

# The Importance of Web Application Scanning

**White paper – November '05**

---

Organizations need a Web application scanning solution that can scan for security loopholes in Web-based applications to prevent would-be hackers from gaining unauthorized access to corporate applications and data. Web applications are proving to be the weakest link in overall corporate security, even though companies have left no stone unturned in installing the better-known network security and anti-virus solutions. Quick to take advantage of this vulnerability, hackers have now begun to use Web applications as a platform for gaining access to corporate data.

Web Applications Are Easy to Hack .....	2
High-Profile Web Application Hacks .....	3
Liability .....	4
Hacking Web Applications: The Modus Operandi .....	4
Hackers' Favorite Web Attack Modes .....	5
The Solution: A Web Application Scanner .....	7
Conclusion: Securing Web Applications Is Imperative .....	8
Resources for More Information .....	8
About Acunetix .....	9

## Web Applications Are Easy to Hack

The hacker's life has become tougher in recent days. Thanks to various intrusion detection and defense mechanisms developed by network security companies, it is no longer easy to breach security perimeters and gain unauthorized access to an organization's network.

Today, firewalls, security scanners and antivirus software protect almost all corporate networks. Hemmed in by such constraints, hackers have been researching alternate ways to breach the security infrastructure.

Unfortunately, hackers have been successful in finding a gaping hole in the corporate security infrastructure, one of which organizations were previously unaware – Web applications.

By design, Web applications are publicly available on the Internet, 24/7. This provides hackers with easy access and allows almost unlimited attempts to hack the application.

While the adoption of Web-based technologies for conducting business has enabled organizations to connect seamlessly with suppliers, customers and other stakeholders, it has also exposed a multitude of previously unknown security risks. According to Pete Lindstrom, Director of Security Strategies with the Hurwitz Group, Web applications are the most vulnerable elements of an organization's IT infrastructure today.

### What is a Web application?

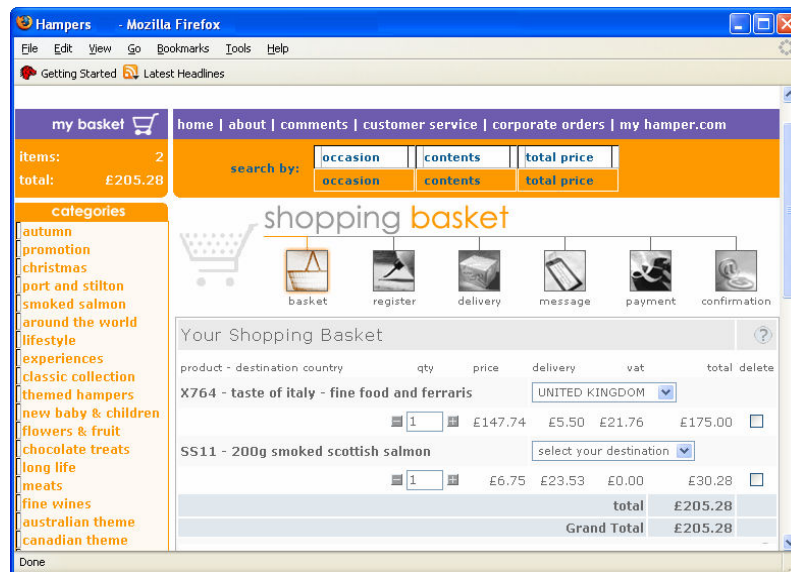
A Web application is an application that resides on a company's Web server, which any authorized user can access over a network, such as the World Wide Web or an Intranet.

A Web application is a three-layered application. Normally, the first layer would be a Web browser, the second would be a content generation technology tool such as Java servlets or ASP (Active Server Pages), and the third layer would be the company database.

The Web browser makes the initial request to the middle layer, which, in turn, accesses the database to perform the requested task, either by retrieving information from the database, or by updating it.

Since Web applications reside on a server, they can be updated and modified at any time without any distribution or installation of software on the client's machines – the main reason for the widespread adoption of Web applications in today's organizations.

Examples of Web applications include shopping carts, forms, login pages, dynamic content, discussion boards and blogs.



A Shopping cart is a typical web application example

## High-Profile Web Application Hacks

The gaping security loophole in Web applications is being exploited by hackers worldwide. According to a survey by the Gartner Group, almost three-fourths of all Internet assaults are targeted at Web applications.

