HO# 2.2: Reconnaissance, Info Gathering & OSINT

Overview (Phase 1- Reconnaissance and Information Gathering)

The Information gathering phase (reconnaissance) is the initial step in the penetration testing lifecycle. *This phase involves collecting as much public information as possible about the organization, systems, networks, applications, and employees to identify potential vulnerabilities and formulate a strategy for further testing.* Information gathering can be divided into two main categories; passive and active.

Passive Information Gathering

Passive information gathering (reconnaissance) involves collecting data without directly interacting with the target system, reducing the risk of detection. Gathering information from publicly available sources like news outlets, blogs and social media platforms (Twitter, Facebook, LinkedIn) is named as Open-Source Intelligence (OSINT). The techniques used for OSINT are Web Scraping, Google Dorking, and social media profiling. The tools that we will be using for Reconnaissance and Information gathering in today's hand out are `host, nslookup, dig, whois, knockpy, netdiscover, traceroute, whatweb, theHarvester, sherlock, wfw00f, Google Dorking, OSINT framework.`

Active Information Gathering

Active information gathering also known as scanning involves directly interacting with the target network, hosts, ports, employees, and so on to collect data. This can include discovering open ports, services, vulnerabilities, and other critical details about the target machine/NW. This approach often leaves traces or logs on the target system, making it more detectable. DONOT perform active network scanning unless you have written permission of the system owner to perform that testing. The most common tool that is used for active information gathering is `nmap`. The Active Information Gathering normally comes in the 2nd phase of penetration testing and will be dealt with in our upcoming handouts.

Host

In order to install DNS related utilities, use the following Linux command:
        `$ sudo apt-get install dnsutils`

The **host** is a utility that performs DNS lookups. Normally *used to convert names to IP address and vice versa*. For details read the man pages.
        `$ host arifbutt.me`

```
┌──(kali㉿kali)-[~]
└─$ host arifbutt.me
arifbutt.me has address 68.65.120.238
;; communications error to 192.168.8.1#53: timed out
arifbutt.me mail is handled by 0 mail.arifbutt.me.

┌──(kali㉿kali)-[~]
└─$ host pucit.edu.pk
pucit.edu.pk has address 202.147.169.205
pucit.edu.pk mail is handled by 1 ASPMX2.GOOGLEMAIL.COM.
pucit.edu.pk mail is handled by 5 ALT2.ASPMX.L.GOOGLE.COM.
pucit.edu.pk mail is handled by 1 ASPMX.L.GOOGLE.COM.
pucit.edu.pk mail is handled by 10 ASPMX3.GOOGLEMAIL.COM.
pucit.edu.pk mail is handled by 5 ALT1.ASPMX.L.GOOGLE.COM.

┌──(kali㉿kali)-[~]
└─$ host bbc.com
bbc.com has address 151.101.64.81
bbc.com has address 151.101.128.81
bbc.com has address 151.101.192.81
bbc.com has address 151.101.0.81
bbc.com has IPv6 address 2a04:4e42:400::81
bbc.com has IPv6 address 2a04:4e42:600::81
bbc.com has IPv6 address 2a04:4e42::81
bbc.com has IPv6 address 2a04:4e42:200::81
bbc.com mail is handled by 20 cluster8a.eu.messagelabs.com.
bbc.com mail is handled by 10 cluster8.eu.messagelabs.com.
```

Description of output `$ host bbc.com`

- o The first four lines show the (**A record**), which are the four server's IPv4 addresses where the website is hosted.
- o The next four lines show the (**AAAA record**), which are the corresponding four server's IPv6 addresses where the website is hosted.
- o The last two lines show the (**MX or Mail Exchange record**), which are the mail servers for `bbc.com`, along with their priorities (lower numbers indicate higher priority).
- o You may come across the (**NS or Name Server record)**, which indicate the authoritative name servers responsible for managing the DNS records of a domain.

Note: For detailed option of this command, the students are advised to read the man pages

Nslookup   https://www.nslookup.io

Like `host` command, **nslookup** (Name Server Lookup) is also used for name to IP address mapping and vice versa. It is more versatile as it can be used to get other specific DNS records such as A (Address), AAAA (IPv6 Address), MX (Mail Exchange), NS (Name Server), TXT (Text), and more.

    $ nslookup arifbutt.me

Description of output:
o   Server shows the IP of the DNS server used to perform the lookup. It is my home router address, in your case it might be 8.8.8.8, which is the IP of Google's public DNS server
o   Address shows the IP along with the default port number for DNS traffic.
o   Then you may get two types of answers, authoritative or non-authoritative. The non-authoritative answer indicates that the answer came from a DNS server cache, which do not have the zone file. The authoritative answer indicates that the answer is returned by the DNS server responsible for that domain.
o   Finally, you have the domain name and the corresponding IP address.

```
 $ nslookup pucit.edu.pk
Server:          192.168.8.1
Address:         192.168.8.1#53

Non-authoritative answer:
Name:    pucit.edu.pk
Address: 202.147.169.205


 (kali kali)-[~]
 $ nslookup arifbutt.me
Server:          192.168.8.1
Address:         192.168.8.1#53

Non-authoritative answer:
Name:    arifbutt.me
Address: 68.65.120.238
```

Reverse DNS Lookup

Reverse DNS (rDNS) lookup is the process of querying the Domain Name System (DNS) to determine the domain name associated with a given IP address. Unlike a forward DNS lookup, which resolves a domain name to an IP address, a reverse DNS lookup resolves an IP address to its corresponding domain name. Following screenshot displays a reverse DNS lookup:

    $ nslookup 68.65.120.238

```
dartsec$ nslookup 68.65.120.238
238.120.65.68.in-addr.arpa      name = server106-5.web-hosting.com.

Authoritative answers can be found from:
```

Dig

The **dig** is also a DNS lookup utility that stands for Domain Information Groper. The dig utility is quite similar to `nslookup` but provides a more detailed and structured output. Do read the man pages for details:

$ dig google.com

```
┌──(kali㉿kali)-[~]
└─$ dig google.com
;; communications error to 192.168.8.1#53: timed out

; <<>> DiG 9.20.1-1-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ──»HEADER«── opcode: QUERY, status: NOERROR, id: 24307
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                        IN      A

;; ANSWER SECTION:
google.com.             6       IN      A       172.217.19.206

;; Query time: 40 msec
;; SERVER: 192.168.8.1#53(192.168.8.1) (UDP)
;; WHEN: Wed Sep 11 10:58:00 EDT 2024
;; MSG SIZE  rcvd: 44
```

Description of output:
1. The first few lines show the version of `dig`, the domain queried, the default global options (`+cmd`), and Got answer, showing that the query was successful and response was received.
2. Query Response Header: Opcode specifies the type of query. In this example QUERY means standard query. Status specifies the status of the query. In this example NOERROR means it was successful. ID is the unique identifier of the query. Flags specifies details about the query: *qr* means query response, *rd* means recursion desired, *ra* means recursion available on the DNS server side. Finally, we have number of queries made and the number of answers received. The number of authority records and number of additional records received if any.
3. QUESTION SECTION displays the query data that was sent:
   o The first column is the domain name queried.
   o The second column is the query type (IN = Internet).
   o The third column specifies the record (A means IPv4, AAAA means IPv6).
4. ANSWER SECTION displays the response:
   o The first column lists the domain name that was queried.
   o The second column is the Time to Live in seconds for which a receiving server can cache this information. After this the mapping becomes invalid and any query must be sent again to the authoritative server.
   o The third column shows the query class. In this case, IN stands for Internet.
   o The fourth column displays the query type. In this case, A stands for IPv4 address, AAAA means IPv6 address.
   o The final column displays the IP address associated with the domain name.
5. Query Statistics displays:
   o The time it took to complete the DNS query. The DNS server used for this query and the port number. The date and time when the query was executed. The size of the DNS response message, in this example it is 44 Bytes.

