can done from 'Configuration > Settings > Database' node.

5. Configuring Acunetix WVS

Introduction

The Acunetix WVS configuration settings can be accessed from the node 'Configuration > Settings' in the tools explorer.



Screenshot 12 - Configuration Settings

Application Settings

Settings > Application Settings > General

From this node, the General Application Settings can be configured.

Updates

- **Updates URL** - The location from where vulnerability and application updates can be downloaded.
- **Check for updates -** Specify when the application should check for new vulnerability and application updates.

HTTP General

- **User agent string** - Configure what user agent header string Acunetix should use when accessing a target website.
- **File size limit in kilobytes** - Maximum file size accepted by the crawler. Larger files than the specified size will not be crawled.
- **HTTP request timeout in seconds** - If no HTTP response is received after the specified interval, the request is cancelled.
- **Display custom HTTP status information** - Display the full HTTP status line header and the corresponding status string.

- **Display HTTPS status icon** – Enable this option to show a padlock icon next to files or directories which are accessed via HTTPS and not HTTP. HTTP Tuning

HTTP Tuning

- **Maximum number of parallel connections** - Specify the maximum number of any type of connection made to a target site at the same time, i.e. these are not only HTTP requests but can also be network connections etc. If overloaded with requests, some target servers might crash or reject new connections.
- **HTTP request queue execution frame** - Specify the maximum number of HTTP requests (a single HTTP request sent to a target website can be a number of smaller HTTP requests) generated by Acunetix WVS passed over to the HTTP protocol implementation as a single request to be sent to the target site. This helps to control memory and processor load and can be fine tuned to maximize server load.
- **Delay between consecutive requests group** - Delay between two execution queues (in milliseconds).
- **Default schemes** - Select predefined schemes of HTTP tuning settings for Internet or Local Intranet. If you select one of these schemes, the values for the previous settings will be overridden.

NOTE: To apply HTTP section changes, restart Acunetix WVS. Be careful when modifying HTTP Settings as it may cause the application to flood or crash the target server.

Memory Optimization

Enabling this option instructs WVS to store temporary data in the specified location instead of system memory. Acunetix WVS must have full access to this folder. This will greatly reduce overall memory usage.

Password Protection

In this section the user can set a password to restrict access to the WVS main application and all the WVS applications including the Reporter, Vulnerability Editor and Scheduler.

To create a new password, enter the password in the fields ‘New Password’ and ‘Confirm New Password’.

To remove password protection, enter the current password in the filed ‘Current Password’ and leave the other 2 fields empty.

Settings > Application Settings > LAN Settings

The LAN Settings options are explained in further details in the section ‘Configuring a HTTP Proxy or SOCKS proxy Server’ of Chapter 2, Installing Acunetix WVS on page 16.

Settings > Application Settings > Database

In this section the user can select a Microsoft Access or MS SQL Server database which will be used to save scan results for reporting purposes. Tick the option ‘Enable database support’ to configure the database settings.

MS Access database

To configure Acunetix WVS to use MS Access database from the ‘Database Type’ drop down menu select ‘MS Access’ and insert the path of the database in the ‘Database’ text field. If the database does not exist, it will be automatically created.

MS SQL Server database

To configure Acunetix WVS to use MS SQL Server database:

1. Select MS SQL Server from the 'Database Type' drop down menu.

2. Insert the Server IP of FQDN in the 'Server' text box and the credentials to connect to the server in the 'Username' and 'Password' text box.

3. Specify a database name in the 'Database' text box. If the database does not exist it will be automatically created. If the database specified already exists, the user is prompted with a confirmation to overwrite the current database structure and data.

Note: To create a new database a user with SQL Administrator privileges must be specified. If an existing database is specified, a user with Administrator privileges on the specified database ONLY is required.

Import Database Configuration

Click on 'Import Database Configuration' and select a dbconfig file generated from the Acunetix Enterprise Reporter to automatically import SQL database settings.

Settings > Application Settings > Certificates

Some websites require client certificates to identify a client before access is granted. These certificates may be configured into Acunetix WVS by specifying the URL to be used during a crawl or a scan.

Configuring Certificates

1. Specify a certificate location by browsing to the certificate by using the Browse icon next to the 'Certificate file' text box and enter the certificate password in the 'Password' text box.

2. Enter the URL for which this certificate should be used. Click on 'Import' and 'Apply' to save the certificate information.

Settings > Application Settings > Logging

This section provides configuration for enabling or disabling logging. It also provides different logging level settings by allowing the user to choose which message types should be logged or not.

Configuring the Scanner

The Scanner configuration settings can be accessed from 'Configuration > Settings > Scanner Settings' node. Here the default scanning options can be configured.

Settings > Scanner Settings > Scanner

- **Disable Alerts generated by crawler** - Select this option to not report broken links which expect user inputs and which naming indicate they can be dangerous, such as dangerous SSL versions alerts.
- **Test known web application vulnerabilities on every directory** - Select this option to launch known specific web applications vulnerabilities on every sub directory of the website not just in the root directory. This will increase the time to complete the scan.
- **Scanning mode** - Select default scanning mode from:
    - **Quick** - In this mode the scanner will test just the first value of every parameter.

- **Heuristic** - In this mode the scanner will try to automatically figure out for which parameters to test all values and for which not to test all values.
- **Extensive** - In this mode the scanner will test all possible combinations for all parameters on the website. In some cases, this can generate a huge number of requests and should be used with caution.

- **Limit crawl recursions to X iterations** - After a site is crawled and vulnerability testing has started, if the scanner finds out new objects, a new crawl is restarted. This is called iteration. Configure the maximum number of iterations that can happen while scanning a website.
- **Enable Port Scanning -** While scanning a website, if this option is enabled a port scan will be launched against the web server hosting the web site.
- **Collect uncommon HTTP Requests -** Select this option to configure Acunetix WVS report back any server response which is not common and which might include sensitive data, such as internal server errors. This is reported in the 'Knowledge Base' node in the Scan Results window.
- **Abort Scan if the server stops responding** - Configure the maximum number of network errors in a scan. If this number of errors is reached the scan is aborted.
- **List of hosts allowed -** By default, Acunetix WVS will not crawl links outside the target URL. However, some links on some websites link to external locations outside the target URL and may require being included in the scan. Configure Acunetix WVS to include and follow these links in the 'list of hosts allowed' field. Enter the host name or IP address of the domain to be included in a vulnerability scan and click the '+' button to add this entry to the list of hosts to be scanned. E.g. when scanning testphp.acunetix.com can be links which link to www.acunetix.com.

Note: Hostnames can be specified using wildcards e.g. '*.domain.com', which includes all websites with a suffix of .domain.com such as sales.domain.com. A question mark can also be used as a wildcard, e.g. 'host?.domain.com', would include all websites with one character added after 'host' such as host1.domain.com.

Settings > Scanner Settings > Login Sequences

The Login Sequence Recorder can be used to record a form based login sequence, to create a pre-defined crawling sequence, and also to mark pages that will require human interaction each time, such as pages with CAPTCHA, One-Time password, Two-Factor authentication etc.

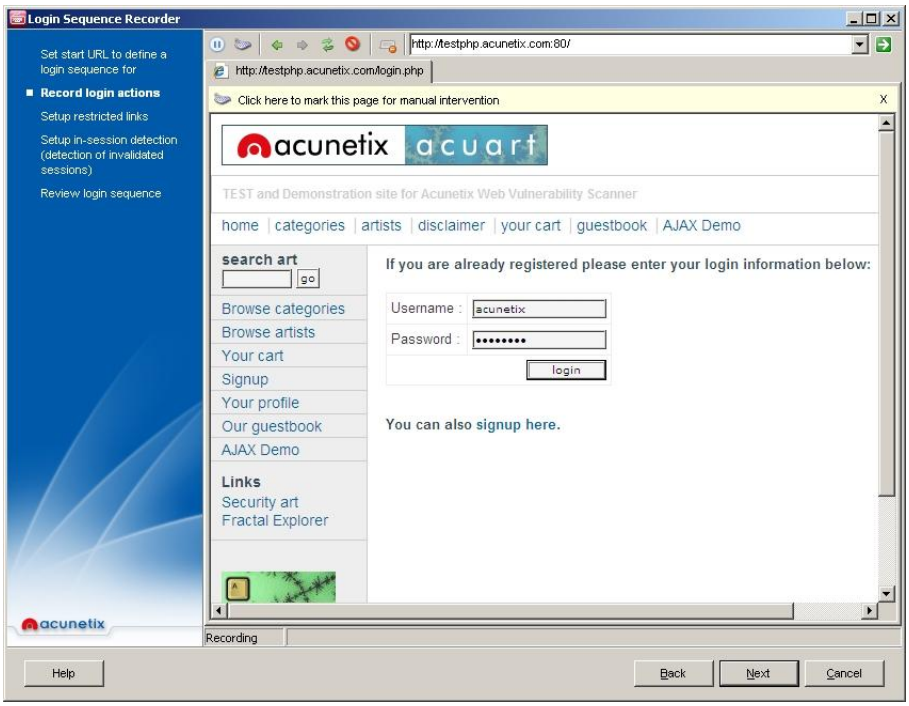
Forms authentication is not handled via HTTP, but via a web form which asks the user for a username and password. This information is sent back to the server for validation by a custom script.

In this configuration screen you can create or edit existing login sequences which are used by Acunetix WVS to access HTML authentication protected areas of a website. Login sequences allow WVS to replicate all events which are manually performed to access the area secured by the login page.

Creating a new login sequence

1. Click on the '  ' button to open up the Login Sequence Recorder. Enter the URL of the website and click on 'Next'. One can also click on 'Check URL' to confirm that the URL entered is reachable from Acunetix Login Sequence Recorder.

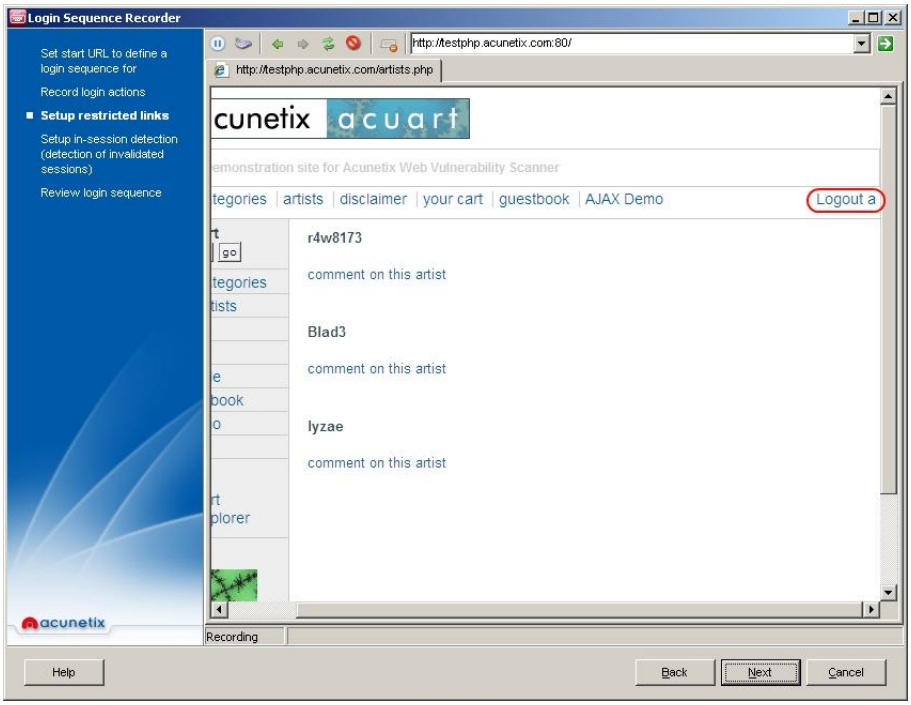
2. Navigate to the Login prompt and enter a valid username and password and authenticate.



Screenshot 13 – Login Sequence Recorder

3. Once authenticated, you also need to identify the logout link, otherwise Acunetix WVS will try to crawl the logout link and will logout of the password protected area. Click 'Next' to proceed with the login recording.

4. If the logout link is not in the same page, click on 'Pause' in the top menu. Navigate to the page where the logout link is, and click 'Pause' again to resume the recording of the excluded link.

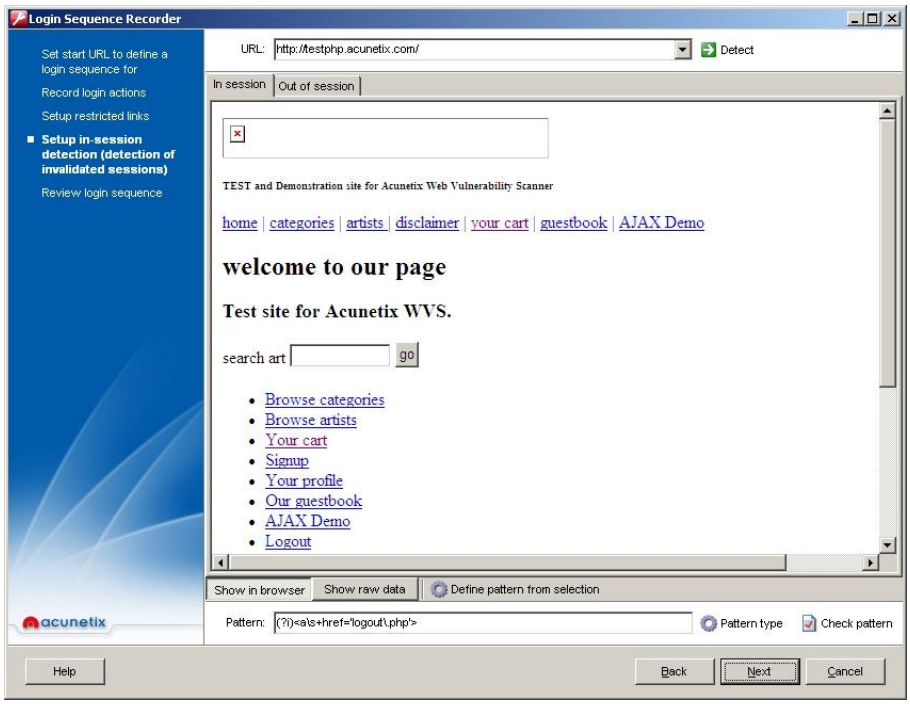


Screenshot 14 – Specify an excluded link

5. Click next to specify the 'In Session' or 'Out of Session' patterns. This is an important step to configure the crawler to automatically detect if during the crawl the session is still logged in or not. If the session for some reason expires during a crawl, the crawler will automatically re-login.

6. Click on 'Detect' to automatically try to detect a particular pattern for the crawler to recognize that it is logged in or not. If the automatic detection does not work, specify a pattern manually. The pattern can be a regular expression. One can also highlight specific content and click on 'Define pattern from selection' and a regular expression will be automatically generated.

7. Also specify where the pattern can be found from the 'Pattern Type' drop down menu. The options are 'In headers', 'Not in headers', 'In body', 'Not in body', 'Status code is' and 'Status code is not'. Click on 'Check Pattern' to verify that the crawler is able to recognize the difference between a logged in session and a logged out session.



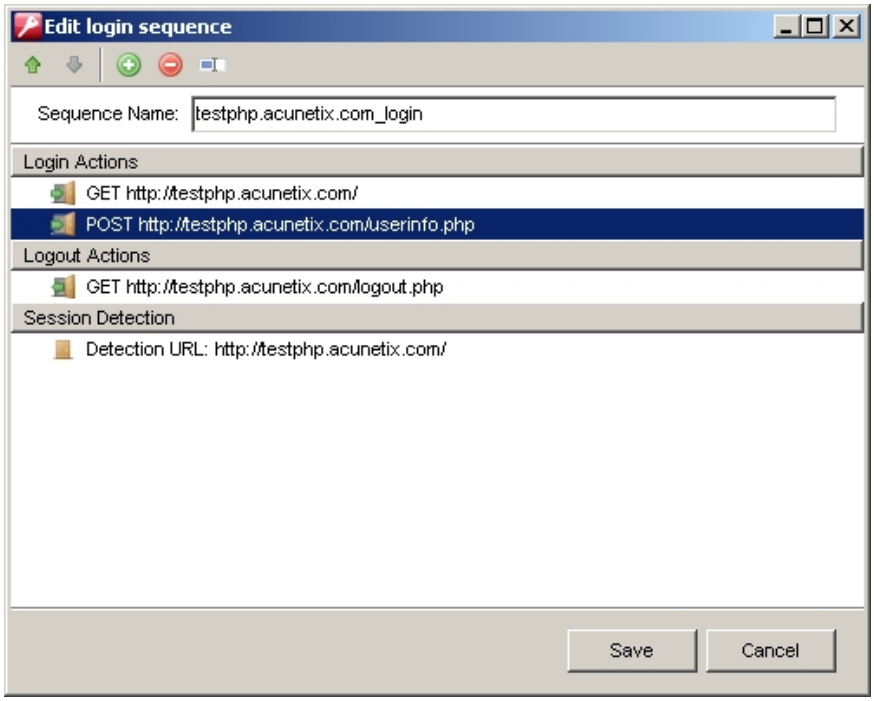
Screenshot 15 – Specify an 'In session' or 'Out of session' pattern

8. Click next to review the recorded login details. One can change priority of url's, edit requests and add or remove requests. Click 'Finish' to finalize the session recording.

Note: Login sequences are saved in the Acunetix Program Files in the sub directories '\Data\General\LoginSequences'.

Editing a Login Sequence

The login sequence can be reviewed by clicking on the 'Edit sequence' button.



Screenshot 16 – Login Sequence Editing

You can change the request priority by highlighting the URL and clicking the up or down arrow in the top right hand side of the window. You can also change the request data by highlighting the request, and click on the 'Edit' button.

Creating a pre-defined crawling sequence

The Login Sequence Recorder can also be used to configure the crawler to crawl a web application in a pre-defined manner, such as a shopping card. To record a pre-defined crawl using the Login Sequence Recorder:

1. Click on the '' button to open up the Login Sequence Recorder. Enter the URL of the website and click on 'Next'. You can also click on 'Check URL' to confirm that the URL entered is reachable from Acunetix Login Sequence Recorder. Click 'Next' to proceed to the 'Record Login Actions' step.

2. Navigate through the web application in the sequence you would like the crawler to crawl it in the 'Record Login Actions' step. Once the whole sequence is recorded, click 'Next'.

3. Specify any restricted links if you would like that the crawler does not crawl them at a later stage. Click the 'Next' button to proceed to the next step.

4. Do not specify any 'In-Session' details. Click 'Next' to verify the pre-defined crawl details. Click 'Finish' to save the crawl.

Marking Pages for Manual Intervention (human input is required)

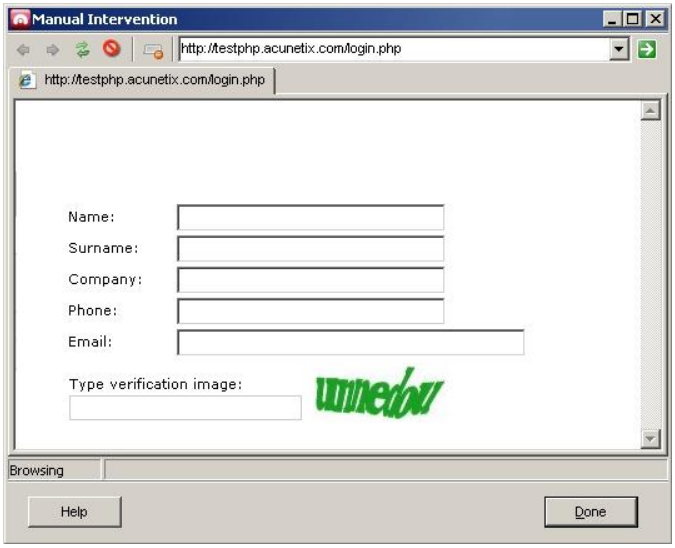
If some pages in your web application require manual intervention, such as pages with CAPTCHA, One-Time password or Two-Factor authentication, use the Login Sequence Recorder to configure the crawler to wait for an input when crawling such page. To mark a page for manual intervention:

1. Launch the Login Sequence Recorder and enter the web application URL in the first step.

2. In the second step of the wizard 'Record Login Sequence', click on the '' (Pause) button to pause the recording, and enter the URL of the page which requires human input in the URL input field.

3. Once the page is loaded, click on '  ' (Manual Intervention) button. Proceed by clicking the 'Next' button till the end of the wizard.

Once a scan is launched, when the crawler crawls the page specified above, a browser window will pop up (as seen in the screen shot below), which requires manual intervention, such as entering a CAPTCHA code. Click 'Done' once the action is ready.



Screenshot 17 – Manual browser window

Note: Only one page has to be marked for manual intervention from the Login Sequence Recorder. If there are more than 1 page which require manual intervention, the first time the browser window pops up during a crawl, enter the required details for the displayed page, and then enter the URL of the other pages manually in the browser and proceed with the entering the details for each page. By doing so, the crawler will automatically crawl those pages as well.

Settings >Scanner settings > Input Fields

In this node, custom values that are sent to forms in websites or web services operations are pre-configured. These values will be submitted by the Scanner during an automated scan when accessing certain parts of the websites or web services which are only accessible when a specific input is given, such as a download links page which is only accessible if a valid email address is submitted to the form.

Note: By default Acunetix WVS already have a generic submit form rule which will submit generic details to any kind of web form it might encounter during scanning.

Creating a web form or Web Service Operations rule for a specific URL

1. Enter the URL of the page or web service containing the specific form or list of operations to which custom parameters are to be passed, and click on "Parse from URL" button. The resulting list will then be automatically completed with the form fields found on the given URL.

2. Enter the values for the required fields from the list by clicking in the value column for that field. Click 'Apply' to save changes.

Note: If specific different details must be specified for each different HTML form, create a new web form rule for each form.

Settings > Scanner Settings > Parameter Manipulation

Parameter manipulation is the changing of data to an attacker's advantage being sent between the web browser and a web server. In this node you can configure the options related to parameter manipulation.

- **Abort combination tests on first alert found** - Tick this option so that the scanner will abandon all remaining XSS variants on a parameter combination if one of them has been found to be positive.
- **Test cookies for all files** - Tick this option so that vulnerabilities that can be executed on cookies will be tested on every file on the server even if it is not clear if they are scripts or not.
- **Exclude normal tests when AcuSensor Technology is used** – When the AcuSensor Technology is enabled, only SQL injections found from the AcuSensor Technology are reported, not to have duplicate results. Tick this option to enable both SQL injections found from AcuSensor Technology and from the scanner.
- **Manipulate the HTTP headers below** - Treat the HTTP parameters listed in the 'HTTP header name' list as a parameter to script. They will be tested by parameter manipulation. Add or remove HTTP headers by clicking on the '+' or '-'buttons.

Settings > Scanner Settings > Parameter Exclusions

In this node specify the parameters to be excluded from a scan. Some parameters cannot be manipulated without affecting the user session therefore will not be manipulated during a scan. You can also select not to test all possible values.

Note: Parameters specified in the Parameter Exclusions list will only be excluded from a scan and not from a crawl; therefore they will still be crawled from the crawler.

Adding a parameter to the exclusion list

1. Specify a URL in the 'URL' textbox to exclude the parameter when scanning the specified URL only. Use '*' wildcard to exclude the parameter from every scan.

2. Type the parameter name to be excluded in the 'Name' textbox and select for which type of HTTP verb it should be excluded from the 'Type' drop down menu. Select 'Any' to exclude the parameter in any type of HTTP verb.

3. Select 'Exclude from Scan' to exclude any kind of parameter manipulation during scan or select 'Do not test all possible values' to try only a limited number of variations during a scan from the 'Action' drop down menu. Click 'Apply' button to save changes.

