9 Network Forensics

On-line lecture: http://asecuritysite.com/subjects/chapter09

9.1 Objectives

The key objectives of this unit are to:

- Understand some of the methodologies used in network forensics.
- Provide an in-depth understanding of the key network protocols, including IP, TCP, ARP, ICMP, DNS, Application Layer protocols, and so on.
- Define a range of audit sources for network activity.

9.2 Introduction

The requirement for network forensics can be many fold, including deconstructing an internal/external network attack, a criminal investigation, and debugging a problem with a system. This unit is focus on the methodology for analysing network traffic, and in determining the key parameters that can be used to determine an evidence base for an investigation.

9.3 The key protocols

The key networking element that are typically used in an analysis of network traffic are:

- **TCP flags**. Most of the communications which occurs on the Internet involves client-server communications using TCP. The start of a connection normally involves an exchange of SYN, SYN/ACK and ACK TCP segments. Thus the start of a connection normally involves this exchange. At the end of the negotiation the TCP ports will be identified.
- **ARP activity**. This is often a sign of a host machine connect to another computer on the local network, or to the default gateway.
- **ICMP activity**. This is often a sign of the discovery of hosts, or for the route to a host.
- **DNS activity**. This is typically seen before some sort of remote access to a host.
- **Application Protocol activity**. This normally identifies the details of the actual transaction.

9.4 Ethernet, IP and TCP headers

Data is normally encapsulated with headers in order to pass the required information to be processed correctly. Figure 9.1 shows an example of a layered model, with the Application Layer (Layer 5-7), the Transport Layer (Layer 4), the Network Layer (Layer 3), the Data Link Layer (Layer 2) and the Physical Layer (Layer 1). As the data goes through these layers extra information is added in sequence, with most layers adding the extra information at the start of the encapsulated data. The Data Link Layer is typically different as it adds information at the start and the end, as this will define the start and end of the encapsulated data. Normally at Layer 2 the transmitted encapsulated data is known as a **data frame**, while at Layer 3 it is known as a **data packet**, and at Layer 4 it is known as a **segment**.

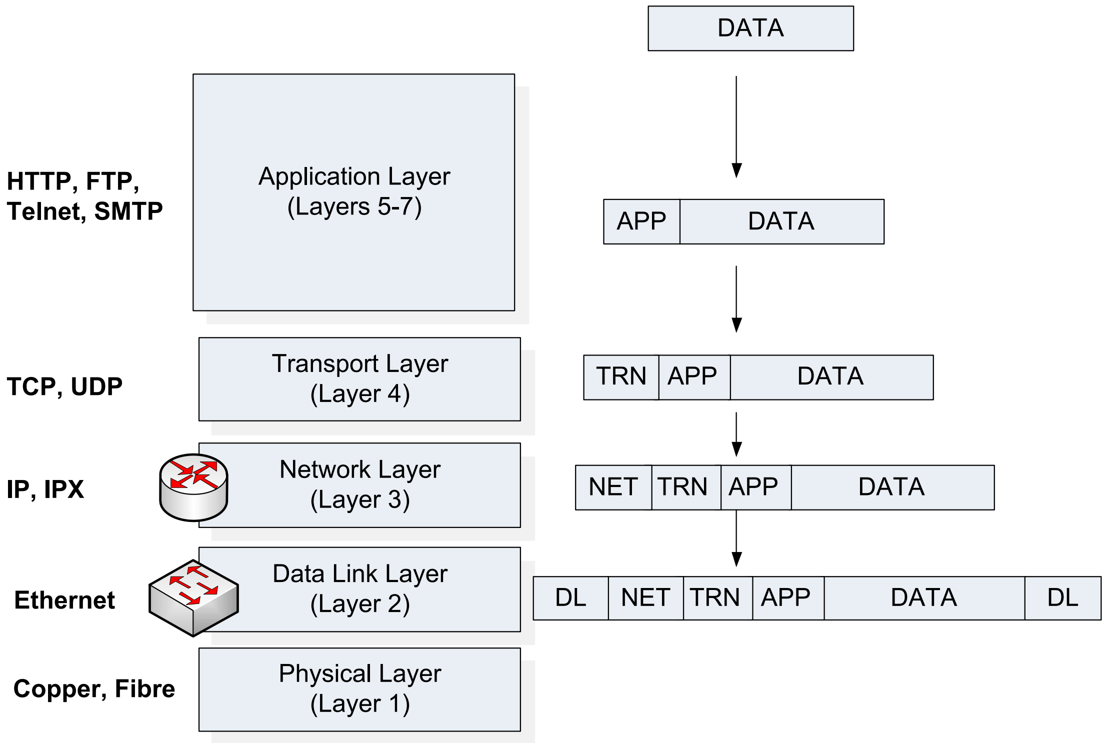


Figure 9.1 Data encapsulation

The most important Layer 2 data frame technology is Ethernet, at Layer 3 it is IP (Internet Protocol) and at Layer 4 it is TCP (Transport Control Protocol). An advantage of using Ethernet is that it will encapsulate a number of Layer 3 protocols. Figure 9.2 shows that the Type field is used to define the format of the data to be contained within its Data field. In this case, 0x800 identifies it will be an IP packet, while 0x806 defines an ARP packet. Then within an IP data packet, there is a Protocol field which will define the Layer 4 protocol. A value of 6 defines TCP, and a value of 17 defines UDP.

For a typical encapsulation of Ethernet, IP and TCP, the key parameters are:

- **Ethernet**. The Src MAC and Dest MAC addresses are 48-bit addresses which define the hardware address of the data frame.
- **IP**. The Src IP and Dest IP addresses defines the 32-bit IP (logical) addresses for the sender and the receiver of the data packet. The TTL field is used to stop the data packet from transversing the Internet infinitely. For this each intermediate routing device decrements this field by a given amount. Once it gets to zero, it will be deleted by the device which receives it. The Version field defines the IP Version, where most data packets use Version 4, while Version 6 is used to extend the address range.

- **TCP**. The reliability of the transmission is normally defined within the TCP operation. The key fields are the TCP Src and TCP Dest ports which define the source and destination TCP ports used in the communication. The Sequence Number and Acknowledge Number defines the sequence numbers for the data segments, and are used to provide acknowledgements for data transmitted and received. The Flags field are used to identify the state of the connection, and the Window field defines the number of data segments that can be received before an acknowledgement is required.

