acunetix

A Complete guide to securing a website

To secure a website or a web application, one has to first understand the target application, how it works and the scope behind it. Ideally, the penetration tester should have some basic knowledge of programming and scripting languages, and also web security.

A website security audit usually consists of two steps. Most of the time, the first step usually is to launch an automated scan. Afterwards, depending on the results and the website's complexity, a manual penetration test follows. To properly complete both the automated and manual audits, a number of tools are available, to simplify the process and make it efficient from the business point of view. Automated tools help the user making sure the whole website is properly crawled, and that no input or parameter is left unchecked. Automated web vulnerability scanners also help in finding a high percentage of the technical vulnerabilities, and give you a very good overview of the website's structure, and security status. Thanks to automated scanners, you can have a better overview and understanding of the target website, which eases the manual penetration process.

For the manual security audit, one should also have a number of tools to ease the process, such as tools to launch fuzzing tests, tools to edit HTTP requests and review HTTP responses, proxy to analyse the traffic and so on.

In this white paper we explain in detail how to do a complete website security audit and focus on using the right approach and tools. We describe the whole process of securing a website in an easy to read step by step format; what needs to be done prior to launching an automated website vulnerability scan up till the manual penetration testing phase.

1. Manual Assessment of target website or web application

Securing a website or a web application with an automated web vulnerability scanner can be a straight forward and productive process, if all the necessary pre-scan tasks and procedures are taken care of. Depending on the size and complexity of the web application structure, launching an automated web security scan with typical 'out of the box' settings, may lead to a number of false positives, waste of time and frustration.

Even though in recent year's web vulnerability scanning technology has improved, a good web vulnerability scanner sometimes needs to be pre-configured. Web vulnerability scanners are designed to scan a wide variety of complex custom made web applications. Therefore most of the times, one would need to fine tune the scanner to his or her needs to achieve the desired correct scan results.

Before launching any kind of automated security scanning process, a manual assessment of the target website needs to be performed. It is a well known fact that an automated scanner will scan every entry point in your website which most likely you tend to forget, and test it for a wide variety of vulnerabilities.

During the manual assessment, familiarize yourself with the website topology and architecture. Keep record of the number of pages and files present in the website, and take record of the directory and file structure. If you have access to the website's root directory and source code, take your time to get to know it. If not, you can manually hover the links throughout the website. This process will help you understand the structure of the URL's. Also, take a note of all the submission and other type of online forms available on the website.

During the pre-automated scan manual assessment, apart from getting used to directory structures and number of files, get to know what web technology is used to develop the target website, e.g. .NET or PHP. There are a number of vulnerabilities which are specific for different types of technologies. Other details you should lookout for when manually assessing a website are;

1. Does the website require client certificates to be accessed?
2. Is the target website using a backend database? If yes, what type of database is it?
3. Is the database server running on the same server as the website?
4. Are all the sensitive records being encrypted?
5. Are there any URL parameters or URL rewrite rules being used for site navigation?
6. When a non existing URL is requested, does the web server return a HTTP Status Code 404, or does it return a custom error page and responds with a HTTP Status Code 200?
7. Are there any particular input forms or one time entry forms (such as CAPTCHA and Single Sign on forms) that need user input during an automated scan?
8. Are there any password protected sections in the website?

Once the manual assessment process is ready, you should know enough about the target website to help you determine if the website was properly crawled from the automated black box scanner before a scan is launched. If the website is not crawled properly, i.e. the scanner is unable to crawl some parts or parameters from the website; the whole "securing the website" point is invalidated. The manual assessment will help you go a long way towards heading off invalid scans and false positives. It will also help you get more familiar with the website itself, and that's the best way to help you configure the automated scanner to cover and check the entire website.