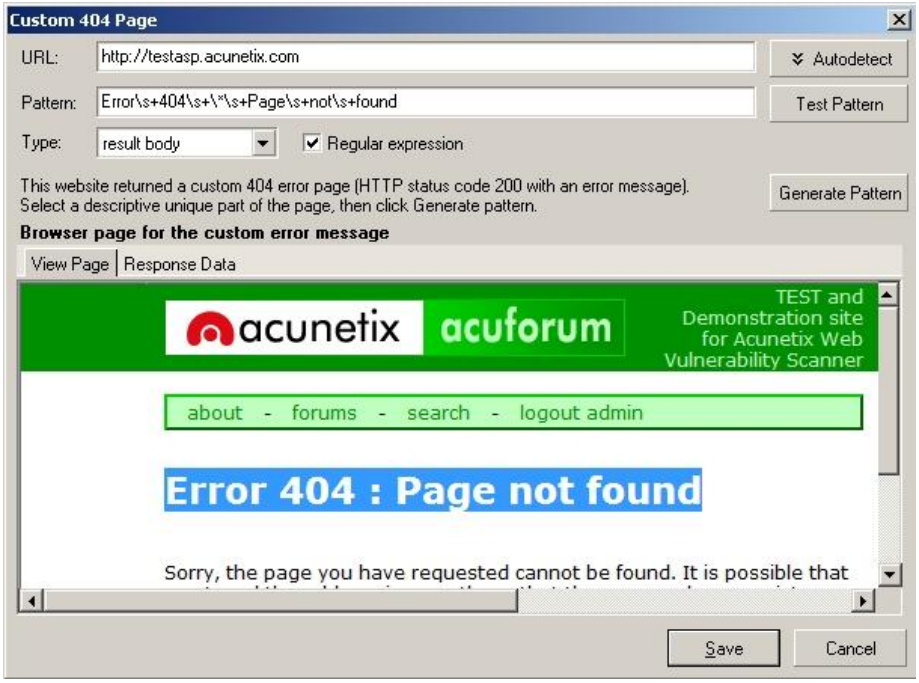
Settings > Scanner Settings > Custom 404 Pages

A HTTP 404 error page (page not found) is the page which appears when a requested page is not found. In many cases though, rather than displaying the standard HTTP 404 error page, many websites show a page formatted according to the look and feel of the website to inform the user that the page requested does not exist. Some of them also return HTTP 200 success code instead of a HTTP 404 error status. In such cases, Acunetix WVS must be able to automatically identify these pages to detect the difference between a non existing URL and a valid web page.

Adding a custom error page

1. Click on the  icon to open the Custom 404 page dialog window. Enter the URL of the website in the URL textbox and click on 'Autodetect' to check

if the website returns a normal 404 error code on Page not found or no. This will extend the current window to show the custom error page, as seen in the screenshot below.



Screenshot 18 - Customer Error Page Configuration

2. Highlight the text that is unique to this custom error page, for example: "Error 404: Page not found"; this text should not be found on any other page of the website.

3. Click on the 'Generate pattern' button to generate a regular expression from the highlighted text. The regular expression generated from the highlighted text will appear in the 'Pattern' textbox.

4. From the 'Type' drop down menu, select one of the following:

- **Location header** - To look for the defined pattern in the location header of the custom error page.
- **Result Body** - To look for the defined patter in the body of the custom error page.
- **Result** - To look for the defined pattern in both the header and body of the custom error page.

5. Click 'Save' to save this custom error page configuration.

Editing a custom error page

To edit a custom error page, select the custom error page you would like to edit and click on 'Edit Pattern'.

Settings > Scanner Settings > GHDB

By default, all GHDB tests (1450+) are launched against a website during a scan. From this node, the vulnerabilities can be unchecked so they are not included by default in a website scan.

Filter the list by entering a keyword (e.g. sql) in the 'Filter GHDB' text box. Click on 'Uncheck Visible' to uncheck all vulnerabilities which match with the keyword and exclude them from a default scan. Click 'Check Visible' to check all entries again and include them in a default scan.

Settings > Scanner Settings > Port Scanner

While scanning a website you can also choose to run a port scan on the web server on which the website is running. The port scanner will scan the web server using a specific list of ports. If a port is found to be open, the port scanner will identify what service is running on that port and will run specific network alerts checks, e.g. if a DNS server is found to be running, tests such as DNS open zone transfer and DNS open recursion tests are run against the service. The Port Scanner configuration options are:

- **Number of sockets used for scanning** - Specify the amount of network sockets to be used from the Port Scanner module.
- **Connection timeout (in seconds)** - Specify the number of seconds for a timeout, i.e. if there is no response when trying to connect to a port within the specified amount of seconds, the port will be considered as closed.
- **List of scanned ports** - The list of specified ports for which the Port Scanner will check. Use the '+' button to add a port and a description and use the '-'button to remove selected ports from the list.

A list of open ports on the server will be displayed in the scan results under Knowledge Base > List of open TCP Ports.

Note: The Network Alert Scripts are fully scriptable. You can modify them from the Vulnerability Scanner or even write new ones.

Settings > Scanner Settings > False Positives

When a specific vulnerability is marked as False Positive from the scan results, it will be listed in the list shown in this node. Press on the '-' button to remove a vulnerability from the list of False Positives.

Note: False positives are specific per site (URL) and file. Therefore if you mark a XSS vulnerability on http://www.testphp.acunetix.com/artists.php as false positive, if you scan another site this vulnerability will show again if it is found.

Scanning Profiles

Scanning profiles may be used to test a website for specific vulnerabilities instead of testing it for all vulnerabilities. From the node 'Configuration > Scanning Profiles' in the tools explorer, new profiles can be created or already created profiles including the default ones can be edited.

Default Scanning Profiles

Acunetix WVS is installed with the following default profiles:

Profile	Description
default	This profile includes all vulnerability checks, excluding only web services related tests.
cgi_tester	The CGI tester scanning profile only searches for common CGI scripts and common sensitive files. This script detects all HTTP methods supported by the targeted web server, for example GET, PUT, DELETE etc. and its functionality.
dir_file_checks	This scanning profile scans the structure of a target website for directories and files. For example when scanning PHP-based websites, this profile would search for such files as phpinfo.php, which could

	contain sensitive information about the PHP configuration of that server.
empty	This profile does not perform any tests. This profile may be used as a clean base to create other profiles.
parameter_manipulation	This scanning profile launches all parameter manipulation attacks, for example SQL injection, XSS 'Cross site scripting' and Command execution.
text_search	The text search scanning profile scans files and filenames for remarks and text. These could contain sensitive information.
version_check	The version_check scanning profile scans the version of the web servers (e.g., Apache), and the different technologies (e.g., PHP, mod_ssl, etc.) in use and compares them to a list of vulnerable versions.
blind_injection	This profile includes only the MultiRequest parameter manipulation section with the Blind SQL / XPath Injection tests.
GHDB	Only GHDB vulnerabilities will be checked. For a more granular selection of the GHDB tests.

Creating/Modifying Scan Profiles

Creating a new Scan Profile

1. Click on the new scanning profile '  ' button in the middle panel at the top and type a name for your profile.

2. Uncheck all the scanning tests to be run when choosing this profile and click on save '  ' button to save the profile.

Modifying a Scan Profile

To modify a profile simply check/uncheck the test modules from an existing profile and save the changes to the current profile.

6. AcuSensor Technology

Introduction

Acunetix AcuSensor Technology is a new security technology that allows you to identify more vulnerabilities than a traditional Web Application Scanner, whilst generating less false positives. In addition it indicates exactly where in your code the vulnerability is and reports also debug information.

The increased accuracy is achieved by combining black box scanning techniques with feedback from sensors placed inside the source code while the source code is executed. Black box scanning does not know how the application reacts and source code analyzers do not understand how the application will behave while it is being attacked. Therefore combining these techniques together achieves more relevant results than using source code analyzers and black box scanning independently.

The AcuSensor Technology does not require .NET source code; it can be injected in already compiled .NET applications! Thus there is no need to install a compiler or obtain the web applications' source code, which is a big advantage when using a third party .NET application. In case of PHP web applications, the source is already available.

To date, Acunetix is the leading Web Vulnerability Scanner to implement this technology.

Advantages of using AcuSensor Technology

- The ability to provide more information about the vulnerability, such as source code line number, stack trace, affected SQL query.
- Allows you to locate and fix the vulnerability faster because of the ability to provide more information about the vulnerability, such as source code line number, stack trace, affected SQL query.
- We can significantly reduce false positives when scanning a website because we can internally understand better the behaviour of the web application.
- Can alert you of web application configuration problems which could result in a vulnerable application or expose internal application details. E.g. If 'custom errors' are enabled in .NET, this could expose sensitive application details to a malicious user.
- It can advise you how to better secure your web application and web server settings, e.g. if write access is enabled on the web server.
- Detect many more SQL injection vulnerabilities. Previously SQL injection vulnerabilities could only be found if database errors were reported or via other common techniques.
- Ability to detect SQL Injection vulnerabilities in all SQL statements, including in SQL INSERT statements. With a black box scanner such SQL injections vulnerabilities cannot be found.
- Ability to know about all the files present and accessible though the web server. If an attacker will gain access to the website and create a backdoor file in the application directory, the file will be found and scanned when using the AcuSensor Technology and you will be alerted.

- The AcuSensor Technology is able to intercept all web application inputs and builds a comprehensive list will all possible inputs in the website and tests them.
- No need to write URL rewrite rules when scanning web applications which use search engine friendly URL's! Using the AcuSensor Technology the scanner is able to rewrite SEO URL's on the fly.
- Ability to test for arbitrary file creating and deletion vulnerabilities. E.g. Through a vulnerable script a malicious user can create a file in the web application directory and execute it to have privileged access, or delete sensitive web application files.
- Ability to test for email injection. E.g. A malicious user may append additional information such as a list or recipients or additional information to the message body to a vulnerable web form, to spam a large number of recipients anonymously.

AcuSensor Technology Vulnerability Reporting

Unlike other vulnerabilities found by typical scans, a vulnerability reported from the AcuSensor Technology contains much more detailed information. It can contain details such as source code line number, POOST variable value, stack trace, affected SQL query etc. A vulnerability reported by the AcuSensor Technology, will be marked with '(AS)' in the title.

Configuring and using the AcuSensor Technology

Step 1: Configure the Sensor

Screenshot 19 – AcuSensor Technology Settings

Run the Acunetix WVS scanner and go to the 'Configuration > Settings' node in the Tools Explorer. Click on 'AcuSensor Technology' under 'Scanner Settings' node and configure the following options:

- **Enable AcuSensor Technology** – Select this option to enable Acunetix AcuSensor Technology during a scan.
- **Request List of files** – Select this option to use the Acunetix AcuSensor Technology to retrieve a list of all files present in the website / web application directory and scan them.
- **Enable Server Alerts** – Select this option so the Acunetix AcuSensor Technology will report back server and platform configuration problems.

- **Password** – Click on the Padlock Icon to generate a random password unless you want to specify one yourself.

Click on 'Set Password in AS Files' to generate the Acunetix AcuSensor Technology client files with the configured specified password.

Note: Each time a password is changed and AcuSensor Technology client files are generated, the AcuSensor Technology client on the server must be updated as well by 'uninjecting and uninstalling' and 'installing and injecting' for .NET. For PHP overwrite the old 'acu_phpaspect.php' with the new one.

Step 2: Installing the Sensor

.NET

1. From the Acunetix AcuSensor Technology page, click on 'Open AcuSensor Technology directory for ASP.NET' and copy 'Setup.exe' to the remote server to be scanned. The application requires Microsoft .NET Framework 3.5 to install. You can download it from here.

2. Click on Setup.exe to start the Acunetix .NET AcuSensor Technology Injector installation and specify the installation path. The application will start automatically after installation unless the option 'Start application after installation is finished' is not selected. If the application is not set to start automatically, click on 'Acunetix .NET AcuSensor Technology Injector' in the program group menu.

Note: On Windows 2008, install IIS 6 Metabase Compatibility from 'Control Panel > Turn Windows features On or Off > Roles > Web Server (IIS) > Management Tools > IIS 6 Management Compatibility > IIS 6 Metabase Compatibility' to be able to list all .NET applications running on server.

PHP

1. From the Acunetix AcuSensor Technology settings page:
    a. click on 'Open AcuSensor Technology directory for PHP'
    b. copy the file 'acu_phpaspect.php' to the remote server to be scanned to a location where it can be accessed from the web sever software.

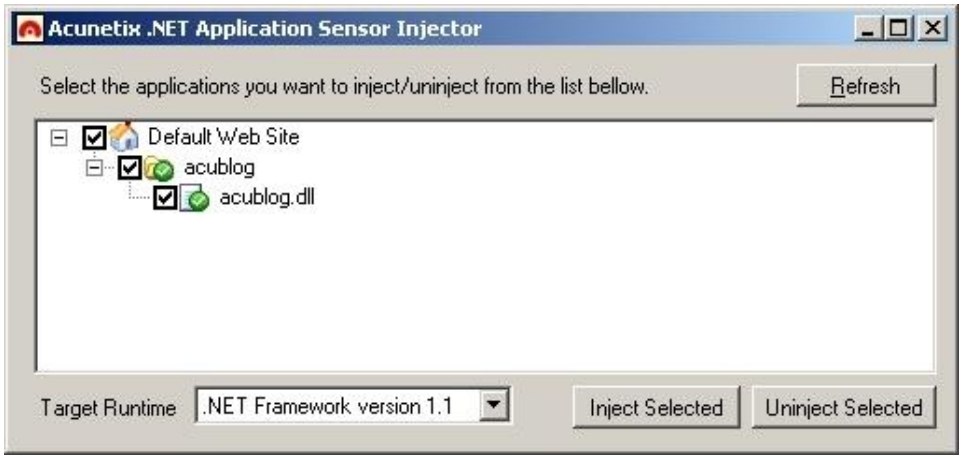
Note: For PHP there is no installation procedure. As explained below, in PHP the AcuSensor Technology file is appended to any PHP file via a configuration in the web server. Acunetix AcuSensor Technology works on PHP 5 and onwards. Previous PHP versions are not supported.

Step 3: Enabling the Sensor

.NET

1. On starting up the Acunetix .NET AcuSensor Technology Injector will retrieve a list of .NET applications setup on your server. Select which applications you would like to inject the AcuSensor Technology code into and select also the Framework version from the drop down menu.

2. Click on 'Inject Selected' to inject the AcuSensor Technology code in the selected .NET applications. Once files are injected, close the success confirmation window and also the AcuSensor Technology Injector.



Screenshot 20 – Acunetix .NET AcuSensor Technology Injector

PHP

In PHP there are 2 methods on how to install the sensor. Method 1 can be used to install the Acunetix AcuSensor Technology on Apache only and Method 2 can be used to install the Acunetix AcuSensor Technology both on Apache and IIS.

Method 1: .htaccess file (Apache)

1. Create a .htaccess file in the web application directory and add the following directive: **php_value auto_prepend_file '[path to acu_phpaspect.php file]'**.

Note: For Windows use 'C:\sensor\acu_phpaspect.php' and for Linux use '/Sensor/acu_phpaspect.php' path declaration formats. If Apache does not execute .htaccess files, it must be configured to do so. Refer to the following configuration guide: http://httpd.apache.org/docs/2.0/howto/htaccess.html. The above directive can also be configured in httpd.conf file.

Method 2: php.ini (IIS and Apache)

1. Locate the file 'php.ini' on the server by using phpinfo() function. Click here for more information about phpinfo() function.

2. Search for the directive **auto_prepend_file**, and specify the path to the acu_phpaspect.php file. If the directive does not exist, add it in the php.ini file: **auto_prepend_file="[path to acu_phpaspect.php file]"**.

3. Save the modifications and restart the web server for the above changes to take effect.

Disabling and uninstalling the Sensor

.NET

1. Run the Acunetix .NET AcuSensor Technology Injector from the program group and select the already injected code. Click on 'Uninject Selected' to remove the AcuSensor Technology code from the .NET applications. On success confirmation, close the confirmation window and the Acunetix .NET AcuSensor Technology Injector.

2. Run uninstall.exe from the application's installation directory.

Note: Just uninstalling the Acunetix .NET AcuSensor Technology Injector without uninjecting the .NET application, does not remove the AcuSensor Technology code from your .NET application.

PHP

1. Delete the directive: **php_value auto_prepend_file="[path to acu_phpaspect.php file]"** from the .htaccess file or from the 'httpd.conf' configuration if method 1 is being used. If method 2 is being used, delete the directive: **auto_prepend_file="[path to acu_phpaspect.php file]"** from the php.ini file.

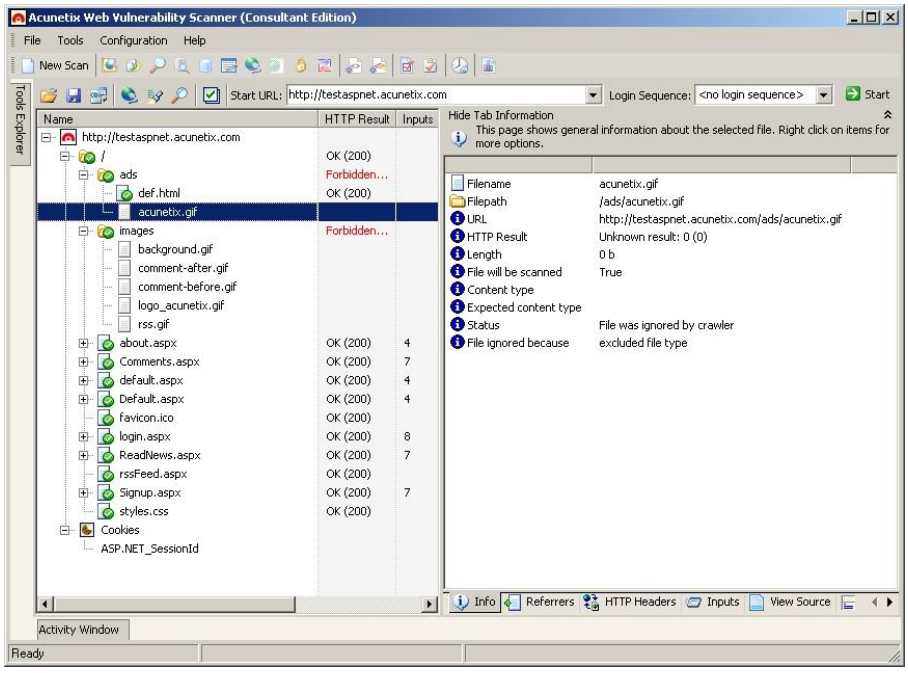
2. Delete the Acunetix AcuSensor Technology php file; acu_phpaspect.php.

Note: Although the Acunetix AcuSensor Technology requires authentication, uninstall / remove the AcuSensor Technology client files from server once not being used anymore.

7. Site Crawler Tool

Introduction

The Site Crawler crawls through the target site and builds the site layout using the information collected, including the site directories and directories. You can use the site crawler tool to analyze the structure of a website without automatically launching the attacks.



Screenshot 21 – The crawler tool interface

The Crawler tool interface consists of:

- **Toolbar** – Here you can specify the URL and start a crawl.
- **Site structure window** (left hand side) – Displays target site information fetched by the crawler, e.g., cookies, robots, files and directories.
- **Details window** (right hand side) – Displays general information about a file selected in the site structure window (e.g., filename, file path etc). At the bottom of the details window, there is a tabbed tool bar. Clicking on the Referrers, Headers, Inputs, View Page or HTML analysis tabs will show further information about the object selected.

Analyzing a Website Structure

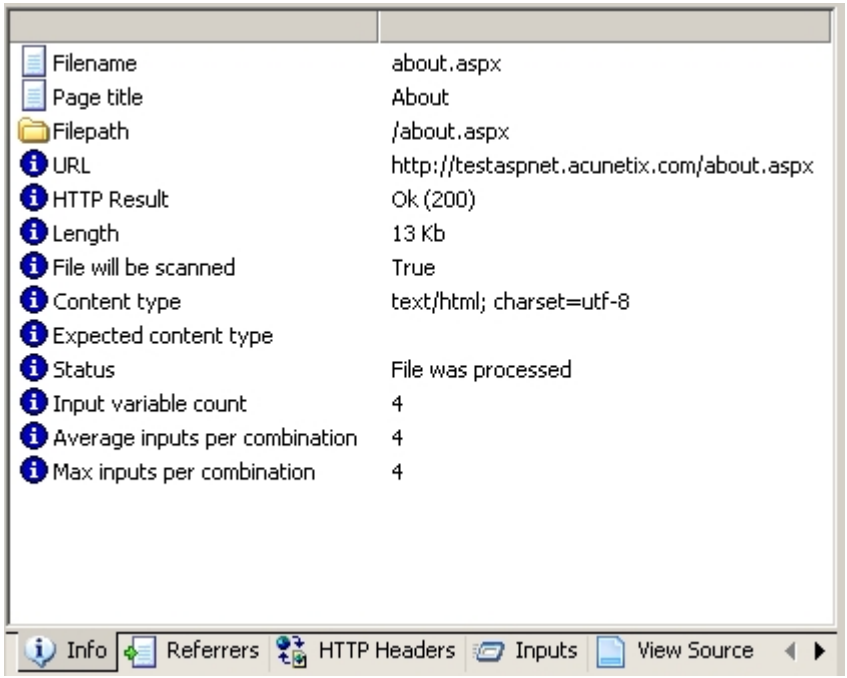
Starting the crawling process

Enter the start URL of the target website from where the crawler should start the site traversal (e.g. http://testphp.acunetix.com/) and click on the 'Start' button.

Note: The crawl process for large sites might take a considerable amount of time, up to several hours for a very large site.

The site structure will be displayed on the left hand side – for each directory found, a node will be created together with sub nodes for each file. At the end of the scan, the site crawler will create a Cookies Node which displays information about the cookies found.

Analyzing the information collected by the crawler



Screenshot 22 – File Details Pane

Clicking on any of the files or directories on the left hand side, will display details about the selected object in the right hand Details Pane (Screenshot 14). From the tabs in the right hand pane the following information is available about the selected object:

- **Info** - Generic information about selected object such as filename, page title, file path, URL, HTTP Result, Length and more.
- **Referrers** - A list of other files on the website from were the crawler found links to the selected object.
- **HTTP Headers** - This tab contains the HTTP request for the selected object and the response received. Details such as content type, date, whether file is cached or not and any relevant server information are found in this tab. You can also edit the HTTP request in the HTTP Editor by clicking the 'Edit with HTTP Editor' icon located on top of the HTTP request pane. This allows the user to analyze how the application will behave when certain parameters are altered.
- **Inputs** - The input tab lists the variables present in the selected object that accept input. The variable name and the variable type are listed. It also displays a list of possible values accepted form the selected variable.
- **View Source** - The source HTML code sent to the scanner when accessing the selected file.
- **View Page** - The tab loads the page as it is displayed in a web browser. However any formatting data such as CSS files, images and client side scripts are disabled for security reasons.

- **HTML Structure Analysis** - Specific HTML code structural information such as HTML tags i.e. EMBED, IMG and LINK found in the selected object.
- **Simple URLs Sub-Tab** - This sub-tab displays the links contained in the selected object. The sub-tag column shows the HTML tag, for example, A for a page link, IMG for an image link and so on. Review this information for pages and links that might reveal sensitive information.
- **Comments Sub-Tab** - This sub-tab displays any comments present within the selected object structure. This information cannot be automatically analyzed but may still reveal interesting developer comments about the construction and coding of the site.
- **Client Script Sub-Tab** - This sub-tab displays the scripts (JavaScript, VBscript etc.) and source code contained in the selected HTML file. These scripts will be executed by the client web browser. Such information might reveal information about the logic of the web application. In the course of a security audit, the user might try to give the application unexpected information and see how it behaves.
- **Input Forms Sub-Tab** - This sub-tab displays any HTML forms in the top window present in the selected file. It also displays a list of fields in the selected form (in the middle window) and also displays the default values for a selected field in the bottom window.
- **META Tags Sub-Tab** - META tags contain information about the web page, for example the description and keywords META tags used by search engines. META tags with an HTTP-EQUIV attribute are equivalent to HTTP headers. Typically, they control the action of browsers and may be used to refine the information provided by the actual headers. Tags using this form should have an equivalent effect when specified as an HTTP header, and in some servers may be translated to actual HTTP headers automatically or by a pre-processing tool.
- **AcuSensor Data** – This tab contains data returned by AcuSensor related to the particular selected object or file.
- **Alerts** – This tab contains all alerts (vulnerabilities) specific to the selected object or file.