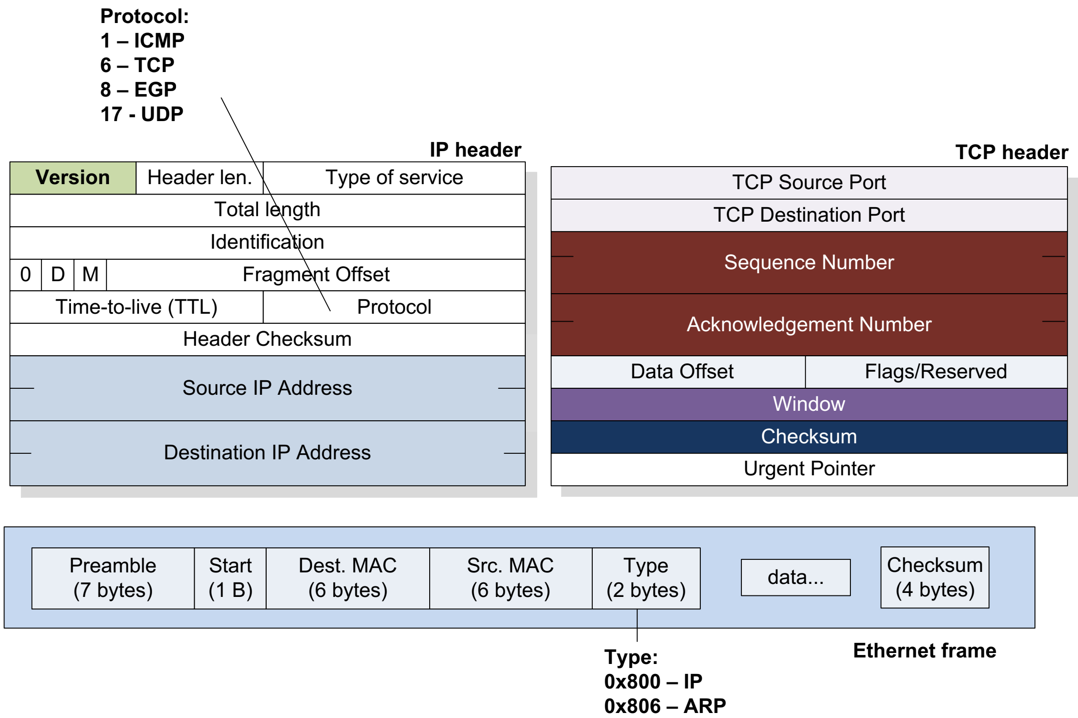


Figure 9.2 Ethernet, IP and TCP

9.5 TCP connection

The key to reliable communications over the Internet is the TCP protocol. At the core of this is the TCP flags, and the three way handshake. Figure 9.3 illustrates this procedure using a practical example. Initially three TCP data segments are exchanged, the first goes from the host (the client) to the server with the S (SYN) flag sent. The client also identifies the TCP port it wishes to use, and connects to the TCP port that the server is listening on. Next the server sends back a TCP segment with the S (SYN) and A (ACK) flags set, which identifies that it wishes to accept the connection. Finally the client sends back a TCP segment with the A (ACK) flag set. Once these TCP segments have been exchanged, there is a unique mapping of:

IP[Host]:TCPport[Host] -> IP[Server]:TCPport[Server]

As part of the three way handshake the host and server also negotiate the Window to be used, as illustrated in Figure 9.3. In this case the final value which they settle on is

66,608. This defines the number of data segments that can be sent before the sender waits for an acknowledgement for the data previously transmitted.

A key element of TCP is that it is reliable, where each data segment has a sequence number, and data is then acknowledged for successful transmission. Figure 9.3 shows an example, where the Ack number defines the data segment that the host expects to receive next. For example in the 7th data segment, the host defines that it expects to see Seq No 70, and the 8th data segment has a Seq No of 70.

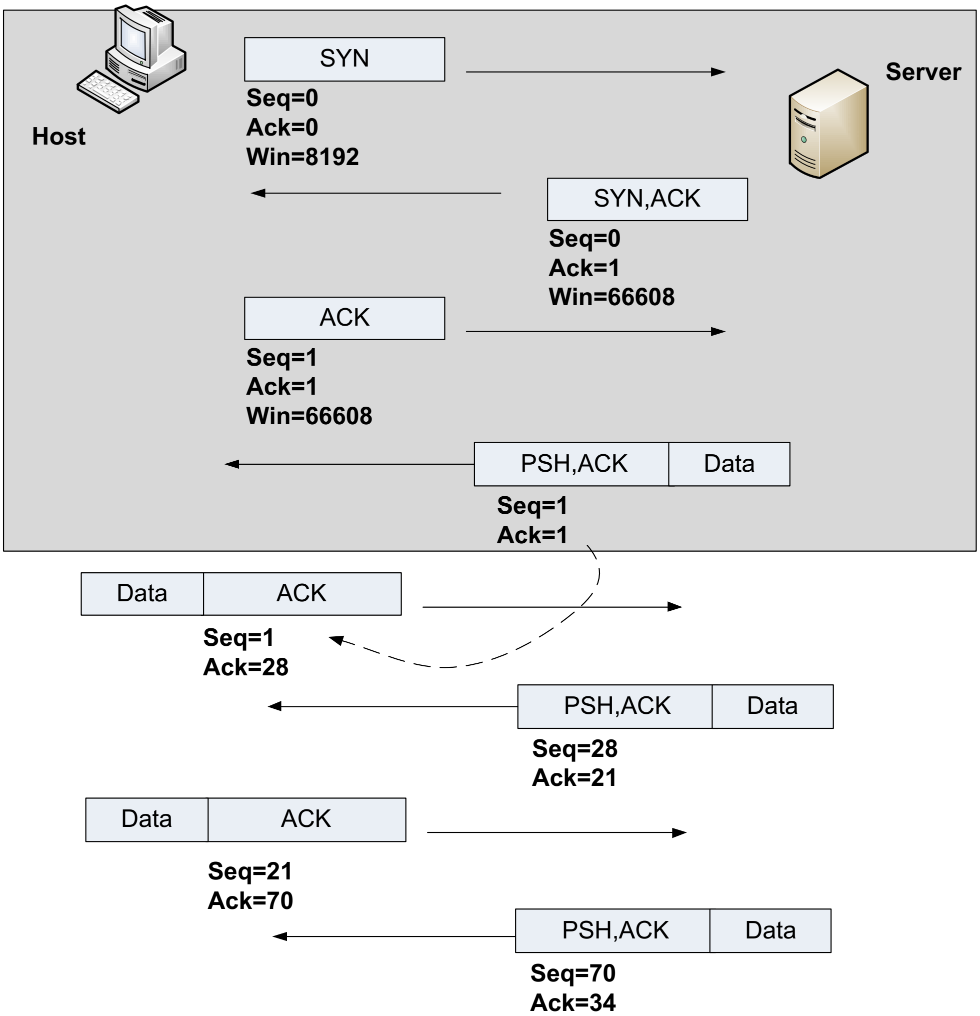


Figure 9.3 TCP flags

9.6 ARP

ARP is used to resolve an IP address to a MAC address, and is used for the first part of the communication path, and also the last part. Often ARP activity is one of the first traces of activity within any type of network connection. If a node communicates with a node within the same subnet, it can discover the MAC address for the node with an ARP broadcast. Figure 9.4 shows an example where Bob (at 192.168.75.132) needs to connect to the Internet, and thus requires the MAC address of the gateway (192.168.75.1). Thus Bob sends out an ARP request:

```
No.     Time          Source                Destination           Protocol Info
```

```
      1 0.000000    Vmware_c0:00:08       Broadcast             ARP      Who has
192.168.75.132?  Tell 192.168.75.1

Frame 1 (42 bytes on wire, 42 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Broadcast
(ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
```

http://buchananweb.co.uk/log/ftp.txt [Packet 1 and 2]

For which the gateway replies with its MAC address (00:0c:29:71:a3):

```
No.     Time        Source                Destination           Protocol Info
      2 0.021830    Vmware_0f:71:a3       Vmware_c0:00:08       ARP
192.168.75.132 is at 00:0c:29:0f:71:a3

Frame 2 (42 bytes on wire, 42 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
Address Resolution Protocol (reply)
```

