

- NAME
 - SYNOPSIS
 - DESCRIPTION
 - OPTIONS
 - ATTACKS
 - SEE ALSO
 - AUTHORS
 - BUGS
 - BUG REPORTS
-

NAME

APSEND - send arbitrary network packets to hosts

SYNOPSIS

apsend -x / -destination <dest ip> [options] / <attacks>

DESCRIPTION

APSEND is a packet sender with which you can build and send arbitrary network packets. It supports the following protocols: **TCP/UDP/IP/ICMP** and **ethernet frames**, but you can (theoretically) build every possible packet (**based on IP**) using the **--generic** option. **APSEND** is based on the excellent **Net::RawIP** module from Sergey Kolychev <ksv@al.lg.ua>.

OPTIONS

APSEND has got alot of options to build network packets. It may be invoked with the following command-line options:

-d <destination IP>

--dest=IP

--destination=IP This sets the destination IP address to <destination IP>. You can use numeric IP addresses like 192.168.1.1 or hostnames like www.foobar.de. You need this option almost every time you use apsend. You won't need it if you just want to see the help (using the **-?**, **-h** or **--help** option), the ICMP types/code (using the **--show** option) or the available network devices (using the **-x** option).

-s <source IP>

--source=IP This option specifies the source IP address (numerical or host name). If use use **0** as your source IP, then APSEND will use a random IP address for each packet it sends (default source IP="127.0.0.1").

-v <IP version>

--version=IP version This option specifies the IP version (default=4).

-o <type of service>

--tos=type of service Use this option to specify the type of service (ToS) in the IP header (default=0).

--tot=total length This option specifies the total length of the IP header (IP header+data=total length). The total length will be calculated, if you don't specify it.

--id=identification IP identification field (default=0).

-fo <fragmentation offset>

--frag-off=fragmentation offset

--frag This options specifies the fragmentation offset as described in RFC791 (internet protocol specifications). The default value for the fragmentation offset is 0x4000.

-ttl <time to live> Use this option to specify the time to live value in the IP header (default=64).

--protocol This option specifies the protocol number for the protocols based on IP. These are for example ICMP(1), TCP (6) and UDP(17). Have a look at the /etc/protocols for other protocol numbers. The default protocol number is TCP(6).

--ipcheck You can use this option to set the IP checksum. You can also specify wrong IP checksums to test firewalls or other network applications for example. **APSEND** will calculate the right IP checksum as default, if you do not specify your own one.

--ihl This option sets the internet header length (IHL) (default=5). The IHL is needed, because of the variable lenght of the option field in the IP header.

-b <source port>

--source-port=source port

--sp=source port This option sets the source port for the TCP/UDP protocol. The default value for TCP and UDP is to set a random port. Have a look at the /etc/services for other ports/services.

-p <destination port>

--dest-port=destination port

--dp This option sets the destination port for the TCP/UDP protocol. The default value for TCP is port 80 (http) and for UDP it is port 7 (echo). Have a look at the /etc/services for other ports/services.

-rs This option sets a random value for the TCP/UDP source port (set the \$field constant in **APSEND** to specify the random port range).

-rd Like the **-rs** option, except that this option specifies the destination port.

-n <sequence number>

--seq-num=sequence number

--seq=sequence number This option specifies the TCP/ICMP sequence number (default=0).

-a <ack number>

--ack-num This option specifies the acknowledgement number (default=0).

-u <urgent pointer>

--urg-ptr=urgent pointer This option specifies the urgent pointer in the TCP header (default=0).

-do <data offset>

--data-off=data offset

--doff=data offset This option specifies the TCP data offset (default=5).

-res1 This option sets the reserved bit 1 in the TCP header.

-res2 This option sets the reserved bit 2 in the TCP header.

-w <window size>

--window=window size This option specifies the TCP window size (default=0xffff).

-tcheck This option sets the TCP checksum. It will be calculated by default, if you don't set your own value.

Here are the different flags in the TCP header:

-ACK Set the ACK flag in the TCP header.

-RST Set the RST flag in the TCP header.

-URG Set the URG flag in the TCP header.

-PSH Set the PSH flag in the TCP header.

-SYN Set the SYN flag in the TCP header.

-FIN Set the FIN flag in the TCP header.

-len <header/data length> This option sets the header+data length of the UDP header (default=will be calculated).

-ucheck <UDP checksum> This option sets the UDP checksum (default=0, no calculation).

-g <ICMP gateway>

--gateway=ICMP gateway This option sets the internet control message protocol (ICMP) gateway (default=0, no gateway).