Configuring the Crawler

Site Crawler Settings

Configuration settings for the Crawler can be found by going to 'Configuration > Settings >Tool Settings > Site Crawler'. The crawler options are:

- **Start HTTP Sniffer for manual crawling at the end of the scan process** - This option will start the HTTP Sniffer at the end of the crawl to allow manual crawling by enabling the user to browse to parts of the site that were not discovered by the crawler. Frequently these pages are linked via JavaScript menus or similar methods. Although the Acunetix WVS handles JavaScript, there may be situations where a manual crawl is still required. The crawler will update the site structure with the newly discovered links and pages.
- **Get first URL only** - Scan only the index or first page of the target site.
- **Do not fetch anything above start folder** - By enabling this option the crawler will not traverse any links which point to a location above the base link. E.g. if http://testphp.acunetix.com/wvs/ is the base URL, the

crawler will not crawl to links which point to a location above the base URL like http://testphp.acunetix.com.

- **Fetch files below base folder** - By enabling this option the crawler will follow links which point to locations outside the base folder. E.g. if http://testphp.acunetix.com/ is the base URL it will traverse the links which point to a location below the base link like http://testphp.acunetix.com/wvs/.
- **Fetch directory indexes even if not linked** - By enabling this option the crawler will try to request the directory index for every discovered directory even if the directory index is not directly linked.
- **Retrieve and process robots.txt, sitemap.xml** - By enabling this option the crawler will scan for a robots.txt and sitemap.xml files in the target website and follow all the links in it.
- **Ignore CASE differences in paths -** By enabling this option the crawler will ignore any case difference in the links found on the website. E.g. "/Admin" will be considered the same as "/admin".
- **Submit forms -** Select this option to automatically fill in and submit web forms with information that you have previously configured in the 'Configuration > Settings (in Tools Explorer) > Scanner Settings > Input Fields' node. (For full details on how to configure the Acunetix WVS see Chapter 0 on page 99 of this manual).
- **Enable CSA (analyze and execute JavaScript/AJAX) -** Select this option to activate the Client Script Analyzer (CSA) during crawling. This will execute JavaScript/AJAX code on the website to gather a more complete site structure.
- **Fetch external resources requested by CSA –** With this option enabled, CSA will fetch all external resources linked through client scripts present on the target. The external resources will only be crawled and will not be scanned. If this option is not enabled and a client script uses external resources, the CSA engine will not be able to analyze the client script correctly.
- **Fetch default index files (index.php, Default.asp …) -** If this option is enabled, the crawler will try to fetch common default index filenames (like index.php, Default.asp) for every folder, even if these files are not directly linked.
- **Try to prevent infinite directory recursion -** There is a small probability that certain website structures will put the scanner in a loop trying to fetch the same directory recursively (e.g. /images/images/images/images/…) Enabling this setting will instruct the scanner to try to prevent this situation by identifying repeated directory names in recursion.
- **Warn user if URL rewrite is detected –** If URL rewrite is detected on the target, a notification window will appear during the crawl notifying the user that URL rewrite was detected. Switch off this option if you do not want to be advised with such a notice.
- **Maximum number of variations -** This option will specify the maximum number of variations for a file. E.g. index.asp has a GET parameter ID of which the crawler discovered 10 possible values of it from links requesting index.asp with the ID set to a different value while crawling. Each link is a variation. Each variation will appear under the file in the Scan Tree during crawling.
- **Link Depth Limitation –** In this option you can configure the maximum link depth level.

- **Structure Depth Limitation –** In this option you can configure the maximum depth level for directories.
- **Authenticate with this username and password combination** - Select this option if the target website you want to crawl uses HTTP authentication.

Site Crawler Settings > File Extension Filters

In this node a list of file extensions which will be included or excluded during a crawl can be configured. This is done by matching the respective extension of the files specified in any of the columns listed below.

- **Include List** - Process all files which fit the wildcards specified.
- **Exclude List** - Ignore all files which fit the wildcard specified.

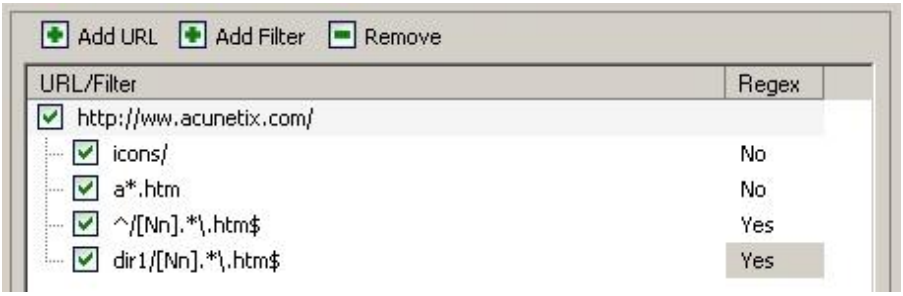
Note: Binary files such as images, movies and archives are excluded by default to avoid unnecessary traffic.

Site Crawler Settings > Directory and File Filters

In this node you can specify a list of directories or filenames to be excluded during a crawl. When specifying a directory or file filter, you can specify the directory name or filename you want to exclude. You can also use wildcards, to match a number of directories or files with one filter. Else, you can use regular expressions to match a number of directories or files. If regular expression is specified as filter, toggle the value under the 'Regex' column to by clicking on the value.

To add a directory or file rule:

1. Click on the 'Add URL' button and specify the address of the website where the directory or file is located.
2. Click on the 'Add Filter' button and specify the directory or filename, a wild card or a regular expression. When specifying a directory name, do not add a slash '/' in front of the directory name.



Screenshot 23 – Directory and File Filter rules

Note: Directory and file filters specified for the root or any other directory, of a website or web application, are not inherited from their sub directories. Therefore a filter for each sub directory should be specified separately, by adding the sub directory name in the filter, as can be seen in the screen shot above.

Site Crawler Settings > URL Rewrite

In this node, a list of URL rewrite rules for websites using this technology can be specified. The purpose of these rules is to configure the crawler to navigate and understand the website structure and not crawl inexistent objects.

Adding a URL rewrite rule manually

1. Click on 'Add Ruleset' button to open up the URL rewrite editor window and enter the host name of the target website for which the URL rewrite rule will be used. Click on  button to open up the Add rule dialogue.

Screenshot 24 – URL Rewrite Configuration

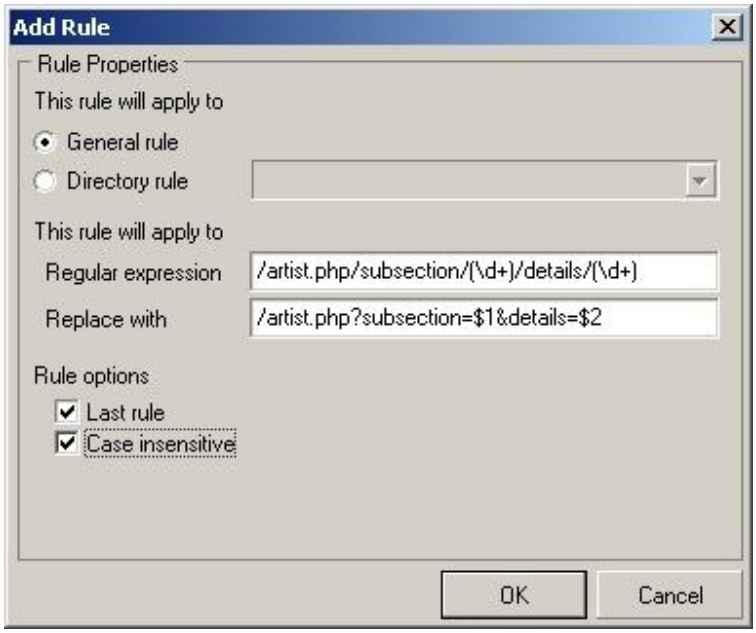
2. Specify if the rule set is generic for all website by ticking 'General rule'. If it will be used for a specific directory, tick 'Directory rule' and specify the directory name.

3. In the 'Regular Expression' input field, enter the part of the URL which is rewritten (as it appears in a web browser). Specify a grouped regulare expression or a group of regular expressions which will be used to match the changing parts of the URL.

4. In the 'Replace with' input field, specify the original URL (before being rewritten by the web server) which Acunetix WVS will request. Specify $1, $2 etc' to specify where the content matched in the regular expression group specified previously should be placed; $1 will be replaced with content matched from the first regular expression group, $2 will be replaced with the content matched in the second regular expression group, etc.

5. Tick 'Last rule' option to specify that if this rule is executed no more rules should be executed afterwards.

6. Tick 'Case insensitive' if the URL's are not case sensitive. Click 'OK' to save the URL rewrite rule.



Screenshot 25 – URL Rewrite Rule

7. Test the URL rewrite rule by specifying a URL in the URL and click on 'Test Rule'.

Importing a URL Rewrite rule configuration from an Apache web server

1. Click on 'Add Ruleset' and then click on import rule button '  'to open the Import Rewrite rules dialogue and enter the path of the Apache httpd.conf or .htaccess file (the file which contains the URL rewrite rule) in the 'Filename' field.

2. Select the type of configuration to import (httpd.conf or .htaccess). If .htaccess is being used, the hostname of the website (e.g. www.acunetix.com) and directory (e.g. sales) in which the URL rewrite configuration is set on the web server needs to be specified.

Site Crawler Settings > Custom Cookies

In this configuration node define custom cookies for each URL to be sent to the web server during a scan. To add a custom cookie:

1. Click on  Add cookie button to add a new blank cookie to the list.

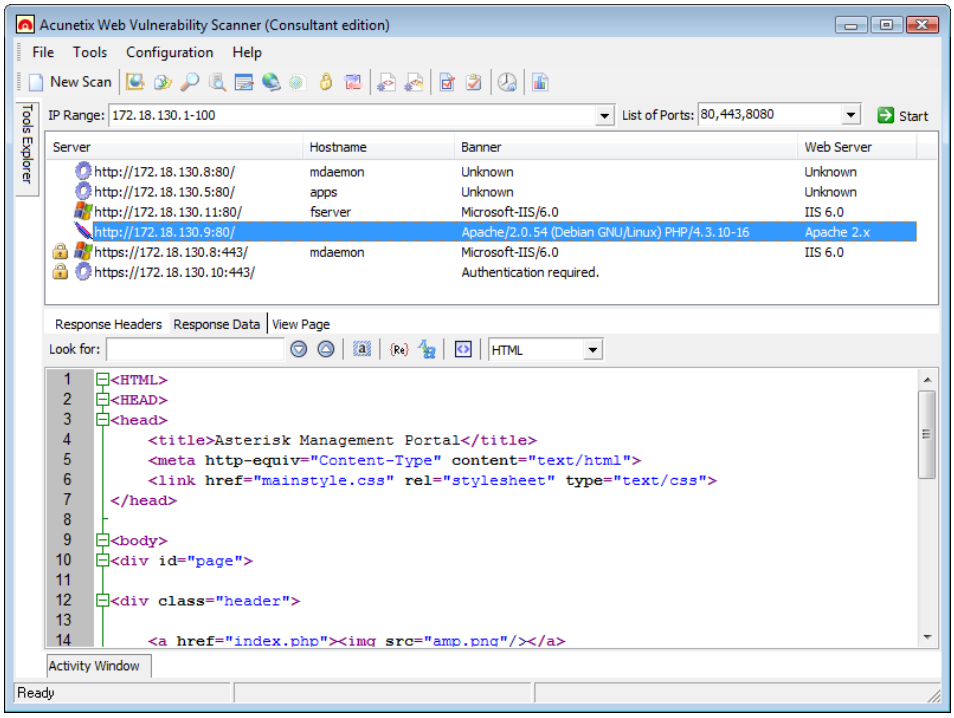
2. Enter the URL of the site for which the cookie will be used.

3. Enter the custom cookie string that will be sent with the cookie. E.g. if cookie name is 'Cookie_Name' and content is 'XYZ' enter 'Cookie_Name=XYZ'. Click 'Apply' button to save changes.

8. Target Finder Tool

Introduction

The Target Finder is a port scanner which can be used to find websites on a given IP or within a range of IP's. The list of ports on which the web servers are listening can also be specified, but by default the scanner will scan only using the default ports, i.e. port 80 for HTTP and port 443 for SSL.



Screenshot 26 - Target Finder view

To Start A Scan

In the Acunetix WVS Tools Explorer select the 'Target Finder' node. You can enter a single IP or a range of IP's to be scanned, e.g. 192.168.0.1-100. If the web servers to be scanned are listening on non default ports add the port numbers to the list of ports.

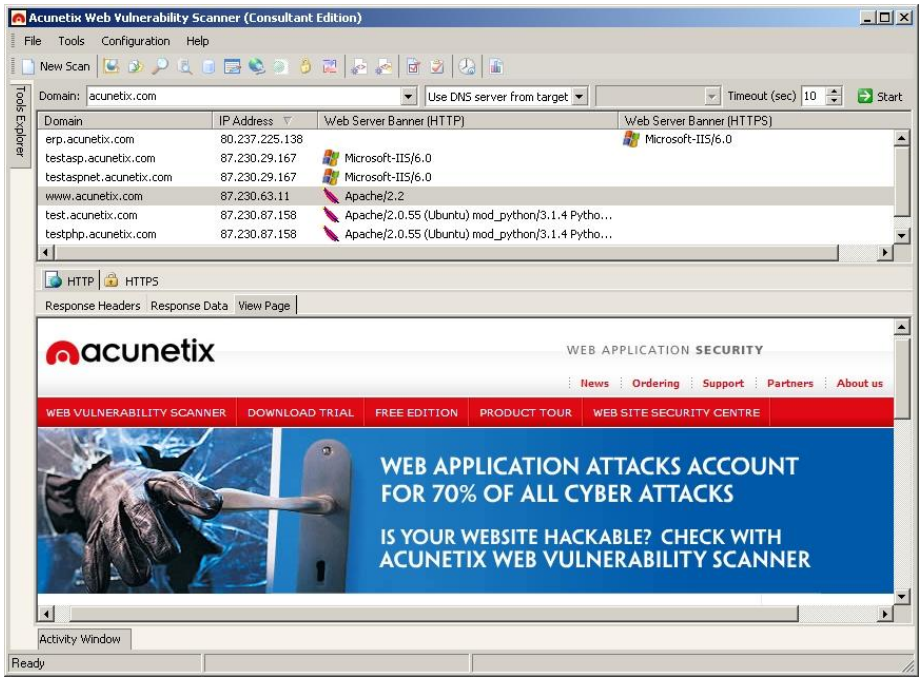
After the scan is complete, the detected web server/s is/are displayed, including the respective server type, hostname and server banner. HTTPS web servers are identified by a padlock icon.

Once the scan is finished you can right-click the discovered web server to launch a scan on the discovered web server, send custom requests using the HTTP Editor, save the list of results as a text-file to be imported into the scan wizard or export the list of servers to a CSV file.

9. Subdomain Scanner Tool

Introduction

The Subdomain Scanner scans a top-level domain to locate any sub domains configured in its hierarchy, by using the target domain's DNS server, or by using any other DNS server specified by the user.



Screenshot 27 – Subdomain Scanner Tool

While scanning, this tool will automatically identify and inform the user if the domain being scanned is using some kind of wildcards such as *.domain.com.

Scanning a Domain for Sub domains

In the Acunetix WVS Tools Explorer select the 'Subdomain Scanner' node. Enter the Top Level Domain Name, such as acunetix.com. Select the DNS Server to use; use the target's DNS server; i.e. the authoritative name servers for the domain, or specify a DNS server of your choice.

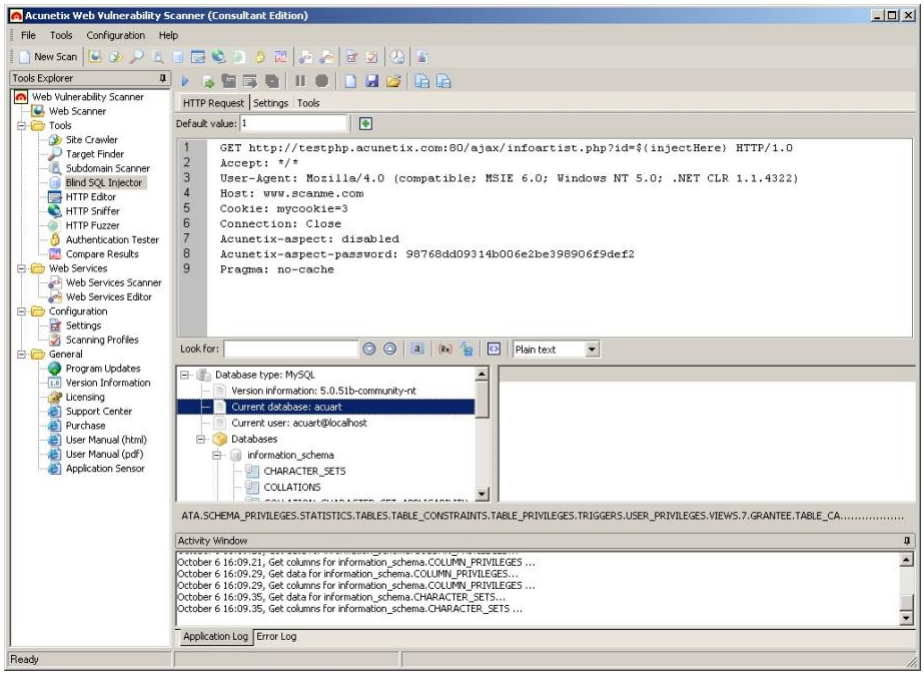
The default timeout specified is an optimal setting; 10 seconds. Increase the timeout if slow responses are encountered.

Once the scan is finished you can right-click the discovered sub domains to launch a scan on the Subdomain, send custom requests using the HTTP Editor, save the list of results as a text-file to be imported into the scan wizard or export the list of servers to a CSV file.

10. Blind SQL Injector

Introduction

Ideal for penetration testers, the Blind SQL injector is an automated database data extraction tool. Using SQL injections found when scanning a website and importing them to this tool, one can see what a serious impact an SQL injection can have on the website. With the Blind SQL Injector tool one can make manual tests to test further a website for SQL injections. One will also be able to enumerate databases, tables, dump data and also read specific files on the file system of the web server. Using this tool, the user can also run custom SQL select queries against the database.



Screenshot 28 - Blind SQL Injector

Importing and writing a HTTP request

The Blind SQL injector needs to know the exact HTTP request from where the remote user can inject data into the database. You can import a HTTP request from a reported SQL injection in a website scan or else write a HTTP request yourself and add an SQL injection point anywhere you would like in the request.

Importing the HTTP request

From the scan results of a website, the user can right click a reported SQL Injection and select 'Import to Blind SQL Injector'. This will import the vulnerable SQL Injection in the tool including the injection point for further analyzes to test against the database.

Writing the HTTP request

The HTTP request can be written manually as plain text in the HTTP Request tab. Specify the exact point where the injection point should be placed by

placing the cursor at the insertion point and click on the '+' icon from the toolbar. This will insert the '${InjectHere}' token which will be replaced dynamically by the injection engine using various injection techniques.

Screenshot 29 - SQL Injection Point

Blind SQL Injector Toolbar – Extracting Data

Once an HTTP request has been imported or written, tests can start to extract the data from a database. Below is an explanation of the Blind SQL Injector toolbar which can be used as a guide to extract data.

- , , - Start, pause or stop the blind SQL injection.
- - Create a new Blind SQL Injection test.
- - Save a Blind SQL Injection test.
- - Open a saved Blind SQL Injection test.
- - 'Get Databases'. Once the Blind SQL injector is able to extract the server banner, you can click on 'Get Databases' icon to get a listing of all databases for which the user has access to running on the SQL server.
- - 'Get Tables'. Once the Blind SQL injector is ready from listing down the databases, select a database from the SQL server viewer and click on this icon to extract all tables of the selected database.
- - 'Get Table Columns'. Select a table from the list of tables extracted by the Blind SQL Injector and click on this icon to extract all the columns present in the database.
- - 'Get Table Data'. Select a table from the list of tables extracted by the Blind SQL injector and click on this icon to extract all the data in the table.
- - 'Export Database Structure / Table Data'. Click on the first icon to export database structure to an XML file or click the second identical icon to export table data to a CSV file.

Blind SQL Injector Tools

File Extraction Tool

With this tool you can extract files from the web server using SQL Injection. This is possible if the injection is already validated. Configure the following options to extract files:

- **File Name -** Specify the exact remote path and filename of the file to extract.

- **Offset** - Specify the character index from where you want to extract data.
- **Length** - How many bytes to extract from such file. Set it to 0 for no limit, i.e. extract all file.
- **Text File** - Tick this option if file is a text file. In this case the extraction algorithm knows it is a text file making the extraction process much faster.

Note: Once 'Extract' is clicked, if the file extraction is successful you will be prompted to specify the location and filename where to save the extracted file.

Execute SQL Query Tool

This tool lets you execute arbitrary SQL queries on remote SQL server. The query can only return 1 row and 1 column therefore the SQL query has to be limited.

- **SQL query –** Write down the SQL query in this text box.
- **Offset** - Specify the character index from where you want to extract data.
- **Length** - How many bytes to extract from the result returned from the SQL query. Set it to 0 for no limit, i.e. extract all result.

Note: Once 'Extract' is clicked, if the SQL query results are successful, you will be prompted to specify the location and filename where to save the results.

Configuring the Blind SQL Injector

Configuration of the Blind SQL Injector can be accessed from the 'Settings' tab in the 'Blind SQL Injector' node.

Settings > General

- **Database Type -** Select 'Automatic' if the database server is not known and the blind SQL Injector will try to guess it. Else, if the SQL server is known, select it from the drop down menu.
- **Extraction Method -** Select 'Automatic' and the tool will try to use the best method possible. 'Condition based' extraction method is the most reliable but slowest. Using 'Union Select', in some limited cases when the SQL query and injection point permits the tool will inject in the existing queries other queries but in a direct way, so this method is up to 8 x faster than the previous one
- **Minimum HTTP Retry -** The number or retries the application will do before reporting a connection error.
- **Encode SQL Spaces with /**/ -** Tick this to encode SQL spaces with /**/. This is a basic way to fool anti SQL injection algorithms.
- **Force HTTP encoding of the SQL string -** Tick this option to automatically encode SQL strings if used in a GET parameter.
- **Encode all characters -** Tick this option to encode all characters not just the special characters.
- **Encode spaces with plus -** Tick this option to encode spaces with a '+' sign instead of %20.
- **Show debug information -** Enable this option to enable debug logging in the application log.

Settings > Condition Based Extractor

- **Injection SQL string > Automatic Detection –** Tick this option if you want that the injection string which will be injected in the SQL is determined automatically by the tool.
- **Injection SQL String > provided by user -**. Select this option to manually input the Injection SQL string yourself. The condition place is given by the ${condition} token, e.g. 1 AND ${condition}/*.
- **True / False condition detector > Automatic -** Select automatic for automatic detection. It may not work if more subtle changes occur in the server response, between consecutive requests.
- **True / False condition detector > Provided by Regex -** Specify the regular expression which must match the response data on true condition.
- **Inverse Regex -** Enable this option when you want that the true condition is triggered when the condition of the above stated regex is false.
- **Character Extractor**
    - **Bit Method -** Select this option to quantize the characters directly to bits and do test on the bits.
    - **Half Method -** If this method is selected, the application will try to find out the numerical value of the character by using the half method, i.e. it will try to find a value in a given interval always splitting the interval in half and testing in which of them the value is and do this recursively.
    - **Try Parallel request -** Tick this option to request all bits in parallel.

