ZAP Scanning Report

Site: https://www.secureblink.com

Generated on Sat, 22 Jan 2022 09:29:56

Summary of Alerts

Risk Level	Number of Alerts
High	2
Medium	4
Low	8
Informational	5
False Positives:	0

Alerts

Name	Risk Level	Number of Instances
Anti-CSRF Tokens Check	High	12
SQL Injection - SQLite	High	2
Application Error Disclosure	Medium	1
Content Security Policy (CSP) Header Not Set	Medium	10
Cross-Domain Misconfiguration	Medium	12
Missing Anti-clickjacking Header	Medium	11
Absence of Anti-CSRF Tokens	Low	12
Application Error Disclosure	Low	3
Cross-Domain JavaScript Source File Inclusion	Low	12
Incomplete or No Cache-control Header Set	Low	12
Private IP Disclosure	Low	3
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	11
Strict-Transport-Security Header Not Set	Low	3
Timestamp Disclosure - Unix	Low	17
Content Security Policy (CSP) Report-Only Header Found	Informational	1
Information Disclosure - Sensitive Information in URL	Informational	10
Information Disclosure - Suspicious Comments	Informational	11
Modern Web Application	Informational	11
Retrieved from Cache	Informational	11

Alert Detail

High	Anti-CSRF Tokens Check
Description	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://www.secureblink.com/company-register
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-869139563 mt-4">
URL	https://www.secureblink.com/company-register?companyName=ZAP&email=foo-bar%40example.com&name=ZAP&password=ZAP
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-869139563 mt-4">
URL	https://www.secureblink.com/contact-us
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-3992399362 mt-12">
URL	https://www.secureblink.com/contact-us?company=ZAP&email=foo-bar%40example.com&job=ZAP&message&name=ZAP&phone=9999999999
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-3992399362 mt-12">
URL	https://www.secureblink.com/register
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-869139563 mt-4">
URL	https://www.secureblink.com/register?email=foo-bar%40example.com&name=ZAP&password=ZAP&username=ZAP
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-869139563 mt-4">
URL	https://www.secureblink.com/signin
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-981351468 mt-4">
URL	https://www.secureblink.com/signin?email_or_username=ZAP&password=ZAP
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-981351468 mt-4">
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-600599106 mt-12">
URL	https://www.secureblink.com/threat-spy?email=foo-bar%40example.com&message&name=ZAP&phone=9999999999
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-600599106 mt-12">

URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2467518051 mt-8">
URL	https://www.secureblink.com/white-paper?email=foo-bar%40example.com&name=ZAP&phone=9999999999
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2467518051 mt-8">
Instances	12
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.
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	20012

High	**SQL Injection - SQLite**
Description	SQL injection may be possible
URL	https://www.secureblink.com/threat-spy?email=foo-bar%40example.com&message&name=ZAP&phone=9999999999
Method	GET
Parameter	message
Attack	case randomblob(10000000) when not null then 1 else 1 end
Evidence	The query time is controllable using parameter value [case randomblob(10000000) when not null then 1 else 1 end], which caused the request to take [459] milliseconds, parameter value [case randomblob(100000000) when not null then 1 else 1 end], which caused the request to take [1,407] milliseconds, when the original unmodified query with value [] took [329] milliseconds.
URL	https://www.secureblink.com/white-paper?email=foo-bar%40example.com&name=ZAP&phone=9999999999
Method	GET
Parameter	email
Attack	case randomblob(10000000) when not null then 1 else 1 end
Evidence	The query time is controllable using parameter value [case randomblob(10000000) when not null then 1 else 1 end], which caused the request to take [638] milliseconds, parameter value [case randomblob(100000000) when not null then 1 else 1 end], which caused the request to take [2,220] milliseconds, when the original unmodified query with

	value [foo-bar@example.com] took [621] milliseconds.
Instances	2
Solution	Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side. If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by '?' If the application uses ASP, use ADO Command Objects with strong type checking and parameterized queries. If database Stored Procedures can be used, use them. Do *not* concatenate strings into queries in the stored procedure, or use 'exec', 'exec immediate', or equivalent functionality! Do not create dynamic SQL queries using simple string concatenation. Escape all data received from the client. Apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input. Apply the privilege of least privilege by using the least privileged database user possible. In particular, avoid using the 'sa' or 'db-owner' database users. This does not eliminate SQL injection, but minimizes its impact. Grant the minimum database access that is necessary for the application.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
CWE Id	89
WASC Id	19
Plugin Id	40024

