



Acunetix Threat Level 2

One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.



High



Medium



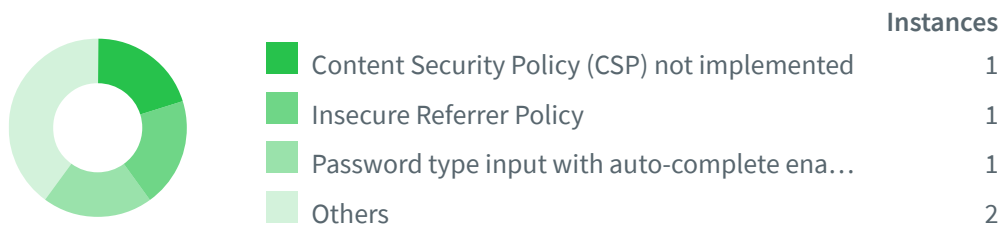
Low



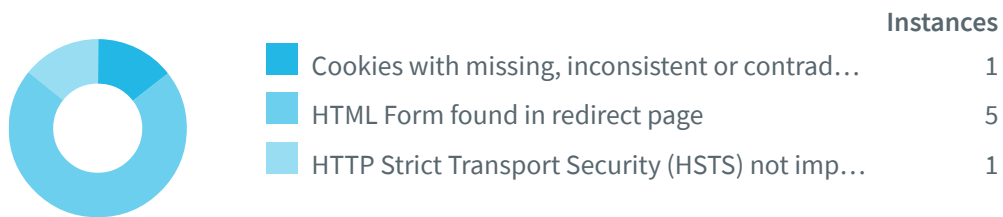
Informational

| Severity | Vulnerabilities | Instances |
|-----------------|-----------------|-----------|
| ! High | 0 | 0 |
| ! Medium | 3 | 3 |
| ! Low | 3 | 7 |
| i Informational | 5 | 5 |
| Total | 11 | 15 |

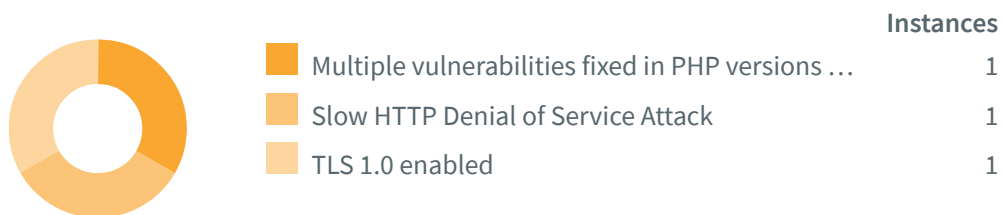
Informational













Low Severity



Medium Severity



Impacts

| SEVERITY | IMPACT |
|---|---|
|  Medium | <div>1</div> Multiple vulnerabilities fixed in PHP versions 5.5.12 and 5.4.28 |
|  Medium | <div>1</div> Slow HTTP Denial of Service Attack |
|  Medium | <div>1</div> TLS 1.0 enabled |
|  Low | <div>1</div> Cookies with missing, inconsistent or contradictory properties |
|  Low | <div>5</div> HTML Form found in redirect page |
|  Low | <div>1</div> HTTP Strict Transport Security (HSTS) not implemented |
|  Informational | <div>1</div> Content Security Policy (CSP) not implemented |
|  Informational | <div>1</div> Insecure Referrer Policy |
|  Informational | <div>1</div> Password type input with auto-complete enabled |
|  Informational | <div>1</div> PHP Version Disclosure |
|  Informational | <div>1</div> TLS 1.1 enabled |

Multiple vulnerabilities fixed in PHP versions 5.5.12 and 5.4.28

List of vulnerabilities that were fixed in PHP versions 5.5.12 and 5.4.28:

Core:

- Fixed bug #61019 (Out of memory on command stream_get_contents).
- Fixed bug #64330 (stream_socket_server() creates wrong Abstract Namespace UNIX sockets).
- Fixed bug #66182 (exit in stream filter produces segfault).
- Fixed bug #66736 (fpassthru broken).
- Fixed bug #67024 (getimagesize should recognize BMP files with negative height).
- Fixed bug #67043 (substr_compare broke by previous change).

cURL:

- Fixed bug #66562 (curl_exec returns differently than curl_multi_getcontent).

