

CRITICAL PROTOCOL FIELDS

(by Mark E. Donaldson)

Field	Field Length	Field Location	Byte Value	Notes
IP				
IP Header Version	4 bits	byte 0 (high order)	x1	31. IDS insertion attack.
IP Header Length	4 bits	byte 0 (low order)	x4	1. Min value 5 (20 bytes). Max value 15 (60 bytes).
IP TOS	1 bytes	byte 1	bit values	1. Called Differentiated Services byte in future. 2. PreDTRCx 3. Two low order for ECN.
IP Total Datagram Length	2 bytes	bytes 2 & 3	x1	1. Min value 46 bytes on Ethernet. Max. value 65,535. If Length < 46 bytes must be padded with 0's.
IP Header ID Number	2 bytes	bytes 4 & 5	x1	1. Fragmentation ID on fragmented packets. 2. Used to reassemble fragments by dest host. 3. Increments by one for each new connection & each retry. 4. Use to find spoofed packets.
IP Fragmentation Flags	3 bits	byte 6 (3 high order)	bit values	1. xDM 2. IDS insertion attack & OS Fingerprinting. 3. MTU Path Discovery with ICMP Type 3-4.
IP Fragment Offset Length	13 bits	bytes 6 & 7	x 8	1. Used to over-write data. 2. Mapping with incomplete fragments and ICMP Type 11-1.
IP TTL	1 byte	byte 8	x1	1. IDS insertion attack if set low & OS Fingerprinting. 2. Trace source to determine spoofing.
IP Embedded Protocol	1 bytes	byte 9	x1	1. Mapping with protocol scan.
IP Header Checksum	2 bytes	bytes 10 & 11	algorithm	1. IDS insertion attack if corrupted.
IP Source Address	4 bytes	bytes 12 to 15	octets	1. Spoofing (Apply Ingress & Egress Filtering.)
IP Destination Address	4 bytes	bytes 16 to 19	octets	
IP Options (if applicable)	0 to 40 bytes	bytes 20 to 59	x1	1. OS Fingerprinting.
Option Type (Options Only) <ul style="list-style-type: none"> ▪ End of Options (Type 0) ▪ No Op (Type 1) ▪ Record Route (Type 7) ▪ Timestamp (Type 68) ▪ Loose Source (Type 131) ▪ Strict Source (Type 137) 	1 byte	byte 20	bit values	1. Consists of 3 sub-fields of copy (1 bit), class (2 bits), and type (5 bits).
Option Length (Options Only)	1 byte	byte 21	x1	1. Includes all option fields and data. Used to determine where option data ends, and where the next option type field may begin.
Option Data (Options Only)	variable	byte 22	x1	
TCP				
Source Port	2 bytes	bytes 0 & 1	x1	1. Initial source port > 1023. 2. Increments by 1 for each new (non-retry) connection.
Destination Port	2 bytes	bytes 2 & 3	x1	1. Port 0 invalid but used for scans.
Sequence Number	4 bytes	bytes 4 to 7	x1	1. ISN randomly generated at Syn. 2. Changes for each new (non-retry) connection. 3. OS Fingerprinting (ISN prediction).

