netsparker

4/26/2025 8:15:13 PM (UTC+05:30)

Detailed Scan Report

https://winlankahospitals.com/

Scan Time	: 4/26/2025 8:11:05 PM (UTC+05:30)
Scan Duration	: 00:00:04:04
Total Requests	: 1,080
Average Speed	: 4.4 r/s

Risk Level:
MEDIUM

12 IDENTIFIED	**5** CONFIRMED	**0** CRITICAL
0 HIGH	2 MEDIUM	3 LOW
	5 BEST PRACTICE	2 INFORMATION

Identified Vulnerabilities

Critical	0
High	0
Medium	2
Low	3
Best Practice	5
Information	2
TOTAL	**12**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	1
Low	1
Best Practice	1
Information	2
TOTAL	**5**

Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://winlankahospitals.com/	
	Weak Ciphers Enabled	GET	https://winlankahospitals.com/	
	Missing Content-Type Header	HEAD	https://winlankahospitals.com/opensearch	
	Missing X-Frame-Options Header	GET	https://winlankahospitals.com/	
	Insecure Transportation Security Protocol Supported (TLS 1.0)	GET	https://winlankahospitals.com/	
	Content Security Policy (CSP) Not Implemented	GET	https://winlankahospitals.com/	
	Expect-CT Not Enabled	GET	https://winlankahospitals.com/	
	Missing X-XSS-Protection Header	GET	https://winlankahospitals.com/	
	Referrer-Policy Not Implemented	GET	https://winlankahospitals.com/	
	Insecure Transportation Security Protocol Supported (TLS 1.1)	GET	https://winlankahospitals.com/	
	Forbidden Resource	GET	https://winlankahospitals.com/opensearch.xml	
	Robots.txt Detected	GET	https://winlankahospitals.com/robots.txt	

1. HTTP Strict Transport Security (HSTS) Policy Not Enabled

MEDIUM | 1

Netsparker identified that HTTP Strict Transport Security (HSTS) policy is not enabled.

The target website is being served from not only HTTPS but also HTTP and it lacks of HSTS policy implementation.

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure (HTTPS) connections. The HSTS Policy is communicated by the server to the user agent via a HTTP response header field named "Strict-Transport-Security". HSTS Policy specifies a period of time during which the user agent shall access the server in only secure fashion.

When a web application issues HSTS Policy to user agents, conformant user agents behave as follows:

- Automatically turn any insecure (HTTP) links referencing the web application into secure (HTTPS) links. (For instance, http://example.com/some/page/ will be modified to https://example.com/some/page/ before accessing the server.)
- If the security of the connection cannot be ensured (e.g. the server's TLS certificate is self-signed), user agents show an error message and do not allow the user to access the web application.

Vulnerabilities

1.1. https://winlankahospitals.com/

Certainty

Remedy

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers_module modules/mod_headers.so

# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
       ServerAlias *
       RewriteEngine On
       RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

# HTTPS-Host-Configuration
<VirtualHost *:443>
      # Use HTTP Strict Transport Security to force client to use secure connections only
      Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

```
    # Further Configuration goes here
    [...]
</VirtualHost>
```

External References

- Wikipedia - HTTP Strict Transport Security
- Configure HSTS (HTTP Strict Transport Security) for Apache/Nginx
- HTTP Strict Transport Security (HSTS) HTTP Header
- Mozilla SSL Configuration Generator

CLASSIFICATION

OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	523
CAPEC	217
WASC	4
ISO27001	A.14.1.2

2. Weak Ciphers Enabled

MEDIUM 1 CONFIRMED 1

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

Vulnerabilities

2.1. https://winlankahospitals.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)

Actions to Take

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

   a. Click Start, click Run, type `regedt32` or type `regedit`, and then click OK.
   b. In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
   c. Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
```

```
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Remedy

Configure your web server to disallow using weak ciphers.