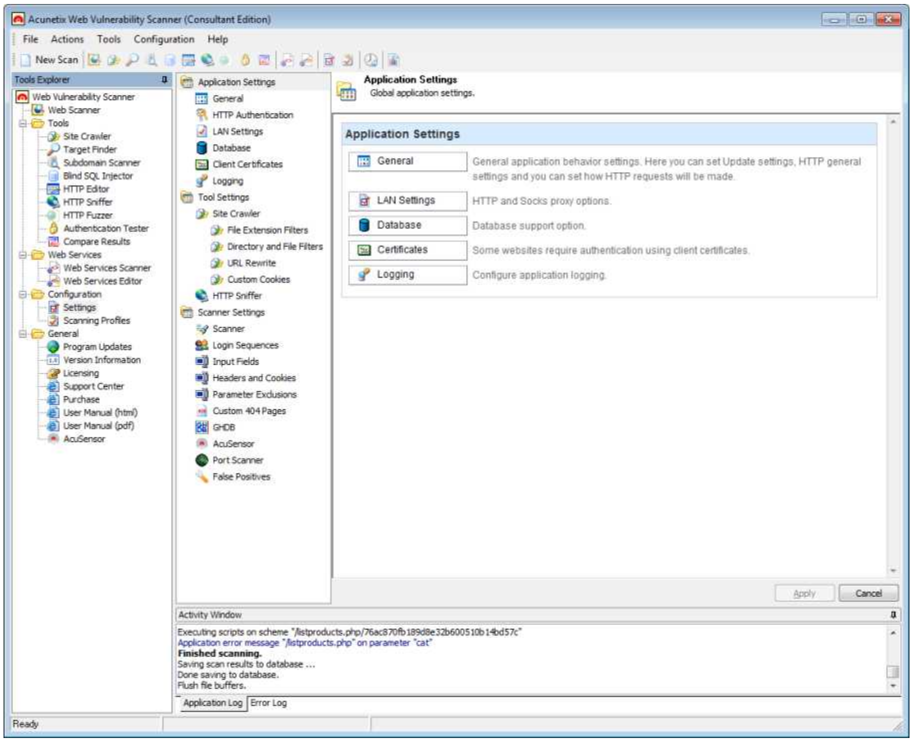
2. Get familiar with the software

Although many automated web vulnerability scanners have a comfortable GUI, if you are new to web security, you might get confused with the number of options and technical terms you'll encounter when trying to use a black box scanner. Though do not give up, it is not rocket science. Commercial black box scanners are backed up by professional support departments, so make sure you use them. You could also find a good amount of information and 'how to's' about the product you are using online. There are also a good number of open source solutions as well, but most of the time you have to dig deep and paddle on your own in rough waters to find support for such solutions. Many commercial software companies are also using social networks to make it easier for you to get to know more about their product, how it works and best practices on how you should use it.

3. Configuring the automated black box scanner

Once you're familiar with the automated black box scanner you will be using, and the target website or web application you will be scanning, it is time to get down to business and get your hands dirty. To start off with, one must first configure the scanner. The most crucial things you should configure in the scanner before launching any automated process are;

1. Custom 404 Pages – If the server returns HTTP status code 200 when a non existing URL is requested.
2. URL Rewrite rules – If the website is using search engine friendly URL's, configure these rules to help the scanner understand the website structure so it can crawl it properly.
3. Login Sequences – If parts of the website are password protected and you would like the scanner to scan them, record a login sequence to train the scanner to automatically login to the password protected section, crawl it and scan it.
4. Mark page which need manual intervention – If the website contains pages which require the user to enter a one time value when accessed, such as CAPTCHA, mark these pages as pages which need manual intervention, so during the crawling process the scanner will automatically prompt you to enter such values.
5. Submission Forms – If you would like to use specific details each time a particular form is crawled from the scanner, configure the scanner with such details. Nowadays scanners make it easy for you by populating the fields automatically (such as in Acunetix WVS).
6. Scanner Filters – Use the scanner filters to specify a file, or a file type, or directory which you would like to be excluded from the scan. You can also exclude specific parameters.



Application Settings
Global application settings.

Application Settings

General	General application behavior settings. Here you can set Update settings, HTTP general settings and you can set how HTTP requests will be made.
LAN Settings	HTTP and Socks proxy options.
Database	Database support option.
Certificates	Some websites require authentication using client certificates.
Logging	Configure application logging.

```
Executing scripts on scheme "/listproducts.php/76ac870fb189d8e32b600510b14bd57c"
Application error message "/listproducts.php" on parameter "cat"
Finished scanning.
Saving scan results to database ...
Done saving to database.
Flush file buffers.
```

Acunetix WVS Settings node

4. Protect your data

From time to time I noticed people complaining that web vulnerability scanners are too invasive, therefore they opt not to run them against their website. This is definitely a bad presumption and wrong solution, because if an automated web vulnerability scanner can break down your website, imagine what a malicious user can do. The solution is to start securing your website and make sure it can handle properly an automated scan.

