

Footprinting overview

- Also known as **fingerprinting** or **reconnaissance**
- 📋 Gathering information about a target system
- E.g. software, network protocols, operating systems or hardware devices.
- End goal is to find a way to break into the system.
- 🕒 Often offered as separate service bought by companies to check against leaks and to see what data is there.
- See also • [Reconnaissance](#) | [Hacking stages](#) and • [Information Gathering](#) | [Penetration testing phases](#)

Footprinting types

Passive footprinting

- Also known as **passive reconnaissance**, **passive fingerprinting** or **passive information gathering**
- 📋 No direct contact with target
- Rely on information that is publicly available.
- Most difficult to detect
- E.g. • News • job postings • [WHOIS](#) databases • government records • document sifting • [dumpster diving](#) | [Social engineering](#) • [competitive analysis](#) • browser search • map lookup • DNS lookup • Facebook/Twitter search

Open-source intelligence (OSINT)

- 📋 Collection and analysis of information that is gathered from public, or open, sources
- ¶ "Open-source" is unrelated to open-source software or collective intelligence
- Categories: • media • internet • public government data • professional and academic publications • commercial data • grey literature
- [awesome-osint](#) | [list of tools](#), [OsintFramework](#) | [graph of tools](#)

Competitive intelligence

- Also known as **competitive analysis**
- Assessment of the strengths and weaknesses of current and potential competitors
- Tools include
 - Traffic statistics: [Alexa](#)
 - News: [Google finance](#)
 - Company plans/finances: • [SEC Info](#) • [experian](#) • [Market Watch](#) • [Wall Street Monitor](#) • [EuroMonitor](#)
 - Company origins and development: • [EDGAR Database](#) • [Hoovers](#) • [LexisNexis](#) • [Business Wire](#)

Active footprinting

- Also known as **active reconnaissance**, **active fingerprinting** or **active information gathering**
- 📋 Direct contact with target including

- Possible for target to be aware e.g. through tasks that may be logged or recorded
- Examples
 - Buying beers for company employees to see what you can extract.
 - Network mapping with `nmap`, perimeter mapping, port scanning, web profiling...
 - • E-mail tracking • Phishing scheme with an email • Querying name servers • File metadata • Social engineering • Extracting DNS information • [Traceroute](#) analysis
- 💡 Easier idea to start with passive footprinting by gathering all publicly available data
 - Then organizing it, and putting in one place.
 - Then use active footprinting with starting probing for ports, networks, possible vulnerabilities etc.
- 💡 Good to learn more about stuff (employees) of a company
 - through them you can learn a lot more and gain a lot more access
 - e.g. contact them through social media and start a conversation
 - e.g. join a conference that you see the person is attending on LinkedIn and meet him.


Footprinting information

- **Network information**
 - Domains, subdomains
 - IP addresses
 - [Whois](#) and DNS records
 - VPN firewalls using e.g. [ike-scan](#)
- **System information**
 - Web server operating systems
 - Server locations
 - Users
 - Passwords
- **Organization information**
 - Employee information
 - Organization's background
 - Phone numbers
 - Locations

Footprinting objectives

- **Learn security posture**
 - Analyze security
 - Find loopholes
 - Create an attack plan
- **Identify focus area**
 - Narrow down the range of IP addresses.
- **Find vulnerabilities**
 - Identify weaknesses in the target's security.
- **Map the network**
 - Graphical representation of target's network a guide during the attack.