On Bob's computer the ARP cache is then updated, such as:

```
C:\> arp -a

Interface: 192.168.75.1 --- 0x1d
  Internet Address      Physical Address      Type
  192.168.75.132        00-0c-29-0f-71-a3     dynamic
  192.168.75.138        00-0c-29-6b-0e-96     dynamic
  192.168.75.255        ff-ff-ff-ff-ff-ff     static
```

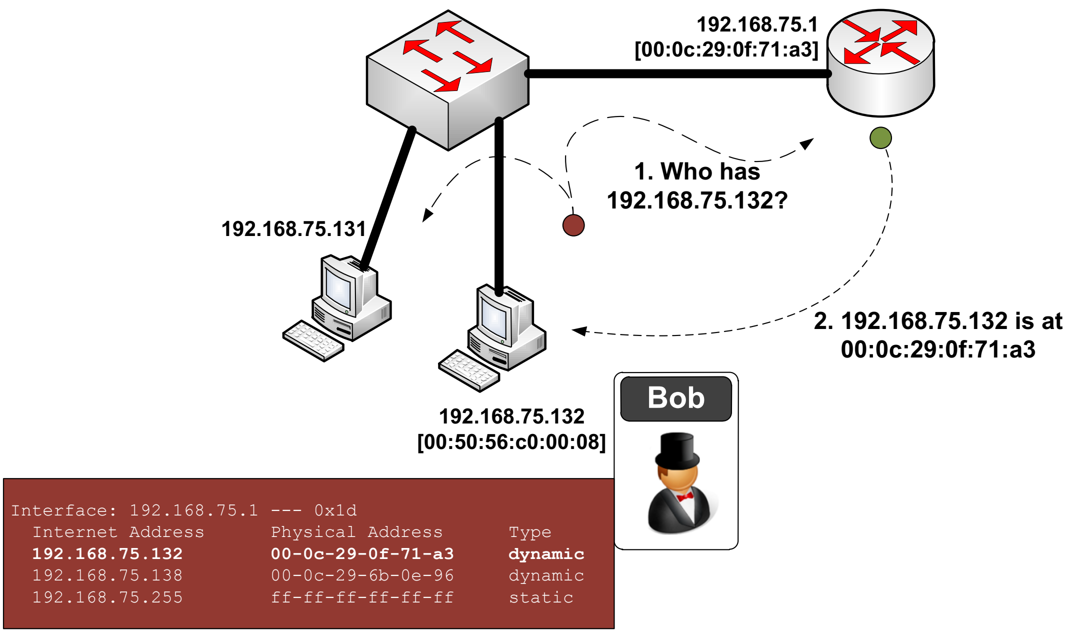


Figure 9.4 ARP activity

Most Windows computers have an ARP timeout of 10 minutes, where Cisco routers timeout after 4 hours.

9.7 SYN

In network forensics, the SYN flag is key to finding the starting point of a connection, as every TCP connection requires a three-way handshake. In the following example the connection details of the connection are:

192.168.75.1:3655 -> 192.168.75.132:21

Where the host is at 192.168.75.1, and the FTP server is at 192.168.75.132.

```
No.     Time        Source                Destination           Protocol Info
      3 0.021867    192.168.75.1          192.168.75.132        TCP      abatemgr >
ftp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2 TSV=683746 TSER=0

Frame 3 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: abatemgr (3655), Dst Port: ftp (21), Seq: 0,
Len: 0

No.     Time        Source                Destination           Protocol Info
      4 0.022961    192.168.75.132        192.168.75.1          TCP      ftp >
abatemgr [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=0 TSV=0 TSER=0

Frame 4 (78 bytes on wire, 78 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: ftp (21), Dst Port: abatemgr (3655), Seq: 0,
Ack: 1, Len: 0

No.     Time        Source                Destination           Protocol Info
      5 0.023078    192.168.75.1          192.168.75.132        TCP      abatemgr >
ftp [ACK] Seq=1 Ack=1 Win=66608 Len=0 TSV=683748 TSER=0

Frame 5 (66 bytes on wire, 66 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: abatemgr (3655), Dst Port: ftp (21), Seq: 1,
Ack: 1, Len: 0
```

View this file: http://buchananweb.co.uk/log/ftp.txt [Packets 3-5]

9.8 Application Layer Analysis - FTP

The FTP application protocol uses commands (USER, PASS, MKD, CWD, QUIT, RMD, and so on), where there is a numeric response value (such as 226 – Transfer complete and 250 – CWD command successful). The following shows the requests and replies passed from a client to a server:

Server	Client
`220 Microsoft FTP Service`	
	USER Administrator
`331 Password required for Administrator.`	
	PASS napier
`230 User Administrator logged in.`	
	SYST
`215 Windows_NT`	
	PWD
`257 "/" is current directory.`	
	PASV

```
227 Entering Passive Mode (192,168,75,132,4,22).
                                                                LIST
125 Data connection already open; Transfer starting.
226 Transfer complete.
                                                               CWD /
250 CWD command successful.
                                                                PASV
227 Entering Passive Mode (192,168,75,132,4,23).
                                                                LIST
125 Data connection already open; Transfer starting.
226 Transfer complete.
                                                                 PWD
257 "/" is current directory.
                                                              TYPE A
200 Type set to A.
                                                                PASV
227 Entering Passive Mode (192,168,75,132,4,24).
                                                        STOR db1.csv
125 Data connection already open; Transfer starting.
226 Transfer complete.
```

This example uses Passive FTP, which creates a server port which the client must connect to. This is determined from:

```
227 Entering Passive Mode (192,168,75,132,4,24).
```

Where the last two digital determine the port that will be created. This is calculated from 256 times the second last digit, plus the last digit. Thus the port created is 1048 (4x256+24). The client will then create a connection on this port, and transfer the information.

The following shows the initial data packets exchanged for the connection defined in the previous two transfers (Sections 9.6 and 9.7):

```
No.     Time        Source                Destination           Protocol Info
      6 0.026461    192.168.75.132        192.168.75.1          FTP      Response: 220 Mi-
crosoft FTP Service

Frame 6 (93 bytes on wire, 93 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1 (192.168.75.1)
Transmission Control Protocol, Src Port: ftp (21), Dst Port: abatemgr (3655), Seq: 1, Ack: 1,
Len: 27
File Transfer Protocol (FTP)

No.     Time        Source                Destination           Protocol Info
      7 0.107380    192.168.75.1          192.168.75.132        FTP      Request: USER Admin-
istrator

Frame 7 (86 bytes on wire, 86 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132 (192.168.75.132)
Transmission Control Protocol, Src Port: abatemgr (3655), Dst Port: ftp (21), Seq: 1, Ack:
28, Len: 20
File Transfer Protocol (FTP)

No.     Time        Source                Destination           Protocol Info
      8 0.108092    192.168.75.132        192.168.75.1          FTP      Response: 331 Pass-
word required for Administrator.

Frame 8 (108 bytes on wire, 108 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1 (192.168.75.1)
Transmission Control Protocol, Src Port: ftp (21), Dst Port: abatemgr (3655), Seq: 28, Ack:
21, Len: 42
```

```
File Transfer Protocol (FTP)

No.     Time        Source                Destination           Protocol Info
    9 0.108387    192.168.75.1          192.168.75.132        FTP      Request: PASS napier

Frame 9 (79 bytes on wire, 79 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132 (192.168.75.132)
Transmission Control Protocol, Src Port: abatemgr (3655), Dst Port: ftp (21), Seq: 21, Ack:
70, Len: 13
File Transfer Protocol (FTP)

No.     Time        Source                Destination           Protocol Info
     10 0.110448    192.168.75.132        192.168.75.1          FTP      Response: 230 User
Administrator logged in.

Frame 10 (101 bytes on wire, 101 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1 (192.168.75.1)
Transmission Control Protocol, Src Port: ftp (21), Dst Port: abatemgr (3655), Seq: 70, Ack:
34, Len: 35
File Transfer Protocol (FTP)
```

