Ingineria Programării în Reţea (IPR)

Unelte software utile în proiectarea şi implementarea unei aplicaţii de reţea

Ping

- Instrumentul cel mai popular pentru testarea conectivităţii într-o reţea IP
- Disponibil pe majoritatea sistemelor de operare
- Codul sursa: http://ftp.arl.army.mil/pub/misc/ping.shar
- Sintaxa:

```
ping –arg. adresaIP(sau hostname)
```

Ping – funcţionare

- Calculatorul sursă trimite pachete de tip ICMP “echo request” către destinaţie şi aşteaptă răspunsuri ICMP “echo response”
- La final se calculează statistici privind:

nr. de pachete trimise/recepţionate, timpul de round trip (min, media, max, mdev)

Ping – exemple (1)

```
linux-413h:~ # ping 192.168.34.7
PING 192.168.34.7 (192.168.34.7) 56(84) bytes of data.
64 bytes from 192.168.34.7: icmp_seq=1 ttl=64 time=0.367 ms
64 bytes from 192.168.34.7: icmp_seq=2 ttl=64 time=0.991 ms
64 bytes from 192.168.34.7: icmp_seq=3 ttl=64 time=0.312 ms
64 bytes from 192.168.34.7: icmp_seq=4 ttl=64 time=0.319 ms
64 bytes from 192.168.34.7: icmp_seq=5 ttl=64 time=0.467 ms
64 bytes from 192.168.34.7: icmp_seq=6 ttl=64 time=0.393 ms
64 bytes from 192.168.34.7: icmp_seq=7 ttl=64 time=0.343 ms

--- 192.168.34.7 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time
   6001ms
rtt min/avg/max/mdev = 0.312/0.456/0.991/0.223 ms
```

Ping – exemple (2)

```
C:\Documents and Settings\ro1v00r7>ping www.ieee.org

Pinging a165.g.akamai.net [88.221.26.88] with 32
   bytes of data:

Reply from 88.221.26.88: bytes=32 time=44ms TTL=56
Reply from 88.221.26.88: bytes=32 time=43ms TTL=56
Reply from 88.221.26.88: bytes=32 time=42ms TTL=56
Reply from 88.221.26.88: bytes=32 time=43ms TTL=56

Ping statistics for 88.221.26.88:
    Packets: Sent = 4, Received = 4, Lost = 0 (0%
   loss),
Approximate round trip times in milli-seconds:
    Minimum = 42ms, Maximum = 44ms, Average = 43ms
```

Ping – protocolul ICMP

- Mesajul de “echo request” – type 8

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type = 8								Code = 0								Header Checksum															
Identifier																Sequence Number															
Data :::																															

- Mesajul de “echo reply” – type 0

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type = 0								Code = 0								Header Checksum															
Identifier																Sequence Number															
Data :::																															

Ping – 4 paşi pentru testarea unei conexiuni

1. `ping 127.0.0.1` – testarea interfeţei de loopback (verificăm că protocolul TCP/IP e instalat)
2. `ping adresa_IP_locala` – testarea interfeţei de reţea locală
3. `ping adresa_IP_gateway_default` – testarea conectivităţii către exterior
4. `ping adresa_IP_destinaţie` – testare conectivitate cu echipamentul destinaţie

Ping – TTL (TimeToLeave)

- TTL = nr. max. de routere prin care pachetul poate trece înainte de a fi aruncat; fiecare router decrementeză acest câmp cu o unitate

TTL exemplu

- Aflăm TTL-ul maşinii locale:

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

- Ping către maşina locală:

```
PING 79.116.226.134 (79.116.226.134) 56(84) bytes of data.
64 bytes from 79.116.226.134: icmp_seq=1 ttl=120 time=21.7 ms
64 bytes from 79.116.226.134: icmp_seq=2 ttl=120 time=5.48 ms
```

TTL exemplu (continuare)

- Verificare TTL:

```
Tracing route to sg5.gazduire.ro [80.96.148.28]
over a maximum of 30 hops:

  1     2 ms     1 ms     1 ms  nas-pppoe-bartolomeu2.brasov.rdsnet.ro
    [86.125.87.71]
  2     2 ms     2 ms     2 ms  dr05.brasov.rdsnet.ro [86.125.87.65]
  3     2 ms     2 ms     2 ms  213-154-120-221.rdsnet.ro [213.154.120.221]
  4     2 ms     2 ms     2 ms  213-154-120-161.rdsnet.ro [213.154.120.161]
  5     4 ms     5 ms     5 ms  213-154-124-105.rdsnet.ro [213.154.124.105]
  6     5 ms    27 ms     5 ms  qr04.bucuresti.rdsnet.ro [213.154.97.2]
  7     5 ms     5 ms     5 ms  82-76-244-98.rdsnet.ro [82.76.244.98]
  8     5 ms     7 ms     5 ms  c3560-v02.gazduire.ro [217.156.103.242]
  9     5 ms     5 ms     5 ms  sg5.gazduire.ro [80.96.148.28]

Trace complete.
```