The first reported instance of a Web application attack was perpetrated in 2000 by a 17 year-old Norwegian boy. While making online transactions with a large bank, he noticed that the URLs of the pages he was opening displayed his account number as one of the parameters. He then substituted his account number with the account numbers of random bank customers to gain access to the customers' accounts and personal details.

On October 31, 2001, the website of Acme Art Inc. was hacked and all the credit card numbers from its online store's database were extracted and displayed on a Usenet newsgroup. This breach was reported to the public by the media and the company lost hundreds of thousands of dollars due to orders withdrawn by wary customers. The company also lost its second phase of funding by a venture capital firm.

Similarly, the 2002 turnover report of a Swedish company was accessed prior to its scheduled publication. The perpetrator simply changed the year parameter in the URL of the previous year's report to that of the present year to gain complete access.

In another 2002 incident, applicants to Harvard Business School accessed their admission status before the results were officially announced by manipulating the online Web application. This third-party Web application was also used by other universities. Upon receiving replies to their applications from these other schools, the applicants examined the URL of the reply and found two parameters that depicted the unique IDs of that school's students. Then, they simply substituted the values in those two parameters in the reply URL with their Harvard IDs, which returned the desired information. This procedure, posted on a businessweek.com online forum, was subsequently employed by over a hundred students eager to know their admission status. When the authorities detected this leakage, these students were denied admission.

In June 2003, hackers detected that the Web applications of the fashion label Guess and pet supply retailer PetCo contained SQL injection vulnerabilities. As a result, the credit card information of almost half a million customers was stolen.

Website defacement is another major problem resulting from Web application attacks. Hackers have learned to modify the source code of many websites. During the 2004 Christmas holidays, the “Santy” worm entered Web application servers, defacing 40,000 websites in a single day. On November 29, 2004, SCO’s website logo was replaced by the text, “We own all your code, pay us all your money.” Similarly, on December 6, 2004, the homepage of Picasa, the picture sharing facility from Google, was hacked and replaced with a totally blank page.

## Liability

---

Companies face a number of legal implications from Web application attacks and lax security measures. Victoria’s Secret, one of the world’s leading lingerie manufacturers, was sued in 2005 when details about individual customers’ purchases became accessible from its database. The company was directed to pay a \$50,000 fine to New York State and settle all monetary claims by customers.

The same method was used in 2005 to access social security numbers and other details of a Tennessee payroll organization. The modus operandi was the same – change the value of the customer ID parameter in the URL.

In 2004, the Federal Trade Commission (FTC) filed judgments against a number of global organizations for privacy and security policy violations when it was discovered that there was a leakage of customer information from company databases caused by Web application intrusions.

For financial as well as legal reasons, it is imperative for companies to make their Web applications totally foolproof.

## Hacking Web Applications: The Modus Operandi

---

Hackers have a wide arsenal of attack mechanisms, from which they choose the one most suited to a particular vulnerability. They use a very systematic plan of action. These steps can be classified as:

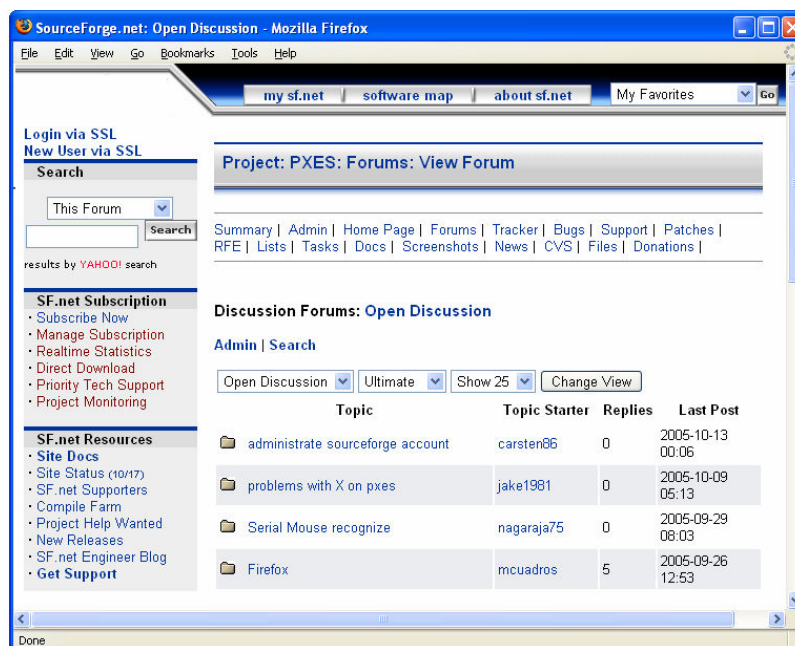
- **Study server infrastructure and server OS/type:** The hacker first analyzes the properties of the server to be hacked, the operating system running on the server, and the server type. A port scan is then initiated to detect all open HTTP and HTTPS ports to single out the port to be attacked.
- **Survey the website/application:** The hacker examines the website for any loopholes that can be exploited. Loopholes could take the form of feedback or inquiry forms that utilize GET and POST variables that hackers can use to their advantage. The hacker also inspects authentication and logon pages for any chances of accessing the server. The success of this method is evident from the 2000 incident involving the Norwegian boy. He was able to bypass required authentication by bookmarking the target page after going through authentication on his initial visit. A good hacker will go through almost every interactive element on a webpage or website in order to gain access to the server. The hacker also goes through the application script to check for any development glitches that can be exploited.
- **Check for presence of input validation:** Input validation consists of the validation that most Web applications incorporate to determine whether particular data input is safe and validated.

Unsafe data is rejected and not processed further. Laxity in input validation is a prime access pathway for hackers. If they manage to outwit the input validation check post, they can use this path to send malicious inputs to the server.

- **Mount the attack:** After examining the entire scenario, from the server to the application, and isolating all the loopholes and vulnerable target areas, the hacker now mounts the attack.

## Hackers' Favorite Web Attack Modes

- **SQL injection:** The hacker transmits SQL query commands to the database residing on the server via the Web application. This is done in two ways: SQL commands are entered in form fields on the webpage, or SQL queries are inserted into required input parameters. Thus, the hacker is able to run SQL queries and commands on the server.
- **Cross-site scripting:** The hacker inserts malicious data into a dynamic webpage. Websites that include only static webpages have control over user interaction because a static webpage is a "read-only" page that does not permit user interaction. Therefore, a would-be hacker can only view the page without being able to cause any damage. However, a dynamic webpage is open to user interaction, so a hacker can insert hazardous content without the website or Web application being able to differentiate this content from innocuous content. The key to the CSS vulnerability is that a hacker can cause the actual Web server to send a webpage with malicious content to the unsuspecting user. The hacker can then transfer the user's input to another server.

