



Web Vulnerability Assessment Report

Target Scanned: www.vulnerable-website.com

Report Generated: Tue Mar 22 12:38:07 2016

Identified Vulnerabilities: 27

Threat Level: High



Screenshot of www.vulnerable-website.com HomePage

The screenshot shows a travel website homepage. At the top, there is a navigation menu with links for Home, LogIn, Bookings, Services, and Contacts. To the right of the navigation is a 'Bookings & Reservations' section with checkboxes for Flight, Hotel, and Cruise, a search input field, and a 'Search' button. Below this is a 'Search For Vacation Package' section with a radio button. The main content area is divided into two columns. The left column features a large image of two hikers in a field, with a 'Hottest Locations' section below it. This section lists 'Orlando - 4 nights \$341' and 'Hawaii - 7 nights \$761', each with a small image and a brief description. The right column features 'Our Packages' section, which includes 'Golden Package' (15-night hotels, air-conditioned motorcoach, admission to Montezuma's Castle, Monument Valley, Flaming Gorge, 17-Mile Drive and all National Parks, Fully Escorted Touring to Grand Canyon, Bryce Canyon, Zion, Canyonlands, Grand Teton, Arches, Yellowstone, Yosemite, Salt Lake City, Las Vegas, San Francisco, Phoenix/Scottsdale, Los Angeles) and 'Silver Package' (5-night hotel, transfers to airports and transportation are included, Visit New York City, Philadelphia, Metropolitan Museum Washington D.C, Hershey Chocolate World, Corning Glass Center, Niagara Falls, Boston and more). There are 'read more' links for both packages. At the bottom of the right column, there are links for '* Varadero - 3 nights \$521' and '* Cienfuegos - 5 nights \$452'. At the bottom of the left column, there is a link to 'Web Hosting Zoom'.



© GamaSec Web Vulnerability Assessment Report

Target Scanned :	www.vulnerable-website.com:80
Report Generated :	Tue Mar 22 12:38:07 2016

GamaScan has identified over 20 vulnerabilities, you should manually verify them to eliminate False-Positive results.

Table Of Contents

Introduction	2
Executive Summary	3
Technical Details	9

Introduction

"Explore Your Web Vulnerabilities"

A remote web vulnerability assessment was conducted by GamaScan against the web-interfaced system. The purpose of this assessment was to remotely identify and quantify vulnerabilities or potential threats in the web-interfaced system before they are exploited by attackers. The tests performed by GamaScan are customized dynamically to the scanned target and includes thousands of known vulnerabilities, dynamic tests and web application vulnerabilities.

By assessing the web-interfaced system remotely GamaScan accurately mimicking the attacker's view.

GamaScan Automated Web-Application Vulnerability Assessment Service Key features:

- 🔄 Zero time setup.
- 🔄 No software installation.
- 🔄 No special hardware.
- 🔄 No special training.
- 🔄 Pre defined or customizable scans.
- 🔄 Detailed, Crystal clear, reports.
- 🔄 Constant updates.



Executive Summary

This section provides an overview of the vulnerability assessment results and shows the distribution of vulnerabilities by severity level and by category.

Security Threat Level

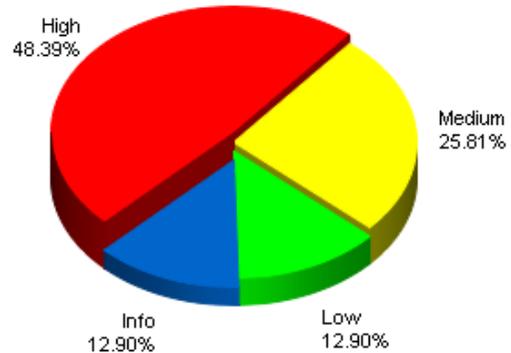
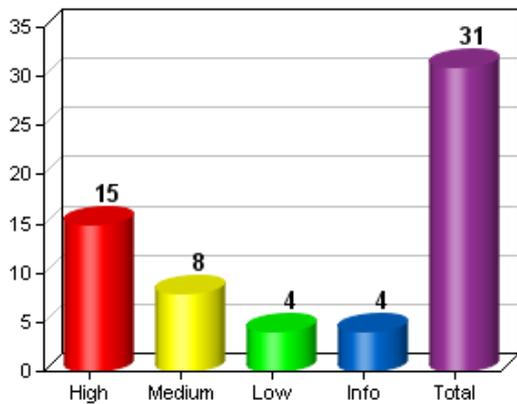
This graph presents the security threat level based on the vulnerabilities identified by GamaScan. The "Threat Level" is classified as being of Informational, Low, Medium or High severity.



Vulnerabilities by Severity

This section shows the number of vulnerabilities identified by GamaScan, grouped according to their severity levels. vulnerabilities are classified as being of High, Medium, Low or Informational severity.

High	15
Medium	8
Low	4
Info	4





Tests Overview

This section shows the performed security tests and their results.

Test Category	Test Result
Web Servers	Pass
Web Server Technologies	Pass
HTTP Methods	Pass
Directory Indexing	Pass
Directory Access	Pass
Directory Permissions	Pass
Sensitive/Common Files	Pass
Backup Files	Pass
SQL Injection	Fail
XPath Injection	Pass
CRLF Injection	Pass
LDAP Injection	Pass
Cookie Manipulation	Fail
Source Code Disclosure	Pass
Directory Traversal	Pass
Installation Path Disclosure	Pass
Platform Exception	Pass
Script Language Error	Pass
Cross-Site Scripting	Pass
Cross-Frame Scripting	Fail
URL Redirection	Fail
Command Execution	Pass
PHP Code Injection	Fail
Remote File Inclusion	Pass
Internal IP Disclosure	Pass



