Are Shallow Copies Dangerous?

Let's assess if using shallow copies is safe or not.

We can conclude that making shallow copies are sometimes not enough. When the copied object is modified, it may result in unwanted consequences. Most of the time, software development best practices and restricted workflow allow us to use shallow copies of objects without problems.

For instance, in BackboneJs, the default implementation of the `toJSON` method of models looks like this:

```
toJSON: function(options) {
    return _.clone(this.attributes);
}
```

I have not seen complaints on BackboneJs forums addressing the problem of retrieving the results of the `toJSON` method and modifying the contents of the model attributes. The reason is that the `toJSON` method is mostly used for one of the two purposes:

- Serialization: preparing a JSON payload for an AJAX request,
- Presentation: preparing a JavaScript object and giving it to a templating engine.

Although some extreme cases may apply, in general, both of these operations only access the clone. Modifying nested data structures are hardly required.

It is very important to detect situations when shallow copies are not enough. In these cases, either think about refactoring your code or using a deep copy.