DigiNinja

Something about Security

Headers and Cookies

https://digi.ninja

Who Am I?

Robin Wood

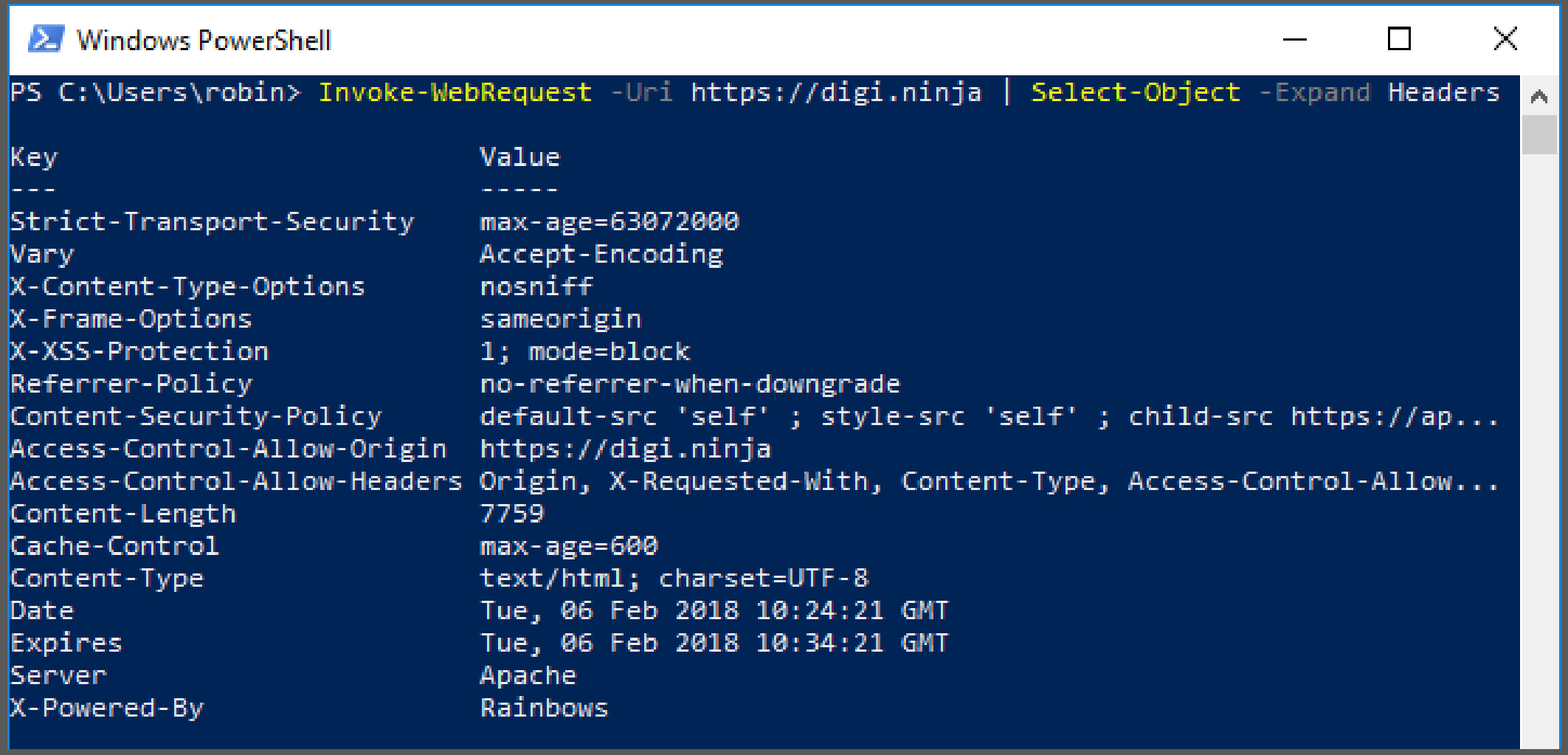
https://digi.ninja

@digininja

Background

- Started work as desktop app developer in 1996
- Moved to web app in 2003
- Moved to security testing in 2009
- Freelance tester and consultant
- Still do bits of web dev on the side
- Published over 50 security tools