Whois

The **`whois`** *command is used to retrieve domain registration information from* `whois` *databases.* These databases store publicly available information about domain names, such as the registry, registrar, registration and expiration dates, name servers, and the contact details of the domain owner (registrant). When trying to come up with a domain name for a new website, performing a WHOIS lets you determine if that name is registered or available.

```
$ sudo apt-get install whois
$ whois google.com
```

```
dartsec$ whois google.com
    Domain Name: GOOGLE.COM
    Registry Domain ID: 2138514_DOMAIN_COM-VRSN
    Registrar WHOIS Server: whois.markmonitor.com
    Registrar URL: http://www.markmonitor.com
    Updated Date: 2019-09-09T15:39:04Z
    Creation Date: 1997-09-15T04:00:00Z
    Registry Expiry Date: 2028-09-14T04:00:00Z
    Registrar: MarkMonitor Inc.
    Registrar IANA ID: 292
    Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
    Registrar Abuse Contact Phone: +1.2086851750
    Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
    Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
    Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
    Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
    Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
    Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
    Name Server: NS1.GOOGLE.COM
    Name Server: NS2.GOOGLE.COM
    Name Server: NS3.GOOGLE.COM
    Name Server: NS4.GOOGLE.COM
    DNSSEC: unsigned
    URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2024-07-12T14:14:04Z <<<
```

Description of output:
o   Registry Domain ID**:** A unique identifier assigned to the domain in the domain registry. This is specific to the top-level domain (in this case, `.com`) and helps uniquely identify the domain within the registry's database. A *registry* is an organization that manages top-level domain names, *registrar* is accredited organization like namecheap or GoDaddythat sell domain names to the public, and finally *registrant* is the person or company who registers a domain name.
o   Registrar WHOIS Server: The address of the WHOIS server that manages information for the domain. This server can be queried for further information about the domain registration.
o   Registrar URL: The website of the domain registrar responsible for the domain's registration.
o   Dates Section: The updated date specifies the last time the domain's registration details were updated. The creation date specifies when the domain was first registered. The expiration date specifies when the domain registration will expire unless renewed. If the domain expires, it can eventually become available for new registration.
o   Registrar Section: Specifies information about the registrar, i.e., the organization that manages the domain registration process of the domain.
o   Domain Status Section: This indicates the current status of the domain in the registry. Where registry is the company that manages a list containing a set of domain names

o   Name Server Section: These are the authoritative DNS servers that resolve the domain name to an IP address.

Using *whois* with IP address

When you pass an IP address to the `whois` command, the output provides information about the organization or entity that owns or manages that IP address. Instead of returning domain registration details (like it would for a domain name), the `whois` output for an IP address focuses on the IP range (netblock), organization details, and sometimes abuse contacts.

$ whois 8.8.8.8

```
NetRange:       8.0.0.0 - 8.127.255.255
CIDR:           8.0.0.0/9
NetName:        LVLT-ORG-8-8
NetHandle:      NET-8-0-0-0-1
Parent:         NET8 (NET-8-0-0-0-0)
NetType:        Direct Allocation
OriginAS:       AS3356
Organization:   Google LLC (GOGL)
RegDate:        1992-12-01
Updated:        2020-05-26
Ref:            https://rdap.arin.net/registry/ip/8.8.8.8

OrgName:        Google LLC
OrgId:          GOGL
Address:        1600 Amphitheatre Parkway
City:           Mountain View
StateProv:      CA
PostalCode:     94043
Country:        US
RegDate:        2000-03-30
Updated:        2019-10-31
Ref:            https://rdap.arin.net/registry/entity/GOGL

OrgAbuseHandle: ABUSE5250-ARIN
OrgAbuseName:   Google LLC - Network Engineering
OrgAbusePhone:  +1-650-253-0000
OrgAbuseEmail:  network-abuse@google.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/GOGL
```

Note: Students should visit the following online web services to get information about a specific domain at their own time:
- https://whois.domaintools.com/
- https://centralops.net/co/
- https://ipinfo.io/

Knockpy

Knockpy is an open-source tool used primarily for subdomain enumeration. It helps in *identifying subdomains associated with a target domain* by sending requests and collecting responses, providing useful information during penetration testing or OSINT (Open-Source Intelligence) activities. It does not come preinstalled on Kali, so you can download its source from GitHub and install it by giving the following commands:

```
$ git clone https://github.com/guelfoweb/knock.git
$ cd knock
$ pip3 install -r requirements.txt
$ which knockpy
$ knockpy --version
$ knockpy -h
$ knockpy -d pu.edu.pk --recon --bruteforce --threads 50
```

```
┌──(kali㉿kali)-[~]
└─$ knockpy -d pu.edu.pk --recon --bruteforce
bruteforce None
Recon.....: 100%|███████████████████████| 6/6
Processing: 100%|███████████████████████| 10775/10775
tomer.pu.edu.pk ['111.68.103.27']
http  [301, 'https://tomer.pu.edu.pk/', 'Apache']
https [200, None, 'Apache']
cert  [True, '2024-09-26']

spamd.pu.edu.pk ['111.68.103.27']
http  [301, 'https://spamd.pu.edu.pk/', 'Apache']
https [200, None, 'Apache']
cert  [True, '2024-09-26']

stamp.pu.edu.pk ['111.68.103.27']
http  [301, 'https://stamp.pu.edu.pk/', 'Apache']
https [200, None, 'Apache']
cert  [True, '2024-09-26']

banco.pu.edu.pk ['111.68.103.27']
http  [301, 'https://banco.pu.edu.pk/', 'Apache']
https [200, None, 'Apache']
cert  [True, '2024-09-26']

contractor.pu.edu.pk ['111.68.103.27']
http  [301, 'https://contractor.pu.edu.pk/', 'Apache']
https [200, None, 'Apache']
cert  [True, '2024-09-26']
```

Netdiscover   https://github.com/netdiscover-scanner/netdiscover

The **netdiscover** is an active/passive network discovery tool that uses Address Resolution Protocol (ARP) to identify hosts in a local area network (LAN). It can be used for both active as well as passive scanning. If netdiscover is not installed on your Kali machine, you can install it using following command:

```
$ sudo apt-get install netdiscover
$ man netdiscover
```

Active Scanning

By default, netdiscover perform active scanning, i.e., by sending ARP requests to every IP address in a given range and waits for responses. The **-r** option is used to specify a range to scan.

```
$ sudo netdiscover -r 10.0.2.0/8
```

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240

   IP              At MAC Address      Count    Len   MAC Vendor / Hostname
 _____

 10.0.2.1         52:54:00:12:35:00      1       60   Unknown vendor
 10.0.2.2         52:54:00:12:35:00      1       60   Unknown vendor
 10.0.2.3         08:00:27:c6:0a:ee      1       60   PCS Systemtechnik GmbH
 10.0.2.4         08:00:27:7a:fc:20      1       60   PCS Systemtechnik GmbH
```

Passive Scanning

If we want to maintain a low profile to avoid getting caught, we can use the **-p** option of netdiscover. This way netdiscover listens to network traffic to detect devices without sending any requests. It gives you the output only when some activity occurs on the machines in the network.