View this file: http://buchananweb.co.uk/log/ftp.txt [Packets 6 and on]

9.9 ICMP

ICMP is a protocol used to provide debug information, such as to determine if a host is operating (using ping) or to trace the route to a destination (using tracert). Unfortunately it can also be used by malicious sources to determine if a device is on-line (and which ones), and the route that data packets take. The following shows a ping request from 192.168.75.1 to 192.168.75.132:

```
No.     Time        Source                Destination           Protocol Info
     10 13.706916   192.168.75.1          192.168.75.132        ICMP     Echo (ping)
request
Frame 10 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Internet Control Message Protocol

No.     Time        Source                Destination           Protocol Info
     11 13.707279   192.168.75.132        192.168.75.1          ICMP     Echo (ping)
reply
Frame 11 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1
(192.168.75.1)
Internet Control Message Protocol
```

View this file: http://buchananweb.co.uk/log/ping.txt

9.10 DNS

DNS lookup is often a key pointer to the start to some form of initial activity. The protocol operates, normally, using UDP on Port 53. In the following example, the host (192.168.0.20) contacts the DNS server at 192.168.0.1. Packet 7 shows that the lookup is for www.intel.com, which, in Packet 8, returns the lookup for www.intel.com, such as:

```
F:\docs\src\clientToolkit\log>nslookup www.intel.com
Server:  UnKnown
Address:  192.168.0.1

Non-authoritative answer:
Name:    a961.g.akamai.net
Addresses:  81.52.140.11
          81.52.140.83
Aliases:  www.intel.com
          www.intel.com.edgesuite.net
          www.intel-sino.com.edgesuite.net
          www.intel-sino.com.edgesuite.net.chinaredirector.akadns.net
```

In this case the UDP details are:

192.168.0.20:63227 -> 192.168.0.1:53

```
No.     Time        Source                Destination           Protocol Info
      7 5.386386    192.168.0.20          192.168.0.1           DNS      Standard que-
ry A www.intel.com

Frame 7 (73 bytes on wire, 73 bytes captured)
Ethernet II, Src: IntelCor_4f:30:1d (00:1f:3c:4f:30:1d), Dst: Netgear_b0:d6:8c
(00:18:4d:b0:d6:8c)
    Destination: Netgear_b0:d6:8c (00:18:4d:b0:d6:8c)
    Source: IntelCor_4f:30:1d (00:1f:3c:4f:30:1d)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.20 (192.168.0.20), Dst: 192.168.0.1 (192.168.0.1)
User Datagram Protocol, Src Port: 63227 (63227), Dst Port: domain (53)
Domain Name System (query)

No.     Time        Source                Destination           Protocol Info
      8 5.461009    192.168.0.1           192.168.0.20          DNS      Standard que-
ry response CNAME www.intel.com.edgesuite.net CNAME www.intel-sino.com.edgesuite.net
CNAME www.intel-sino.com.edgesuite.net.chinaredirector.akadns.net CNAME
a961.g.akamai.net A 92.122.126.176 A 92.122.126.146

Frame 8 (547 bytes on wire, 547 bytes captured)
Ethernet II, Src: Netgear_b0:d6:8c (00:18:4d:b0:d6:8c), Dst: IntelCor_4f:30:1d
(00:1f:3c:4f:30:1d)
    Destination: IntelCor_4f:30:1d (00:1f:3c:4f:30:1d)
    Source: Netgear_b0:d6:8c (00:18:4d:b0:d6:8c)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.20 (192.168.0.20)
User Datagram Protocol, Src Port: domain (53), Dst Port: 63227 (63227)
Domain Name System (response)
```

View this file: http://buchananweb.co.uk/log/dnslookup.txt

9.11 Port scan

A port scan is often seen as a sign of malicious activity, where an intruder tries to find the ports which are open on a computer. The following shows an NMAP scan from 192.168.75.1 to 192.168.75.132, where it sends SYNs for key ports, such as Telnet (23), RAP (256), IMAPS (993), POP3 (110), and so on. If a connection is made on the port, there will be a response, otherwise NMAP continues to scan the ports. A continual accessing of a range of a ports over a time interval, often shows intruder activity.

```
No.     Time        Source                Destination           Protocol Info
     85 25.420710   192.168.75.1          192.168.75.132        TCP      54370 >
telnet [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 85 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: 54370 (54370), Dst Port: telnet (23), Seq: 0,
Len: 0

No.     Time        Source                Destination           Protocol Info
     86 25.420836   192.168.75.1          192.168.75.132        TCP      54370 > rap
[SYN] Seq=0 Win=2048 Len=0 MSS=1460

Frame 86 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: 54370 (54370), Dst Port: rap (256), Seq: 0,
Len: 0

No.     Time        Source                Destination           Protocol Info
     87 25.420897   192.168.75.1          192.168.75.132        TCP      54370 > imaps
[SYN] Seq=0 Win=3072 Len=0 MSS=1460

Frame 87 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: 54370 (54370), Dst Port: imaps (993), Seq: 0,
Len: 0

No.     Time        Source                Destination           Protocol Info
     88 25.420941   192.168.75.1          192.168.75.132        TCP      54370 > pop3s
[SYN] Seq=0 Win=2048 Len=0 MSS=1460

Frame 88 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: 54370 (54370), Dst Port: pop3s (995), Seq: 0,
Len: 0

No.     Time        Source                Destination           Protocol Info
     89 25.420984   192.168.75.1          192.168.75.132        TCP      54370 > mi-
crosoft-ds [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 89 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
```

```
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: 54370 (54370), Dst Port: microsoft-ds (445),
Seq: 0, Len: 0

No.     Time        Source                Destination           Protocol Info
     90 25.421026   192.168.75.1          192.168.75.132        TCP      54370 > smux
[SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 90 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: 54370 (54370), Dst Port: smux (199), Seq: 0,
Len: 0

No.     Time        Source                Destination           Protocol Info
     91 25.421069   192.168.75.1          192.168.75.132        TCP      54370 > pptp
[SYN] Seq=0 Win=2048 Len=0 MSS=1460
```

View this file: http://buchananweb.co.uk/log/webpage.txt [Packet 85 on]

9.12 SYN flood

Distributed Denial-of-Service (DDoS) is one of the most difficult attacks to defend against, as it is often difficult to differentiate malicious connections from non-malicious ones. The following shows an example of a host (192.168.75.137) connecting to port 80 on 192.168.75.1, and results in the connections of:

192.168.75.137:1608 -> 192.168.71.1:80
192.168.75.137:1609 -> 192.168.71.1:80

```
No.     Time        Source                Destination           Protocol Info
      2 4.510329    192.168.75.137        192.168.75.1          HTTP     Continuation
or non-HTTP traffic

Frame 2 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_6b:0e:96 (00:0c:29:6b:0e:96), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_6b:0e:96 (00:0c:29:6b:0e:96)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.137 (192.168.75.137), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: smart-lm (1608), Dst Port: http (80), Seq: 0,
Len: 4
Hypertext Transfer Protocol

No.     Time        Source                Destination           Protocol Info
      3 5.514164    192.168.75.137        192.168.75.1          HTTP     Continuation
or non-HTTP traffic

Frame 3 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_6b:0e:96 (00:0c:29:6b:0e:96), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_6b:0e:96 (00:0c:29:6b:0e:96)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.137 (192.168.75.137), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: isysg-lm (1609), Dst Port: http (80), Seq: 0,
Len: 4
Hypertext Transfer Protocol

No.     Time        Source                Destination           Protocol Info
```

```
      4 6.517235    192.168.75.137        192.168.75.1          HTTP     Continuation
or non-HTTP traffic

Frame 4 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_6b:0e:96 (00:0c:29:6b:0e:96), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_6b:0e:96 (00:0c:29:6b:0e:96)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.137 (192.168.75.137), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: taurus-wh (1610), Dst Port: http (80), Seq:
0, Len: 4
Hypertext Transfer Protocol

No.     Time        Source                Destination           Protocol Info
      5 7.520267    192.168.75.137        192.168.75.1          HTTP     Continuation
or non-HTTP traffic

Frame 5 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_6b:0e:96 (00:0c:29:6b:0e:96), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_6b:0e:96 (00:0c:29:6b:0e:96)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.137 (192.168.75.137), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: ill (1611), Dst Port: http (80), Seq: 0, Len:
4
Hypertext Transfer Protocol
```

View this file: http://buchananweb.co.uk/log/hping_port80.txt [Packet 2 on]

A FIN flood is shown in http://buchananweb.co.uk/log/hping_fin.txt

9.13 Spoofed addresses

One method that an intruder can use to hide their tracks is to substitute their IP address with another address. In the following example the intruder has used NMAP with a spoofed address of 10.0.0.1:

```
nmap -e eth0 192.168.75.132 -S 10.0.0.1 –sS
```

to give a result of:

```
No.     Time        Source                Destination           Protocol Info
      5 0.044549    10.0.0.1              192.168.75.132        TCP      40484 > https
[SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 5 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_6b:0e:96 (00:0c:29:6b:0e:96), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
Internet Protocol, Src: 10.0.0.1 (10.0.0.1), Dst: 192.168.75.132 (192.168.75.132)
Transmission Control Protocol, Src Port: 40484 (40484), Dst Port: https (443), Seq: 0,
Len: 0

No.     Time        Source                Destination           Protocol Info
      6 0.044857    10.0.0.1              192.168.75.132        TCP      40484 > pptp
[SYN] Seq=0 Win=2048 Len=0 MSS=1460

Frame 6 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: Vmware_6b:0e:96 (00:0c:29:6b:0e:96), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
Internet Protocol, Src: 10.0.0.1 (10.0.0.1), Dst: 192.168.75.132 (192.168.75.132)
Transmission Control Protocol, Src Port: 40484 (40484), Dst Port: pptp (1723), Seq: 0,
Len: 0

No.     Time        Source                Destination           Protocol Info
```

```
     7 0.044871    192.168.75.132        10.0.0.1              TCP      https > 40484
[RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Frame 7 (54 bytes on wire, 54 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_f5:2e:f3
(00:50:56:f5:2e:f3)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 10.0.0.1 (10.0.0.1)
Transmission Control Protocol, Src Port: https (443), Dst Port: 40484 (40484), Seq: 1,
Ack: 1, Len: 0

No.     Time        Source                Destination           Protocol Info
     8 0.045043    192.168.75.132        10.0.0.1              TCP      pptp > 40484
[RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

View this file: http://buchananweb.co.uk/log/spoof_address.txt [Packet 5 on]

Private addresses within a public address space normally shows maliciousness. These addresses are in the following ranges:

10.0.0.0 - 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255

9.14 Application Layer Analysis - HTTP

The foundation protocol of the WWW is the Hypertext Transfer Protocol (HTTP) which can be used in any client/server application involving hypertext. It is used on the WWW for transmitting information using hypertext jumps and can support the transfer of plaintext, hypertext, audio, images, or any Internet-compatible information. The most recently defined standard is HTTP 1.1, which has been defined by the IETF standard.

HTTP is a stateless protocol where each transaction is independent of any previous transactions. Thus when the transaction is finished the TCP/IP connection is disconnected, as illustrated in Figure 9.5. The advantage of being stateless is that it allows the rapid access of WWW pages over several widely distributed servers. It uses the TCP protocol to establish a connection between a client and a server for each transaction then terminates the connection once the transaction completes.

HTTP also supports many different formats of data. Initially a client issues a request to a server which may include a prioritized list of formats that it can handle. This allows new formats to be easily added and also prevents the transmission of unnecessary information.

A client's WWW browser (the user agent) initially establishes a direct connection with the destination server which contains the required WWW page. To make this connection the client initiates a TCP connection between the client and the server. After this is established the client then issues an HTTP request, such as the specific command (the method), the URL, and possibly extra information such as request parameters or client information. When the server receives the request, it attempts to perform the requested action. It then returns an HTTP response, which includes status information, a success/error code, and extra information. After the client receives this, the TCP connection is closed.

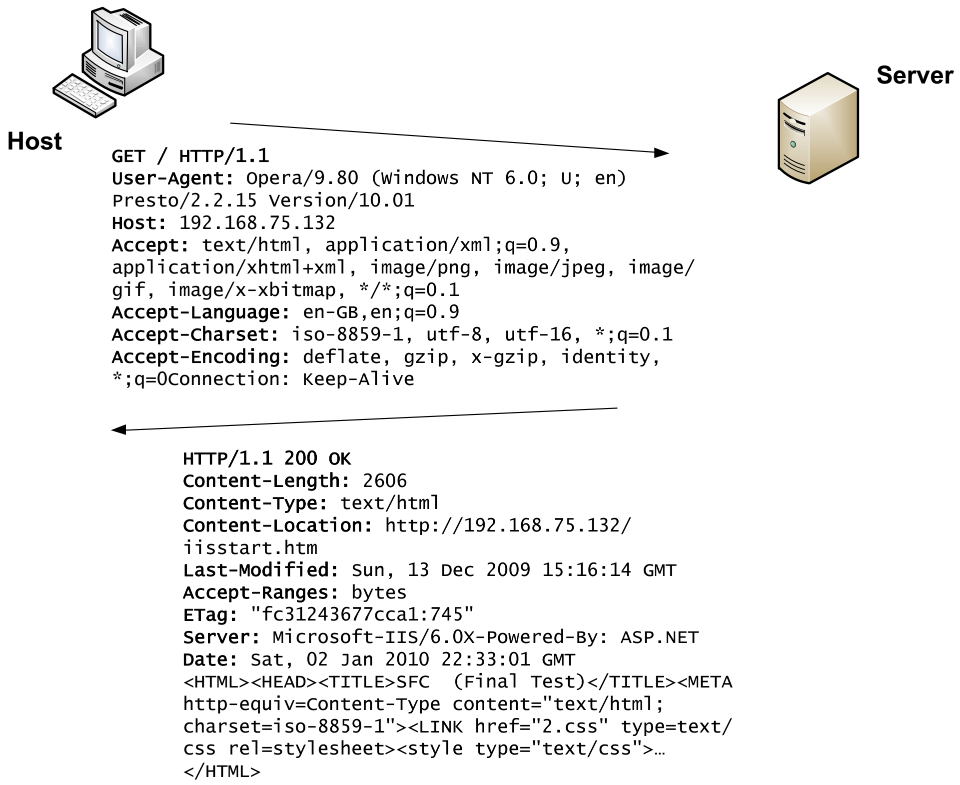


Figure 9.5 Example HTTP transaction

HTTP messages

The simple request is a `GET` command with the requested URI such as:

```
GET  /info/dept/courses.html
```

The simple response is a block containing the information identified in the URI (called the entity-body).

Full requests/responses

Very few security measures or enhanced services are built into the simple requests/responses. HTTP Version 1.0/1.1 improves on the simple requests/responses by adding many extra requests and responses, as well as adding extra information about the data supported. Each message header consists of a number of fields which begin on a new line and consist of the field name followed by a colon and the field value. A full request starts with a request line command (such as `GET`, `MOVE` or `DELETE`) and is then followed by one or more of the following:

- General-headers which contain general fields that do not apply to the entity being transferred (such as MIME version, date, and so on).
- Request-headers which contain information on the request and the client (e.g. the client's name, its authorization, and so on).
- Entity-headers which contain information about the resource identified by the request and entity-body information (such as the type of encoding, the language, the title, the time when it was last modified, the type of resource it is, when it expires, and so on).

- Entity-body which contains the body of the message (such as HTML text, an image, a sound file, and so on).

A full response starts with a response status code (such as `OK`, `Moved Temporarily`, `Accepted`, `Created`, `Bad Request`, and so on) and is then followed by one or more of the following:

- General-headers, as with requests, contain general fields which do not apply to the entity being transferred (MIME version, date, and so on).
- Response-headers which contain information on the response and the server (e.g. the server's name, its location and the time the client should retry the server).
- Entity-headers, as with request, which contain information about the resource identified by the request and entity-body information (such as the type of encoding, the language, the title, the time when it was last modified, the type of resource it is, when it expires, and so on).
- Entity-body, as with requests, which contains the body of the message (such as HTML text, an image, a sound file, and so on).

The following example shows an example request. The first line is always the request method; in this case it is `GET`. Next there are various headers. The general-header field is `Content-Type`, the request-header fields are `If-Modified-Since` and `From`. There are no entity parts to the message as the request is to get an image (if the command had been to `PUT` then there would have been an attachment with the request). Notice that a single blank line delimits the end of the message as this indicates the end of a request/response. Note that the headers are case sensitive, thus `Content-Type` with the correct types of letters (and `GET` is always in uppercase letters).

An example is:

```
No.     Time        Source                Destination           Protocol Info
      3 0.000362    192.168.75.1          192.168.75.132        TCP      mgcp-gateway
> http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2 TSV=344415 TSER=0

Frame 3 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: mgcp-gateway (2427), Dst Port: http (80),
Seq: 0, Len: 0

No.     Time        Source                Destination           Protocol Info
      4 0.000602    192.168.75.132        192.168.75.1          TCP      http > mgcp-
gateway [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=0 TSV=0 TSER=0

Frame 4 (78 bytes on wire, 78 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: mgcp-gateway (2427),
Seq: 0, Ack: 1, Len: 0
```

```
No.     Time        Source                Destination           Protocol Info
      5 0.000681    192.168.75.1          192.168.75.132        TCP      mgcp-gateway
> http [ACK] Seq=1 Ack=1 Win=66608 Len=0 TSV=344415 TSER=0

Frame 5 (66 bytes on wire, 66 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: mgcp-gateway (2427), Dst Port: http (80),
Seq: 1, Ack: 1, Len: 0

No.     Time        Source                Destination           Protocol Info
      6 0.000835    192.168.75.1          192.168.75.132        HTTP     GET /
HTTP/1.1

Frame 6 (475 bytes on wire, 475 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: mgcp-gateway (2427), Dst Port: http (80),
Seq: 1, Ack: 1, Len: 409
Hypertext Transfer Protocol

No.     Time        Source                Destination           Protocol Info
      7 0.055477    192.168.75.132        192.168.75.1          TCP      [TCP segment
of a reassembled PDU]

Frame 7 (1514 bytes on wire, 1514 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: mgcp-gateway (2427),
Seq: 1, Ack: 410, Len: 1448

No.     Time        Source                Destination           Protocol Info
      8 0.055715    192.168.75.132        192.168.75.1          TCP      [TCP segment
of a reassembled PDU]

Frame 8 (1514 bytes on wire, 1514 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: mgcp-gateway (2427),
Seq: 1449, Ack: 410, Len: 1448

No.     Time        Source                Destination           Protocol Info
      9 0.055759    192.168.75.1          192.168.75.132        TCP      mgcp-gateway
> http [ACK] Seq=410 Ack=2897 Win=66608 Len=0 TSV=344421 TSER=15586

Frame 9 (66 bytes on wire, 66 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: mgcp-gateway (2427), Dst Port: http (80),
Seq: 410, Ack: 2897, Len: 0
```

```
No.     Time        Source                Destination           Protocol Info
     10 0.056010    192.168.75.132        192.168.75.1          HTTP     HTTP/1.1 200
OK  (text/html)

Frame 10 (79 bytes on wire, 79 bytes captured)
Ethernet II, Src: Vmware_0f:71:a3 (00:0c:29:0f:71:a3), Dst: Vmware_c0:00:08
(00:50:56:c0:00:08)
    Destination: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Source: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.132 (192.168.75.132), Dst: 192.168.75.1
(192.168.75.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: mgcp-gateway (2427),
Seq: 2897, Ack: 410, Len: 13
[Reassembled TCP Segments (2909 bytes): #7(1448), #8(1448), #10(13)]
Hypertext Transfer Protocol
Line-based text data: text/html

No.     Time        Source                Destination           Protocol Info
     11 0.090363    192.168.75.1          192.168.75.132        HTTP     GET /2.css
HTTP/1.1

Frame 11 (565 bytes on wire, 565 bytes captured)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_0f:71:a3
(00:0c:29:0f:71:a3)
    Destination: Vmware_0f:71:a3 (00:0c:29:0f:71:a3)
    Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.75.1 (192.168.75.1), Dst: 192.168.75.132
(192.168.75.132)
Transmission Control Protocol, Src Port: mgcp-gateway (2427), Dst Port: http (80),
Seq: 410, Ack: 2910, Len: 499
Hypertext Transfer Protocol
```

View this file: http://buchananweb.co.uk/log/webpage.txt

The request/response sequence is then:

Client **Server**

```
GET / HTTP/1.1
User-Agent: Opera/9.80 (Windows NT 6.0; U; en) Presto/2.2.15 Version/10.01
Host: 192.168.75.132
Accept: text/html, application/xml;q=0.9, application/xhtml+xml, image/png, im-
age/jpeg, image/gif, image/x-xbitmap, */*;q=0.1
Accept-Language: en-GB,en;q=0.9
Accept-Charset: iso-8859-1, utf-8, utf-16, *;q=0.1
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0
Connection: Keep-Alive
                                                                  HTTP/1.1 200 OK
                                                             Content-Length: 2606
                                                          Content-Type: text/html
                                Content-Location: http://192.168.75.132/iisstart.htm
                                        Last-Modified: Sun, 13 Dec 2009 15:16:14 GMT
                                                             Accept-Ranges: bytes
                                                       ETag: "fc31243677cca1:745"
                                                         Server: Microsoft-IIS/6.0
                                                             X-Powered-By: ASP.NET
                                                Date: Sat, 02 Jan 2010 22:33:01 GMT
                                                                            <HTML>
                                                                            <HEAD>
                                                   <TITLE>SFC  (Final Test)</TITLE>
                 <META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
                                       <LINK href="2.css" type=text/css rel=stylesheet>
                                                           <style type="text/css">
                                                                                 …
                                                                           </HTML>
GET /2.css HTTP/1.1
User-Agent: Opera/9.80 (Windows NT 6.0; U; en) Presto/2.2.15 Version/10.01
Host: 192.168.75.132
Accept: text/html, application/xml;q=0.9, application/xhtml+xml, image/png, im-
age/jpeg, image/gif, image/x-xbitmap, */*;q=0.1
Accept-Language: en-GB,en;q=0.9
Accept-Charset: iso-8859-1, utf-8, utf-16, *;q=0.1
```