Vulnerability Descriptions	
This section shows the common application vulnerabilities with their risk.	
Vulnerability	Description
SQL injection	SQL injection may allow a remote attacker to inject or manipulate SQL queries in the back-end database, allowing for the manipulation or disclosure of arbitrary data.
CRLF Injection	CRLF Injection may allow an attacker to inject CRLF sequences within a Location element of an HTTP response header, helping to facilitate XSS and other attacks.
LDAP Injection	LDAP Injection may allow an attacker to inject arbitrary LDAP queries, this can lead to disclosure of sensitive information.
XPath Injection	XPath Injection may allow an attacker to inject or manipulate XQueries, this can lead to disclosure of sensitive information.
Cross-Site Scripting (XSS)	Cross-Site Scripting may allow an attacker to create a specially crafted request that would execute arbitrary script code in a user's browser session within the trust relationship between their browser and the server.
Cross-Frame Scripting (XFS)	Cross-Frame Scriptings may allow an attacker to use one web page to load content from another, concealing the origin of one web site. This method is useful in phishing attacks, to load legitimate content from one site, while loading a malicious form embedded in that content.
URL Redirection	URL Redirection may allow an attacker to create a specially crafted URL, that if clicked, would redirect a victim from the intended legitimate web site to an arbitrary web site of the attacker's choosing. Such attacks are useful as the crafted URL initially appear to be a web page of a trusted site. This could be leveraged to direct an unsuspecting user to a web page containing attacks that target client side software such as a web browser or document rendering programs.
Cookie Manipulation	Cookie Manipulation may allow an attacker to alter or modify the cookies values on the clients web browser. This flaw may lead to an attacker gaining unauthorised access, leading to a loss of integrity.
Installation Path Disclosure	Installation Path Disclosure vulnerability disclose the web root's installation path resulting in a loss of confidentiality, it is often useful in carrying out additional, more focused attacks.
Internal IP Disclosure	Internal IP Disclosure may allow an attacker to gain better knowledge of the internal networking scheme, making subsequent attackers more easy.
Directory Traversal	Directory Traversal may allow an attacker to access arbitrary files outside of the web path.
Exception Handling	Exception error message may provide attacker useful attack information, it is often useful in carrying out additional, more focused attacks.
Source Code Disclosure	Source Code Disclosure vulnerability disclose the source code of the application to a remote attacker.
Command Execution	Command Execution may allow an attacker to inject arbitrary OS commands that will be executed on the web server, this can lead to complete host takeover.
PHP Code Injection	PHP Code Injection may allow an attacker to inject PHP code that will be executed on the web server.
Remote File Inclusion	Remote File Inclusion may allow an attacker to include a file from a third-party remote host that contains commands or code that will be executed by the vulnerable script with the same privileges as the web server.
HTTP Methods	Misconfigured web server supporting a potentially dangerous HTTP methods like DELETE, PUT, TRACE, TRACK may lead to disclosure of sensitive information and to additional, more focused attacks.
Security / Server Misconfiguration	Security Misconfiguration may allow an attacker to gain access to default files, sample files, configuration files, sensitive files, common files, unprotected files, unprotected directories and unsecure admin interface. this can lead to disclosure of sensitive information and to complete host takeover.

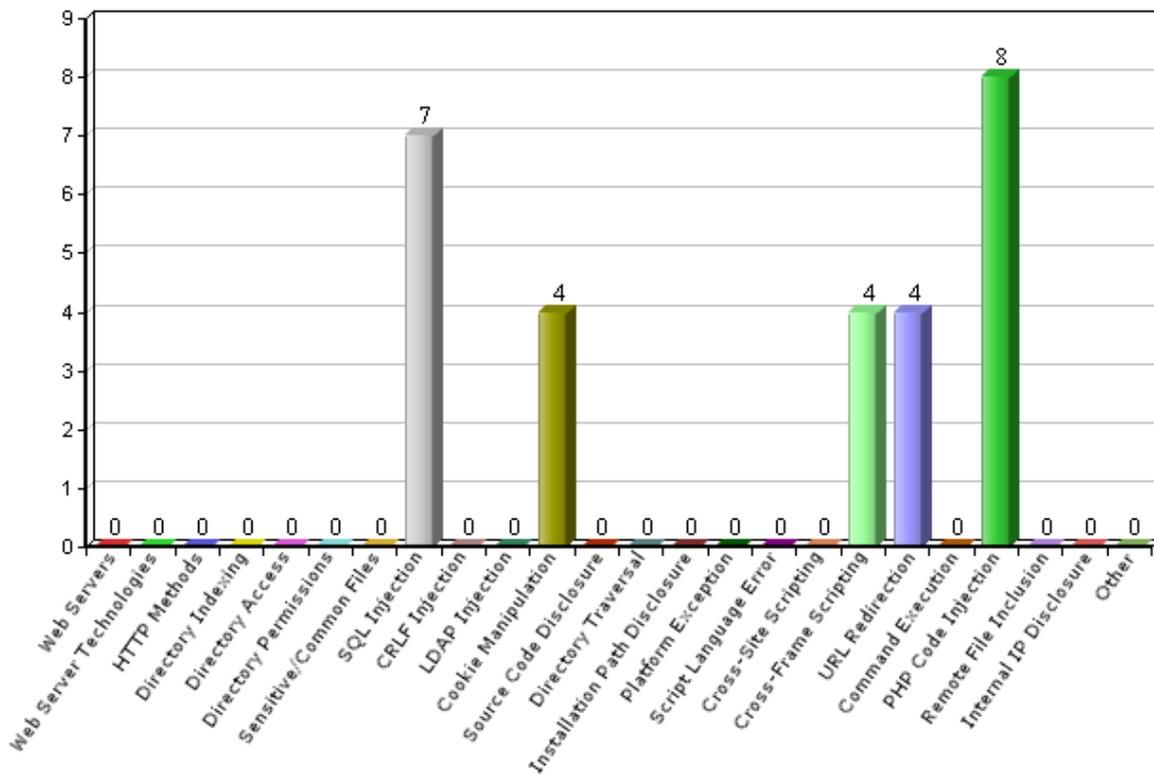


Vulnerabilities by Category

This section shows the number of vulnerabilities identified by GamaScan, grouped according to their categories.

