Netfilter Log Format

HOME

Network:

- NetCalc
- NetBitCalc
- Whois

Linux:

- Firewall
- REJECT or DENY?
- ipchains log format
- Netfilter log format
  - Issues
- NetfilterLogAnalyzer

Links:

- Linux HOWTOs
- c.o.l.security FAQ
- FAQs.ORG
- RFC Index (full)

RFC

Here is a quick reference for the format used by the netfilter log messages. This is all derived from the source of the netfilter kernel modules (Linux kernel 2.4.2).

Below is a hypothetical log message generated by netfilter. It is based on a real log entry but I have added all possible IP and TCP flags as well as a fragment offset for illustrative purposes.

Note: If you want to cut-n-paste this into the Netfilter Log Analyzer, then you will have to edit out the fragment offset or set it to zero.

```
Apr 16 00:30:45 megahard kernel: NF: D(I, Priv) IN=eth1 OUT=
MAC=00:80:8c:1e:12:60:00:10:76:00:2f:c2:08:00
SRC=211.251.142.65 DST=203.164.4.223 LEN=60
TOS=0x00 PREC=0x00 TTL=44 ID=31526 CE DF MF FRAG=179
OPT (072728CBA404DFCBA40253CBA4032ECBA403A2CBA4033ECBA40
2C1180746EA18074C52892734A200)
PROTO=TCP SPT=4515 DPT=111 SEQ=1168094040 ACK=0
WINDOW=32120 RES=0x03 URG ACK PSH RST SYN FIN URGP=0
OPT (020405B40402080A05E3F3C40000000001030300)
```

The items are explained in sequence:

`Apr 16 00:30:45 megahard kernel:`	syslog prefix. It is not present if you read log messages from the console.
`NF: D(I, Priv)`	Enabled with: **`--log-prefix`** *`'prefix'`* An arbitrary, user defined log prefix. **Including the spaces.** A trailing space is necessary to keep the prefix separate from the next token; this is a bug in netfilter.
`IN=eth1`	Interface the packet was received from. Empty value for locally generated packets.
`OUT=`	Interface the packet was sent to. Empty value for locally received packets.

`MAC= 00:80:8c:1e:12:60: 00:10:76:00:2f:c2: 08:00`	Destination MAC=00:80:8c:1e:12:60, Source MAC=00:10:76:00:2f:c2, Type=08:00 (ethernet frame carried an IPv4 datagram)
`SRC=211.251.142.65`	Source IP address
`DST=203.164.4.223`	Destination IP address
`LEN=60`	Total length of IP packet in bytes
`TOS=0x00`	Type Of Service, "Type" field. Increasingly being replaced by **DS** and **ECN**. Refer to the IP header info below.
`PREC=0x00`	Type Of Service, "Precedence" field. Increasingly being replaced by **DS** and **ECN**. Refer to the IP header info below.
`TTL=44`	remaining Time To Live is 44 hops.
`ID=31526`	Unique ID for this IP datagram, shared by all fragments if fragmented.
`CE`	Presumably the "ECN CE" flag (Congestion Experienced). This seems to be wrong because according to RFC2481, the CE bit is located in the TOS field. Refer to the IP header info below.
`DF`	"Don't Fragment" flag.
`MF`	"More Fragments following" flag.
`FRAG=179`	Fragment offset in units of "8-bytes". In this case the byte offset for data in this packet is 179*8=1432 bytes.
`OPT (0727..A200)`	Enabled with: **`--log-ip-options`** IP options. This variable length field is rarely used. Certain IP options, f.e. source routing, are often disallowed by netadmins. Even harmless options like "Record Route" may only be allowed if the transport protocol is ICMP, or not at all.
`PROTO=TCP`	Protocol name or number. Netfilter uses names for TCP, UDP, ICMP, AH and ESP. Other protocols are identified by number. A list is in your */etc/protocols*. A complete list is in the file ***protocol-numbers***
`SPT=4515`	Source port (TCP and UDP). A list of port numbers is in your */etc/services*. A complete list is in the file ***port-numbers***
`DPT=111`	Destination port (TCP and UDP). See SPT above.