Field	Field Length	Field Location	Byte Value	Notes
Acknowledgement Number	4 bytes	bytes 8 to 11	x1	<ol style="list-style-type: none"> 1. Last received sequence number + 1. 2. Must be next expected sequence number. 3. Valid values must > 0 unless wrapped. 4. 0 an impossible initial A as S takes one sequence number. 5. Ack storm from hijacked session.
TCP Header Length	4 bits	byte 12 (high order)	x4	<ol style="list-style-type: none"> 1. Min. value 5 (20 bytes). Max value 15 (60 bytes).
Reserved	6 bits	byte 12 (4 low order) byte 13 (2 high order)	bit values	<ol style="list-style-type: none"> 1. Two high order reserved bits in byte 13 for future ECN (CWR & ECN-echo).
TCP Flags	6 bits	byte 13 (low order)	bit values	<ol style="list-style-type: none"> 1. UAPRSF 2. RESET/ACK returned by non-listening port. 3. RESET returned by unsolicited ACK. 4. PUSH flag only set when sending host empties sending buffer, or all data sent. Tells receiving host to immediately PUSH data up stack to application & not wait for more data. PUSH is set by sending side application telling TCP to create a segment, set the PUSH flag, and send whatever data is in receive buffer now. Telnet needs to do this to echo a keystroke immediately back to the client. 5. OS Fingerprinting & Port Scanning.
Window Size	2 bytes	bytes 14 & 15	x1	<ol style="list-style-type: none"> 1. Size of receive buffer (dynamic). 2. Allows receiver to control flow. 3. Window size of 0 halts data transmission. 4. Initial window size used for OS Fingerprinting. 5. Max receive buffer limited by field size to 65,000 bytes.
TCP Checksum	2 bytes	bytes 16 & 17	algorithm	<ol style="list-style-type: none"> 1. IDS insertion attack if corrupted.
Urgent Pointer	2 bytes	bytes 18 & 19	algorithm	<ol style="list-style-type: none"> 1. Sequence number of last byte of any urgent data in current segment.
TCP Options (if applicable)	0 to 40 bytes	bytes 20 to 59	x1	<ol style="list-style-type: none"> 1. OS Fingerprinting (option order) & IDS insertion attacks. 2. Padding used to make segments end on 32 bit boundary. 3. NOOP used as internal padding for 32 bit boundary.
Option Type (Options Only) <ul style="list-style-type: none"> ▪ End of Options (Type 0) ▪ No Op (Type 1) ▪ MSS (Type 2) ▪ Window Scale (Type 3) ▪ Selective ACK (Type 4) ▪ Sel ACK Data (Type 5) ▪ Timestamp (Type 8) 	1 byte	byte 20	x1	<ol style="list-style-type: none"> 1. All TCP option except Timestamp are negotiated on SYN.
Option Length (Options Only)	1 byte	byte 21	x1	<ol style="list-style-type: none"> 1. Includes all option fields and data. Used to determine where option data ends or where next option field begins.
Option Data (Options Only)	variable	byte 22	x1	
UDP				
Source Port	2 bytes	bytes 0 & 1	x1	<ol style="list-style-type: none"> 1. Initial source port > 1023. 2. Increments by 1 for each new (non-retry) connection.
Destination Port	2 bytes	bytes 2 & 3	x1	<ol style="list-style-type: none"> 1. Port 0 invalid but used for scans. 2. Scans rely on ICMP response.
UPD Length	2 bytes	bytes 4 & 5	x1	<ol style="list-style-type: none"> 1. Represents UPD header and data. 2. UDP header always 8 bytes. 3. Min value 8 bytes. Max value 65515 bytes. 4. DNS UDP 512 bytes max.

