aselsan

STAJ PROJE SUNUMU

Yekta Güngör PARLAK | 19 Eylül 2022

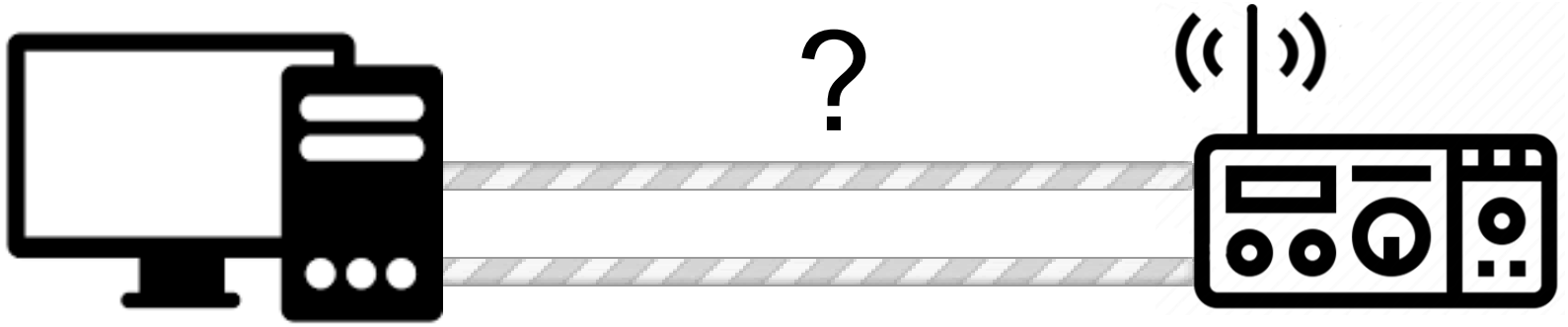
Savunma Sistem Teknolojileri (SST) Sektör Başkanlığı

Gömülü ve Gerçek Zamanlı Yazılım Tasarım Birimi

İçindekiler

Problem Hakkında

Sayısal haberleşmede kullanılan telsizlerin bağlantı durumunun kontrol edilmesi için bir yöntem arıyorduk. Telsizle aramızda bağlantının kurulup kurulmadığına dair bize geri bildirim verecek bir sistem gerekiyordu.



Çözüm Yolu

Problemin çözümü, ağ iletişiminin canlılığını teşhis edebilen bir program geliştirmekti.

Bunu mümkün kılacak programı geliştirmek için ise, ICMP adı verilen İnternet Kontrol Mesajı Protokolü kullanıldı.

Projenin amacı, bu protokolü kullanarak ağ bağlantısının aktif olup olmadığını anlık olarak tespit etmekti.

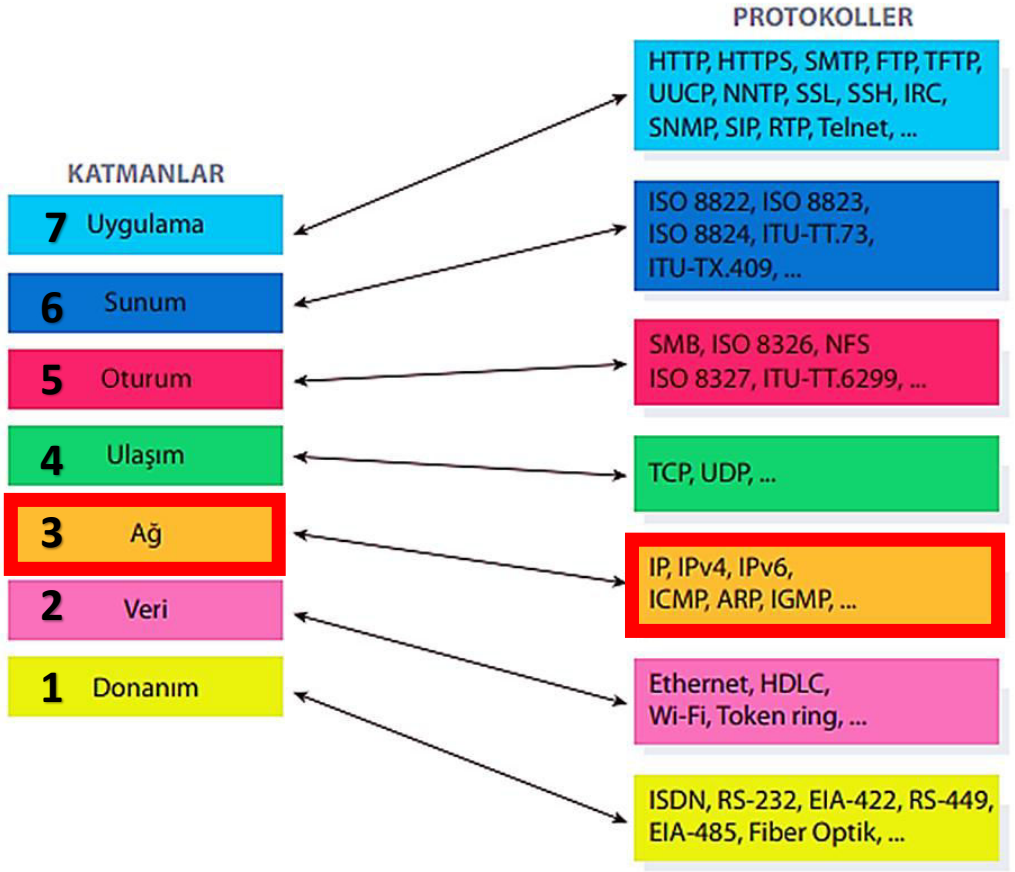
ICMP Nedir?

