Assignment 1 - Report on Broken Access Control

Site: http://127.0.0.1:9090

Generated on Mon, 9 Oct 2023 12:56:55

ZAP Version: 2.13.0

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	3
Low	2
Informational	6

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	11
Content Security Policy (CSP) Header Not Set	Medium	8
Vulnerable JS Library	Medium	1
Cookie No HttpOnly Flag	Low	4
Cookie without SameSite Attribute	Low	4
Authentication Request Identified	Informational	1
Information Disclosure - Sensitive Information in URL	Informational	2
Information Disclosure - Suspicious Comments	Informational	1
Modern Web Application	Informational	8
Session Management Response Identified	Informational	7
User Controllable HTML Element Attribute (Potential XSS)	Informational	2

Alert Detail

Medium	Absence of Anti-CSRF Tokens ✓
CSRF token should include in form submission just like id, name suggested below	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.

Description	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	http://127.0.0.1:9090/jwt
Method	GET
Attack	
Evidence	<form id="decodeForm">
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: ""].
URL	http://127.0.0.1:9090/jwt
Method	GET
Attack	
Evidence	<form id="encodeForm">
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 2: ""].
URL	http://127.0.0.1:9090/jwt?header&payload
Method	GET
Attack	
Evidence	<form id="decodeForm">
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: ""].
URL	http://127.0.0.1:9090/jwt?header&payload
Method	GET
Attack	
Evidence	<form id="encodeForm">
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 2: ""].
URL	http://127.0.0.1:9090/jwt?header&payload&token
Method	GET
Attack	
Evidence	<form id="decodeForm">
Other	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token,

Info	_csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: ""].
URL	http://127.0.0.1:9090/jwt?header&payload&token
Method	GET
Attack	
Evidence	<form id="encodeForm">
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 2: ""].
URL	http://127.0.0.1:9090/jwt?token
Method	GET
Attack	
Evidence	<form id="decodeForm">
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: ""].
URL	http://127.0.0.1:9090/jwt?token
Method	GET
Attack	
Evidence	<form id="encodeForm">
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 2: ""].
URL	http://127.0.0.1:9090/login
Method	GET
Attack	
Evidence	<form action="/login" 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: "password" "username"].
URL	http://127.0.0.1:9090/login?error=true
Method	GET
Attack	
Evidence	<form action="/login" 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: "password" "username"].
URL	http://127.0.0.1:9090/login?logout
Method	GET
Attack	
Evidence	<form action="/login" 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: "password" "username"].
Instances	11
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

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	http://127.0.0.1:9090/home
Method	GET
Attack	
Evidence	
Other Info	
URL	http://127.0.0.1:9090/jwt

NO CSP policy has been found like

"Content-Security-Policy: default-src 'self';
script-src 'self' https://example.com;
style-src 'self' https://example.com;
img-src 'self' https://example.com;"

In this situation, any images, content, script can be accessed from anywhere which is vulnerable .

Method	GET
Attack	
Evidence	
Other Info	
URL	http://127.0.0.1:9090/jwt?header&payload
Method	GET
Attack	
Evidence	
Other Info	
URL	http://127.0.0.1:9090/jwt?header&payload&token
Method	GET
Attack	
Evidence	
Other Info	
URL	http://127.0.0.1:9090/jwt?token
Method	GET
Attack	
Evidence	
Other Info	
URL	http://127.0.0.1:9090/login
Method	GET
Attack	
Evidence	
Other Info	
URL	http://127.0.0.1:9090/login?error=true
Method	GET
Attack	
Evidence	
Other Info	
URL	http://127.0.0.1:9090/login?logout
Method	GET
Attack	
Evidence	
Other Info	
Instances	8
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