Date:

- Fixed bug #66721 (__wakeup of DateTime segfaults when invalid object data is supplied).

Embed:

- Fixed bug #65715 (php5embed.lib isn't provided anymore).

Fileinfo:

- Fixed bug #66987 (Memory corruption in fileinfo ext / bigendian).

FPM:

- Fixed bug #66482 (unknown entry 'priority' in php-fpm.conf).
- Fixed bug #67060 (possible privilege escalation due to insecure default configuration). (CVE-2014-0185)).

Json:

- Fixed bug #66021 (Blank line inside empty array/object when JSON_PRETTY_PRINT is set).

LDAP:

- Fixed issue with null bytes in LDAP bindings.

mysqli:

- Fixed problem in mysqli_commit()/mysqli_rollback() with second parameter (extra comma) and third parameters (lack of escaping).

Openssl:

- Fixed bug #66942 (memory leak in openssl_seal()).
- Fixed bug #66952 (memory leak in openssl_open()).

SimpleXML:

- Fixed bug #66084 (simplexml_load_string() mangles empty node name).

SQLite:

- Fixed bug #66967 (Updated bundled libsqlite to 3.8.4.3)

XSL:

- Fixed bug #53965 (<xsl:include> cannot find files with relative paths when loaded with "file:///")

Apache2 Handler SAPI:

- Fixed Apache log issue caused by APR's lack of support for %zu (APR issue https://issues.apache.org/bugzilla/show_bug.cgi?id=56120)

Impact

Multiple vulnerabilities were fixed with this update (impact is different for each vulnerability).

<https://fowom.dialog.lk/>

Recommendation

Upgrade to the latest version of PHP.

References

[PHP 5 ChangeLog](https://www.php.net/ChangeLog-5.php#5.5.12)

<https://www.php.net/ChangeLog-5.php#5.5.12>

Slow HTTP Denial of Service Attack

Your web server is vulnerable to Slow HTTP DoS (Denial of Service) attacks.

Slowloris and Slow HTTP POST DoS attacks rely on the fact that the HTTP protocol, by design, requires requests to be completely received by the server before they are processed. If an HTTP request is not

complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this creates a denial of service.

Impact

A single machine can take down another machine's web server with minimal bandwidth and side effects on unrelated services and ports.

<https://fowom.dialog.lk/>

Time difference between connections: 10003 ms

Recommendation

Consult Web references for information about protecting your web server against this type of attack.

References

[Slowloris DOS Mitigation Guide](https://www.funtoo.org/Slowloris_DOS_Mitigation_Guide)

https://www.funtoo.org/Slowloris_DOS_Mitigation_Guide

[Protect Apache Against Slowloris Attack](https://web.archive.org/web/20180329210925/http://blog.secaserver.com/2011/08/protect-apache-slowloris-attack/)

<https://web.archive.org/web/20180329210925/http://blog.secaserver.com/2011/08/protect-apache-slowloris-attack/>

TLS 1.0 enabled

The web server supports encryption through TLS 1.0. TLS 1.0 is not considered to be "strong cryptography" as defined and required by the PCI Data Security Standard 3.2(.1) when used to protect sensitive information transferred to or from web sites. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.

Impact

An attacker may be able to exploit this problem to conduct man-in-the-middle attacks and decrypt communications between the affected service and clients.

<https://fowom.dialog.lk/>

Confidence: 100%

The SSL server (port: 443) encrypts traffic using TLSv1.0.

Recommendation

It is recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher.