To start off with, automated web vulnerability scanners tend to perform invasive scans against the target website, since they try to input data which a website has not been designed to handle. If the automated vulnerability scanner is not that invasive against a target website, then it is not really checking for all vulnerabilities and is not doing an in-depth security check. Such security checks could and will lead to a number of unwanted results; such as deletion of database records, change a blog's theme, a number of garbage posts placed on your forum, a huge number of emails in your mailbox, and even worse, a non functional website. This is expected, because like a malicious user would do, the automated black box scanner will try its best to find security holes in your website, and tries to find ways and means how to get unauthorized access.

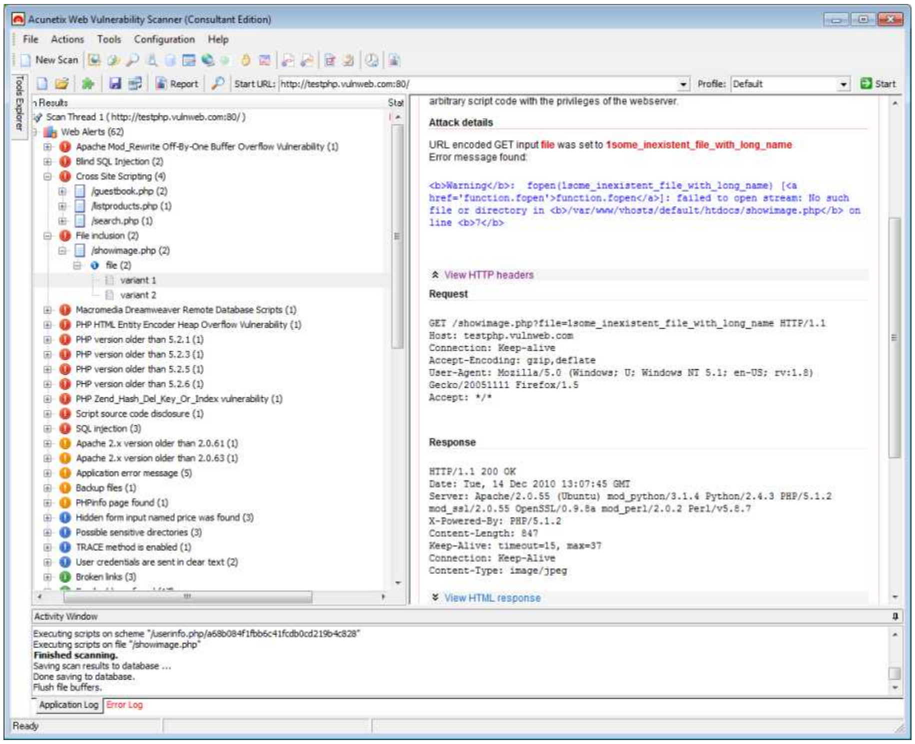
Therefore it is imperative that such scans are not launched against live servers. Ideally a replica of the live environment should be created in a test lab, so if something goes wrong, only the replica is affected. Though, if a test lab is not available, make sure you have latest backups. If something goes wrong, the live website can be restored and be functional again in the shortest time possible.

5. Launching the scan

Once the manual website analysis is ready, and the black box scanner is configured, we are ready to launch the automated scan. If time permits, you should first run a crawl of the website, so once the crawl is ready, you can confirm that all the files in the website and input parameters are crawled from the scanner. Once you confirm that all the files are crawled, you can safely proceed with the automated scan.

6. After the scan – Analysing the results

Once the automated security scan is ready, you already have a good overview of your website's security level. Look into the details of every reported vulnerability and make sure you have all the required information to fix the vulnerability. A typical black box scanner such as Acunetix Web Vulnerability Scanner will report a good amount of detail about the discovered vulnerability, such as the HTTP request and response headers, HTML response, a description of the vulnerability and a number of web links from where you can learn more about the vulnerability reported, and how to fix it.



- Scan Thread 1 (http://testphp.vulnweb.com:80/)
  - Web Alerts (62)
    - Apache Mod_Rewrite Off-By-One Buffer Overflow Vulnerability (1)
    - Blind SQL Injection (2)
    - Cross Site Scripting (4)
      - /guestbook.php (2)
      - /listproducts.php (1)
      - /search.php (1)
    - File inclusion (2)
      - /showimage.php (2)
        - file (2)
          - variant 1
          - variant 2
    - Macromedia Dreamweaver Remote Database Scripts (1)
    - PHP HTML Entity Encoder Heap Overflow Vulnerability (1)
    - PHP version older than 5.2.1 (1)
    - PHP version older than 5.2.3 (1)
    - PHP version older than 5.2.5 (1)
    - PHP version older than 5.2.6 (1)
    - PHP Zend_Hash_Del_Key_Or_Index vulnerability (1)
    - Script source code disclosure (1)
    - SQL injection (3)
    - Apache 2.x version older than 2.0.61 (1)
    - Apache 2.x version older than 2.0.63 (1)
    - Application error message (5)
    - Backup files (1)
    - PHPinfo page found (1)
    - Hidden form input named price was found (3)
    - Possible sensitive directories (3)
    - TRACE method is enabled (1)
    - User credentials are sent in clear text (2)
    - Broken links (3)

arbitrary script code with the privileges of the webserver.