External References

- OWASP - Insecure Configuration Management
- OWASP Top 10-2017 A3-Sensitive Data Exposure
- Zombie Poodle - Golden Doodle (CBC)
- Mozilla SSL Configuration Generator
- Strong Ciphers for Apache, Nginx and Lighttpd

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	327
CAPEC	217
WASC	4
ISO27001	A.14.1.3

CVSS 3.0 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String

CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

CVSS 3.1 SCORE

Base	6.8 (Medium)
Temporal	6.8 (Medium)
Environmental	6.8 (Medium)

CVSS Vector String
CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

3. Insecure Transportation Security Protocol Supported (TLS 1.0)

LOW 1 CONFIRMED 1

Netsparker detected that insecure transportation security protocol (TLS 1.0) is supported by your web server.

TLS 1.0 has several flaws. An attacker can cause connection failures and they can trigger the use of TLS 1.0 to exploit vulnerabilities like BEAST (Browser Exploit Against SSL/TLS).

Websites using TLS 1.0 are considered non-compliant by PCI since 30 June 2018.

Impact

Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.

Vulnerabilities

3.1. https://winlankahospitals.com/

CONFIRMED

Actions to Take

We recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher. See Remedy section for more details.

Remedy

Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod_ssl module. This directive can be set either at the server level or in a virtual host configuration.

```
SSLProtocol +TLSv1.2
```

- For Nginx, locate any use of the directive ssl_protocols in the `nginx.conf` file and remove `TLSv1`.

```
ssl_protocols TLSv1.2;
```

- For Microsoft IIS, you should make some changes on the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**
    1. Click on Start and then Run, type `regedt32` or `regedit`, and then click OK.
    2. In Registry Editor, locate the following registry key or create if it does not exist:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\
```

3. Locate a key named `Server` or create if it doesn't exist.
4. Under the `Server` key, locate a DWORD value named `Enabled` or create if it doesn't exist and set its value to "0".

- For lighttpd, put the following lines in your configuration file:

```
ssl.use-sslv2 = "disable"
ssl.use-sslv3 = "disable"
ssl.openssl.ssl-conf-cmd = ("Protocol" => "-TLSv1.1, -TLSv1, -SSLv3") # v1.4.48 or up
ssl.ec-curve = "secp384r1"
```

External References

- How to Disable TLS v1.0
- OWASP - Insecure Configuration Management
- OWASP Top 10 - 2017 A3 - Sensitive Data Exposure
- How to disable PCT 1.0, SSL 2.0, SSL 3.0, or TLS 1.0 in Internet Information Services
- IIS Crypto is a free tool that gives administrators the ability to enable or disable protocols, ciphers, hashes and key exchange algorithms on Windows Server 2003, 2008 and 2012
- Date Change for Migrating from SSL and Early TLS
- Browser Exploit Against SSL/TLS Attack (BEAST)
- Are You Ready for 30 June 2018? Saying Goodbye to SSL/early TLS

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	326
CAPEC	217
WASC	4
HIPAA	164.306
ISO27001	A.14.1.3

4. Missing Content-Type Header

LOW | 1

Netsparker detected a missing `Content-Type` header which means that this website could be at risk of a MIME-sniffing attacks.

Impact

MIME type sniffing is a standard functionality in browsers to find an appropriate way to render data where the HTTP headers sent by the server are either inconclusive or missing.

This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the intended content type.

The problem arises once a website allows users to upload content which is then published on the web server. If an attacker can carry out XSS (Cross-site Scripting) attack by manipulating the content in a way to be accepted by the web application and rendered as HTML by the browser, it is possible to inject code in e.g. an image file and make the victim execute it by viewing the image.

Vulnerabilities

4.1. https://winlankahospitals.com/opensearch

Certainty

Remedy

1. When serving resources, make sure you send the content-type header to appropriately match the type of the resource being served. For example, if you are serving an HTML page, you should send the HTTP header:

```
Content-Type: text/html
```

2. Add the X-Content-Type-Options header with a value of "nosniff" to inform the browser to trust what the site has sent is the appropriate content-type, and to not attempt "sniffing" the real content-type.