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	**Vulnerable JS Library**
Description	The identified library bootstrap, version 3.3.7 is vulnerable.
URL	http://127.0.0.1:9090/webjars/bootstrap/3.3.7/js/bootstrap.min.js
Method	GET
Attack	
Evidence	/3.3.7/js/bootstrap.min.js
Other Info	CVE-2019-8331 CVE-2018-14041 CVE-2018-20677 CVE-2018-20676 CVE-2018-14042 CVE-2016-10735
Instances	1
Solution	Please upgrade to the latest version of bootstrap.
Reference	https://github.com/twbs/bootstrap/issues/28236 https://github.com/twbs/bootstrap/issues/20184 https://github.com/advisories/GHSA-ph58-4vrj-w6hr https://github.com/twbs/bootstrap/issues/20631 https://github.com/advisories/GHSA-4p24-vmcr-4gqj https://nvd.nist.gov/vuln/detail/CVE-2018-20676
CWE Id	829
WASC Id	
Plugin Id	10003

Low	**Cookie No HttpOnly Flag** ✓
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://127.0.0.1:9090/files WEBWOLFSESSION=dXO4LogWKqHyR6ZbBjug1nYDnQP6eC38vivgeclr
Method	GET
Attack	
Evidence	Set-Cookie: WEBWOLFSESSION
Other Info	
URL	http://127.0.0.1:9090/mail WEBWOLFSESSION=nS-gBZda4Bl3WD809TIR3TaMyZduEGid8Q209Pur
Method	GET
Attack	
Evidence	Set-Cookie: WEBWOLFSESSION
Other Info	

URL	http://127.0.0.1:9090/requests
Method	GET
Attack	
Evidence	Set-Cookie: WEBWOLFSESSION
Other Info	
URL	http://127.0.0.1:9090/login
Method	POST WEBWOLFSESSION=T1VP8yjcfPnlrq48SR4CUQklRM5tgFXuvx1Y_1SH
Attack	
Evidence	Set-Cookie: WEBWOLFSESSION
Other Info	
Instances	4
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly
CWE Id	[1004](1004)
WASC Id	13
Plugin Id	[10010](10010)

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	http://127.0.0.1:9090/files
Method	GET
Attack	
Evidence	Set-Cookie: WEBWOLFSESSION
Other Info	
URL	http://127.0.0.1:9090/mail
Method	GET
Attack	
Evidence	Set-Cookie: WEBWOLFSESSION
Other Info	
URL	http://127.0.0.1:9090/requests
Method	GET
Attack	
Evidence	Set-Cookie: WEBWOLFSESSION
Other Info	
URL	http://127.0.0.1:9090/login
Method	POST
Attack	
Evidence	Set-Cookie: WEBWOLFSESSION

Other Info	
Instances	4
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

Informational	Authentication Request Identified ✓
Description	The given request has been identified as an authentication request. The 'Other Info' field contains a set of key=value lines which identify any relevant fields. If the request is in a context which has an Authentication Method set to "Auto-Detect" then this rule will change the authentication to match the request identified.
URL	http://127.0.0.1:9090/login
Method	POST
Attack	
Evidence	password username=ZAP&password=ZAP
Other Info	userParam=username userValue=ZAP passwordParam=password referer=http://127.0.0.1:9090/login
Instances	1
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/auth-req-id/
CWE Id	
WASC Id	
Plugin Id	10111