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://www.secureblink.com/_next/static/chunks/pages/index-ed95606da377e7c58d7a.js
Method	GET
Parameter	
Attack	
Evidence	Internal Server Error
Instances	1
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
Plugin Id	90022

Medium	**Content Security Policy (CSP) Header Not Set**
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/about-us
Method	GET
Parameter	
Attack	
Evidence	

URL	https://www.secureblink.com/blog
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/contact-us
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/robots.txt
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/solutions
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/threat-research
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	
Attack	
Evidence	
Instances	10
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header, to achieve optimal browser support: "Content-Security-Policy" for Chrome 25+, Firefox 23+ and Safari 7+, "X-Content-Security-Policy" for Firefox 4.0+ and Internet Explorer 10+, and "X-WebKit-CSP" for Chrome 14+ and Safari 6+.
Reference	https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html http://www.w3.org/TR/CSP/ http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html http://www.html5rocks.com/en/tutorials/security/content-security-policy/ http://caniuse.com/#feat=contentsecuritypolicy http://content-security-policy.com/
CWE Id	693
WASC Id	15
Plugin Id	10038

Medium	Cross-Domain Misconfiguration
Description	Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server
URL	https://www.secureblink.com/about-us
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/blog
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/careers
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/contact-us
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/robots.txt
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/rss-feeds
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/solutions
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/terms-and-conditions
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/threat-research
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *

URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	
Attack	
Evidence	access-control-allow-origin: *
Instances	12
Solution	Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance). Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.
Reference	https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy
CWE Id	264
WASC Id	14
Plugin Id	10098

Medium	**Missing Anti-clickjacking Header**
Description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
URL	https://www.secureblink.com/
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/about-us
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/blog
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/careers
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/contact-us
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/rss-feeds
Method	GET

Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/solutions
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/threat-research
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	X-Frame-Options
Attack	
Evidence	
Instances	11
Solution	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app. If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	1021
WASC Id	15
Plugin Id	10020

Low	**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://www.secureblink.com/company-register
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-869139563 mt-4">

URL	https://www.secureblink.com/contact-us
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-3992399362 mt-12">
URL	https://www.secureblink.com/contact-us?company=ZAP&email=foo-bar%40example.com&job=ZAP&message&name=ZAP&phone=9999999999
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-3992399362 mt-12">
URL	https://www.secureblink.com/register
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-869139563 mt-4">
URL	https://www.secureblink.com/signin
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-981351468 mt-4">
URL	https://www.secureblink.com/signin?email_or_username=ZAP&password=ZAP
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2209493400 jsx-981351468 mt-4">
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-600599106 mt-12">
URL	https://www.secureblink.com/threat-spy?email=foo-bar%40example.com&message&name=ZAP&phone=9999999999
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-600599106 mt-12">
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2467518051 mt-8">
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2467518051 mt-8">
URL	https://www.secureblink.com/white-paper?email=foo-bar%40example.com&name=ZAP&phone=9999999999
Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2467518051 mt-8">
URL	https://www.secureblink.com/white-paper?email=foo-bar%40example.com&name=ZAP&phone=9999999999

Method	GET
Parameter	
Attack	
Evidence	<form class="jsx-2467518051 mt-8">
Instances	12
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.
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

Low	**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://www.secureblink.com/cyber-security-news/moncler-group-becomes-the-first-victim-of-alphv-
Method	GET
Parameter	
Attack	
Evidence	HTTP/1.1 500 Internal Server Error
URL	https://www.secureblink.com/cyber-security-news/sfile-
Method	GET
Parameter	
Attack	
Evidence	HTTP/1.1 500 Internal Server Error
URL	https://www.secureblink.com/threat-research/mozi-p2p-botnet-evolved-executed-new-capabilities-to-target-its-victims
Method	GET
Parameter	
Attack	
Evidence	HTTP/1.1 500 Internal Server Error
Instances	3
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
Plugin Id	90022

