ZAP Scanning Report

Site: https://192.168.0.2

Generated on Fri, 19 Apr 2024 22:18:54

ZAP Version: 2.14.0

ZAP is supported by the Crash Override Open Source Fellowship

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	5
Low	6
Informational	3

Alerts

Name	Risk Level	Number of Instances
Path Traversal	High	1
Absence of Anti-CSRF Tokens	Medium	2
CSP: Wildcard Directive	Medium	1
CSP: script-src unsafe-inline	Medium	1
CSP: style-src unsafe-inline	Medium	1
HTTPS to HTTP Insecure Transition in Form Post	Medium	2
Cookie Without Secure Flag	Low	3
Cookie with SameSite Attribute None	Low	1
Cookie without SameSite Attribute	Low	3
Private IP Disclosure	Low	1
Secure Pages Include Mixed Content	Low	2
Strict-Transport-Security Header Not Set	Low	5
Re-examine Cache-control Directives	Informational	1
Session Management Response Identified	Informational	6
User Agent Fuzzer	Informational	60

Alert Detail

High	Path Traversal
	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.

Description	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://192.168.0.2/auth/saml/login?nextUrl=%2Flogin
Method	GET
Attack	/login
Evidence	
Other Info	
Instances	1
Solution	Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list 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 deny list). However, deny lists 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 allow lists 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 an allow list 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 allow list 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	https://owasp.org/www-community/attacks/Path_Traversal https://cwe.mitre.org/data/definitions/22.html
CWE Id	22
WASC Id	33
Plugin Id	6

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://192.168.0.2
Method	GET
Attack	
Evidence	<form id="kc-form-login" onsubmit="login.disabled = true; return true;" action="http://192.168.0.7:8080/realms/test/login-actions/authenticate?session_code=6fZC2GCPyw5_ClxsQpUItW5nZeTdYgh6Jv7Fya8DJig&execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem&tab_id=wM9EbTDDoXw" method="post">
	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken,

Other Info	csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "id-hidden-input" "kc-login" "password" "username"].
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	<form id="kc-form-login" onsubmit="login.disabled = true; return true;" action="http://192.168.0.7:8080/realms/test/login-actions/authenticate?session_code=771fCjEbHQ-7QIPFYcSBLDj8WyNRH1X2kZmFnGdJI7g&execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem&tab_id=pvlUNxwYFcU" method="post">
Other Info	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "id-hidden-input" "kc-login" "password" "username"].
Instances	2
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	https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html https://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

Medium	**CSP: Wildcard Directive**
	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) Cross Site Scripting (XSS), and data