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	http://127.0.0.1:9090/jwt?header&payload&token
Method	GET
Attack	
Evidence	token
Other Info	The URL contains potentially sensitive information. The following string was found via the pattern: token token
URL	http://127.0.0.1:9090/jwt?token
Method	GET
Attack	
Evidence	token
Other Info	The URL contains potentially sensitive information. The following string was found via the pattern: token token
Instances	2
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	http://127.0.0.1:9090/webjars/jquery/3.5.1/jquery.min.js
Method	GET
Attack	
Evidence	username
Other Info	The following pattern was used: \bUSERNAME\b and was detected in the element starting with: "!function(e,t){"use strict";"object"==typeof module&&"object"==typeof module.exports? module.exports=e.document?t(e,!0):function(", see evidence field for the suspicious comment/snippet.
Instances	1
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	http://127.0.0.1:9090/home
Method	GET
Attack	
Evidence	<a href="#"> </a>
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
URL	http://127.0.0.1:9090/jwt
Method	GET
Attack	
Evidence	<a href="#"> </a>
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
URL	http://127.0.0.1:9090/jwt?header&payload
Method	GET
Attack	
Evidence	<a href="#"> </a>
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
URL	http://127.0.0.1:9090/jwt?header&payload&token
Method	GET
Attack	

```
/*! jQuery v3.5.1 | (c) JS
Foundation and other
contributors |
jquery.org/license */
!function(e,t){"use strict";"
object"==typeof module&&"
object"==typeof
module.exports?
module.exports=e.document?t
(e,!0):function(e){if(!
e.document)throw new Error("
jQuery requires a window with
a document");return t(e)}:t(e)}
("undefined"!=typeof window?
window:this,function(C,e){"
use strict";var t=
[],r=Object.getPrototypeOf,s=t.
slice,g=t.flat?function(e)
{return t.flat.call(e)}:function(e)
{return t.concat.apply
([],e)},u=t.push,i=t.indexOf,n=
{},o=n.toString,v=n.hasOwnPr
operty,a=v.toString,l=a.call
(Object),y={},m=function(e)
{return"function"==typeof e&&"
number"!=typeof
e.nodeType},x=function(e)
{return null!
=e&&e===e.window},E=C.doc
ument,c={type:!0,src:!
0,nonce:!0,noModule:!0};
function b(e,t,n){var r,i,o=
(n=n||E).createElement("
script");if(o.text=e,t)for(r in c)
(i=t[r]
||t.getAttribute&&t.getAttribute
(r))&&o.setAttribute(r,i);
n.head.appendChild
(o).parentNode.removeChild
(o)}function w(e){return
null==e?e+"":"object"==typeof
e||"function"==typeof e?n
[o.call(e)]||"object":typeof e}
var f="3.5.1",S=function(e,t)
{return new S.fn.init(e,t)};
function p(e){var t=!!e&&"
length"in e&&e.length,n=w(e);
return!m(e)&&!x(e)&&("array"
===n||0===t||"number"
==typeof t&&0<t&&t-1 in e)}
S.fn=S.prototype={jquery:
f,constructor:S,length:
0,toArray:function(){return
s.call(this)},get:function(e)
{return null==e?s.call(this):
e<0?this[e+this.length]:this
[e]},pushStack:function(e){var
t=S.merge(this.constructor
(),e);return
t.prevObject=this,t},each:
function(e){return S.each
(this,e)},map:function(n)
{return this.pushStack(S.map
(this,function(e,t){return n.call
(e,t,e)}))},slice:function()
{return this.pushStack(s.apply
(this,arguments))},first:function
(){return this.eq(0)},last:
function(){return this.eq
(-1)},even:function(){return
this.pushStack(S.grep
(this,function(e,t){return(t+1)
%2}))},odd:function(){return
this.pushStack(S.grep
(this,function(e,t){return
t%2}))},eq:function(e){var
t=this.length,n=+e+(e<0?t:0);
return this.pushStack
(0<=n&&n<t?[this[n]]:[])},end:
function(){return
this.prevObject||this.constructo
r()},push:u,sort:t.sort,splice:
t.splice},S.extend=S.fn.extend
=function(){var
e,t,n,r,i,o,a=arguments[0]||
{},s=1,u=arguments.length,l=!
1;for("boolean"==typeof a&&
(l=a,a=arguments[s]||{},s++),"
object"==typeof a||m(a)||(a=
{}),s===u&&(a=this,s--);s<u;
s++)if(null!=(e=arguments[s]))
for(t in e)r=e[t],"__proto__"!
==t&&a!==r&&(l&&r&&
(S.isPlainObject(r)||
(i=Array.isArray(r)))?(n=a
[t],o=i&&!Array.isArray(n)?[]:
i||S.isPlainObject(n)?n:{},i=!1,a
[t]=S.extend(l,o,r)):void 0!
==r&&(a[t]=r));return
a},S.extend({expando:"jQuery"
+(f+Math.random()).replace
(/\D/g,""),isReady:!0,error:
function(e){throw new Error
```

