Overview Known Protocols

Offensive Network Security
Florida State University
Spring 2014

Outline

- Obtain an understanding of known protocols
  - Address Resolution Protocol
  - Internet Control Message Protocol
  - Dynamic Host Configuration Protocol
  - Domain Name System
  - Hyper Text Transfer Protocol
  - Simple Mail Transfer Protocol
  - Universal Plug and Play
- This is just an overview of some protocols and not exhaustive

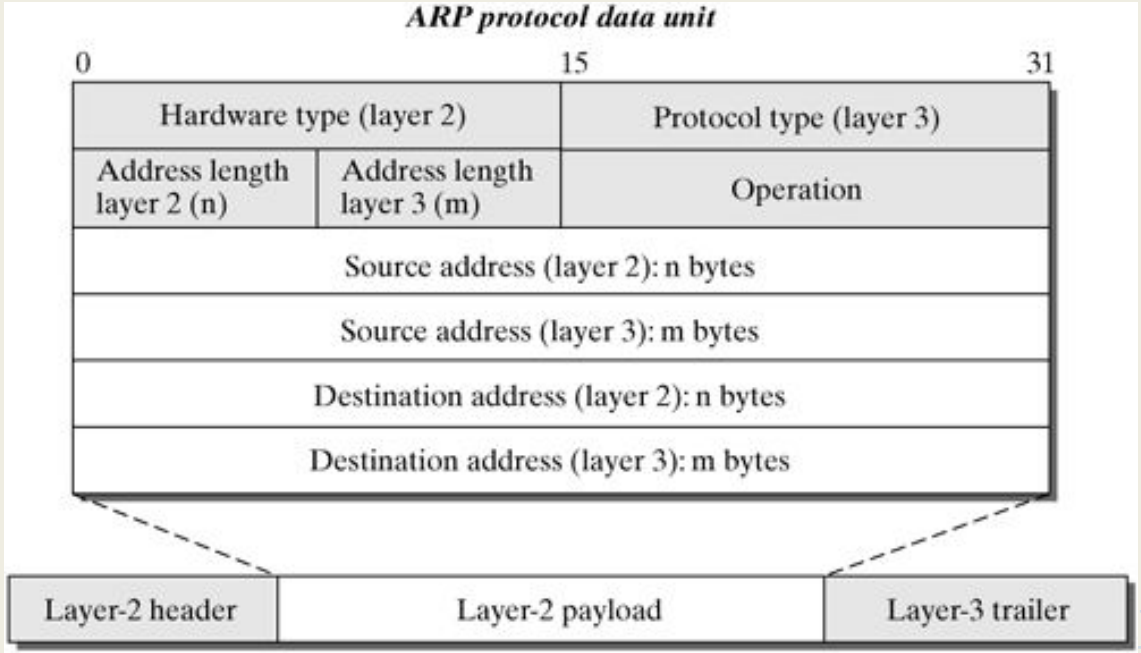
Address Resolution Protocol

- RFC 826 (1982)
- Internet Standard (IS) 37
- Link-Layer protocol due to the nature of working in a single network
- Used to map IPv4 address to physical address
  - IPv4 -> DECNet
  - IPv4 -> ATM
  - IPv4 -> Ethernet MAC (most common)
- Ethernet Type is 0x0806
- Is replaced with Neighbor Discovery Protocol (NDP) in IPv6

ARP Packet

- Hardware Type: Hardware type, i.e. 0x1 is Ethernet
- Protocol Type: Layer 3 protocol type, i.e. 0x800 is IPv4
- Hardware Length: Length in octets of the hardware address
- Protocol Length: Length in octets of physical address
- Operation: Sender operation, i.e. 0x1 (request), 0x2 (reply)
- Sender Hardware Address: layer 2 address of the sender
- Sender Protocol Address: layer 3 address of the sender
- Target Hardware Address: layer 2 address of the target
- Target Protocol Address: layer 3 address of the target

ARP Packet


ARP protocol data unit
0
15
31
Hardware type (layer 2)
Protocol type (layer 3)
Address length layer 2 (n)
Address length layer 3 (m)
Operation
Source address (layer 2): n bytes
Source address (layer 3): m bytes
Destination address (layer 2): n bytes
Destination address (layer 3): m bytes
Layer-2 header
Layer-2 payload
Layer-3 trailer

ARP Thoughts

- Abused mostly to spoof IPv4 to physical address mapping
- What if we specify a different protocol type?
- What if we specify an incorrect address length for protocol type?
- What if these incorrect values were broadcasted?