```
$ sudo netdiscover -p -r 10.0.2.0/8
```

```
Currently scanning: (passive)   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 240

   IP              At MAC Address      Count    Len   MAC Vendor / Hostname
 _____

 10.0.2.4         08:00:27:7a:fc:20      2      120   PCS Systemtechnik GmbH
 10.0.2.1         52:54:00:12:35:00      1       60   Unknown vendor
 10.0.2.3         08:00:27:c6:0a:ee      1       60   PCS Systemtechnik GmbH
```

TraceRoute

The **traceroute** command is used to trace the path that packets take from your device to any remote server (a domain name or an IP address) in a LAN or on the Internet. The output of the command shows the list of routers or gateways (hops) that the packets pass through on their way to the final destination. The output also shows the round-trip-time (RTT) for each hop, indicating how long it took for packets to travel to that hop and back. Each hop represents a router or gateway that the packets pass through on their way to the final destination. The `-I` option specifies that use ICMP echo requests instead of the default UDP packets

$ traceroute -I arifbutt.me

```
┌──(kali㉿kali)-[~]
└─$ traceroute -I arifbutt.me
traceroute to arifbutt.me (68.65.120.238), 30 hops max, 60 byte packets
 1  192.168.8.1 (192.168.8.1)  22.436 ms  21.907 ms *
 2  * * *
 3  * * *
 4  * * *
 5  10.80.238.134 (10.80.238.134)  300.230 ms  300.099 ms  299.973 ms
 6  * * *
 7  10.253.8.44 (10.253.8.44)  94.570 ms * *
 8  10.253.4.36 (10.253.4.36)  86.375 ms  221.792 ms  214.626 ms
 9  * * *
10  19551.sgw.equinix.com (27.111.229.74)  146.192 ms  146.081 ms  176.450 ms
11  * * *
12  172.21.0.62 (172.21.0.62)  359.882 ms  359.787 ms  359.686 ms
13  * * *
14  100.65.240.57 (100.65.240.57)  484.020 ms *  427.231 ms
15  100.65.220.66 (100.65.220.66)  345.213 ms  351.408 ms  451.546 ms
16  * * server106-5.web-hosting.com (68.65.120.238)  313.488 ms
```

Description of output:
- o The 68.65.120.238 is the IP of arifbutt.me to trace.
- o Each line displays the hop number, the IP of the next NW router along the path, and three times in milliseconds, representing the time taken for three packets to travel to that hop and back, showing the latency in the NW for each hop.
- o If a router doesn't respond to the ICMP request within the timeout limit, an asterisk (*) is shown instead of a time value. Multiple asterisks might indicate that a hop is unreachable or is configured to drop ICMP requests.
- o The command continues until it either reaches the final destination or hits the maximum number of hops (default is usually 30).

Whatweb   https://whatweb.net

The **whatweb** is a tool that is used to *identify and recognize all the web technologies available on the target website*. It can identify technologies used by websites such as blogging, content management system, and all JavaScript libraries. This tool can give us the following information:

- **HTTP Headers**: Displays the HTTP headers returned by the web server, including server type, content type, and other HTTP metadata.
- **Web Server Information**: Identifies the type of web server software running (e.g., Apache, Nginx, IIS).
- **CMS (Content Management System)**: Detects if the website is using a CMS such as WordPress, Joomla, Drupal, etc.
- **Frameworks and Libraries**: Identifies web application frameworks (e.g., Django, Ruby on Rails) and JavaScript libraries (e.g., jQuery, React).
- **Plugins and Extensions**: Lists any plugins or extensions detected on the website.

```
$ sudo apt-get install whatweb
$ whatweb -v pucit.edu.pk
```

```
WhatWeb report for https://pucit.edu.pk/
Status    : 200 OK
Title     : FCIT || Faculty of Computing and Information Technology
IP        : 202.147.169.205
Country   : PAKISTAN, PK

Summary   : Apache[2.4.10][mod_perl/2.0.8-dev], Bootstrap, HTML5, HTTPServer[Unix][Apache/2.4.10 (Unix) OpenSSL/1.0.1i PHP/5.5.15 mod_perl/2.0.8-dev Perl/v5.
16.3], JQuery[1.12.4,6.8.2], Lightbox, MetaGenerator[Powered by LayerSlider 6.8.2 - Multi-Purpose, Responsive, Parallax, Mobile-Friendly Slider Plugin for Wo
rdPress.,Powered by WPBakery Page Builder - drag and drop page builder for WordPress.,WordPress 5.1.1], Modernizr, OpenSSL[1.0.1i], Perl[5.16.3], PHP[5.5.15]
, PoweredBy[LayerSlider,WPBakery], Script[text/javascript], UncommonHeaders[link], WordPress[5.1.1], X-Powered-By[PHP/5.5.15]

Detected Plugins:
[ Apache ]
        The Apache HTTP Server Project is an effort to develop and
        maintain an open-source HTTP server for modern operating
        systems including UNIX and Windows NT. The goal of this
        project is to provide a secure, efficient and extensible
        server that provides HTTP services in sync with the current
        HTTP standards.

        Version       : 2.4.10 (from HTTP Server Header)
        Module        : mod_perl/2.0.8-dev
        Google Dorks: (3)
        Website       : http://httpd.apache.org/

[ Bootstrap ]
        Bootstrap is an open source toolkit for developing with
        HTML, CSS, and JS.

        Website       : https://getbootstrap.com/

[ HTML5 ]
        HTML version 5, detected by the doctype declaration
```

Description of output: The above screenshot is a cropped portion of the detailed output generated by whois command. Students are advised to understand the details and keep a note of it, so that it can be used in the next phases of pen-testing ☺

10

Aggressive Scan on IP Range

We can use the aggression level using **-a** option and give it address range. We will get errors on addresses that don't exist. We can ignore them by using the **--no-errors** option.

