Based on the results of different penetration tests, security of Hydras 3 net has been further improved in Hydras 3 net version 4.40.

Setting up a Hydras 3 net server in the most secure way without breaking compatibility to the existing IT environment requires however some configuration effort and a deeper understanding of HTTPS communication.

In this document the detailed improvements and options for setting up a secure Hydras 3 net server are explained.

The most basic step in setting up a secure Hydras 3 net server is by activating HTTPS in the options dialog on page “Web Server”. If no certificate is specified, Hydras 3 net will automatically create a self-signed certificate. However this type of certificate is often no longer accepted by browsers or you get warning messages, so a certificate from an official Certificate Authority should be used. Typically you import this certifcate in the Windows certificate store in the root area. In this case it is sufficient to enter ROOT in the certificate field in the options dialog.

If possible, you should disable the insecure HTTP server. However in some cases you still need support for HTTP, e.g. when older OTT ecoLog 500 or DuoSens data loggers transmit data to Hydras 3 net, as they do not support HTTPS.

General security enhancements

Some of the security enhancements are always active without explicit configuration by an admin.

These enhancements are explained in this chapter.

Strict server-side authorization

While in in the past the client did check if a user is authorized to perform a certain action, this is now always done on client and server side. On client side to offer only actions in the UI to the user that he is allowed to perform and in addition on the server side in order to prevent that a hacker bypassed the client and sent a request directly to the server. Please note that as default all new users that are created have full privileges. In order to restrict privileges for users the specific privileges for the users have to be adjusted using menu item “Extras > Privilege manager” in the Hydras 3 main form.

The server-side authorization is used both for Hydras 3 net windows and web clients.

Enforcing a content security policy (CSP)

The idea of a content security policy is that the web server tells the browser what kind of actions are allowed in a web page and where the elements of this page can come from in order to protect the user from attacks. The content security policy is transmitted to the browser with the HTTP header “Content-Security-Policy”.

There are different aspects that are defined by the CSP in order to protect the user. These aspects are explained in the next chapters

Use of Javascript

While Javascript is required to provide user interaction on a web page, it is also commonly used to attack web pages and steal sensitive data. The ***script-src*** directive in the CSP header of a Hydras 3 net server looks like this:

script-src self A.B.C.D 'nonce-41110Ql583105722583OIl41110';

where A.B.C.D is the server address.

This tells the browser that only script code that comes directly from the server itself is allowed to be loaded and executed. In addition a so called nonce is defined, which is a session specific code, that has to be used in any <script> block, that is inserted in the loaded web page. So injecting generic script code in the web page will not work, as the surrounding <script> element will not know the nonce.

An implicit aspect of the CSP header for Javascript is that no inline Javascript is allowed (e.g. assigning an event handler directly in a button element using <button onClick="…") and that the Javascript eval() function cannot be used. This function executes dynamic script code that could have been loaded from a server at runtime. The Hydras 3 net web interface has been rewritten, so that no inline code and eval functions are used since V 4.40 and that all embedded <script> elements within an HTML page use the nonce attribute.

Images

The *img-src* directive tells the browser from which location images can be safely loaded.

The ***img-src*** directive in the CSP header of a Hydras 3 net server looks like this:

*img-src self A.B.C.D *.openstreetmap.org data*

This tells the browser that only image files from the Hydras 3 net server are allowed, that delivered ther HTML page and that images from openstreetmap.org can be loaded, which are required to show the map.

CSS Styles

The *style-src* directive tells the browser from which location style sheets can be safely loaded.

The ***style-src*** directive in the CSP header of a Hydras 3 net server looks like this:

style-src A.B.C.D 'unsafe-inline'

This restricts the origin of the style sheets to the Hydras 3 net server and allow the use of inline style information, which is used ocassionally to provide style information directly in the HTML elements.

Other file types

The *default-src* directive tells the browser that all file types that were not explicitly covered by the specific directives are only allowed from the Hydras 3 net server.

Anti CSRF (Cross site request forgery) tokens

All HTML forms of the Hydras 3 net web client use now so called Anti CSRF (Cross site request forgery) tokens in order to make sure that all submitted HTML forms are actually from the original web client and the authenticated user and not from a malicious web page.