Low	**Cross-Domain JavaScript Source File Inclusion**
Description	The page includes one or more script files from a third-party domain.
URL	https://www.secureblink.com/
Method	GET
Parameter	https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js
Attack	
Evidence	<script data-ad-client="ca-pub-3214656650762790" async="" src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js" type="582e645760f3ad141a44b7fb-text/javascript"></script>
URL	https://www.secureblink.com/
Method	GET
Parameter	https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js
Attack	
Evidence	<script data-ad-client="ca-pub-3214656650762790" async="" src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
URL	https://www.secureblink.com/
Method	GET
Parameter	https://www.googletagmanager.com/gtag/js?id=UA-151054930-1
Attack	
Evidence	<script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-151054930-1" type="582e645760f3ad141a44b7fb-text/javascript"></script>
URL	https://www.secureblink.com/
Method	GET
Parameter	https://www.googletagmanager.com/gtag/js?id=UA-151054930-1
Attack	
Evidence	<script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-151054930-1"></script>
URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js
Attack	
Evidence	<script data-ad-client="ca-pub-3214656650762790" async="" src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js" type="0a75bc98daeadc6d2b0eb5ce-text/javascript"></script>
URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	https://www.googletagmanager.com/gtag/js?id=UA-151054930-1
Attack	
Evidence	<script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-151054930-1" type="0a75bc98daeadc6d2b0eb5ce-text/javascript"></script>
URL	https://www.secureblink.com/robots.txt
Method	GET
Parameter	https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js
Attack	
Evidence	<script data-ad-client="ca-pub-3214656650762790" async="" src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js" type="df2ec037b84378012e05c1d9-text/javascript"></script>
URL	https://www.secureblink.com/robots.txt
Method	GET
Parameter	https://www.googletagmanager.com/gtag/js?id=UA-151054930-1
Attack	
Evidence	<script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-151054930-1" type="df2ec037b84378012e05c1d9-text/javascript"></script>

URL	https://www.secureblink.com/solutions
Method	GET
Parameter	https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js
Attack	
Evidence	<script data-ad-client="ca-pub-3214656650762790" async="" src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js" type="24e3c35fd3ef5c4d6a6a40cd-text/javascript"></script>
URL	https://www.secureblink.com/solutions
Method	GET
Parameter	https://www.googletagmanager.com/gtag/js?id=UA-151054930-1
Attack	
Evidence	<script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-151054930-1" type="24e3c35fd3ef5c4d6a6a40cd-text/javascript"></script>
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js
Attack	
Evidence	<script data-ad-client="ca-pub-3214656650762790" async="" src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js" type="fa6aff380164ed621e9f4ac1-text/javascript"></script>
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	https://www.googletagmanager.com/gtag/js?id=UA-151054930-1
Attack	
Evidence	<script async="" src="https://www.googletagmanager.com/gtag/js?id=UA-151054930-1" type="fa6aff380164ed621e9f4ac1-text/javascript"></script>
Instances	12
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
Plugin Id	10017

Low	**Incomplete or No Cache-control Header Set**
Description	The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content.
URL	https://www.secureblink.com/about-us
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/blog
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/careers
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/contact-us
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate

URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/rss-feeds
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/sitemap.xml
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/solutions
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/terms-and-conditions
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/threat-research
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	Cache-Control
Attack	
Evidence	public, max-age=0, must-revalidate
Instances	12
Solution	Whenever possible ensure the cache-control HTTP header is set with no-cache, no-store, must-revalidate.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#web-content-caching https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control
CWE Id	525
WASC Id	13
Plugin Id	10015

Low	**Private IP Disclosure**
Description	A private IP (such as 10.x.x.x, 172.x.x.x, 192.168.x.x) or an Amazon EC2 private hostname (for example, ip-10-0-56-78) has been found in the HTTP response body. This information might be helpful for further attacks targeting internal systems.
URL	https://www.secureblink.com/cyber-security-news/vmware-vulnerability
Method	GET

Parameter	
Attack	
Evidence	10.0.0.2
URL	https://www.secureblink.com/rss-feeds/threat-research
Method	GET
Parameter	
Attack	
Evidence	10.64.100.51:8080
URL	https://www.secureblink.com/threat-research/shareit:-unpatched-vulnerability-targeting-to-remote-code-execution
Method	GET
Parameter	
Attack	
Evidence	10.64.100.51:8080
Instances	3
Solution	Remove the private IP address from the HTTP response body. For comments, use JSP/ASP/PHP comment instead of HTML/JavaScript comment which can be seen by client browsers.
Reference	https://tools.ietf.org/html/rfc1918
CWE Id	200
WASC Id	13
Plugin Id	2

Low	**Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)**
Description	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/cereberus-banking-virus
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/clop-ransomware
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/dorkbot-malware
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/ekans-ransomware
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/netwalker-ransomware
Method	GET
Parameter	
Attack	

Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/taidoor-malware
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/the-iron-liberty-group
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/user-datagram-protocol
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/wastedlocker-and-evil-corp
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
URL	https://www.secureblink.com/threat-research/zloader-malware
Method	GET
Parameter	
Attack	
Evidence	x-powered-by: Next.js
Instances	11
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.
Reference	http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx http://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html
CWE Id	200
WASC Id	13
Plugin Id	10037

Low	**Strict-Transport-Security Header Not Set**
Description	HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.
URL	https://www.secureblink.com/cdn-cgi/l/email-protection
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/cdn-cgi/styles/cf.errors.css
Method	GET
Parameter	
Attack	
Evidence	
URL	https://www.secureblink.com/cdn-cgi/styles/cf.errors.ie.css
Method	GET
Parameter	
Attack	
Evidence	
Instances	3

Solution	Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html https://owasp.org/www-community/Security_Headers http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security http://caniuse.com/stricttransportsecurity http://tools.ietf.org/html/rfc6797
CWE Id	319
WASC Id	15
Plugin Id	10035

Low	**Timestamp Disclosure - Unix**
Description	A timestamp was disclosed by the application/web server - Unix
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	1042565032
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	1108566402
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	1143614438
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	1264948985
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	1420229099
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	1490040372
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	151054930
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	1689842603
URL	https://www.secureblink.com/
Method	GET
Parameter	

Attack	
Evidence	1703666734
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	1970145647
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	414251047
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	502299912
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	68498547
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	770682051
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	864116505
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	900867076
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	978765007
Instances	17
Solution	Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.
Reference	http://projects.webappsec.org/w/page/13246936/Information%20Leakage
CWE Id	200
WASC Id	13
Plugin Id	10096

Informational	**Content Security Policy (CSP) Report-Only Header Found**
Description	The response contained a Content-Security-Policy-Report-Only header, this may indicate a work-in-progress implementation, or an oversight in promoting pre-Prod to Prod, etc. Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft

	to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	https://www.secureblink.com/about-us
Method	GET
Parameter	
Attack	
Evidence	
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header, to achieve optimal browser support: "Content-Security-Policy" for Chrome 25+, Firefox 23+ and Safari 7+, "X-Content-Security-Policy" for Firefox 4.0+ and Internet Explorer 10+, and "X-WebKit-CSP" for Chrome 14+ and Safari 6+.
Reference	https://www.w3.org/TR/CSP2/ https://w3c.github.io/webappsec-csp/ http://caniuse.com/#feat=contentsecuritypolicy http://content-security-policy.com/
CWE Id	693
WASC Id	15
Plugin Id	10038

Informational	**Information Disclosure - Sensitive Information in URL**
Description	The request appeared to contain sensitive information leaked in the URL. This can violate PCI and most organizational compliance policies. You can configure the list of strings for this check to add or remove values specific to your environment.
URL	https://www.secureblink.com/company-register?companyName=ZAP&email=foo-bar%40example.com&name=ZAP&password=ZAP
Method	GET
Parameter	email
Attack	
Evidence	foo-bar@example.com
URL	https://www.secureblink.com/company-register?companyName=ZAP&email=foo-bar%40example.com&name=ZAP&password=ZAP
Method	GET
Parameter	password
Attack	
Evidence	password
URL	https://www.secureblink.com/contact-us?company=ZAP&email=foo-bar%40example.com&job=ZAP&message&name=ZAP&phone=9999999999
Method	GET
Parameter	email
Attack	
Evidence	foo-bar@example.com
URL	https://www.secureblink.com/register?email=foo-bar%40example.com&name=ZAP&password=ZAP&username=ZAP
Method	GET
Parameter	email
Attack	
Evidence	foo-bar@example.com
URL	https://www.secureblink.com/register?email=foo-bar%40example.com&name=ZAP&password=ZAP&username=ZAP
Method	GET
Parameter	password
Attack	
Evidence	password
URL	https://www.secureblink.com/register?email=foo-bar%40example.com&name=ZAP&password=ZAP&username=ZAP
Method	GET
Parameter	username
Attack	

Evidence	username
URL	https://www.secureblink.com/signin?email_or_username=ZAP&password=ZAP
Method	GET
Parameter	email_or_username
Attack	
Evidence	email_or_username
URL	https://www.secureblink.com/signin?email_or_username=ZAP&password=ZAP
Method	GET
Parameter	password
Attack	
Evidence	password
URL	https://www.secureblink.com/threat-spy?email=foo-bar%40example.com&message&name=ZAP&phone=9999999999
Method	GET
Parameter	email
Attack	
Evidence	foo-bar@example.com
URL	https://www.secureblink.com/white-paper?email=foo-bar%40example.com&name=ZAP&phone=9999999999
Method	GET
Parameter	email
Attack	
Evidence	foo-bar@example.com
Instances	10
Solution	Do not pass sensitive information in URIs.
Reference	
CWE Id	200
WASC Id	13
Plugin Id	10024

