

Kali Linux Tools

Complete Reference

800+ documented tools — install, usage & commands
compiled from official kali.org tool documentation

ICYWALL

icywall.tech

How to use this reference

Each documented tool has its own section with what it does, how to install it, and real usage / help output. Tools are sorted alphabetically and grouped by first letter — use the table of contents or your PDF viewer's search to jump to a tool. The appendix indexes every binary package shipped in Kali.

Priority legend

- **Essential** – in the `kali-tools-top10` metapackage. Learn these first.
- **High** – installed by default in standard Kali images.
- **Medium** – shipped in `kali-linux-large` / NetHunter images.
- **Low** – only in `kali-linux-everything`; install on demand.

Reading the usage blocks

Dark boxes show the install command. Framed blocks show real help output and command examples exactly as the tool prints them — lines starting `root@kali:~#` show the command that was run. Very long outputs (man pages etc.) are truncated; run the tool locally for the rest.

#

.1 Otrace

Medium Information Gathering

<https://lcamtuf.coredump.cx>

Traceroute tool that can run within an existing TCP connection The package is traceroute tool that can be run within an existing, open TCP connection, therefore bypassing some types of stateful packet filters with ease.

```
sudo apt install Otrace
```

Otrace.sh

```
root@kali:~# Otrace.sh -h
Usage: /usr/bin/Otrace.sh iface target_ip [ target_port ]
```

sendprobe

```
root@kali:~# sendprobe -h
Usage: sendprobe src_ip dst_ip sport dport seq ack
```

Suspend execution for microsecond intervals

usleep

```
root@kali:~# man usleep
usleep(3)                                Library Functions Manual                usleep(3)

NAME
    usleep - suspend execution for microsecond intervals

LIBRARY
    Standard C library (libc, -lc)

SYNOPSIS
    #include <unistd.h>

    int usleep(useconds_t usec);

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

usleep():
    Since glibc 2.12:
        (_XOPEN_SOURCE >= 500) && ! (_POSIX_C_SOURCE >= 200809L)
        || /* glibc >= 2.19: */ _DEFAULT_SOURCE
        || /* glibc <= 2.19: */ _BSD_SOURCE
    Before glibc 2.12:
        _BSD_SOURCE || _XOPEN_SOURCE >= 500

DESCRIPTION
    The usleep() function suspends execution of the calling thread for (at
    least) usec microseconds. The sleep may be lengthened slightly by any
    system activity or by the time spent processing the call or by the granu-
    larity of system timers.

RETURN VALUE
    The usleep() function returns 0 on success. On error, -1 is returned,
```

N.30 nmap

Essential Exploitation, Identification, Information Gathering, Post Exploitation, Reverse Engineering, Social Engineering, VoIP, Vulnerability Analysis, Web Applications <https://nmap.org/>

NMAP netcat reimplementatoin ncat is a reimplementatoin of Netcat by the NMAP project, providing most of the features present in the original implementations, along with some new features such as IPv6 and SSL support. Port scanning support has been removed.

```
sudo apt install ncat
```

Concatenate and redirect sockets

ncat

```
root@kali:~# ncat -h
Ncat 7.99 ( https://nmap.org/ncat )
Usage: ncat [options] [hostname] [port]

Options taking a time assume seconds. Append 'ms' for milliseconds,
's' for seconds, 'm' for minutes, or 'h' for hours (e.g. 500ms).
-4                               Use IPv4 only
-6                               Use IPv6 only
-U, --unixsock                   Use Unix domain sockets only
--vsock                         Use vsock sockets only
-C, --crlf                      Use CRLF for EOL sequence
-c, --sh-exec <command>        Executes the given command via /bin/sh
-e, --exec <command>           Executes the given command
--lua-exec <filename>          Executes the given Lua script
-g hop1[,hop2,...]              Loose source routing hop points (8 max)
-G <n>                           Loose source routing hop pointer (4, 8, 12, ...)
-m, --max-conns <n>            Maximum <n> simultaneous connections
-h, --help                     Display this help screen
-d, --delay <time>             Wait between read/writes
-o, --output <filename>        Dump session data to a file
-x, --hex-dump <filename>      Dump session data as hex to a file
-i, --idle-timeout <time>      Idle read/write timeout
-p, --source-port port          Specify source port to use
-s, --source-addr               Specify source address to use (doesn't affect -l)
-l, --listen                   Bind and listen for incoming connections
-k, --keep-open                Accept multiple connections in listen mode
-n, --nodns                    Do not resolve hostnames via DNS
-t, --telnet                   Answer Telnet negotiations
-u, --udp                      Use UDP instead of default TCP
--sctp                         Use SCTP instead of default TCP
-v, --verbose                  Set verbosity level (can be used several times)
-w, --wait <time>             Connect timeout
-z                             Zero-I/O mode, report connection status only
--append-output                Append rather than clobber specified output files
--send-only                    Only send data, ignoring received; quit on EOF
--recv-only                    Only receive data, never send anything
--no-shutdown                  Continue half-duplex when receiving EOF on stdin
-q <time>                      After EOF on stdin, wait <time> then quit.
--allow                        Allow only given hosts to connect to Ncat
--allowfile                     A file of hosts allowed to connect to Ncat
--deny                          Deny given hosts from connecting to Ncat
--denyfile                     A file of hosts denied from connecting to Ncat
--broker                       Enable Ncat's connection brokering mode
--chat                          Start a simple Ncat chat server
--proxy <addr[:port]>           Specify address of host to proxy through
--proxy-type <type>            Specify proxy type ("http", "socks4", "socks5")
--proxy-auth <auth>            Authenticate with HTTP or SOCKS proxy server
--proxy-dns <type>             Specify where to resolve proxy destination
--ssl                           Connect or listen with SSL
--ssl-cert                     Specify SSL certificate file (PEM) for listening
--ssl-key                      Specify SSL private key (PEM) for listening
--ssl-verify                   Verify trust and domain name of certificates
--ssl-trustfile                 PEM file containing trusted SSL certificates
--ssl-ciphers                  Cipherlist containing SSL ciphers to use
--ssl-servername               Request distinct server name (SNI)
--ssl-alpn                     ALPN protocol list to use
--version                      Display Ncat's version information and exit
```