HTTP Headers



https://digi.ninja

Main Headers

- X-Content-Type-Options
- X-Frame-Options
- X-XSS-Protection
- Referrer-Policy
- Strict-Transport-Security
- Content-Security-Policy
- Public-Key-Pins
- Expect-CT

X-Content-Type-Options

Prevents a browser from trying to guess the file type of content

Protects against download attacks where browser makes bad choices

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

X-Content-Type-Options

No header – browser can sniff

One options:

- nosniff – honour the type specified*

* Recommended

https://digi.ninja

X-Content-Type-Options

Example header:

x-content-type-options: nosniff

X-Frame-Options

Specifies how a site can be (or cannot be) used in frames and iframes

Protects against Clickjacking*

Demo https://vuln-demo.com/clickjack/

* https://www.owasp.org/index.php/Clickjacking

X-Frame-Options

No header – any site can frame this one

Three options:

- ALLOW-FROM – specify domains which can frame this one
- SAMEORIGIN – this site can frame itself
- DENY – nothing can frame this site*

* Recommended

https://digi.ninja

X-Frame-Options

Example headers:

x-frame-options: sameorigin

x-frame-options: deny

X-XSS-Protection

Enable or disable a browsers built in Cross-Site Scripting protections

Affects Chrome and IE/Edge

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

X-XSS-Protection

No header – default browser behaviour

Four options:

- 0 – disable protections
- 1 – enable protections and sanitize output
- 1; report=<reporting-uri>*
- 1; mode=block – enable protections and block malicious content*

* Recommended

https://digi.ninja

X-XSS-Protection

Example headers:

x-xss-protection: 0

x-xss-protection: 1; mode=block

Referrer-Policy

Newest header on the block

Specifies when a browser should pass a referer header

Useful when you have sensitive data in querystrings

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

Referrer-Policy

No header – default browser behaviour, usually just pass the header

Eight options:

- no-referrer
- no-referrer-when-downgrade
- origin
- origin-when-cross-origin
- same-origin
- strict-origin
- strict-origin-when-cross-origin*
- unsafe-url

https://www.w3.org/TR/referrer-policy/

* Recommended

https://digi.ninja

Referrer-Policy

Example headers:

referrer-policy: strict-origin-when-cross-origin

referrer-policy: origin

Referrer-Policy

Can break tracking/logging software

Obviously breaks referrer programs if not done right

Strict-Transport-Security

Also known as HSTS

Enforces HTTPS on all requests

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/strict-transport-security

Strict-Transport-Security

No header – traffic can use HTTP or HTTPS

Three options:

- max-age=<expire-time>
- max-age=<expire-time>; includeSubDomains*
- max-age=<expire-time>; preload

* Recommended

https://digi.ninja

Strict-Transport-Security

Example headers:

strict-transport-security: max-age=31536000; includeSubDomains

strict-transport-security: max-age=0;

Strict-Transport-Security

Site still vulnerable on first load

Can be mitigated with preloading

Submit at https://hstspreload.org/

https://src.chromium.org/viewvc/chrome/trunk/src/net/http/transport_security_state_static.json

Content-Security-Policy

Hardest one on the list to implement

Locks down how and what resources can be used by a site by use of whitelisting

Two modes, enabled and report only

Content-Security-Policy

Mozilla scrapes Google's list for Firefox
https://blog.mozilla.org/security/2012/11/01/preloading-hsts/

Mozilla's Guide
https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

Publishers Guide
https://content-security-policy.com/

Content-Security-Policy

Example header:

Content-Security-Policy: default-src https:

Content-Security-Policy: default-src site.com

Content-Security-Policy: default-src https://site.com

Content-Security-Policy

Example header:

Content-Security-Policy: default-src ‘none’; script-src https://site.com; style-src https://site.com https://image.site.com

