ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	1
Low	5
Informational	0

Alert Detail

High (Medium)	Path Traversal
Description	The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTP-based interface is potentially vulnerable to Path Traversal. Most web sites restrict user access to a specific portion of the file-system, typically called the "web document root" or "CGI root" directory. These directories contain the files intended for user access and the executable necessary to drive web application functionality. To access files or execute commands anywhere on the file-system, Path Traversal attacks will utilize the ability of special-characters sequences. The most basic Path Traversal attack uses the "../" special-character sequence to alter the resource location requested in the URL. Although most popular web servers will prevent this technique from escaping the web document root, alternate encodings of the "../" sequence may help bypass the security filters. These method variations include valid and invalid Unicode-encoding ("..%u2216" or "..%c0%af") of the forward slash character, backslash characters ("..\") on Windows-based servers, URL encoded characters "%2e%2e%2f"), and double URL encoding ("..%255c") of the backslash character. Even if the web server properly restricts Path Traversal attempts in the URL path, a web application itself may still be vulnerable due to improper handling of user-supplied input. This is a common problem of web applications that use template mechanisms or load static text from files. In variations of the attack, the original URL parameter value is substituted with the file name of one of the web application's dynamic scripts. Consequently, the results can reveal source code because the file is interpreted as text instead of an executable script. These techniques often employ additional special characters such as the dot (".") to reveal the listing of the current working directory, or "%00" NULL characters in order to bypass rudimentary file extension checks.
URL	https://dev.investorstatelawguide.com/start-your-trial?City=ZAP&Country=Albania&Email=start-your-trial&EnableEditProfile=1&Firm_Institution=ZAP&FirstName=ZAP&I_am_18years_or_Older=1&LastName=ZAP&Position=ZAP&SheduleTutorial=false&TermsConditions=1&active=0&g-recaptcha-response=&id=54&save=y&signup=y&toc=content
Method	POST
Parameter	Email
Attack	start-your-trial
Instances	1
Solution	Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not

	strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a blacklist). However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue." For filenames, use stringent whitelists that limit the character set to be used. If feasible, only allow a single "." character in the filename to avoid weaknesses, and exclude directory separators such as "/". Use a whitelist of allowable file extensions. Warning: if you attempt to cleanse your data, then do so that the end result is not in the form that can be dangerous. A sanitizing mechanism can remove characters such as '.' and ';' which may be required for some exploits. An attacker can try to fool the sanitizing mechanism into "cleaning" data into a dangerous form. Suppose the attacker injects a '.' inside a filename (e.g. "sensi.tiveFile") and the sanitizing mechanism removes the character resulting in the valid filename, "sensitiveFile". If the input data are now assumed to be safe, then the file may be compromised. Inputs should be decoded and canonicalized to the application's current internal representation before being validated. Make sure that your application does not decode the same input twice. Such errors could be used to bypass whitelist schemes by introducing dangerous inputs after they have been checked. Use a built-in path canonicalization function (such as realpath() in C) that produces the canonical version of the pathname, which effectively removes ".." sequences and symbolic links. Run your code using the lowest privileges that are required to accomplish the necessary tasks. If possible, create isolated accounts with limited privileges that are only used for a single task. That way, a successful attack will not immediately give the attacker access to the rest of the software or its environment. For example, database applications rarely need to run as the database administrator, especially in day-to-day operations. When the set of acceptable objects, such as filenames or URLs, is limited or known, create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs. Run your code in a "jail" or similar sandbox environment that enforces strict boundaries between the process and the operating system. This may effectively restrict which files can be accessed in a particular directory or which commands can be executed by your software. OS-level examples include the Unix chroot jail, AppArmor, and SELinux. In general, managed code may provide some protection. For example, java.io.FilePermission in the Java SecurityManager allows you to specify restrictions on file operations. This may not be a feasible solution, and it only limits the impact to the operating system; the rest of your application may still be subject to compromise.
Reference	http://projects.webappsec.org/Path-Traversal http://cwe.mitre.org/data/definitions/22.html
CWE Id	22
WASC Id	33
Source ID	1

Medium (Medium)	**Format String Error**
Description	A Format String error occurs when the submitted data of an input string is evaluated as a command by the application.
URL	https://dev.investorstatelawguide.com/Home/Register?