See the ncat(1) manpage for full options, descriptions and usage examples

ndiff

The Network Mapper - result compare utility Ndiff is a tool to aid in the comparison of Nmap scans. It takes two Nmap XML output files and prints the differences between them: hosts coming up and down, ports becoming open or closed, and things like that. It can produce output in human-readable text or machine-readable XML formats.

```
sudo apt install ndiff
```

ndiff

```
root@kali:~# ndiff -h
```

```
Usage: /usr/bin/ndiff [option] FILE1 FILE2
Compare two Nmap XML files and display a list of their differences.
Differences include host state changes, port state changes, and changes to
service and OS detection.

-h, --help      display this help
-v, --verbose   also show hosts and ports that haven't changed.
--text         display output in text format (default)
--xml          display output in XML format
```

The Network Mapper Nmap is a utility for network exploration or security auditing. It supports ping scanning (determine which hosts are up), many port scanning techniques, version detection (determine service protocols and application versions listening behind ports), and TCP/IP fingerprinting (remote host OS or device identification). Nmap also offers flexible target and port specification, decoy/stealth scanning, sunRPC scanning, and more. Most Unix and Windows platforms are supported in both GUI and commandline modes. Several popular handheld devices are also supported, including the Sharp Zaurus and the iPAQ.

```
sudo apt install nmap
```

Network exploration tool and security / port scanner

```
root@kali:~# nmap -h
Nmap 7.99 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY/PY[portlist]: TCP SYN, TCP ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sV/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports sequentially - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
    directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
    <Lua scripts> is a comma-separated list of script-files or
    script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
# ... (56 more lines truncated -- run locally for full output)
```

Network packet generation tool / ping utility

nping

```
root@kali:~# nping -h
Nping 0.7.99 ( https://nmap.org/nping )
Usage: nping [Probe mode] [Options] {target specification}

TARGET SPECIFICATION:
  Targets may be specified as hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.*.1-24
PROBE MODES:
```

metadata. See [/usr/share/doc/metagoofil/README.md.gz](#).

```
sudo apt install metagoofil
```

```

root@kali:~# metagoofail -h
usage: metagoofail.py [-h] -d DOMAIN [-e DELAY] [-f [SAVE_FILE]]
                    [-i URL_TIMEOUT] [-l SEARCH_MAX] [-n DOWNLOAD_FILE_LIMIT]
                    [-o SAVE_DIRECTORY] [-r NUMBER_OF_THREADS] -t FILE_TYPES
                    [-u [USER_AGENT]] [-w]

Metagoofail v1.2.0 - Search Google and download specific file types.

options:
-h, --help                show this help message and exit
-d DOMAIN                 Domain to search.
-e DELAY                  Delay (in seconds) between searches. If it's too small
                        Google may block your IP, too big and your search may
                        take a while. Default: 30.0
-f [SAVE_FILE]            Save the html links to a file.
                        no -f = Do not save links
                        -f = Save links to html_links_<TIMESTAMP>.txt
                        -f SAVE_FILE = Save links to SAVE_FILE
-i URL_TIMEOUT            Number of seconds to wait before timeout for
                        unreachable/stale pages. Default: 15
-l SEARCH_MAX             Maximum results to search. Default: 100
-n DOWNLOAD_FILE_LIMIT    Maximum number of files to download per filetype.
                        Default: 100
-o SAVE_DIRECTORY         Directory to save downloaded files. Default is current
                        working directory, "."
-r NUMBER_OF_THREADS      Number of downloader threads. Default: 8
-t FILE_TYPES             file_types to download
                        (pdf,doc,xls,ppt,odp,ods,docx,xlsx,pptx). To search
                        all 17,576 three-letter file extensions, type "ALL"
-u [USER_AGENT]           User-Agent for file retrieval against -d domain.
                        no -u = "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
                        -u = Randomize User-Agent
                        -u "My custom user agent 2.0" = Your customized User-Agent
-w                        Download the files, instead of just viewing search
                        results.

```

```
root@kali:~# metagoofil -d kali.org -t pdf -l 100 -n 25 -o kalipdf -f kalipdf.html
```

```
*****
 *      /\ \   _ __ | | ___| |__| |___) | *
 *     /  \/  \_||_| | | | | | |___) | *
 *    /  /\ \   _ || | | | | | |___) | *
 *   V  /  \  \_\_\_| |_| | | | |___) | *
 *                               |___/  *
 * Metagoofil Ver 2.2                *
 * Christian Martorella              *
 * Edge-Security.com                 *
 * cmartorella_at_edge-security.com *
 *****
 [pdf']

[-] Starting online search...

[-] Searching for pdf files, with a limit of 100
    Searching 100 results...
Results: 21 files found
Starting to download 25 of them:
```

M.22 metasploit-framework

Essential Exploitation, Identification, Information Gathering, Post Exploitation, Reverse Engineering, Social Engineering, Vulnerability Analysis, Web Applications <https://www.metasploit.com/>

Framework for exploit development and vulnerability research The Metasploit Framework is an open source platform that supports vulnerability research, exploit development, and the creation of custom security tools.

```
sudo apt install metasploit-framework
```

msf-egg hunter

```

root@kali:~# msf-egghunter -h
Usage: msf-egghunter [options]
Example: msf-egghunter -f python -e WOOT

Specific options:
  -f, --format <String>          See --list-formats for a list of supported output formats
  -b, --badchars <String>       (Optional) Bad characters to avoid for the egg
  -e, --egg <String>            The egg (Please give 4 bytes)
  -p, --platform <String>       (Optional) Platform
  --startreg <String>           (Optional) The starting register
  --forward                      (Optional) To search forward
  --depreg <String>             (Optional) The DEP register
  --depdest <String>           (Optional) The DEP destination
  --depsize <Integer>          (Optional) The DEP size
  --depmethod <String>         (Optional) The DEP method to use (virtualprotect/virtualalloc/copy/copy_size)