Field	Field Length	Field Location	Byte Value	Notes
UDP Checksum	2 bytes	bytes 6 & 7	algorithm	1. IDS insertion attack if corrupted.
ICMP				
ICMP Message Type	1 byte	byte 0	x1	1. Messages are either error or query. 2. Error messages 8 bytes + IP Header + 8 Original bytes. 3. Mapping & Discovery.
ICMP Message Code	1 byte	byte 1	x1	1. Sub-category of Type.
ICMP Checksum	2 bytes	bytes 2 & 3	algorithm	1. IDS insertion attack if corrupted.
Message Data (Error Only)	4 bytes	bytes 4 to 7	sub-fields	1. Content depends on error message type. 2. Error Message Types 3, 4, 5, 11, & 12. 3. Parameter message contains 1 byte pointer to problem octet in original datagram.
Additional Fields (Query Only)	4 bytes	bytes 4 to 7	sub-fields	1. Content depends on query message type. 2. Query Message Types 0, 8, 13, 14, 17, & 18.
ICMP Identifier (Query Only)	2 bytes	bytes 4 & 5	x1	1. Provides a session identification number for request/reply messages. 2. Usually same as PID (process ID). 3. Covert Channel exploit.
ICMP Sequence Number (Query Only)	2 bytes	bytes 6 & 7	x1	1. Provides a counter for request/reply messages, allowing multiple messages for a single identifier. 2. Increments sequentially starting from 0 for each message within ICMP Identifier set (e.g. multiple ping packets). 3. Covert channel exploit.
Original IP Header (Error Only)	20 to 60 bytes	byte 8	x1	1. Valid for ICMP "error messages" only.
Original Data (Error Only)	8 bytes	start point variable	x1	1. Valid for ICMP "error messages" only.
ARP/RARP (ETHERNET)				
Hardware Type	2 bytes	bytes 0 & 1	x1	Specifies a hardware interface type for which the sender requires a response.
Protocol Types	2 bytes	bytes 2 & 3	x1	Specifies the type of high-level protocol address the sender has supplied.
HLen	1 byte	byte 4	x1	Hardware address length in bytes.
PLen	1 byte	byte 5	x1	Protocol address length in bytes.
Operation Code (Opcode) <ul style="list-style-type: none"> ▪ 1 ARP request ▪ 2 ARP response ▪ 3 RARP request ▪ 4 RARP response. ▪ 5 Dynamic RARP request ▪ 6 Dynamic RARP reply ▪ 7 Dynamic RARP error ▪ 8 InARP request ▪ 9 InARP reply 	2 bytes	bytes 6 & 7	x1	
Sender Hardware Address	6 bytes	bytes 8 to 13	octets	HLen bytes in length.
Sender Protocol Address	4 bytes	bytes 14 to 17	octets	PLen bytes in length.
Target Hardware Address	6 bytes	bytes 18 to 23	octets	HLen bytes in length.
Target Protocol Address	4 bytes	bytes 24 to 23	octets	PLen bytes in length.
DNS				
DNS Identification Number	2 bytes	bytes 0 & 1		1. Unique number for each query/response pair.

Field	Field Length	Field Location	Byte Value	Notes
DNS Flags <ul style="list-style-type: none"> ▪ QR ▪ Opcode ▪ AA ▪ TC ▪ RD ▪ RA ▪ Z ▪ RCODE 	2 bytes 1 bit 4 bits 1 bit 1 bit 1 bit 1 bit 3 bits 4 bits	bytes 2 & 3 <ul style="list-style-type: none"> ▪ bit 0 ▪ bits 1 to 4 ▪ bit 5 ▪ bit 6 ▪ bit 7 ▪ bit 8 ▪ bits 9 to 11 ▪ bit 12 to 15 	bit values	<ol style="list-style-type: none"> 1. Query (0) or Response (1) 2. Standard (0), Inverse (1), or Server Status (2) 3. Authoritative Answer (1) tcpdump = * 4. Truncation (1) tcpdump = 5. Recursion Desired (1) tcpdump = + 6. Recursion Not Available (0) tcpdump = - 7. Reserved 8. No Error (0), Format Error (FORMERR) (1), Server Failure (2), Non-existent Domain (NXDOMAIN) (3), Query Type Not Implemented (4), Query Refused (5)
QDCOUNT	2 bytes	bytes 4 & 5	x1	1. Number of entries in question.
ANCOUNT	2 bytes	bytes 6 & 7	x1	1. Number of resource records in answer section. Tcpdump 6/8/8 = 6 Answer Records 8 Authoritative Records 8 Additional Records
NSCOUNT	2 bytes	bytes 8 & 9	x1	1. Number of server records in authority section.
ARCOUNT	2 bytes	bytes 10 & 11	x1	1. Number of records in additional info section.
Question Section	variable	starting point byte 12	x1	
Answer Section	variable	variable	x1	
Authority Section	variable	variable	x1	
Additional Info Section	variable	variable	x1	
RPC				
Transaction ID (XID)	4 bytes	bytes 0 to 3	x1	<ol style="list-style-type: none"> 1. RPC header immediately follows UDP header for UDP transport. For TCP transport, RPC header preceded by 4 byte RPC total length field. 2. Found in call & reply packets.
Message Type	4 bytes	bytes 4 to 7	x1	<ol style="list-style-type: none"> 1. Found in call & reply packets. 2. Call = 0 Reply = 1.
RPC Version (Call or Request) Status (Reply)	4 bytes	bytes 8 to 11	x1	1. Status 0 = Accepted Status != 0 = Rejected.
RPC Program (Call or Request) Verifier (Reply)	4 bytes	bytes 12 to 15	x1	<ol style="list-style-type: none"> 1. 0001 86a0 = 100000 = portmap. 2. 0001 86a5 = 100005 = mounted. 3. Verifier up to 408. 4. NULL in verifier field means nothing follows.
Program Version (Call or Request) Accept Status (Reply)	4 bytes	bytes 16 to 19	x1	1. 0000 0002 = version 2
Procedure Number (Call)	4 bytes	bytes 20 to 23	x1	1. 0000 0003 = getport()
Data (Call & Reply)	variable	byte 24 (Call) byte 20 (Reply)	x1	1. Call can include optional Auth Credentials & Auth Verification
ETHERNET				
Source MAC Address	6 bytes	bytes 0 to 5	x1	
Destination MAC Address	6 bytes	bytes 6 to 11	x1	
EtherType	2 bytes	bytes 12 & 13	x1	1. IP = 0x0800, ARP = 0x0806