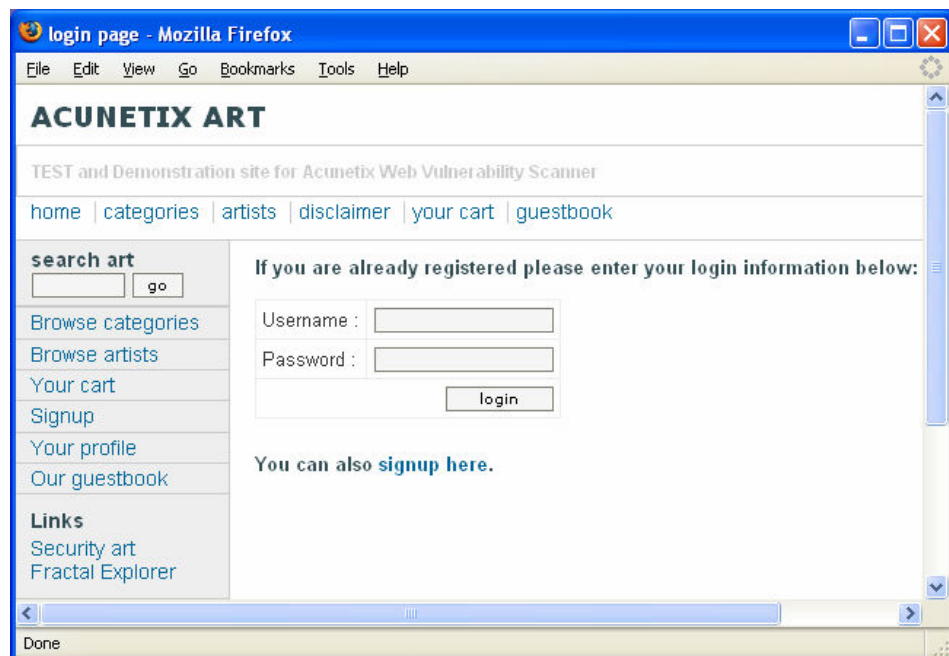


Forums are often vulnerable to Cross site scripting attacks

- **Directory traversal attacks:** This attack is also called the ../ (dot dot slash) attack. With this attack, the Web application is manipulated to allow access to files or other resources on the server that are not normally accessible. The attack works by changing the parameter that an application would use to access a certain file. For instance, suppose the value of the parameter includes the path of a particular file. Placing ../ at the beginning of the parameter value forces the application to access the file in the parent directory. By placing a series of ../

and then giving a different file name at the end, a particular file in the root directory can be retrieved.

- **Parameter manipulation:** This involves manipulating data transmitted between the browser and Web application. Parameter manipulation can be carried out in the following ways:
  - **Cookie manipulation:** Cookies maintain a certain state in HTTP by storing user preferences and information related to session maintenance. All cookies can be changed at the client end and then sent to the server with URL requests. Thus, a hacker can easily manipulate the data residing within a cookie.
  - **HTTP header manipulation:** HTTP headers consist of control information that is sent from the Web client to the Web server during HTTP requests, and sent from Web servers to Web clients during HTTP responses. Since the HTTP request headers originate from the client, a hacker can easily modify them.
  - **HTML form field manipulation:** Form fields contain values of all the check boxes checked, radio buttons selected, text fields filled or any other action by a user on a particular webpage. This data is then sent to the server. Moreover, there can be hidden fields not visible to the user on the page that are sent to the server. A potential hacker can manipulate the form fields to send any value. One example of this manipulation is to simply right-click the mouse on the webpage to view the source code, alter it, save the changes and then reload the page in the browser.
  - **URL manipulation:** The HTML forms mentioned above are submitted in a process that requires a certain result to be displayed to the user before the result is displayed on a fresh webpage. The URL of this page will contain all the form field names and their respective values, which can be easily manipulated.



An example of an HTML form-based login

- **Authentication attacks:** The hacker searches for valid authentication to access and enter the server from a Web application. For this kind of attack, a database of usernames and passwords is maintained in order to maximize authentication and thereby obtain access to restricted domains.

- **Known exploits:** The hacker community is very close-knit; newly discovered Web application intrusions are posted on a number of community forums and websites known only to members of that group. These postings are updated on a daily basis and are used to facilitate further hacking.
- **Directory enumeration:** Analyzing the website's entire directory structure, the hacker seeks out hidden directories. These hidden directories could contain administrative data that the hacker may find valuable when launching attacks.

## The Solution: A Web Application Scanner

Clearly, Web applications are the biggest Achilles heel in an organization's security strategy. They are much more difficult to protect than traditional applications that reside behind a firewall. Web application security needs to be stringently checked using an automated Web application security scanner.

A Web application scanner is an automated security program that searches for software vulnerabilities within Web applications. A Web application scanner first crawls the entire website, analyzing in-depth each file it finds, and displaying the entire website structure. After this discovery stage, it performs an automatic audit for common security vulnerabilities by launching a series of Web attacks. Web application scanners check for vulnerabilities on the Web server, proxy server, Web application server and even on other Web services.

Important features of Web applications scanners are:

- Ability to analyze different Web technologies, such as PHP, ASP.NET, ASP, etc.
- Ability to scale: Should be fast enough to process large websites
- Ability to produce readable and actionable results without extensive Web security know-how.

## List of Web Application Scanning Solutions<sup>1</sup>

Vendor name	Product name	Features
Acunetix	Web Vulnerability Scanner	Provides protection from the following attacks: <ul style="list-style-type: none"> <li>• CRLF injection attacks</li> <li>• Code execution attacks</li> <li>• Directory traversal attacks</li> <li>• File inclusion attacks</li> <li>• Input validation attacks</li> <li>• Authentication attacks</li> </ul> Creates professional security audit reports
Application Security Inc.	AppDetective	A network-based vulnerability assessment tool that rates the security strength of applications within your network.
Imperva	SecureSphere Dynamic Profiling Firewall	Provides total protection for Web application and Web service attacks, database breaches and worm infections. Incorporates a Web firewall, a database firewall, database auditing, Web services firewall, Intrusion Prevention System (IPS) and a network firewall. Includes one gigabit performance sub-millisecond latency.
Kavado	Scando Web application scanner	Detects and eliminates Web application vulnerabilities before exploitation by hackers and thieves. Compatible with every stage of the software life cycle — from development and installation right up to the auditing stage. Works together with the Kavado InterDo firewall to ensure continuous security monitoring of the Web application. Compatible with all Web technologies like Flash, ASP, JavaScript, XML and Web Services.