Content-Security-Policy

child-src

connect-src

default-src

font-src

form-action

frame-ancestors

frame-src*

img-src

media-src

object-src

plugin-types

report-uri

sandbox

script-src

style-src

* frame-src is deprecated, use child-src

https://digi.ninja

Cont Policy

New compatibility tables are in beta

	Desktop						Mobile						
	Chrome	Edge	Firefox	IE	Opera	Safari	Android	Chrome for Android	Edge Mobile	Firefox for Android	Opera Mobile	Safari Mobile	Samsung Internet
Content-Security-Polic	25 *	14	23 *	10 *	15	7 *	Yes	Yes	Yes	23	?	7.1 *	?
base-uri	40	No	35	No	27	10	Yes	Yes	No	35	?	9.3	?
block-all-mixed-conten	Yes	?	48	No	Yes	?	Yes	Yes	?	48	?	?	?
child-src	40	15	45	No	27	10	Yes	Yes	No	45	?	9.3	?
connect-src	25	14	23 *	No	15	7	Yes	Yes	?	23	?	7.1	?
default-src	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
disown-opener	No	No	No	No	No	No	No	No	No	No	No	No	?
font-src	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
form-action	40	15	36	No	27	10	Yes	Yes	No	36	?	9.3	?
frame-ancestors	40	15	33 *	No	26	10	?	Yes	No	33 *	?	9.3	?
frame-src	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
img-src	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
manifest-src	Yes	No	41	No	Yes	No	Yes	Yes	No	41	?	No	?
media-src	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
navigation-to	No	No	No	No	No	No	No	No	No	No	No	No	?
object-src	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
plugin-types	40	15	No *	No	27	10	Yes	Yes	No	No	?	9.3	?
referrer	33 — 56	No	37 *	No	? — 43	No	33 — 56	33 — 56	No	37 *	? — 43	No	?
report-sample	59	?	?	?	46	?	59	59	?	?	46	?	?
report-to	No	No	No	No	No	No	No	No	No	No	No	No	?
report-uri	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
require-sri-for	54	No	49	No	41	No	54	54	No	49	41	No	?
sandbox	25	14	50	10	15	7	Yes	Yes	?	50	?	7.1	?
script-src	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
strict-dynamic	52	No	52	No	39	No	52	52	No	No	39	No	?
style-src	25	14	23	No	15	7	Yes	Yes	?	23	?	7.1	?
upgrade-insecure-reque	43	No *	42	No	30	No	43	43	No	42	30	No	?
worker-src	59 *	No	58	No	48	No	59 *	59 *	No	58	48	No	?

- Full support
- No support
- Compatibility unknown
- Experimental. Expect behavior to change in the future.
- Non-standard. Expect poor cross-browser support.
- Deprecated. Not for use in new websites.
- * See implementation notes.
- User must explicitly enable this feature.

https://digi.ninja

Content-Security-Policy

Reporting of failures can be done by adding the following to the header

report-uri https://report-uri.com

e.g.

Content-Security-Policy: default-src https://site.com; report-uri https://report-uri.com/

Content-Security-Policy

Strongly recommend setting up an account with Report URI and sending reports there

https://report-uri.com/

Just remember to monitor them!

Content-Security-Policy

View 100 records

Filter

Action	Date	URI	Directive	Blocked URI	Raw	Count
All	Hours 07/02/2018 11	hostname path	All	blocked hostname blocked path		All
Enforced	07 Feb 2018 11:17:45	https://digi.ninja/projects_metasploit.php	font-src	data	show/hide	2
Enforced	07 Feb 2018 11:07:35	https://digi.ninja/projects/ip_camera_finder.php	font-src	data	show/hide	2
Enforced	07 Feb 2018 11:17:35	https://digi.ninja/projects_networking.php	font-src	data	show/hide	1

View 100 records

```
{
    "csp-report": {
        "document-uri": "https://di
        "effective-directive": "fon
        "original-policy": "default
        "blocked-uri": "data"
    }
}
```

https://digi.ninja

Public Key Pinning

Short version – Specify in a header which CAs can sign your certificates

Longer version is a bit more complicated than that

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

Public Key Pinning

Current advice – Don’t do it!