Informational	**Information Disclosure - Suspicious Comments**
Description	The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/about-us
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/blog
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/careers
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/contact-us
Method	GET

Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/robots.txt
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/solutions
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/threat-research
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	
Attack	
Evidence	query
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	
Attack	
Evidence	query
Instances	11
Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.
Reference	
CWE Id	200
WASC Id	13
Plugin Id	10027

Informational	**Modern Web Application**
Description	The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	Product
URL	https://www.secureblink.com/about-us
Method	GET
Parameter	
Attack	
Evidence	Product

URL	https://www.secureblink.com/blog
Method	GET
Parameter	
Attack	
Evidence	 <h2 class="jsx-3802875118">Automated Vulnerability Interception</h2>
URL	https://www.secureblink.com/careers
Method	GET
Parameter	
Attack	
Evidence	Product
URL	https://www.secureblink.com/contact-us
Method	GET
Parameter	
Attack	
Evidence	Product
URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	
Attack	
Evidence	 <h2 class="jsx-3802875118">Automated Vulnerability Interception</h2>
URL	https://www.secureblink.com/robots.txt
Method	GET
Parameter	
Attack	
Evidence	<noscript></noscript>
URL	https://www.secureblink.com/solutions
Method	GET
Parameter	
Attack	
Evidence	Product
URL	https://www.secureblink.com/threat-research
Method	GET
Parameter	
Attack	
Evidence	 <h2 class="jsx-3802875118">Automated Vulnerability Interception</h2>
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	
Attack	
Evidence	Product
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	
Attack	
Evidence	 <h2 class="jsx-3802875118">Automated Vulnerability Interception</h2>

Instances	11
Solution	This is an informational alert and so no changes are required.
Reference	
CWE Id	
WASC Id	
Plugin Id	10109

Informational	**Retrieved from Cache**
Description	The content was retrieved from a shared cache. If the response data is sensitive, personal or user-specific, this may result in sensitive information being leaked. In some cases, this may even result in a user gaining complete control of the session of another user, depending on the configuration of the caching components in use in their environment. This is primarily an issue where caching servers such as "proxy" caches are configured on the local network. This configuration is typically found in corporate or educational environments, for instance.
URL	https://www.secureblink.com/
Method	GET
Parameter	
Attack	
Evidence	Age: 0
URL	https://www.secureblink.com/about-us
Method	GET
Parameter	
Attack	
Evidence	Age: 355413
URL	https://www.secureblink.com/blog
Method	GET
Parameter	
Attack	
Evidence	Age: 161024
URL	https://www.secureblink.com/careers
Method	GET
Parameter	
Attack	
Evidence	Age: 426294
URL	https://www.secureblink.com/contact-us
Method	GET
Parameter	
Attack	
Evidence	Age: 161023
URL	https://www.secureblink.com/cyber-security-news
Method	GET
Parameter	
Attack	
Evidence	Age: 332793
URL	https://www.secureblink.com/sitemap.xml
Method	GET
Parameter	
Attack	
Evidence	Age: 0
URL	https://www.secureblink.com/solutions
Method	GET
Parameter	
Attack	
Evidence	Age: 161024
URL	https://www.secureblink.com/threat-research
Method	GET

Parameter	
Attack	
Evidence	Age: 161022
URL	https://www.secureblink.com/threat-spy
Method	GET
Parameter	
Attack	
Evidence	Age: 161024
URL	https://www.secureblink.com/white-paper
Method	GET
Parameter	
Attack	
Evidence	Age: 161024
Instances	11
Solution	Validate that the response does not contain sensitive, personal or user-specific information. If it does, consider the use of the following HTTP response headers, to limit, or prevent the content being stored and retrieved from the cache by another user: Cache-Control: no-cache, no-store, must-revalidate, private Pragma: no-cache Expires: 0 This configuration directs both HTTP 1.0 and HTTP 1.1 compliant caching servers to not store the response, and to not retrieve the response (without validation) from the cache, in response to a similar request.
Reference	https://tools.ietf.org/html/rfc7234 https://tools.ietf.org/html/rfc7231 http://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html (obsoleted by rfc7234)
CWE Id	
WASC Id	
Plugin Id	10050