```
$ whatweb -v -a 3 10.0.2.1-10.0.2.254
```

```
dartsec$ whatweb -v -a 3 10.0.2.1-10.0.2.254
ERROR Opening: http://10.0.2.15 - Connection refused - connect(2) for "10.0.2.15" port 80
ERROR Opening: http://10.0.2.1 - Connection refused - connect(2) for "10.0.2.1" port 80
ERROR Opening: http://10.0.2.3 - Protocol not available - connect(2) for "10.0.2.3" port 80
ERROR Opening: http://10.0.2.13 - No route to host - connect(2) for "10.0.2.13" port 80
ERROR Opening: http://10.0.2.21 - No route to host - connect(2) for "10.0.2.21" port 80
ERROR Opening: http://10.0.2.23 - No route to host - connect(2) for "10.0.2.23" port 80
ERROR Opening: http://10.0.2.7 - No route to host - connect(2) for "10.0.2.7" port 80
ERROR Opening: http://10.0.2.27 - No route to host - connect(2) for "10.0.2.27" port 80
ERROR Opening: http://10.0.2.6 - No route to host - connect(2) for "10.0.2.6" port 80
ERROR Opening: http://10.0.2.26 - No route to host - connect(2) for "10.0.2.26" port 80
WhatWeb report for http://10.0.2.2
Status     : 200 OK
Title      : IIS Windows
IP         : 10.0.2.2
Country    : RESERVED, ZZ

Summary    : HTTPServer[Microsoft-IIS/10.0], Microsoft-IIS[10.0], X-Powered-By[ASP.NET]

Detected Plugins:
[ HTTPServer ]
        HTTP server header string. This plugin also attempts to
        identify the operating system from the server header.

        String      : Microsoft-IIS/10.0 (from server string)

[ Microsoft-IIS ]
        Microsoft Internet Information Services (IIS) for Windows
        Server is a flexible, secure and easy-to-manage Web server
        for hosting anything on the Web. From media streaming to
        web application hosting, IIS's scalable and open
        architecture is ready to handle the most demanding tasks.

        Version     : 10.0
        Website     : http://www.iis.net/

[ X-Powered-By ]
        X-Powered-By HTTP header

        String      : ASP.NET (from x-powered-by string)

HTTP Headers:
        HTTP/1.1 200 OK
        Content-Type: text/html
        Last-Modified: Tue, 10 Jan 2023 06:18:09 GMT
        Accept-Ranges: bytes
        ETag: "cb63b14fbb24d91:0"
        Server: Microsoft-IIS/10.0
```

Note:
- Students are advised to read the Linux man pages for details and understand the different aggression levels that controls the trade-off between speed/stealth and reliability.
- Students can visit the online web service available at https://sitereport.netcraft.com to find out the infrastructure and technologies used by a domain
- Students can visit the Google Chrome Web Store and install the **wappalyzer** extension, that will also help to find out different technology stacks used to build a specific website that they have opened in their browser

TheHarvester   https://github.com/laramies/theHarvester

TheHarvester is a command line utility that can be used to gather open-source intelligence (OSINT) about targets, such as domain names, IP addresses, email addresses and more. It is often employed during the initial information-gathering phase of a penetration test or security audit to collect publicly available data from a variety of sources. TheHarvester gathers data by querying various public data sources like search engines (Google, Bing, Yahoo), social media platforms (LinkedIn), and public databases to retrieve information related to a domain or company. Key Features of theHarvester are listed below:

- **Email Address Gathering**: Collects email addresses associated with a domain from various public sources.
- **Subdomain Enumeration**: Identifies subdomains of a target domain using multiple search engines and data sources.
- **IP Address and Hostname Discovery**: Retrieves IP addresses and hostnames related to a domain.

```
$ sudo apt-get install theharvester
$ theharvester –help
```

The `-d` option specifies the domain, `-l` option specifies the limit of search results (default is 500), `-b` option lets us specify the source(s) from which to search (yahoo, google, bing, baidu, shodan).

$ theHarvester -d pucit.edu.pk -l 100 -b yahoo

Description: As we can see from the results of the above command, we were able to find 24 email addresses. This tool does not always give result or same result on multiple runs. Another tool that you can try at your own is **hunter.io**

```
dartsec$ theHarvester -d pucit.edu.pk -b yahoo
Read proxies.yaml from /home/kali/.theHarvester/proxies.yaml
*******************************************************************
*                                                                 *
* |_|_|_  __    ^ /\ _ .__ __ __ ___  _|_ _ .__  *
* | | | \ / _) / /  (_| | \ \/ / _)  | |(/_ | *
* \_|| |_|\_| v /_/ \_,_| | \ \/\ _/\_ _/| *
*                                                                 *
* theHarvester 4.6.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] Target: pucit.edu.pk

[*] Searching Yahoo.

[*] No IPs found.

[*] Emails found: 24
_____

admissions@pucit.edu.pk
amina.mustansir@pucit.edu.pk
arif@pucit.edu.pk
asim@pucit.edu.pk
bsef20m537@pucit.edu.pk
chairman.dit@pucit.edu.pk
ejaz@pucit.edu.pk
hassankhan@pucit.edu.pk
imranj@pucit.edu.pk
info@pucit.edu.pk
kamran.malik@pucit.edu.pk
nfatimaa@pucit.edu.pk
phdcsf18m002@pucit.edu.pk
phdcsf20m004@pucit.edu.pk
principal@pucit.edu.pk
saadia.shahzad@pucit.edu.pk
shahid@pucit.edu.pk
shuja@pucit.edu.pk
swjaffry@pucit.edu.pk
umair.babar@pucit.edu.pk
waheed.iqbal@pucit.edu.pk
webmaster@pucit.edu.pk
zia@pucit.edu.pk
znawaz@pucit.edu.pk
```
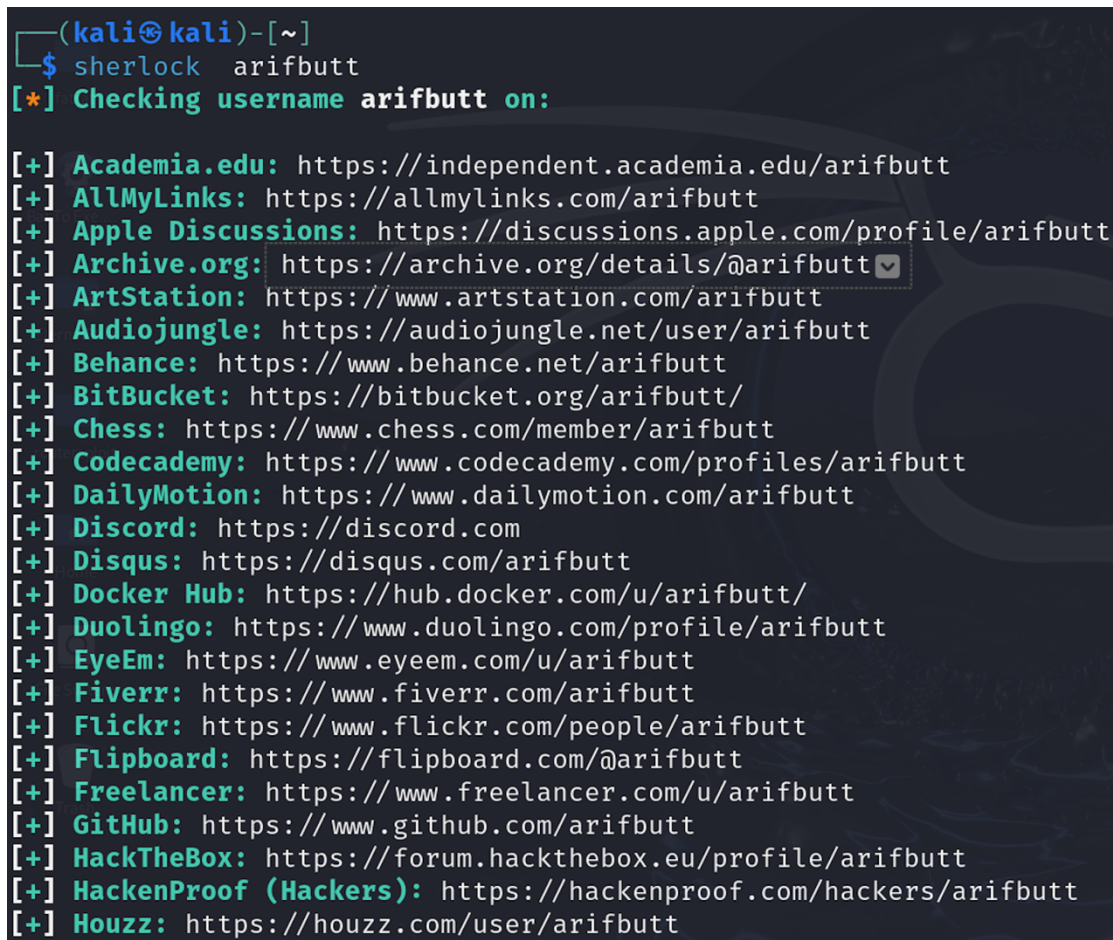
Sherlock   https://github.com/sherlock-project/sherlock

Sherlock is a command-line tool used to *search for usernames across various social media platforms and websites*. It is popular for OSINT (Open-Source Intelligence) investigations and can help users, penetration testers, or investigators find profiles associated with a specific username across a wide range of platforms. This can help in discovering potential online profiles associated with the target, which may be useful for social engineering or gathering more information about a person or organization.

Installing Steps for `sherlock` (from source):
```
$ sudo apt update
$ sudo apt install python3 python3-pip git
$ git clone https://github.com/sherlock-project/sherlock.git
$ cd sherlock
$ sudo pip3 install -r requirements.txt
$ sherlock -h
$ sherlock --version
$ sherlock <username>
```