```

-a, --arch <String>	(Optional) Architecture
--list-formats	List all supported output formats
-v, --var-name <name>	(Optional) Specify a custom variable name to use for certain output formats
-h, --help	Show this message

msf-exe2vba

```
root@kali:~# msf-exe2vba -h
Usage: msf-exe2vba [exe] [vba]
```

msf-exe2vbs

```
root@kali:~# msf-exe2vbs -h
Usage: msf-exe2vbs [exe] [vbs]
```

msf-find_badchars

```
root@kali:~# msf-find_badchars -h

Usage: msf-find_badchars <options>

OPTIONS:

-b The list of characters to avoid: '\x00\xff'
-h Help banner
-i Read memory contents from the supplied file path
-t The format that the memory contents are in (empty to list)
```

msf-halfml_second

```
root@kali:~# msf-halfml_second -h

Usage: msf-halfml_second <options>

OPTIONS:

-h Display this help information
-n The encrypted LM hash to crack
-p The decrypted LANMAN password for bytes 1-7
-s The server challenge (default value 1122334455667788)
```

msf-hmac_shai_crack

```
root@kali:~# msf-hmac_shai_crack -h

Usage: msf-hmac_shai_crack hashes.txt <wordlist | - >
The format of hash file is <identifier>:<hex-salt>:<hash>
```

msf-java_deserializer

```
root@kali:~# msf-java_deserializer -h
Usage: msf-java_deserializer <file> [option]
-a, --array=ID          Print detailed information about content array
-o, --object=ID         Print detailed information about content object
-h, --help              Prints this help
```

msf-jsobfu

```
root@kali:~# msf-jsobfu -h
Usage: msf-jsobfu [options]

Specific options:
-t, --iteration <Integer>      Number of times to obfuscate the JavaScript
-i, --input <String>          The JavaScript file you want to obfuscate (default=1)
-o, --output <String>         Save the obfuscated file as
-p id1,id2,                   The identifiers to preserve
  --preserved-identifiers
-h, --help                    Show this message
```

msf-md5_lookup

```
root@kali:~# msf-md5_lookup -h
Usage: msf-md5_lookup [options]

Specific options:
-i, --input <file>            The file that contains all the MD5 hashes (one line per hash)
-d, --databases <names>      (Optional) Select databases: all, authsecu, i337, md5_my_addr, md5_net, md5crack, md5cracker, md5decryption,
  md5online, md5pass, netmd5crack, tmt0 (Default=all)
-o, --out <filepath>         (Optional) Save the results to a file (Default=md5_results.txt)
-h, --help                    Show this message
```

msf-metasm_shell

```
root@kali:~# msf-metasm_shell -h

Usage: msf-metasm_shell <options>

OPTIONS:

-a The architecture to encode as (ARM, Ia32, MIPS, X86_64)
```

```
sudo apt install winregfs
```

Check a Windows registry file

fsck.winregfs

```
root@kali:~# fsck.winregfs -h
Windows Registry Hive File Checker 0.8 (2021-09.15)

Usage: fsck.winregfs [options] hivename
```

Windows registry FUSE filesystem

mount.winregfs

```
root@kali:~# mount.winregfs -h
Windows Registry Filesystem 0.8 (2021-09.15)

Usage: mount.winregfs [-o ro] [fuse_options] hivename mountpoint
```

W.29 wireshark

Essential Wireless (802.11), Forensics, Incident Response, Sniffing & Spoofing, VoIP, Web Applications, Wireless <https://www.wireshark.org/>

Network packet dissection library – data files The libwireshark library provides the network packet dissection services developed by the Wireshark project. This package contains the platform independent files.

```
sudo apt install libwireshark-data
```

libwireshark-dev

Network packet dissection library – development files The "libwireshark" library provides the network packet dissection services developed by the Wireshark project. This package contains the static library and the C header files that are needed for applications to use libwireshark services.

```
sudo apt install libwireshark-dev
```

libwireshark19

Network packet dissection library – shared library The libwireshark library provides the network packet dissection services developed by the Wireshark project.

```
sudo apt install libwireshark19
```

libwiretap-dev

Network packet capture library – development files Wiretap, part of the Wireshark project, is a library that allows one to read and write several packet capture file formats. Supported formats are: * Libpcap * Sniffer * LANalyzer * Network Monitor * "snoop" * "iptrace" * Sniffer Basic (NetXRay)/Windows Sniffer Pro * RADCOM WAN/LAN Analyzers * Lucent/Ascend access products * HP-UX nettl * Toshiba ISDN Router * ISDN4BSD "i4btrace" utility * Cisco Secure Intrusion Detection System iplogging facility * pppd logs (pppdump-format files) * VMS TCPTRACE * DBS Etherwatch (text format) * Catapult DCT2000 (.out files) Wiretap's shortcomings are: no filter capability and no support for packet capture. This package contains the static library and the C header files.

```
sudo apt install libwiretap-dev
```

libwiretap16

Network packet capture library – shared library Wiretap, part of the Wireshark project, is a library that allows one to read and write several packet capture file formats. Supported formats are: * Libpcap * Sniffer * LANalyzer * Network Monitor * "snoop" * "iptrace" * Sniffer Basic (NetXRay)/Windows Sniffer Pro * RADCOM WAN/LAN Analyzers * Lucent/Ascend access products * HP-UX nettl * Toshiba ISDN Router * ISDN4BSD "i4btrace" utility * Cisco Secure Intrusion Detection System iplogging facility * pppd logs (pppdump-format files) * VMS TCPTRACE * DBS Etherwatch (text format) * Catapult DCT2000 (.out files) Wiretap's shortcomings are: no filter capability and no support for packet capture.

```
sudo apt install libwiretap16
```

libwsutil-dev

Network packet dissection utilities library – development files The libwsutil library provides utility functions for Wireshark

and related binaries and shared libraries. This package contains the static library and the C header files that are needed for applications to use the libwsutil library.

```
sudo apt install libwsutil-dev
```

libwsutil17

Network packet dissection utilities library – shared library The libwsutil library provides utility functions for Wireshark and related binaries and shared libraries.

```
sudo apt install libwsutil17
```

stratoshark

System call and log analyzer - graphical and console interfaces Stratoshark is a system call and log analyzer. It can capture and analyze system calls on Linux and capture and analyze log data from other sources. The package provides stratoshark as the graphical interface and strato as the console interface.

```
sudo apt install stratoshark
```