Footprinting tools

- Collects and visualizes information e.g. • IP location • routing • business • address • phone number • social security number • source of an email and a file • DNS • domain
-  [Maltego](#)
 - Proprietary software for open-source intelligence (OSINT)
 - Provides graphical link for investigative tasks.
- [Recon-ng \(The Recon-ng Framework\)](#)
 - Open source CLI tools for open source web-based reconnaissance
- [FOCA](#)
 - Fingerprinting Organizations with Collected Archives
 - Open-source tool to find metadata and hidden information in the documents:
 1. Finds documents (e.g. PDF, SVG) through search engines or manual upload
 2. Analyze them and identify which documents are created by same team, using which servers/clients.
- [Recon-dog](#)
 - Open-source CLI tool self-claimed as Reconnaissance Swiss Army Knife
 - Can extract targets from STDIN (piped input) and act upon them
 - Passive reconnaissance tool extracting all information with APIs without any contact with target
- [Dmitry](#) (DeepMagic Information Gathering Tool)
 - CLI tool to analyze a website e.g. `dmitry https://cloudarchitecture.io`
 - • Performs [WHOIS](#) lookup on IP and domain • Retrieves [Netcraft](#) information • Search for subdomains/email addresses • Performs TCP scanning • Grabs banner for each port

Footprinting reports

- Includes
 - Details about the performed tests
 - Used techniques
 - Test results
- It should also include
 - List of vulnerabilities and how they can be fixed
 - E.g. wrong configuration in webserver because you're allowing a forward and somebody is using your proxy for reflection attacks.
 - Reflection attack = Send a packet from A to B, A gives wrong source IP for DDoS attacks.
 - List sources of information e.g. DNS, social media, social engineering.
 - List what information you gathered from each source
 - E.g. login pages, technologies, files, contact details, GPS location, IP address, email servers.
- Should be kept highly confidential

Countermeasures


- Enforcing security policies
- Educating employees about security threats
 - Raises awareness, reduces risks dramatically
- Encrypting sensitive information

- 💡 Use proper encryption everywhere
 - 🚫 Many companies uses VPN/proxy with encryption for outside communication, but service communicate with each other without any encryption.
- Disabling protocols that are not required
- Proper service configuration
 - Double check all services that application depends.
 - Do not disable/enable configuration without knowing consequences.
- Scrutinize information released to the public domain
 - E.g. you post on social media which routers the company has just bought
 - Allows hacker to
 - know default router configurations
 - get image of OS in the router and conduct tests in a VM
- Limit site caching
 - Inform search engines what they're supposed to index through e.g. `robots.txt`
 - E.g `User-agent: * Disallow: /` prevents indexing any page (`Disallow: /`) for any crawler (`User-agent: *`)
- Use Whois Guard
- Restricting access to social media
 - Extra risk as you click on many links and giving away companies IP address


Search engines and online resources

- For e.g. information about the target organization's employees, intranet, login pages...
- Sources include • social networking sites • people search services • alerting services • financial services • job sites showing target infrastructure details, physical location, and employee details • deep and dark web

Google hacking

- Involves using a set of search operators (**dorks**) and building complex queries.
-  Form of [passive reconnaissance](#)
- Common dorks:

Dork	Definition	Example
<code>site</code>	Only from the specified domain	<code>azure</code> <code>site:cloudarchitecture.io</code>
<code>inurl</code>	Only pages that has the query in its URL.	<code>inurl: cloudarchitecture</code>
<code>intitle</code>	Only pages that has the query in its title.	<code>intitle: cloud</code> <code>architecture</code>
<code>cache</code>	Cached versions of the queried page	<code>cache:cloudarchitecture.io</code>
<code>link</code>	Only pages that contain the queried URL. Discontinued.	<code>link:cloudarchitecture.io</code>
<code>filetype</code>	Only results for the given filetype	<code>filetype:sql</code>

-  Usual to combine `filetype` and `site` dorks as see in [metagoofil](#)
- Google logical query operators

Operator	Definition	Example
<code>OR, </code>	X or Y but not both	<code>jobs OR gates, jobs gates</code>
<code>AND</code>	Results related to both X and Y, google default.	<code>jobs AND gates</code>
<code>-</code>	Exclude a term or phrase	<code>jobs -apple</code>
<code>*</code>	Wildcard that will match any word or phrase.	<code>"Google * my life" > google changed my life, google runs my life...</code>
<code>(,)</code>	Group multiple terms	<code>(iPad OR iPhone) apple</code>

- E.g. finding passwords: `intext:"please change your" password | code | login`
`file:pdf | doc | txt | docx -github`
 - `intext:` in the text of the website