		<p>Follows a structured three-stage scanning process:</p> <ol style="list-style-type: none"> <li>1. Studies the structure and content of the Web application.</li> <li>2. Executes dummy hacking instances to detect vulnerabilities.</li> <li>3. Displays scan results in systematic reports along with suggestions for remedial solutions.</li> </ol>
Watchfire	AppScan	Security software that automates the complex, manual task of auditing Web applications.
	AppScan DE	Integrated seamlessly into VS.NET, AppScan DE is a powerful automated unit-testing tool that enables rapid development of secure Web applications.
SPI Dynamics	WebInspect	Efficiently detects vulnerabilities in Web applications. Ensures that there's no chance of an attack at any point in the Web application development and implementation of the lifecycle.
<b>Source: Company websites</b>		

## Conclusion: Securing Web Applications Is Imperative

Attacks on Web applications are increasing at a rapid pace. As per a report from the Computer Emergency Response Team (CERT), the number of successful Web application attacks is on the rise, from around 60% in 2002 to 80% in 2003. If Web application infringements continue to grow at this rate, customers' confidence in online commerce will further diminish. As observed by Gartner, rampant attacks on Web applications make customers wary of making online purchases for fear of credit card tampering and leakage of credit information.

When companies fail to recognize application vulnerabilities, hackers have free rein attacking security loopholes. Hackers are increasingly focusing on Web applications for monetary gains and their attack modes are becoming more advanced and difficult to prevent.

Recent examples demonstrate the unfortunate after effects that companies have faced after such Web application breaches. Companies have borne the brunt of lawsuits, incurred financial losses, lost their credibility in the eyes of the public and, last but not least, have seen their company secrets siphoned off right under their noses.

The only way to combat the Web application security threat is to proactively scan websites and Web applications for vulnerabilities and then fix them. Implementing a Web application scanning solution must be a crucial part of any organization's overall strategy.

## Resources for More Information

For more information on Web application security and related documentation, visit: OWASP (The Open Web Application Security Project) [www.owasp.org](http://www.owasp.org); Acunetix Web Attacks Info page (<http://www.acunetix.com/websitesecurity/web-site-security.htm>)

## About Acunetix

---

Acunetix™ is a new company specializing in Web security technology. Its product, the Acunetix Web Vulnerability Scanner, is the result of several years of development and utilizes unique technology to allow companies to check the security of their websites. The Acunetix product development team consists of highly specialized developers, each of whom has extensive experience in the computer security field and is familiar with the latest hacking techniques. The management team is backed by many years of marketing and selling network software. Acunetix Ltd. is a privately held firm. For more information, please visit [www.acunetix.com](http://www.acunetix.com).

<sup>1</sup>Disclaimer: The information for the table comparing the features of various commercially available Web application scanning solutions have been sourced “as-is” from the literature provided in the company websites as of September 2005. The table does not provide a feature-to-feature technical comparison of the products and neither does it suppose a live testing of these products in any laboratory environment for the stated features nor an acknowledgement of the authenticity or the validity of the facts stated in the company literatures. The table is meant solely for the purposes of drawing a study comparison and Acunetix cannot be held liable for any financial decisions made by inferring the contents of this table.

© 2005 Acunetix Ltd. All rights reserved. The information contained in this document represents the current view of Acunetix on the issues discussed as of the date of publication. Because Acunetix must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Acunetix, and Acunetix cannot guarantee the accuracy of any information presented after the date of publication. This White Paper is for informational purposes only. Acunetix MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT. Acunetix, Acunetix Web Vulnerability Scanner and their product logos are either registered trademarks or trademarks of Acunetix Ltd. in the United States and/or other countries. All product or company names mentioned herein may be the trademarks of their respective owners.