```
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0
Referer: http://192.168.75.132/
Connection: Keep-Alive, TE
TE: deflate, gzip, chunked, identity, trailers
                                                            HTTP/1.1 200 OK

                                                    Content-Length: 14135
                                                   Content-Type: text/css
                         Last-Modified: Sun, 13 Dec 2009 15:15:09 GMT
                                                    Accept-Ranges: bytes
                                                 ETag: "744c36f77cca1:745"
                                                 Server: Microsoft-IIS/6.0
                                                     X-Powered-By: ASP.NET
                                      Date: Sat, 02 Jan 2010 22:33:01 GMT
                                                                     ...H1
                  {font: bold 16pt Verdana, Arial, Helvetica, sans-serif;
                                                 background: transparent;
```

It can be seen that the data format that the client and the server can accept are identified in the header sent.

9.15 Network logs on hosts

Captured network packets are useful for analyzing systems in real-time, but often malicious activity can take place over long intervals. It is thus difficult to analyze a trail of evidence of network packets over a relatively long period of time. Most systems, though, have audit logs which can provide evidence of activities. In Windows the Web server stores its log at:

C:\WINDOWS\system32\LogFiles

For the instance of Web instance of W3SVC1, a sample log is:

```
#Software: Microsoft Internet Information Services 6.0
#Version: 1.0
#Date: 2010-01-02 22:29:25
#Fields: date time s-sitename s-ip cs-method cs-uri-stem cs-uri-query s-port cs-
username c-ip cs(User-Agent) sc-status sc-substatus sc-win32-status
2010-01-02 22:29:25 W3SVC1 192.168.75.132 GET /iisstart.htm - 80 - 192.168.75.1 Mozil-
la/5.0+(Windows;+U;+Windows+NT+6.0;+en-
US;+rv:1.8.1.20)+Gecko/20081217+Firefox/2.0.0.20 200 0 0
2010-01-02 22:29:25 W3SVC1 192.168.75.132 GET /2.css - 80 - 192.168.75.1 Mozil-
la/5.0+(Windows;+U;+Windows+NT+6.0;+en-
US;+rv:1.8.1.20)+Gecko/20081217+Firefox/2.0.0.20 200 0 0
2010-01-02 22:29:25 W3SVC1 192.168.75.132 GET /favicon.ico - 80 - 192.168.75.1 Mozil-
la/5.0+(Windows;+U;+Windows+NT+6.0;+en-
US;+rv:1.8.1.20)+Gecko/20081217+Firefox/2.0.0.20 404 0 2
2010-01-02 22:29:35 W3SVC1 192.168.75.132 GET /iisstart.htm - 80 - 192.168.75.1 Mozil-
la/5.0+(Windows;+U;+Windows+NT+6.0;+en-
US;+rv:1.8.1.20)+Gecko/20081217+Firefox/2.0.0.20 304 0 0
2010-01-02 22:29:35 W3SVC1 192.168.75.132 GET /2.css - 80 - 192.168.75.1 Mozil-
la/5.0+(Windows;+U;+Windows+NT+6.0;+en-
US;+rv:1.8.1.20)+Gecko/20081217+Firefox/2.0.0.20 304 0 0
2010-01-02 22:33:01 W3SVC1 192.168.75.132 GET /iisstart.htm - 80 - 192.168.75.1
Opera/9.80+(Windows+NT+6.0;+U;+en)+Presto/2.2.15+Version/10.01 200 0 0
2010-01-02 22:33:01 W3SVC1 192.168.75.132 GET /2.css - 80 - 192.168.75.1
Opera/9.80+(Windows+NT+6.0;+U;+en)+Presto/2.2.15+Version/10.01 200 0 0
2010-01-02 22:33:01 W3SVC1 192.168.75.132 GET /favicon.ico - 80 - 192.168.75.1
Opera/9.80+(Windows+NT+6.0;+U;+en)+Presto/2.2.15+Version/10.01 404 0 2
```

Where it can be seen that the host 192.168.75.132 has been accessing Web pages on the server (192.168.75.1). It can also be seen that the accesses have been from Firefox and Presto (Opera). For FTP, we can access the instance of an FTP server (\MSFTPSVC1) gives:

```
#Software: Microsoft Internet Information Services 6.0
#Version: 1.0
#Date: 2010-01-04 19:36:08
#Fields: time c-ip cs-method cs-uri-stem sc-status sc-win32-status
19:36:08 192.168.75.138 [2]closed - 426 10054
20:18:05 192.168.75.1 [3]USER Administrator 331 0
20:18:05 192.168.75.1 [3]PASS - 230 0
20:18:44 192.168.75.1 [3]created index_asfc.html 550 5
20:19:21 192.168.75.1 [3]QUIT - 226 0
```

Where the client IP address and the requests are defined in the log file. Error logs are also an important place to look for maliciousness.

In Linux, the /var/log folder contains a host of log files. For example, the Apache Web server stores its log files in:

/var/log/apache2

An example of the access file is:

```
napier@ubuntu:/var/log/apache2$ cat access.log.1
192.168.75.1 - - [29/Dec/2009:09:09:25 -0800] "GET / HTTP/1.1" 200 471 "-" "Mozil-
la/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.8.1.20) Gecko/20081217
Firefox/2.0.0.20"
192.168.75.132 - - [30/Dec/2009:11:42:37 -0800] "GET / HTTP/1.0" 200 430 "-" "-"
192.168.75.132 - - [30/Dec/2009:11:42:39 -0800] "GET / HTTP/1.1" 200 430 "-" "Mozil-
la/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
192.168.75.132 - - [30/Dec/2009:11:42:39 -0800] "GET /robots.txt HTTP/1.1" 404 491 "-"
"Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
192.168.75.132 - - [30/Dec/2009:11:42:39 -0800] "GET /favicon.ico HTTP/1.1" 404 492 "-
" "Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)"
```

In Linux many of the system messages are stored to messages, or syslog. For example from messages:

```
Jan  4 13:07:51 ubuntu ftpd[2044]: connection from bills.local
Jan  4 13:08:02 ubuntu ftpd[2044]: FTP LOGIN FROM bills.local as napier
```

9.16 Tripwire

Tripwire is a useful method of watching key file and auditing their changes. In Ubuntu a profile is created by editing:

```
/usr/tripwire/twpol.txt
```

This is then is compiled into an encrypted policy file with (and produces tw.cfg):

```
twadmin --create-polfile --cfgfile ./tw.cfg --site-keyfile  ./site.key
./twpol.txt
```

of which the database is created with (using the tw.pol file):