- **8** (=**128**-**120**) routere (noduri) între sursă şi destinaţie

Atacuri folosind ping

- Ping of death:

-atacul e bazat pe pachete ICMP fragmentate (RFC 791 nu permite pachete mai mari de 65,535 bytes)

-la destinaţie pachetul este reasamblat şi poate genera un buffer overflow, urmat de crash-ul sistemului

- Ping flood:

-un atac simplu de tip DOS (Denial Of Service)

-reuşeşte dacă banda atacatorului este mai mare decât banda victimei

Ping şi securitatea

- Multe firewall-uri din ziua de azi fac calculatoarele "invizibile" (nu răspund la ping)
- Unii provideri de Internet filtrează prin routerele lor mesajele de tip ICMP, pentru a opri un eventual atac cu ping

Ping din Internet

- Câteva site-uri de unde puteţi da ping din Internet:

http://www.pingme.ro/

http://www.eckes.org/modules.php?name=Content&pa=showpage&pid=2

Traceroute

- Utilitar care permite vizualizarea rutei de la sursă la destinaţie (nivelul 3)
- Sintaxa:

```
traceroute -arg. adresaIP(sau
  hostname)
```

Sau, pentru Windows:

```
tracert -arg. adresaIP(sau
  hostname)
```

Traceroute – exemple (1)

```
Tracing route to google.com [72.14.207.99]
over a maximum of 30 hops:

  1    11 ms    15 ms    15 ms  nas-pppoe-bartolomeu2.brasov.rdsnet.ro
    [86.125.87.71]
  2    15 ms    15 ms    15 ms  dr05.brasov.rdsnet.ro [86.125.87.65]
  3    15 ms    15 ms    15 ms  213-154-120-221.rdsnet.ro [213.154.120.221]
  4    15 ms    15 ms    15 ms  213-154-120-161.rdsnet.ro [213.154.120.161]
  5    31 ms    30 ms    46 ms  br01.frankfurt.rdsnet.ro [213.154.125.41]
  6   187 ms   171 ms    31 ms  xr01.frankfurt.rdsnet.ro [213.154.125.34]
  7    31 ms    30 ms    31 ms  72.14.198.221
  8    31 ms    30 ms    31 ms  209.85.255.170
  9    46 ms    62 ms    46 ms  72.14.233.104
 10   109 ms   140 ms   124 ms  72.14.236.220
 11   124 ms   124 ms   124 ms  72.14.233.113
 12   124 ms   124 ms   124 ms  66.249.94.90
 13   124 ms   124 ms   124 ms  66.249.94.50
 14   124 ms   124 ms   124 ms  eh-in-f99.google.com [72.14.207.99]

Trace complete.
```

Traceroute – exemple (2)

```
traceroute to cern.ch (137.138.28.241), 40 hops max, 40 byte packets
 1  129.194.9.1  0.506 ms  0.370 ms  0.310 ms
 2  192.33.214.3  0.544 ms  0.532 ms  0.533 ms
 3  130.59.36.121  0.748 ms  0.645 ms  0.646 ms
 4  130.59.36.141  0.869 ms  0.808 ms  0.749 ms
 5  130.59.36.209  0.784 ms  0.765 ms  0.756 ms
 6  192.65.184.222  0.838 ms  0.793 ms  0.784 ms
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  137.138.28.241  1.675 ms  1.469 ms  1.437 ms
```

Traceroute - utilizare

- Cu traceroute se pot identifica rutele congestionate
- Se poate determina cel mai rapid mirror pentru download

Traceroute - securitate

- Unele routere/noduri îşi “ascund” identitatea pentru a oferi protecţie
- Unii atacatori au folosit traceroute pentru a avea o imagine asupra arhitecturii unei reţele

Tcpdump

- Utilitar destinat monitorizării traficului în reţea şi achiziţionării de date.
- Plăcile de reţea Ethernet capurează la nivelul data-link doar pachetele destinate lor sau cele de broadcast. Pentru a captura toate frame-urile tcpdump trece interfaţa într-un mod special de lucru, numit promiscuous.