İki cihaz internet üzerinden bağlanır ve veri paketleri veya datagramlar aracılığıyla veri alışverişinde bulunurlar.

ICMP, verilerin hedefe ulaşıp ulaşmadığını belirleyen bir hata raporlamasıdır ve iki ağ cihazı arasında olan iletişimdeki hatalar, ICMP mesajı olarak kaynak cihaza bildirilir.

Internet protokolünün (IP), yönetimine yardımcı olan bir protokoldür. RFC 792 standardı ile belirlenmiştir.

(IANA) İnternet Tahsisli Numaralar Otoritesi dokümantasyonunda yazıldığı üzere ICMP için atanan IP Protokol numarası 1'dir.



*https://www.rfc-editor.org/rfc/rfc792#ref-1
*https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

ICMP Nedir?

IP DATAGRAM				
	Bits 0-7	**Bits 8-15**	**Bits 16-23**	**Bits 24-31**
IP Başlığı (20 Bytes)	Version	Type of service	Length	
	Identification		Flags	
	TTL	Protocol	Checksum	
	Source IP Address (kaynak)			
	Destination IP Address (varış)			
ICMP Başlığı (8 Bytes)	Type	Code	Checksum	
	Header			
ICMP Data	Payload			

Datagram: Kaynak uçtan alıcı uca gönderilen veri birimidir.

RFC 792 ICMP Standardı Dokümantasyonunda söylendiği üzere ICMP bilgileri, IP paketleri içine yerleştirilir.

RFC 1122 İnternet Sunucuları için Gereksinimler Standardı Dokümantasyonu, Bölüm 3.2.’de şöyle der, "Her bir Host (Ana Bilgisayar), ICMP mesaj işlevini implement etmesi gerekmektedir.”

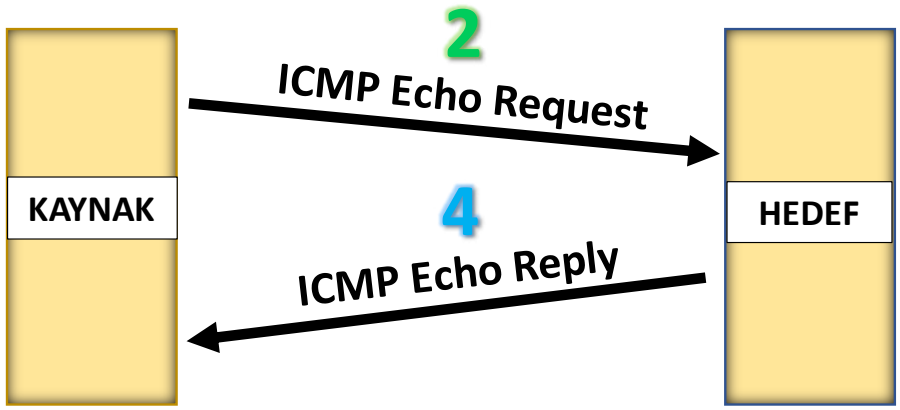
Ayrıca ICMP bilgisi içinde, bir bağlantı noktası (port) yer almaz. Çünkü TCP, UDP gibi protokollerin bulunduğu 4. katmanda değildir.

*https://www.rfc-editor.org/rfc/rfc792#ref-1
*https://www.rfc-editor.org/rfc/rfc1122

4 ICMP Nasıl Çalışır?

aselsan

ICMP protokolü, Ping adı ile anılan Echo Request & Echo Reply yolu ile iletişim sağlar. Nasıl çalıştığına dair genel anlatım, aşağıda verilmiştir. ICMP bu işlemi gerçekleştirmek için, denetlenmek istenen cihaza "echo request" (tip 8) paketi gönderir. Eğer hedef cihaz, bu paketi başarılı şekilde alırsa, "echo reply" (tip 0) mesaj paketini kaynak cihaza gönderir.