	City=ZAP&Country=Albania&Email=ZAP+%251%21s%252%21s%253%21s%254%21s%255%21s%256%21s%257%21s%258%21s%259%21s%2510%21s%2511%21s%2512%21s%2513%21s%2514%21s%2515%21s%2516%21s%2517%21s%2518%21s%2519%21s%2520%21s%2521%21n%2522%21n%2523%21n%2524%21n%2525%21n%2526%21n%2527%21n%2528%21n%2529%21n%2530%21n%2531%21n%2532%21n%2533%21n%2534%21n%2535%21n%2536%21n%2537%21n%2538%21n%2539%21n%2540%21n%0A&EnableEditProfile=1&Firm_Institution=ZAP&FirstName=ZAP&I_am_18years_or_Older=1&LastName=ZAP&Length=0&Position=ZAP&SheduleTutorial=false&TermsConditions=1&active=0&g-recaptcha-response=&id=54&save=y&signup=y&toc=content
Method	POST
Parameter	Email
Attack	ZAP %1!s%2!s%3!s%4!s%5!s%6!s%7!s%8!s%9!s%10!s%11!s%12!s%13!s%14!s%15!s%16!s%17!s%18!s%19!s%20!s%21!n%22!n%23!n%24!n%25!n%26!n%27!n%28!n%29!n%30!n%31!n%32!n%33!n%34!n%35!n%36!n%37!n%38!n%39!n%40!n
URL	https://dev.investorstatelawguide.com/start-your-trial?City=ZAP&Country=Albania&Email=ZAP+%251%21s%252%21s%253%21s%254%21s%255%21s%256%21s%257%21s%258%21s%259%21s%2510%21s%2511%21s%2512%21s%2513%21s%2514%21s%2515%21s%2516%21s%2517%21s%2518%21s%2519%21s%2520%21s%2521%21n%2522%21n%2523%21n%2524%21n%2525%21n%2526%21n%2527%21n%2528%21n%2529%21n%2530%21n%2531%21n%2532%21n%2533%21n%2534%21n%2535%21n%2536%21n%2537%21n%2538%21n%2539%21n%2540%21n%0A&EnableEditProfile=1&Firm_Institution=ZAP&FirstName=ZAP&I_am_18years_or_Older=1&LastName=ZAP&Position=ZAP&SheduleTutorial=false&TermsConditions=1&active=0&g-recaptcha-response=&id=54&save=y&signup=y&toc=content
Method	POST
Parameter	Email
Attack	ZAP %1!s%2!s%3!s%4!s%5!s%6!s%7!s%8!s%9!s%10!s%11!s%12!s%13!s%14!s%15!s%16!s%17!s%18!s%19!s%20!s%21!n%22!n%23!n%24!n%25!n%26!n%27!n%28!n%29!n%30!n%31!n%32!n%33!n%34!n%35!n%36!n%37!n%38!n%39!n%40!n
Instances	2
Solution	Rewrite the background program using proper deletion of bad character strings. This will require a recompile of the background executable.
Other information	Potential Format String Error. The script closed the connection on a microsoft format string error
Reference	https://www.owasp.org/index.php/Format_string_attack
CWE Id	134
WASC Id	6
Source ID	1

Low (Medium)	**Cross-Domain JavaScript Source File Inclusion**
Description	The page includes one or more script files from a third-party domain.
URL	https://dev.investorstatelawguide.com/contact-us?g-recaptcha-response
Method	GET
Parameter	https://www.google.com/recaptcha/api.js?render=reCAPTCHA_site_key
Evidence	<script type="text/javascript" src="https://www.google.com/recaptcha/api.js?render=reCAPTCHA_site_key"></script>
URL	https://dev.investorstatelawguide.com/assets/icons/android-chrome-512x512.png
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js

Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/about-islg/Ivaylo_Dimitrov
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/Home/ForGetPassword?token=%2fJJ1P12gJvRpftktHjszEg%3d%3d
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/sitemap.xml
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/about-islg/Ximena_Iturriaga
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/about-islg/Madalena_Beaudet
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/about-islg/Sapna_Jhangiani
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/Home/Register?Length=0
Method	GET
Parameter	https://www.google.com/recaptcha/api.js?render=reCAPTCHA_site_key
Evidence	<script type="text/javascript" src="https://www.google.com/recaptcha/api.js?render=reCAPTCHA_site_key"></script>
URL	https://dev.investorstatelawguide.com/about-islg/Jennifer_Radford

Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/about-islg/Irit_Weinfeld
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/about-islg/Olga_Boltenko
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/about-islg/Diana_Ruiz
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/product
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/assets/icons/apple-touch-icon.png
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/Home/ForGetPassWord?Length=0
Method	POST
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/Views/MasterPages/image/Gear-Favicon.ico
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
URL	https://dev.investorstatelawguide.com/recaptcha/api/fallback?k=6Le-wvkSAAAAAPBMRTvw0Q4Muexq9bi0DJwx_mJ-
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
Instances	70
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Reference	

CWE Id	829
WASC Id	15
Source ID	3

Low (Medium)	**Absence of Anti-CSRF Tokens**
Description	No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