Public Key Pinning

Deprecated by Google in Chrome 67

https://www.theregister.co.uk/2017/10/30/google_hpkp/

Public Key Pinning

PKP Suicide

https://www.smashingmagazine.com/be-afraid-of-public-key-pinning/

Public Key Pinning

Replaced by…

Expect-CT

Replaces Key Pinning

Tells the browser to only accept a certificate if there is an entry for it in the certificate transparency register

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

Expect-CT

No header – browser dependent

One options:

- enforce – Only accept the cert if CT found
- max-age – The number of seconds to honour the header
- report-uri – URI to report failures to

Expect-CT

Example headers:

Expect-CT: max-age=0, report-uri="<report URI>"

Expect-CT: enforce, max-age=60, report-uri="<report URI>"

Expect-CT

Check CT logs here

https://crt.sh

Facebook monitoring

https://www.facebook.com/notes/protect-the-graph/introducing-our-certificate-transparency-monitoring-tool/1811919779048165/

Expect-CT

Criteria	Identity = 'digi.ninja'

Certificates

crt.sh ID	Logged At ⇧	Not Before	Issuer Name
272121594	2017-12-06	2017-12-06	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
250449914	2017-11-09	2017-11-09	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
223131904	2017-10-03	2017-10-03	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
205068152	2017-09-06	2017-09-06	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
205061299	2017-09-06	2017-09-06	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
155189092	2017-06-16	2017-06-16	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
108156775	2017-03-24	2017-03-23	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
71780150	2016-12-31	2016-12-30	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
40995830	2016-10-07	2016-10-07	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
24760643	2016-07-16	2016-07-16	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
17197415	2016-04-24	2016-04-24	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
17190771	2016-04-23	2016-04-23	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
12473468	2016-02-01	2016-01-31	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X1
11838949	2016-01-05	2016-01-05	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X1
11838948	2016-01-05	2016-01-05	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X1
11838565	2016-01-05	2016-01-05	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X1
7212947	2015-04-18	2015-03-03	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2

https://digi.ninja

Cookies

```
PS C:\Users\robin> Invoke-WebRequest -Uri https://vuln-demo.com/cookie_flags/index.php -SessionVariable s
PS C:\Users\robin> $s.Cookies.GetCookies("https://vuln-demo.com/cookie_flags/index.php")


Comment    :
CommentUri :
HttpOnly   : True
Discard    : False
Domain     : vuln-demo.com
Expired    : False
Expires    : 01/01/0001 00:00:00
Name       : justhttponly
Path       : /cookie_flags/index.php
Port       :
Secure     : False
TimeStamp  : 07/02/2018 12:25:56
Value      : no+JS
Version    : 0

Comment    :
CommentUri :
HttpOnly   : False
Discard    : False
Domain     : vuln-demo.com
Expired    : False
Expires    : 01/01/0001 00:00:00
Name       : justsecure
Path       : /cookie_flags/index.php
Port       :
Secure     : True
TimeStamp  : 07/02/2018 12:25:56
Value      : it%27s+secure
Version    : 0

Comment    :
CommentUri :
HttpOnly   : True
Discard    : False
Domain     : vuln-demo.com
Expired    : False
Expires    : 01/01/0001 00:00:00
Name       : both
Path       : /cookie_flags/index.php
Port       :
Secure     : True
TimeStamp  : 07/02/2018 12:25:56
Value      : it%27s+secure+and+no+JS
Version    : 0
```

The Flags

- Secure
- HttpOnly
- SameSite

Secure

Ensures the cookie is only sent over HTTPS

Stops cookies being sniffed while in transit

Should be set on all session cookies

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

Secure

Not needed if no HTTP:// site exists?

http://site.com:443/page

HttpOnly

Prevents JavaScript from accessing the cookie

Blocks session hijacking through cookie theft

Should be set on all session cookies

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

SameSite

New flag from around November 2017

Chrome 62 onwards, Firefox 59 onwards

Not in IE, Edge or Safari

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

SameSite

Cookie only sent with a request if the request comes from the same site

Designed to prevent Cross-Site Request Forgery (CSRF) attacks

SameSite

No header – no restriction on cookies

Two options:

- strict – never send the cookie unless the request originates same site
- lax – send the cookie for “Safe” methods (GET, HEAD, OPTIONS, TRACE)*

* Recommended

Any Questions?

Robin Wood

https://digi.ninja

@digininja