Settings > Union Select based extractor

- **Start Column number –** Specify the minimum number of columns expected in a database.
- **Max column number –** Specify the maximum number of columns expected in a database.
- **Visible column index –** Specify a column which the Blind SQL injector can already extract. This setting is used as a reference from the tool. Leave as 0 to set as auto.

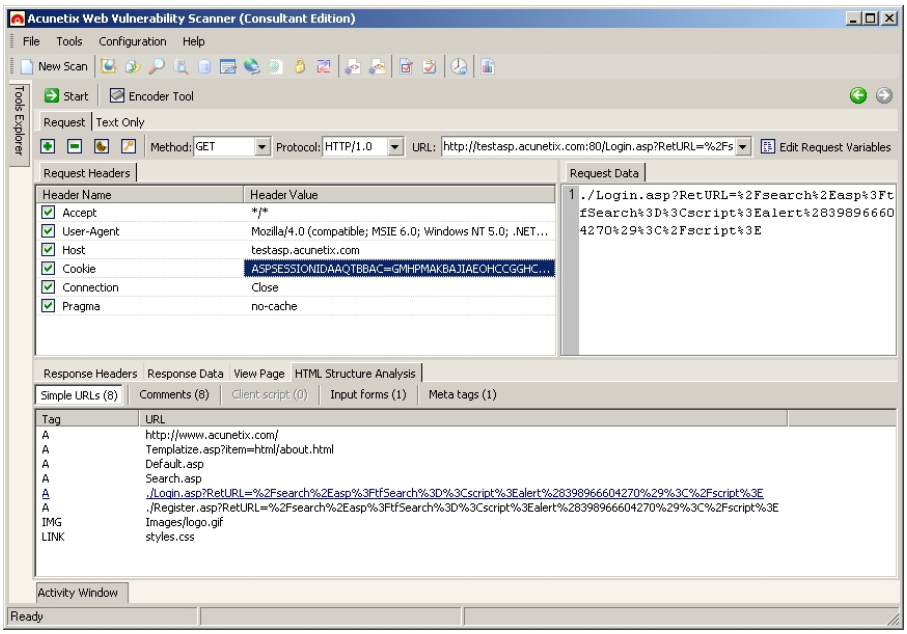
Note: If a database you are scanning may include more than 20 columns per table, increase the value in ‘Max Column Number’.

11. HTTP Editor Tool

Introduction

The HTTP Editor tool allows you to create, analyze and edit client HTTP requests and server responses. You can start the HTTP Editor from the 'Tools' node within the Tools Explorer.

The Top pane in the HTTP editor displays the HTTP request data and headers. The bottom pane displays the HTTP response headers data.



Screenshot 30 - The HTTP Editor

Editing a Request

1. From a Scan or crawl, select a file and right click and select 'Edit with HTTP Editor'.

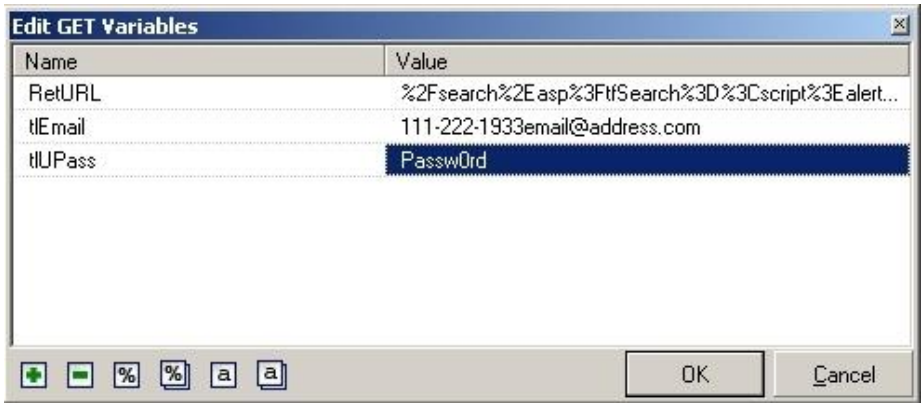
2. From the HTTP Editor Toolbar, the following options can be edited:

- **Method -** Select one of the standard methods supported by all web servers such as GET, POST and HEAD or a custom method supported only by specific web servers such as OPTIONS, TRACE or DELETE.
- **Protocol -** Select the HTTP Protocol (HTTP/1.0 or HTTP/1.1) version to be used for the request.
- **URL -** Specify the fully qualified URL, including the hostname of target object that you want to request (e.g. http://192.168.0.28/). You can specify a relative URL without hostname and request the hostname via the request headers.

3. The Request tab shows the headers of the HTTP request. You can edit any of the headers by specifying the Header name e.g. Cookie or User-Agent and assigning the header text associated to it, e.g. ID=1.

4. To make a request that requires user data apart from the headers (e.g. a POST request with variables); enter the data in the 'Request Data' window.

The variable data can be edited by the variable editor only if it is URL encoded.



Screenshot 31 - Variable Editor

Click on the 'Edit query Variables' button to edit variables in the URL using the variable editor. Query variables are separated from the URL by a "?" and are encoded in the URL-Encode standard. With the variable editor you can edit query variables, cookies and other request data. You can add, remove, URL-encode and URL-decode variables using the buttons in the small toolbar at the bottom of the variable editor window. Click OK when you have entered all the variables.

You can supply data other than the URL encoded variables, such as XML documents for PROPFIND request. Specify the content length and the content type through the appropriate ('content length' and 'content type') headers. In the case that no content length or type is specified, the HTTP Editor will use "application/x-www-form-urlencoded" as the default content type, whilst the content length is automatically calculated.

5. Use the toolbar at the top of the request page to add and remove request headers, add cookie variables, open the encoder-decoder tool and/or specify any HTTP authentication which might be required by the target server receiving the request.

Note: For websites or web applications with AcuSensor Technology enabled, you can manually add AcuSensor Technology headers to the HTTP request. To do this, right click the 'Request Headers' pane and select 'Add AcuSensor headers'. If AcuSensor Technology headers are added to the request, you can view specific AcuSensor Technology data in the response 'AcuSensor Data' tab.

6. Click the icon to specify HTTP authentication details. Select the authentication type (NTLM v1 and NTLM V2 or HTTP Basic) and enter the username and a password.

7. Click the 'Encoder Tool' button to encode-decode any text data that you want to send with a request or that you got back in response. This tool can currently make use of two encoding / decoding techniques to convert plain text data to send in a request. These are Base64 and URL-encoding. Click 'Start' to request to URL.

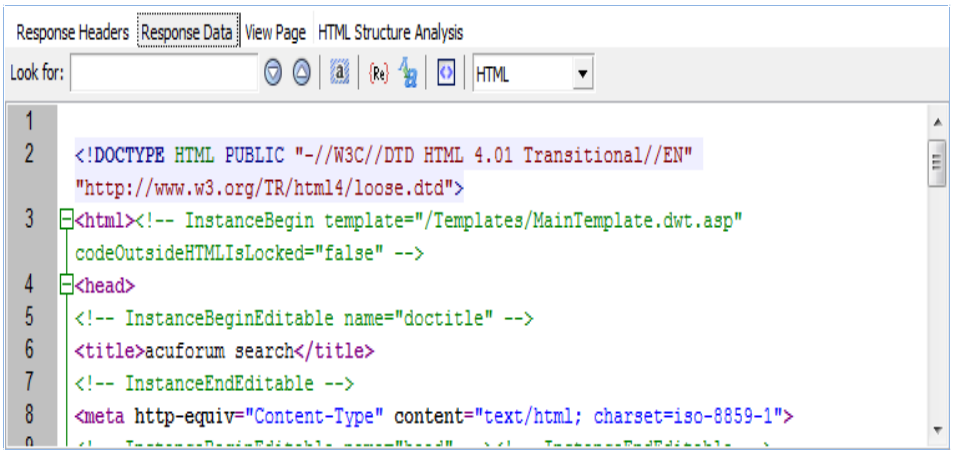
Text Only Tab

This tab displays the request in plain text. You can make changes to the request by editing the text directly on display.

Fine-Tuning Requests and Analyzing Responses

After the request to the server is launched, the server response in the bottom pane of the HTTP Editor can be analyzed. The server response is shown in the tabs 'Response headers', 'Response data', 'View Page', and 'HTML structure analysis'.

Response Headers and Response Data tabs



Screenshot 32 - Response data tab

The 'Response Headers' and 'Response Data' tabs show the headers and data of the response sent back by the server. Cookies information sent by the server can be viewed by clicking on the cookie icon button  located in the HTTP Editor Toolbar.

View Page Tab

The view page tab displays the web page without relevant images or CSS. Clicking on any of the links will display the request of that link in the 'Request Headers' tab and will update the URL in the HTTP Editor Toolbar.

HTML Structure Analysis Tab

In the 'HTML Structure Analysis' a list or links, comments, client scripts, web forms and META tags in the HTML document is displayed.

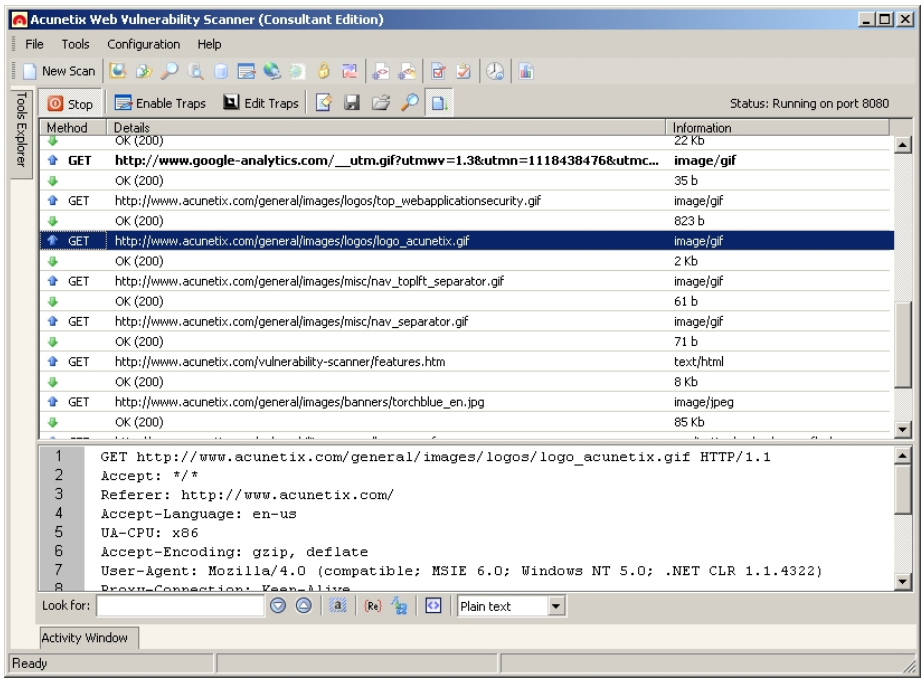
AcuSensor Data Tab

In the 'AcuSensor data' tab, a list of AcuSensor Technology parameters is shown, if the AcuSensor headers are added in the request.

12. HTTP Sniffer Tool

Introduction

The HTTP Sniffer tool is a proxy server which allows you to capture, edit and filter requests made between a web client (browser or other http application) and a web server or vice versa.



Screenshot 33 – The HTTP Sniffer

You can use the HTTP Sniffer tool to create a rule to trap particular POST, GET requests and change them manually, create a rule that automatically changes particular requests and also create a rule to automatically log information in requests or responses.

Enabling the HTTP Sniffer

Toggle the Start / Stop button to enable and disable the HTTP Sniffer. All connection requests and responses will be listed in the main window. To view the complete request or response, click on the entry and all the request or response will be displayed in the lower pane.

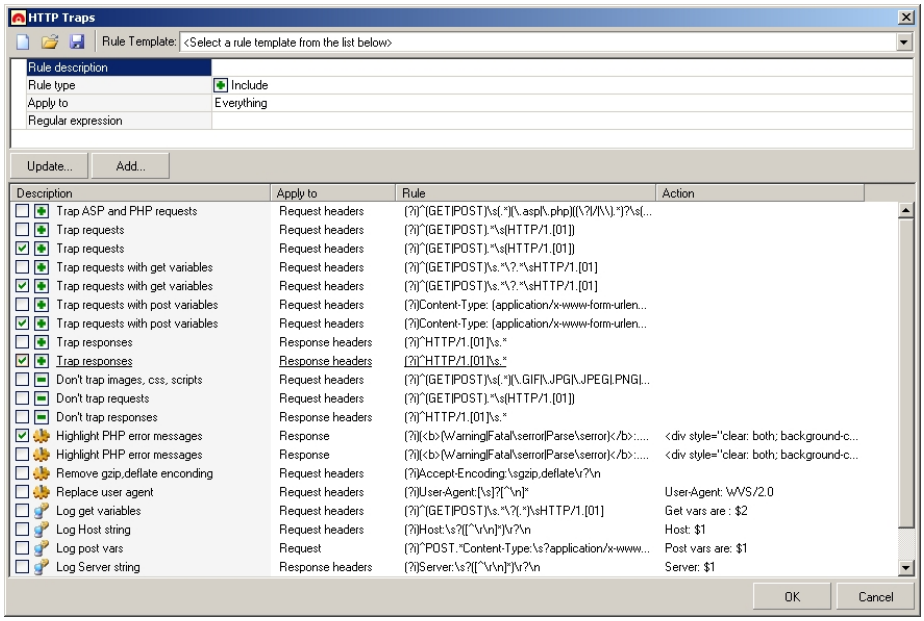
HTTP Sniffer Trap Filters

You can configure the HTTP Sniffer to intercept an HTTP request before being sent so you can make changes to this request and the sniffer will send the modified request to the server.

You can also do the same for HTTP responses; review and edit a particular request before it is sent to the client. This can be done by creating HTTP Proxy trap filters.

Creating a HTTP Sniffer Trap Filter

1. In the HTTP Sniffer toolbar, click on the 'Edit traps' button to bring up the HTTP traps window.



Screenshot 34 - HTTP Sniffer Edit Trap window

2. You can select a rule trap template, e.g. trap requests, and trap ASP or PHP requests. This will load up a preconfigured trap which you can edit.

3. Alternatively you can create a new trap by entering a description, rule type, to what traffic it applies and a regular expression. The following rule types are available:

Trap rules - Configure what requests/responses should be trapped for editing.

Don't trap rules - Configure what trapped requests/responses should be ignored.

Replace or change rules - Configure which requests should be automatically changed based on the given expression.

Logging rules - Configure which requests or responses should be logged in the 'Activity window'.

5. You can now configure whether to apply the trap to all of the response, or just the Request headers, request body and so on.

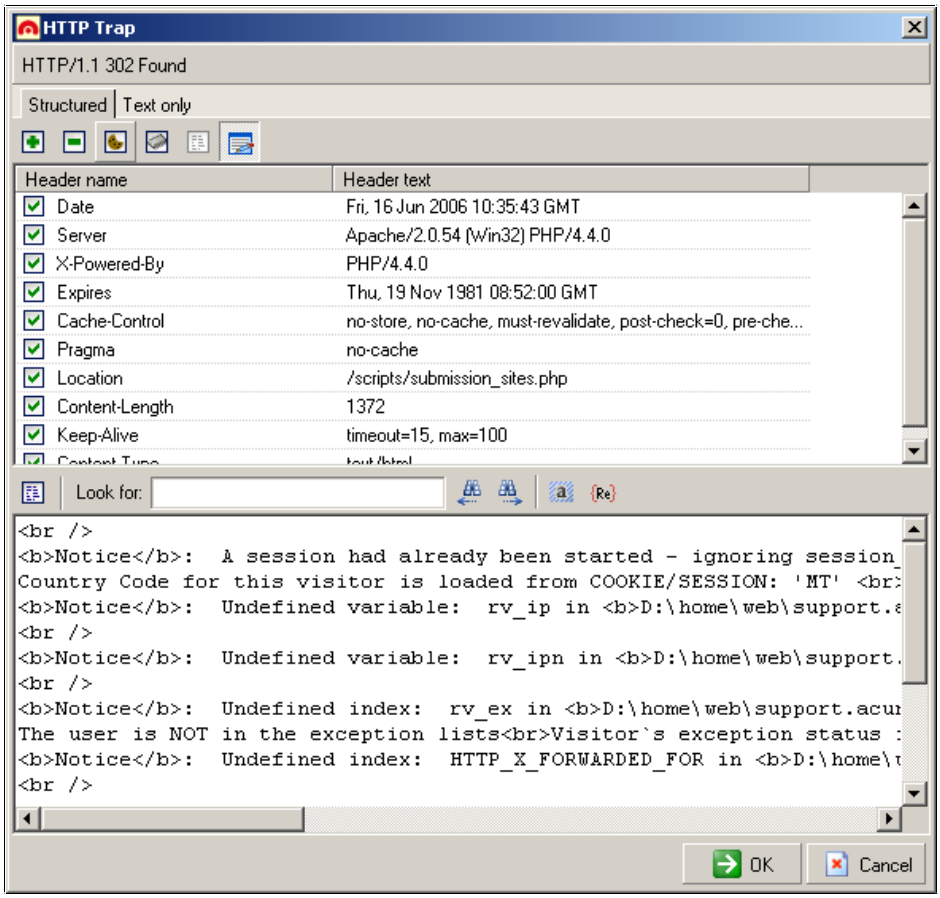
6. Enter a regular expression.

7. To add the trap to the list below, click 'Add'. This will add the trap and automatically enable it. You can enable/disable traps by clicking on the tick box in front of the trap rule.

8. When you have created your trap rules, click the 'OK' button to return to the HTTP Sniffer dialog.

9. Click on the 'Enable traps' button to activate the traps.

The Trap Form



Screenshot 35 - HTTP Sniffer Trap form

When a request or a response is trapped by the HTTP Sniffer, the 'HTTP trap' window will pop up to allow you to edit the request/reply. Similar to the HTTP Editor, the Trap Form editor allows you to edit cookies, query and post variables.

When done, click 'OK' to send the request/response to the server/client.

Analyzing and Responding To the Trapped Requests

After you have created your trap filters and enabled them, the sniffer will follow the steps described below to decide which actions should be taken when handling a certain request or response:

1. Is it included in the log rules? If yes, make a log entry.
2. Is it included in auto change rules? If yes make the requested changes.
3. Is it included in the trapping rules? If no then go to Action 6.
4. Is it included in the exclusion rules for trapping? If yes the go to Action 6.
5. Trap the request or response by using the trap form
6. Forward the request or the response.

Editing an HTTP Request without a Trap

If you want to edit a request without setting up an HTTP trap, right click on a request or a response and select 'Edit with the HTTP Editor'. Then click Start to send the request/response to the server using the HTTP editor.

Configuring the HTTP Sniffer

By default, the HTTP Sniffer proxy server will listen on port 8080 and on local host only; 127.0.0.1. This limits the proxy to capture only traffic from web client applications running on the same machine on which Acunetix WVS is installed.

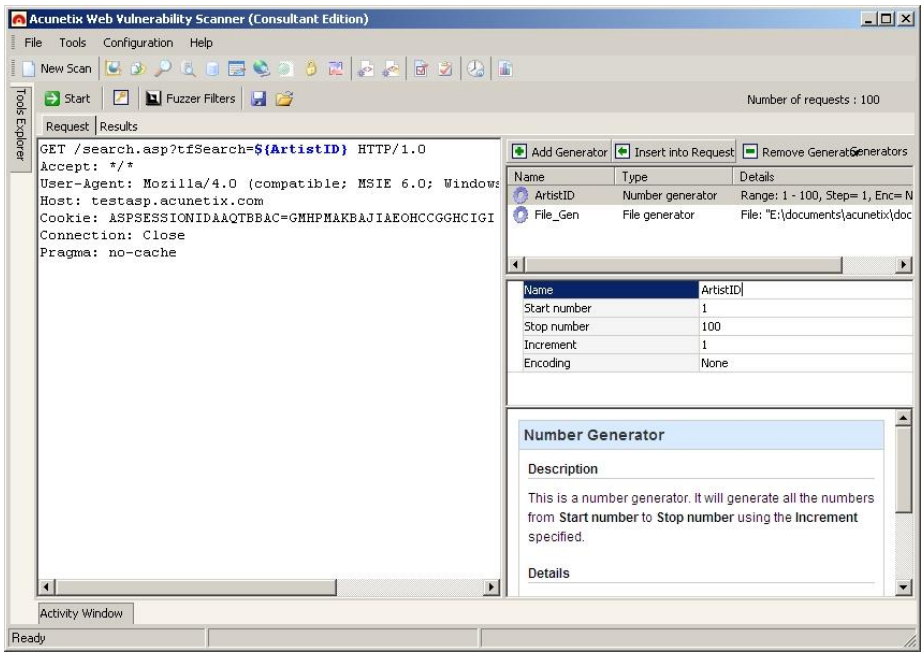
The HTTP Sniffer options can be accessed from 'Configuration > Settings > Tools Settings > HTTP Sniffer'.

From the options the HTTP Sniffer can be set to listen on all interfaces, so web client applications running on other machines can direct traffic through the HTTP Sniffer for analyzes. The HTTP Sniffer port can also be configured; the default is 8080.

For a web client application to send traffic through the HTTP Sniffer, its proxy server settings must be configured to point to the machine where Acunetix WVS is installed.

13. HTTP Fuzzer Tool

Introduction



Screenshot 36 – The HTTP Fuzzer

The HTTP Fuzzer tool allows testing for buffer overflows and input validation. Rules can be created to automatically test a range of variables.

An example URL: http://testphp.acunetix.com/listproducts.php?cat=1

Using the HTTP Fuzzer a rule can be created to automatically replace the last part of the URL (1) with numbers from 1 to 999. Only valid results will be reported. This gives the advantage to quickly test 1000 queries while significantly reducing the amount of time and manual input.

Creating a Rule to Automatically Test a Series of Inputs

A rule will be created to test the products section of the Acunetix test website using a range of values to find out what products are listed in the database.

The scanner will be set to automatically replace the variable part of a URL with a series of values. In the URL, the last part?cat=1 is the variable part.

http://testphp.acunetix.com/listproducts.php?cat=1

Note: The scanner automatically guesses variable sections in a URL and tries to extract valid variables. With the HTTP Fuzzer more advanced tests can be done.

Gathering a HTTP Request

If a valid HTTP request is known, paste it in the 'Request' tab in the HTTP Fuzzer, else load a scan or crawl result from a previously scanned website.

Right click one of the files in the results tree and select 'Export to HTTP Fuzzer'.

Creating Data Generators

Determine which part of the request will be used for testing. This value will be replaced by a data generator.

1. To create a Data Generator, click on the 'Add Generator' button on the right part of the HTTP Fuzzer window.

2. Select the appropriate generator from the drop-down list, which can be:

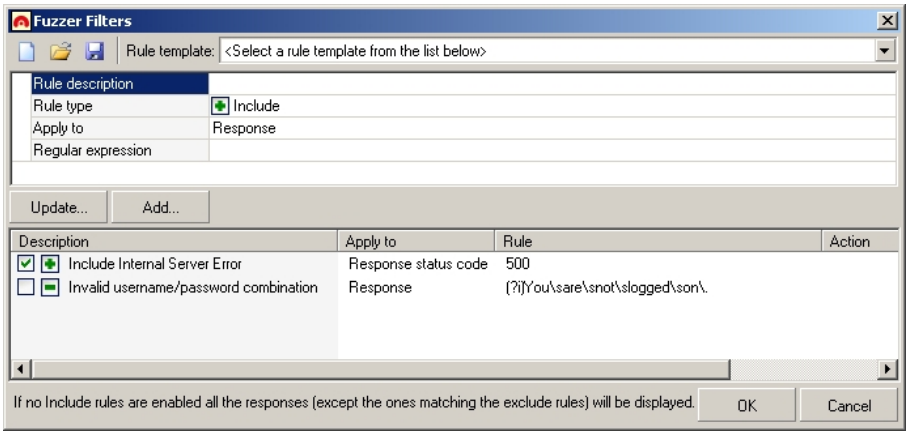
- **Number generator** - This will generate all range of numbers from a start number variable to a stop number variable, using the specified increment.
- **Character generator** - This will generate all the ASCII characters contained between a Start character variable and a Stop character variable using the specified increment.
- **File generator** - This will feed all the strings from a specified file. In the file, each variable string should be entered on a new line.
- **String generator** - This will generate all the string combinations with the characters from a Character set variable of the length specified.
- **Random string generator** - This will generate a specified number of random strings with the characters from a Character set variable of a given length.
- **Character repeater** - This will repeat a specified character/string for a given number of times (commonly used for buffer overflow testing).

