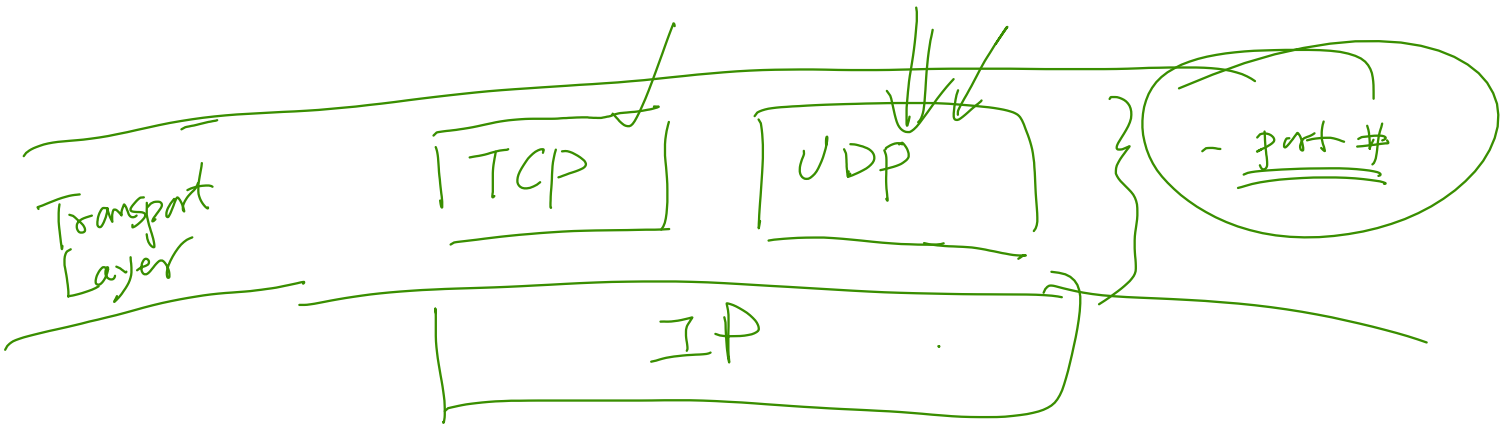
Internet Security

UDP and Attacks

What the IP Layer Does and Does Not Do



Transport Layer and Port Numbers

port ⟺ Application

16-bit. 0 ~ 65535

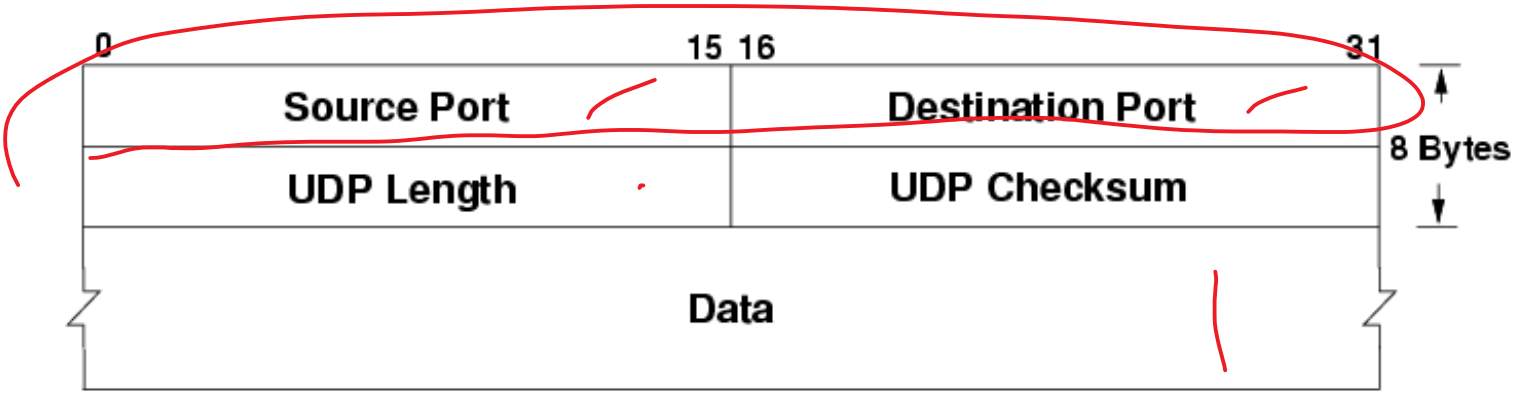
23 : tcp: telnet
udp

0 ~ 1023 : well-known applications
root
53 : DNS

1024 ~ 49151 : less well-known register

SRC port → 49152 ~ 65535 : Dynamic/Private port

UDP Header and Protocol



UDP Client/Server Programs

❖ UDP client

```c
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/ip.h>

void main()
{
    struct sockaddr_in dest_info;
    char *data = "Hello Server.\n";

    // Create a network socket.
    int sock = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);

    // Provide needed information about destination.
    memset((char *) &dest_info, 0, sizeof(dest_info));
    dest_info.sin_family = AF_INET;
    dest_info.sin_addr.s_addr = inet_addr("10.0.2.5");
    dest_info.sin_port = htons(9090);

    // Send the packet out.
    sendto(sock, data, strlen(data), 0,
                (struct sockaddr *)&dest_info, sizeof(dest_info));
    close(sock);
}
```

❖ UDP server

```c
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/ip.h>

void main()
{
    struct sockaddr_in server;
    struct sockaddr_in client;
    int clientlen;
    char buf[1500];

    int sock = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);

    memset((char *) &server, 0, sizeof(server));
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl(INADDR_ANY);
    server.sin_port = htons(9090);

    if (bind(sock, (struct sockaddr *) &server, sizeof(server)) < 0)
        error("ERROR on binding");

    while (1) {
        bzero(buf, 1500);
        recvfrom(sock, buf, 1500-1, 0,
                        (struct sockaddr *) &client, &clientlen);
        printf("%s\n", buf);
    }
    close(sock);
}
```