```
┌──(kali㊸kali)-[~]
└─$ sherlock   arifbutt
[*] Checking username arifbutt on:

[+] Academia.edu: https://independent.academia.edu/arifbutt
[+] AllMyLinks: https://allmylinks.com/arifbutt
[+] Apple Discussions: https://discussions.apple.com/profile/arifbutt
[+] Archive.org: https://archive.org/details/@arifbutt
[+] ArtStation: https://www.artstation.com/arifbutt
[+] Audiojungle: https://audiojungle.net/user/arifbutt
[+] Behance: https://www.behance.net/arifbutt
[+] BitBucket: https://bitbucket.org/arifbutt/
[+] Chess: https://www.chess.com/member/arifbutt
[+] Codecademy: https://www.codecademy.com/profiles/arifbutt
[+] DailyMotion: https://www.dailymotion.com/arifbutt
[+] Discord: https://discord.com
[+] Disqus: https://disqus.com/arifbutt
[+] Docker Hub: https://hub.docker.com/u/arifbutt/
[+] Duolingo: https://www.duolingo.com/profile/arifbutt
[+] EyeEm: https://www.eyeem.com/u/arifbutt
[+] Fiverr: https://www.fiverr.com/arifbutt
[+] Flickr: https://www.flickr.com/people/arifbutt
[+] Flipboard: https://flipboard.com/@arifbutt
[+] Freelancer: https://www.freelancer.com/u/arifbutt
[+] GitHub: https://www.github.com/arifbutt
[+] HackTheBox: https://forum.hackthebox.eu/profile/arifbutt
[+] HackenProof (Hackers): https://hackenproof.com/hackers/arifbutt
[+] Houzz: https://houzz.com/user/arifbutt
```

Wafw00f

The **wafw00f** (Web Application Firewall (WAF) Foot printing Tool) is an essential tool for security professionals looking to *access the security posture of web applications*. This tool is used for following purposes:

- o **Identify WAFs:** It helps in detecting whether a web application is protected by a WAF and, if so, identifies which WAF is in use.
- o **Analyze WAF Types:** It can distinguish between various WAF vendors and types, and knowing the specific WAF can help in understanding its capabilities and limitations.
- o **Inform Security Testing:** When performing penetration testing or vulnerability assessments, knowing the WAF in place helps in crafting appropriate test cases and avoiding detection or false positives. Some WAFs have known bypass techniques, so identifying them can improve the effectiveness of security testing.