Dump and analyze system calls and event logs

strato

```
root@kali:~# strato -h
strato (Stratoshark) 0.9.6
Dump and analyze network traffic.
See https://www.wireshark.org for more information.

Usage: strato [options] ...

Input file:
  -r <infile>, --read-file <infile>
                                set the filename to read from (or '-' for stdin)

Processing:
  -2                             perform a two-pass analysis
  -M <packet count>              perform session auto reset
  -R <read filter>, --read-filter <read filter>
                                packet Read filter in Wireshark display filter syntax
                                (requires -2)
  -Y <display filter>, --display-filter <display filter>
                                packet display filter in Wireshark display filter
                                syntax
  -n                             disable all name resolutions (def: "mNd" enabled, or
                                as set in preferences)
  -N <name resolve flags>        enable specific name resolution(s): "mtndsNvg"
  -d <layer_type>==<selector>,<decode_as_protocol> ...
                                "Decode As", see the man page for details
                                Example: tcp.port==8888,http
  -H <hosts file>               read a list of entries from a hosts file, which will
                                then be written to a capture file. (Implies -W n)
  --enable-protocol <proto_name>
                                enable dissection of proto_name
  --disable-protocol <proto_name>
                                disable dissection of proto_name
  --only-protocols <protocols>
                                Only enable dissection of these protocols, comma
                                separated. Disable everything else
  --disable-all-protocols
                                Disable dissection of all protocols
  --enable-heuristic <short_name>
                                enable dissection of heuristic protocol
  --disable-heuristic <short_name>
                                disable dissection of heuristic protocol

Output:
  -w <outfile|->                write packets to a pcapng-format file named "outfile"
                                (or '-' for stdout). If the output filename has the
                                .gz extension, it will be compressed to a gzip archive
  --capture-comment <comment>
                                add a capture file comment, if supported
  -C <config profile>           start with specified configuration profile
  --global-profile              use the global profile instead of personal profile
  -F <output file type>         set the output file type; default is pcapng.
                                an empty "-F" option will list the file types
  -V                             add output of packet tree (Packet Details)
  -O <protocols>               Only show packet details of these protocols, comma
                                separated
  -P, --print                  print packet summary even when writing to a file
  -S <separator>               the line separator to print between packets
  -x                             add output of hex and ASCII dump (Packet Bytes)
  --hexdump <hexoption>        add hexdump, set options for data source and ASCII dump
                                all
                                dump all data sources (-x default)
                                frames
                                dump only frame data source
# ... (78 more lines truncated -- run locally for full output)
```

System call and event log analyzer

stratoshark

```

root@kali:~# bully -h

bully v1.4
the fork that actually works!
maintained by kimocoder - https://twitter.com/kimocoder

usage: bully <options> interface

Required arguments:

    interface      : Wireless interface in monitor mode (root required)

Or
    -b, --bssid macaddr  : MAC address of the target access point
    -e, --essid string    : Extended SSID for the access point

Optional arguments:

    -c, --channel N[,N...] : Channel number of AP, or list to hop [b/g]
    -i, --index N          : Starting pin index (7 or 8 digits) [Auto]
    -l, --lockwait N       : Seconds to wait if the AP locks WPS [43]
    -o, --outfile file     : Output file for messages [stdout]
    -p, --pin N            : Starting pin number (7 or 8 digits) [Auto]
    -s, --source macaddr   : Source (hardware) MAC address [Probe]
    -u, --lua              : Lua script file
    -v, --verbosity N      : Verbosity level 1-4, 1 is quietest [3]
    -w, --workdir path     : Location of pin/session files [~/bully/]
    -5, --5ghz             : Hop on 5GHz a/n default channel list [No]
    -B, --bruteforce       : Bruteforce the WPS pin checksum digit [No]
    -F, --force            : Force continue in spite of warnings [No]
    -S, --sequential       : Sequential pins (do not randomize) [No]
    -T, --test             : Test mode (do not inject any packets) [No]

Advanced arguments:

    -d, --pixiewps         : Attempt to use pixiewps [No]
    -a, --acktime N        : Deprecated/ignored [Auto]
    -r, --retries N        : Resend packets N times when not acked [2]
    -m, --m13time N       : Deprecated/ignored [Auto]
    -t, --timeout N        : Deprecated/ignored [Auto]
    -1, --pin1delay M,N    : Delay M seconds every Nth nack at M5 [0,1]
    -2, --pin2delay M,N    : Delay M seconds every Nth nack at M7 [5,1]
    -A, --noacks           : Disable ACK check for sent packets [No]
    -C, --nocheck          : Skip CRC/FCS validation (performance) [No]
    -D, --detectlock       : Detect WPS lockouts unreported by AP [No]
    -E, --eapfail          : EAP Failure terminate every exchange [No]
    -L, --lockignore       : Ignore WPS locks reported by the AP [No]
    -M, --m57nack          : M5/M7 timeouts treated as WSC_NACK's [No]
    -N, --nofcs            : Packets don't contain the FCS field [Auto]
    -P, --probe            : Use probe request for nonbeaconing AP [No]
    -Q, --wpsinfo          : Use probe request to gather WPS info [No]
    -R, --radiotap         : Assume radiotap headers are present [Auto]
    -W, --windows7         : Masquerade as a Windows 7 registrar [No]
    -Z, --suppress         : Suppress packet throttling algorithm [No]
    -V, --version          : Print version info and exit
    -h, --help             : Display this help information

```

```

root@kali:~# bully -e 6F36E6 wlan0mon
[!] Bully v1.0-22 - WPS vulnerability assessment utility
[X] Unknown frequency '-113135872' reported by interface 'mon0'
[!] Using '00:1f:33:f3:51:13' for the source MAC address
[+] Datalink type set to '127', radiotap headers present
[+] Scanning for beacon from '6F36E6' on channel 'unknown'
[+] Got beacon for '6F36E6' (9c:d3:6d:b8:ff:56)
[+] Switching interface 'mon0' to channel '8'
[!] Beacon information element indicates WPS is locked
[!] Creating new randomized pin file '/root/.bully/pins'
[+] Index of starting pin number is '00000000'
[+] Last State = 'NoAssoc' Next pin '54744431'

```

B.33 burpsuite

Essential Web Applications

<https://portswigger.net>

Platform for security testing of web applications Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities. Burp gives you full control, letting you combine advanced manual techniques with state-of-the-art automation, to make your work faster, more effective, and more fun.

```
sudo apt install burpsuite
```

```

root@kali:~# burpsuite --help
Usage:
--help          Print this message
--version       Print version details
--disable-extensions Prevent loading of extensions on startup
--diagnostics   Print diagnostic information
--use-defaults  Start with default settings
--collaborator-server Run in Collaborator server mode
--collaborator-config Specify Collaborator server configuration file; defaults to collaborator.config
--data-dir      Specify data directory
--project-file  Open the specified project file; this will be created as a new project if the file does not exist