Tcpdump - utilizare

- Sintaxa: `tcpdump -arg.`
- Exemple de argumente:
  - `-i eth0` capturează doar pe interfaţa eth0
  - `-c21` se opreşte după 21 packete
  - ....

Tcpdump – exemple (1)

```
7:39:05.839538 IP 192.168.34.1 > linux: ICMP echo request, id 512, seq 7680,
    length 40
07:39:05.839592 IP linux > 192.168.34.1: ICMP echo reply, id 512, seq 7680,
    length 40
07:39:06.803323 IP 192.168.34.1 > linux: ICMP echo request, id 512, seq 7936,
    length 40
07:39:06.811921 IP linux > 192.168.34.1: ICMP echo reply, id 512, seq 7936,
    length 40
07:39:07.806359 IP 192.168.34.1 > linux: ICMP echo request, id 512, seq 8192,
    length 40
07:39:07.814251 IP linux > 192.168.34.1: ICMP echo reply, id 512, seq 8192,
    length 40
07:39:08.841584 IP 192.168.34.1 > linux: ICMP echo request, id 512, seq 8448,
    length 40
07:39:08.841705 IP linux > 192.168.34.1: ICMP echo reply, id 512, seq 8448,
    length 40
```

Tcpdump – exemple (2)

```
07:39:10.837408 arp who-has 192.168.34.1 tell linux
07:39:10.837731 arp reply 192.168.34.1 is-at 00:50:56:c0:00:01 (oui Unknown)
07:39:13.483936 IP 192.168.34.1.4468 > linux.ssh: P 3196084637:3196084765(128)
    ack 2284433388 win 65215
07:39:13.484004 IP 192.168.34.1.4468 > linux.ssh: F 128:128(0) ack 1 win 65215
07:39:13.490429 IP linux.ssh > 192.168.34.1.4468: F 1:1(0) ack 129 win 25024
07:39:13.490917 IP 192.168.34.1.4468 > linux.ssh: . ack 2 win 65215
07:39:15.892148 IP 192.168.34.1.4788 > linux.ssh: S 1439964200:1439964200(0)
    win 65535 <mss 1460,nop,nop,sackOK>
07:39:15.892427 IP linux.ssh > 192.168.34.1.4788: S 4043622980:4043622980(0)
    ack 1439964201 win 5840 <mss 1460,nop,nop,sackOK>
07:39:15.892802 IP 192.168.34.1.4788 > linux.ssh: . ack 1 win 65535
07:39:15.952458 IP linux.ssh > 192.168.34.1.4788: P 1:21(20) ack 1 win 5840
07:39:15.954542 IP 192.168.34.1.4788 > linux.ssh: P 1:45(44) ack 21 win 65515
07:39:15.954790 IP linux.ssh > 192.168.34.1.4788: . ack 45 win 5840
07:39:15.955172 IP 192.168.34.1.4788 > linux.ssh: P 45:381(336) ack 21 win
    65515
07:39:15.955467 IP linux.ssh > 192.168.34.1.4788: . ack 381 win 6432
```

Tcpdump explicaţii

- **07:39:15.954542** – Timpul în care pachetul TCP a intrat în reţeaua locală
- **IP** – tip pachet
- **192.168.34.1.4788** – adresa şi portul sursă
- **linux.ssh** – destinaţia şi portul destinaţie
- **P 1:45(44) ack 21 win 65515** – proprietăţi tcp: intervalul numerelor de secvenţă, urmat de nr. de octeţi din interval, dimensiunea ferestrei pe care sursa e capabilă sa o accepte

Wireshark

- Cel mai popular analizor de reţea
- Cunoscut iniţial sub numele de Ethereal
- Similar cu tcpdump, diferenţa majoră o reprezintă GUI-ul

Wireshark – features

- Suportă 935 de protocoale
- Pachetele pot fi capturate din medii ca Ethernet, FDDI (Fiber Distributed Data Interface), PPP (Point to Point Protocol), Token Ring, IEEE802.11, Loopback
- Permite definirea de filtre