Description	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://192.168.0.2
Method	GET
Attack	
Evidence	frame-src 'self'; frame-ancestors 'self'; object-src 'none';
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: script-src, style-src, img-src, connect-src, font-src, media-src, manifest-src, worker-src, form-action The directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	https://www.w3.org/TR/CSP/ https://caniuse.com/#search=content+security+policy https://content-security-policy.com/ https://github.com/HtmlUnit/htmlunit-csp https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Medium	**CSP: script-src unsafe-inline**
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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://192.168.0.2
Method	GET
Attack	
Evidence	frame-src 'self'; frame-ancestors 'self'; object-src 'none';
Other Info	script-src includes unsafe-inline.
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	https://www.w3.org/TR/CSP/ https://caniuse.com/#search=content+security+policy https://content-security-policy.com/ https://github.com/HtmlUnit/htmlunit-csp https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Medium	CSP: style-src unsafe-inline
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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://192.168.0.2
Method	GET
Attack	
Evidence	frame-src 'self'; frame-ancestors 'self'; object-src 'none';
Other Info	style-src includes unsafe-inline.
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	https://www.w3.org/TR/CSP/ https://caniuse.com/#search=content+security+policy https://content-security-policy.com/ https://github.com/HtmlUnit/htmlunit-csp https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Medium	HTTPS to HTTP Insecure Transition in Form Post
Description	This check identifies secure HTTPS pages that host insecure HTTP forms. The issue is that a secure page is transitioning to an insecure page when data is uploaded through a form. The user may think they're submitting data to a secure page when in fact they are not.
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	http://192.168.0.7:8080/realms/test/login-actions/authenticate?session_code=6fZC2GCPyw5_ClxsQpUItW5nZeTdYgh6Jv7Fya8DJig&execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem&tab_id=wM9EbTDDoXw
Other Info	The response to the following request over HTTPS included an HTTP form tag action attribute value: https://192.168.0.2 The context was: <form id="kc-form-login" onsubmit=" login.disabled = true; return true;" action="http://192.168.0.7:8080/realms/test/login-actions /authenticate?session_code=6fZC2GCPyw5_ClxsQpUItW5nZeTdYgh6Jv7Fya8DJig& execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem& tab_id=wM9EbTDDoXw" method="post"> <label for="username" class="pf-c-form__label pf-c-form__label-text">Username or email</label> <input tabindex=" 1" id="username" class="pf-c-form-control" name="username" value="" type="text" autofocus autocomplete="off" aria-invalid="" /> <label for=" password" class="pf-c-form__label pf-c-form__label-text">Password</label> <input tabindex="2" id="password" class="pf-c-form-control" name="password" type="password" autocomplete="off" aria-invalid="" /> <input type="hidden" id="id-hidden-input" name="credentialId" /> <input tabindex="4" class="pf-c-button pf-m-primary pf-m-block btn-lg" name="login" id="kc-login" type="submit" value="Sign In"/> </form>
URL	https://192.168.0.2
Method	GET

Attack	
Evidence	http://192.168.0.7:8080/realms/test/login-actions/authenticate?session_code=771fCjEbHQ-7QIPFYcSBLDj8WyNRH1X2kZmFnGdJI7g&execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem&tab_id=pvlUNxwYFcU
Other Info	The response to the following request over HTTPS included an HTTP form tag action attribute value: https://192.168.0.2 The context was: <form id="kc-form-login" onsubmit=" login.disabled = true; return true;" action="http://192.168.0.7:8080/realms/test/login-actions /authenticate?session_code=771fCjEbHQ-7QIPFYcSBLDj8WyNRH1X2kZmFnGdJI7g& execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem& tab_id=pvlUNxwYFcU" method="post"> <label for="username" class="pf-c-form__label pf-c-form__label-text">Username or email</label> <input tabindex=" 1" id="username" class="pf-c-form-control" name="username" value="" type="text" autofocus autocomplete="off" aria-invalid="" /> <label for=" password" class="pf-c-form__label pf-c-form__label-text">Password</label> <input tabindex="2" id="password" class="pf-c-form-control" name="password" type="password" autocomplete="off" aria-invalid="" /> <input type="hidden" id="id-hidden-input" name="credentialId" /> <input tabindex="4" class="pf-c-button pf-m-primary pf-m-block btn-lg" name="login" id="kc-login" type="submit" value="Sign In"/> </form>
Instances	2
Solution	Ensure sensitive data is only sent over secured HTTPS channels.
Reference	
CWE Id	319
WASC Id	15
Plugin Id	10042

Low	**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://192.168.0.2
Method	GET
Attack	
Evidence	Set-Cookie: AUTH_SESSION_ID_LEGACY
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	Set-Cookie: KC_RESTART
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	
Evidence	set-cookie: security_authentication
Other Info	
Instances	3

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	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
CWE Id	614
WASC Id	13
Plugin Id	10011

Low	**Cookie with SameSite Attribute None**
Description	A cookie has been set with its SameSite attribute set to "none", which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	Set-Cookie: AUTH_SESSION_ID
Other Info	
Instances	1
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	1275
WASC Id	13
Plugin Id	10054