```

afl-clang-fast Afl-clang-fast++ (8) - afl-cc++5.02c by Michal Zalewski, Laszlo Szekeres, Marc Heuse afl-cc

afl-clang-fast++ Afl-cc++5.02c by Michal Zalewski, Laszlo Szekeres, Marc Heuse afl-cc

afl-clang-lto Afl-clang-lto++ (8) - afl-cc++5.02c by Michal Zalewski, Laszlo Szekeres, Marc Heuse afl-cc

afl-clang-lto++ (unknown subject)

afl-cmin (unknown subject)

afl-cmin.awk (unknown subject)

afl-cmin.bash (unknown subject)

afl-cmin.py (unknown subject)

afl-fuzz afl-g++ Afl-g++-fast (8) - afl-cc++5.02c by Michal Zalewski, Laszlo Szekeres, Marc Heuse afl-cc

afl-g++-fast afl-gcc Afl-cc++5.02c by Michal Zalewski, Laszlo Szekeres, Marc Heuse afl-cc

afl-gcc-fast (unknown subject)

afl-gotcpu (unknown subject)

afl-health Afl-cc++5.02c by Michal Zalewski, Laszlo Szekeres, Marc Heuse afl-cc

afl-lto Afl-lto++ (8) - afl-cc++5.02c by Michal Zalewski, Laszlo Szekeres, Marc Heuse afl-cc

afl-lto++ afl-network-client afl-network-server (unknown subject)

afl-persistent-config (unknown subject)

afl-plot (unknown subject)

afl-showmap (unknown subject)

afl-system-config (unknown subject)

afl-tmin (unknown subject)

afl-whatsup afl+-doc

Instrumentation-driven fuzzer for binary formats - documentation American fuzzy lop is a fuzzer that employs compile-time instrumentation and genetic algorithms to automatically discover clean, interesting test cases that trigger new internal states in the targeted binary. This package provides the documentation, a collection of special crafted test cases, vulnerability samples and experimental stuff.

```
sudo apt install afl+-doc
```

Additional usage output omitted – run the tool locally for the full reference.

A.8 aircrack-ng

Essential Wireless (802.11), Exploitation, Sniffing & Spoofing, Social Engineering, Wireless

<https://www.aircrack-ng.org/>

Wireless WEP/WPA cracking utilities aircrack-ng is an 802.11a/b/g WEP/WPA cracking program that can recover a 40-bit, 104-bit, 256-bit or 512-bit WEP key once enough encrypted packets have been gathered. Also it can attack WPA1/2 networks with some advanced methods or simply by brute force. It implements the standard FMS attack along with some optimizations, thus making the attack much faster compared to other WEP cracking tools. It can also fully use a multiprocessor system to its full power in order to speed up the cracking process. aircrack-ng is a fork of aircrack, as that project has been stopped by the upstream maintainer.

```
sudo apt install aircrack-ng
```

Multi-purpose tool aimed at attacking clients as opposed to the Access Point (AP) itself

airbase-ng

```
root@kali:~# airbase-ng --help
Airbase-ng 1.7 - (C) 2008-2022 Thomas d'Otreppe
Original work: Martin Beck
https://www.aircrack-ng.org
```

```
usage: airbase-ng <options> <replay interface>

Options:
  -a bssid      : set Access Point MAC address
  -i iface      : capture packets from this interface
  -w WEP key    : use this WEP key to en-/decrypt packets
  -h MAC        : source mac for MITM mode
  -f disallow   : disallow specified client MACs (default: allow)
  -W 0|1       : [don't] set WEP flag in beacons 0|1 (default: auto)
  -q           : quiet (do not print statistics)
  -v           : verbose (print more messages)
  -A           : Ad-Hoc Mode (allows other clients to peer)
  -Y in|out|both : external packet processing
  -c channel    : sets the channel the AP is running on
  -X           : hidden ESSID
  -s           : force shared key authentication (default: auto)
  -S           : set shared key challenge length (default: 128)
  -L           : Caffe-Latte WEP attack (use if driver can't send frags)
  -N           : cfrag WEP attack (recommended)
  -x nbpps      : number of packets per second (default: 100)
  -y           : disables responses to broadcast probes
  -O           : set all WPA,WEP,open tags. can't be used with -z & -Z
  -z type       : sets WPA1 tags. 1=WEP40 2=TKIP 3=WRAP 4=CCMP 5=WEP104
  -Z type       : same as -z, but for WPA2
  -V type       : fake EAPOL 1=MD5 2=SHA1 3=auto
  -F prefix     : write all sent and received frames into pcap file
  -P           : respond to all probes, even when specifying ESSIDs
  -I interval   : sets the beacon interval value in ms
  -C seconds    : enables beaconing of probed ESSID values (requires -P)
  -n hex        : User specified ANonce when doing the 4-way handshake

Filter options:
  --bssid MAC    : BSSID to filter/use
  --bssids file  : read a list of BSSIDs out of that file
  --client MAC   : MAC of client to filter
  --clients file : read a list of MACs out of that file
  --essid ESSID  : specify a single ESSID (default: default)
  --essids file  : read a list of ESSIDs out of that file

  --help        : Displays this usage screen
```

A 802.11 WEP / WPA-PSK key cracker

```
root@kali:~# aircrack-ng --help

Aircrack-ng 1.7 - (C) 2006-2022 Thomas d'Otreppe
https://www.aircrack-ng.org

usage: aircrack-ng [options] <input file(s)>

Common options:

  -a <amode> : force attack mode (1/WEP, 2/WPA-PSK)
  -e <essid> : target selection: network identifier
  -b <bssid> : target selection: access point's MAC
  -p <nbcpu> : # of CPU to use (default: all CPUs)
  -q        : enable quiet mode (no status output)
  -C <macs> : merge the given APs to a virtual one
  -l <file> : write key to file. Overwrites file.

Static WEP cracking options:

  -c      : search alphanumeric characters only
  -t      : search binary coded decimal chr only
  -h      : search the numeric key for Fritz!BOX
  -d <mask> : use masking of the key (A1:XX:CF:YY)
  -m <maddr> : MAC address to filter usable packets
  -n <nbits> : WEP key length : 64/128/152/256/512
  -i <index> : WEP key index (1 to 4), default: any
  -f <fudge> : bruteforce fudge factor, default: 2
  -k <korek> : disable one attack method (1 to 17)
  -x or -x0 : disable bruteforce for last keybytes
  -x1      : last keybyte bruteforcing (default)
  -x2      : enable last 2 keybytes bruteforcing
  -X       : disable bruteforce multithreading
  -y       : experimental single bruteforce mode
  -K       : use only old KoreK attacks (pre-PTW)
  -s       : show the key in ASCII while cracking
  -M <num> : specify maximum number of IVs to use
  -D       : WEP decloak, skips broken keystreams
  -P <num> : PTW debug: 1: disable Klein, 2: PTW
  -l       : run only 1 try to crack key with PTW
  -V       : run in visual inspection mode

WEP and WPA-PSK cracking options:

  -w <words> : path to wordlist(s) filename(s)
  -N <file>  : path to new session filename
  -R <file>  : path to existing session filename

WPA-PSK options:

  -E <file> : create EWSA Project file v3
  -l <str>  : PMKID string (hashcat -m 16800)
  -j <file> : create Hashcat v3.6+ file (HCCAPX)
  -J <file> : create Hashcat file (HCCAP)
  -S       : WPA cracking speed test
  -Z <sec> : WPA cracking speed test length of execution.
  -r <DB>  : path to airolib-ng database
```

```
--color|--nocolor    => enable/disable colored output [default is ENABLED]
--corelan            => display corelan reference
--help              => displays help
--man               => displays extended help with examples
--version           => displays version information

-s                 => suppress (display result only)
-f|--file <file1>,<file2> => use file(s) as input
# ... (1 more lines truncated -- run locally for full output)
```

```
root@kali:~# hURL -b "S2FsaSBMaW51eAo="

Original string      :: S2FsaSBMaW51eAo=
base64 DEcoded string :: Kali Linux
```