- `"please change your" password`: Placing something in quote marks means it must contain the text as whole, not parts of it.
- `file:pdf`: specify what kind of file you want.
- `-github`: minus + word tells to exclude results containing that word(s).
- For complex searches use:
 - [Google Advanced Search](#) (no need for dorks)
 - [Google Advanced Image Search](#)
- 💡 Easier way may be using [Google Advanced Search](#) or [Advanced Image Search](#)

Google hacking tools

- [Google hack honeypot](#)
 - Logs google hacking queries against your resources
- [Google hacking database](#)
 - Helps you with
 - finding various types of files, including those that contain usernames and passwords.
 - VoIP footprinting using e.g. `intitle:"D-Link VoIP Router" "welcome"` to find pages containing D-Link login portals
 - VPN footprinting using e.g. `filetype:pcf "cisco" "GroupPwd"` to find Cisco VPN files with passwords
 - 💡 Once you find password lists and you can guess similar ones as people usually have similar passwords.

metagoofil

- [Open-source](#) tool to extract metadata of public documents (pdf,doc,xls,ppt,etc) available in the target websites
- Also helps with [website footprinting](#)
- Flow
 1. Queries Google for different filetypes that may have metadata
 - Combining `site:` and `filetype` dorks
 2. Downloads the documents to disk and extracts the metadata of the file
 3. Parses files using different libraries for metadata (e.g. Hachoir, pdfminer)

Online services

- Searching domain gives you some data about e.g. IP address, server, geolocation.
 - ⚠ Careful, can be fairly inaccurate, Generic results = No guarantee.
 - Far better to do your own search
 - Generic results = No guarantee
- [Website Watcher](#) to get notified if a web page is changed.

Reverse image search

- Allows tracking original source of an image
- E.g. • Google Image Search • TinEye Reverse Image Search • Yahoo Image Search

Video search engines

- Search video related to target and extract video information
- E.g. • YouTube • Google Videos
- Video analysis tools include • YouTube DataViewer • EZGif • VideoReverser.com,

Meta data engines

- Uses other search engines to build meta data of Internet
- Can give more information such as images, videos, blogs, news, articles about target
- E.g. • Startpage • MetaGer

FTP search engines

- Search files on FTP servers
- E.g. • NAPALM FTP Indexer • Global FTP Search Engine
- Can help to find tax documents, business strategies etc.


IoT search engines

- Can allow finding e.g. manufacturer details, geographical location, IP address, hostname, open ports
- E.g. [Shodan](#), Censys, and Thingful
- See [IoT footprinting.](#) | [IoT security](#)

Shodan

- Online [search engine](#)
- Finds specific types of IoT (webcams, routers, servers, etc.) connected to the internet using a variety of filters.
-  You can e.g. search for open ports `port: 1433`

Censys

- Online [censys](#)
-  Provides internet asset discovery i.e. scanning for unknown internet resources.
- Available on [search.censys.io](#)

Netcraft

- Allows you search web by domain (DNS) through [search DNS](#) service.
- Reports more information such as
 - If it uses HTML5 or flash (flash has many vulnerabilities)
 - `X-Frame-Options`: Do not allow this site to be rendered in an iframe
 - If it's allowed it allows for a phishing scheme such as [clickjacking](#)

[CrimeFlare](#)

- Helps you find IP addresses behind a CDN (e.g. CloudFlare)
- **CDN**: Protects against DDoS, geolocation of servers by having different IP address.
- People often use real IP addresses before CDN, you can then look at past DNS records to find it.





WHOIS, GeolpLocation and DNS interrogation

- All public records, accessing is not illegal.


WHOIS

- Query and response protocol (port 43)
- Used for retrieving information about assigned Internet resources
- To get WHOIS information you can
 - Use different websites such as whois.net
 - Use command-line: `whois cloudarchitecture.io`
- Two models
 - **Thick WHOIS**: information from all registrars for the specified set of data.
 - **Thin WHOIS**: limited information about the specified set of data.

