What is NetCURLv6

- (is) Licensed under the Apache License, Version 2.0
- NetCurl ***was not*** an attempt to rebuild the wheel.
- NetCurl was back in time a regular (yet another) curl-wrapper, with the ability to test simpler input- and output data that was returned from proxies (socks, TOR, etc). The primary goal of the module was to be able to both auto scan different kind of websites for open proxies and then test them. As soon as there was proxies detected, they was added to DNS blacklists..
- NetCurl is a preconfigured wrapper of standard libraries, created to get started as quick as possible without any further knowledge than the phrase *"I'd like to get the content of XXX site, in the format of YYY, regardless of what they run and return"*. Instead of setting up classes, calls and methods, NetCurl activates a state of high verbosity, so that the developer can pick the data freely.
- NetCurl can, if WordPress is present, switch over to the WP_HTTP-class, instead of using internal functions for the calls.
- NetCurl can, if GuzzleHttp is present, switch over to GuzzleHttp to utilize and automatically set up a communication link via this external library. If curl is available, Guzzle will use curl as the first option (which makes this a bit useless). However, if curl is not present, NetCurl can instead utilize the streams driver provided in Guzzle. NetCurl will become a *GuzzleWrapper*, so to speak.
- NetCurl follows the PHP and curl development and handles certificates with a best practice. It can however, override this and lower security by completely ignore SSL validations.
- NetCurl parses and splits up data in sections of **header**, **response code**, **body** and a **pre-parsed data container**. The header can normally be viewed as an array, the body contains the received data content as is (raw, untouched), while the parsed data tries to automatically detect content and convert it to readable arrays/objects.
- NetCurl supports SoapClient. It's done by the class **Tornevall_simpleSoap**. This module has dependencies to xml and SoapClient. SoapCalls are handled automatically from the base class: Adding ?wsdl to the URL fetched by NetCurl, will automatically initiate a session for soap. The call can also be sent without ?wsdl, but in that case you have to tell the module to go SOAP.
- NetCurl supports native basic authentication (curl) and enables similar support when going through a GuzzleStream. For curl, authentication comes for free, while you're pretty much on your own when going stream. NetCurl also makes sure that the SoapClient adopts authentication data set by the **setAuthentication()**-method.
- NetCurl was back in time more focused to be a "network attack protection tool", rather than a functional network communications tool. Many things has happened since then.

Parsed responses

If you make a web request to a site that responds in json formatted code, the parsed content will be returned as an object when running getParsedContent(). If the request returns XML, the parsed content will make sure it is converted in the same way. By running NetCurl in an older state (setChain=off) the raw data containers can be accesses without the method getParsedContent().