H.37 hydra

Essential Information Gathering, Password Attacks, Vulnerability Analysis, Web Applications <https://github.com/vanhauser-thc/thc-hydra>

Very fast network logon cracker Hydra is a parallelized login cracker which supports numerous protocols to attack. It is very fast and flexible, and new modules are easy to add. This tool makes it possible for researchers and security consultants to show how easy it would be to gain unauthorized access to a system remotely. It supports: Cisco AAA, Cisco auth, Cisco enable, CVS, FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET, HTTP(S)-HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB(NT), SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP.

```
sudo apt install hydra
```

Generates a (d)efault (p)assword (l)ist as input for THC hydra

dp14hydra

```
root@kali:~# dp14hydra -h
dp14hydra v0.9.9 (c) 2012 by Roland Kessler (@rokkessler)

Syntax: dp14hydra [help] | [refresh] | [BRAND] | [all]

This script depends on a local (d)efault (p)assword (l)ist called
/root/.dp14hydra/dp14hydra_full.csv. If it is not available, regenerate it with
'dp14hydra refresh'. Source of the default password list is
http://open-sez.me

Options:
  help      Help: Show this message
  refresh   Refresh list: Download the full (d)efault (p)assword (l)ist
            and generate a new local /root/.dp14hydra/dp14hydra_full.csv file. Takes time!
  BRAND     Generates a (d)efault (p)assword (l)ist from the local file
            /root/.dp14hydra/dp14hydra_full.csv, limiting the output to BRAND systems, using
            the format username:password (as required by THC hydra).
            The output file is called dp14hydra_BRAND.lst.
  all       Dump list of all systems credentials into dp14hydra_all.lst.

Example:
# dp14hydra linksys
File dp14hydra_linksys.lst was created with 20 entries.
# hydra -C ./dp14hydra_linksys.lst -t i 192.168.1.1 http-get /index.asp
```

A very fast network logon cracker which supports many different services

```
root@kali:~# hydra -h
Hydra v9.7 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this
is non-binding, these *** ignore laws and ethics anyway).

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT]
] [-x MIN:MAX:CHARSET] [-c TIME] [-IS0uvVd46] [-m MODULE_OPT] [service://server[:PORT]/[OPT]]

Options:
-R      restore a previous aborted/crashed session
-I      ignore an existing restore file (don't wait 10 seconds)
-S      perform an SSL connect
-s PORT if the service is on a different default port, define it here
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-x MIN:MAX:CHARSET password brute-force generation, type "-x -h" to get help
-y      disable use of symbols in brute-force, see above
-r      use a non-random shuffling method for option -x
-e nsr   try "n" null password, "s" login as pass and/or "r" reversed login
-u      loop around users, not passwords (effective! implied with -x)
-C FILE  colon separated "login:pass" format, instead of -L/-P options
-M FILE  list of servers to attack, one entry per line, ':' to specify port
-D XofY  Divide wordlist into Y segments and use the Xth segment.
-o FILE  write found login/password pairs to FILE instead of stdout
-b FORMAT specify the format for the -o FILE: text(default), json, jsonv1
-f / -F  exit when a login/pass pair is found (-M: -f per host, -F global)
-t TASKS run TASKS number of connects in parallel per target (default: 16)
-T TASKS run TASKS connects in parallel overall (for -M, default: 64)
```

```

-w / -W TIME wait time for a response (32) / between connects per thread (0)
-c TIME wait time per login attempt over all threads (enforces -t 1)
-4 / -6 use IPv4 (default) / IPv6 addresses (put always in [] also in -M)
-v / -V / -d verbose mode / show login+pass for each attempt / debug mode
-O use old SSL v2 and v3
-K do not redo failed attempts (good for -M mass scanning)
-q do not print messages about connection errors
-U service module usage details
-m OPT options specific for a module, see -U output for information
-h more command line options (COMPLETE HELP)
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy
http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mongodb mssql mysql nntp oracle-listener oracle-sid panywhere
pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smb2 smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak
telnet[s] vmauthd vnc xmpm

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)
These services were not compiled in: afp ncp oracle sapr3.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.
E.g. % export HYDRA_PROXY=socks5://l:p@127.0.0.1:9150 (or: socks4:// connect://)
% export HYDRA_PROXY=connect_and_socks_proxylist.txt (up to 64 entries)
% export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080
% export HYDRA_PROXY_HTTP=proxylist.txt (up to 64 entries)

Examples:
hydra -l user -P passlist.txt ftp://192.168.0.1
hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS/DIGEST-MD5
hydra -l admin -p password ftp://[192.168.0.0/24]/
# ... (1 more lines truncated -- run locally for full output)