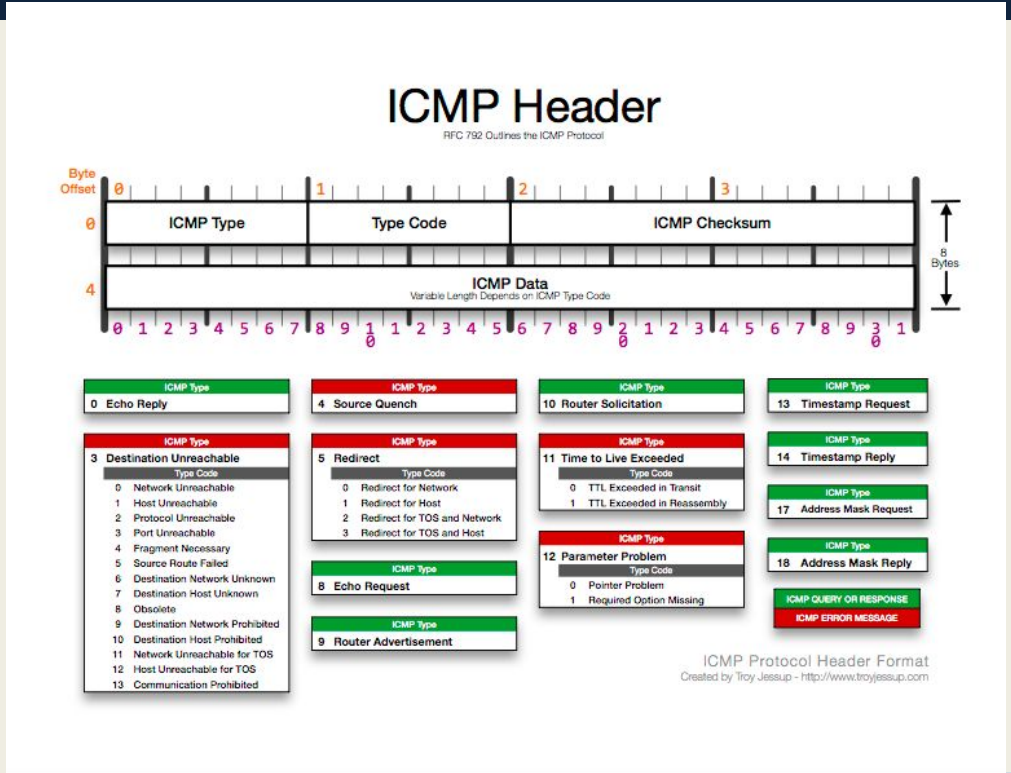
ICMP

- RFC 792 (1981)
- IP protocol number: 0x01
- Integral to IP
- Must be implemented by IP modules according to RFC
- IPv6 implements ICMPv6
- Purpose: provide feedback in case of problems since IP is not reliable

ICMP Header

- Type: Type of ICMP message
  - Echo-reply
  - Destination Unreachable
  - Source-Quench
  - Redirect Message
  - Echo Request
  - Router Advertisement
- Code: Subtype of ICMP message
  - What type of Destination Unreachable
  - What type of Redirect Message
- Checksum: Error checking ICMP header+data
- Varying Fields: More fields in header depends on ICMP message type