UDP Applications

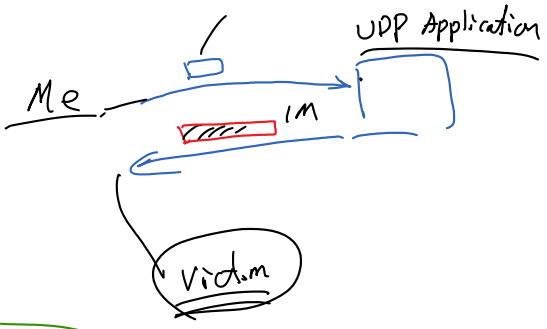
- DNS Protocol
- Video/Audio Streaming
- Real-Time Applications

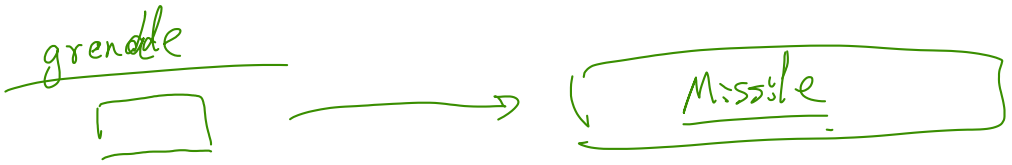
Question

UDP does not preserve order and does not handle packet loss. If an application does care about packet loss and order, can it still use UDP? Please explain.

UDP Flooding Attacks

UDP Amplification Attack



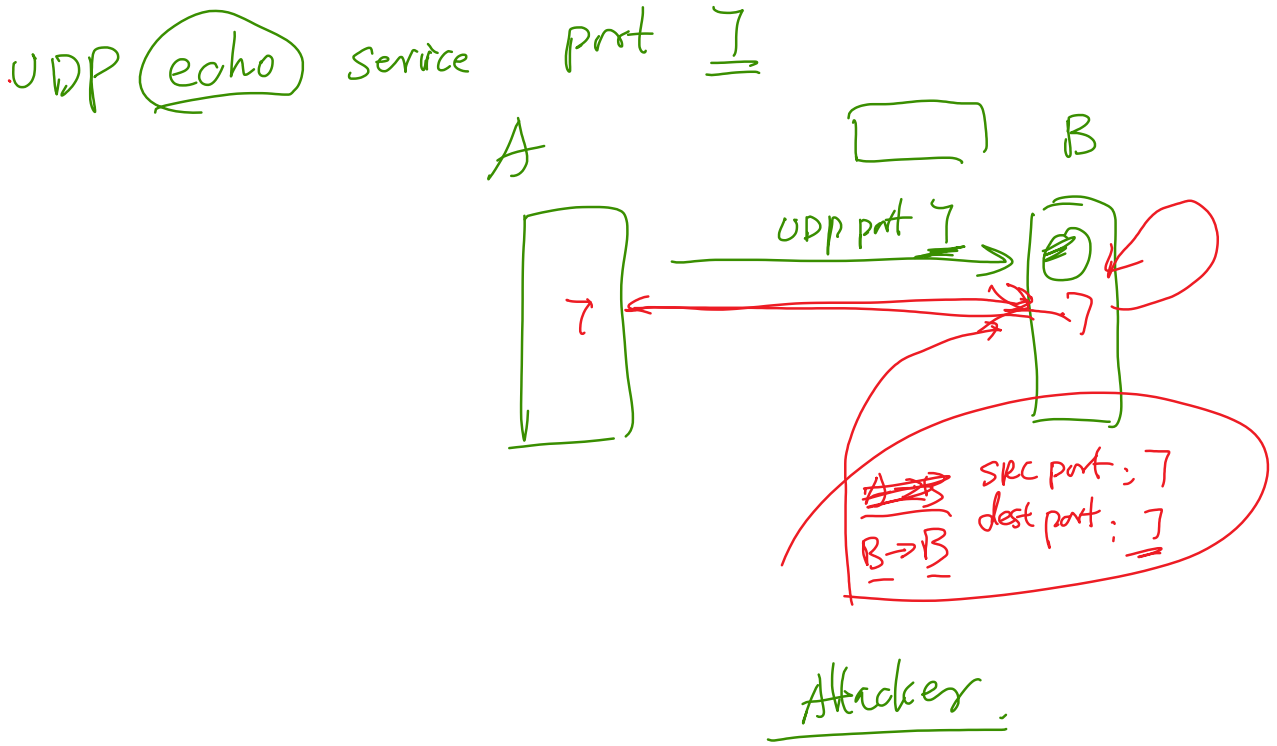


UDP-Based Amplification Attacks

Protocol	Bandwidth Amplification Factor	Vulnerable Command
DNS	28 to 54	see: TA13-088A [4]
NTP	556.9	see: TA14-013A [5]
SNMPv2	6.3	GetBulk request
NetBIOS	3.8	Name resolution
SSDP	30.8	SEARCH request
CharGEN	358.8	Character generation request
QOTD	140.3	Quote request
BitTorrent	3.8	File search
Kad	16.3	Peer list exchange
Quake Network Protocol	63.9	Server info exchange
Steam Protocol	5.5	Server info exchange
Multicast DNS (mDNS)	2 to 10	Unicast query
RIPv1	131.24	Malformed request
Portmap (RPCbind)	7 to 28	Malformed request
LDAP	46 to 55	Malformed request [6]

Source: Christian Rossow

UDP Ping-Pong Attack



Summary: Strategies for DOS attacks