Attack details

URL encoded GET input file was set to 1some_inexistent_file_with_long_name
Error message found:

```
<b>Warning</b>:  fopen(1some_inexistent_file_with_long_name) [<a
href='function.fopen'>function.fopen</a>]: failed to open stream: No such
file or directory in <b>/var/www/vhosts/default/htdocs/showimage.php</b> on
line <b>7</b>
```

View HTTP headers

Request

```
GET /showimage.php?file=1some_inexistent_file_with_long_name HTTP/1.1
Host: testphp.vulnweb.com
Connection: Keep-alive
Accept-Encoding: gzip,deflate
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8)
Gecko/20051111 Firefox/1.5
Accept: */*
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 14 Dec 2010 13:07:45 GMT
Server: Apache/2.0.55 (Ubuntu) mod_python/3.1.4 Python/2.4.3 PHP/5.1.2
mod_ssl/2.0.55 OpenSSL/0.9.8a mod_perl/2.0.2 Perl/v5.8.7
X-Powered-By: PHP/5.1.2
Content-Length: 847
Keep-Alive: timeout=15, max=37
Connection: Keep-Alive
Content-Type: image/jpeg
```

View HTML response

```
Executing scripts on scheme "/userinfo.php/a68b084f1fbb6c41fcdb0cd219b4c828"
Executing scripts on file "/showimage.php"
Finished scanning.
Saving scan results to database ...
Done saving to database.
Flush file buffers.
```

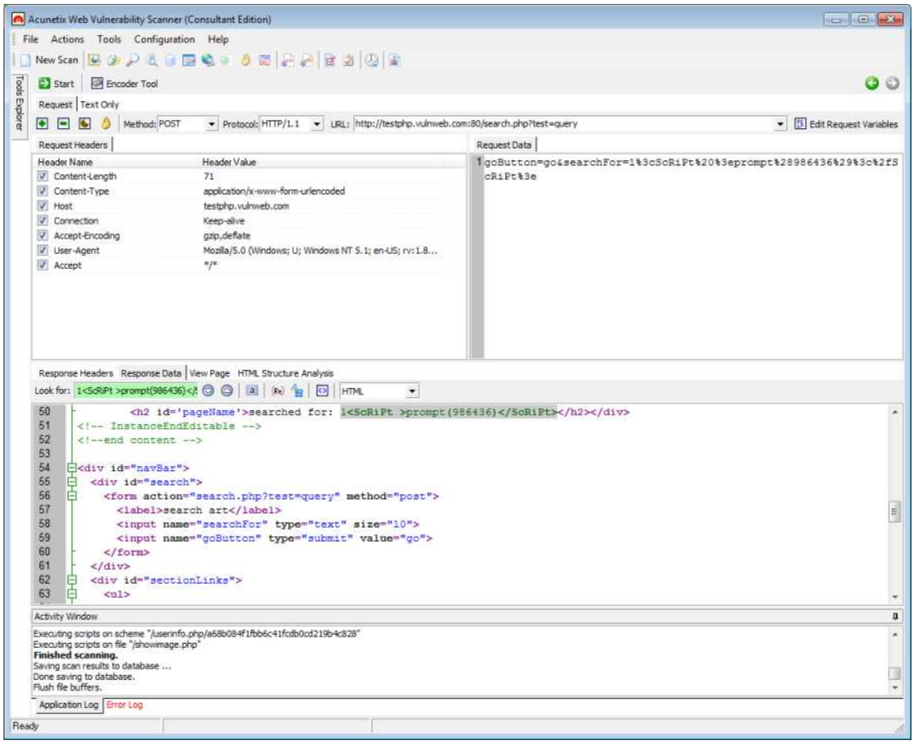
Acunetix WVS website security scan results

If AcuSensor Technology (Acunetix WVS) is enabled, much more debug information is reported; the line of code which leads to the reported vulnerability, SQL stack trace in case of SQL injection etc.