ICMP Header


ICMP Header
RFC 792 Outlines the ICMP Protocol
Byte Offset
0
4
ICMP Type
Type Code
ICMP Checksum
8 Bytes
ICMP Data
Variable Length Depends on ICMP Type Code
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
ICMP Type
0 Echo Reply
ICMP Type
3 Destination Unreachable
Type Code
0 Network Unreachable
1 Host Unreachable
2 Protocol Unreachable
3 Port Unreachable
4 Fragment Necessary
5 Source Route Failed
6 Destination Network Unknown
7 Destination Host Unknown
8 Obsolete
9 Destination Network Prohibited
10 Destination Host Prohibited
11 Network Unreachable for TOS
12 Host Unreachable for TOS
13 Communication Prohibited
ICMP Type
4 Source Quench
ICMP Type
5 Redirect
Type Code
0 Redirect for Network
1 Redirect for Host
2 Redirect for TOS and Network
3 Redirect for TOS and Host
ICMP Type
8 Echo Request
ICMP Type
9 Router Advertisement
ICMP Type
10 Router Solicitation
ICMP Type
11 Time to Live Exceeded
Type Code
0 TTL Exceeded in Transit
1 TTL Exceeded in Reassembly
ICMP Type
12 Parameter Problem
Type Code
0 Pointer Problem
1 Required Option Missing
ICMP Type
13 Timestamp Request
ICMP Type
14 Timestamp Reply
ICMP Type
17 Address Mask Request
ICMP Type
18 Address Mask Reply
ICMP QUERY OR RESPONSE
ICMP ERROR MESSAGE
ICMP Protocol Header Format
Created by Troy Jessup - http://www.troyjessup.com

ICMP Thoughts

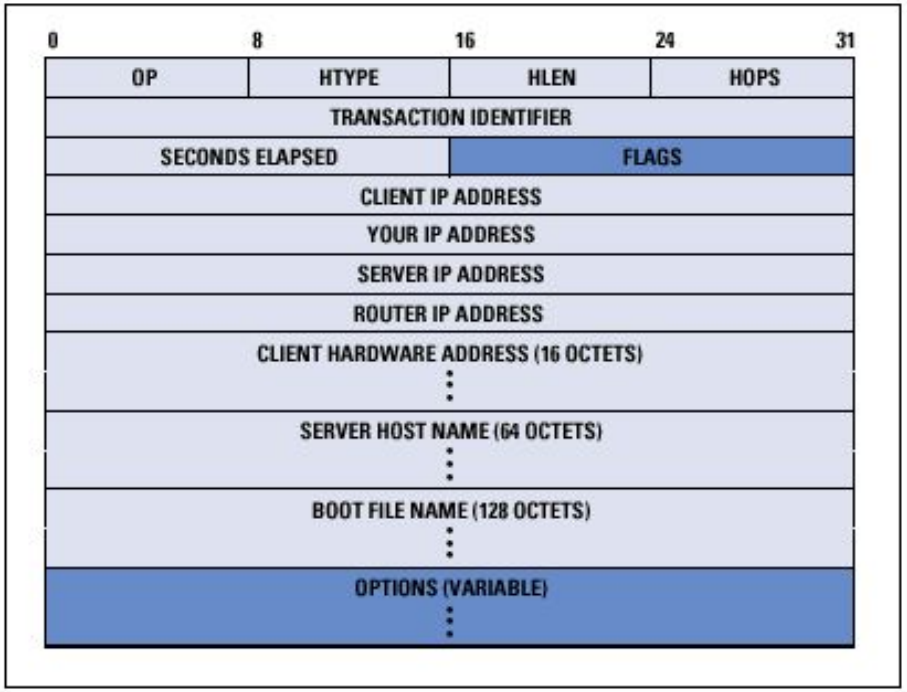
- How can we prevent these messages from being sent by the server?
- Can we send arbitrary data using ICMP?
- How much error checking is needed due to varying header fields?

DHCP

- Originally described in RFC 1531 as an extension to Bootstrap Protocol (BOOTP)
- Currently described in RFC 2131 (http://tools.ietf.org/html/rfc2131) for IPv4
- Currently uses BOOTP relay agents to allow interaction with BOOTP
- Provides ability to pass host configuration through TCP/IP network