```
X-Content-Type-Options: nosniff
```

External References

- MIME Sniffing: feature or vulnerability?
- X-Content-Type-Options HTTP Header

CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
SANS Top 25	16
WASC	15
ISO27001	A.14.1.2

5. Missing X-Frame-Options Header

LOW | 1

Netsparker detected a missing `X-Frame-Options` header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options` HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a `frame` or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Vulnerabilities

5.1. https://winlankahospitals.com/

Certainty

Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
  - `X-Frame-Options: DENY` It completely denies to be loaded in frame/iframe.
  - `X-Frame-Options: SAMEORIGIN` It allows only if the site which wants to load has a same origin.
  - `X-Frame-Options: ALLOW-FROM` *`URL`* It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

External References

- Clickjacking
- Can I Use X-Frame-Options
- X-Frame-Options HTTP Header

Remedy References

- Clickjacking Defense Cheat Sheet

CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
SANS Top 25	693
CAPEC	103
ISO27001	A.14.2.5

6. Content Security Policy (CSP) Not Implemented

BEST PRACTICE | 1

CSP is an added layer of security that helps to mitigate mainly Cross-site Scripting attacks.

CSP can be enabled instructing the browser with a Content-Security-Policy directive in a response header;

```
Content-Security-Policy: script-src 'self';
```

or in a meta tag;

```
<meta http-equiv="Content-Security-Policy" content="script-src 'self';">
```

In the above example, you can restrict script loading only to the same domain. It will also restrict inline script executions both in the element attributes and the event handlers. There are various directives which you can use by declaring CSP:

- **script-src:** Restricts the script loading resources to the ones you declared. By default, it disables inline script executions unless you permit to the evaluation functions and inline scripts by the unsafe-eval and unsafe-inline keywords.
- **base-uri:** Base element is used to resolve relative URL to absolute one. By using this CSP directive, you can define all possible URLs which could be assigned to base-href attribute of the document.
- **frame-ancestors**: It is very similar to X-Frame-Options HTTP header. It defines the URLs by which the page can be loaded in an iframe.
- **frame-src / child-src**: frame-src is the deprecated version of child-src. Both define the sources that can be loaded by iframe in the page. (Please note that frame-src was brought back in CSP 3)
- **object-src** : Defines the resources that can be loaded by embedding such as Flash files, Java Applets.
- **img-src**: As its name implies, it defines the resources where the images can be loaded from.
- **connect-src**: Defines the whitelisted targets for XMLHttpRequest and WebSocket objects.
- **default-src**: It is a fallback for the directives that mostly ends with -src suffix. When the directives below are not defined, the value set to default-src will be used instead:
    - child-src
    - connect-src
    - font-src
    - img-src
    - manifest-src
    - media-src
    - object-src
    - script-src
    - style-src

When setting the CSP directives, you can also use some CSP keywords:

- **none**: Denies loading resources from anywhere.
- **self** : Points to the document's URL (domain + port).
- **unsafe-inline**: Permits running inline scripts.
- **unsafe-eval**: Permits execution of evaluation functions such as eval().

In addition to CSP keywords, you can also use wildcard or only a scheme when defining whitelist URLs for the points. Wildcard can be used for subdomain and port portions of the URLs:

```
Content-Security-Policy: script-src https://*.example.com;
Content-Security-Policy: script-src https://example.com:*;
Content-Security-Policy: script-src https:;
```

It is also possible to set a CSP in Report-Only mode instead of forcing it immediately in the migration period. Thus you can see the violations of the CSP policy in the current state of your web site while migrating to CSP:

```
Content-Security-Policy-Report-Only: script-src 'self'; report-uri: https://example.com;
```

Impact

There is no direct impact of not implementing CSP on your website. However, if your website is vulnerable to a Cross-site Scripting attack CSP can prevent successful exploitation of that vulnerability. By not implementing CSP you'll be missing out this extra layer of security.

Vulnerabilities

6.1. https://winlankahospitals.com/

Certainty

Actions to Take

- Enable CSP on your website by sending the `Content-Security-Policy` in HTTP response headers that instruct the browser to apply the policies you specified.
- Apply the whitelist and policies as strict as possible.
- Rescan your application to see if Netsparker identifies any weaknesses in your policies.

Remedy

Enable CSP on your website by sending the `Content-Security-Policy` in HTTP response headers that instruct the browser to apply the policies you specified.

External References

- An Introduction to Content Security Policy
- Content Security Policy (CSP) HTTP Header
- Content Security Policy (CSP)

CLASSIFICATION

SANS Top 25	16
WASC	15
ISO27001	A.14.2.5

7. Expect-CT Not Enabled

BEST PRACTICE | 1

Netsparker identified that Expect-CT is not enabled.

Certificate Transparency is a technology that makes impossible (or at least very difficult) for a CA to issue an SSL certificate for a domain without the certificate being visible to the owner of that domain.

Google announced that, starting with April 2018, if it runs into a certificate that is not seen in Certificate Transparency (CT) Log, it will consider that certificate invalid and reject the connection. Thus sites should serve certificate that takes place in CT Logs. While handshaking, sites should serve a valid Signed Certificate Timestamp (SCT) along with the certificate itself.

Expect-CT can also be used for detecting the compatibility of the certificates that are issued before the April 2018 deadline. For instance, a certificate that was signed before April 2018, for 10 years it will be still posing a risk and can be ignored by the certificate transparency policy of the browser. By setting Expect-CT header, you can prevent misisssued certificates to be used.

Vulnerabilities

7.1. https://winlankahospitals.com/

Certainty

Remedy

Configure your web server to respond with Expect-CT header.

```
Expect-CT: enforce, max-age=7776000, report-uri="https://ABSOLUTE_REPORT_URL"
```

Note: We strongly suggest you to use Expect-CT header in **report-only mode** first. If everything goes well and your certificate is ready, go with the Expect-CT enforce mode. To use **report-only mode** first, omit **enforce** flag and see the browser's behavior with your deployed certificate.

```
Expect-CT: max-age=7776000, report-uri="https://ABSOLUTE_REPORT_URL"
```

External References

- Expect-CT Extension for HTTP
- Expect-CT HTTP Header
- Expect-CT Header

CLASSIFICATION

SANS Top 25	16
WASC	15
ISO27001	A.14.1.2

8. Insecure Transportation Security Protocol Supported (TLS 1.1)

BEST PRACTICE | 1 | CONFIRMED | 1

Netsparker detected that a deprecated, insecure transportation security protocol (TLS 1.1) is supported by your web server.

TLS 1.1 will be considered as deprecated by major web browsers (i.e. Chrome, Firefox, Safari, Edge, Internet Explorer) starting in 2020.

Impact

Your website will be inaccessible due to web browser deprecation.

Vulnerabilities

8.1. https://winlankahospitals.com/

CONFIRMED

Actions to Take

We recommended to disable TLS 1.1 and replace it with TLS 1.2 or higher. See Remedy section for more details.

Remedy

Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod_ssl module. This directive can be set either at the server level or in a virtual host configuration.

```
SSLProtocol +TLSv1.2
```

- For Nginx, locate any use of the directive ssl_protocols in the `nginx.conf` file and remove `TLSv1.1`.

```
ssl_protocols TLSv1.2;
```

- For Microsoft IIS, you should make some changes on the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**
    1. Click on Start and then Run, type `regedt32` or `regedit`, and then click OK.
    2. In Registry Editor, locate the following registry key or create if it does not exist:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\
```

3. Locate a key named `Server` or create if it doesn't exist.
4. Under the `Server` key, locate a DWORD value named `Enabled` or create if it doesn't exist and set its value to "0".

