👇

Fingerprinting a webserver + finding new web applications

Tools

- Nmap
- Nikto
- Netcraft online tool
- Wappalyzer browser plugin
- Curl/wget to send malformed requests

Test for

- nmap -sV ip_adr

-sV is the flag for banner grabbing

- Nikto —host ip_addr

Nikto will automatically try banner grabbing

- Wapplyzer browser plugin

Will auto analyse if possible

- curl http://ip_addr/BAD_REQUEST

Some servers will respond differently to bad requests

Enumerating different webapps

- Different baseURLs may refer to different applications, often we can only find these by directory brute forcing, for example https://google.com might go to the search engine but https://google.com/mail might point to a totally different webapp
- Besides port 80 and 443 we should investigate anything that looks like a webserver. Use tools like nmap to enumerate all the open ports, -p- for all ports instead of top 1000 and -sU for UDP ports included
- vHosts are different hosts on the same webserver, for example mail.google.com and www.google.com might point to the same webserver but they might return a different application based on the routing of the URL. We can use vHost brute forcing tools.

Looking for information on the webpage

- Investigate comments made by developers
- Investigate metadata
- Review JS files
- Identify if any debug features exists that we might be able to use
- Map the application flows in xmind

Tools

General

- Linkfinder (https://github.com/GerbenJavado/LinkFinder)
- SecretFinder (https://github.com/m4ll0k/SecretFinder)
- Burp suite, right click a target > engagement tools > extract comments (Only in pro)
- ZAP proxy
- Wget to download JS files
- Google maps API scanner https://github.com/ozguralp/gmapsapiscanner/
- httprint – http://net-square.com/httprint.html
- httprecon – http://www.computec.ch/projekte/httprecon/
- Netcraft – http://www.netcraft.com
- Nmap – https://nmap.org/
- Netcat – https://sectools.org/tool/netcat/

WhatWeb

Website: https://github.com/urbanadventurer/WhatWeb

Currently one of the best fingerprinting tools on the market. Included in a default Kali Linux build. Language: Ruby Matches for fingerprinting are made with:

- Text strings (case sensitive)
- Regular expressions
- Google Hack Database queries (limited set of keywords)
- MD5 hashes
- URL recognition
- HTML tag patterns
- Custom ruby code for passive and aggressive operations

Sample output is presented on a screenshot below:

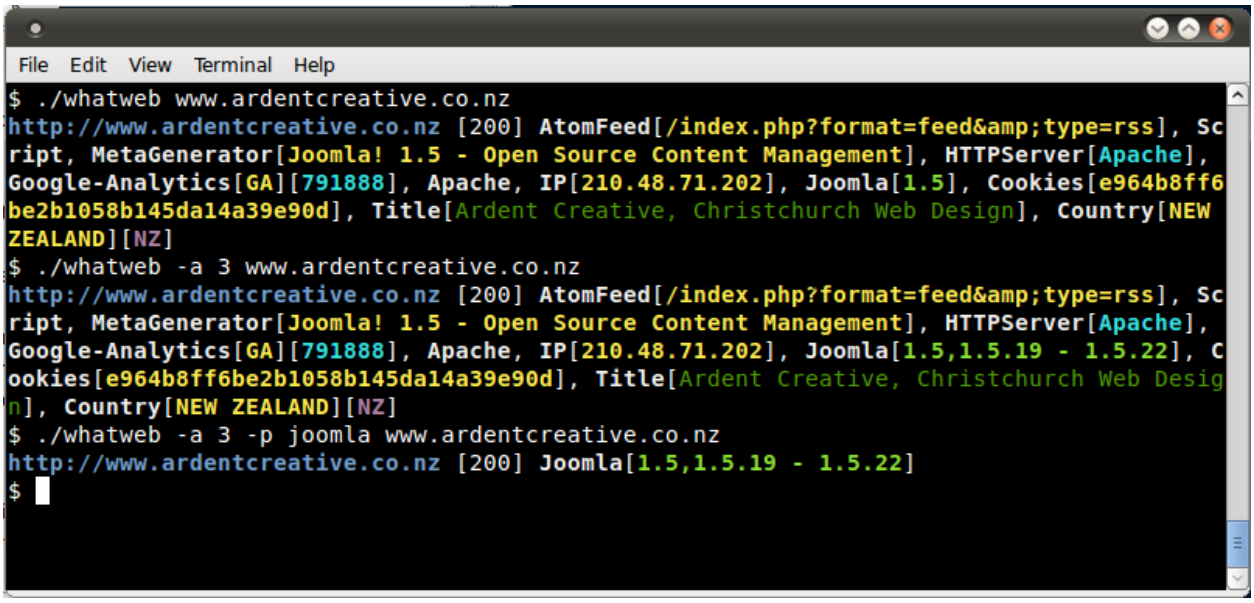

Figure 4.1.8-8: Whatweb Output sample

Wappalyzer

Website: https://www.wappalyzer.com/

Wapplyzer is available in multiple usage models, the most popular of which is likely the Firefox/Chrome extensions. They work only on regular expression matching and doesn't need anything other than the page to be loaded in browser. It works completely at the browser level and gives results in the form of icons. Although sometimes it has false positives, this is very handy to have notion of what technologies were used to construct a target website immediately after browsing a page.

Sample output of a plug-in is presented on a screenshot below.

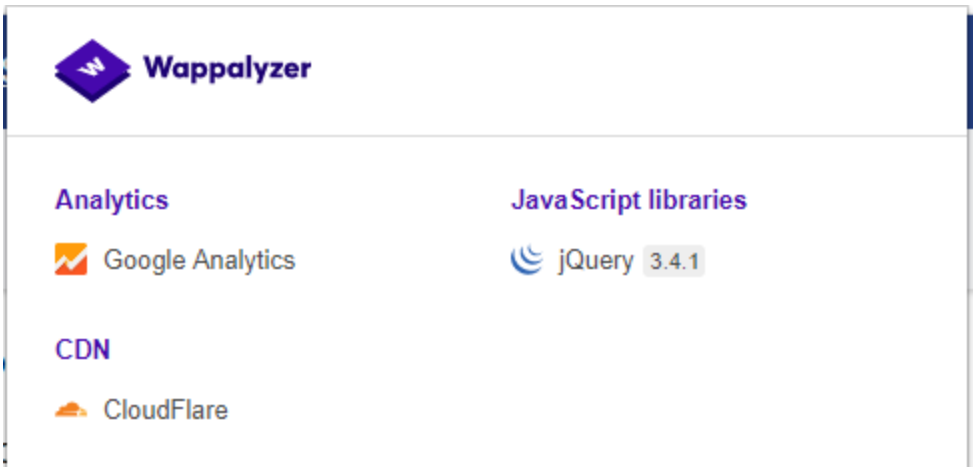

Figure 4.1.8-9: Wappalyzer Output for OWASP Website

Interesting cookies indicating frameworks

Common cookies

Framework	Cookie name
Zope	zope3
CakePHP	cakephp
Kohana	kohanasession
Laravel	laravel_session
phpBB	phpbb3_
WordPress	wp-settings
1C-Bitrix	BITRIX_
AMPcms	AMP
Django CMS	django
DotNetNuke	DotNetNukeAnonymous
e107	e107_tz
EPiServer	EPiTrace, EPiServer
Graffiti CMS	graffitibot
Hotaru CMS	hotaru_mobile
ImpressCMS	ICMSession

Framework	Cookie name
Indico	MAKACSESSION
InstantCMS	InstantCMS[logdate]
Kentico CMS	CMSPreferredCulture
MODx	SN4[12symb]
TYPO3	fe_typo_user
Dynamicweb	Dynamicweb
LEPTON	lep[some_numeric_value]+sessionid
Wix	Domain=.wix.com
VIVVO	VivvoSessionId

HTML Source Code

Application	Keyword
WordPress	`<meta name="generator" content="WordPress 3.9.2" />`
phpBB	`<body id="phpbb"`
Mediawiki	`<meta name="generator" content="MediaWiki 1.21.9" />`
Joomla	`<meta name="generator" content="Joomla! - Open Source Content Management" />`
Drupal	`<meta name="Generator" content="Drupal 7 (http://drupal.org)" />`
DotNetNuke	`DNN Platform - [http://www.dnnsoftware.com](http://www.dnnsoftware.com)`

General Markers

- `%framework_name%`
- `powered by`
- `built upon`
- `running`

Specific Markers

Framework	Keyword

Framework	Keyword
Adobe ColdFusion	`<!-- START headerTags.cfm`
Microsoft ASP.NET	`__VIEWSTATE`
ZK	`<!-- ZK`
Business Catalyst	`<!-- BC_OBNW -->`
Indexhibit	`ndxz-studio`

Remediation

While efforts can be made to use different cookie names (through changing configs), hiding or changing file/directory paths (through rewriting or source code changes), removing known headers, etc. such efforts boil down to “security through obscurity”. System owners/admins should recognize that those efforts only slow down the most basic of adversaries. The time/effort may be better used on stakeholder awareness and solution maintenance activities.