DHCP Header

Figure 2: DHCP Message Format



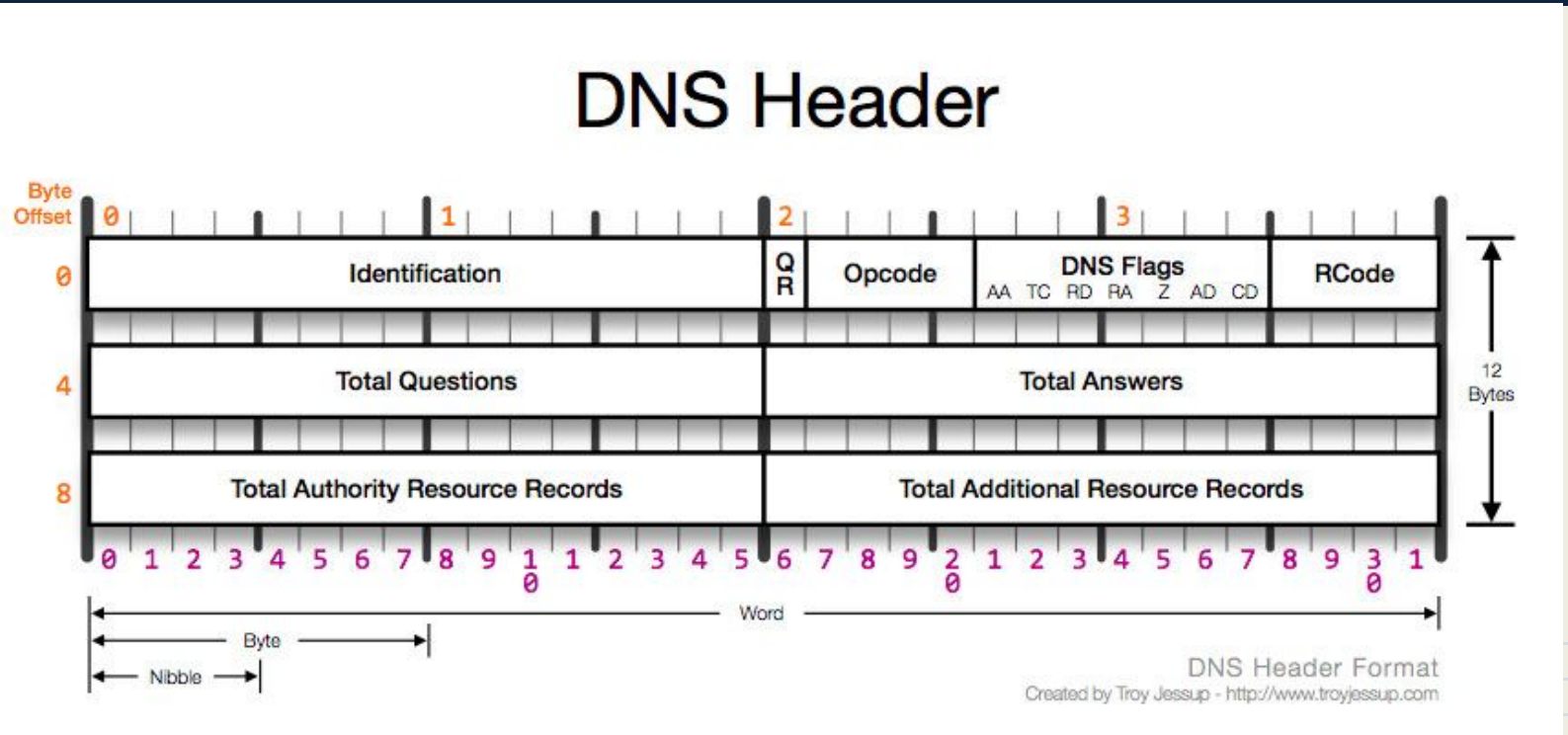
DHCP Thoughts

- Could an attacker assign a malicious gateway?
- Could an attacker revoke an assigned IP?
- What are some of the setbacks for

DNS

- Original RFCs: 882, 883
- Current RFCs: 1034, 1035
- Berkeley Internet Name Domain (BIND)
  - Most common DNS server
  - Implemented for Unix, ported to Windows NT
- Naming conventions/syntax applied to domain names
  - Length?
  - Accepted Characters, ASCII/Unicode?
- DNS servers are distributed and use client/server model

DNS Header


DNS Header
Byte Offset
0
1
2
3
0
Identification
QR
Opcode
DNS Flags
AA TC RD RA Z AD CD
RCode
4
Total Questions
Total Answers
8
Total Authority Resource Records
Total Additional Resource Records
12 Bytes
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
Word
Byte
Nibble
DNS Header Format
Created by Troy Jessup - http://www.troyjessup.com

DNS Thoughts

- DNS is a complex protocol / standard
- over 30 RFCs out for DNS and DNS securtiy
- Are all of these implemented correctly?
- Malformed DNS requests causes…?

