Exploiting JSON Framework : 7 Attack Shots

[Unvieling Insecurities In JSON]

By: Aditya K Sood
http://zeroknock.metaeye.org
Metaeye Security.

Abtract:
This article define the layout of the exploiting factors of web attacks ie where the JSON framework is compromised.The article is consistent in explaining the pros of the web attack related to JSON.

String Defracturing:
First of all the string defracturing relates to fusing of two different strings and manipulate by joining together to form one large string that is used as malformed input.This exploiting factor comes to play as a result of a feature of JSON.what happens in this is that the JSON pass string as normal string through function toJSONString() function.The malicious attacker can easily inject script as:

0x01] Direct Injecting in which the string passed is the script only bu not any kind of desired string.
0x02] Indirect Injecting means the script is going to be fused with the string as encoded string so that it seems as generic string and the contents are not readable by any user or the security enhanced program.

```
var myJSONText = myObject.toJSONString();
```

So one can see very clearly how the string is processed and manipulated in the best possible way by the attacker to inject malicious content in the normal string.

Exploiting Features: Malicious Hashing And Listing:
This is another exploiting factor of JSON based on the language feature.The exploitation in real sense is to get the rogue work done by manipulating the system oriented and language features.The JSON framework provides object oriented feature like hashing , listing etc.The point of talk is that whenn ever a hash data structure or listing is designed the reference objects are adhered to the keys. The keys hold the referential objects that point to the definitive data to be located by that key.In this the attacker can easily inscribed a script or document object based exploit and let it to trigger.It work as a desired injection through the JSON or considered to be as XSS kind.Lets see :

```
{"Tools":[{"Link":"www.example.com","Refer":"Interesting link"}]}
```

The Link and Refer keys can be injected with rogue scripts in the reference objects and the script will automatically burst out.So the feature can be exploited very easily in this way.

Exploiting JSON Libraries
The JSON libraries are somewhat mismanaged in parsing the requests or the function callings.Getting into deeper aspect ,look at the eval function that performs unescape ie what ever the objects or entities present in the response text are parsed in the definite manner ie and during that get replaced with the specified characters or references that further processed as javascript.To betray this the entity should be encoded and wrapped with manual unencode

call which means second level encoding and unencoding must be there in the libraries that are specific to the JSON.The problem arise in the structuraldevelopment of the language as the libraries are unable to handle these doubly encoded request as a result of which the manipulation can be done.

CSRF JSON:

The CSRF stands for cross site request forgery.This CSRF can cause major security havoc for the web sites using JSON.The malicious code works in a very crafty way.The exploitation starts by having you visit a page with a refernce that is embedded in the source that relates to third party.The third party uses cookie to verify once identity.In some major browsers, cookies will be sent to the third party site and by unknowingly the access is provided to the malicious user that further control the user's private data.This issue has been exploited the attackers to launch third party attacks and exploitation is getting easy in the prescribed sense. The issue can be resolved as:

0x01] The values can not be guessed.
0x02] Encoding in the urls.
0x03] Embedding Dynamic values.
0x04] Domain cookies transfer should be restricted.
0x05] XMLHttpRequest advised.Stringent in its usage.

So CSRF has also become the base of JSON exploitation.

Fused Proxy Attacks : Domain Crossing:

As one know the JSON can be fused with the XMLHttp method to perform the requisite function in the domain.This done to ensure security feature in the defined domain.Actually the web page restriction sets the page to be attached to the domain in which it is called.No cross domain referencing can be done.Like if the website A called a script then this script cannot be called from the other website B because security restriction is there. This means domain bypassing is not possible.This can be easily bypassed by setting proxy in the same language on the defined server.The proxy act as a bridge between websiteA and website B which makes the domain crossing possible.This make the data exchange possible across the domain.

Internet Explorer shows an alert stating that a potential security risk exists but gives the user a choice of whether to continue with the request. Firefox simply stops the request and shows an error message in the JavaScript console.

JSON Padding : Cross Site Service Attacks:

The JSONP relates to the cross site services provided by the clients.This is the outcome of JSON RPC.The services are applied on the six basic steps:

A] The page which used for creating service registers callback to recieve response.
B] Afterwards page called script element in conjunction toJSON used.
C] The request is encoded and the callback is feeded thescript URL.
D] On the server side the request is executed
E] Callback invocation is created with result data as an argument.
F] The page loads and executes result invoking the callback.

This is definitive way of JSON RPC.The malicious attacker can code or insert a script in such a way that when ever script element is called with the data , the malicious content gets fused with the required data and the response extracted is based on that.

The manipulation can be done in the system context too.

Banner Grabbing : JSON-XML:

The XMLHttpRequest object provides two properties that provide access to the server response.The first property, responseText,simply provides the response as a string. The second property,responseXML,provides the response as an XML object. Retrieving the response as simple text is fine for simple use cases, such as when the response is displayed in an alert box or the response is a simple one-word phrase indicating success or failure.But if the code is set in detailed layout there are other calling functions such as getAllResponseHeader or getReponseHeader in which specific request is set and the response extracted will dethrone the required information needed.

Conclusion:

Every single technology comes to play have both positive and negative layouts.The JSON framework no doubt has enhanced the working of web making it web 2.0 but also increased the risk of security breaching.