-iid <ICMP id>

--icmp-id=ICMP id Specify the ICMP identification (ICMP id).

-icheck Set the ICMP checksum (default=will be calculated).

-mtu <ICMP mtu> This options specifies the ICMP maximal transfer unit (ICMP mtu).

-i <ICMP type>

--icmp-type=ICMP type This option specifies the ICMP type. You can show all possible ICMP types using the **-show** option (see also **--icmp-code**).

-c <ICMP code>

--icmp-code This options specifies the ICMP code. You can show all possible ICMP codes using the **-show** option (see also **--icmp-type**).

-show Show all ICMP codes/types and exit.

--eth-device <device> Specify ethernet device.

-ems <MAC source address>

--eth-mac-source=MAC source address Set the ethernet MAC source address (default=00:00:00:00:00:00).

-emd <MAC destination address>

--eth-mac-source=MAC destination address Set the ethernet MAC destination address (default=00:00:00:00:00:00).

-rems Use a random source MAC address.

-remd Use a random destination MAC address.

--generic <header/data> This is a very powerfull option, as you can use this option, to build every possible packet based on IP. The only problem is, that you have to calculate all of the checksums/whatever yourself. I'll write more (examples and other things) to this option, in the next version of **APSEND**.

--li <port>

--listen=port If you use this option, then **APSEND** will listen on the port <port> and wait for incoming connections. If someone connects to this port (via TCP or UDP), then **APSEND** will print all data it gets to **STDOUT**.

--co <port>

--connect=port Use this option, if you want to connect to a remote host (via TCP or UDP). If you use this option, then **APSEND** will make a REAL connection (three way handshake) to the remote host. If you want to send data to the remote host, just type it in (**STDIN**).

--regex=Regular expression If **APSEND** runs in listening mode (using the **--li <port>** option, then you can use this regex to print only the data the regex matches.

--nregex=Regular expression Like **--regex**, but print only the data NOT being matched by this regex.

-sop <protocol number>

--socket-protocol=protocol number Use this protocol for the **--listen** or **--connect** mode of **APSEND** (currently are only TCP and UDP supported).

-f <script file> (not implemented)

--file=script file (not implemented) This option specifies the scripting file of **APSEND**. But it's not yet implemented, but it'll be available in one of the next versions of **APSEND**. Sorry.

--plugin <file> Use plugin file <file>. Look at the plugin.txt for a few examples!

-m <number of packets to send>

--number=number of packets to send This option specifies the number of packets to send. Use **0** if you want to send unlimited numbers of packets (default=1).

-q

--quiet Don't display anything while sending the packets (quiet mode).

-y <data>

--data=data This option specifies the data to send with the TCP/UDP packets (default='').

-x Show network devices and exit.

ATTACKS

APSEND includes a few DoS attacks like the land attack and others. Please use this attacks only against your OWN machines. They're only for educational purposes, so if you do anything illegal with this attacks, then it's not my fault.

-sf

--syn-flood

--flood This is a SYN flood attack (I think everyone knows how it works).

--land Land attack against WIN95/98 and possibly NT (I didn't test this). Please notify: The source port has to be the same as the destination port. If not, then **APSEND** will autoselect it.

--stream Stream attack (default src/dest port=22).

--bd

--bed

--be-dos DoS attack against BeOS.

-td

--tcp-dos

--tcpd DoS attack against **tcpdump 3.4a** (we all love this program :)). We just need to send an IP packet with protocol=4 and IHL=0 to crash tcpdump 3.4a (from **BLADI** **bladi@EUSKALNET.NET**).

-uf

--udp-flood Just a simple UDP flood. This can crash an ascend router for example (use destination port 7 (**echo**) to do this).

-pf

--ping-flood A simple ping flood :)

SEE ALSO

RFC791, **RFC792**, **RFC768**, **RFC793**, **perl(1)**, **Net::RawIP(1)**, the **perl cookbook**, **TCP/IP illustrated** (Volume 3) [Stevens, 1996], **UNIX Network programming** (Volume 1: Sockets and XTI) [Stevens] and **TCP/IP - Internet Protokolle im professionellen Einsatz** [Mathias Hein].

AUTHORS

Anarchy <anarchy@elxsi.de>

BUGS

I think there are still alot of bugs in **APSEND**.

BUG REPORTS

If you find any bugs in **APSEND** please report it to B<anarchy@elxsi.de>> and if you want me to implement any other options/whatever in **APSEND** then you can also drop me a line. Thanks!