WhiteHat SECURITY (https://www.whitehatsec.com)

Blog

⌂ (https://www.whitehatsec.com) › Blog (https://www.whitehatsec.com/blog/) › Technical Insight (https://www.whitehatsec.com/blog-category/technical-insight/) › Handling Untrusted JSON Safely

INDUSTRY SOLUTIONS

Podcast (Https://Www.whitehatsec.com/Blog-Category/Podcast/)

SECURITY RESEARCH

Aviator (Https://Www.whitehatsec.com/Blog-Category/Aviator/)

Technical Insight (Https://Www.whitehatsec.com/Blog-Category/Technical-Insight/)

Tools And Applications (Https://Www.whitehatsec.com/Blog-Category/Tools-And-Applications/)

True Stories Of The TRC (Https://Www.whitehatsec.com/Blog-Category/True-Stories-Of-The-Trc/)

Unsung Heroes (Https://Www.whitehatsec.com/Blog-Category/Unsung-Heroes/)

THOUGHT LEADERSHIP

Industry Observations (Https://Www.whitehatsec.com/Blog-Category/Industry-Observations/)

WHITEHAT SENTINEL

Events (Https://Www.whitehatsec.com/Blog-Category/Events/)

Web Application Security (Https://Www.whitehatsec.com/Blog-Category/Web-Application-Security/)

WhiteHat Security Products (Https://Www.whitehatsec.com/Blog-Category/Whitehat-Security-Products/)

THREAT BULLETINS

Breaking News (Https://Www.whitehatsec.com/Blog-Category/Breaking-News/)

TECHNICAL INSIGHT (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/TECHNICAL-INSIGHT/)

Handling Untrusted JSON Safely

Jim Manico (https://www.whitehatsec.com/author/jimmanico/) | January 11, 2013

JSON (JavaScript Object Notation) is quickly becoming the de-facto way to transport structured text data over the Web, a job also performed by XML. JSON is a limited subset of the object literal notation inherent to JavaScript, so you can think of JSON as just part of the JavaScript language. As a limited subset of JavaScript object notation, JSON objects can represent simple name-value pairs, as well as lists of values.

BUT, with JSON comes JavaScript and the potential for JavaScript Injection, the most critical type of Cross Site Scripting (XSS) (https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)).

Just like XML, JSON data need to be parsed to be utilized in software. The two major locations within a Web application architecture where JSON needs to be parsed are in the browser of the client and in application code on the server.

Parsing JSON can be a dangerous procedure if the JSON text contains untrusted data (https://www.owasp.org/index.php/Injection_Theory#Untrusted_Data). For example, if you parse untrusted JSON in a browser using the JavaScript “eval” function, and the untrusted JSON text itself contains JavaScript code, the code will execute during parse time.

From http://www.json.org/js.html (http://www.json.org/js.html) :

“To convert a JSON text into an object, you can use the eval() function. eval() invokes the JavaScript compiler. Since JSON is a proper subset of JavaScript, the compiler will correctly parse the text and produce an object structure. The text must be wrapped in parentheses to avoid tripping on an ambiguity in JavaScript’s syntax.

```
var myObject = eval(‘(‘ + myJSONtext + ‘)’);
```

The eval function is very fast. However, it can compile and execute any JavaScript program, so there can be security issues.”

So the essential question is: How can programmers and applications parse untrusted JSON safely?

Parsing JSON safely, *Client Side*

The most common way to parse JSON safely in a modern browser is to utilize the JSON.parse (http://msdn.microsoft.com/en-us/library/ie/cc836466(v=vs.94).aspx) method inherent to JavaScript. Here is a good reference that describes the state of JSON.parse browser support (http://caniuse.com/#search=JSON.parse). And for legacy browsers that do not support native JSON parsing, there is always Douglas Crockford’s JSON parsing library for legacy browsers (https://github.com/douglascrockford/JSON-js).

Parsing JSON in the browser is often the result of an asynchronous request returning JSON to the browser. Another technique that is becoming more common is to embed JSON directly in a Web page server side, and then to parse and render that JSON on the client side. The mechanism of embedding JSON safely in a Web page is described here: https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet#RULE_.233.1_-_HTML_escape_JSON_values_in_an_HTML_context_and_read_the_data_with_JSON.parse (https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet#RULE_.233.1_-_HTML_escape_JSON_values_in_an_HTML_context_and_read_the_data_with_JSON.parse)

In this description, Step 1 shows safely embedded JSON on a Web page through HTML Entity Encoding:

```
<span style=”display:none” id=”init_data”>

<%= data.to_json %>  <– data is HTML escaped –>

</span>
```

Steps 2 and 3 includes decoding the JSON data and then parsing it safely:

```
<script>

// unescapes the content of the span

var jsonText = document.getElementById(‘init_data’).innerHTML;

// parse untrusted JSON safely

var initData = JSON.parse(jsonText);

</script>
```

Parsing JSON safely, *Server Side*

It’s important to use a formal JSON parser when handling untrusted JSON on the server side. For example, the Java Programing language can utilize the OWASP JSON Sanitizer (https://www.owasp.org/index.php/OWASP_JSON_Sanitizer) for Java. The OWASP JSON Sanitizer project aspires to accomplish the following goals:

“Given JSON-like content, converts it to valid JSON.

This can be attached at either end of a data-pipeline to help satisfy Postel’s principle:

Be conservative in what you do; be liberal in what you accept from others.

Applied to JSON-like content from others, the OWASP JSON Sanitizer will produce well-formed JSON that should satisfy any parser you use.

Applied to your own output before you send, it will coerce minor mistakes in encoding and make it easier to embed your JSON in HTML and XML.”