PROTOCOL HEADER LENGTHS

Protocol	Header Length (bytes)	Comment
Ethernet	14	Also has 4 byte CRC
IP	20 to 60	Depends on options used
TCP	20 to 60	Depends on options used
UDP	8	
ICMP Error	8 + 8 + (20 to 60)	Depends on options used in mother message
ICMP Query	4 + (0 to 65,000)	Depends on query type
ARP/RARP	24	
DNS	20	
RPC	24 20	Call or Request Header Reply Header

DNS Output Meanings:

(31) = 31 payload bytes not counting IP & UDP Headers

+ = Recursion Desired

- = Recursion not available

*** = Authorative Response**

| = Record Truncated

6/8/8 = 6 Answer Records

8 Authorative Records

8 Additional Records

TCP Flag Responses

- SYN on non-listening port => RST,ACK
- Unsolicited SYN/ACK (closed) => RST
- Unsolicited SYN/ACK (open) => No Response
- Unsolicited ACK => RST
- RST on non-connection => No Response

Inverse Mapping

- RST Scan => RST or host unreachable
- Unsolicited DNS reply => host unreachable
- Fragmented IP Datagrams => host unreachable
- Unsolicited ICMP response => host unreachable

Significant High Ports

- Port 9100 => HP LaserJet Printers
- Port 1524 => ingreslock
- Port 9704 => RPC & WU-FTP buffer overflow
- Port 9876 => T)rn rootkit backdoor
- Port 31337 => eleet port
- Port 371 => clearcase
- Port 6699 => Napster

- Port 4000 => Terabase & ICQ
- Port 3128 => Squid Proxy
- Port 31789 => Hack a Tack

Severity/Lethality States/Formula

Indications & Warnings

Event (Intrusion) Categories

- Privilege access
- Limited access
- Reconnaissance (mapping, fingerprinting)
- Stealth reconnaissance (FIN, inverse mapping)
- Denial of Service
- Distributed Denial of Service
- AUP (acceptable use policy) violations

Critical Logs

- System (syslog, event log)
- Web Server (Apache, IIS)
- Firewall
- NIDS
- HIDS

Stages of Incident Response

- Preparation
- Detection
- Containment
- Eradication
- Recovery
- Follow-up