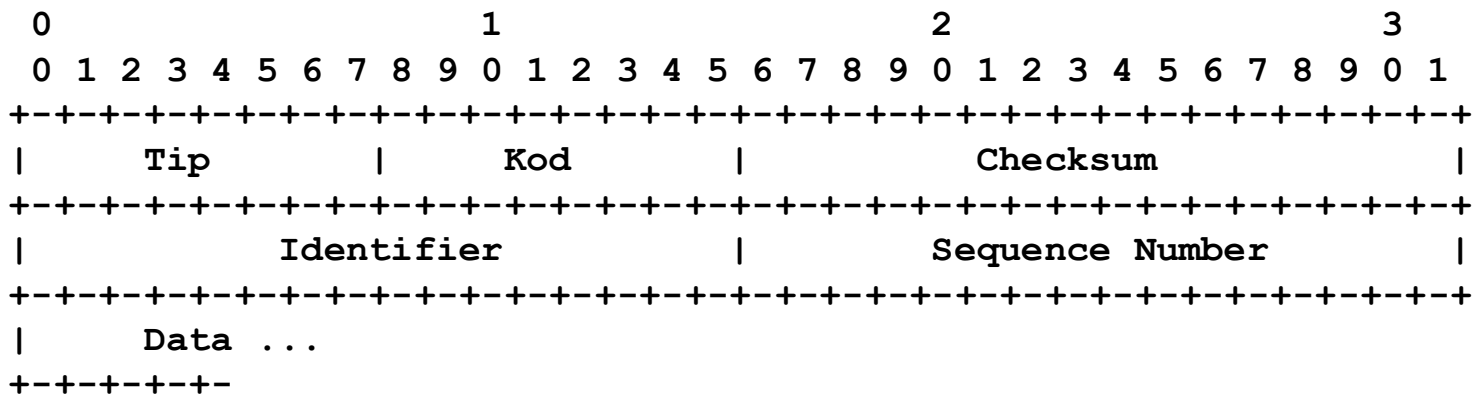


1. **Echo Request işlemini başlat.**
2. **Echo Request'i gönder.**
3. **Echo Reply'ı bekle.**
4. **Echo Reply'ı al.**
5. **Sonucu raporla.**

Tip	Diğer Örnek ICMP Mesaj Tipleri
3	Hedefe Erişilemedi (Destination Not Reachable)
11	Time to Live Exceed (TTL Zaman Aşımı)

*http://www.csc.villanova.edu/~mdamian/Past/networksfa12/Notes/ICMP.pdf

ICMP Mesajı Formatı



Tip (1 Bytes): ICMP mesaj türünü tanımlar. Echo Request mesajları tip 8'dir; Echo Reply mesajları tip 0'dır.

Kod (1 Bytes): Echo Request ve Echo Reply mesajları için 0 olarak tanımlanmıştır.

Checksum (2 Bytes): 16 bitlik sağlama toplamı alanıdır. Eğer bilgi yolda bozulmamışsa, kaynak noktasındaki hesaplanan sayı ile varış noktasındaki hesaplanan sayı aynı çıkar.

Sequence Number (2 Bytes): Gönderilen her Echo Request'te bir artar. Echo Reply, aynı değerle geri dönmesi gerekmektedir.

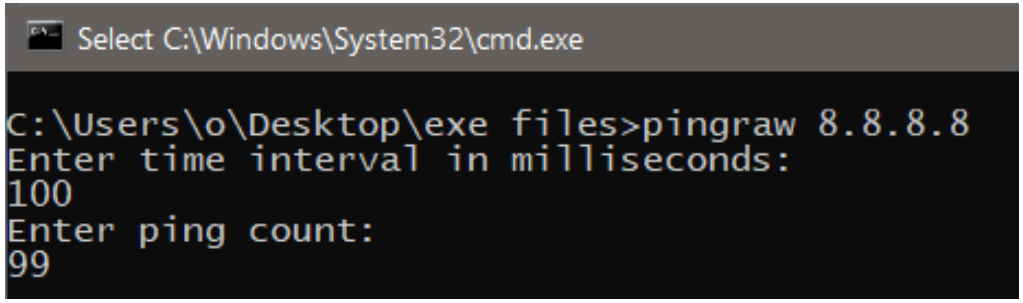
Identifier (2 Bytes): Echo Request ve Echo Reply mesajlarını eşleştirmeye yardımcı olmak için kullanılır.

Data: İsteğe göre veri yerleştirilebilir.

*https://www.rfc-editor.org/rfc/rfc792#ref-1
*http://www.tcpipguide.com/free/t_ICMPGeneralOperation.htm

Program Girdisi

Raw Socket Yaklaşımı



(Platformdan bağımsız, kullanımı daha esnektir.)