3. Once a generator is selected, set the parameters according to the test from the window underneath the generators list.

4. After configuring the generator(s), place the text cursor in the specific part of the HTTP Request where the generator will replace the static value. Highlight the static value (e.g. /artists.php?artist=1). And click on 'Insert into Request'. The static value will be replaced with the generator variable, e.g. /artists.php?artist=${artists_id}.

5. Click the 'Insert into Request' button to replace the static value with the Generator variable (e.g. result will be: /artists.php?artist=${artists_id}) and click on 'Start' to start generating the HTTP requests.

Creating Fuzzer Filters



Screenshot 37 – HTTP Fuzzer filter

To create a Fuzzer filter, click on the 'Fuzzer Filters' button in the toolbar to open the filters dialog. To use a standard filter, select a predefined rule

template from the dropdown list; otherwise custom filters can be created by defining the following parameters:

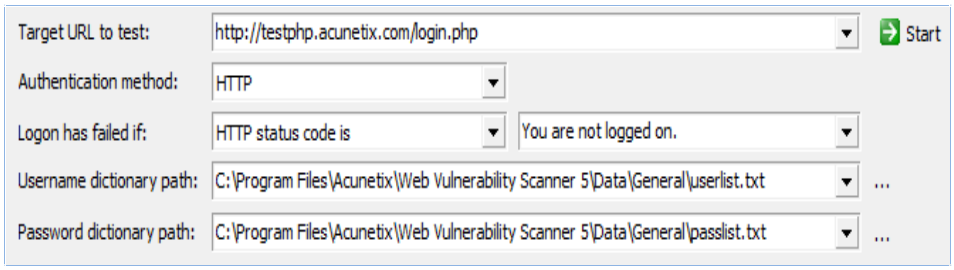
- **Rule description** - A name to describe the rule.
- **Rule Type** - Select an Include type or Exclude type of rule.
- **Apply To** - Indicates where to search for the matching expression, if in the request or response etc.
- **Regular expression** - The regular expression or text which will be searched to match the rule.

Ensure that the relevant checkboxes are ticked to enable the created filters.

14. Authentication Tester Tool

Introduction

The authentication tester is a tool used to test the strength of passwords within HTTP or web forms authentication environments via a dictionary attack.



Screenshot 38 – Authentication Tester

Testing HTTP Authentication

What is HTTP Authentication?

HTTP authentication is part of the HTTP specification. If a site performs HTTP authentication, then the browser will display a password pop-up dialog. With HTTP authentication, the web server validates the logon against a database of users (with IIS these are local Windows user accounts and with Apache these are stored in a file).

Testing the Password Strength

1. From the Tools Explorer, select the 'Authentication Tester' node and in the 'Target URL to test' edit box, specify the target URL e.g. www.test.com/login/

2. Select 'HTTP' as the authentication method to be used for the attack.

3. The default dictionaries will be used. You can also specify your own Username and Password dictionaries by specifying the full path to a plain text file containing the list of usernames or passwords to attempt to login with. Click 'Start' to start the Authentication tester.

Note: By default the Authentication tester will classify a failed logon if the server returns a HTTP response value of 401. However, if custom error pages are used, a matching string or regular expression must be specified 'Logon has failed if' field.

Testing HTML Form Authentication

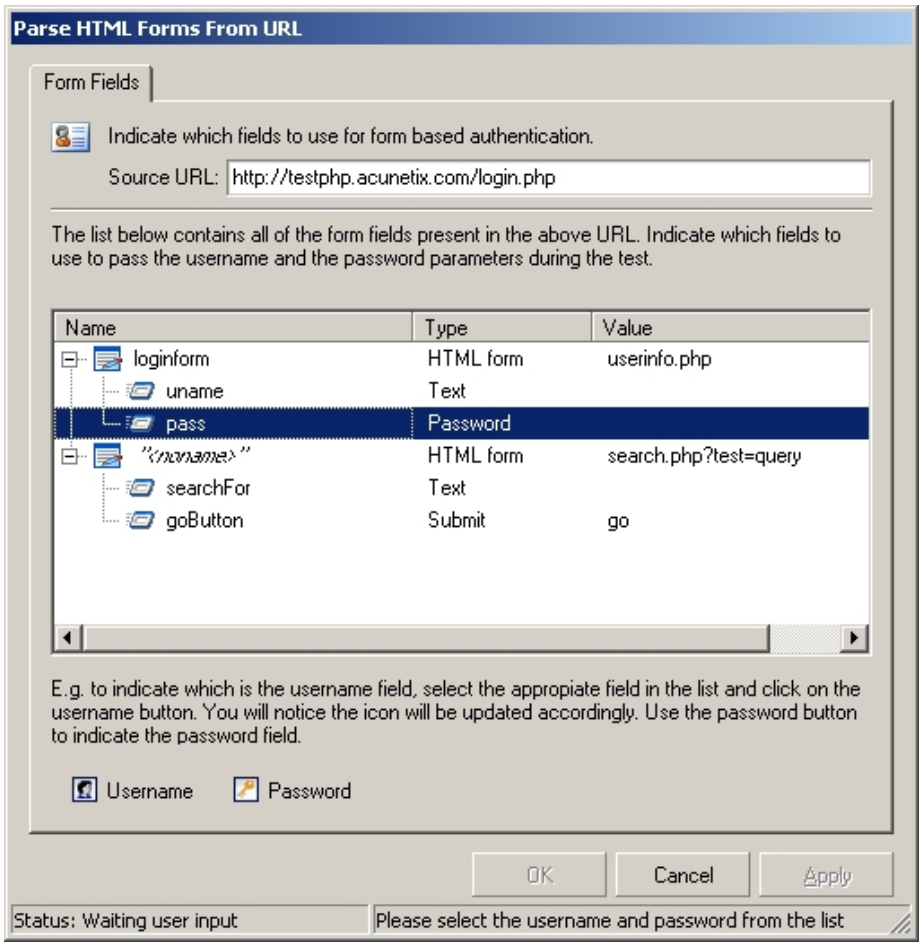
What is Web Forms Authentication?

A logon sequence that implements web forms authentication asks the user for credentials via a web form, which is then validated on the server via a custom script, rather than by the web server directly.

Testing Password Strength

1. From the Tools Explorer, select the 'Authentication Tester' node and in the 'Target URL to test' edit box, specify the target URL e.g. www.test.com/login/

2. Select 'HTML form based' as the authentication method to be used for the attack and click on 'Select user/password form fields to use'.



Screenshot 39 – Specifying HTML Form fields

3. In the 'Parse HTML Forms from URL' screen, the application will display all the available fields contained in the target page, as shown in the screenshot above. Indicate the form field that represents the Username, by clicking on the field and clicking on 'Username' button and indicate the form field that represents the Password by clicking on the field and clicking on the 'Password' button at the bottom of the window.

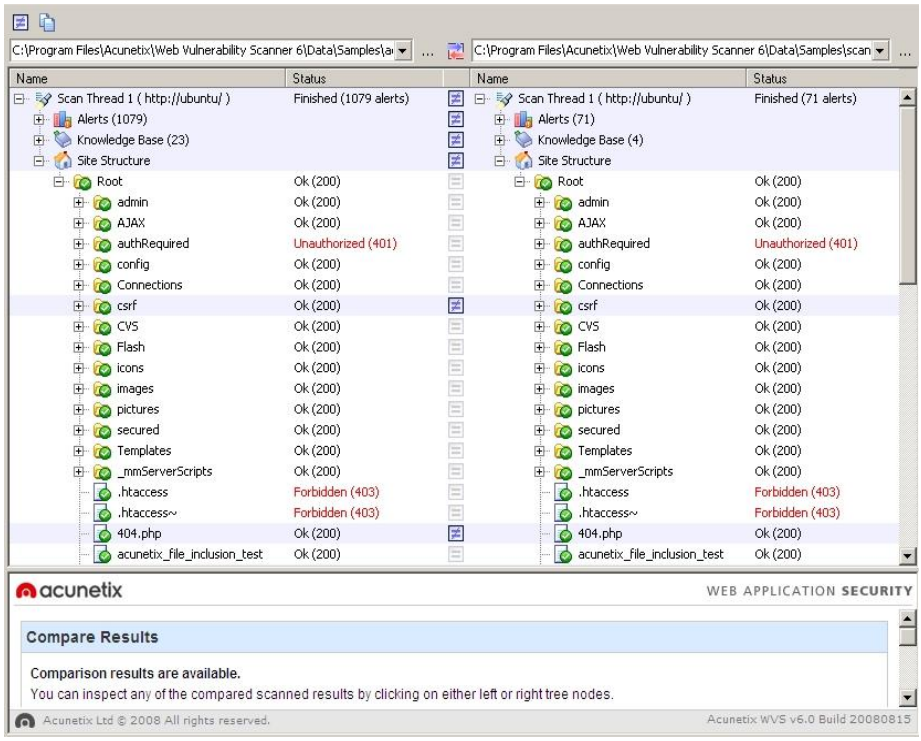
Note: If there are multiple forms on the page, they will be parsed and shown in this dialog. Select the form which contains the relevant authentication fields.

4. Acunetix WVS must be instructed what constitutes a failed login page so the application realizes the appropriate behavior upon successful login. Attempt to logon to the page to generate a login error and note down the text that indicates a login failure. Set 'Logon has failed if' to 'Result contains' and copy the text that indicates a login failure in the input text box. Regular expressions can also be specified by choosing 'Result matches regular expression'. Click 'Start' to start the Authentication tester.

15. Compare Results Tool

Introduction

The compare results tool allows you to analyze the differences between 2 scans performed at different dates. You can compare a full security scan or just the site crawler output. To compare results you need to save the scan results to a scan file using the save scan results function in the file menu.



Screenshot 40 – Compare Results Tool

Comparing Results

To compare scan results:

1. Go to the 'Compare Results' node in the Tools Explorer and in the Compare results toolbar specify the path of a scan file. In the second edit box specify the path of the scan file you want to compare the already specified scan.

3. Click on the Compare '' button to launch the compare results wizard. Specify which items you want to compare such as Referrers, HTTP headers etc. The list of items that are enabled for compare can be saved as a new template by renaming the template and clicking the 'Save' button. Click 'Start' to start the compare process.

Note: For large websites, the file structure comparison process may take a long time to complete.

Analyzing the Results Comparison

Once the comparison is completed, the results are shown in a two-pane interface with a column down the middle. The left pane contains the contents of the original scan while the right hand side pane contains the results of the second specified scan. The middle column shows icons indicating the comparison result of the items in that line. The legend of possible comparison results is shown below:

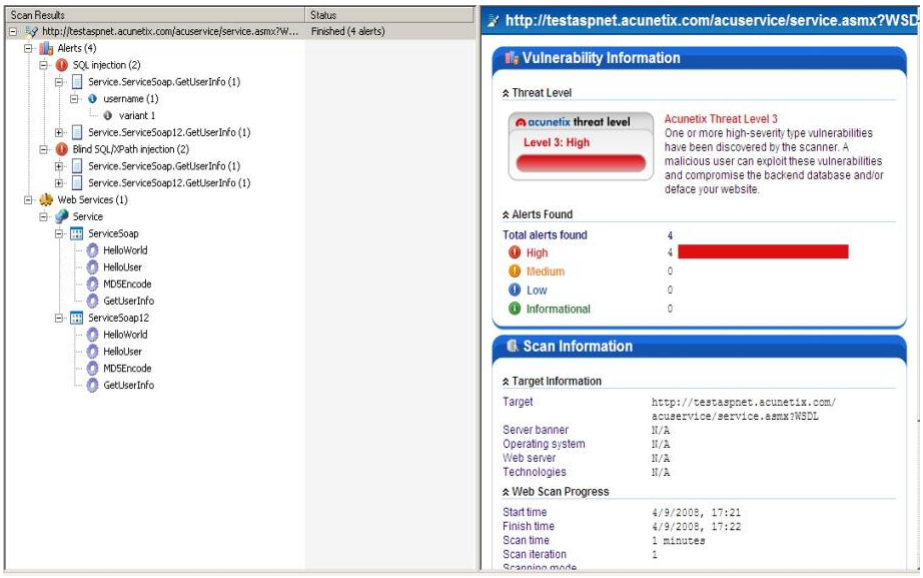
	There are no changes.
	This item was added in the new version.
	This item was deleted from the new version.
	This item was changed in the new version.

Click on the result icon in the middle column to display the comparison result details in the window below the comparison. These details show the changes detected between the two scans such as the number of items present in each scan and the items that have been added or deleted.

16. Scanning Web Services

Introduction

Many organizations are implementing the Web Services architecture to increase the availability of information, and to improve process executions of the internet. Web Services, like any other internet-dependent system, present new exploit possibilities and increase the need for security audits. The Web Services Scanner performs automated vulnerability scans for Web Services and generates a detailed security report from the results.



Screenshot 41 – Web Services Scanner

Starting a Web Service Scan

To start a web service scan ideally you should use the Web Services Scan Wizard as it provides a series of steps to ask for the required details and configuration to be used during the scan.

1. From the 'Tools Explorer' select 'Web Services Scanner' and click on 'New Scan' button in the toolbar to launch the Web Service Wizard. Specify the URL of an online or local WSDL and choose a scanning profile. Click on 'Next' to proceed to the next step.

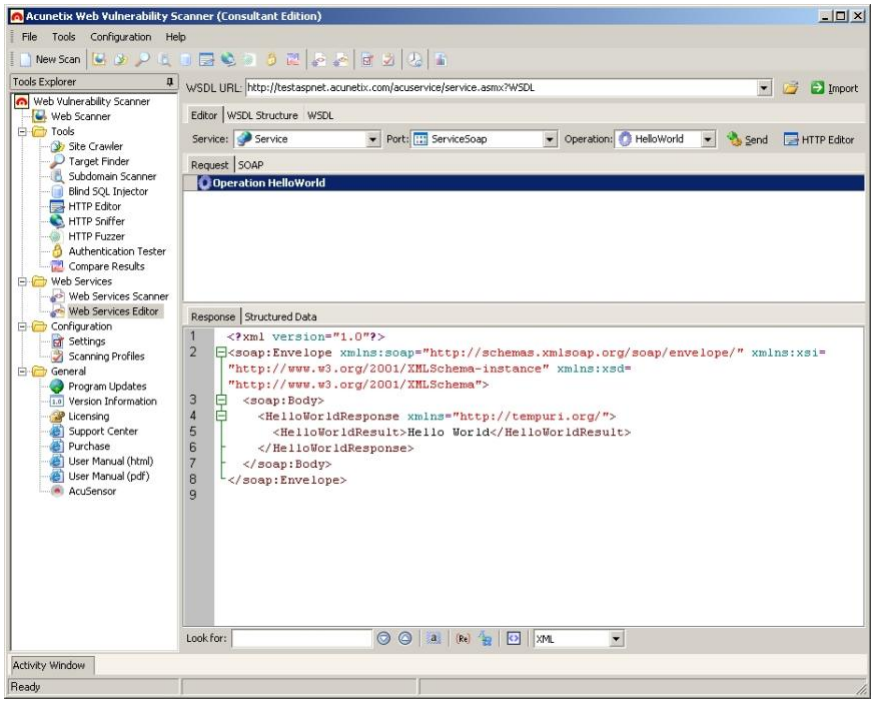
2. In the 'Selection' step, select the Web Services, Ports and Operations that will be scanned. The number of inputs accepted by each operation and the URL of the ports will be displayed in the Details section.

3. Enter specific input values (optional) for the scanner to use custom values for Web Service Operations during the scan in the 'Default Values' step. Proceed to the scan summary, review it and click 'Finish' to start the scan.

Web Services Editor

The Web Services Editor allows importation of online or local WSDL for custom editing and execution of various web service operations over different

port types for in depth analyses of WSDL requests and responses. The editor also features syntax highlighting for all languages to easily edit SOAP headers and customize manual attacks. Editing and sending Web Services SOAP messages is very similar to editing normal requests sent via the HTTP Editor.



Screenshot 42 – Web Services Editor

Importing WDSL and Sending Request

1. Click on the 'Web Services Editor' node in the tools explorer and enter the URL of the WSDL or locate the local directory, and then click 'Import'.

2. In the Editor Tab select the Service, Port and the Operation from the drop down menus in WSDL Services Editor toolbar which will be used to perform the test. Click 'Send' and the editor will build the SOAP request as defined by the operation, sends the request to the web service and displays the server response in a structured or XML view type.

Response Tab

This tab displays the response sent back from the web service in raw XML format.

Structured Data Tab

This tab presents the XML data received from the web service response in a different way by showing the elements in a hierarchy of nodes showing the value for each element.

WSDL Structure Tab

This tab provides a detailed view of the web service data as provided by the WSDL Structure.

The WSDL information is structured in the form of nodes and sub-nodes and the main nodes of the tree structure are XML Schema and Services.

The XML Schema node lists all the ComplexTypes and the Elements of the web service. The Services node lists all the web service ports and their

respective operations together with the resource details of the source of the SOAP data.

A more detailed WSDL structure can also be shown by ticking the 'Show detailed WSDL structure' at the bottom of the screen. This will provide extensive information for each sub-node of the Services node structure such as input messages and parameters.

WSDL Tab

This tab shows the actual WDSL data in the form of XML tags. Using the toolbar provided at the bottom of the screen you can search for certain keywords or elements in the source code and also change the syntax highlighting if needed.

HTTP Editor Export

In the Web Services Editor you can export a SOAP request to the HTTP Editor by clicking on the 'HTTP Editor' button in the Web Services Editor toolbar. The HTTP Editor tool will automatically import the data and now the request can be further customized and sent as an HTTP POST request. Refer to Chapter 10 for further details about the HTTP Editor tool and how to use it.

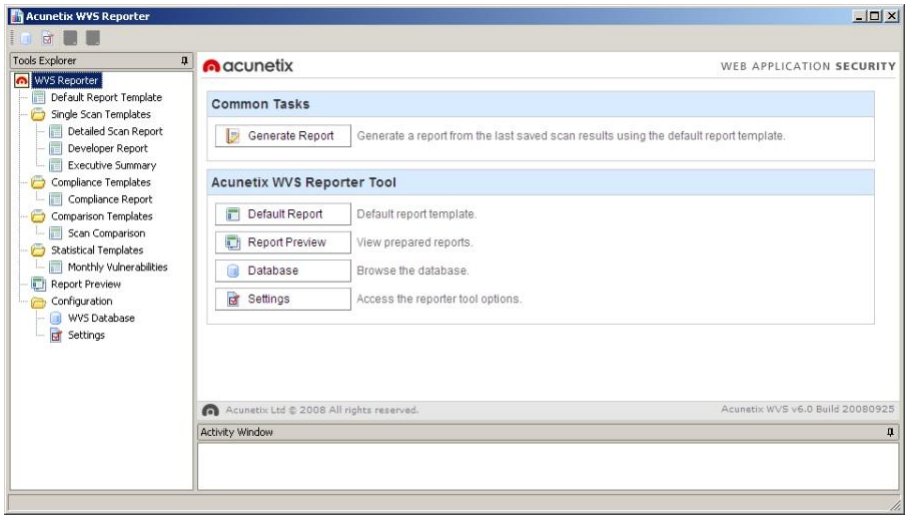
17. The Reporter

Introduction to the Reporter

The Reporter Application is a separate application included in the scanner's installation, which provides the ability to generate several types or reports. It can be launched directly from Acunetix WVS once a scan is complete to generate on-the-fly reports according to the chosen default template or can be launched from the Acunetix WVS program group.

Different reporting templates can be used to categorize scan results according to vulnerability-class, affected pages, general exploit summary, comparison and statistical analysis, and to present exploit details as specified by several compliance standards.

The Reporter Application also allows you to view and manage the scan database and other existing reports.



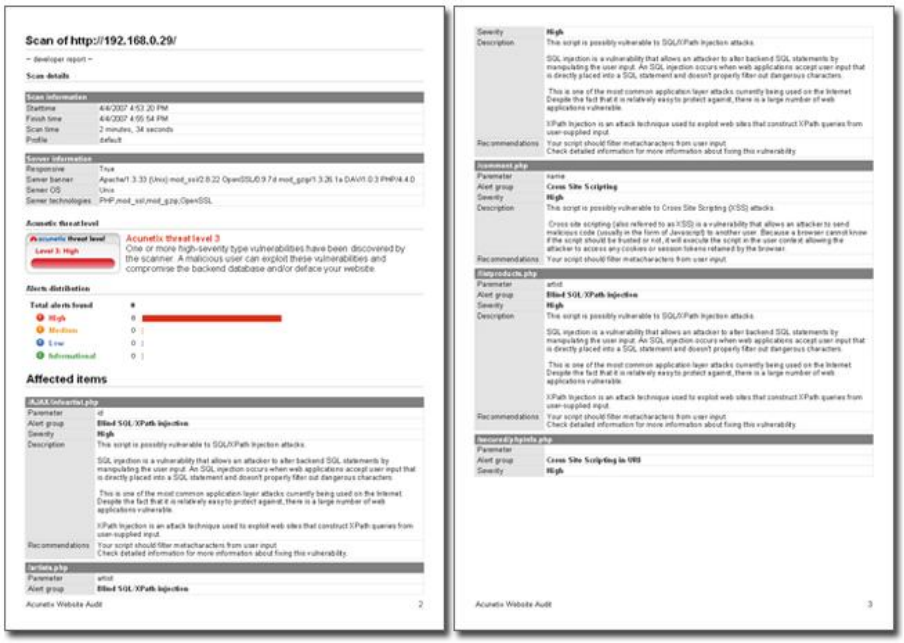
Screenshot 43 – The Reporter Application

Report Templates

The Reporter offers the functionality for creating different type of reports. The packaged templates allow you to launch the specific wizard for a selected report-style, and to quickly present your scan results into the desired format.

Developer Report

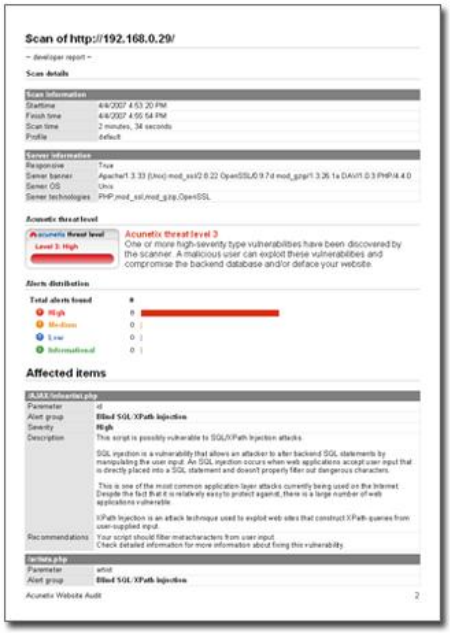
The developer report groups the scan results according to the affected pages and files. This creates an easy workflow for the developer to quickly identify and resolve vulnerabilities detected on the site. This report style also features detailed remediation examples and best-practice recommendations for securing the vulnerable items.