WHOIS results

- Domain details
-  Domain owner details
 - Includes contact information of the owner
 - Can be hidden by a **WHOIS guard**
 - A proxy between the owner of the domain and who's accessing
 - Emails are usually still redirected to the owner.
 -  Allows for e-mail phishing to learn who the actual owner is.
- Domain server
 - Who it's registered with e.g. NameCheap.com, Gandi.net
 -  Site owner might have account in the server, and you can test passwords there.
- Net range
- Domain expiration
 -  If auto-renewal fails, someone can transfer a domain to another address for malicious behaviors or just to sell it back to you.
- Creation and last update dates


Regional internet registries

- WHOIS databases are maintained by the Regional Internet Registries (RIRs) such as:
 - **ARIN**: American Registry for Internet Numbers
 - **AFRINIC**: African Network Information Center
 - **APNIC**: Asia Pacific Network Information Center
 - **RIPE**: Réseaux IP Européens Network Coordination Centre
 - **LACNIC**: Latin American and Caribbean Network Information Center
-  Every ISP, hosting company etc. must be member of one of the registries to get IP addresses.

IP geolocation

- Helps find location information about a target
- Includes country, city, postal code, ISP, and so on
 - Country is mostly accurate but city, coordinates are not but approximated
- Helps with social engineering attacks
- E.g. [GeolpTool.com](https://geolpTool.com)

DNS interrogation

- Collecting information about DNS zone data.
 - e.g. server types and their locations
- Includes information about key hosts in the network
-  E.g. `host -t a cloudarchitecture.com`
 - `t` stands for type of domain record `a` gives A type of domain records.
 - Returns something like this:


```
cloudarchitecture.io has address 13.33.17.159
cloudarchitecture.io has address 13.33.17.136
```

- A records returns multiple IP addresses to increase speed and availability e.g. when hosting same content in multiple continents.
- See also [DNS enumeration](#)

Reverse DNS lookup

- Use one of IP addresses that's listed as an A
- `host 13.33.17.159`
 - Returns `159.17.33.13.in-addr.arpa domain name pointer server-13-33-17-159.arn53.r.cloudfront.net.`
- Multiple IP addresses can be tied to same domain
 - multiple domain addresses that are tied to the same IP

MX records

- Can be retrieved with `-t mx`
- Exposes which e-mail service they use
- Have a preference number to tell the SMTP client to try (and retry) each of the relevant addresses in the list in order, until a delivery attempt succeeds
 - The smallest preference number has the highest priority
-  Once a hacker knows who the e-mail provider is, he/she can create fake-mails using the provider to test e.g.
 - What kind of content is allowed
 - If a file be modified so it appears as PDF but make it executable
 - When an e-mail is labeled as spam / malicious

Email footprinting

- By monitoring the email delivery and inspecting the e-mail headers
- Information includes
 - IP address of the recipient
 - Geolocation of the recipient
 - Delivery information
 - Visited links
 - Browser and OS information
 - Reading time
- Can track emails using various **email tracking tools**
 - E.g. notifies sender of the email being delivered and opened by the recipient
 - Used by marketers, sellers etc.

Email header analysis

- Helps to determine an e-mail contains something malicious or not
- Email-headers include
 - Sender's name
 - IP/Email address of the sender
 - Mail server
 - Mail server authentication system
 - Send and delivery stamps
 - Unique number of the message

Authentication protocol headers

- Allows you to detect forged sender addresses.
- The goal is for sender to identify itself to the receiver.
- E-mail headers include information about their pass status

SPF: Sender Policy Framework

- E.g. 'PASS' with IP 209.85.220.69 or 'NEUTRAL' ...
- Verifies if the domain of the e-mail owned by the sending server.
 - If not passed, many e-mail providers just block it.
- Based on e-mail servers who publish records and says "here's the IP addresses we'll send e-mails"

DKIM: DomainKeys Identified Mail

- E.g. 'PASS' with domain accounts.google.com
- Allows the receiver to verify that an email claimed to have come from a specific domain was authorized by the owner of that domain using a digital signature on the domain.