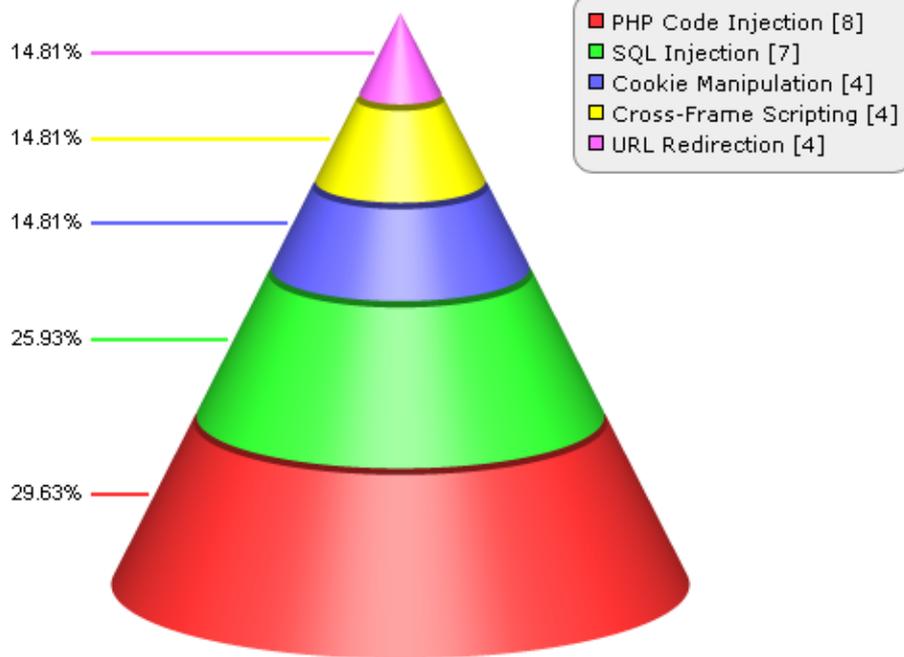
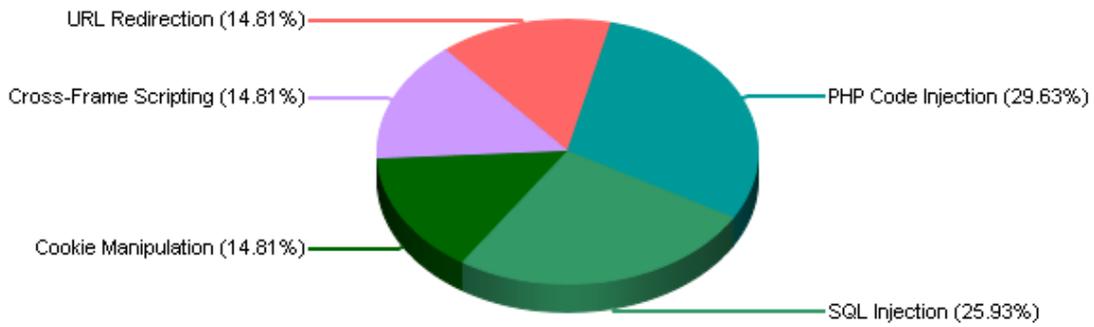
Number of Vulnerabilities by Category

Category	Number Of Vulnerabilities
SQL Injection	7 25.92%
Cookie Manipulation	4 14.81%
Cross-Frame Scripting	4 14.81%
URL Redirection	4 14.81%
PHP Code Injection	8 29.62%





Percent of Vulnerabilities by Category

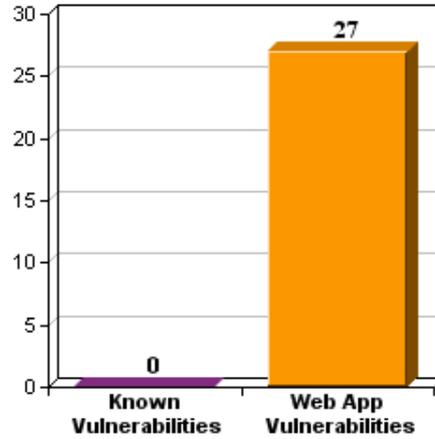
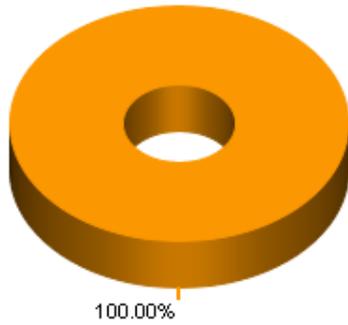




Vulnerabilities by Attacks Category

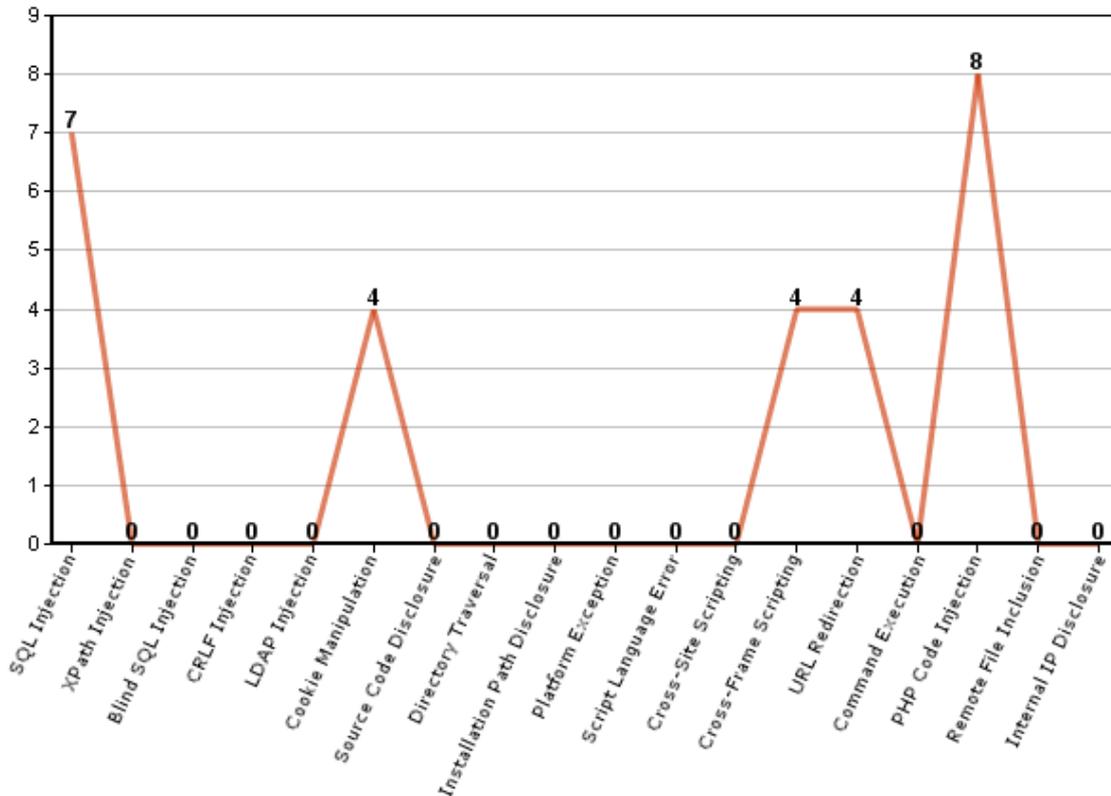
This chart shows the distribution of vulnerabilities by "Known Vulnerabilities" and by "Web Application Vulnerabilities". "Known Vulnerabilities" are identified using known attack signatures. "Web Application Vulnerabilities" are identified using web attacks customized to the scanned web application.

Web App Vulnerabilities



Vulnerabilities by "Web Application" Category

This chart shows the distribution of vulnerabilities identified using web attacks customized to the scanned web application.





Technical Details

This section provides details on the open ports, web server, vulnerabilities and threats identified on the web-interfaced system.

Port Scan

This table shows the open ports on the system. not each open port is a security threat, but open ports on the system are invitations to attackers. In general, the number of open ports should be kept to a minimum and only the mission-critical ports should be open.