Screenshot 44 – Developer Report

Executive Report

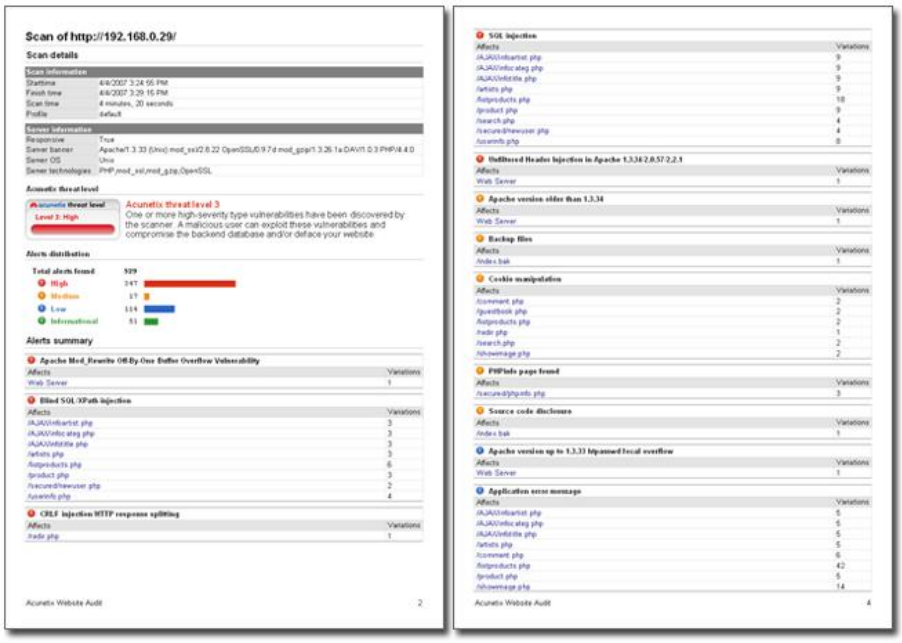
The Executive report creates a summary of the total number of exploits found in every vulnerability class. This makes it ideal for management to review the results without needing to include unnecessary technical detailing.



Screenshot 45 – Executive Report

Vulnerability Report

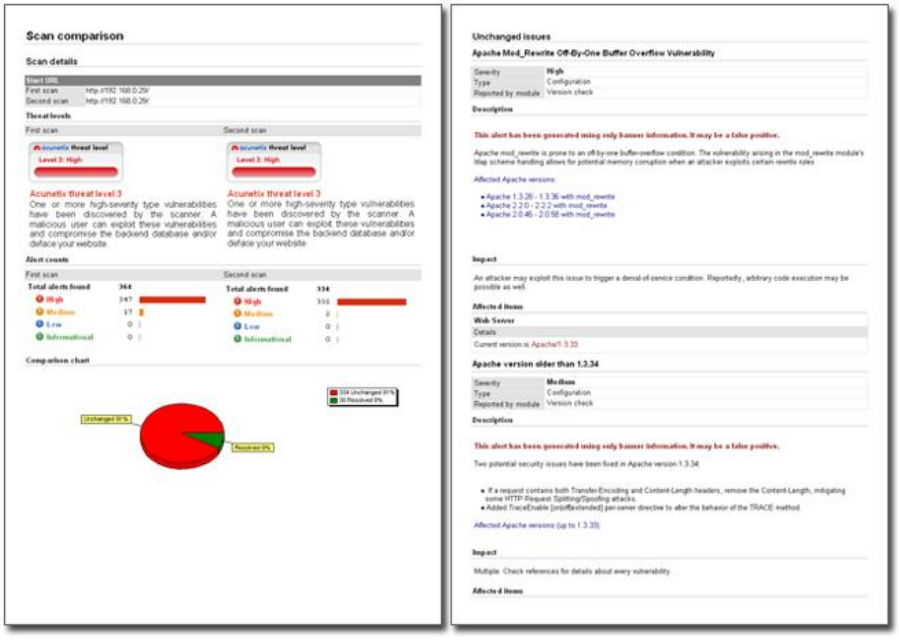
The Vulnerability report style presents a technical summary of the scan results and groups all the exploits according to their vulnerability class. Each vulnerability class contains information about the exposed pages, the attack headers and the specific test details.



Screenshot 46 – Vulnerability Report

Scan Comparison Report

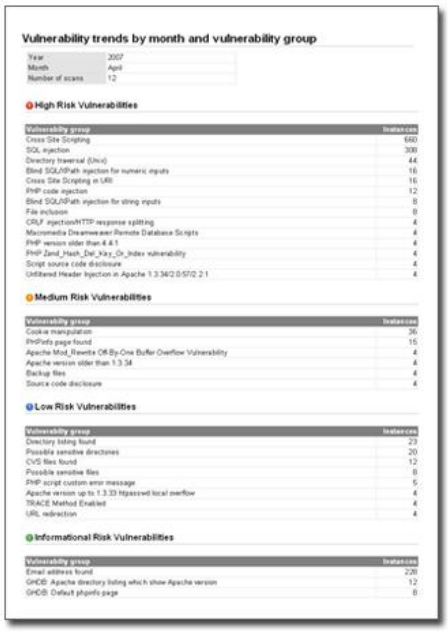
The Scan Comparison report template allows the user to document the changes tracked between 2 sets of scan results. This report will document resolved and unchanged exploits, and new vulnerability details. This report style makes it easy to periodically track development changes for a web application.



Screenshot 47 – Comparison Report

Statistical Reports

This set of reporting templates allows you to gather exploit information from the results database and present the information for periodical vulnerability statistics. This report style is particularly suitable for both developers and management to track security changes and to compile trend analysis reports.



Screenshot 48 – Statistical Report

Compliance Reports

This group of report styles allows you to generate a report according to the various compliance standard specifications. An easy to use wizard will prioritize and report specific vulnerabilities and exploits according to the standardized format as specified by the following compliance bodies; The Health Insurance Portability and Accountability Act (HIPAA), OWASP 2004 Top10, OWASP 2007 Top10, Payment Card Industry (PCI) standards, Sarbanes Oxley Act of 2002, and the Web Application Security Consortium Threat Classification.

Screenshot 49 – Compliance Report

Generating a Report

Once a scan is finished, click on the 'Report' button from the Web Scanner Toolbar. This will generated the configured default report-type from the scan result. Else launch the Acunetix WVS Reporter and proceed with the following steps:

Single Scan Report

1. Click on one of the 'Single Scan Template' sub-nodes from the Tools Explorer panel to select Developer, Executive Summary or Vulnerability Report and click on 'Report Wizard' button to launch the report wizard.

2. Configure the report filter to identify specific results, or leave the default selection to display all scan results and click 'Next' to proceed to select the specific scan for which to generate a report.

3. Select the scan and proceed to select what properties and details should the report include. Click on 'Generate' button to finalize the wizard and generate the report.

Comparison Report

1. Click on the 'Scan Comparison' sub-node under the Comparison Templates node from the Tools Explorer panel and click on 'Report Wizard' button to launch the report wizard.

2. Configure the report filter to identify specific results, or leave the default selection to display all scan results and click 'Next' to proceed to select the 1st scan which is going to be compared. Click 'Next' again to select the 2nd scan.

3. Once both scans have been selected, proceed to select what properties and details should the report include. Click on 'Generate' button to finalize the wizard and generate the report.

Statistical Templates

1. Click on one of the 'Monthly Vulnerabilities' sub-nodes from the Tools Explorer panel and click on 'Report Wizard' button to launch the report wizard.

2. Select for which year and month (number) the report should be generated for and click on 'Generate' button to finalize the wizard and generate the report.

Compliance Templates

1. Click on one of the 'Compliance Reports' sub-nodes from the Tools Explorer panel and click on 'Report Wizard' button to launch the report wizard.

2. Select which type of compliance template should be used for the report generation and proceed to choose the scan for which the report should be generated for. Click on 'Generate' button to finalize the wizard and generate the report.

The Report Viewer

The Repot Viewer allows you to view the generated report and also to save, export or print the report. The reports can be exported to PDF, HTML, Text, Word Document and BMP.

WVS Database

The Reporter can also be used to view the information on the database and the scan results saved in the database. To view such data click on 'WVS Database' node in the Tools Explorer.

By right clicking a scan, the user has the options to generate a report for the selected scan, to show information (such as scan time and duration, lists of vulnerabilities etc) about the selected scan and also delete the selected scan from the database.

The Reporter Settings

The Reporter settings allow you to configure the layout and style of the generated reports.

Report Options

This configuration node consists of two sections which can be used to customize the layout, titles and images in the headers of the report.

General Settings - Configure the default report style when generating a report directly from Acunetix WVS.

Report Options - Select custom icons, logos, headers and footer to customize the report.

You can use these settings to change the report layout to suit your needs and also to brand them for your own company. These settings are general default settings and will be used for all the reports generated with the WVS Reporter.

Page Settings

The page settings node allows you to configure the default page size, orientation and border dimensions of your reports. These settings are general default settings and will be used for all the reports generated with the WVS Reporter.

18. Command Line Support

Introduction

Acunetix WVS supports also command line. This have some advantages such as Acunetix WVS can be called from batch files and also through scripting languages, thereby allowing the user to automate repetitive tasks.

The WVS Console Scanner

The Acunetix WVS Console Scanner is installed with Acunetix WVS and can be accessed from the default installation folder of the application. The default location is: 'C:\Program Files\Acunetix\Web Vulnerability Scanner 6\wvs_console.exe'.

If the executable is run without parameters, usage information is presented together with all the details of every parameter and option accepted by the console application for your quick reference. For more WVS console scanner help, use the '/?' switch.

WVS Console Scanner Command Line Parameters

The Acunetix WVS Console Scanner supports most of the graphical user interface options and allows the same degree of customization and flexibility via a set of command line switches.

Acunetix WVS Console Scanner Parameters:

Parameter	Description
/scan	Scan a single website. **Syntax:** /scan [url] **Example:** /scan http://testphp.acunetix.com
/crawl	Crawl a single website. **Syntax:** /crawl [url] **Example:** /crawl http://testphp.acunetix.com
/scanfromcrawl	Start a scan from a saved crawl. **Syntax:** /scanfromcrawl [path to file\file name] **Example:** /scanfromcrawl c:\crawl\sitecrawl.cwl
/scanlist	Scan a group of websites defined in a text file of websites you want to scan. **Syntax:**

	/scanlist [path to file\file name] **Example:** /scanlist c:\lists\sites.txt
/scanwsdl	Start a web services scan. **Syntax:** /scanwsdl [wsdlurl] **Example:** /scanwsdl http://test.acunetix.com /service.asmx?WSDL
/profile	Use specified profile name during the scan. **Syntax:** /profile [profile name] **Example:** /profile default
/loginseq	Use specified login sequence during the scan. **Syntax:** /loginseq [filename] **Example:** /loginseq testphp_seq
/save	Save scan results to filename. **Syntax:** /save [filename] **Example:** /save c:\results\scan1.wvs
/exportxml	Exports results as. **Syntax:** /exportxml [filename] **Example:** /exportxml c:\results\scan1.xml
/exportavdl	Exports results as AVDL. **Syntax:** /exportavdl [filename] **Example:** /exportavdl c:\results\scan1.xml
/savetodatabase	Save scan results to reporting database. If this option is not specified, reports cannot be generated after the scan. **Syntax:** /savetodatabase
/savelogs	Save scan logs to the non default location. **Syntax:** /savelogs [filename] **Example:** /savelogs c:\logs\scan1logs.csv

/generatereport	Save the report of a scan directly to a directory. **Syntax:** /generatereport [dir] **Example:** /generatereport c:\reports\
/ReportFormat	Generate the report in one of the specified formats; REP, PDF, RTF, HTML etc. **Syntax:** /ReportFormat [format] **Example:** /ReportFormat PDF
/ReportExtraParams	Extra parameters for reporter, such as report template, compliance type etc. See the section Acunetix Reporter CLI reference for more information on this parameter. **Syntax:** /ReportExtraParams [parameter=value] **Example:** /ReportExtraParams "/r WVSComplianceReport.rep /k PCI12.xml"
/sendmail	When a scan finishes, an email will be sent using the details configured in the scheduler settings. **Syntax:** /sendmail
/verbose	Enable verbose mode; the log file entries will also be displayed in the command line window. **Syntax:** /verbose
/Password	Application password if password is required. Password can be enabled from the application settings. **Syntax:** /Password [password string] **Example:** /Password TestPass123!

WVS Console Scanner Command Line Options

Option	Description
--GetFirstOnly	Specify if the crawler should get the first URL only. **Syntax:** --GetFirstOnly=[true | false]
--RestrictToBaseFolder	Specify if crawler should fetch anything above start directory. **Syntax:** --RestrictToBaseFolder=[true | false]
--FetchSubdirs	Specify if the crawler should fetch directories below

	base directory. **Syntax:** --FetchSubdirs=[true | false]
--ForceFetchDirindex	Specify if the crawler should fetch directory indexes even if not linked. **Syntax:** --ForceFetchDirindex=[true | false]
--UseHTTPAuthentication	Use HTTP authentication when crawling the site. If this switch is used, AuthUser and AuthPass switches should be used to specify the credentials. **Syntax:** --UseHTTPAuthentication=[true | false]
--AuthUser	HTTP authentication username. The username passed with this option will be used with an NTLM login on the website if the option UseHTTPAuthentication is set to true. **Syntax:** --AuthUser=[username string]
--AuthPass	HTTP authentication password. The password passed with this option will be used with an NTLM login on the website if the option UseHTTPAuthentication is set to true. **Syntax:** --AuthPass=[password string]
--SubmitForms	Submit forms during crawl to discover links. **Syntax:** --SubmitForms=[true | false]
--RobotsTxt	Retrieve and process robots.txt and sitemap.xml during crawl to discover links. **Syntax:** --RobotsTxt=[true | false]
--CaseInsensitivePaths	Specify if the crawler should cater for case insensitive / sensitive paths. **Syntax:** --CaseInsensitivePaths=[true | false]
--UseCSA	Enable Client Script Analyzer engine to analyze JavaScript and other client side scripts during crawling. **Syntax:** --UseCSA=[true | false]
--scanningMode	Specify what scanning mode to use for this scan. Options available are Quick, Heuristic or extensive. **Syntax:** --scanningMode= [Quick | Heuristic | Extensive]
--TestWebAppsInAllDirs	Test for known web applications vulnerabilities in all directories. **Syntax:**

	--TestWebAppsInAllDirs=[True | False]
--ManipHTTPHeaders	Manipulate HTTP headers during scan. **Syntax:** --ManipHTTPHeaders=[True | False]
--UseAcuSensor	Enable AcuSensor technology for this scan. AcuSensor Technology sensor files must be installed on target. **Syntax:** --UseAcuSensor=[True | False]
--EnablePortScanning	Port scan target and run network alerts tests against target during scan. **Syntax:** --EnablePortScanning=[True | False]
--UseSensorDataFromCrawl	You can specify to use Sensor data from a saved crawl to proceed with scan or to re-crawl the target. **Syntax:** --UseSensorDataFromCrawl=[Yes | No | Revalidate]

Note: The only mandatory parameter is the scan URL. All other parameters and options are not mandatory. If not specified, the default graphical user interface settings will be used.

The Acunetix WVS console Reporter

The Acunetix WVS console Report is installed with Acunetix WVS and can be accessed from the default installation folder of the application. The default location is: 'C:\Program Files\Acunetix\Web Vulnerability Scanner 6\reporter_console.exe'.

If the executable is run without parameters, usage information is presented together with all the details of every parameter and option accepted by the console application for your quick reference. For more WVS console Reporter help, use the '/?' switch.

The Acunetix WVS console Reporter command line options

Option	Description
/v or /View	View a *.pre format report in the Acunetix reporter. **Syntax:** /v [path to report] **Example:** /v c:\report.pre
/o or /Output	The destination path where the report should be generated and the filename of the report. **Syntax:** /o [path to report destination/report name] **Example:** /o c:\reports\report
/r or /Report	Specify the report template to use for generating the report.

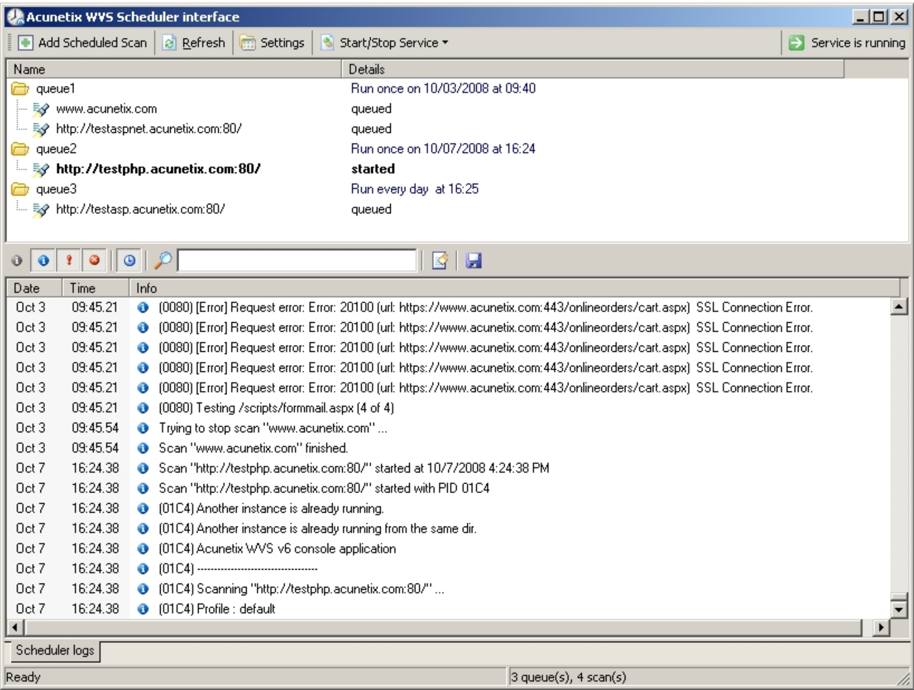
	Available report templates: WVSComplianceReport.rep: Compliance report. WVSDeveloperReport.rep: Developer report. WVSScanCompare.rep: Scan comparison report. WVSSingleScan.rep: Detailed Scan report. WVSSingleScanExecutive.rep: Executive Summary WVSVulnGroupTrends.rep: Monthly Vulnerabilities report. **Syntax:** /r [report template] **Example:** /r WVSDeveloperReport.rep
/k or /Kind	This parameter may be used only for compliance type reports. In fact, such parameter should only be used when the /r or /Report switches are set to WVSComplianceReport.rep. Specify which type of compliance report from the following: HIPAA.xml: The Health Insurance Portability and Accountability Act (HIPAA) NIST_SP800_53.xml: NIST special publication 800-53. OWASP_Top_10_2004.xml: OWASP Top 10 Vulnerabilities 2004. OWASP_Top_10_2007.xml: OWASP Top 10 vulnerabilities 2007. PCI.xml: Payment Card Industry Data Security Standard version 1.1 (PCI DSS) PCI12.xml: Payment Card Industry Data Security Standard version 1.2 (PCI DSS) Sarbanes_Oxley.xml: Sarbanes-Oxley Act of 2002. WASC_Threat_Classification.xml: Web Application Security Consortium: Threat Classification. **Syntax:** /r WVSComplianceReport.rep /k [compliance type template] **Example:** /r WVSComplianceReport.rep /k PCI12.xml
/p or /Password	Application password, if Acunetix WVS GUI application is protected with a startup password. **Syntax:** /p [password]
/c or /Console	Do not load Acunetix Reporter user interface. If this option is not specified, the user interface of the Acunetix Reporter will load. **Syntax:** /c
/a or /Action	Specify the file type in which the exported report should be generated. File types available: PDF, RTF, HTML, REP (Acunetix WVS proprietary format). **Syntax:** /a [format type]

	Example: /a PDF
/p or /Parameters	For each type or report template, there are different parameters. If no parameters are specified, the default will be used. To specify the parameters to be passed to the report, us the “name=value” format delimited by “;”. To find out what parameters are available for each type report template, use the following syntax: Reporter_console.exe /r ReportTemplate /? **Syntax:** /r [report template] /p [parameter=True/False] **Usage Example:** /r WVSSingleScan.rep /p "ShowHTTP=False "
/t or /Target	Scan identifiers from the database to use as a report source. From the Acunetix WVS reporter, in the Configuration > WVS Database node, one can find the ID for each scan stored in the reporting database. Can be one integer for single target templates, two integers for comparison templates delimited by “;”. Can also be omitted for reports without specific scan target. For single scan templates, you can use “last” as target to indicate the last saved scan from the database. **Syntax:** /t [report ID] **Example:** /t 24

19. Scheduler

Introduction

The Scheduler application allows you to schedule scans; neither the WVS nor the Acunetix WVS Scheduler Interface needs to be running for the scans to launch at the scheduled.



Screenshot 50 – Acunetix WVS Scheduler

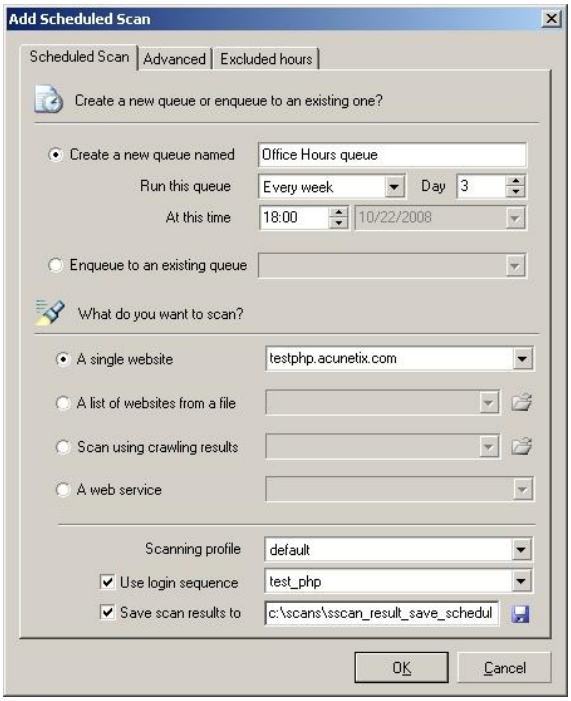
Creating a Scheduled scan

NOTE: First a queue with a specific schedule must be created and then scans are assigned to the queue. Therefore schedule time and exclude hours must be set for a queue. Multiple scans will run sequentially in a queue and they cannot run simultaneously. Therefore the schedule is set for a particular queue and not for a particular scan. Multiple queues with different schedules can be created with multiple scans assigned to each queue.

Creating a queue and a schedule

1. Start the Scheduler interface by clicking on the Scheduler Icon  on the toolbar in the main program interface or the Acunetix program group and click on the 'Add scheduled Scan' button.

2. Enter a queue name in the field 'Create a new queue named' to create a new queue or tick 'Enqueue to an existing Queue' to add the new scan to an already existing queue. Select the scheduled recurrence of the queue from: Once, Every Day, Every Week, Every Month or Continuous. Set a specific day number if schedule is set to weekly or monthly, e.g. 2nd day of the week or 21st day of the month.



Screenshot 51 - Creating a schedule

3. You can specify the hours when an ongoing scan should be paused by clicking on 'Excluded hours' tab, ticking 'Enable Excluded Hours" and highlight in red where you do not want the scan to run. This is useful if you want to stop scanning of your website during normal business hours.

Note: If a scan is still running during excluded hours, the scan will be automatically paused and resumed again in the next available hours when scanning is allowed.

Configuring the scan

3. In the 'Scheduled Scan' specify the target by specifying a URL, or a text file containing a list of URL's, or a saved crawl result or a web service.

4. Select also which scanning profile should be used to scan the target, specify a Logon sequence if needed and also a location where the scan results should be saved to. Click 'OK' to save the scheduled job.