URL	https://dev.investorstatelawguide.com/start-your-trial
Method	GET
Evidence	<form class="form">
URL	https://dev.investorstatelawguide.com/admin/adminlogin
Method	GET
Evidence	<form action="/admin/adminlogin" id="form0" method="post">
URL	https://dev.investorstatelawguide.com/start-your-trial?City=ZAP&Country=Albania&Email=ZAP&EnableEditProfile=1&Firm_Institution=ZAP&FirstName=ZAP&I_am_18years_or_Older=1&LastName=ZAP&Position=ZAP&SheduleTutorial=false&TermsConditions=1&active=0&g-recaptcha-response&id=54&save=y&signup=y&toc=content
Method	GET
Evidence	<form class="form">
URL	https://dev.investorstatelawguide.com/contact-us
Method	GET
Evidence	<form class="form" novalidate>
URL	https://dev.investorstatelawguide.com/contact-us?g-recaptcha-response
Method	GET
Evidence	<form class="form" novalidate>
URL	https://dev.investorstatelawguide.com/Home/Register?Length=0
Method	GET
Evidence	<form class="form">
URL	https://dev.investorstatelawguide.com/Home/Register?City=ZAP&Country=Albania&Email=ZAP&EnableEditProfile=1&Firm_Institution=ZAP&FirstName=

	ZAP&I_am_18years_or_Older=1&LastName=ZAP&Length=0&Position=ZAP&SheduleTutorial=false&TermsConditions=1&active=0&g-recaptcha-response&id=54&save=y&signup=y&toc=content
Method	GET
Evidence	<form class="form">
URL	https://dev.investorstatelawguide.com/admin/adminlogin
Method	POST
Evidence	<form action="/admin/adminlogin" id="form0" method="post">
URL	https://dev.investorstatelawguide.com/Home/Register?City=ZAP&Country=Albania&Email=ZAP&EnableEditProfile=1&Firm_Institution=ZAP&FirstName=ZAP&I_am_18years_or_Older=1&LastName=ZAP&Length=0&Position=ZAP&SheduleTutorial=false&TermsConditions=1&active=0&g-recaptcha-response&id=54&save=y&signup=y&toc=content
Method	POST
Evidence	<form class="form">
Instances	9
Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS. Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.
Other information	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token] was found in the following HTML form: [Form 2: "toc" "id" "signup" "save" "active" "EnableEditProfile" "textLabel" "FirstName" "LastName" "txtEmail" "Firm_Institution" "Position" "City" "TermsConditions" "I_am_18years_or_Older" "SheduleTutorial" "SheduleTutorial"].
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352

WASC Id	9
Source ID	3

Low (Medium)	**Information Disclosure - Debug Error Messages**
Description	The response appeared to contain common error messages returned by platforms such as ASP.NET, and Web-servers such as IIS and Apache. You can configure the list of common debug messages.
URL	https://dev.investorstatelawguide.com/start-your-trial
Method	POST
Evidence	customErrors mode
URL	https://dev.investorstatelawguide.com/Home/Register?Length=0
Method	POST
Evidence	customErrors mode
Instances	2
Solution	Disable debugging messages before pushing to production.
Reference	
CWE Id	200
WASC Id	13
Source ID	3

Low (Medium)	**Cookie Without Secure Flag**
Description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
URL	https://dev.investorstatelawguide.com/notepad-login
Method	GET
Parameter	ASP.NET_SessionId
Evidence	Set-Cookie: ASP.NET_SessionId
URL	https://dev.investorstatelawguide.com/notepad-login
Method	GET
Parameter	__RequestVerificationToken_Lw__
Evidence	Set-Cookie: __RequestVerificationToken_Lw__
URL	https://dev.investorstatelawguide.com/notepad-login
Method	GET
Parameter	UserId
Evidence	Set-Cookie: UserId
URL	https://dev.investorstatelawguide.com/notepad-login
Method	GET
Parameter	Password
Evidence	Set-Cookie: Password
Instances	4
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such

	sensitive information.
Reference	http://www.owasp.org/index.php/Testing_for_cookies_attributes_(OWASP-SM-002)
CWE Id	614
WASC Id	13
Source ID	3

Low (Medium)	**Application Error Disclosure**
Description	This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page.
URL	https://dev.investorstatelawguide.com/start-your-trial
Method	POST
Evidence	HTTP/1.1 500 Internal Server Error
URL	https://dev.investorstatelawguide.com/Home/Register?Length=0
Method	POST
Evidence	HTTP/1.1 500 Internal Server Error
Instances	2
Solution	Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.
Reference	
CWE Id	200
WASC Id	13
Source ID	3