Port Number	Service	Description
80 (tcp)	http	World Wide Web HTTP
443 (tcp)	https	secure http (SSL)

Web Server

This table provides general details on the web server identified by GamaScan.

Target Banner	Unknown
HTTP Methods	
Cookie	pron_ip=7; path=/;



Vulnerability Findings

This section provides technical details on the vulnerabilities identified by GamaScan, vulnerabilities are classified as being of High, Medium, Low or Informational severity and grouped according to their severity levels.

High Severity Findings

Possibility of SQL Injection Script: Search.aspx Variable: txtSearch Value: '

Description	<p>GamaScan has identified a flaw that may allow an attacker to carry out an SQL injection attack. This flaw exists because the application does not properly sanitize user-supplied input to the variable. This may allow an attacker to inject or manipulate SQL queries in the backend database.</p> <div style="border: 1px solid gray; padding: 2px; margin: 5px 0;">Form Method: POST</div> <div style="border: 1px solid gray; padding: 2px; margin: 5px 0;">The Form Inputs</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #cccccc;"> <th style="text-align: left;">Type</th> <th style="text-align: left;">Name</th> </tr> </thead> <tbody> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td>checkbox</td><td>flight</td></tr> <tr><td>checkbox</td><td>hotel</td></tr> <tr><td>checkbox</td><td>cruise</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtSearch</td></tr> <tr><td>submit</td><td>Button1</td></tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
Type	Name																								
hidden	__LASTFOCUS																								
hidden	__VIEWSTATE																								
hidden	__VIEWSTATEGENERATOR																								
hidden	__EVENTTARGET																								
hidden	__EVENTARGUMENT																								
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								
Scan Request	www.vulnerable-website.com:80/Search.aspx?key='																								
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> Wikipedia sqlsecurity 																								



Possibility of PHP Code Injection
Script: Search.aspx **Variable:** txtSearch
Value: print(md5(1234567));exit;

Description	<p>GamaScan has identified a flaw that may allow an attacker to compromise the system. The flaw occurs due to use of unsanitized user-supplied data being used in a PHP eval() function call. It is possible that the flaw may allow arbitrary PHP code execution resulting in a loss of integrity.</p> <div style="background-color: #f0f0f0; padding: 2px; text-align: center; border: 1px solid #ccc;">Form Method: POST</div> <div style="background-color: #f0f0f0; padding: 2px; text-align: center; border: 1px solid #ccc;">The Form Inputs</div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f0f0f0;"> <th style="text-align: left;">Type</th> <th style="text-align: left;">Name</th> </tr> </thead> <tbody> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td>checkbox</td><td>flight</td></tr> <tr><td>checkbox</td><td>hotel</td></tr> <tr><td>checkbox</td><td>cruise</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtSearch</td></tr> <tr><td>submit</td><td>Button1</td></tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
Type	Name																								
hidden	__LASTFOCUS																								
hidden	__VIEWSTATE																								
hidden	__VIEWSTATEGENERATOR																								
hidden	__EVENTTARGET																								
hidden	__EVENTARGUMENT																								
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								
Scan Request	www.vulnerable-website.com:80/Search.aspx?key=print(md5(1234567));exit;																								
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> PHP wikipedia 																								

Possibility of SQL Injection
Script: Bookings.aspx **Variable:** CityName
Value: '

Description	<p>GamaScan has identified a flaw that may allow an attacker to carry out an SQL injection attack. This flaw exists because the application does not properly sanitize user-supplied input to the variable. This may allow an attacker to inject or manipulate SQL queries in the backend database.</p>
Scan Request	www.vulnerable-website.com:80/Bookings.aspx?CityName='&CuntryID=2
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> Wikipedia sqlsecurity



Possibility of PHP Code Injection
Script: Bookings.aspx **Variable:** CityName
Value: print(md5(1234567));exit;

Description	GamaScan has identified a flaw that may allow an attacker to compromise the system. The flaw occurs due to use of unsanitized user-supplied data being used in a PHP eval() function call. It is possible that the flaw may allow arbitrary PHP code execution resulting in a loss of integrity.
Scan Request	www.vulnerable-website.com:80/Bookings.aspx?CityName=print(md5(1234567));exit;&CuntryID=2
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: PHP wikipedia

Possibility of SQL Injection
Script: Bookings.aspx **Variable:** CuntryID
Value: '

Description	GamaScan has identified a flaw that may allow an attacker to carry out an SQL injection attack. This flaw exists because the application does not properly sanitize user-supplied input to the variable. This may allow an attacker to inject or manipulate SQL queries in the backend database.
Scan Request	www.vulnerable-website.com:80/Bookings.aspx?CityName=Orlando%20-%204%20nights%20\$341&CuntryID='
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: Wikipedia sqlsecurity



Possibility of PHP Code Injection Script: Bookings.aspx Variable: CuntryID Value: print(md5(1234567));exit;	
Description	GamaScan has identified a flaw that may allow an attacker to compromise the system. The flaw occurs due to use of unsanitized user-supplied data being used in a PHP eval() function call. It is possible that the flaw may allow arbitrary PHP code execution resulting in a loss of integrity.
Scan Request	www.vulnerable-website.com:80/Bookings.aspx?CityName=Orlando%20-%204%20nights%20\$341&CuntryID=print(md5(1234567));exit;
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: PHP wikipedia



Possibility of PHP Code Injection
Script: LogIn.aspx **Variable:** txtUserName
Value: print(md5(1234567));exit;

Description	<p>GamaScan has identified a flaw that may allow an attacker to compromise the system. The flaw occurs due to use of unsanitized user-supplied data being used in a PHP eval() function call. It is possible that the flaw may allow arbitrary PHP code execution resulting in a loss of integrity.</p> <table border="1" style="width: 100%; border-collapse: collapse; background-color: #f0f0f0;"> <tr><td colspan="2" style="text-align: center;">Form Method: POST</td></tr> <tr><td colspan="2" style="text-align: center;">The Form Inputs</td></tr> <tr> <th style="width: 30%;">Type</th> <th>Name</th> </tr> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtUserName</td></tr> <tr><td>password</td><td>txtPassword</td></tr> <tr><td>submit</td><td>btnLogIn</td></tr> </table>	Form Method: POST		The Form Inputs		Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	text	txtUserName	password	txtPassword	submit	btnLogIn
Form Method: POST																									
The Form Inputs																									
Type	Name																								
hidden	__LASTFOCUS																								
hidden	__VIEWSTATE																								
hidden	__VIEWSTATEGENERATOR																								
hidden	__EVENTTARGET																								
hidden	__EVENTARGUMENT																								
hidden	__EVENTVALIDATION																								
text	txtUserName																								
password	txtPassword																								
submit	btnLogIn																								
Scan Request	www.vulnerable-website.com:80/LogIn.aspx?UserName=print(md5(1234567));exit;																								
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> PHP wikipedia 																								