```

Wizard to use hydra from command line

hydra-wizard

```

root@kali:~# man hydra-wizard
HYDRA-WIZARD(1)          General Commands Manual          HYDRA-WIZARD(1)

NAME
    HYDRA-WIZARD - Wizard to use hydra from command line

DESCRIPTION
    This script guide users to use hydra, with a simple wizard that will make
    the necessary questions to launch hydra from command line a fast and eas-
    ily
    1. The wizard ask for the service to attack
    2. The target to attack
    3. The username o file with the username what use to attack
    4. The password o file with the passwords what use to attack
    5. The wizard ask if you want to test for passwords same as login, null or
    reverse login
    6. The wizard ask for the port number to attack
    Finally, the wizard show the resume information of attack, and ask if you
    want launch attack

SEE ALSO
    hydra(1), dpl4hydra(1),

AUTHOR
    hydra-wizard was written by Shivang Desai <shivang.ice.2010@gmail.com>.

    This manual page was written by Daniel Echeverry <epsilon77@gmail.com>,
    for the Debian project (and may be used by others).

19/01/2014          HYDRA-WIZARD(1)

```

A tool to reduce the password list

pw-inspector

```

root@kali:~# pw-inspector -h
PW-Inspector v0.2 (c) 2005 by van Hauser / THC vh@thc.org [https://github.com/vanhauser-thc/thc-hydra]

Syntax: pw-inspector [-i FILE] [-o FILE] [-m MINLEN] [-M MAXLEN] [-c MINSETS] -l -u -n -p -s

Options:
  -i FILE    file to read passwords from (default: stdin)
  -o FILE    file to write valid passwords to (default: stdout)
  -m MINLEN  minimum length of a valid password
  -M MAXLEN  maximum length of a valid password
  -c MINSETS the minimum number of sets required (default: all given)

Sets:
  -l        lowercase characters (a,b,c,d, etc.)
  -u        upcase characters (A,B,C,D, etc.)
  -n        numbers (1,2,3,4, etc.)
  -p        printable characters (which are not -l/-u/-n, e.g. $,!/,(*, etc.)
  -s        special characters - all others not within the sets above

PW-Inspector reads passwords in and prints those which meet the requirements.
The return code is the number of valid passwords found, 0 if none was found.
Use for security: check passwords, if 0 is returned, reject password choice.

```

```
sudo apt install sqlitebrowser
```

Light GUI editor for SQLite databases

```
root@kali:~# man sqlitebrowser
SQLITEBROWSER(1)          General Commands Manual          SQLITEBROWSER(1)

NAME
  sqlitebrowser - light GUI editor for SQLite databases

SYNOPSIS
  sqlitebrowser [file]

DESCRIPTION
  SQLite Database Browser is a visual tool used to create, design and edit
  database files compatible with SQLite. It is meant to be used for users
  and developers that want to create databases, edit and search data using a
  familiar spreadsheet-like interface, without the need to learn complicated
  SQL commands.

SEE ALSO
  SQLitebrowser's home page: http://sqlitebrowser.sourceforge.net/.

AUTHOR
  sqlitebrowser was developed originally by Mauricio Piacentini from Tab-
  uleiro Producoes. Icons were contributed by Raquel Ravanini, also from
  Tabuleiro. Jens Miltner contributed the code to support SQLite 3.x data-
  bases for the 1.2 release.
  In the spirit of the original SQLite source files, the authors disclaim
  copyright to this source code. It may be used as the basis for other pro-
  grams, public domain, open source or commercial. Do whatever you want with
  it.

  This manual page was written by Francois Fevotte <francois.fevotte@en-
  sta.org>, for the Debian project (but may be used by others).
  Permission is granted to copy, distribute and/or modify this document un-
  der the terms of the GNU General Public License, Version 2 or any later
  version published by the Free Software Foundation.
  On Debian systems, the complete text of the GNU General Public License can
  be found in /usr/share/common-licenses/GPL.

                        August  4, 2007                      SQLITEBROWSER(1)
```

S.69 sqlmap

Essential Database Assessment, Exploitation, Information Gathering, Vulnerability Analysis, Web Applications <https://sqlmap.org/>

Automatic SQL injection tool sqlmap goal is to detect and take advantage of SQL injection vulnerabilities in web applications. Once it detects one or more SQL injections on the target host, the user can choose among a variety of options to perform an extensive back-end database management system fingerprint, retrieve DBMS session user and database, enumerate users, password hashes, privileges, databases, dump entire or user's specific DBMS tables/columns, run his own SQL statement, read specific files on the file system and more.

```
sudo apt install sqlmap
```

Automatic SQL injection tool

```
root@kali:~# sqlmap -h
  __
 --H--
  [.]
--- [.] --- {1.10.9#stable}
|_ -| . [.] | .'| . |
|---|_ [.]|_|_|_|_|_|
|_|V... |_| https://sqlmap.org

Usage: python3 sqlmap [options]

Options:
  -h, --help          Show basic help message and exit
  -hh                 Show advanced help message and exit
  --version            Show program's version number and exit
  -v VERBOSE           Verbosity level: 0-6 (default 1)

Target:
  At least one of these options has to be provided to define the
  target(s)
  -u URL, --url=URL    Target URL (e.g. "http://www.site.com/vuln.php?id=1")
  -g GOOGLEDORK         Process Google dork results as target URLs

Request:
  These options can be used to specify how to connect to the target URL
  --data=DATA           Data string to be sent through POST (e.g. "id=1")
  --cookie=COOKIE       HTTP Cookie header value (e.g. "PHPSESSID=a8d127e..")
  --random-agent        Use randomly selected HTTP User-Agent header value
  --proxy=PROXY         Use a proxy to connect to the target URL
  --tor                 Use Tor anonymity network
  --check-tor           Check to see if Tor is used properly
```

```

root@kali:~# ferret-sidejack -h
-- FERRET 3.0.1 - 2007-2012 (c) Errata Security
-- build = Sep 29 2024 02:38:07 (64-bits)
-- libpcap version 1.10.6 (64-bit time_t, with TPACKET_V3)
options:
-i <adapter>   Sniffs the wire(less) attached to that network adapter.
                Must have libpcap or winpcap installed to work.
-r <files>     Read files in off-line mode. Can use wildcards, such as
                using "ferret -r *.pcap". Doesn't need libpcap to work.
-c <file>     Reads in more advanced parameters from a file.