Analysing the automated scan results in detail will also help you understand more the way the web application works and how the input parameters are used, thus giving you an idea of what type of tests to launch in the manual penetration test and which parameters to target.

7. Manual penetration test

There are a number of advantages in using a commercial black box security scanner such as Acunetix Web Vulnerability Scanner. Apart from benefitting from professional support and official documentation, it also includes a number of manual advanced penetration testing tools. Having all the web penetration testing tools available in a centralized web security solution has the advantage that all the tools support importing and exporting of data from one to the other, which you will definitely need. It also helps manually analyzing the scan results by exporting the automated scan results to the manual tools and further look into the issues.



Method: POST Protocol: HTTP/1.1 URL: http://testphp.vulnweb.com:80/search.php?test=query

Header Name	Header Value
☑ Content-Length	71
☑ Content-Type	application/x-www-form-urlencoded
☑ Host	testphp.vulnweb.com
☑ Connection	Keep-alive
☑ Accept-Encoding	gzip,deflate
☑ User-Agent	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8...
☑ Accept	*/*

```
goButton=go&searchFor=1%3cScRiPt%20%3eprompt%28986436%29%3c%2fS
cRiPt%3e
```

Look for: 1<ScRiPt >prompt(986436)</ HTML

```
        <h2 id='pageName'>searched for: 1<ScRiPt >prompt(986436)</ScRiPt></h2></div>
<!-- InstanceEndEditable -->
<!--end content -->

<div id="navBar">
  <div id="search">
    <form action="search.php?test=query" method="post">
      <label>search art</label>
      <input name="searchFor" type="text" size="10">
      <input name="goButton" type="submit" value="go">
    </form>
  </div>
  <div id="sectionLinks">
    <ul>
```

```
Executing scripts on scheme "/userinfo.php/a68b084f1fbb6c41fcdb0cd219b4c828"
Executing scripts on file "/showimage.php"
Finished scanning.
Saving scan results to database ...
Done saving to database.
Flush file buffers.
```

Acunetix WVS HTTP Editor

As much as the automated scan, the manual penetration test is also a very important step in securing a website. If the advanced manual penetration testing tools are used properly, they can ease the manual penetration test process and help you be more efficient. The manual penetration testing helps you audit your website and check for logical vulnerabilities. Even though automated scans can hint you of such vulnerabilities, and help you in pin pointing them out, most of them can only be discovered and verified manually.

Below are two examples of logical vulnerabilities

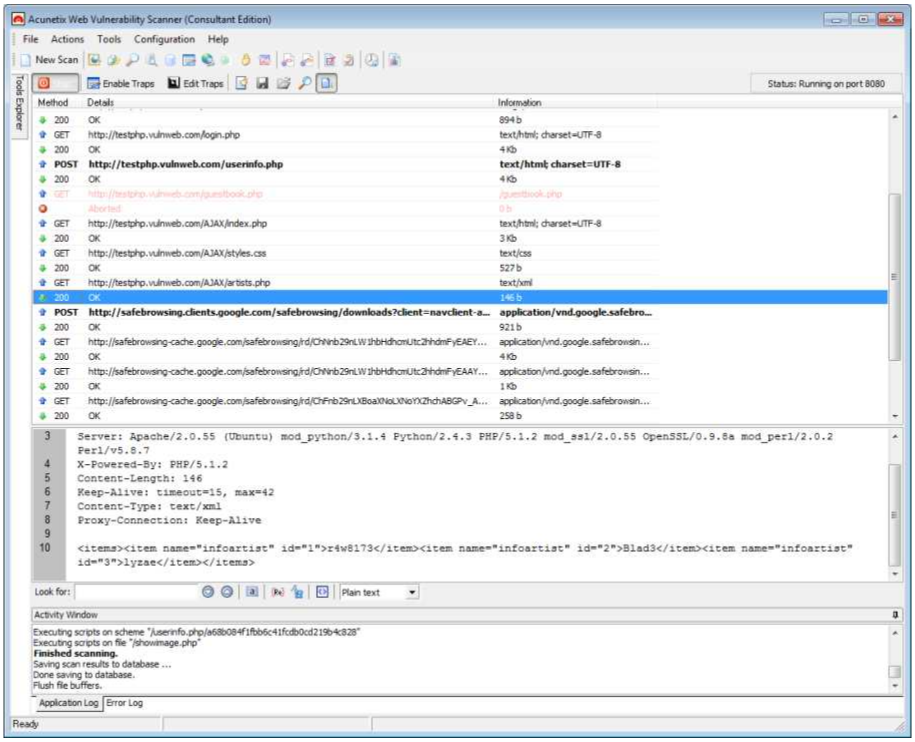
While auditing a shopping cart, you notice that if you manually set the price parameter to 0, in the checkout request, the customer can get the product for free without being asked for the payment details.