Advanced Options tab

In the Advanced Tab the following options are available:

- Save options
- Logging options
- Scan options (e.g. scanning mode, AcuSensor technology etc)
- Crawling options
- Reporting options

Scheduler Settings

In the general Tab the options to Start Scheduler Interface on Windows startup and to minimize interface to system tray are available.

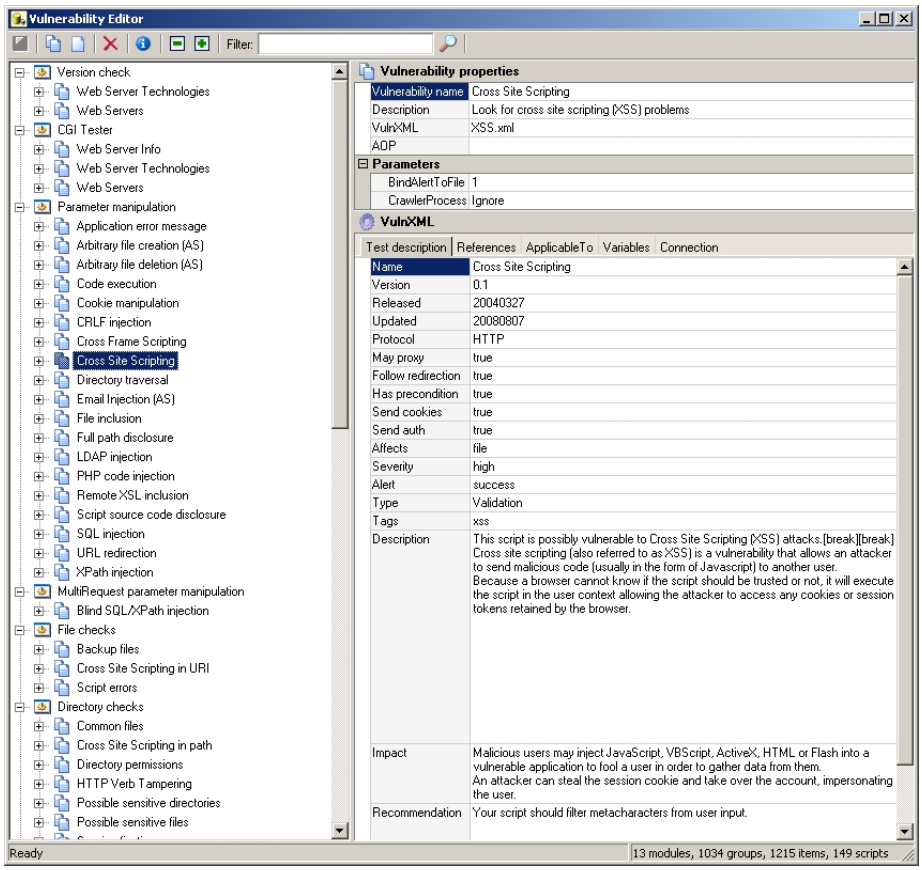
In the Email Notifications tab the user can specify email settings such as SMTP server IP or FQDN and port, SMTP server authentication (optional) and email addresses which will be used for notification.

Scheduled Scan Controls

By right clicking a running scan, you can select to pause or resume the scan. A paused scan will not be automatically resumed. You need to manually resume the scan. You can also stop the running scan and save partial scan results by right clicking the scan and select ‘Stop Scan (save partial results).

20. Vulnerability Editor

Introduction



Screenshot 52 – The vulnerability editor

The Vulnerability Editor allows you to edit the database which contains the definitions of all the vulnerability tests that can be performed during an audit. You can create a new or edit an existing vulnerability. You can start the Vulnerability Editor from the Acunetix program group in the Start Menu, or by clicking 'Tools > Vulnerability Editor' on the file menu within the Acunetix WVS user interface.

Note:

Be careful when editing the vulnerability checks; these are core to Acunetix WVS and can corrupt the Acunetix WVS installation. All tests are organized into 6 main nodes, each node being the particular module that performs the actual audit tests. Below each node, you can create vulnerabilities and vulnerability items. Vulnerabilities are stored in a modified version of the VulnXML format – a web vulnerability standard defined by the OWASP group.

Also, changes to built-in vulnerabilities are not retained during upgrade or updates (if that particular file is updated). Once an upgrade or update takes place, a backup copy (.bak file) is created in the vulnerabilities directory. The default path of this directory is 'C:\Program Files\Acunetix\Web Vulnerability Scanner 6\Data\Profiles\VulnXML'. If you want to retain the changes, redo the changes once the updates or upgrade is finished.

If a new vulnerability is added or an existing one is modified, you need to restart Acunetix WVS or press CTRL+ALT+R to reload all the modules in Acunetix WVS.

Acunetix WVS audit modules

Acunetix includes the following auditing modules:

Version Check - TM_ Version_Check.dll

This module analyses the server banners to determine web server versions (e.g., Apache, IIS etc) and different technologies used (e.g., PHP, mod_ssl, etc.). The version is then compared to the database of vulnerable versions of that particular software. With the version check module you can create checks for more recent versions or for other types of software, thereby allowing for checks in spite of which vulnerable version is installed.

CGI Tester - TM_CGI_Tester.dll

The CGI tester module will search for common CGI scripts. It can also be used to determine both the presence of sensitive files on the web server (e.g., the Apache manual directory) and the methods allowed on the web server (e.g., GET, PUT, DELETE etc.). It is similar to a CGI scanner, but with a configurable XML interface.

Parameter Manipulation - TM_parameter_manipulation.dll

This module will try to manipulate the inputs of a file (like server side script), to test common vulnerabilities e.g. SQL injection, XSS 'Cross site scripting', command execution and many others.

File Checks - TM_Backup_Files.dll

This module analyses "interesting" files within the web structure (e.g. Files with input parameters and probable scripts) and will try to manipulate their names in the http requests. For example, if WVS finds a file 'login.php', the module will try to look for files such as 'login.php.bak', 'login.bak', 'login.zip' etc. You can add new file parameters and different extensions using the Vulnerability Editor.

Directory Checks - TM_Common_Files.dll

This module scans for common files left in directories of the website or web application being scanned. It will look at the structure of the website and will try to request files and directories that should not be there. For example, a common file present on PHP-based websites would be phpinfo.php, which displays information about the PHP configuration on that server.

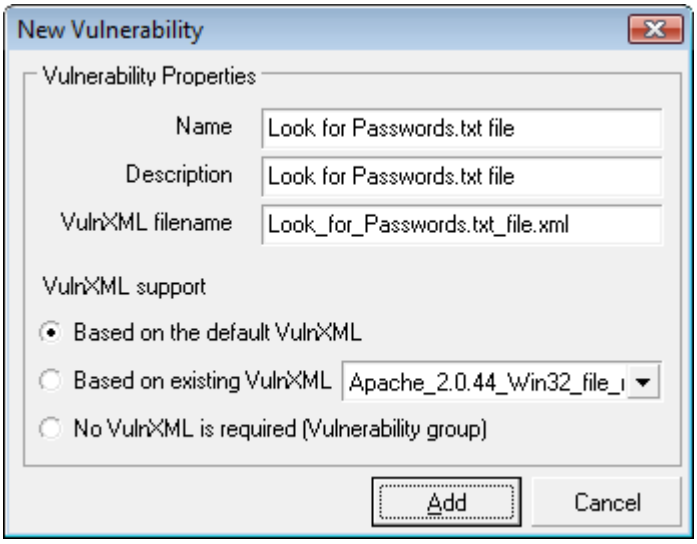
Text Search - TM_ Text_Search.dll

The text search will look for certain texts within the files/filenames retrieved from the web server. It will search for remarks left by the web administrator, including username and password information.

Adding a Vulnerability

To add a new vulnerability:

1. Right click on an existing module or Vulnerability and select 'Add Vulnerability'.



Screenshot 53 - New Group details

2. Specify the name of the Vulnerability, a short description and the name of the VulnXML file where the test parameters will be stored.

3. Specify whether the test must be based on VulnXML or not:

- **Based Default VulnXML** - Uses the default/built-in VulnXML test parameters.
- **Based on existing VulnXML** - Copies the test parameters from an existing VulnXML file.
- **No VulnXML is required** - Used if the test does not perform any HTTP requests but only specifies the condition which will make it successful. (E.g. tests in the 'Version Checks' module, only specify a VersionRegex parameter. The test is successful if the VersionRegex value matches the target web server banner).

4. Click on the 'Add' button to create the new Vulnerability.

5. Now click on the created Vulnerability to bring up the details in the Vulnerability properties page (the right hand pane), which contains the Vulnerability Properties, the Parameters and the VulnXML sections. The properties are the ones already set when you created the new vulnerability.

6. Set the vulnerability property **AOP** to 1 if the vulnerability being configured will be using AcuSensor Technology. The default setting is '0', i.e. the vulnerability will not be using AcuSensor Technology.

Vulnerability Parameters

Different vulnerability classes have different parameters. Below is a list of all parameters available. When a vulnerability is highlighted, only the parameters which apply will be displayed.

- **Affects** - Identifies the object which is affected by this test, for example details about a Web Server (e.g. if the vulnerability effects the web server), a file or an object which is identified by the module (when set_by_module is specified). This parameter is dependent on the type of test being carried out.
- **BindAlertToFile** - Set this to '1' to enable the test to add any new discovered files to the crawler directory structure for use in future scans.
- **CrawlerProcessingMode** – From this parameter you can configure the action the crawler should take when discovering new files while launching this vulnerability check. Set to 'Ignore' to ignore all discovered links. Set

it to 'ParseOnly' if you want the crawler to parse the links from the discovered file and try to discover more files from it (but they won't be added to the file structure in the Scan Results tab). Set to 'FullProcess' to discover more files from it, get the file as is and add the file to the site structure in the Scan Results tab, as if it was discovered from the crawler.

- **StopAtFirstMatch** – Set this parameter to '1' if you want the scanner to stop checking other files on the target web application for the same vulnerability once it is discovered. Else set this parameter to '0'.
- **SearchVariations** – Set this parameter to '1' if you want to test different variants of this vulnerability once it is discovered.
- **SearchOnlyHTTP** – If you want to launch this attack on HTTP sites only, and not HTTPS, set this parameter to '1'.
- **AlertIfTextNotFound** – This parameter applies to text search checks only. Set this parameter to '1' if you would like to be alerted if the specified text or regular expression is not matched on the target.

Editing test parameters in VulnXML section

This section is organized into 5 subsections, each represented by a tab each of which is described in the subsequent subsections:

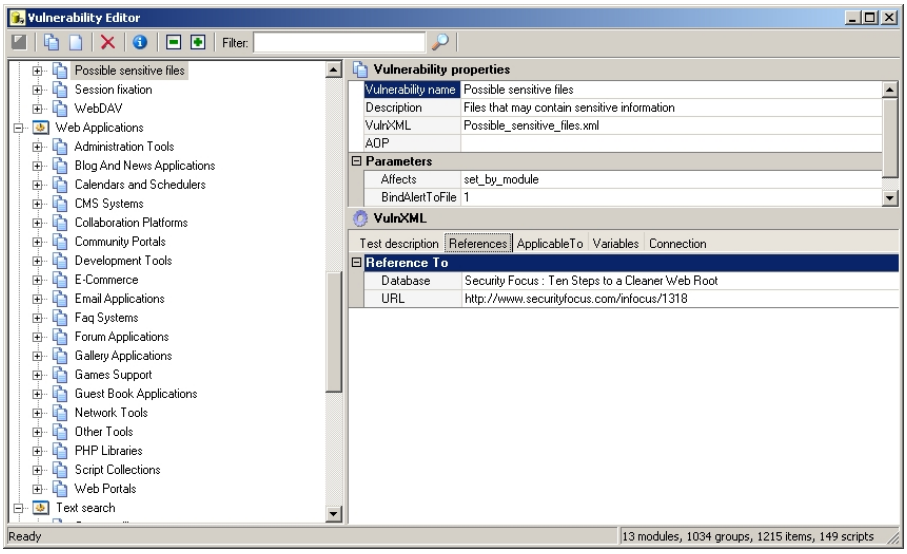
- **Test Description** Tab - Edit generic information.
- **References** Tab - Specify links to be added to the additional information section when reporting the vulnerability.
- **Applicable To** Tab - Specify for which operating systems, web servers or technologies you want this test to be performed.
- **Variables** Tab - Create/edit variables to be used by the test.
- **Connection** Tab - Specify what HTTP requests should be made, what response to look for and what defines success or failure of the test.

Editing the Vulnerability Description

In the vulnerability 'Test Description' tab you can edit generic information:

- **Name** -The name of the vulnerability (e.g., could be the same as the name given to the VulnXML file.).
- **Version** - Test Version number.
- **Released** - Date showing when this Test/Vulnerability was created (yyyy/mm/dd).
- **Updated** - Date showing the last time that this Test/Vulnerability was updated (yyyy/mm/dd).
- **Protocol** - Defines the Protocol that this test will use for sending request to a target during a scan (i.e. HTTP).
- **May Proxy** - Defines whether this test may be performed through a proxy server. If Acunetix WVS is configured to use a proxy server, set this option to true to execute the test.
- **Follow Redirection** – Specify if a discovered redirection (link) should be followed when launching the vulnerability check.
- **Has Precondition** – For some specific vulnerability checks like Cross Site Scripting checks, there is a precondition. The precondition is always the first item in the vulnerabilities group and its item name is "precondition". Set 'Has Precondition' to 'True' if you do not want to run this check unless the precondition is met.

- **Send Cookies** – If the site being scanned utilizes cookies, you can configure the vulnerability check to send the cookies or not, by setting the 'Send Cookie' value to 'True' or 'False' respectively.
- **Validate Session** – If the vulnerability check is running in a validated session, and for any reason the session is invalidated, you can configure this option to 'True' if you would want the scanner to re-launch the login sequence and have a valid session again. Set to 'False' if once the session is invalidated, you do not want the scanner to re-validate the session.
- **Send Auth** – This applies for HTTP authentication only. If during the configured vulnerability check HTTP authentication is required, if set to 'True' the scanner will send authentication details. If this option is set to 'False', once asked for HTTP authentication the scanner will not send the authentication details.
- **Affects** - Defines which components of the target site structure will be tested.
- **Severity** - Defines the vulnerability level of a target should this test fail (i.e. High Severity indicates that if this test generates failures, the target being scanned has a severe vulnerability).
- **Alert** - Defines whether the Alert is to be triggered on success or failure of the test.
- **Description** - Contains the test function description.
- **Impact** - Contains information on the effect that the vulnerability detected by this test has on your target site.
- **Recommendation** - Contains information on what you should do to eliminate the vulnerability detected by this test.



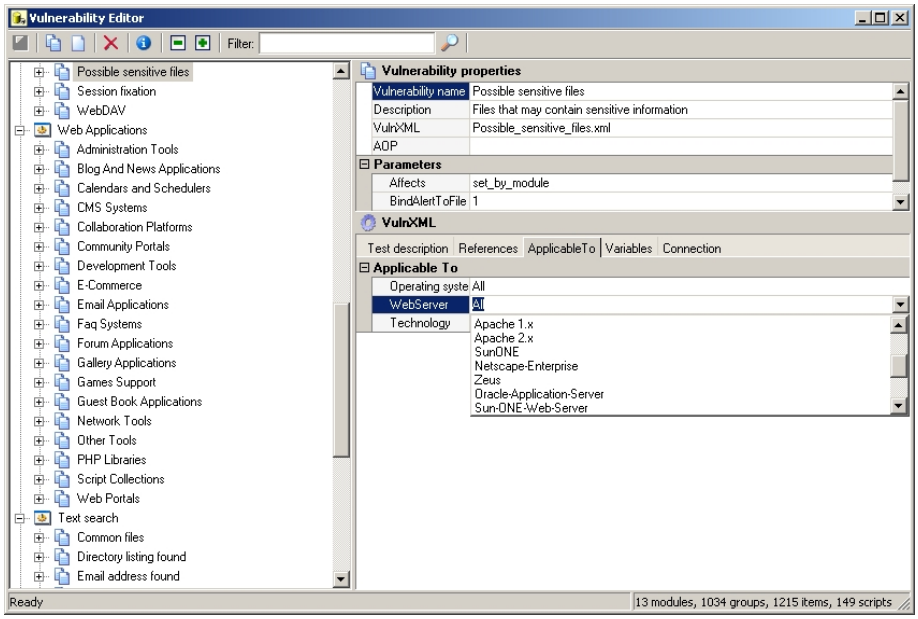
Screenshot 54 - References tab page

In the 'References' tab you can specify links to additional information about the vulnerability (e.g., cause and related fix).

- **Database** - Specify the Link heading/title of the article/information.
- **URL** - Contains the URL.

You can add additional references by right clicking and selecting 'Add reference'.

Specifying When the Vulnerability Check is Applicable



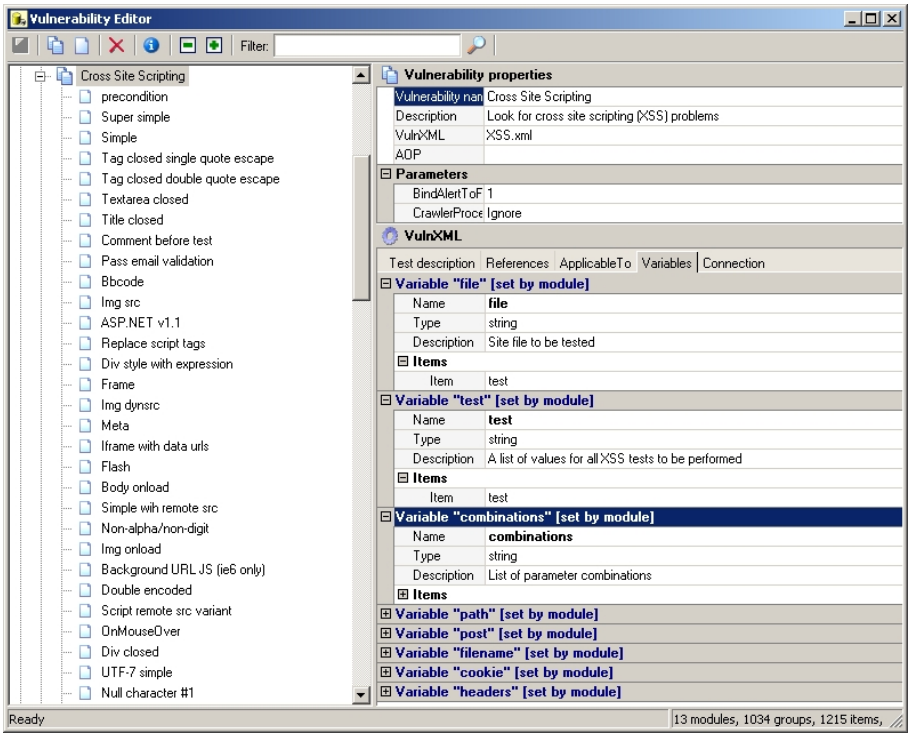
Screenshot 55 - Applicable to tab

In the 'ApplicableTo' tab you can specify for which operating systems, web servers or web technologies you want this test to be performed. The test will only be performed if all of the conditions are true.

- **Operating System** - Defines the Operating systems. You can choose Windows, Unix/Linux or all.
- **Web Server** - Defines which Web Server types must be checked using by this test. For example Apache, IIS etc.
- **Technology** - Define which technologies (e.g. ASP/PHP) must be checked by this test.

You can add additional conditions by right-clicking and selecting 'Add applicable to'.

Specifying Test Variables



Screenshot 56 - Variables page

In the 'Variables' tab you can create/edit variables to be used by the test, which can be specified in the 'Request', 'Response' and 'TestCriteria' sub-nodes in the 'Connection Tab'. The types of variables that you can create are dependent on which module is performing the test. For example, if creating a vulnerability check within the CGI Tester node, only the File variable will be available. The following is a list of variables that each module supports:

Version Check

- No variables

CGI Tester

- No variables

Parameter Manipulation

- **File** - The site file to be tested (e.g. /dir/a.asp).
- **Test** - This specifies that it should perform the check for each parameter created under 'Vulnerability parameters'.
- **Combinations** - This will contain all the combinations of parameter values, for example: ?param1=${test}&param2=1, ?param1=1&param2=${test}.
- **Path** - The actual URL for the test, for example ${file}${combinations}.
- **Post** - Same as combinations but for POST variables.
- **Filename** - Same as file however it does not include the path, only the filename, for example a.asp.

File Checks

- **File** - The site file to be tested (e.g. /dir/a.asp).

- **Test** - This specifies that it should perform the check for each parameter created under 'Vulnerability parameters'.
- **Path** - The actual URL for the test, for example ${file}${test}.

Directory Checks

- **File** - The site file to be tested (e.g. /dir/a.asp).
- **Test** - This specifies that it should perform the check for each parameter created under 'Vulnerability parameters'.
- **Path** – The actual URL for the test, for example ${directory}${test}.

Text Search

- no variables

Variables Explained

Defining the variables is the 'hardest part' in creating a vulnerability check and is best explained using an example such a SQL injection check. For example we have a website with 1 file: '/dir1/a.asp'. On that file, we want to create an HTTP request with a " ' " and a "1" character. We would setup this vulnerability check with these variables:

- **File** - /dir1/a.asp
- **Test** - ' (a single quote)
- **Combinations** -?param1=${test}&param2=1, ?param1=1&param2={test}
- **Path** - ${file}${combinations}
- **Post** - <empty>
- **Filename** - a.asp

With these variables, the vulnerability will be executed with the following request:

${scheme}://${host}:${port}${path} - scheme, host, port are default variables that will contain the values of currently scanned website. e.g. scheme=http://, host=testwebsite.com, port=80

Path is defined as ${file}${combinations}, so it will be evaluated as /dir1/a.asp${combinations}

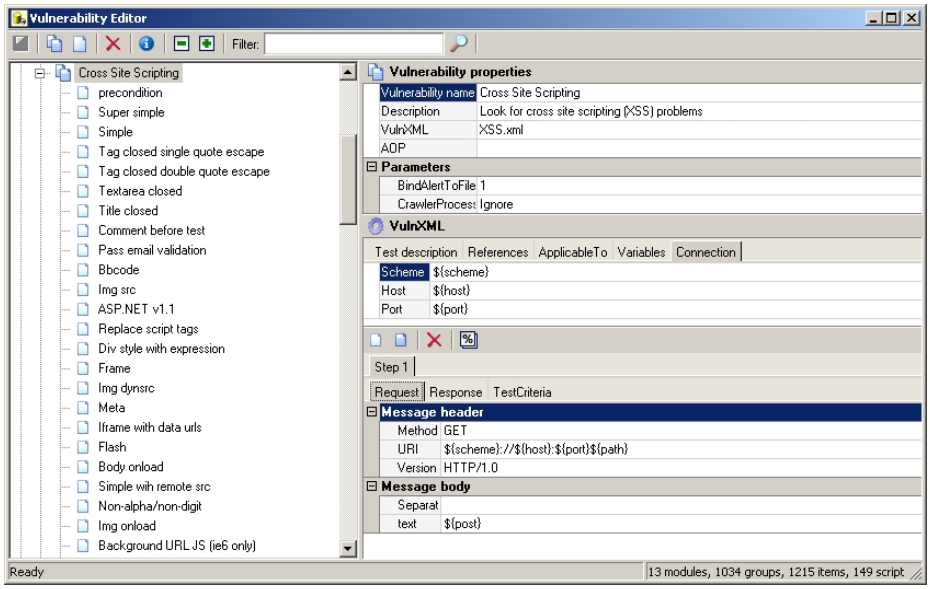
${combinations} is ?param1=${test}&param2=1, ${test} is ', so, in the end we will have 2 requests:

/dir1/a.asp?param1='&param2=1

/dir1/a.asp?param1=1&param2='

You can edit the existing variables or add new ones. To create a new variable, Right-click on the Variable page and select 'Add Variable'. To delete a user-created variable right click on the variable name and select 'Delete'.