Possibility of SQL Injection
Script: Orders.aspx **Variable:** CityName
Value: '

Description	<p>GamaScan has identified a flaw that may allow an attacker to carry out an SQL injection attack. This flaw exists because the application does not properly sanitize user-supplied input to the variable. This may allow an attacker to inject or manipulate SQL queries in the backend database.</p>
Scan Request	www.vulnerable-website.com:80/Orders.aspx?CityName='&CuntryID=1
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> Wikipedia sqlsecurity



Possibility of PHP Code Injection
Script: Orders.aspx **Variable:** CityName
Value: print(md5(1234567));exit;

Description	GamaScan has identified a flaw that may allow an attacker to compromise the system. The flaw occurs due to use of unsanitized user-supplied data being used in a PHP eval() function call. It is possible that the flaw may allow arbitrary PHP code execution resulting in a loss of integrity.
Scan Request	www.vulnerable-website.com:80/Orders.aspx?CityName=print(md5(1234567));exit;&CuntryID=1
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: PHP wikipedia

Possibility of SQL Injection
Script: Orders.aspx **Variable:** CuntryID
Value: %27

Description	GamaScan has identified a flaw that may allow an attacker to carry out an SQL injection attack. This flaw exists because the application does not properly sanitize user-supplied input to the variable. This may allow an attacker to inject or manipulate SQL queries in the backend database.
Scan Request	www.vulnerable-website.com:80/Orders.aspx?CityName=Hawai%20-%207%20nights%20\$761&CuntryID=%27
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: Wikipedia sqlsecurity



Possibility of PHP Code Injection
Script: Orders.aspx **Variable:** CuntryID
Value: print(md5(1234567));exit;

Description	GamaScan has identified a flaw that may allow an attacker to compromise the system. The flaw occurs due to use of unsanitized user-supplied data being used in a PHP eval() function call. It is possible that the flaw may allow arbitrary PHP code execution resulting in a loss of integrity.
Scan Request	www.vulnerable-website.com:80/Orders.aspx?CityName=Hawai%20-%207%20nights%20\$761&CuntryID=print(md5(1234567));exit;
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: PHP wikipedia

Possibility of SQL Injection
Script: Search.aspx **Variable:** key
Value: '

Description	GamaScan has identified a flaw that may allow an attacker to carry out an SQL injection attack. This flaw exists because the application does not properly sanitize user-supplied input to the variable. This may allow an attacker to inject or manipulate SQL queries in the backend database.
Scan Request	www.vulnerable-website.com:80/Search.aspx?key='
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: Wikipedia sqlsecurity



Possibility of PHP Code Injection Script: Search.aspx Variable: key Value: print(md5(1234567));exit;	
Description	GamaScan has identified a flaw that may allow an attacker to compromise the system. The flaw occurs due to use of unsanitized user-supplied data being used in a PHP eval() function call. It is possible that the flaw may allow arbitrary PHP code execution resulting in a loss of integrity.
Scan Request	www.vulnerable-website.com:80/Search.aspx?key=print(md5(1234567));exit;
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: PHP wikipedia



Possibility of SQL Injection Script: View.aspx Variable: txtSearch Value: '																									
Description	GamaScan has identified a flaw that may allow an attacker to carry out an SQL injection attack. This flaw exists because the application does not properly sanitize user-supplied input to the variable. This may allow an attacker to inject or manipulate SQL queries in the backend database.																								
	Form Method: POST																								
	The Form Inputs																								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Type</th> <th style="text-align: left;">Name</th> </tr> </thead> <tbody> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td>checkbox</td><td>flight</td></tr> <tr><td>checkbox</td><td>hotel</td></tr> <tr><td>checkbox</td><td>cruise</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtSearch</td></tr> <tr><td>submit</td><td>Button1</td></tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
	Type	Name																							
hidden	__LASTFOCUS																								
hidden	__VIEWSTATE																								
hidden	__VIEWSTATEGENERATOR																								
hidden	__EVENTTARGET																								
hidden	__EVENTARGUMENT																								
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								
Scan Request	www.vulnerable-website.com:80/View.aspx?nameid='																								
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: Wikipedia sqlsecurity																								



Possibility of PHP Code Injection Script: View.aspx Variable: txtSearch Value: print(md5(1234567));exit;																													
Description	<p>GamaScan has identified a flaw that may allow an attacker to compromise the system. The flaw occurs due to use of unsanitized user-supplied data being used in a PHP eval() function call. It is possible that the flaw may allow arbitrary PHP code execution resulting in a loss of integrity.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="background-color: #cccccc;">Form Method: POST</th> </tr> <tr> <th colspan="2" style="background-color: #cccccc;">The Form Inputs</th> </tr> <tr> <th style="width: 30%;">Type</th> <th>Name</th> </tr> </thead> <tbody> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td>checkbox</td><td>flight</td></tr> <tr><td>checkbox</td><td>hotel</td></tr> <tr><td>checkbox</td><td>cruise</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtSearch</td></tr> <tr><td>submit</td><td>Button1</td></tr> </tbody> </table>	Form Method: POST		The Form Inputs		Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
Form Method: POST																													
The Form Inputs																													
Type	Name																												
hidden	__LASTFOCUS																												
hidden	__VIEWSTATE																												
hidden	__VIEWSTATEGENERATOR																												
hidden	__EVENTTARGET																												
hidden	__EVENTARGUMENT																												
hidden	__EVENTVALIDATION																												
checkbox	flight																												
checkbox	hotel																												
checkbox	cruise																												
text	txtSearch																												
submit	Button1																												
Scan Request	www.vulnerable-website.com:80/View.aspx?nameid=print(md5(1234567));exit;																												
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> PHP wikipedia 																												



Medium Severity Findings

Possibility of Cross Frame Scripting
Script: Search.aspx **Variable:** txtSearch
Value: "></script><iframe src=http://www.gamascan.com>

Description	<p>GamaScan has identified a flaw that may allow an attacker to bypass certain frame restrictions. The flaw occurs due to use of unsanitized user-supplied data, which may allow an attacker to conduct phishing attacks.</p> <div style="background-color: #f0f0f0; padding: 2px; text-align: center; border: 1px solid #ccc;">Form Method: POST</div> <div style="background-color: #f0f0f0; padding: 2px; text-align: center; border: 1px solid #ccc;">The Form Inputs</div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f0f0f0;"> <th style="text-align: left;">Type</th> <th style="text-align: left;">Name</th> </tr> </thead> <tbody> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td>checkbox</td><td>flight</td></tr> <tr><td>checkbox</td><td>hotel</td></tr> <tr><td>checkbox</td><td>cruise</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtSearch</td></tr> <tr><td>submit</td><td>Button1</td></tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
Type	Name																								
hidden	__LASTFOCUS																								
hidden	__VIEWSTATE																								
hidden	__VIEWSTATEGENERATOR																								
hidden	__EVENTTARGET																								
hidden	__EVENTARGUMENT																								
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								
Scan Request	<p><a ><="" href="http://www.vulnerable-website.com:80/Search.aspx?key=" script><iframe%20src='http://www.gamascan.com"'>www.vulnerable-website.com:80/Search.aspx?key="></script><iframe%20src=http://www.gamascan.com</p>																								
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <p> owasp</p>																								