- o **Reconnaissance:** During reconnaissance, understanding the security landscape (including WAF presence) can help you make better strategies.

Installing Steps for `wafw00f` (from source):
```
$ sudo apt update
$ sudo apt install python3 python3-pip git
$ git clone https://github.com/EnableSecurity/wafw00f.git
$ cd wafw00f
$ sudo pip3 install .
$ man wafw00f
$ wafw00f <target_url >
$ wafw00f -i <urls.txt> # To check multiple URLs, specify them in text file
```

Google Hacking/Dorking

Google Dorking, also known as Google Hacking, is a technique that utilizes advance search operators to uncover information on the Internet that may not be readily available through standard search queries on Google. We know the term "hacking" suggests an illicit activity, however, Google Hacking/Dorking is entirely legal and often used by security professionals to identify vulnerabilities in their systems. These terms, when used with regular search keywords, can help us discover hidden resources crawled by Google. These resources include sensitive information such as usernames, passwords, credit card numbers, email addresses, shell scripts, user accounts, and so on.

- The `filetype:` operator allows you to search for specific file types, such as PDFs or Word documents. The following example will display only pdf files that matches the search string "Advanced Network Security"

<div align="center">

`filetype:pdf "Advanced Network Security"`

</div>
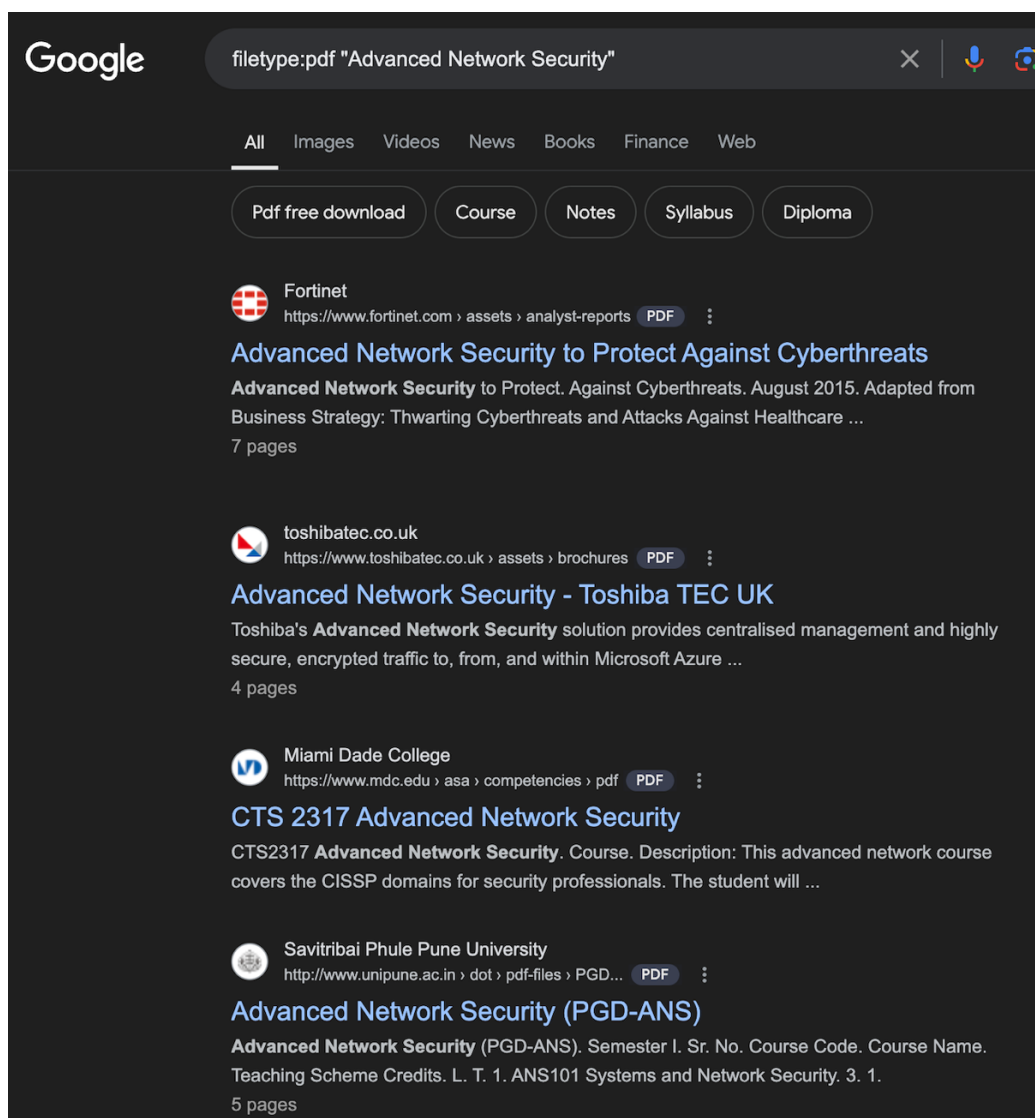
- The `inurl:` operator allows you to find specific words within the URL of a page. The following example will return only those pages that has the string `admin.php` in the URL.

$$\texttt{inurl:admin.php}$$
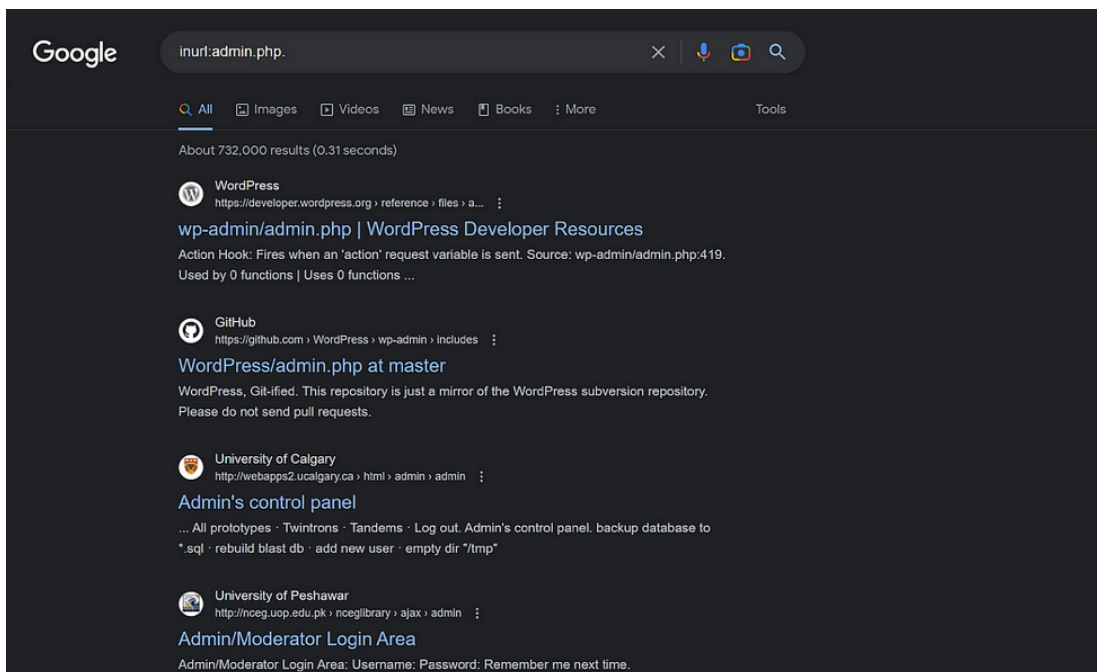
- The `intitle:` operator is used to search for specific terms in the title of a webpage. For example, the following example could revel web servers with directory listing enabled.

$$\texttt{intitle:"index of"}$$
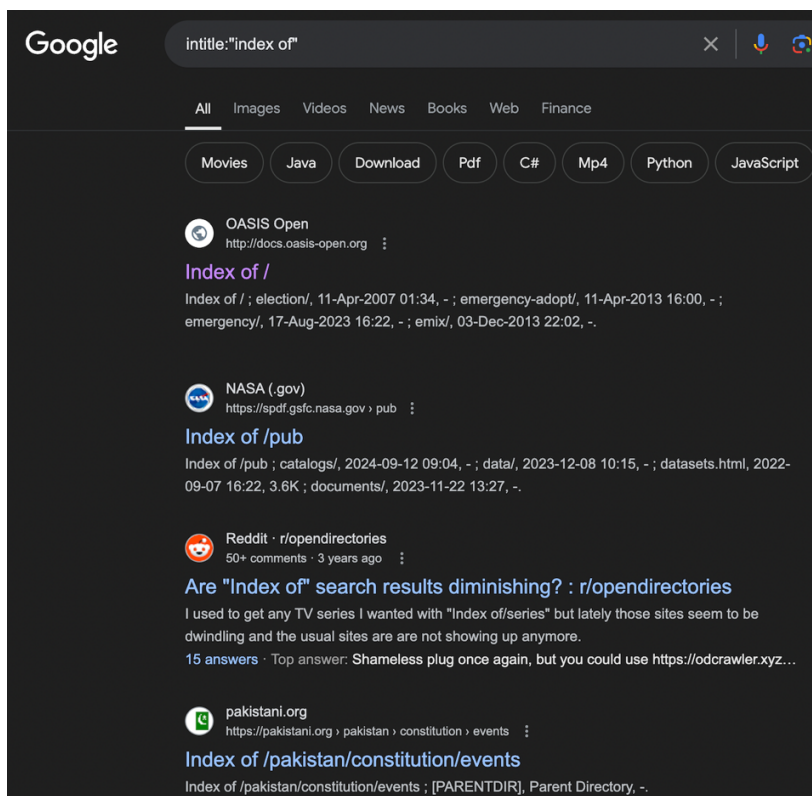