Wireshark – filtre

- Operatori de comparare suportaţi:
  - ==
  - !=
  - >
  - <
  - <=
  - >=

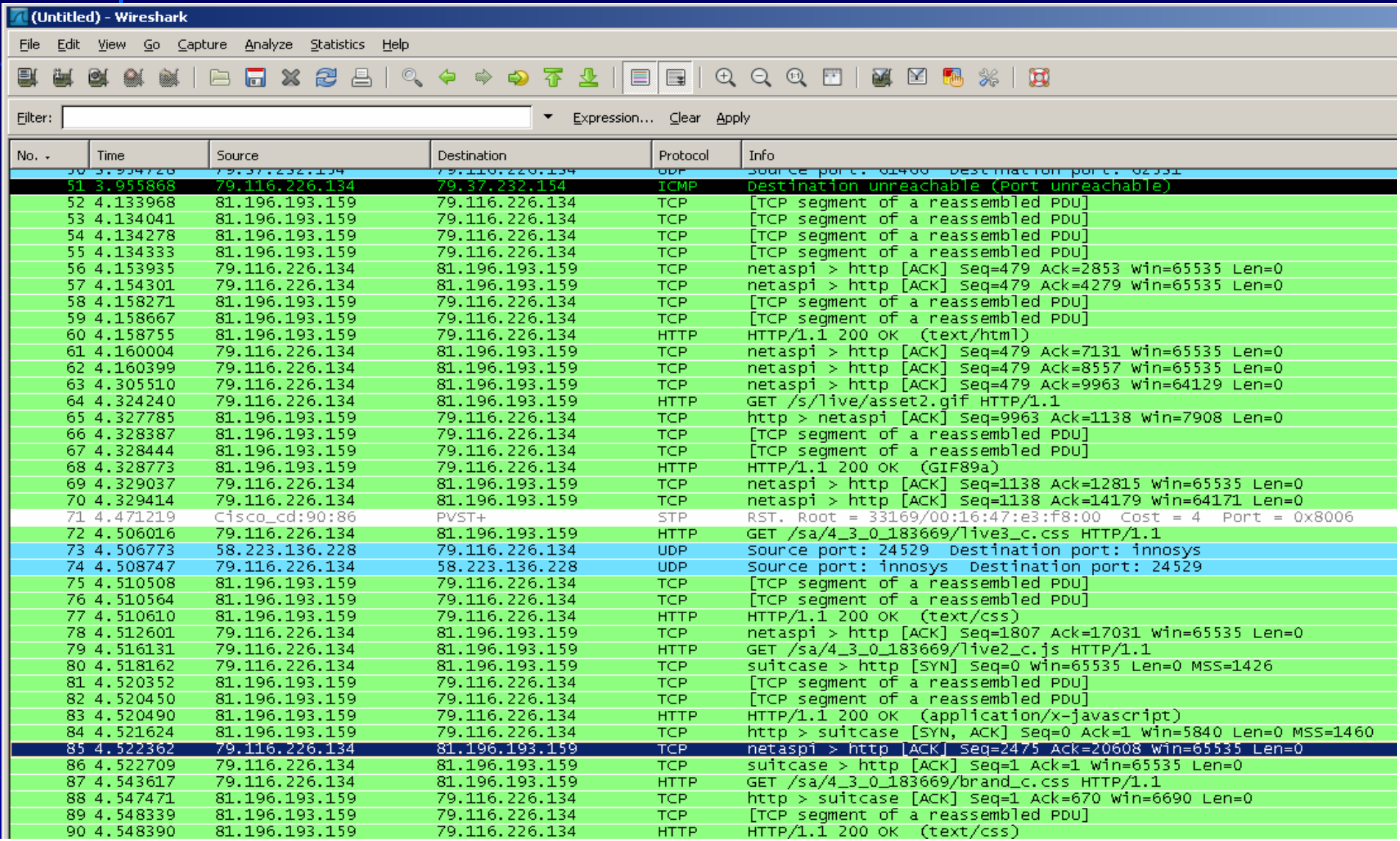
Wireshark – filtre

- Operaţii logice
  - &&
  - ||
  - ^^
  - !