DMARC: Domain-based Message Authentication, Reporting and Conformance

- E.g. `PASS` or `FAIL`
- Combination of two protocols SPF + DKIM
- It builds on them and adds more policy

Verifying email legitimacy

- Double check `FROM`
- Check the spelling in domain name so it's coming from the domain of the company
 - If it's random e-mail check if it's from one of the biggest domain providers or if something legit.
- Check IP of the domain
 - It can be someones computer (home router IP) or a private server
 - Major mail service providers checks to determine if domain of the e-mail is tied to the source IP of the e-mail (e.g. have a record)
 - 🧑 You can tie a public WiFi (e.g. coffee shop) IP to domain and send the e-mails from there.

E-mail policies

- Different e-mail service provider have different policies regarding to their SMTP
- 🧐 Once hacker recognizes e-mail servers then then he/she can create accounts there, send e-mails back and further to figure out what the rules are.
- E.g. google does not allow you to see the IP address of the sender
 - They proxy it behind one of their servers
 - Workarounds are not so efficient.
- Each have own ruling list
 - Determines e.g. what kind of files that can be send

Getting an IP address from an e-mail

- You can then get IP and a lot from browser headers including
 - browser information, OS info, device types
 - Revealing your IP is not safe as even home routers have pretty static IP addresses
 - Last usually 30 days up to 3 months
 - 🧐 You can still release DHCP lease in your home router settings to get a new IP from the ISP.
- You can send an image from a back-end server that you own
 - Some e-mail providers request it and hide users IP
- You can send a direct link
 - No e-mail provider can protect you from that
 - 🧑 Can be done through social engineering e.g.
 - You know from social media that Bob was celebrating yesterday. You send an e-mail stating "Hi Bob, crew and I had a great time last night, you're never going to guess what Sam did in toilet, threw himself up, check out his pictures"
 - E.g.

1. Install apache `yum install httpd`
2. Start apache `systemctl start httpd`
3. Create a file: `cd /var/www/html/` then `touch <RESOURCE_NAME>;`
4. Check logs live: `tail -f /var/log/httpd/access_log`
5. You'll get the IP address when the link (`<IP_ADDRESS>/<RESOURCE_NAME>`) is opened
 - You can find out self IP address using `curl ifconfig.me`
6. And you can look at the location of IP using `geoiplookup <IP_ADDRESS>;`

Website footprinting

- Hackers can map the entire website of the target without being noticed
- Gives information about:
 - Software
 - Operating system
 - Subdirectories
 - Contact information
 - Scripting platform
 - Query details

Web spiders

- Programs designed to help in website footprinting
- Methodically browse a website in search of specific information.
- Information collected this way can help attackers perform social engineering attacks.

Cookie examination

- Reveals what software that is running on the server and its behavior
- Possible to identify the scripting platforms.

Examining website headers

- By examining the website headers, it is possible to obtain information about:
 - Content-Type
 - Accept-Ranges
 - Connection Status
 - Last-Modified Information
 - X-Powered-By Information
 - E.g. ZendServer 8.5.0, ASP.NET
 - Web Server Information
 - Server header can give you e.g. Apache Server on CentOS
- You can also analyze what website pulls
 - In debugging developer tool of most browsers (ctrl+shift+c) network section
 - For each request you can see remote IP address, and response headers for further analysis.

Source code examination

Comment analysis

- Possible to extract information from the comments
- In most of browsers you can right click and how source
- Walkthrough
 - In almost any browser: Right click => Show source

- Check for HTML `<!-- comment -->` or JavaScript `// comment` comments
- They are skipped by interpreters and compilers, only for human eyes
- They can be instructions for other developers, notes for themselves
 - E.g. this library won't work as this element is not supported
 - Gives you clues about what technology (frameworks, languages) they use in the background


Observing link and image tags

- Html links: `href=cloudarchitecture.io`
- Gain insight into the file system structure
- You can find e.g. a caching server and check vulnerabilities for that caching server.