- For lighttpd, put the following lines in your configuration file:

```
ssl.use-sslv2 = "disable"
ssl.use-sslv3 = "disable"
ssl.openssl.ssl-conf-cmd = ("Protocol" => "-TLSv1.1, -TLSv1, -SSLv3") # v1.4.48 or up
ssl.ec-curve = "secp384r1"
```

External References

- Deprecating TLSv1.0 and TLSv1.1 draft-ietf-tls-oldversions-deprecate-00
- Google Security Blog: Modernizing Transport Security
- OWASP - Insecure Configuration Management
- OWASP Top 10 - 2017 A3 - Sensitive Data Exposure
- IIS Crypto is a free tool that gives administrators the ability to enable or disable protocols, ciphers, hashes and key exchange algorithms on Windows Server 2003, 2008 and 2012
- Date Change for Migrating from SSL and Early TLS

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	326
CAPEC	217
WASC	4
HIPAA	164.306
ISO27001	A.14.1.3

9. Missing X-XSS-Protection Header

BEST PRACTICE | 1

Netsparker detected a missing `X-XSS-Protection` header which means that this website could be at risk of a Cross-site Scripting (XSS) attacks.

Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

Vulnerabilities

9.1. https://winlankahospitals.com/

Certainty

Remedy

Add the X-XSS-Protection header with a value of "1; mode= block".

- ```
  X-XSS-Protection: 1; mode=block
  ```

External References

- Internet Explorer 8 Security Features - MSDN
- X-XSS-Protection HTTP Header
- Internet Explorer 8 XSS Filter

CLASSIFICATION

SANS Top 25	16
WASC	15
HIPAA	164.308(A)
ISO27001	A.14.2.5
  ```

10. Referrer-Policy Not Implemented

BEST PRACTICE | 1

Netsparker detected that no Referrer-Policy header implemented.

Referrer-Policy is a security header designed to prevent cross-domain Referer leakage.

Impact

Referer header is a request header that indicates the site which the traffic originated from. If there is no adequate prevention in place, the URL itself, and even sensitive information contained in the URL will be leaked to the cross-site.

The lack of Referrer-Policy header might affect privacy of the users and site's itself

Vulnerabilities

10.1. https://winlankahospitals.com/

Certainty

Actions to Take

In a response header:

```
Referrer-Policy: no-referrer | same-origin | origin | strict-origin | no-origin-when-downgrading
```

In a META tag

```
<meta name="Referrer-Policy" value="no-referrer | same-origin"/>
```

In an element attribute

```
<a href="http://crosssite.example.com" rel="noreferrer"></a>
```

or

```
<a href="http://crosssite.example.com" referrerpolicy="no-referrer | same-origin | origin | strict-
origin | no-origin-when-downgrading"></a>
```

Remedy

Please implement a Referrer-Policy by using the Referrer-Policy response header or by declaring it in the meta tags. It's also possible to control referrer information over an HTML-element by using the rel attribute.

External References

- Referrer Policy
- Referrer Policy - MDN
- Referrer Policy HTTP Header
- A New Security Header: Referrer Policy
- Can I Use Referrer-Policy

CLASSIFICATION

OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	200
ISO27001	A.14.2.5

11. Forbidden Resource

INFORMATION 1 CONFIRMED 1

Netsparker identified a forbidden resource.

Access to this resource has been denied by the web server. This is generally not a security issue, and is reported here for informational purposes.

Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

Vulnerabilities

11.1. https://winlankahospitals.com/opensearch.xml

CONFIRMED

CLASSIFICATION

OWASP Proactive Controls	C8
ISO27001	A.8.1.1

12. Robots.txt Detected

INFORMATION 1 | CONFIRMED 1

Netsparker detected a `Robots.txt` file with potentially sensitive content.

Impact

Depending on the content of the file, an attacker might discover hidden directories and files.

Vulnerabilities

12.1. https://winlankahospitals.com/robots.txt

CONFIRMED

Interesting Robots.txt Entries

- Disallow: /wp-admin/
- Sitemap: https://winlankahospitals.com/sitemap_index.xml

Remedy

Ensure you have nothing sensitive exposed within this file, such as the path of an administration panel. If disallowed paths are sensitive and you want to keep it from unauthorized access, do not write them in the `Robots.txt`, and ensure they are correctly protected by means of authentication.
`Robots.txt` is only used to instruct search robots which resources should be indexed and which ones are not.

The following block can be used to tell the crawler to index files under /web/ and **ignore the rest**:

```
User-Agent: *
Allow: /web/
Disallow: /
```

Please note that when you use the instructions above, **search engines will not index your website** except for the specified directories.

If you want to hide certain section of the website from the search engines `X-Robots-Tag` can be set in the response header to tell crawlers whether the file should be indexed or not:

```
X-Robots-Tag: googlebot: nofollow
X-Robots-Tag: otherbot: noindex, nofollow
```

By using `X-Robots-Tag` you don't have to list the these files in your `Robots.txt`.

It is also not possible to prevent media files from being indexed by putting using Robots Meta Tags. `X-Robots-Tag` resolves this issue as well.

For Apache, the following snippet can be put into `httpd.conf` or an `.htaccess` file to restrict crawlers to index multimedia files without exposing them in `Robots.txt`

```
<Files ~ "\.pdf$">
 # Don't index PDF files.
 Header set X-Robots-Tag "noindex, nofollow"
</Files>
```

```
<Files ~ "\.(png|jpe?g|gif)$">
 #Don't index image files.
 Header set X-Robots-Tag "noindex"
</Files>
```

External References

- What Content Is Not Crawled? - Google
- How Search organizes information
- X-Robots-Tag: A Simple Alternate For Robots .txt and Meta Tag

CLASSIFICATION

OWASP Proactive Controls	C7
ISO27001	A.18.1.3

Show Scan Detail

Enabled Security Checks : Apache Struts S2-045 RCE,
Apache Struts S2-046 RCE,
BREACH Attack,
Code Evaluation,
Code Evaluation (Out of Band),
Command Injection,
Command Injection (Blind),
Content Security Policy,
Content-Type Sniffing,
Cookie,
Cross Frame Options Security,

Cross-Origin Resource Sharing (CORS),
Cross-Site Request Forgery,
Cross-site Scripting,
Cross-site Scripting (Blind),
Custom Script Checks (Active),
Custom Script Checks (Passive),
Custom Script Checks (Per Directory),
Custom Script Checks (Singular),
Drupal Remote Code Execution,
Expect Certificate Transparency (Expect-CT),
Expression Language Injection,
File Upload,
Header Analyzer,
Heartbleed,
HSTS,
HTML Content,
HTTP Header Injection,
HTTP Methods,
HTTP Status,
HTTP.sys (CVE-2015-1635),
IFrame Security,
Insecure JSONP Endpoint,
Insecure Reflected Content,
JavaScript Libraries,
Local File Inclusion,
Login Page Identifier,
Mixed Content,
Open Redirection,
Referrer Policy,
Reflected File Download,
Remote File Inclusion,
Remote File Inclusion (Out of Band),
Reverse Proxy Detection,
RoR Code Execution,
Server-Side Request Forgery (DNS),
Server-Side Request Forgery (Pattern Based),
Server-Side Template Injection,
Signatures,
SQL Injection (Blind),
SQL Injection (Boolean),
SQL Injection (Error Based),
SQL Injection (Out of Band),
SSL,
Static Resources (All Paths),
Static Resources (Only Root Path),
Unicode Transformation (Best-Fit Mapping),
WAF Identifier,
Web App Fingerprint,
Web Cache Deception,
WebDAV,
Windows Short Filename,
XML External Entity,
XML External Entity (Out of Band)

URL Rewrite Mode	: Heuristic
Detected URL Rewrite Rule(s)	: None
Excluded URL Patterns	: (log|sign)\-?(out|off) exit endsession gtm\.js WebResource\.axd ScriptResource\.axd
Authentication	: None
Scheduled	: No
Additional Website(s)	: https://www.winlankahospitals.com/

This report created with 5.8.1.28119-master-bca4e4e
https://www.netsparker.com