Low	**Cookie without SameSite Attribute**
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	Set-Cookie: AUTH_SESSION_ID_LEGACY
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	Set-Cookie: KC_RESTART
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	
Evidence	set-cookie: security_authentication

Other Info	
Instances	3
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	1275
WASC Id	13
Plugin Id	10054

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://192.168.0.2
Method	GET
Attack	
Evidence	192.168.0.7:8080
Other Info	192.168.0.7:8080
Instances	1
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	**Secure Pages Include Mixed Content**
Description	The page includes mixed content, that is content accessed via HTTP instead of HTTPS.
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	http://192.168.0.7:8080/realms/test/login-actions/authenticate?session_code=6fZC2GCPyw5_ClxsQpUItW5nZeTdYgh6Jv7Fya8DJig&execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem&tab_id=wM9EbTDDoXw
Other Info	tag=form action=http://192.168.0.7:8080/realms/test/login-actions/authenticate?session_code=6fZC2GCPyw5_ClxsQpUItW5nZeTdYgh6Jv7Fya8DJig&execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem&tab_id=wM9EbTDDoXw
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	http://192.168.0.7:8080/realms/test/login-actions/authenticate?session_code=771fCjEbHQ-7QIPFYcSBLDj8WyNRH1X2kZmFnGdJI7g&execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem&tab_id=pvlUNxwYFcU
Other Info	tag=form action=http://192.168.0.7:8080/realms/test/login-actions/authenticate?session_code=771fCjEbHQ-7QIPFYcSBLDj8WyNRH1X2kZmFnGdJI7g&execution=5e15b64d-fd26-4734-87d3-a5d932069132&client_id=siem&tab_id=pvlUNxwYFcU
Instances	2

Solution	A page that is available over SSL/TLS must be comprised completely of content which is transmitted over SSL/TLS. The page must not contain any content that is transmitted over unencrypted HTTP. This includes content from third party sites.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html
CWE Id	311
WASC Id	4
Plugin Id	10040

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://192.168.0.2/auth
Method	GET
Attack	
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml
Method	GET
Attack	
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	
Evidence	
Other Info	

Instances	5
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 https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security https://caniuse.com/stricttransportsecurity https://datatracker.ietf.org/doc/html/rfc6797
CWE Id	319
WASC Id	15
Plugin Id	10035

Informational	**Re-examine Cache-control Directives**
Description	The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	no-store, must-revalidate, max-age=0
Other Info	
Instances	1
Solution	For secure content, ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".
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 https://grayduck.mn/2021/09/13/cache-control-recommendations/
CWE Id	525
WASC Id	13
Plugin Id	10015

Informational	**Session Management Response Identified**
Description	The given response has been identified as containing a session management token. The 'Other Management Method set to "Auto-Detect" then this rule will change the session management to
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	81e0e855-a126-4f08-923c-ddaa3642a63b
Other Info	cookie:AUTH_SESSION_ID_LEGACY cookie:KC_RESTART cookie:AUTH_SESSION_ID
URL	https://192.168.0.2
Method	GET
Attack	
Evidence	969b1f69-d5a8-42c2-992a-bec9c9d95f35
Other	