Cloning websites

- Also called **website mirroring**
- Helps in
 - browsing the site offline
 - searching the website for vulnerabilities
 - discovering valuable information and metadata.
- Can be protected with some detections based on e.g. page pull speed, behavior, known scrapers, AI.
- 💡 Good tool for setting up fake websites.
 - E.g. manually recreate login pages
 - If you control the DNS you can do a redirect.
- Allows you to save social media pages with this however most are protected, and illegal to clone.
- **Website monitoring tools** can send notifications on detected changes.
- 💡 Protection against fake websites
 - Always check domain name for misspelling
 - Make sure it's HTTPS, if it's not the data can be sniffed easily
 - Protects against someone taking over DNS
 - If the other part does not have the certificate, browser does not accept communication
 - Check SSL certificate authority, if it's changing, it can prompt a question.
 - Certificates expire usually in a year.

Website cloning tools

- [httrack](#)
 - `httrack https://testwebpage.com` to copy
-  `wget`
 - Basic utility that can be used for mirroring website
- Or one could manually copy paste source code of HTML + CSS

Extracting metadata

- You can extract metadata of files (e.g. images) from a webpage

- Metadata can include
 - Owner of the file
 - GPS coordinates (images)
 - File type metadata
 - 🐧 Linux does not work with extensions e.g. `.pdf` but checks for the metadata.
 - Helpful as you will not be fooled by the extension

Tools for extracting metadata

- `hexdump`
 - Dump file as ASCII and inspect manually
 - E.g. `hexdump -C TEST_DOCUMENT.docx`
 - ⚠ Not recommended as it's pretty hard to extract information from binary.
- [ExifTool](#)
 - Reads + writes metadata of audio, video, PDF, docs etc.
 - E.g. `exiftool TEST_DOCUMENT.docx` would return something like `Microsoft Office word, version: 16.0`
- 📄 [Metagoofil | Google hacking tool](#)
 - Search for files that may have metadata for a website using Google and dump their metadata.

Network footprinting

- Collecting network range information to use the information to map the target's network
- Gives insights into how the network is structured and which machines belong to the network.

Nmap

- Used for network discovery
- Uses raw IP packets to determine e.g.
 - the available hosts on the network
 - the services offered by those hosts
 - operating systems they are
 - firewall types that are being used
 - and more...
- Not only used for malicious purposes but also for checking something is working as intended
 - e.g. check why a port is open and confirm it's closed
- E.g. `nmap -v -p 0-2000 -o -sv 178.128.203.1`
 - `-v`: verbose, more output than usual
 - `-d` prints even more.
 - `-p`: for port
 - default: 0-1024
 - the higher the ranges is the longer it takes.
 - `-o`: os detection (best guess)
 - `-sv`: versions of all detected services (best guess)
 - 💡 Allows you to check for vulnerabilities of a specific version of that services e.g. through [exploit database](#)
 - `178.128.203.1`: can also specify subnet also e.g. `/24`
- 🇬🇧 In UK and Germany it's illegal to conduct a scan on a network, more [Nmap | legal issues](#)
- Read more about Nmap in [Nmap | Scanning Tools](#)

Traceroute

- 📄 Programs used for discovering routers that are on the path to the target host.
- You always go through multiple hops before you reach target
 - E.g. first hop being your router, then routers & switches ISP provider and the router that sends traffic out of the country...
- Helps hacker to collect information about
 - network topology
 - trusted routers
 - firewall locations
- Can use protocols such as `ICMP` (often), `TCP`, `UDP`, `DCPP`..
- 📌 There can be hops that are invisible/undetectable
 - 💡 You can craft special packets to detect them with custom time to lives, their failure

- Uses TTL field in the IP header to discover the route.
 - Starts by setting TTL to 1
 - Stops at each hop on the way to the destination and providing information to the sender about that hop
 - The TTL is incremented by 1 for each hop discovered
- Used to create network diagrams and plan attacks.
- Helps with e.g. man-in-the-middle attacks.
- It records IP addresses and DNS names of discovered routers.
- Commands
 - Unix tool: `traceroute 178.128.203.1` (uses UDP)
 - Using Nmap: `nmap traceroute --script traceroute-geolocation 178.128.203.1 -d`
 - Using hping: `hping3 -tracert -S {target ip}`
 - Windows tool: `tracert 178.128.203.1` (uses ICMP)