Prevent session re-negotiation attacks

The Hydras 3 net server does no longer allow to perform a so called TLS session renegotiation, which could be used to perform a man-in-the-middle attack or a Denial of Service (DoS) attack.

Additional secure HTTP headers

X-Content-Type-Options: nosniff

With this header the browser does not automatically try to determine the content-type of a dynamically loaded file, but relies on the content-type header coming from the server. This helps to prevent potential cross-site-scripting (XSS) attacks.

X-Frame-Options: sameorigin

With this header the browser is only allowed to display content in a frame that originates from the same Hydras 3 net server, as the currently displayed web page. This helps to prevent potential cross-site-scripting (XSS) attacks.

Configurable security enhancements

While all security enhancements described so far are always active and cannot be configured by the user, there are more options, that require explicit configuration by the user. These are described in this chapter.

HTTP Strict-Transport-Security

With the Strict-Transport-Security header, a web server can instruct the browser to use always HTTPS instead of HTTP for communication with the domain referenced in the URL.

To activate this HTTP header, the registry key below must be set to 1.

HKEY_CURRENT_USER\Software\Ott-Hydrometry\Hydras3\Communication\Web_Server2_HSTS

This will activate the Strict-Transport-Security header with the setting “max-age=31536000”, which means this setting will be valid for one year.

With the optional registry key below, the full value of the Strict-Transport-Security header can be explicitly configured:

HKEY_CURRENT_USER\Software\Ott-Hydrometry\Hydras3\Communication\Web_Server2_HSTS_Details

Example:

“max-age=31536000; includeSubDomains”

The setting is also valid for one year and in addition HTTPS will be enforced for all sub domains.

With the “preload” directive in the key, the domain can be registered with Google Chrome to always use HTTPS. Details can be found on https://hstspreload.org.

Caution: This header has to be used carefully, as access to some sites might be blocked unintentionally.
If some pages within the domain (or subdomains if “includeSubDomains” is used) are for legacy reasons only accessible via HTTP, this header will lockout the browser from accessing these pages, as HTTPS will always be enforced!

This header has only an effect on browsers. A Hydras 3 net windows client or an OTT data logger is not affected by this setting. If e.g. older devices like OTT ecoLog 500 are sending data to Hydras 3 net, you still have to use an insecure HTTP port, as these devices do not support HTTPS. However as this header only forces browser communication to use HTTPS, the ecoLog 500s could still send data via HTTP.

When you are running Hydras 3 as a system service, plase make sure to use HKEY_USERS\.Default\... instead of HKEY_CURRENT_USER\...

Alternatively you can invoke menu item “File > Save options for execution as system service” in the Hydras 3 main window (running as application) after you have set the registry keys for CURRENT_USER, so that they will also be applied for the system service.

Preventing HTTP Host header attacks

The HTTP host header is often used by attackers for attempts to inject malicious code in web pages (https://portswigger.net/web-security/host-header). The Hydras 3 net server automatically sanitizes this header, so that <script> elements cannot be injected. However in some cases, the host header is read by server side code and inserted in HTML pages to reference other content. To avoid that a manipulated host header could redirect the user to a different page, the most secure way is to explicitly configure the host domain name in a registry key, that is then used by server side logic instead the actual content of the host header field.

The host domain name can be explicitly set with the registry key below:

HKEY_CURRENT_USER\Software\Ott-Hydrometry\Hydras3\Communication\Web_Server_Host

Enforce certificate chain

By default, the Hydras 3 net server does not send the full certificate chain to the client while establishing a TLS connection. The reason for this is that older netDL dataloggers with firmware before 3.02.0 do not handle this correctly. Due to this an online test with SSL Labs (https://www.ssllabs.com/ssltest) results in a rating of B, as the server's certificate chain is incomplete.

To enforce sending of the complete certifcate chain to the client set the registry key below to 1.

HKEY_CURRENT_USER\Software\Ott-Hydrometry\Hydras3\Communication\Web_Server2_ForceCertificateChain

However, be aware that in this case netDL loggers with firmware older than 3.02.0 will not be able to communicate via HTTPS with the Hydras 3 net server.

Links:

https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html

https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

https://www.invicti.com/blog/web-security/protecting-website-using-anti-csrf-token/

https://scanrepeat.com/web-security-knowledge-base/x-content-type-options-header-missing

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html

https://portswigger.net/web-security/host-header