Evidence	<a href="#"> </a>
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
URL	http://127.0.0.1:9090/jwt?token
Method	GET
Attack	
Evidence	<a href="#"> </a>
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
URL	http://127.0.0.1:9090/login
Method	GET
Attack	
Evidence	<a href="#"> </a>
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
URL	http://127.0.0.1:9090/login?error=true
Method	GET
Attack	
Evidence	<a href="#"> </a>
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
URL	http://127.0.0.1:9090/login?logout
Method	GET
Attack	
Evidence	<a href="#"> </a>
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
Instances	8
Solution	This is an informational alert and so no changes are required.
Reference	
CWE Id	
WASC Id	
Plugin Id	10109

Informational	**Session Management Response Identified**
Description	The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.
URL	http://127.0.0.1:9090/files
Method	GET
Attack	
Evidence	dXO4LogWKqHyR6ZbBjug1nYDnQP6eC38vivgeclr
Other Info	cookie:WEBWOLFSESSION

URL	http://127.0.0.1:9090/logout
Method	GET
Attack	
Evidence	RNEShTs3OQKJDBt0pMDrxEawZ_9UYHbaHOrOXnYE
Other Info	cookie:WEBWOLFSESSION
URL	http://127.0.0.1:9090/mail
Method	GET
Attack	
Evidence	nS-gBZda4Bl3WD809TIR3TaMyZduEGid8Q209Pur
Other Info	cookie:WEBWOLFSESSION
URL	http://127.0.0.1:9090/requests
Method	GET
Attack	
Evidence	RNEShTs3OQKJDBt0pMDrxEawZ_9UYHbaHOrOXnYE
Other Info	cookie:WEBWOLFSESSION
URL	http://127.0.0.1:9090/login
Method	POST
Attack	
Evidence	T1VP8yjcfPnlrq48SR4CUQklRM5tgFXuvx1Y_1SH
Other Info	cookie:WEBWOLFSESSION
URL	http://127.0.0.1:9090/logout
Method	GET
Attack	
Evidence	RNEShTs3OQKJDBt0pMDrxEawZ_9UYHbaHOrOXnYE
Other Info	cookie:WEBWOLFSESSION
URL	http://127.0.0.1:9090/login
Method	POST
Attack	
Evidence	T1VP8yjcfPnlrq48SR4CUQklRM5tgFXuvx1Y_1SH
Other Info	cookie:WEBWOLFSESSION
Instances	7
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 Controllable HTML Element Attribute (Potential XSS)** ✓

Description	This check looks at user-supplied input in query string parameters and POST data to identify where certain HTML attribute values might be controlled. This provides hot-spot detection for XSS (cross-site scripting) that will require further review by a security analyst to determine exploitability.
URL	http://127.0.0.1:9090/login?error=true
Method	GET
Attack	
Evidence	
Other Info	User-controlled HTML attribute values were found. Try injecting special characters to see if XSS might be possible. The page at the following URL: http://127.0.0.1:9090/login?error=true appears to include user input in: a(n) [input] tag [autofocus] attribute The user input found was: error=true The user-controlled value was: true
URL	http://127.0.0.1:9090/login?error=true
Method	GET
Attack	
Evidence	
Other Info	User-controlled HTML attribute values were found. Try injecting special characters to see if XSS might be possible. The page at the following URL: http://127.0.0.1:9090/login?error=true appears to include user input in: a(n) [input] tag [required] attribute The user input found was: error=true The user-controlled value was: true
Instances	2
Solution	Validate all input and sanitize output it before writing to any HTML attributes.
Reference	http://websecuritytool.codeplex.com/wikipage?title=Checks#user-controlled-html-attribute
CWE Id	20
WASC Id	20
Plugin Id	10031