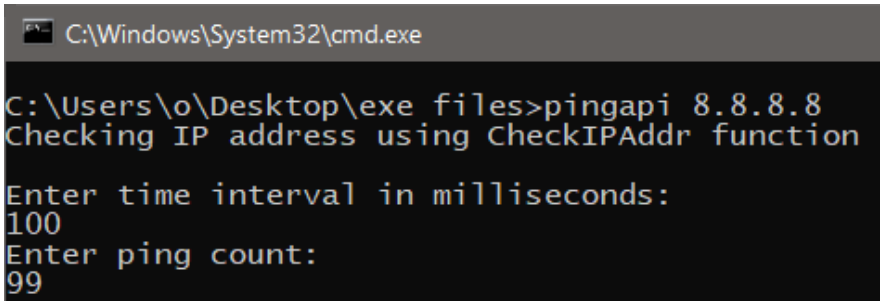
Hedef cihaza bir Echo Request göndermek için sendto işlevi çağrılır:

```
nRet = sendto(sRaw, buff, sizeof(ICMP_Header) + DataLength, 0,
              (SOCKADDR *)&RecvAddr, sizeof(RecvAddr));
```

Echo Reply almak için recvfrom işlevi çağrılır:

```
nRet = recvfrom(sRaw, recvBuf, 1024, 0, (sockaddr*)&from, &nLen);
```

Win32 API Yaklaşımı



(Windows işletim sistemine bağımlıdır.)

Hem Echo Request hem de Echo Reply için IcmpSendEcho işlevi çağrılır:

```
dwRetVal = IcmpSendEcho(IcmpHandle, ipaddr, (LPVOID)
           SendData, sizeof(SendData), &ipOptions,
           ReplyBuffer, ReplySize, Timeout);
```

7 Program Çıktısı (Başarılı Ping)

aselsan

Raw Socket Yaklaşımı

```
>pingraw 8.8.8.8
Enter time interval in milliseconds:
1000
Enter ping count:
10

32 Bytes payload, sent to 8.8.8.8
Round Trip Time (RTT): 125 ms
Time Interval: 1000 ms
Ping Count: 1
OK

32 Bytes payload, sent to 8.8.8.8
Round Trip Time (RTT): 78 ms
Time Interval: 1000 ms
Ping Count: 2
OK
```

Win32 API Yaklaşımı

```
 Parlak_2022\ICMP_Final_ProjectFiles_YektaParlak_2022\exe files
>pingapi 8.8.8.8
Checking IP address using CheckIPAddr function

Enter time interval in milliseconds:
1000
Enter ping count:
10
 Sent ICMP echo request to 8.8.8.8 with 32 Bytes payload
 Received 1 ICMP response successfully
 Round Trip Time (RTT): 90 ms
 Time Interval: 1000 ms
 Ping Count: 1
 TTL: 128
OK
 Sent ICMP echo request to 8.8.8.8 with 32 Bytes payload
 Received 1 ICMP response successfully
 Round Trip Time (RTT): 83 ms
 Time Interval: 1000 ms
 Ping Count: 2
 TTL: 128
OK
```

Program Çıktısı (Başarısız Ping)

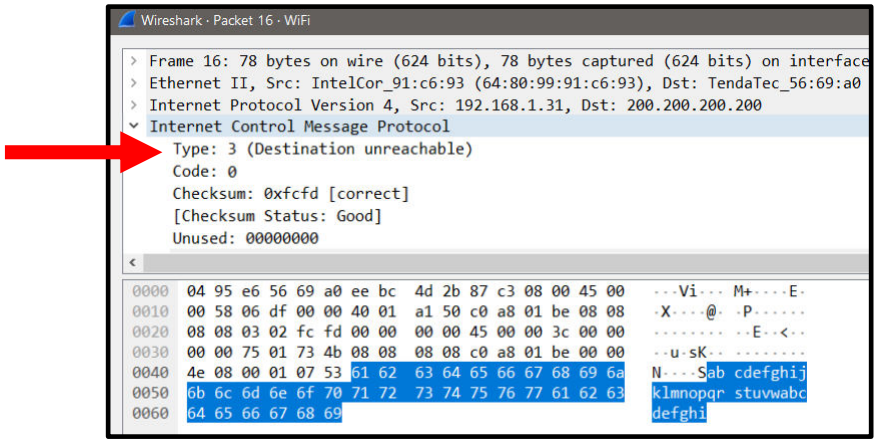
aselsan

Raw Socket Yaklaşımı

```
>pingraw 200.200.200.200
Enter time interval in milliseconds:
100
Enter ping count:
10
 timed out!

C:\Users\Administrator\Desktop\test\Aselsan_Icmp_
aParlak_2022\ICMP_Final_ProjectFiles_YektaParlak_
>
```