Wireshark – exemple aplicare filtre

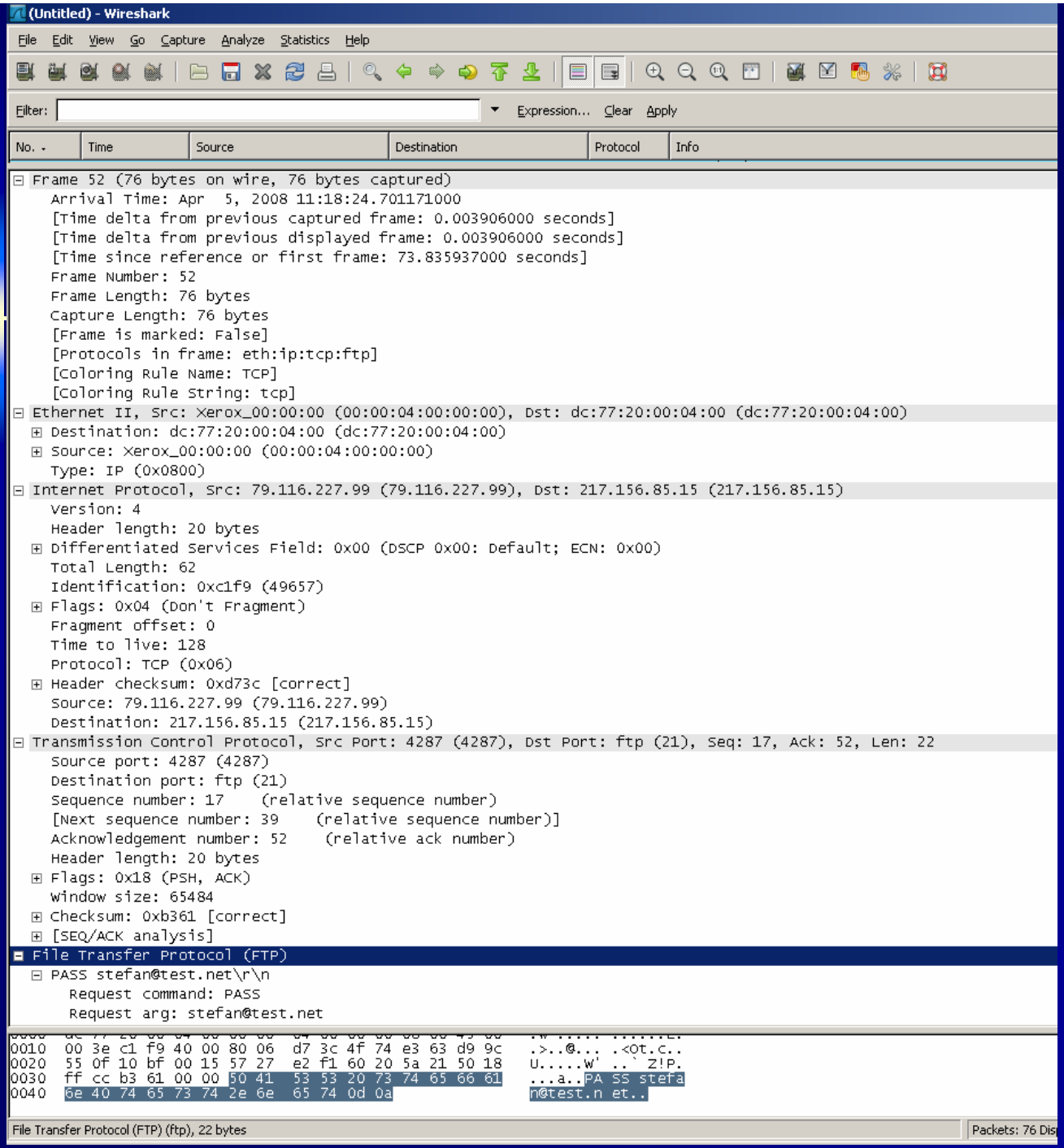
- `dns || icmp`
- `ip.addr == 10.10.1.1`
- `ip.src != 10.1.2.3 and ip.dst != 10.4.5.6`
- `tcp.port == 21`
- `tcp.dstport == 23`
- `tcp.flags`
- `tcp.flags.syn == 0x02`

Wireshark – exemple (1)



Wireshark – exemple (2)