Or else imagine an online ads company promotes a new campaign; create an online account, buy $100 worth of ads and they will give you an extra $100 worth of ads for free. During development stage, the developers should make some kind of check statement like the following;

IF new account AND deposits $100 THEN give $100

If the developers forgot the AND statement, then upon opening an account and without the need to purchase $50 worth of adverts, you will still get your $100 worth of free ads.

One might think that such logical vulnerabilities are very remote, or that they are a joke, but we do encounter them when analysing production web applications. Such vulnerabilities are typically discovered by using several manual penetration testing tools together, like the HTTP Sniffer to analyze the application logic, and then the HTTP Editor to build HTTP requests, send them and analyze the server response.



Method	Details	Information
200	OK	894 b
GET	http://testphp.vulnweb.com/login.php	text/html; charset=UTF-8
200	OK	4 Kb
POST	**http://testphp.vulnweb.com/userinfo.php**	**text/html; charset=UTF-8**
200	OK	4 Kb
GET	http://testphp.vulnweb.com/guestbook.php	/guestbook.php
	Aborted	0 b
GET	http://testphp.vulnweb.com/AJAX/index.php	text/html; charset=UTF-8
200	OK	3 Kb
GET	http://testphp.vulnweb.com/AJAX/styles.css	text/css
200	OK	527 b
GET	http://testphp.vulnweb.com/AJAX/artists.php	text/xml
200	OK	146 b
POST	**http://safebrowsing.clients.google.com/safebrowsing/downloads?client=navclient-a...**	**application/vnd.google.safebro...**
200	OK	921 b
GET	http://safebrowsing-cache.google.com/safebrowsing/rd/ChNnb29nLW1hbHdhcmUtc2hhdmFyEAEY...	application/vnd.google.safebrowsin...
200	OK	4 Kb
GET	http://safebrowsing-cache.google.com/safebrowsing/rd/ChNnb29nLW1hbHdhcmUtc2hhdmFyEAAY...	application/vnd.google.safebrowsin...
200	OK	1 Kb
GET	http://safebrowsing-cache.google.com/safebrowsing/rd/ChFnb29nLXBoaXNoLXNoYXZhchABGPv_A...	application/vnd.google.safebrowsin...
200	OK	258 b

```
Server: Apache/2.0.55 (Ubuntu) mod_python/3.1.4 Python/2.4.3 PHP/5.1.2 mod_ssl/2.0.55 OpenSSL/0.9.8a mod_perl/2.0.2
Perl/v5.8.7
X-Powered-By: PHP/5.1.2
Content-Length: 146
Keep-Alive: timeout=15, max=42
Content-Type: text/xml
Proxy-Connection: Keep-Alive

<items><item name="infoartist" id="1">r4w8173</item><item name="infoartist" id="2">Blad3</item><item name="infoartist"
id="3">lyzae</item></items>
```

```
Executing scripts on scheme "/userinfo.php/a68b084f1fbb6c41fcdb0cd219b4c828"
Executing scripts on file "/showimage.php"
Finished scanning.
Saving scan results to database ...
Done saving to database.
Flush file buffers.
```

Acunetix WVS HTTP Sniffer

Conclusion

As we can see from the above, web security is very different from network security. As a concept, network security can be simplified to "allow good guys in and block the bad guys." Web security is different; it is much more than that. Though never give up. There are tools available out there which will automate most of the job for you, assist you and make the whole process easier and faster.

About Acunetix Web Vulnerability Scanner

Acunetix Web Vulnerability Scanner ensures website security by automatically checking for SQL injection, Cross-Site Scripting and other vulnerabilities. The scanner checks password strength on authentication pages and automatically audits shopping carts, forms, dynamic content and other web applications. Detailed reports resulting from the scan identify where vulnerabilities exist. The Acunetix WVS Reporting Application allows security alerts to be presented in a document which abides by the PCI Compliance specification.

About Acunetix

Acunetix is a market leader in web application security technology, founded to combat the alarming rise in web attacks. Its flagship product, Acunetix Web Vulnerability Scanner, is the result of several years of work by a team of highly experienced security developers. Acunetix customers include the US Army, US Airforce, AT&T, KPMG, Telstra, Fujitsu, and Adidas. More information can be found here http://www.acunetix.com/.