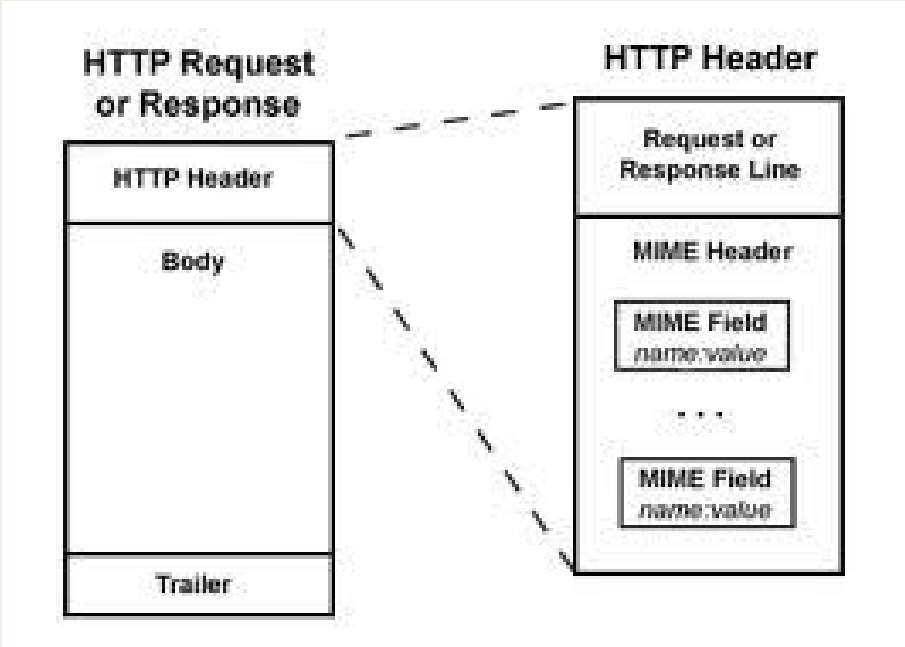
HTTP

- Version 1.1 described in RFC 2616
- Text Based Protocol
- Application Layer Protocol (Layer 5)
  - Designed for IP networks
  - Assumes transport protocol will be used
    - TCP (mainly)
    - UDP (SSDP)
- Stateless protocol
- Request and response headers + data
- Servers file type + data

HTTP Request Methods

- CONNECT
- DELETE
- GET
- HEAD
- OPTIONS
- POST
- PUT
- TRACE

HTTP Header


HTTP Request
or Response
HTTP Header
Body
Trailer
HTTP Header
Request or
Response Line
MIME Header
MIME Field
name:value
. . .
MIME Field
name:value

HTTP Thoughts

- Complex protocol / header
- Plenty of variable length methods, sizes, etc.
- Does a HTTP client/server handle an unknown method correctly?
- Does a HTTP client/server handle connection break correctly?
- What could a malicious HTTP server do to a client?
- What could a malicious HTTP client to do to a server?

Know a Protocol (RFC)

- Request for Comments
  - Published by Internet Engineering Task Force / Internet Society
  - Used to describe internet-connected systems
  - RFC can be elevated to Internet Standard status by IETF
  - Not at protocols are published using RFC (looking at you IEC standards)
- tl;dr: read RFC, understand protocol
- This could give clues to what the server/client is expecting in the packet
  - Programmer could make assumptions, leads to vulnerabilities
  - Make a protocol perform unexpectedly
  - Possible Denial of Service

Official IP Standards

- http://www.rfc-editor.org/rfcxx00.html