[Skip to main content](https://lms.alnafi.com/xblock/block-v1:alnafi+DCCS102+2025_DCCS+type@vertical+block@53116eacc6a744c1ae43b2d5180b76f0?exam_access=&recheck_access=1&show_bookmark=0&show_title=0&view=student_view#main)

Sensitive data must be protected when it is transmitted through the network. If data is transmitted over HTTPS or encrypted in another way the protection mechanism must not have limitations or vulnerabilities, as explained in the broader article [Testing for Weak Transport Layer Security](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/01-Testing_for_Weak_Transport_Layer_Security) and in other OWASP documentation:

As a rule of thumb if data must be protected when it is stored, this data must also be protected during transmission. Some examples for sensitive data are:

* Information used in authentication (e.g. Credentials, PINs, Session identifiers, Tokens, Cookies…)
* Information protected by laws, regulations or specific organizational policy (e.g. Credit Cards, Customers data)

If the application transmits sensitive information via unencrypted channels - e.g. HTTP - it is considered a security risk. Some examples are Basic authentication which sends authentication credentials in plain-text over HTTP, form based authentication credentials sent via HTTP, or plain-text transmission of any other information considered sensitive due to regulations, laws, organizational policy or application business logic.

Examples for Personal Identifying Information (PII) are:

* Social security numbers
* Bank account numbers
* Passport information
* Healthcare related information
* Medical insurance information
* Student information
* Credit and debit card numbers
* Drivers license and State ID information

**How to Test**

Various types of information that must be protected, could be transmitted by the application in clear text. It is possible to check if this information is transmitted over HTTP instead of HTTPS, or whether weak ciphers are used. See more information about insecure transmission of credentials [OWASP Top 10 2017 A3-Sensitive Data Exposure](https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure) or [Transport Layer Protection Cheat Sheet](https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html).

**Example 1: Basic Authentication over HTTP**

A typical example is the usage of Basic Authentication over HTTP. When using Basic Authentication, user credentials are encoded rather than encrypted, and are sent as HTTP headers. In the example below the tester uses [curl](https://curl.haxx.se/) to test for this issue. Note how the application uses Basic authentication, and HTTP rather than HTTPS

$ curl -kis http://example.com/restricted/ HTTP/1.1 401 Authorization Required Date: Fri, 01 Aug 2013 00:00:00 GMT WWW-Authenticate: Basic realm="Restricted Area"  
Accept-Ranges: bytes Vary: Accept-Encoding Content-Length: 162 Content-Type: text/html   
<html><head><title>401 Authorization Required</title></head> <body bgcolor=white> <h1>401 Authorization Required</h1> Invalid login credentials! </body></html>

**Example 2: Form-Based Authentication Performed over HTTP**

Another typical example is authentication forms which transmit user authentication credentials over HTTP. In the example below one can see HTTP being used in the action attribute of the form. It is also possible to see this issue by examining the HTTP traffic with an interception proxy.

<form action="http://example.com/login"> <label for="username">User:</label> <input type="text" id="username" name="username" value=""/><br /> <label for="password">Password:</label> <input type="password" id="password" name="password" value=""/> <input type="submit" value="Login"/>  
</form>

**Tools**

* Identity Finder
* Wireshark
* TCPDUMP