- The **link:** operator allows you to find pages that link to a specific URL. The following example will return only those pages that link to arifbutt.me domain.

<div align="center">

link:arifbutt.me

</div>
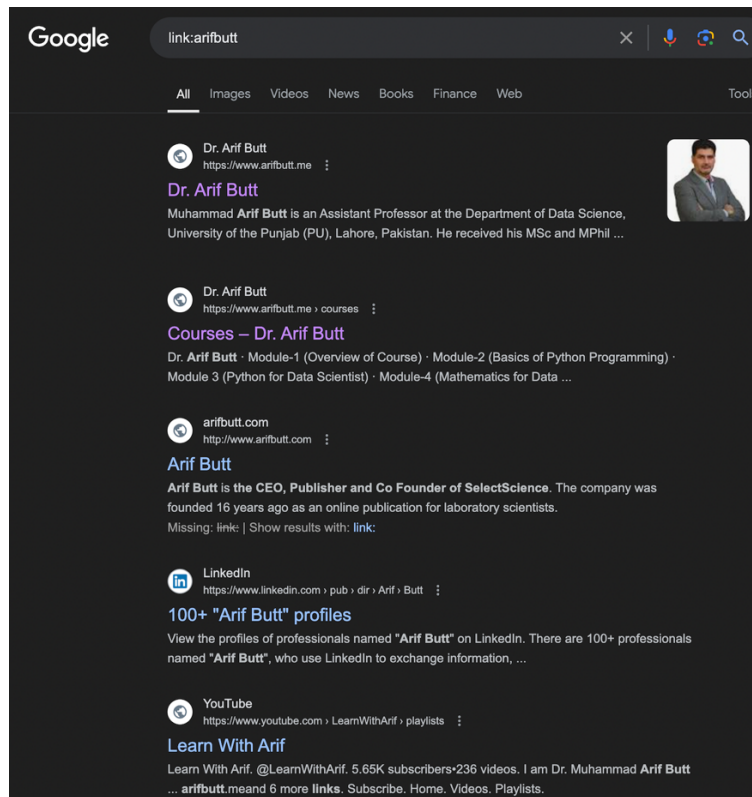
- The **site:** operator allows you to search within a specific site. For example, the combination of following two operators will display the links that has the string admin within the url and further narrow down the search to only pucit.edu.pk domain.

<div align="center">

site:pucit.edu.pk inurl:admin

</div>
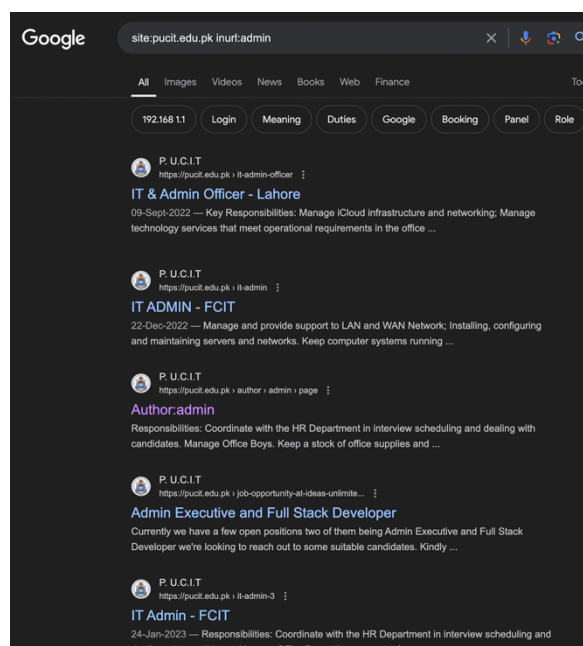
- The `intext:` operator allows you to search for specific text within the content of a web page. For example, the combination of following two operators will display the links that has the string `admin` within the text of that webpage and further narrow down the search to only `daraz.pk` domain.

site:daraz.pk intext:admin
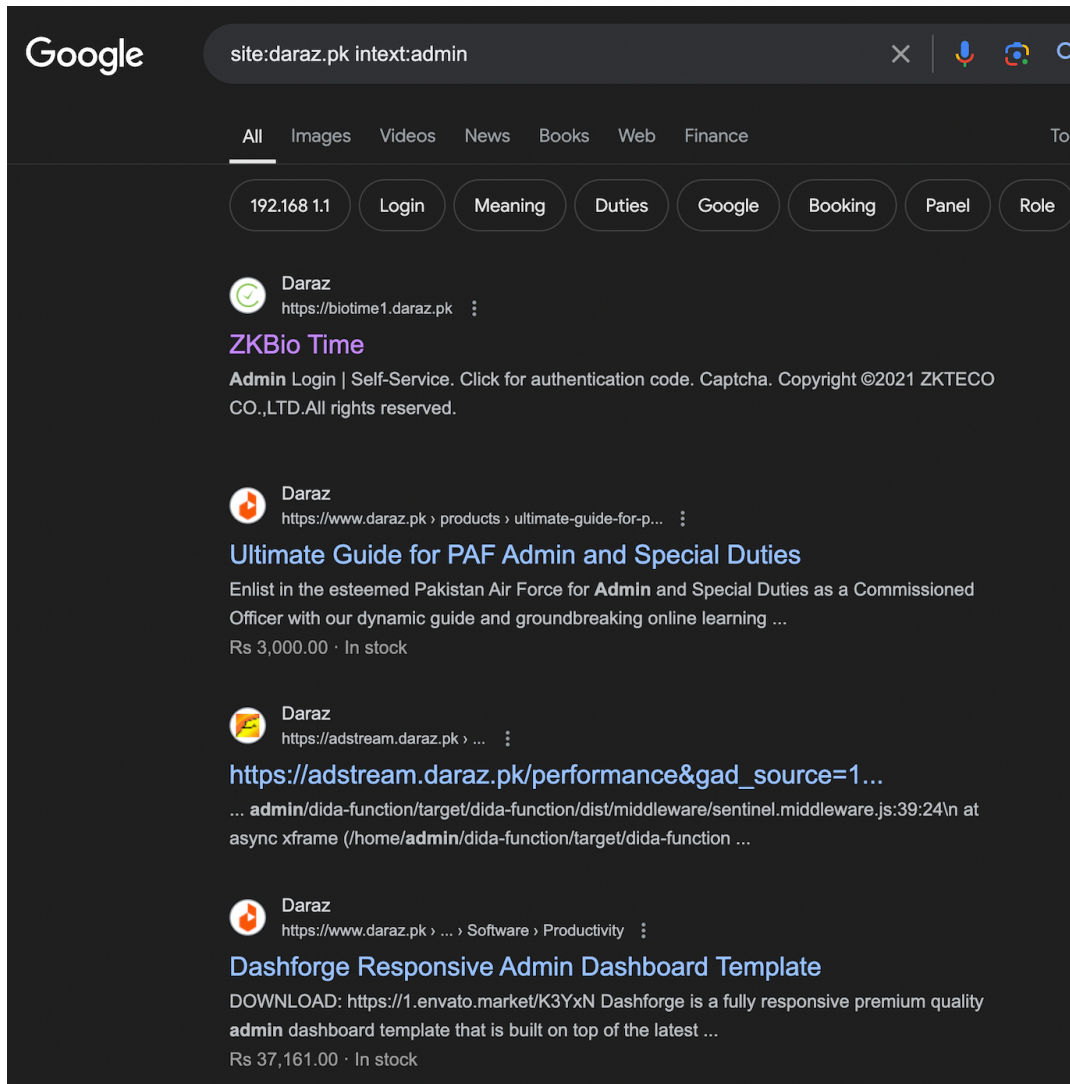
OSINT Framework

- **Open-Source Intelligence (OSINT)** refers to the process of collecting and analyzing publicly available information from open sources, such as websites, social media, public records, forums, and more. OSINT is used in various fields, including cybersecurity, law enforcement, journalism, and competitive intelligence, to uncover details about individuals, organizations, or topics. The OSINT tools and techniques can be used by anyone looking to gather information from public sources.

- The best tool for OSINT is the **OSINT Framework** (https://osintframework.com/), which is a collection of free tools, resources, and methods for conducting Open-Source Intelligence (OSINT) investigations. It organizes a variety of OSINT tools and techniques into an easy-to-navigate, categorized structure, allowing investigators, security professionals, and researchers to access information efficiently. The framework is particularly useful in gathering information from public sources related to individuals, organizations, IP addresses, domains, social media, and more.

- The OSINT Framework organizes tools into categories such as:

  - **Search Engines**: Specialized tools for web searches, metadata extraction, and search operators.

  - **People Search**: Resources for finding information about individuals (e.g., names, email addresses, phone numbers).

  - **Usernames and social media**: Tools for tracking social media profiles, posts, and activity across platforms like Twitter, Facebook, Instagram, etc. Username enumeration tools to find common aliases across various platforms.

  - **Email Addresses**: Tools to find email addresses linked to individuals or domains, verify their validity, and even search for email breaches.

  - **Domain and IP Information**: Resources for gathering information about websites, domain registrations, IP addresses, and DNS records.

  - **Public Records**: Access to government and organizational databases, like court records, property records, or financial disclosures.

  - **Geolocation**: Tools for extracting geolocation data from images or social media posts.

  - **Malware and Threat Intelligence**: Tools for analyzing malware samples, IP blacklists, and known threat actors.

  - **Metadata and File Analysis**: Helps extract metadata from files, documents, and images for investigative purposes.

  - **Dark Web Tools**: Resources to navigate or search the dark web, along with Tor-specific tools for hidden services.

(T) - Indicates a link to a tool that must be installed and run locally
(D) - Google Dork, for more information: Google Hacking
(R) - Requires registration
(M) - Indicates a URL that contains the search term and the URL itself must be edited manually
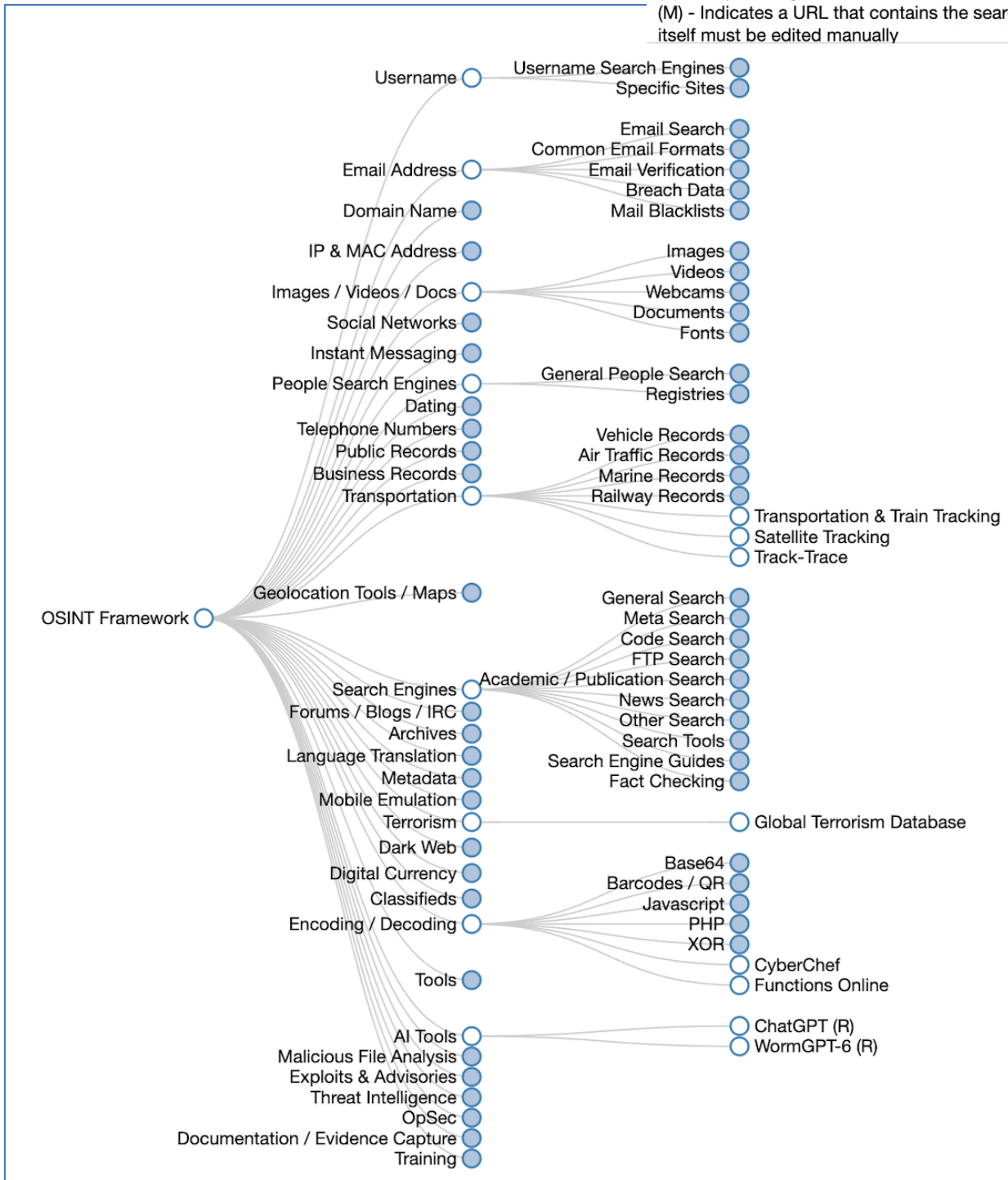
Disclaimer

The series of handouts distributed with this course are only for educational purposes. Any actions and or activities related to the material contained within this handout is solely your responsibility. The misuse of the information in this handout can result in criminal charges brought against the persons in question. The authors will not be held responsible in the event any criminal charges be brought against any individuals misusing the information in this handout to break the law.