References

[Are You Ready for 30 June 2018? Saying Goodbye to SSL/early TLS](https://blog.pcisecuritystandards.org/are-you-ready-for-30-june-2018-sayin-goodbye-to-ssl-early-tls)

<https://blog.pcisecuritystandards.org/are-you-ready-for-30-june-2018-sayin-goodbye-to-ssl-early-tls>

[PCI 3.1 and TLS 1.2 \(Cloudflare Support\)](https://support.cloudflare.com/hc/en-us/articles/205043158-PCI-3-1-and-TLS-1-2)

<https://support.cloudflare.com/hc/en-us/articles/205043158-PCI-3-1-and-TLS-1-2>

Cookies with missing, inconsistent or contradictory properties

At least one of the following cookies properties causes the cookie to be invalid or incompatible with either a different property of the same cookie, of with the environment the cookie is being used in. Although this is not a vulnerability in itself, it will likely lead to unexpected behavior by the application, which in turn may cause secondary security issues.

Impact

Cookies will not be stored, or submitted, by web browsers.

<https://fowom.dialog.lk/>

Verified

List of cookies with missing, inconsistent or contradictory properties:

- <https://fowom.dialog.lk/pages/login.php>

Cookie was set via:

```
Set-Cookie: PHPSESSID=b4ejcoptollp4tufu3j6eker22; path=/;HttpOnly;Secure
```

This cookie has the following issues:

- Cookie without SameSite attribute.

When cookies lack the SameSite attribute, Web browsers may apply different and some

Request

```
GET /pages/login.php HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36
Connection: Keep-alive
```

Recommendation

Ensure that the cookies configuration complies with the applicable standards.

References

[MDN | Set-Cookie](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie)

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie>

[Securing cookies with cookie prefixes](https://www.sjoerdlangkemper.nl/2017/02/09/cookie-prefixes/)

<https://www.sjoerdlangkemper.nl/2017/02/09/cookie-prefixes/>

[Cookies: HTTP State Management Mechanism](https://tools.ietf.org/html/draft-ietf-httpbis-rfc6265bis-05)

<https://tools.ietf.org/html/draft-ietf-httpbis-rfc6265bis-05>

[SameSite Updates - The Chromium Projects](https://www.chromium.org/updates/same-site)

<https://www.chromium.org/updates/same-site>

[draft-west-first-party-cookies-07: Same-site Cookies](https://tools.ietf.org/html/draft-west-first-party-cookies-07)

<https://tools.ietf.org/html/draft-west-first-party-cookies-07>

HTML Form found in redirect page

Manual confirmation is required for this alert.

An HTML form was found in the response body of this page. However, the current page redirects the visitor to another page by returning an HTTP status code of 301/302. Therefore, all browser users will not see the contents of this page and will not be able to interact with the HTML form.

Sometimes programmers don't properly terminate the script after redirecting the user to another page. For example:


```

<?php
    if (!isset($_SESSION["authenticated"])) {
        header("Location: auth.php");
    }
?>
<title>Administration page</title>
<form action="/admin/action" method="post">
    <!-- ... form inputs ... -->
</form>

<!-- ... the rest of the administration page ... -->

```

This script is incorrect because the script is not terminated after the "header("Location: auth.php");" line. An attacker can access the content the administration page by using an HTTP client that doesn't follow redirection (like HTTP Editor). This creates an authentication bypass vulnerability. The correct code would be

```

<?php
    if (!isset($_SESSION[auth])) {
        header("Location: auth.php");
        exit();
    }
?>
<title>Administration page</title>
<form action="/admin/action" method="post">
    <!-- ... form inputs ... -->
</form>

<!-- ... the rest of the administration page ... -->

```

Impact

The impact of this vulnerability depends on the affected web application.

https://fowom.dialog.lk/pages/BOQ_create.php

Form action='/pages/BOQ_create.php'

Request

```

GET /pages/BOQ_create.php HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
Cookie: PHPSESSID=hshfh5adfb1rfh32hni8u80bb7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36

```

https://fowom.dialog.lk/pages/BOQ_edit.php

Form action='/pages/BOQ_edit.php'