`SEQ=1168094040`	Enabled with: **`--log-tcp-sequence`** Receive Sequence number. By cleverly chosing this number, a cryptographic "cookie" can be implemented while still satisfying TCP protocol requirements. These "SYN-cookies" defeat some types of SYN-flooding DoS attacks and should be enabled on all systems running public TCP servers. `echo 1 > /proc/sys/net/ipv4/tcp_syncookies`
`ACK=0`	Same as the Receive Sequence number, but for the other end of the TCP connection.
`WINDOW=32120`	The TCP Receive Window size. This may be scaled by bit-shifting left by a number of bits specified in the "Window Scale" TCP option. If the host supports ECN, then the TCP Receive Window size will also be controlled by that.
`RES=0x03`	Reserved bits. The ECN flags "**CWR**" and "**ECNE**" will show up in the two least significant bits of this field. Refer to the TCP header info below.
`URG`	Urgent flag. See URGP below.
`ACK`	Acknowledgement flag.
`PSH`	Push flag.
`RST`	RST (Reset) flag.
`SYN`	SYN flag, only exchanged at TCP connection establishment.
`FIN`	FIN flag, only exchanged at TCP disconnection.
`URGP=0`	The Urgent Pointer allows for urgent, "out of band" data transfer. Unfortunately not all protocol implementations agree, so this facility is hardly ever used.
`OPT (020405...300)`	enabled with: **`--log-tcp-options`** TCP options. This variable length field gets a lot of use. Important options include: Window Scaling, Selective Acknowledgement and Explicit Congestion Notification. Refer to the TCP header info below.
	Unfortunately the rule number in the chain which matched the packet is for architectural reasons not available in netfilter logs. You will have to "cook your own" by using the user-prefix feature.

More interesting files, such as ***multicast-addresses***, can be found in http://www.isi.edu/in-notes/iana/assignments/.

Issues with netfilter log format

Also suggests an alternative log format.

The ULOG module looks like a suitable future remedy.

Protocol Header Information

IP Header Format as defined in RFC-791:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
IP Version				Hdr.Length				**TOS / DS,ECN**								Total Length															
Identification																-	DF	MF	Fragment Offset												
Time To Live								Protocol Number								Header Checksum															
32 bit Source Address																															
32 bit Destination Address																															
Options (0 to 10 Words of 32 Bits)																															
IP Payload (including header of heigher protocol)																															

The header of an IP packet consists of 5 or more words of 32 bits (4 bytes) each. The minimum header length (no options) is therefore 20 bytes. The Version field for the shown type of packet is 4 = IPv4 (Internet Protocol version 4). The header Length field is the header length in 32bit words, this would be 5 without options, and at most 15 with options. The Total Length is in bytes and includes the header. Data length can then be calculated from the supplied values.

TOS / DS / ECN: This field has had an unstable history. This is briefly explained in RFC2481, section 19 (near the end).

	0	1	2	3	4	5	6	7
TOS	Precedence			Type				-
DS,ECN	DS Codepoint						ECT	CE

Many sites are starting to implement Differentiated Services DS [RFC2474] in their routers. DS uses *code-points* which are stored in bits 0 to 5 of the old TOS field. The content and meaning of this field can change at network boundaries.

If the host is ECN [RFC2481] capable and the payload is a TCP packet, then up to two flag bits will be needed in the old TOS field. Bit 6 becomes the **ECT** (ECN-capable Transport) flag, and Bit 7 becomes the **CE** (Congestion Experienced) flag.

IP datagrams can be fragmented if the link layer cannot fit it into a single link layer data unit. The fragment offset is specified in units of *8-bytes*, thus allowing the available 13 bits to cover the necessary values for up to 64K of data.

IP packets usually carry a higher level protocol such as TCP. In the case of TCP, the PROTO field would be set to 6 and the TCP *Protocol Data Unit* (PDU) is carried in the IP Payload field of the packet. See below.

TCP Header Format (as defined in RFC-793):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Source Port																Destination Port															
Sequence Number																															
Acknowledgement Number																															
Data Offset				-	-	-	-	CWR	ECNE	URG	ACK	PSH	RST	SYN	FIN	Window															
Checksum																Urgent Pointer															
Options (0 to 10 Words of 32 Bits)																															
TCP Payload																															

The header of a TCP packet consists of 5 or more words of 32 bits (4 bytes) each. The minimum header length (no options) is therefore 20 bytes. The *Data Offset* field is the header length in 32bit words, this would be 5 without options, and at most 15 with options.

Explicit Congestion Notification (ECN) [RFC2481] adds 2 new flags to the TCP header: *Congestion Window Reduced* (CWR) and *ECN-Echo* (ECNE). ECN also requires 1 or 2 additional flags in the IP header.

Commonly, the TCP header will carry options related to enhancements of the TCP protocol. Important options are Window Scaling, Selective Acknowledgement (SACK) [RFC2018, RFC2883] and Explicit Congestion Notification (ECN) [RFC2481].

TCP data payload length is the IP payload length minus the TCP header length.

TCP packets usually carry an application level data stream, f.e. HTTP, FTP, Telnet, SSH, etc.

UDP Header format (as defined in RFC-768):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Source Port																Destination Port															
Total Length																Checksum (optional)															
UDP Payload																															

The header of a UDP packet consists of 2 words of 32 bits (4 bytes) each. The header length is therefore always 8 bytes. The *Total Length* field includes the UDP header and is measured in bytes.

UDP packets usually carry an application level datagram as their payload, f.e. DNS, NTP, NFS, etc.

All good books on TCP/IP explain the IP, TCP, UDP header formats in detail. There are also various RFCs covering different aspects of IP, ICMP, TCP, UDP and other protocols. Another good starting point is Uri's TCP/IP Resources List.