```

F.10 ffuf

High General Purpose

<https://github.com/ffuf/ffuf>

Fast web fuzzer written in Go (program) ffuf is a fast web fuzzer written in Go that allows typical directory discovery, virtual host discovery (without DNS records) and GET and POST parameter fuzzing. This program is useful for pentesters, ethical hackers and forensics experts. It also can be used for security tests.

```
sudo apt install ffuf
```

Fast web fuzzer written in Go

```

root@kali:~# ffuf -h
Fuzz Faster U Fool - 2.1.0-dev

HTTP OPTIONS:
-H           Header "Name: Value", separated by colon. Multiple -H flags are accepted.
-X           HTTP method to use
-b           Cookie data "NAME1=VALUE1; NAME2=VALUE2" for copy as curl functionality.
-cc          Client cert for authentication. Client key needs to be defined as well for this to work
-ck          Client key for authentication. Client certificate needs to be defined as well for this to work
-d           POST data
-http2       Use HTTP2 protocol (default: false)
-ignore-body Do not fetch the response content. (default: false)
-r           Follow redirects (default: false)
-raw         Do not encode URI (default: false)
-recursion   Scan recursively. Only FUZZ keyword is supported, and URL (-u) has to end in it. (default: false)
-recursion-depth Maximum recursion depth. (default: 0)
-recursion-strategy Recursion strategy: "default" for a redirect based, and "greedy" to recurse on all matches (default: default)
-replay-proxy Replay matched requests using this proxy.
-sni         Target TLS SNI, does not support FUZZ keyword
-timeout     HTTP request timeout in seconds. (default: 10)
-u           Target URL
-x           Proxy URL (SOCKS5 or HTTP). For example: http://127.0.0.1:8080 or socks5://127.0.0.1:8080

GENERAL OPTIONS:
-V           Show version information. (default: false)
-ac          Automatically calibrate filtering options (default: false)
-acc         Custom auto-calibration string. Can be used multiple times. Implies -ac
-ach         Per host autocalibration (default: false)
-ack         Autocalibration keyword (default: FUZZ)
-acs         Custom auto-calibration strategies. Can be used multiple times. Implies -ac
-c           Colorize output. (default: false)
-config      Load configuration from a file
-json        JSON output, printing newline-delimited JSON records (default: false)
-maxtime     Maximum running time in seconds for entire process. (default: 0)
-maxtime-job Maximum running time in seconds per job. (default: 0)
-noninteractive Disable the interactive console functionality (default: false)
-p           Seconds of 'delay' between requests, or a range of random delay. For example "0.1" or "0.1-2.0"
-rate        Rate of requests per second (default: 0)
-s           Do not print additional information (silent mode) (default: false)
-sa          Stop on all error cases. Implies -sf and -se. (default: false)
-scraperfile Custom scraper file path
-scrappers   Active scraper groups (default: all)
-se          Stop on spurious errors (default: false)
-search      Search for a FFUFHASH payload from ffuf history
-sf          Stop when > 95% of responses return 403 Forbidden (default: false)
-t           Number of concurrent threads. (default: 40)
-v           Verbose output, printing full URL and redirect location (if any) with the results. (default: false)

MATCHER OPTIONS:
-mc          Match HTTP status codes, or "all" for everything. (default: 200-299,301,302,307,401,403,405,500)
-ml          Match amount of lines in response
-mm          Matcher set operator. Either of: and, or (default: or)
-mr          Match regexp
-ms          Match HTTP response size
-mt          Match how many milliseconds to the first response byte, either greater or less than. EG: >100 or <100
-mw          Match amount of words in response

FILTER OPTIONS:
-fc          Filter HTTP status codes from response. Comma separated list of codes and ranges
-fl          Filter by amount of lines in response. Comma separated list of line counts and ranges
# ... (44 more lines truncated -- run locally for full output)

```

F.11 fierce

High Information Gathering

<https://github.com/mschwager/fierce>

F.8 feroxbuster

High Information Gathering, Vulnerability Analysis

<https://github.com/epi052/feroxbuster>

Fast, simple, recursive content discovery tool written in Rust feroxbuster is a tool designed to perform Forced Browsing. Forced browsing is an attack where the aim is to enumerate and access resources that are not referenced by the web application, but are still accessible by an attacker. feroxbuster uses brute force combined with a wordlist to search for unlinked content in target directories. These resources may store sensitive information about web applications and operational systems, such as source code, credentials, internal network addressing, etc... This attack is also known as Predictable Resource Location, File Enumeration, Directory Enumeration, and Resource Enumeration.

```
sudo apt install feroxbuster
```

Manual page for feroxbuster 2.13.1

```
root@kali:~# feroxbuster --help
A fast, simple, recursive content discovery tool.

Usage: feroxbuster [OPTIONS]

Options:
  -h, --help            Print help (see a summary with '-h')

  -V, --version          Print version

Target selection:
  -u, --url <URL>       The target URL (required, unless [--stdin || --resume-from ||
                        --request-file] used)

  --stdin                Read url(s) from STDIN

  --resume-from <STATE_FILE>
                        State file from which to resume a partially complete scan (ex.
                        --resume-from ferox-1606586780.state)

  --request-file <REQUEST_FILE>
                        Raw HTTP request file to use as a template for all requests

Composite settings:
  --burp                 Set --proxy to http://127.0.0.1:8080 and set --insecure to true

  --burp-replay          Set --replay-proxy to http://127.0.0.1:8080 and set --insecure to true

  --data-urlencoded <DATA>
                        Set -H 'Content-Type: application/x-www-form-urlencoded', --data to
                        <data-urlencoded> (supports @file) and -m to POST

  --data-json <DATA>    Set -H 'Content-Type: application/json', --data to <data-json>
                        (supports @file) and -m to POST

  --smart                Set --auto-tune, --collect-words, and --collect-backups to true

  --thorough             Use the same settings as --smart and set --collect-extensions and
                        --scan-dir-listings to true

Proxy settings:
  -p, --proxy <PROXY>   Proxy to use for requests (ex: http(s)://host:port,
                        socks5(h)://host:port)

  -P, --replay-proxy <REPLAY_PROXY>
                        Send only unfiltered requests through a Replay Proxy, instead of all
                        requests

  -R, --replay-codes <REPLAY_CODE>...
                        Status Codes to send through a Replay Proxy when found (default:
  # ... (247 more lines truncated -- run locally for full output)
```

F.9 ferret-sidejack

Medium Sniffing & Spoofing, Web Applications

<https://github.com/robertdavidgraham/ferret>

Monitors data and extracts interesting data This tool extracts interesting bits from network traffic. One use is to feed the "hamster" tool. Another use is to dump the output into a text file, then use indexers and grep programs to analyze it.

```
sudo apt install ferret-sidejack
```

This is a free sample.

The full reference covers 820 tools across 931 pages -
every tool with install commands and real usage output.

Get it at icywall.tech/store

ICYWALL