Possibility of Cookie Manipulation																									
Script: Search.aspx Variable: txtSearch																									
Value: "><meta http-equiv="Set-cookie" content="GamaScan=CM">																									
Description	GamaScan has identified a flaw that may allow an attacker to inject a META tag. The flaw occurs due to use of unsanitized user-supplied data , which enables attacker to alter or modify the cookies values, on the clients web browser. This flaw may lead to an attacker gaining unauthorised access, leading to a loss of integrity.																								
	Form Method: POST																								
	The Form Inputs																								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Type</th> <th>Name</th> </tr> </thead> <tbody> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td>checkbox</td><td>flight</td></tr> <tr><td>checkbox</td><td>hotel</td></tr> <tr><td>checkbox</td><td>cruise</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtSearch</td></tr> <tr><td>submit</td><td>Button1</td></tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
	Type	Name																							
	hidden	__LASTFOCUS																							
	hidden	__VIEWSTATE																							
	hidden	__VIEWSTATEGENERATOR																							
	hidden	__EVENTTARGET																							
	hidden	__EVENTARGUMENT																							
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								
Scan Request	<a %20content="GamaScan=CM" >"="" ><meta%20http-equiv="Set-cookie" href="http://www.vulnerable-website.com:80/Search.aspx?key=">www.vulnerable-website.com:80/Search.aspx?key="><meta%20http-equiv="Set-cookie"%20content="GamaScan=CM">																								
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> owasp cgisecurity 																								



Possibility of Cross Frame Scripting																					
Script: LogIn.aspx Variable: txtUserName																					
Value: "></script><iframe src=http://www.gamascan.com>																					
Description	GamaScan has identified a flaw that may allow an attacker to bypass certain frame restrictions. The flaw occurs due to use of unsanitized user-supplied data, which may allow an attacker to conduct phishing attacks.																				
	Form Method: POST																				
	The Form Inputs																				
	<table border="1"> <thead> <tr> <th>Type</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>hidden</td> <td>__LASTFOCUS</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATE</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATEGENERATOR</td> </tr> <tr> <td>hidden</td> <td>__EVENTTARGET</td> </tr> <tr> <td>hidden</td> <td>__EVENTARGUMENT</td> </tr> <tr> <td>hidden</td> <td>__EVENTVALIDATION</td> </tr> <tr> <td style="color: red;">text</td> <td style="color: red;">txtUserName</td> </tr> <tr> <td>password</td> <td>txtPassword</td> </tr> <tr> <td>submit</td> <td>btnLogIn</td> </tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	text	txtUserName	password	txtPassword	submit	btnLogIn
	Type	Name																			
	hidden	__LASTFOCUS																			
	hidden	__VIEWSTATE																			
	hidden	__VIEWSTATEGENERATOR																			
	hidden	__EVENTTARGET																			
	hidden	__EVENTARGUMENT																			
hidden	__EVENTVALIDATION																				
text	txtUserName																				
password	txtPassword																				
submit	btnLogIn																				
Scan Request	<a ><="" href="http://www.vulnerable-website.com:80/LogIn.aspx?UserName=" script><iframe%20src='http://www.gamascan.com>"'>www.vulnerable-website.com:80/LogIn.aspx?UserName="></script><iframe%20src=http://www.gamascan.com>																				
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: owasp																				



Possibility of Cookie Manipulation																					
Script: LogIn.aspx Variable: txtUserName Value: "><meta http-equiv="Set-cookie" content="GamaScan=CM">																					
Description	GamaScan has identified a flaw that may allow an attacker to inject a META tag. The flaw occurs due to use of unsanitized user-supplied data, which enables attacker to alter or modify the cookies values, on the clients web browser. This flaw may lead to an attacker gaining unauthorised access, leading to a loss of integrity.																				
	Form Method: POST																				
	The Form Inputs																				
	<table border="1"> <thead> <tr> <th>Type</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>hidden</td> <td>__LASTFOCUS</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATE</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATEGENERATOR</td> </tr> <tr> <td>hidden</td> <td>__EVENTTARGET</td> </tr> <tr> <td>hidden</td> <td>__EVENTARGUMENT</td> </tr> <tr> <td>hidden</td> <td>__EVENTVALIDATION</td> </tr> <tr> <td>text</td> <td>txtUserName</td> </tr> <tr> <td>password</td> <td>txtPassword</td> </tr> <tr> <td>submit</td> <td>btnLogIn</td> </tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	text	txtUserName	password	txtPassword	submit	btnLogIn
	Type	Name																			
hidden	__LASTFOCUS																				
hidden	__VIEWSTATE																				
hidden	__VIEWSTATEGENERATOR																				
hidden	__EVENTTARGET																				
hidden	__EVENTARGUMENT																				
hidden	__EVENTVALIDATION																				
text	txtUserName																				
password	txtPassword																				
submit	btnLogIn																				
Scan Request	<a %20content="GamaScan=CM" >"="" ><meta%20http-equiv="Set-cookie" href="http://www.vulnerable-website.com:80/LogIn.aspx?UserName=">www.vulnerable-website.com:80/LogIn.aspx?UserName="><meta%20http-equiv="Set-cookie"%20content="GamaScan=CM">																				
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> owasp cgisecurity 																				

Possibility of Cross-Frame Scripting	
Script: Search.aspx Variable: key Value: "></script><iframe src=http://www.gamascan.com>	
Description	GamaScan has identified a flaw that may allow an attacker to bypass certain frame restrictions. The flaw occurs due to use of unsanitized user-supplied data, which may allow an attacker to conduct phishing attacks.
Scan Request	<a ><="" href="http://www.vulnerable-website.com:80/Search.aspx?key=" script><iframe%20src='http://www.gamascan.com>"'>www.vulnerable-website.com:80/Search.aspx?key="></script><iframe%20src=http://www.gamascan.com>
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> owasp