Info	cookie:AUTH_SESSION_ID_LEGACY cookie:KC_RESTART cookie:AUTH_SESSION_ID
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	
Evidence	Fe26.2**3a70eb1cf04cd3114cd8449ead492f704129519f4b2fc9af95dd8b93a6fa075d*Me1qYK9 VQrxbUf1fEm8BCi2mCTIVl8wb2qsRI7zrK7G9Ar25lYW_7dLT**c3569d322092ec563a3f0eec05
Other Info	cookie:security_authentication
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	
Evidence	Fe26.2**7105d9ae51e03c55c260d0ad7217d1b5e5eb668bea6a12b8ed58b45541827ac5*Gtv42 _0DJYH0seoO_FmQTAuZPfIbuu1A9MbZGQ**f232cd67d3662aa121b7fd6d56efd5da24b699cd
Other Info	cookie:security_authentication
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	
Evidence	Fe26.2**7f5f2eee7856d61ff5070e0632b7faec2f1999c6d7f3a30824521b46fc5b78b8*n8K5erX25 ghXXWufxJmc3j26pW9AMoh5Q7fZHpJa4RmRLtFYIhDeYVcijjFekmTmqMuzeyfZ9CRUZtbBy7 UMKQbpBuctzgep0U
Other Info	cookie:security_authentication
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	
Evidence	Fe26.2**aa04c58af5d4ef7da719bbd357fdb28a703ea3bae8f692cd57960895b147be99*MYgVIJ vwjhkwhQWAOm0mJQRAFF2m34lqFO5DEpuC7fwNiftqcSLyb**29d91161e86f413684bf64e8d3
Other Info	cookie:security_authentication
Instances	6
Solution	This is an informational alert rather than a vulnerability and so there is nothing to fix.
Reference	https://www.zaproxy.org/docs/desktop/addons/authentication-helper/session-mgmt-id
CWE Id	
WASC Id	
Plugin Id	10112

Informational	**User Agent Fuzzer**
Description	Check for differences in response based on fuzzed User Agent (eg. mobile sites, access as a Search Engine Crawler). Compares the response statuscode and the hashcode of the response body with the original response.
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Evidence	
Other Info	

URL	https://192.168.0.2
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Trident/7.0; rv:11.0) like Gecko
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3739.0 Safari/537.36 Edg/75.0.109.0
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:93.0) Gecko/20100101 Firefox/91.0
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Evidence	
Other Info	
URL	https://192.168.0.2

Method	GET
Attack	Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12A366 Safari/600.1.4
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341 Safari/528.16
Evidence	
Other Info	
URL	https://192.168.0.2
Method	GET
Attack	msnbot/1.1 (+http://search.msn.com/msnbot.htm)
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)
Evidence	
Other Info	
URL	https://192.168.0.2/

Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Trident/7.0; rv:11.0) like Gecko
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3739.0 Safari/537.36 Edg/75.0.109.0
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:93.0) Gecko/20100101 Firefox/91.0
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12A366 Safari/600.1.4
Evidence	
Other Info	
URL	https://192.168.0.2/

Method	GET
Attack	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341 Safari/528.16
Evidence	
Other Info	
URL	https://192.168.0.2/
Method	GET
Attack	msnbot/1.1 (+http://search.msn.com/msnbot.htm)
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Trident/7.0; rv:11.0) like Gecko
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3739.0 Safari/537.36 Edg/75.0.109.0
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET

Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:93.0) Gecko/20100101 Firefox/91.0
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12A366 Safari/600.1.4
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341 Safari/528.16
Evidence	
Other Info	
URL	https://192.168.0.2/auth/saml/login?nextUrl=%2F
Method	GET
Attack	msnbot/1.1 (+http://search.msn.com/msnbot.htm)
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET

Attack	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Trident/7.0; rv:11.0) like Gecko
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3739.0 Safari/537.36 Edg/75.0.109.0
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:93.0) Gecko/20100101 Firefox/91.0
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)

Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12A366 Safari/600.1.4
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341 Safari/528.16
Evidence	
Other Info	
URL	https://192.168.0.2/robots.txt
Method	GET
Attack	msnbot/1.1 (+http://search.msn.com/msnbot.htm)
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)

Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Trident/7.0; rv:11.0) like Gecko
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3739.0 Safari/537.36 Edg/75.0.109.0
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:93.0) Gecko/20100101 Firefox/91.0
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12A366 Safari/600.1.4

Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341 Safari/528.16
Evidence	
Other Info	
URL	https://192.168.0.2/sitemap.xml
Method	GET
Attack	msnbot/1.1 (+http://search.msn.com/msnbot.htm)
Evidence	
Other Info	
Instances	60
Solution	
Reference	https://owasp.org/wstg
CWE Id	
WASC Id	
Plugin Id	10104