10 6.092239	217.156.85.15	79.116.226.23	TCP	ftp > dvapps [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
11 6.093662	79.116.226.23	217.156.85.15	TCP	dvapps > ftp [ACK] Seq=1 Ack=1 Win=65535 Len=0
12 6.099003	217.156.85.15	79.116.226.23	FTP	Response: 220 FTP server ready.
13 6.104492	79.116.226.23	217.156.85.15	FTP	Request: USER anonymous
14 6.108388	217.156.85.15	79.116.226.23	TCP	ftp > dvapps [ACK] Seq=24 Ack=17 Win=5840 Len=0
15 6.110172	217.156.85.15	79.116.226.23	FTP	Response: 331 Any password will work
16 6.114153	79.116.226.23	217.156.85.15	FTP	Request: PASS stefan@test.net
17 6.118241	217.156.85.15	79.116.226.23	FTP	Response: 230 Any password will work
18 6.122343	79.116.226.23	217.156.85.15	FTP	Request: SYST
19 6.126944	217.156.85.15	79.116.226.23	FTP	Response: 215 UNIX Type: L8
20 6.131093	79.116.226.23	217.156.85.15	FTP	Request: FEAT
21 6.137110	217.156.85.15	79.116.226.23	FTP	Response: 211-Extensions supported:
22 6.139831	Cisco_cd:90:86	PVST+	STP	RST. Root = 33169/00:16:47:e3:f8:00 Cost = 4 Port = 0x8006
23 6.162457	79.116.226.23	217.156.85.15	FTP	Request: PWD
24 6.169566	217.156.85.15	79.116.226.23	FTP	Response: 257 "/" is your current location
25 6.187017	79.116.226.23	217.156.85.15	FTP	Request: TYPE A
26 6.192078	217.156.85.15	79.116.226.23	FTP	Response: 200 TYPE is now ASCII


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
Frame 52 (76 bytes on wire, 76 bytes captured)
Arrival Time: Apr 5, 2008 11:18:24.701171000
[Time delta from previous captured frame: 0.003906000 seconds]
[Time delta from previous displayed frame: 0.003906000 seconds]
[Time since reference or first frame: 73.835937000 seconds]
Frame Number: 52
Frame Length: 76 bytes
Capture Length: 76 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:tcp:ftp]
[Coloring Rule Name: TCP]
[Coloring Rule String: tcp]
Ethernet II, Src: Xerox_00:00:00 (00:00:04:00:00:00), Dst: dc:77:20:00:04:00 (dc:77:20:00:04:00)
Destination: dc:77:20:00:04:00 (dc:77:20:00:04:00)
Source: Xerox_00:00:00 (00:00:04:00:00:00)
Type: IP (0x0800)
Internet Protocol, Src: 79.116.227.99 (79.116.227.99), Dst: 217.156.85.15 (217.156.85.15)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 62
Identification: 0xc1f9 (49657)
Flags: 0x04 (Don't Fragment)
Fragment offset: 0
Time to live: 128
Protocol: TCP (0x06)
Header checksum: 0xd73c [correct]
Source: 79.116.227.99 (79.116.227.99)
Destination: 217.156.85.15 (217.156.85.15)
Transmission Control Protocol, Src Port: 4287 (4287), Dst Port: ftp (21), Seq: 17, Ack: 52, Len: 22
Source port: 4287 (4287)
Destination port: ftp (21)
Sequence number: 17 (relative sequence number)
[Next sequence number: 39 (relative sequence number)]
Acknowledgement number: 52 (relative ack number)
Header length: 20 bytes
Flags: 0x18 (PSH, ACK)
Window size: 65484
Checksum: 0xb361 [correct]
[SEQ/ACK analysis]
File Transfer Protocol (FTP)
PASS stefan@test.net\r\n
Request command: PASS
Request arg: stefan@test.net
0010 00 3e c1 f9 40 00 80 06 d7 3c 4f 74 e3 63 d9 9c .>..@... .<Ot.c..
0020 55 0f 10 bf 00 15 57 27 e2 f1 60 20 5a 21 50 18 U.....W' ..` Z!P.
0030 ff cc b3 61 00 00 50 41 53 53 20 73 74 65 66 61 ...a..PA SS stefa
0040 6e 40 74 65 73 74 2e 6e 65 74 0d 0a n@test.n et..
File Transfer Protocol (FTP) (ftp), 22 bytes
Packets: 76 Dis

Iptraf

- Utilitar de consolă pentru Linux
- Oferă statistici privind interfeţele de reţea: traficul in şi out (în kbits/sec şi pachete/sec), traficul total pe o interfaţă
- Statistici în funcţie de tipul pachelor: IP, Non-IP
- Trafic detaliat în funcţie de protocol: TCP, UDP, ICMP, Other IP

Iptraf

- Suportă Ethernet, FDDI, ISDN, SLIP, PPP şi Loopback
- Sunt suportate majoritatea plăcilor de reţea deorece este folosită interfaţa de tip raw socket a kernelului Linux
- Logging
- Un modul de statisică pe întregul LAN

Iptraf – exemplu