Possibility of Cookie Manipulation Script: Search.aspx Variable: key Value: "><meta http-equiv="Set-cookie" content="GamaScan=CM">	
Description	GamaScan has identified a flaw that may allow an attacker to inject a META tag. The flaw occurs due to use of unsanitized user-supplied data, which enables attacker to alter or modify the cookies values, on the clients web browser. This flaw may lead to an attacker gaining unauthorised access, leading to a loss of integrity.
Scan Request	<a %20content="GamaScan=CM" >"="" ><meta%20http-equiv="Set-cookie" href="http://www.vulnerable-website.com:80/Search.aspx?key=">www.vulnerable-website.com:80/Search.aspx?key="><meta%20http-equiv="Set-cookie"%20content="GamaScan=CM">
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none">  owasp  cgisecurity



Possibility of Cross Frame Scripting																									
Script: View.aspx Variable: txtSearch																									
Value: "></script><iframe src=http://www.gamascan.com>																									
Description	GamaScan has identified a flaw that may allow an attacker to bypass certain frame restrictions. The flaw occurs due to use of unsanitized user-supplied data, which may allow an attacker to conduct phishing attacks.																								
	Form Method: POST																								
	The Form Inputs																								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Type</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>hidden</td> <td>__LASTFOCUS</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATE</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATEGENERATOR</td> </tr> <tr> <td>hidden</td> <td>__EVENTTARGET</td> </tr> <tr> <td>hidden</td> <td>__EVENTARGUMENT</td> </tr> <tr> <td>hidden</td> <td>__EVENTVALIDATION</td> </tr> <tr> <td>checkbox</td> <td>flight</td> </tr> <tr> <td>checkbox</td> <td>hotel</td> </tr> <tr> <td>checkbox</td> <td>cruise</td> </tr> <tr> <td style="color: red;">text</td> <td style="color: red;">txtSearch</td> </tr> <tr> <td>submit</td> <td>Button1</td> </tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
	Type	Name																							
	hidden	__LASTFOCUS																							
	hidden	__VIEWSTATE																							
	hidden	__VIEWSTATEGENERATOR																							
	hidden	__EVENTTARGET																							
	hidden	__EVENTARGUMENT																							
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								
Scan Request	<a "="" href="http://www.vulnerable-website.com:80/View.aspx?nameid=">></script><iframe%20src=http://www.gamascan.com>																								
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <p>owasp</p>																								



Possibility of Cookie Manipulation																									
Script: View.aspx Variable: txtSearch																									
Value: "><meta http-equiv="Set-cookie" content="GamaScan=CM">																									
Description	GamaScan has identified a flaw that may allow an attacker to inject a META tag. The flaw occurs due to use of unsanitized user-supplied data , which enables attacker to alter or modify the cookies values, on the clients web browser. This flaw may lead to an attacker gaining unauthorised access, leading to a loss of integrity.																								
	Form Method: POST																								
	The Form Inputs																								
	<table border="1"> <thead> <tr> <th>Type</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>hidden</td> <td>__LASTFOCUS</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATE</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATEGENERATOR</td> </tr> <tr> <td>hidden</td> <td>__EVENTTARGET</td> </tr> <tr> <td>hidden</td> <td>__EVENTARGUMENT</td> </tr> <tr> <td>hidden</td> <td>__EVENTVALIDATION</td> </tr> <tr> <td>checkbox</td> <td>flight</td> </tr> <tr> <td>checkbox</td> <td>hotel</td> </tr> <tr> <td>checkbox</td> <td>cruise</td> </tr> <tr> <td>text</td> <td>txtSearch</td> </tr> <tr> <td>submit</td> <td>Button1</td> </tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
	Type	Name																							
	hidden	__LASTFOCUS																							
	hidden	__VIEWSTATE																							
	hidden	__VIEWSTATEGENERATOR																							
	hidden	__EVENTTARGET																							
	hidden	__EVENTARGUMENT																							
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								
Scan Request	<a %20content="GamaScan=CM" >"="" ><meta%20http-equiv="Set-cookie" href="http://www.vulnerable-website.com:80/View.aspx?nameid=">www.vulnerable-website.com:80/View.aspx?nameid="><meta%20http-equiv="Set-cookie"%20content="GamaScan=CM">																								
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <ul style="list-style-type: none"> owasp cgisecurity 																								



Low Severity Findings

Possibility of URL redirection
Script: Search.aspx **Variable:** txtSearch
Value: "><meta http-equiv="Refresh" content="0;url=http://www.gamascan.com/">

Description	<p>GamaScan has identified a flaw that may allow an attacker to redirect visitors from your website to another one. The flaw occurs due to use of unsanitized user-supplied data, which may allow an attacker to conduct phishing attacks.</p> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px; text-align: center;">Form Method: POST</div> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 2px; text-align: center;">The Form Inputs</div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #d3d3d3;"> <th style="text-align: left;">Type</th> <th style="text-align: left;">Name</th> </tr> </thead> <tbody> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td>checkbox</td><td>flight</td></tr> <tr><td>checkbox</td><td>hotel</td></tr> <tr><td>checkbox</td><td>cruise</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtSearch</td></tr> <tr><td>submit</td><td>Button1</td></tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
Type	Name																								
hidden	__LASTFOCUS																								
hidden	__VIEWSTATE																								
hidden	__VIEWSTATEGENERATOR																								
hidden	__EVENTTARGET																								
hidden	__EVENTARGUMENT																								
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								

Scan Request	<a %20content="0;url=http://www.gamascan.com/" >"="" ><meta%20http-equiv="Refresh" href="http://www.vulnerable-website.com:80/Search.aspx?key=">www.vulnerable-website.com:80/Search.aspx?key="><meta%20http-equiv="Refresh"%20content="0;url=http://www.gamascan.com/">
---------------------	--

Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References:</p> <p>wikipedia</p>
-----------------------	--



Possibility of URL redirection Script: LogIn.aspx Variable: txtUserName Value: "><meta http-equiv="Refresh" content="0;url=http://www.gamascan.com/">																									
Description	<p>GamaScan has identified a flaw that may allow an attacker to redirect visitors from your website to another one. The flaw occurs due to use of unsanitized user-supplied data, which may allow an attacker to conduct phishing attacks.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="background-color: #cccccc;">Form Method: POST</th> </tr> <tr> <th colspan="2" style="background-color: #cccccc;">The Form Inputs</th> </tr> <tr> <th style="width: 30%;">Type</th> <th>Name</th> </tr> </thead> <tbody> <tr><td>hidden</td><td>__LASTFOCUS</td></tr> <tr><td>hidden</td><td>__VIEWSTATE</td></tr> <tr><td>hidden</td><td>__VIEWSTATEGENERATOR</td></tr> <tr><td>hidden</td><td>__EVENTTARGET</td></tr> <tr><td>hidden</td><td>__EVENTARGUMENT</td></tr> <tr><td>hidden</td><td>__EVENTVALIDATION</td></tr> <tr><td style="color: red;">text</td><td style="color: red;">txtUserName</td></tr> <tr><td>password</td><td>txtPassword</td></tr> <tr><td>submit</td><td>btnLogIn</td></tr> </tbody> </table>	Form Method: POST		The Form Inputs		Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	text	txtUserName	password	txtPassword	submit	btnLogIn
Form Method: POST																									
The Form Inputs																									
Type	Name																								
hidden	__LASTFOCUS																								
hidden	__VIEWSTATE																								
hidden	__VIEWSTATEGENERATOR																								
hidden	__EVENTTARGET																								
hidden	__EVENTARGUMENT																								
hidden	__EVENTVALIDATION																								
text	txtUserName																								
password	txtPassword																								
submit	btnLogIn																								
Scan Request	<a %20content="0;url=http://www.gamascan.com/" >"="" ><meta%20http-equiv="Refresh" href="http://www.vulnerable-website.com:80/LogIn.aspx?UserName=">www.vulnerable-website.com:80/LogIn.aspx?UserName="><meta%20http-equiv="Refresh"%20content="0;url=http://www.gamascan.com/">																								
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References: wikipedia</p>																								