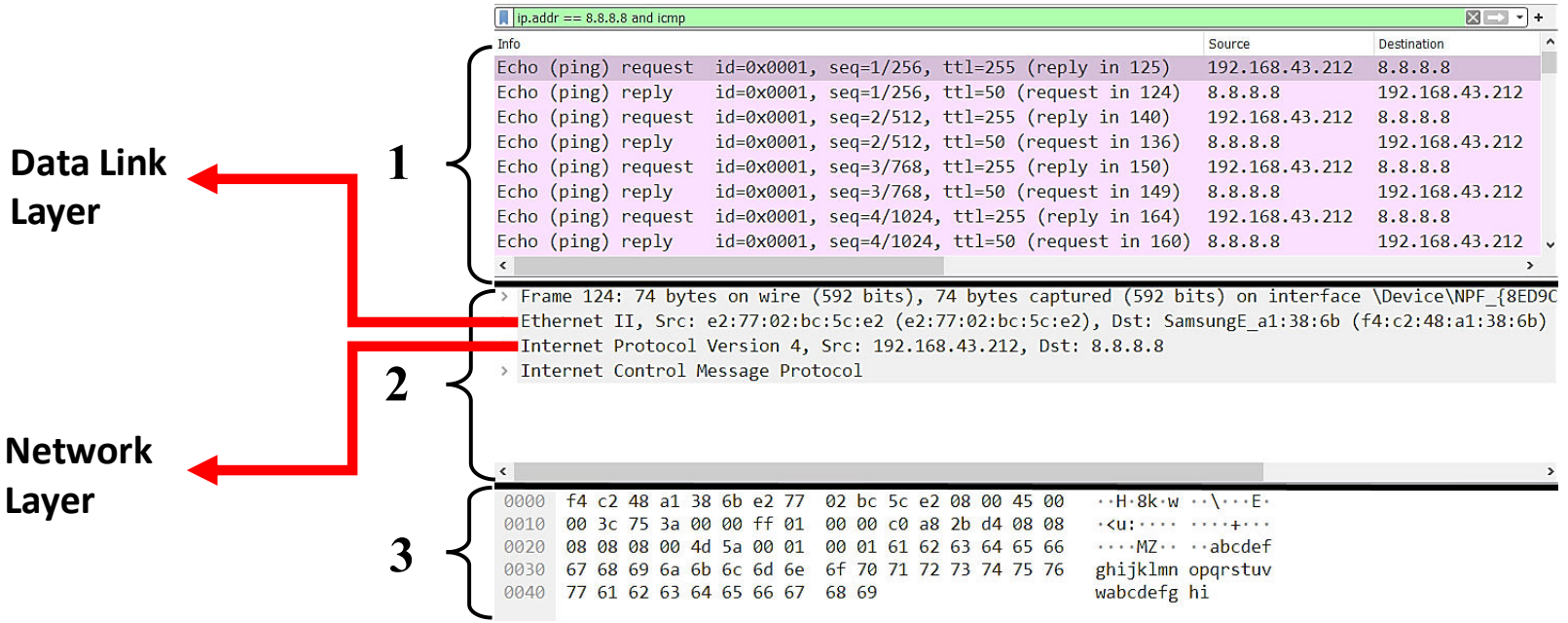
Win32 API Yaklaşımı

```
>pingapi 200.200.200.200
Checking IP address using CheckIPAddr function

Enter time interval in milliseconds:
100
Enter ping count:
10
IcmpSendEcho returned error code: 11010
 Request timed out

C:\Users\Administrator\Desktop\test\Aselsan_Icmp_Code_File_Yekt
aParlak_2022\ICMP_Final_ProjectFiles_YektaParlak_2022\exe files
>
```



Wireshark ICMP Paketi Analizi

aselsan



Bölüm 1:
Bu bölüm, yakalanan tüm paketlerin renkli bir listesidir.

Bölüm 2:
Alan 1'deki paketlerden birine tıkladığımızda, paket yapısı doğrudan alan 2'de gösterilir.

Bölüm 3:
Paketi oluşturan hexadecimal koddur. Sağda, bu hexadecimal kodun Unicode sürümü var.

**https://www.rfc-editor.org/rfc/rfc792#ref-1*
**http://www.tcpipguide.com/free/t_ICMPGeneralOperation.htm*

Wireshark ICMP Paketi Analizi

aselsan

Request Mesajı:

```
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence Number (BE): 8397 (0x20cd)
Sequence Number (LE): 52512 (0xcd20)
```

Reply Mesajı:

```
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence Number (BE): 8397 (0x20cd)
Sequence Number (LE): 52512 (0xcd20)
```

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d5a [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 8397 (0x20cd)
    Sequence Number (LE): 52512 (0xcd20)
    [Response frame: 125]
  > Data (32 bytes)
```

```
0000  f4 c2 48 a1 38 6b e2 77  02 bc 5c e2 08 00 45 00   ··H·8k·w ··\···E·
0010  00 3c 75 3a 00 00 ff 01  00 00 c0 a8 2b d4 08 08   ·<u:···· ····+···
0020  08 08 08 00 4d 5a 00 01  00 01 61 62 63 64 65 66   ····MZ·· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

Tip (1 Byte)

Checksum (2 Bytes)