Request

GET /pages/BOQ_edit.php HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
Cookie: PHPSESSID=hshfh5adfb1rfh32hni8u80bb7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36
Connection: Keep-alive

https://fowom.dialog.lk/pages/BOQ_search.php

Form action='/pages/BOQ_search.php'

Request

GET /pages/BOQ_search.php HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
Cookie: PHPSESSID=hshfh5adfb1rfh32hni8u80bb7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36
Connection: Keep-alive

https://fowom.dialog.lk/pages/incident_create.php

Form action='/pages/incident_create.php'

Request

GET /pages/incident_create.php HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
Cookie: PHPSESSID=b4ejcoptollp4tufu3j6eker22
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36

https://fowom.dialog.lk/pages/incident_search.php

Form action='/pages/incident_search.php'

Request

```
GET /pages/incident_search.php HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
Cookie: PHPSESSID=b4ejcoptollp4tufu3j6eker22
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36
Connection: Keep-alive
```

Recommendation

Make sure the script is terminated after redirecting the user to another page.

References

[HTML Form Found in Redirect Page Web Vulnerability](https://www.acunetix.com/blog/web-security-zone/html-form-found-in-redirect-page/)

<https://www.acunetix.com/blog/web-security-zone/html-form-found-in-redirect-page/>

HTTP Strict Transport Security (HSTS) not implemented

HTTP Strict Transport Security (HSTS) tells a browser that a web site is only accessible using HTTPS. It was detected that your web application doesn't implement HTTP Strict Transport Security (HSTS) as the Strict Transport Security header is missing from the response.

Impact

HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks

<https://fowom.dialog.lk/pages/>

Request

```
GET /pages/ HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
Cookie: PHPSESSID=hshfh5adfb1rfh32hni8u80bb7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36
Connection: Keep-alive
```

Recommendation

It's recommended to implement HTTP Strict Transport Security (HSTS) into your web application. Consult web references for more information

References

hstspreload.org

<https://hstspreload.org/>

[Strict-Transport-Security](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security)

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security>

Content Security Policy (CSP) not implemented

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

Content Security Policy (CSP) can be implemented by adding a **Content-Security-Policy** header. The value of this header is a string containing the policy directives describing your Content Security Policy. To implement CSP, you should define lists of allowed origins for the all of the types of resources that your site utilizes. For example, if you have a simple site that needs to load scripts, stylesheets, and images hosted locally, as well as from the jQuery library from their CDN, the CSP header could look like the following:

```
Content-Security-Policy:
  default-src 'self';
  script-src 'self' https://code.jquery.com;
```

It was detected that your web application doesn't implement Content Security Policy (CSP) as the CSP header is missing from the response. It's recommended to implement Content Security Policy (CSP) into your web application.

Impact

CSP can be used to prevent and/or mitigate attacks that involve content/code injection, such as cross-site scripting/XSS attacks, attacks that require embedding a malicious resource, attacks that involve malicious use of iframes, such as clickjacking attacks, and others.

<https://fowom.dialog.lk/pages/login.php>

Request

```
GET /pages/login.php HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36
Cookie: PHPSESSID=b4ejc0ptollp4tufu3j6eker22
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
Connection: Keep-alive
```

Recommendation

It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the **Content-Security-Policy** HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.

References

[Content Security Policy \(CSP\)](#)

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

[Implementing Content Security Policy](#)

<https://hacks.mozilla.org/2016/02/implementing-content-security-policy/>

Insecure Referrer Policy

Referrer Policy controls behaviour of the Referer header, which indicates the origin or web page URL the request was made from. The web application uses insecure Referrer Policy configuration that may leak user's information to third-party sites.