Possibility of URL redirection Script: Search.aspx Variable: key Value: "><meta http-equiv="Refresh" content="0;url=http://www.gamascan.com/">	
Description	<p>GamaScan has identified a flaw that may allow an attacker to redirect visitors from your website to another one. The flaw occurs due to use of unsanitized user-supplied data, which may allow an attacker to conduct phishing attacks.</p>
Scan Request	<a %20content="0;url=http://www.gamascan.com/" >"="" ><meta%20http-equiv="Refresh" href="http://www.vulnerable-website.com:80/Search.aspx?key=">www.vulnerable-website.com:80/Search.aspx?key="><meta%20http-equiv="Refresh"%20content="0;url=http://www.gamascan.com/">
Recommendation	<p>It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input.</p> <p>External References: wikipedia</p>



Possibility of URL redirection Script: View.aspx Variable: txtSearch Value: "><meta http-equiv="Refresh" content="0;url=http://www.gamascan.com/">																									
Description	GamaScan has identified a flaw that may allow an attacker to redirect visitors from your website to another one. The flaw occurs due to use of unsanitized user-supplied data, which may allow an attacker to conduct phishing attacks.																								
	Form Method: POST																								
	The Form Inputs																								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Type</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>hidden</td> <td>__LASTFOCUS</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATE</td> </tr> <tr> <td>hidden</td> <td>__VIEWSTATEGENERATOR</td> </tr> <tr> <td>hidden</td> <td>__EVENTTARGET</td> </tr> <tr> <td>hidden</td> <td>__EVENTARGUMENT</td> </tr> <tr> <td>hidden</td> <td>__EVENTVALIDATION</td> </tr> <tr> <td>checkbox</td> <td>flight</td> </tr> <tr> <td>checkbox</td> <td>hotel</td> </tr> <tr> <td>checkbox</td> <td>cruise</td> </tr> <tr> <td style="color: red;">text</td> <td style="color: red;">txtSearch</td> </tr> <tr> <td>submit</td> <td>Button1</td> </tr> </tbody> </table>	Type	Name	hidden	__LASTFOCUS	hidden	__VIEWSTATE	hidden	__VIEWSTATEGENERATOR	hidden	__EVENTTARGET	hidden	__EVENTARGUMENT	hidden	__EVENTVALIDATION	checkbox	flight	checkbox	hotel	checkbox	cruise	text	txtSearch	submit	Button1
	Type	Name																							
	hidden	__LASTFOCUS																							
	hidden	__VIEWSTATE																							
	hidden	__VIEWSTATEGENERATOR																							
	hidden	__EVENTTARGET																							
	hidden	__EVENTARGUMENT																							
hidden	__EVENTVALIDATION																								
checkbox	flight																								
checkbox	hotel																								
checkbox	cruise																								
text	txtSearch																								
submit	Button1																								
Scan Request	<a "="" href="http://www.vulnerable-website.com:80/View.aspx?nameid=">><meta%20http-equiv="Refresh"%20content="0;url=http://www.gamascan.com/">																								
Recommendation	It is possible to correct the flaw by manually editing the code to properly sanitize the user-supplied input. External References: wikipedia																								



Info Severity Findings

Directory Enumeration	
Description	GamaScan has identified a directory on the server.
Scan Request	www.vulnerable-website.com:80/system/
Recommendation	No fix is required. External References: Web Security Glossary

Directory Enumeration	
Description	GamaScan has identified a directory on the server.
Scan Request	www.vulnerable-website.com:80/js/
Recommendation	No fix is required. External References: Web Security Glossary

Directory Enumeration	
Description	GamaScan has identified a directory on the server.
Scan Request	www.vulnerable-website.com:80/images/
Recommendation	No fix is required. External References: Web Security Glossary

Directory Enumeration	
Description	GamaScan has identified a directory on the server.
Scan Request	www.vulnerable-website.com:80/aspnet_client/
Recommendation	No fix is required. External References: Web Security Glossary

This product includes data from the Open Source Vulnerability Database developed by OSVDB (www.osvdb.org) and its contributors.

Please consider your environmental responsibility before printing this report.



GamaScan The Web Application Security Solution

GamaScan is a remote online web vulnerability-assessment service that tests web servers, web-interfaced systems and web-based applications against thousands of known vulnerabilities with dynamic testing, and by simulating web-application attacks during online scanning. The service identifies security vulnerabilities and produces recommended solutions that can fix, or provide a viable workaround to the identified vulnerabilities. GamaScan provides a flexible, interactive security scan which requires no installation, no setup, no new hardware, no software development, no security expertise and no special training.



Scan your web site today with GamaScan and find vulnerabilities before hackers do!



Technical	support@gamasec.com
Sales	sales@gamasec.com
Information	info@gamasec.com
Partner Program	partnerprogram@gamasec.com



By displaying the GamaSec security seal your customers know your website is secure.



GamaSec next generation of Web application Security Scanner and malware detection allow you to identify and eradicate Web Vulnerabilities and destructive Malware.

GAMASEC VERIFIED THIS WEBSITE

Domain Name	www.gamasec.com
Valid	DD/MM/YYYY
Service	GamaShield - Application Scanner & Malware Detection

GamaSec runs a number of cloud security services. The security seal specifies that the website uses one of the services developed by GamaSec.

Seal Disclaimer:GamaSec makes no warranty or guarantee of any kind of the accuracy of information presented on the Site, nor that the Site is completely secure or safe, nor that user data can't be compromised by hackers or other third parties. Furthermore, GamaSec is in no way responsible for and shall be held harmless against any claims for the security of or use of any information stored or utilized on the Site.

For more info Visit our website : www.gamasec.com

