

Claroty JSON SQLi Vulnerabilities

<https://campus.barracuda.com/doc/98217566/>

This article provides an update on the recently discovered JSON-based [SQL Injection Vulnerability](#) by [Team82](#).

The [Claroty T82](#) research team released a [blog](#) last week demonstrating a newly identified SQL injection in JSON-based SQL and how it bypasses many name-brand WAF vendors.

Exploit

The attack technique involves appending JSON syntax to SQL injection payloads. The attack affects only web applications using JSON.

Barracuda Load Balancer ADC Mitigation

The Barracuda Load Balancer ADC protect against this attack with an update in the existing SQL injection category of the Smart Signatures.


The default SQL injection medium and strict checks do not detect this variant, which employs JSON syntax. The new signature detects all identified variants of the JSON syntax-based attacks.

Barracuda Networks has pushed the new signature through Attack Definition Update version 1.222. The [Release Notes](#) is updated to reflect the changelog.

The Attack Definitions are available only as part of the Energize Updates subscription.

Action Required

1. Set **Automatic Updates** to **ON** for the Load Balancer ADC devices to receive the latest Attack Definition version 1.222.
2. Set the **Operating Mode** for the new attack pattern "*sql-tautology-conditions-json-bypass-string*" to **Active** in the **SECURITY > View Internal Patterns > Attack Types > sql-injection-medium** group.


Barracuda | Load Balancer ADC
admin Sign out English

BASIC TRAFFIC **SECURITY** ACCESS CONTROL NETWORK ADVANCED

Security Policies Allow/Deny Rules Website Profiles Advanced Security DDoS Prevention Libraries **View Internal Patterns** FTP Security

Search help topics

Attack Types
Help

Group	Pattern Name	Pattern Regex	Pattern Algorithm	Status	Case Sensitive	Operating Mode	
cross-site-scripting-strict	script-comments	<code>\s*(.)*\s*</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Copy Detail:
	script-...concat	<code>[""](.)*[""]</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	closing...html-tag	<code><v([\x08-\x0a\p{ph}]](1...</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	opening...html-tag	<code>\<([\x08-\x0a\p{ph}]](1...</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	arbitra...jection	<code>("")(.)*([\p{ph}]]+</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
sql-injection-medium	sql-tau...n-dbcmd	<code>[^:alnum:]]+(O...09]...</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Copy Detail:
	sql-dec...simple	<code>[^:alnum:]]+(D...[:al...</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-quote	<code>("")([\x09]*or)(\x09]</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-sel...command	<code>(([:!])[""](.)*[\p{ph}]]...</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-tau...string	<code>[^:alnum:]]+(O...")</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-tau...string	<code>("")(.)*(")</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-tau...extract	<code>sql-tautology-conditions-json-bypass-string d ...</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-tau...e-dbcmd	<code>[^:alnum:]]+(O...")</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-tau...n-dbcmd	<code>[^:alnum:]]+(O...")</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-exec-simple	<code>[^:alnum:]]+(E...'+')</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-uni...command	<code>union.*[""][:alnu...om[...</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:
	sql-tau...simple	<code>[^:alnum:]]+(O...+~[...</code>		On	No	<input type="radio"/> Passive <input checked="" type="radio"/> Active	Detail:

It is advised to watch out for false positives from this pattern and to contact [Barracuda Networks Technical Support](#) as required.

Related Articles:

- <https://claroty.com/team82/research/js-on-security-off-abusing-json-based-sql-to-bypass-waf>
- <https://securityaffairs.co/wordpress/139445/hacking/web-application-firewalls-waf-bypass.html>
- <https://www.techtarget.com/searchsecurity/news/252528217/Claroty-unveils-web-application-firewall-bypassing-technique>
- <https://www.itworldcanada.com/post/claroty-discovers-method-to-bypass-vendors-web-application-firewalls-waf>
- <https://gbhackers.com/bypass-web-application-firewalls/>

Figures

1. View_Internal_Patterns.png
2. Attack_Types.png
3. Pattern.png

© Barracuda Networks Inc., 2024 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.