Note: Default module variables cannot be deleted.



Screenshot 57 - Connection tab sub-tabs

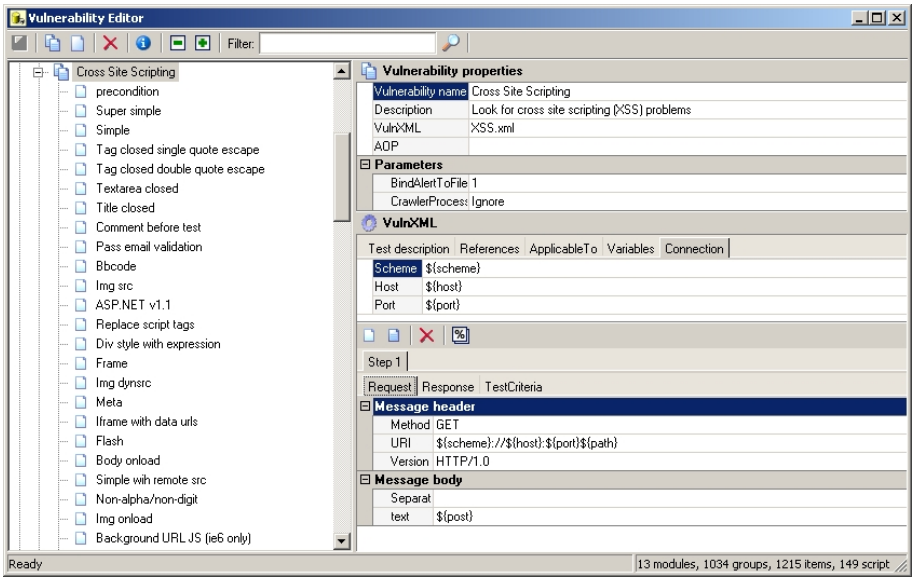
In the 'Connection' tab you can specify what HTTP requests should be made, what response to look for and what defines success or failure of the test. These parameters are set via the 'Connection', 'Request', 'Response' and 'TestCriteria' sub-tabs. It's usually not necessary to modify the connection sub-tab, since the test will automatically use the scheme, hostname and port of the active scan. However you can choose to specify a custom connection scheme (HTTP/HTTPS), host name and the port for the test.

Variable Types

There are four variable types which you can configure:

- **String** – the variable type is of string value
- **Int** – the variable type is of integer value
- **htmlparser** – the content of the variable is needed to be parsed from the html parser
- **aspectdata** – the variable type is of Acunetix AcuSensor Technology data type (i.e. the content should be filled from the AcuSensor Technology).

Defining the Requests to be Made in the Test



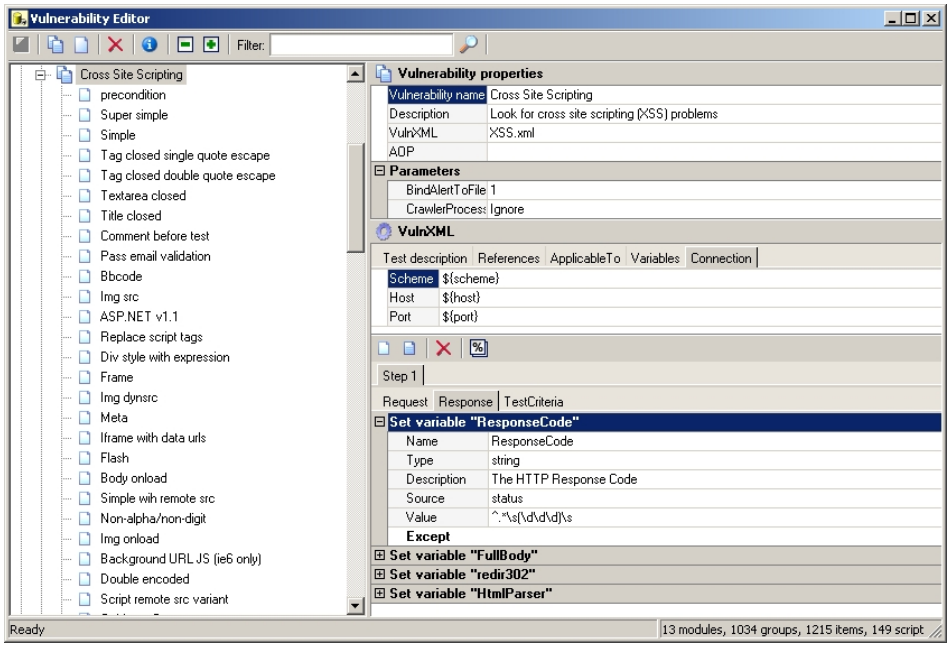
Screenshot 58 - Request sub-tab page

In the Request sub-tab, you must specify the exact HTTPrequest to be made:

- **Message header**
  - **Method** - Define HTTP request method, e.g. GET, POST, HEAD and PUT.
  - **URI** - Define the destination of the request. The URI parameter is by default set to path since this variable encloses the value of variables $file$test. This means that the path variable will be set to various combinations of $file and $test according to the request and target website being scanned.
  - **Version** - Define the HTTP protocol version to be used for the request, e.g. HTTP/1.0 or HTTP/1.1.
- **Message body**
  - **Separator -** Specify the separator.
  - **Text -** Specify the text for the body or a variable, e.g. ${post}.

Note: The URI is not necessarily a URL. For more information on subject, please refer to http://www.pierobon.org/iis/url.htm.

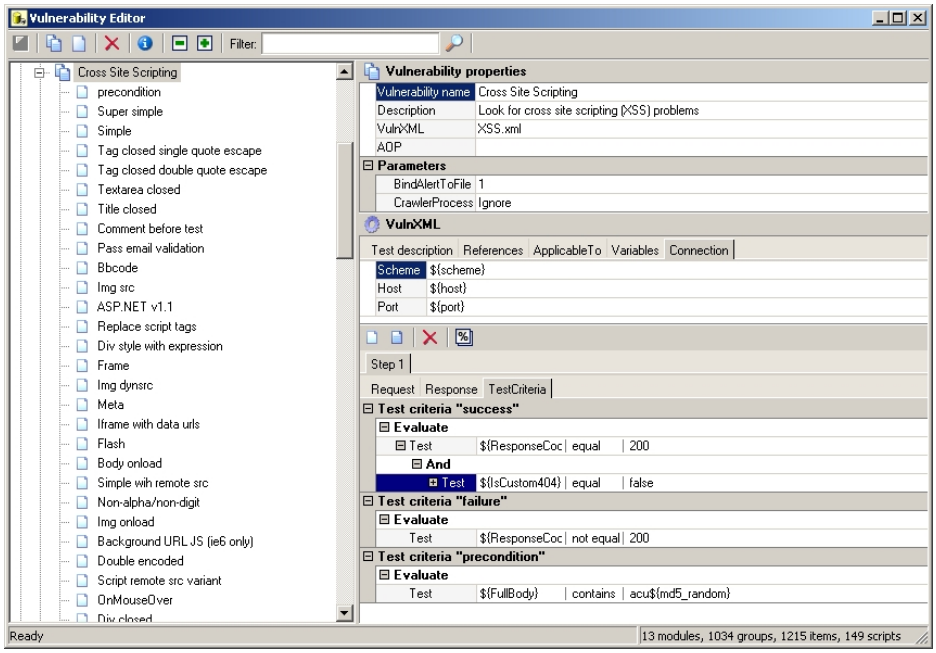
Analyzing the Response



Screenshot 59 - Response sub-tab page *In this response tab you can edit/create the responses that the test should look for.*

Define

- **Name** - Variable name.
- **Type** - Variable type.
- **Description** - Variable description.
- **Source** - Specify where to apply the regular expression (on status code, on response headers or response body).
- **Value** - Specify the regular expression used to extract the variable value from the source. Defining the test criteria/conditions.



Screenshot 60 – Test Criteria sub-tab page

The last step is to define what conditions cause the success or failure of the test. You can add 'failure' or 'success' conditions: If a failure condition evaluates to true, then the test fails. If a success condition evaluates to true then the test passes.

You can create multiple success or failure conditions: If any of the failure conditions evaluates to true, independent of the other conditions, then the test fails. If you add a success condition, then the success condition must evaluate to true for the test to pass.

You can use equal, not equal, contains, not contains, lower than and greater than operators in a condition.

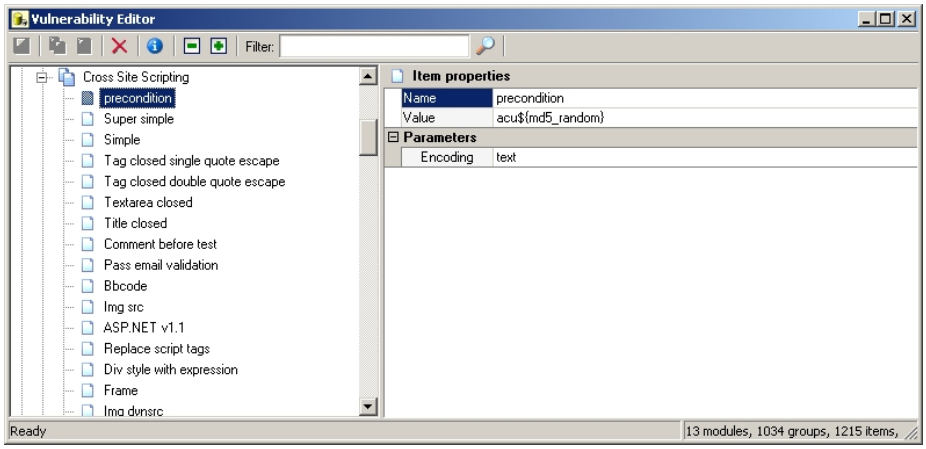
To create a new test criteria, right click and select 'Add test criteria – success' to create a success condition or 'Add test criteria – failure' to create a failure condition.

After you have created the vulnerability, click on the Save '  ' button in the Tool bar to save the test information. Now close Acunetix WVS, including the vulnerability editor and launch it again to perform the test. You will need to enable the test first in one of the scanning profiles. You can do this from the Configuring > Scanning Profiles node.

Adding a Vulnerability Item

Vulnerability items are additional parameters which Vulnerabilities require during a scan. Vulnerability items are kept within the relative Vulnerability and can be created as follows:

1. Right click on the Vulnerability w here you want to create the new Vulnerability parameter and select 'Add Vulnerability item'.



Screenshot 61 - Vulnerability parameter parameters

2. In the Item Properties, define the 'Name' (i.e. the Item name) and 'Value' (e.g. a file name) that will be attributed to this parameter.

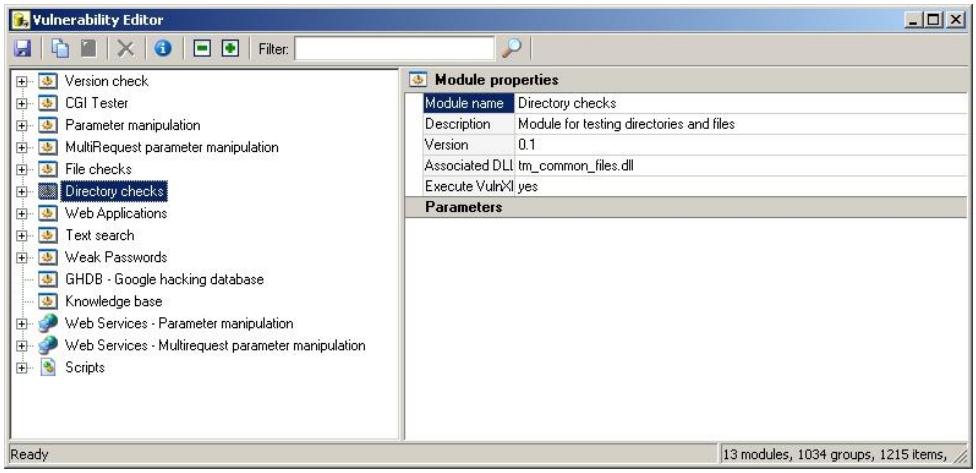
3. Click on the save '  ' button in the Toolbar at the top of the Vulnerability editor window. This will save the new Vulnerability item which will be referenced by the test variable.

Example: Creating a Test Which Searches for a Particular File

In this section we will present a walk-through of the process of creating a new vulnerability check – in this case looking for a file called "passwords.txt".

Step 1: Creating a Vulnerability

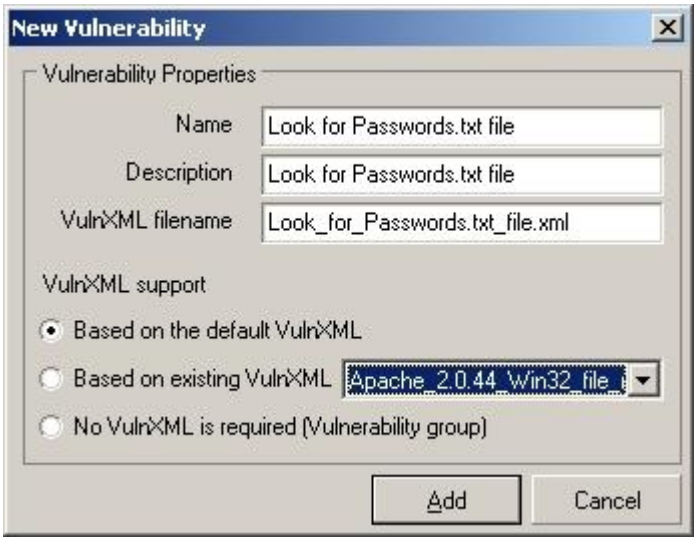
Create a new Vulnerability. We will call it "Look for Passwords.txt file".



Screenshot 62 – Vulnerability Editor Modules

1. Launch the Vulnerability Editor from Acunetix WVS.

2. Since we are looking for a file in any of the site's directories, we will use the Directory check module. Click on the 'Directory Checks' node, right-click and select 'Add vulnerability'.



Screenshot 63 - New group properties window

3. In the New vulnerability dialog, specify the following details:

- **Name -** 'Look for a Passwords.txt. file'
- **Description** - This test will scan the target site and look for a file called passwords.txt
- **VulnXML -** Leave default suggested filename
- **VulnXML support -** Based on Default VulnXML.

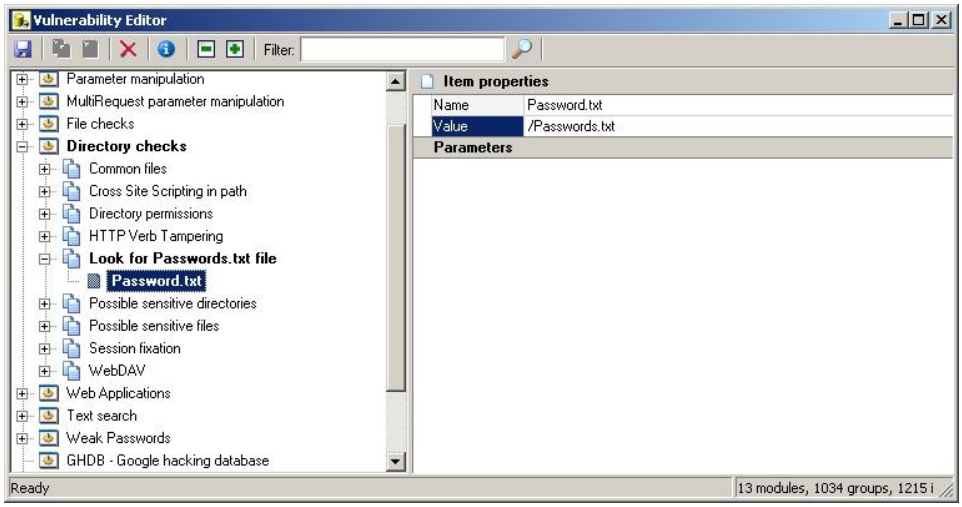
Click on the 'Add' button to create the new Vulnerability. It will be listed under the 'Directory Checks' node.

Step 2: Adding a Vulnerability Item

Now that we have created the test, we need to define the parameters of the test. This is done by creating a Vulnerability item.

In this example, we need to create a Vulnerability parameter which contains the name of the file to be searched for (i.e. passwords.txt):

1. Right click on the Look for Passwords.txt Vulnerability, right-click and select 'Add vulnerability item'.



Screenshot 64 - Creating an item

2. In the Item properties section, specify the following information:

   - **Name** - Password.txt
   - **Value** -/Passwords.txt

   The web scanner will now look for a file called Passwords.txt in all the directories it finds. E.g. Assume that the crawler finds 2 directories "*/secured*" and "*/*" after scanning a target site. Based on the value of the ${path} variable (in the VulnXML file properties) and the corresponding *Vulnerability parameter* value, it will look for:

   "/passwords.txt"

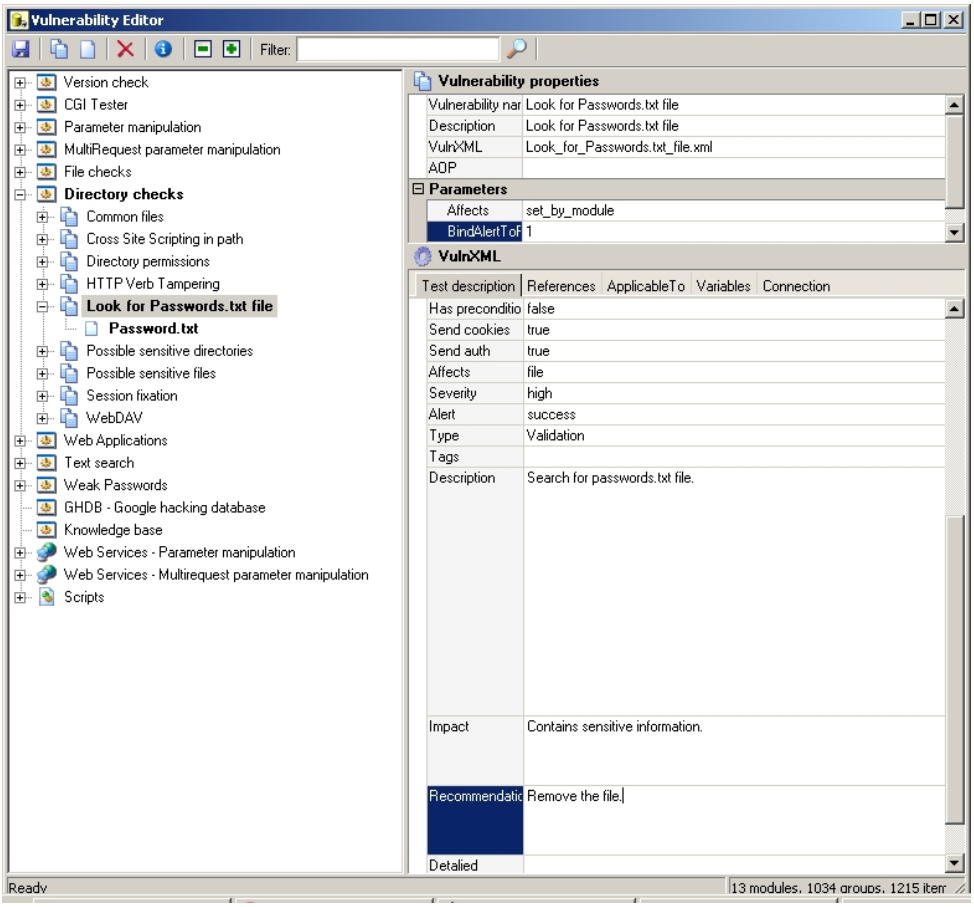
   "/secured/passwords.txt".

3. Click on the save button to save the new Vulnerability parameter.

Step 3: Configuring the Test Properties

Now we need to configure the test properties:

1. Click on the Look for the Passwords.txt Vulnerability.
2. In the parameters section, leave the *Affects* and *BindAlertToFile* as default (i.e. "set_by_module" and "1" respectively).



Screenshot 65 - Specifying the test description

3. In the VulnXML section, specify the following details for these fields in the test description tab:

   - **Name** - Look for Passwords.txt file.
   - **Affects** - File
   - **Severity** - High
   - **Alert** - Success (i.e. alert is generated if file is found).
   - **Description** - Search for passwords.txt file.
   - **Impact** - Contains sensitive information.
   - **Recommendation** - Delete the file.

4. Alternatively, in the 'References' tab, specify any references on the web to the vulnerability:

   - **Database** - Link Title.
   - **URL** - Full URL to the reference.

5. In the 'Applicable To' tab, leave the settings as default, since checking for the file independent of the web server, operating system or technology used.

6. In the 'variables' tab specify the variables of the test. The Directory checks module makes use of three variables called *File*, *Test* and *Path*.

- The 'File' variable value is automatically set by the scanner for every directory it finds.
- The 'Test' variable is retrieved from the Vulnerability parameter created previously. In our example, the "test" variable will contain "/Passwords.txt" which is the value specified when having added a new

Vulnerability parameter (i.e. in our example the Vulnerability parameter is called “The Pword file to be searched” and is the sub node that we have added to the ‘Look for passwords.txt file’ Vulnerability).

- The ‘Path’ variable value is set by combining the values of $file$test explained above.

However, since already having created the vulnerability item which is referenced by the test variable, there is no need to make any changes in this dialog.

7. In the Connection tab specify the HTTP requests and the success/fail criteria that this test will make. Since there is no need to make any specific HTTP requests in this example, leave the values of the Connection tab default.

Step 4: Save the Test and Re-Launch Acunetix WVS

1. Click on the save button to save the Vulnerability check and close the Vulnerability Editor as well as Acunetix WVS.

2. Launch Acunetix WVS again and check if the new Vulnerability has been added to the scanning profiles by clicking on the ’Scanning Profiles’ located in the ‘Configuration’ node.

3. Mark the box at the left of the new test in order to enable the use of the new test in the next scan. Click on the ‘Web Scanner’ node, specify a target and start a scan so that you can check the new test.

4. If the test identifies the file, then it will be displayed in the alerts node during a scan.

21. Troubleshooting

Introduction

The troubleshooting guide explains how you should go about resolving the issues that may result.

The main sources of information available to users are:

- The Manual - most issues can be solved by reading the manual.
- Email Support - contact the Acunetix support department by email at support@acunetix.com
- The Acunetix Support Center – http://www.acunetix.com/support

Request Support via E-Mail

If you have problems that you cannot resolve, please contact the Acunetix support department. The best way to do this is via e-mail, since you can include vital information to enable us to solve the issues you have more quickly.

The 'Troubleshooter' included in the program group, automatically generates a number of files needed for Acunetix to provide technical support. The files would include the configuration settings etc. To generate these files, start the troubleshooter and follow the instructions in the application.

In addition to collecting all the information, the troubleshooter will also ask you several questions. Answer these questions accurately as without proper information it will not be possible to diagnose your problem.

Then go to the support directory, located under the main program directory, **ZIP the files** and send the generated files to support@acunetix.com.

We will answer your query within 24 hours or less, depending on your time zone. We will try our best to resolve the issue as quickly as possible.

Credits

Acunetix Web Vulnerability Scanner uses technology from the following entities:

- OpenSSL Project (http://www.openssl.org/). The product contains and uses the unmodified version of OpenSSL 0.9.7c.
- PCRE Wrapper for Delphi (http://renatomancuso.com/software/dpcre/dpcre.htm) based on PCRE (Perl Compatible Regular Expression) library.
- Regular expression support is provided by the PCRE library package, which is open source software written by Philip Hazel and copyright by the University of Cambridge, England. (ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/)
- Internet Component Suite developed by François Piette (http://www.overbyte.be/).
- Virtual TreeView component developed by Mike Lischke (http://www.delphi-gems.com/VirtualTreeview/VT.php).
- GHDB (Google hacking database) (http://johnny.ihackstuff.com/index.php?module=prodreviews).

Index