What are HTTP Headers?

HTTP headers are the names and values for request and response messages for Hypertext Transport Protocol. Locking down or implementing protection HTTP headers will minimize risk of compromise and exfiltration of data or server configuration information.

While this article goes into IIS HTTP Header configuration these HTTP headers can be applied to other webservers like Apacheor Nginx.

How do I modify HTTP headers in IIS?

Log into your Windows Webserver

Open the IIS Manager

Click on the server or the site you want to update the HTTP headers on

Click HTTP Response Headers

This isn't the only way but is an easier way for beginners who are used to using an interface.

There are two options: lock down headers at the IIS server level or website level. Which is better and why?

Locking down HTTP headers from the IIS server level enforces all websites under the IIS server inherit these HTTP headers.

Good: Means consistent HTTP headers for all sites

Bad: mistakes mean all sites can be unprotected, not versatile if sites need different HTTP headers

Locking down headers from a single site under IIS

Good: versatile and flexible for every site

Bad: more maintenance and time consuming, inconsistent HTTP headers can lead to an unprotected site

Changing any of these HTTP Headers will require a restart of IIS

Access-Control-Allow-Origin

Allows websites to share content between each other with the given origin

Name	Value	Meaning
Access-Control-Allow-Origin	URI_Value	Only allow a specific domain name
Access-Control-Allow-Origin	*	Allows all which is very insecure

X-XSS-Protection

Used to prevent Cross-Site Scripting (XSS)

Name	Value	Meaning
X-XSS-Protection	0	Disables XSS filtering
X-XSS-Protection	1	Enables XSS filtering
X-XSS-Protection	1; mode=block	Enables XSS Filtering, Prevents page loading
X-XSS-Protection	1; report=<reporting-uri>	Enables XSS filtering and reporting to a specified URI

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

X-Frame-Options

Used to prevent Clickjacking vulnerability

Name	Value	Meaning
X-Frame-Options	SAMEORIGIN	Frame/iframe of content is allowed from originating site
X-Frame-Options	DENY	Prevent any domain to embed content into frame/iframe
X-Frame-Options	ALLOW-FROM	Allow embedded content from a specific URI

Recommended:X-Frame-Options SAMEORIGIN

X-Content-Type-Options

Disallows content sniffing

Name	Value	Meaning
X-Content-Type-Options	nosniff	disallows content sniffing

Recommended:X-Content-Type-Options nosniff

HTTP Public Key Pinning (Not recommended)

Minimize man in the middle attacks by pinning the certificate to an HTTP header

This is not widely supported by all browsers and if not maintained could prevent websites from loading properly.

Name	Value	Meaning
Public-Key-Pins	pin-sha256=”pin value”	Base64 encoded Subject Public Key Information Fingerprint
Public-Key-Pins	max-age=<expire-time>	Time in seconds that the browser should remember the site

Public-Key-Pins	includeSubDomains	optional parameter for to apply to all subdomains
Public-Key-Pins	report-uri="<URI>"	If failure report to URL

Content-Security-Policy

Prevent XSS, Click Jacking and code injection

Name	Value	Meaning
Content-Security-Policy	default-src ‘self’	Loads all resource types (Catch all)
Content-Security-Policy	script-src ‘self’	Load only scripts from defined source
Content-Security-Policy	style-src ‘self’	Load only CSS from defined source
Content-Security-Policy	img-src ‘self’	Image Loading strategy
Content-Security-Policy	content-src ‘self’	Defines XHR, Ajax, Websockets etc. load strategy
Content-Security-Policy	font-src ‘self’	Defines font loading policy
Content-Security-Policy	object-src ‘self’	Defines flash loading policy
Content-Security-Policy	media-src ‘self’	Defines media loading policy

Content-Security-Policy	frame-src ‘self’	Defines frame/iframe loading policy
Content-Security-Policy	sandbox ‘allow-forms’	Defines sandboxing resource load policy

Arguments that can be used in values:

Value Meaning

* Allow any content

None No content is allowed

Self Run homogeneous content

Data running the data: protocol (Base64 image)

www.domainname.org allows loading from a specified domain

123.123.123.123 allows loading from a specified IP address

*.domainname.org allows loading from subdomains

|https://domainname.org Allows loading of the specified domain

https: Allows loading https resources

unsafe-inline Allows loading of inline assets

unsafe-eval Allows dynamic loading of JS

HTTP Header Example:

Content-Security-Policy default-src ‘self’

Content-Security-Policy script-src ‘self’ ‘unsafe-inline’ ‘unsafe=eval’ *.google-analystics.com;

Recommended: Content-Security-Policy default-src ‘self’

Strict-Transport-Security

Enforces the use of HTTPS by the browser client but is ignored if the site being accessed is HTTP.

Name	Value	Meaning
Strict-Transport-Security	max-age=<expire-time>	Time in seconds the browser should remember the site

Recommended: Strict-Transport-Security max-age=31536000; includeSubDomains;Preload

Referrer-Policy

Determines which referrer information sent in the Referrer header should be included with requests made.

Name	Value	Meaning
Referrer-Policy	no-referrer	referrer will be omitted
Referrer-Policy	no-referrer-when-downgrade	default; The URL is sent as a referrer when the protocol security level stays the same but isn't sent to a less secure destination
Referrer-Policy	origin	Only send the origin of the document
Referrer-Policy	origin-when-cross-origin	Send a full URL when performing a same-origin request, but only send the origin of the document for other cases
Referrer-Policy	same-origin	A referrer will be sent for same-site origins, but cross-origin requests

		will contain no referrer information
Referrer-Policy	strict-origin	Only send the origin of the document as the referrer when the protocol security level stays the same, but don't send to less secure destination
Referrer-Policy	strict-origin-when-cross-origin	Send full URL when performing a same-origin request, only send the origin when the protocol security level stays the same and send no header to a less secure destination
Referrer-Policy	unsafe-url	send full URL when performing a same-origin or cross-origin request.

Content Source

https://geekflare.com/http-header-implementation/

https://scotthelme.co.uk/hardening-your-http-response-headers/

https://securityonline.info/configure-security-http-headers-prevent-vulnerabilities/

https://securityonline.info/csp-content-security-policycontent-security-policy-header/

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

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

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Allow-Origin

Additional Tools

Scan Website Security Headers:

https://securityheaders.com/

OWASP Zap Tool

https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

Fiddler Web Debugging Tool

https://www.telerik.com/fiddler

This certainly isn't all of the HTTP headers we as security professionals should be aware of it but is a fairly detailed list. I will continue to update this resource as new HTTP header information is posted or released