Kod (1 Byte)

```
Data (32 bytes)
    Data: 6162636465666768696a6b6c6d6e6f707172737475767761626364656667686 9
    [Length: 32]
```

```
0000  f4 c2 48 a1 38 6b e2 77  02 bc 5c e2 08 00 45 00   ··H·8k·w ··\···E·
0010  00 3c 75 3a 00 00 ff 01  00 00 c0 a8 2b d4 08 08   ·<u:···· ····+···
0020  08 08 08 00 4d 5a 00 01  00 01 61 62 63 64 65 66   ····MZ·· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

*https://www.rfc-editor.org/rfc/rfc792#ref-1
*http://www.tcpipguide.com/free/t_ICMPGeneralOperation.htm

10 Hatalı IP Adresi Girişi Tespiti

aselsan

Raw Socket Yaklaşımı

```
>pingraw 8.8.8.8.8.8.8.8.8.8

Wrong IP address format. Host not found!


C:\Users\Administrator\Desktop\test\Aselsan_Icmp_Code_File_Yekt
aParlak_2022\ICMP_Final_ProjectFiles_YektaParlak_2022\exe files
>pingraw 865645869576456

Wrong IP address format. Host not found!


C:\Users\Administrator\Desktop\test\Aselsan_Icmp_Code_File_Yekt
aParlak_2022\ICMP_Final_ProjectFiles_YektaParlak_2022\exe files
>pingraw 88.8.88. 456

Check your argument again!


C:\Users\Administrator\Desktop\test\Aselsan_Icmp_Code_File_Yekt
aParlak_2022\ICMP_Final_ProjectFiles_YektaParlak_2022\exe files
>
```

Win32 API Yaklaşımı

```
aParlak_2022\ICMP_Final_ProjectFiles_YektaParlak_2022\exe files
>pingapi 8.8.8.8.8.8..88
Checking IP address using CheckIPAddr function


8.8.8.8.8.8..88 is an invalid IPv4 address usage
pingapi
C:\Users\Administrator\Desktop\test\Aselsan_Icmp_Code_File_Yekt
aParlak_2022\ICMP_Final_ProjectFiles_YektaParlak_2022\exe files
>pingapi 8.8.88.  46

8.8.88. is an invalid IPv4 address usage
pingapi
C:\Users\Administrator\Desktop\test\Aselsan_Icmp_Code_File_Yekt
aParlak_2022\ICMP_Final_ProjectFiles_YektaParlak_2022\exe files
>
```

11 Gelecekte ekstra neler eklenebilir?

aselsan

- Arayüz tasarlama ve girilecek argümanları menü olarak atama.

- Ek komut satırı argümanı ekleyerek, örneğin istenildiği zaman çok fazla ping göndermek için büyük bir sayı girmek yerine, “-t” yazarak sonsuz defa gönderebilmeyi sağlamak.

Teşekkürler

Yekta Güngör PARLAK

Ekstra Yansı – Raw Socket Yaklaşımı Özeti

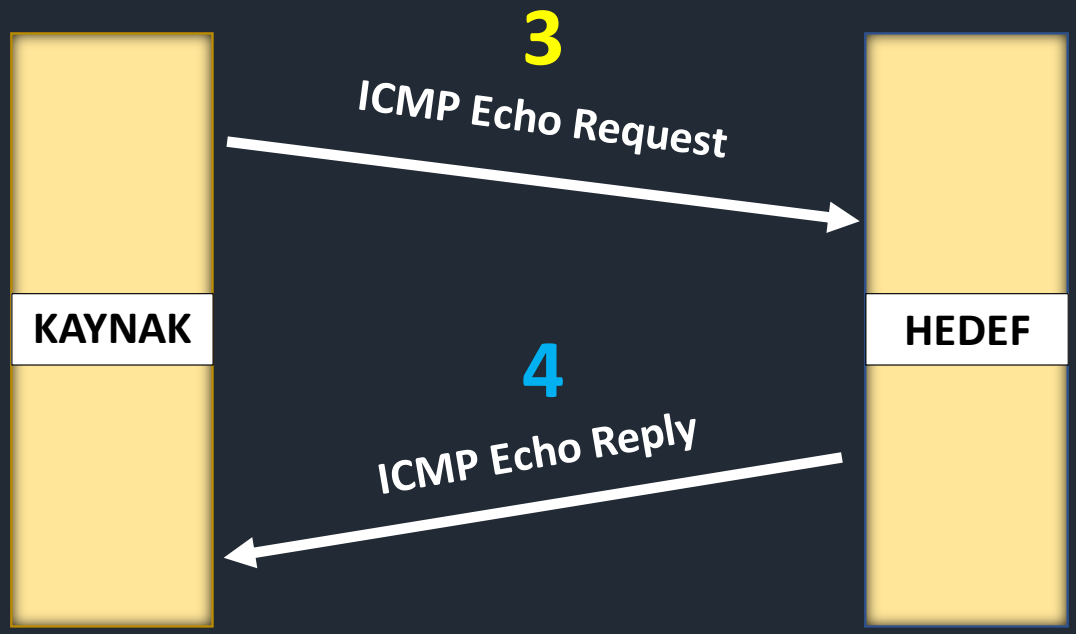
Özetle,

1. SOCK_RAW tipinde bir soket oluşturduk ve IPPROTO_ICMP protokolünü ayarladık. Ardından ise soketin özelliklerini tanımladık.

2. ICMP başlığını tanımladık, oluşturduk.

3. Hedef ağ cihazına bir ICMP Request göndermek için sendto işlevini çağırdık.

4. Herhangi bir ICMP Reply almak için recvfrom işlevini çağırdık.



12 Ekstra Yansı – Raw Socket Yaklaşımı Kodu

aselsan

ICMP Başlığını Tanıt:

```
typedef struct ICMP_Header
{
    unsigned char   icmp_type;
    unsigned char   icmp_code;
    unsigned short  icmp_checksum;
    unsigned short  icmp_id;
    unsigned short  icmp_sequence;
    unsigned long   icmp_timestamp;
} ICMP_Header;
```

ICMP Checksum Hesapla:

```
unsigned short checksum(unsigned short* buff, int size)
{   unsigned long cksum = 0;
    while(size>1)
    {
        cksum += *buff++;
        size -= sizeof(unsigned short);
    }
    if(size)
    cksum += *(char*)buff;
    cksum = (cksum >> 16) + (cksum & 0xffff);
    cksum += (cksum >> 16);
    return (unsigned short)(~cksum);    }
```

Winsock başlat:

```
WSADATA wsaData;
int ret;
ret = WSAStartup(MAKEWORD(2,2), &wsaData);
```

Girilen ve depolanan hedef IP adresini 32 bitlik binary gösterimine dönüştür:

```
char szDestIp[256] ={0};
strcpy(szDestIp,argv[1]);
unsigned long ulDestIP = inet_addr(szDestIp);
```

12 Ekstra Yansı – Raw Socket Yaklaşımı Kodu

aselsan

Raw Socket oluşturmak için kullanılan standart socket() çağrısı şunları içerir:

- Family, TCP veya UDP için olduğu gibi AF_INET'tir
- Soket türü, SOCK_STREAM veya SOCK_DGRAM yerine SOCK_RAW'dir
- Soket protokolünün belirtilmesi gerekir, örn. IPPROTO_ICMP (genellikle UDP veya TCP soketleri için 0'da bırakılır)

Böylece, Soket olarak kullandığımız fonksiyon **(AF_INET, SOCK_RAW, IPPROTO_ICMP)** olur.

SOCK_RAW türünde bir soket oluştur ve IPPROTO_ICMP protokolünü düzenle ve de soketin özelliklerini ayarlayıp hedef adresi oluştur:

```
SOCKET sRaw = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP);

sockaddr_in RecvAddr;
RecvAddr.sin_family = AF_INET;                    // adres ailesi
RecvAddr.sin_port = htons(0);                     // port numarası
RecvAddr.sin_addr.s_addr = inet_addr(szDestIp);   // hedef IP adres
```

12 Ekstra Yansı – Raw Socket Yaklaşımı Kodu

aselsan

Timeout (Zaman Aşımı) Ayarla:

```
//SO_RCVTIMEO ile ayarlanır
int Timeout=10000;
ret = setsockopt(sRaw, SOL_SOCKET, SO_RCVTIMEO,
      (char*)&Timeout, sizeof(Timeout));
```

ICMP Paketini Oluştur:

```
char buff[sizeof(ICMP_HDR) + 32];
ICMP_HDR* pIcmp = (ICMP_HDR*)buff;
```

ICMP Paket Verisinin İçini Doldur:

```
pIcmp->icmp_type = 8;
pIcmp->icmp_code = 0;
pIcmp->icmp_id = (USHORT)::GetCurrentProcessId();
pIcmp->icmp_checksum = 0;
pIcmp->icmp_sequence = 0;
```

Data (Veri) Kısmını Doldur (Opsiyonel):

```
memset(&buff[sizeof(ICMP_HDR)], 'Y', 32);
```

ICMP paketini göndermeye ve almaya başla:

```
USHORT nSeq = 1;              //Sequence number Başlangıcı
char recvBuf[1024] = { 0 };   /* Receive Buffer'ı tanıt. Paketin
                              ikili verilerinin depolandığı bellek
                              alanına ihtiyacımız var */
sockaddr_in from;             // Alınan verinin IP adresini kaydet
int nLen = sizeof(from);     //Adres uzunlugu
```

Hedef cihaza bir ICMP Request göndermek için sendto işlevini çağır:

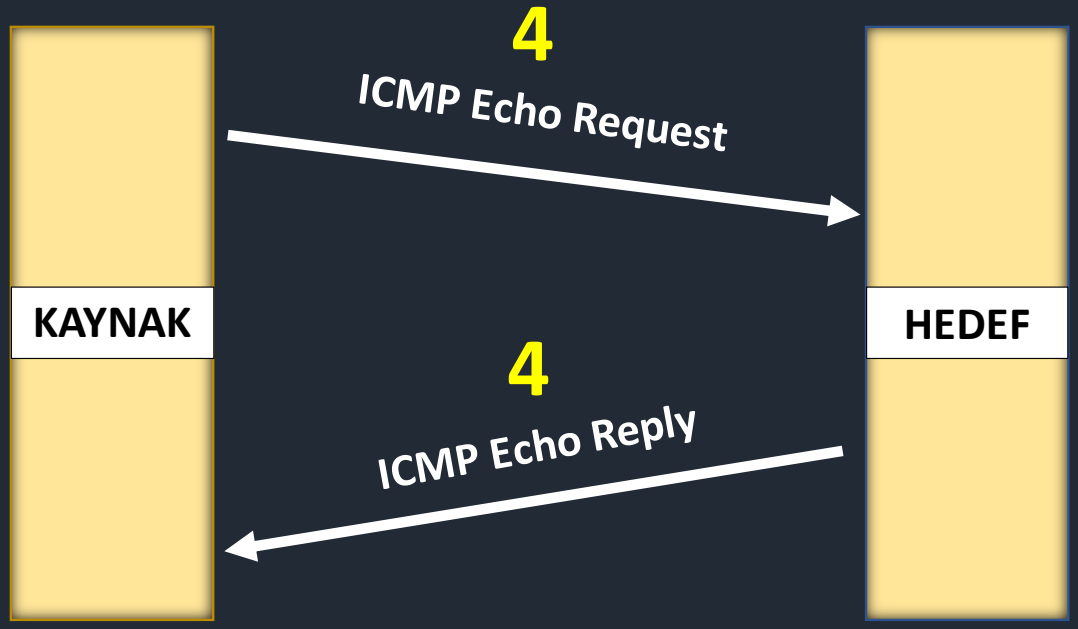
```
nRet = sendto(sRaw, buff, sizeof(ICMP_Header) + DataLength, 0,
              (SOCKADDR *)&RecvAddr, sizeof(RecvAddr));
```

ICMP Reply almak için recvfrom işlevini çağır:

```
nRet = recvfrom(sRaw, recvBuf, 1024, 0, (sockaddr*)&from, &nLen);
```

Ekstra Yansı – Win32 API Yaklaşımı Özeti

aselsan



Özetle,

1. iphlpapi.dll (dynamic link library) tanıtıldı.

2. Bir Icmp Handle oluşturmak için IcmpCreateFile işlevini kullandık.

3. API parametrelerini oluşturun (IcmpHandle, ipaddr, SendData, sizeof(SendData), &ipOptions, ReplyBuffer, ReplySize, Timeout)

4. Echo Request ve Echo Reply için IcmpSendEcho işlevini çağırdık.

5. IcmpCreateFile çağrısıyla açılan tanıtıcıyı, IcmpCloseHandle işlevi ile kapattık.

Ekstra Yansı – Win32 API Yaklaşımı Kodu

aselsan

iphlpapi.dll (dynamic link library) tanıt:

```
#pragma comment(lib, "iphlpapi.lib")
#include <iphlpapi.h>
```

IcmpSendEcho işlevini çağır:

```
dwRetVal = IcmpSendEcho(IcmpHandle,
           ipaddr,
           (LPVOID) SendData,
           sizeof(SendData),
           &ipOptions,
           ReplyBuffer,
           ReplySize,
           Timeout);
```

IcmpCreateFile ile Handle oluştur:

```
IcmpHandle = IcmpCreateFile();
```

IcmpCreateFile çağrısıyla açılan tanıtıcıyı, IcmpCloseHandle işlevi ile kapat:

```
IcmpCloseHandle(IcmpHandle);
```

Ekstra Yansı – Win32 API Yaklaşımı Kodu

aselsan

IcmpSendEcho Parametre Detayları:

IcmpHandle:	IcmpCreateFile işlevi tarafından döndürülen tanıtıcı.
DestinationAddress:	IPAddr yapısı biçiminde Echo Request'in IPv4 hedef adresi.
RequestData:	Echo Request'te gönderilecek verileri içeren bir arabelleğe işaretleyen.
RequestSize:	RequestData parametresinin işaret ettiği Echo Request veri arabelleğinin Byte cinsinden boyutu.
ReplyBuffer:	Echo Request'te verilen Echo Reply'ı tutacak bir arabellektir.
ReplySize:	Echo Reply arabelleğinin Byte cinsinden ayrılan boyutudur.
Timeout:	Echo Reply'ı beklemek için milisaniye cinsinden süre.