Impact

In some situations, an attacker may leak user's private data

<https://fowom.dialog.lk/>

URLs where Referrer Policy configuration is insecure:

- <https://fowom.dialog.lk/pages/>
- <https://fowom.dialog.lk/pages/img/>
- <https://fowom.dialog.lk/pages/config.php>
- <https://fowom.dialog.lk/pages/test.php>
- https://fowom.dialog.lk/pages/BOQ_create.php
- https://fowom.dialog.lk/pages/BOQ_edit.php
- https://fowom.dialog.lk/pages/BOQ_search.php
- <https://fowom.dialog.lk/>

Request

```
GET /pages/ HTTP/1.1
Referer: https://fowom.dialog.lk/pages/login.php
Cookie: PHPSESSID=hshfh5adfb1rfh32hni8u80bb7
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36
Connection: Keep-alive
```

Recommendation

Consider setting Referrer-Policy header to 'strict-origin-when-cross-origin' or a stricter value

References

[Referrer-Policy](#)

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy>

Password type input with auto-complete enabled

When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved. Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.

Impact

Possible sensitive information disclosure.

<https://fowom.dialog.lk/>

Pages with auto-complete password inputs:

- <https://fowom.dialog.lk/pages/login.php>

```
Form name: myForm
Form action: <empty>
Form method: POST
Password input: password
```

Request

```
POST /pages/login.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Referer: https://fowom.dialog.lk/pages/login.php
Cookie: PHPSESSID=b4ejcoptollp4tufu3j6eker22
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 51
Host: fowom.dialog.lk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/83.0.4103.61 Safari/537.36
Connection: Keep-alive
```

```
password=g00dPa%24%24w0rD&signin=&username=pHqghUme
```

Recommendation

The password auto-complete should be disabled in sensitive applications.

To disable auto-complete, you may use a code similar to:

```
<INPUT TYPE="password" AUTOCOMPLETE="off">
```

PHP Version Disclosure

The web server is sending the X-Powered-By: response headers, revealing the PHP version.

Impact

An attacker might use the disclosed information to harvest specific security vulnerabilities for the version identified.

<https://fowom.dialog.lk/>

Version detected: PHP/5.4.16.

Recommendation

Configure your web server to prevent information leakage from its HTTP response.

References

[PHP Documentation: header_remove\(\)](#)

<https://www.php.net/manual/en/function.header-remove.php>

[PHP Documentation: php.ini directive expose_php](#)

<https://www.php.net/manual/en/ini.core.php#ini.expose-php>

TLS 1.1 enabled

The web server supports encryption through TLS 1.1. When aiming for Payment Card Industry (PCI) Data Security Standard (DSS) compliance, it is recommended (although at the time of writing not required) to use TLS 1.2 or higher instead. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.

Impact

An attacker may be able to exploit this problem to conduct man-in-the-middle attacks and decrypt communications between the affected service and clients.

<https://fowom.dialog.lk/>

Confidence: 100%

The SSL server (port: 443) encrypts traffic using TLSv1.1.

Recommendation

It is recommended to disable TLS 1.1 and replace it with TLS 1.2 or higher.

References

[Are You Ready for 30 June 2018? Saying Goodbye to SSL/early TLS](https://blog.pcisecuritystandards.org/are-you-ready-for-30-june-2018-sayin-goodbye-to-ssl-early-tls)

<https://blog.pcisecuritystandards.org/are-you-ready-for-30-june-2018-sayin-goodbye-to-ssl-early-tls>

[PCI 3.1 and TLS 1.2 \(Cloudflare Support\)](https://support.cloudflare.com/hc/en-us/articles/205043158-PCI-3-1-and-TLS-1-2)

<https://support.cloudflare.com/hc/en-us/articles/205043158-PCI-3-1-and-TLS-1-2>

Coverage

<https://fowom.dialog.lk/>

<https://fowom.dialog.lk/pages/>

https://fowom.dialog.lk/pages/BOQ_create.php

https://fowom.dialog.lk/pages/BOQ_edit.php

https://fowom.dialog.lk/pages/BOQ_search.php

https://fowom.dialog.lk/pages/incident_create.php

https://fowom.dialog.lk/pages/incident_search.php

<https://fowom.dialog.lk/pages/login.php>