The OWASP JSON Sanitizer project was created and is maintained by Mike Samuel, an esteemed member of the Google Application Security Team. For more information on the OWASP JSON Sanitizer, please visit the OWASP JSON Sanitizer Google Code page (http://code.google.com/p/json-sanitizer/).

I hope this article helps you develop safer parsing of JSON in your applications. Please drop me a line if you have any questions at jim@owasp.org. (mailto:jim@owasp.org)

0 Comments **whitehatsec** 1 **Login**

♡ **Recommend** **Share** **Sort by Best**

Start the discussion…

Be the first to comment.

Subscribe **Add Disqus to your site Add Disqus Add** **Privacy**

Related Articles

STATIC ANALYSIS (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/STATIC-ANALYSIS/) - WHITEHAT SECURITY PRODUCTS (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/WHITEHAT-SECURITY-PRODUCTS/)

Find and Fix JavaScript Vulnerabilities Early in the SDLC

(https://www.whitehatsec.com/blog/find-and-fix-javascript-vulnerabilities/) Ruchika Mishra (https://www.whitehatsec.com/author/ruchika-mishra/) | April 04, 2016

INDUSTRY OBSERVATIONS (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/INDUSTRY-OBSERVATIONS/) - VULNERABILITIES (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/VULNERABILITIES/) - WEB APPLICATION SECURITY (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/WEB-APPLICATION-SECURITY/) - WHITEHAT HACKERKAST (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/WHITEHAT-HACKERKAST/)

#HackerKast 8: Recap ofJPMC Breach, Hacking Rewards Programs and TOR Version of Facebook

(https://www.whitehatsec.com/blog/hackerkast-8/) Matt Johansen (https://www.whitehatsec.com/author/mattjohansen/) | November 11, 2014

INDUSTRY OBSERVATIONS (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/INDUSTRY-OBSERVATIONS/) - TECHNICAL INSIGHT (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/TECHNICAL-INSIGHT/)

North Korea's Naenara Web Browser: It's Weirder Than We Thought

(https://www.whitehatsec.com/blog/north-koreas-naenara-web-browser-its-weirder-than-we-thought/) Robert Hansen (https://www.whitehatsec.com/author/roberthansen/) | January 08, 2015

Most Recent Articles

VULNERABILITIES (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/VULNERABILITIES/) - WEB APPLICATION SECURITY (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/WEB-APPLICATION-SECURITY/)

Apache Struts 2 CVE-2017-5638: Are My Applications Vulnerable to Remote Code Execution?

(https://www.whitehatsec.com/blog/apache-struts-cve-2017/) Peter Monahan (https://www.whitehatsec.com/author/peter-monahan/) | March 10, 2017

INDUSTRY OBSERVATIONS (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/INDUSTRY-OBSERVATIONS/)

Evolution and the Movement of AppSec to the Cloud

(https://www.whitehatsec.com/blog/movement-of-appsec-to-the-cloud/) Jeannie Warner (https://www.whitehatsec.com/author/jeannie-warner/) | March 09, 2017

INDUSTRY OBSERVATIONS (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/INDUSTRY-OBSERVATIONS/) - VULNERABILITIES (HTTPS://WWW.WHITEHATSEC.COM/BLOG-CATEGORY/VULNERABILITIES/)

Verizon Needed a Security Facts Label

(https://www.whitehatsec.com/blog/verizon-needed-a-security-facts-label/) Eric Sheridan (https://www.whitehatsec.com/author/ericsheridan/) | March 07, 2017

Subscribe to Communications:

(https://info.whitehatsec.com/Opt-In-Website.html)

Products (https://www.whitehatsec.com/products/)

Dynamic Application Security Testing (https://www.whitehatsec.com/products/dynamic-application-security-testing/)

Static Application Security Testing (https://www.whitehatsec.com/products/static-application-security-testing/)

Mobile Application Security Testing (https://www.whitehatsec.com/products/mobile-application-security-testing/)

Company (https://www.whitehatsec.com/company/)

Leadership (https://www.whitehatsec.com/company/leadership/)

Threat Research Center (https://www.whitehatsec.com/company/threat-research-center/)

In The News (/news/)

(https://www.whitehatsec.com)

Computer-Based Training (https://www.whitehatsec.com/products/computer-based-training/)

Solutions By Role (/products/#solutions)

Solutions By Need (/products/#solutions)

Solutions By Industry (/products/#solutions)

Customers (https://www.whitehatsec.com/customers/)

Case Studies (https://www.whitehatsec.com/customers/case-studies/)

Support (https://www.whitehatsec.com/customers/support/)

Partners (https://www.whitehatsec.com/partners/)

Technology Partners (https://www.whitehatsec.com/partners/technology-partners/)

Resale Partners (https://www.whitehatsec.com/partners/resale-partners/)

Industry Recognition (https://www.whitehatsec.com/company/industry-recognition/)

Careers (https://www.whitehatsec.com/company/careers/)

Events Calendar (/events/)

Community (https://www.whitehatsec.com/community/)

Contact (https://www.whitehatsec.com/company/contact/)

Blog (https://www.whitehatsec.com/blog/)

Resources (https://www.whitehatsec.com/resources/)

Trending Now (https://www.whitehatsec.com/trending/)

The front line of application security.

Terms & Conditions (https://www.whitehatsec.com/terms-conditions/)

Safe Harbor Privacy Policy (https://www.whitehatsec.com/safe-harbor-privacy-policy/)

(https://sentinel.whitehatsec.com/gateway/certified/confirmed.html?badge_key=fce35782-c672-47d6-887d-74944439b2d1)