```
tripwire --init --cfgfile /etc/tripwire/tw.cfg --polfile
/etc/tripwire/tw.pol --site-keyfile /etc/tripwire/site.key --local-keyfile
/etc/tripwire/ubuntu-local.key
```

Then using:

```
tripwire --check
```

produces a report of the system changes:

```
Database file used:           /var/lib/tripwire/ubuntu.twd
Command line used:            tripwire --check

===============================================================================
Rule Summary:
===============================================================================

-------------------------------------------------------------------------------
  Section: Unix File System
-------------------------------------------------------------------------------

  Rule Name                       Severity Level    Added    Removed  Modified
  ---------                       --------------    -----    -------  --------
  Invariant Directories           66                0        0        0
* Tripwire Data Files             100               1        0        0
  Other binaries                  66                0        0        0
  Tripwire Binaries               100               0        0        0
  Other libraries                 66                0        0        0
  Root file-system executables    100               0        0        0
  System boot changes             100               0        0        0
  Root file-system libraries      100               0        0        0
  (/lib)
  Critical system boot files      100               0        0        0
* Other configuration files       66                0        0        2
  (/etc)
  Boot Scripts                    100               0        0        0
  Security Control                66                0        0        0
  Root config files               100               0        0        0
* Devices & Kernel information    100               159      155      0

Total objects scanned:  70781
Total violations found:  317

===============================================================================
Object Summary:
===============================================================================

-------------------------------------------------------------------------------
# Section: Unix File System
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
Rule Name: Tripwire Data Files (/var/lib/tripwire/ubuntu.twd)
Severity Level: 100
-------------------------------------------------------------------------------

Added:
"/var/lib/tripwire/ubuntu.twd"

-------------------------------------------------------------------------------
Rule Name: Other configuration files (/etc)
Severity Level: 66
-------------------------------------------------------------------------------

Modified:
"/etc/tripwire"
```

```
"/etc/tripwire/list"
```

Where we can see that the file "list" has been changed. A sample rule is given next, where Tripwire watches the passwd and shadow files:

```
(
  rulename = "Security Control",
  severity = $(SIG_MED)
)
{
   /etc/passwd    -> $(SEC_CONFIG) ;
   /etc/shadow    -> $(SEC_CONFIG) ;
}
```

When the /etc/passwd file changes it results in:

```
-------------------------------------------------------------------------------
Rule Name: Security Control (/etc/passwd)
Severity Level: 66
-------------------------------------------------------------------------------

Modified:
"/etc/passwd"
```

Tripwire Demo Link:
http://buchananweb.co.uk/adv_security_and_network_forensics/tripwire/

9.17 Tutorial

The main tutorial is at:

On-line tutorial: http://buchananweb.co.uk/adv03.html

9.18 Network Forensics tutorial

A. Download, install and run client.exe from:

   http://buchananweb.co.uk/dotnetclientserver.zip

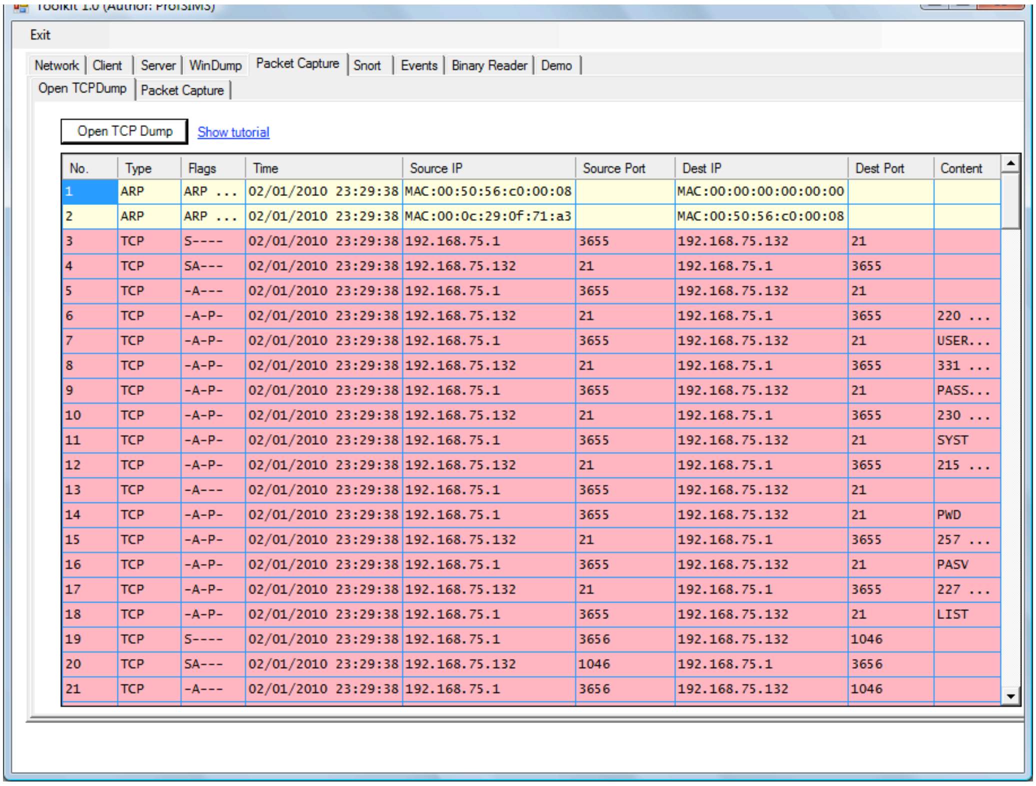
B. Within Toolkit, select the Packet Capture tab and then the Open TCPDump tab.

FTP Analysis Demo:
http://buchananweb.co.uk/adv_security_and_network_forensics/tcpdump01/tcpdump01.htm

Note: If you prefer to use Wireshark, the Pcap dump files are at:

http://buchananweb.co.uk/log/

9.1 Open **ftp** dump (see Figure 9.1).



No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	ARP	ARP ...	02/01/2010 23:29:38	MAC:00:50:56:c0:00:08		MAC:00:00:00:00:00:00		
2	ARP	ARP ...	02/01/2010 23:29:38	MAC:00:0c:29:0f:71:a3		MAC:00:50:56:c0:00:08		
3	TCP	S----	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	
4	TCP	SA---	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	
5	TCP	-A---	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	
6	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	220 ...
7	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	USER...
8	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	331 ...
9	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	PASS...
10	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	230 ...
11	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	SYST
12	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	215 ...
13	TCP	-A---	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	
14	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	PWD
15	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	257 ...
16	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	PASV
17	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.132	21	192.168.75.1	3655	227 ...
18	TCP	-A-P-	02/01/2010 23:29:38	192.168.75.1	3655	192.168.75.132	21	LIST
19	TCP	S----	02/01/2010 23:29:38	192.168.75.1	3656	192.168.75.132	1046	
20	TCP	SA---	02/01/2010 23:29:38	192.168.75.132	1046	192.168.75.1	3656	
21	TCP	-A---	02/01/2010 23:29:38	192.168.75.1	3656	192.168.75.132	1046	

Figure 9.1: FTP Dump

Determine the following:

Host src TCP port (Hint: Examine the Source Port on Packet 3):

Server src TCP port (Hint: Examine the Destination Port on Packet 3):

Host src IP address (Hint: Examine the Source IP on Packet 3):

Server src IP address (Hint: Examine the Dest IP on Packet 3):

What is the MAC address of the server (Hint: Examine the reply for Packet 2):

Identify the packets used for the SYN, SYN/ACK and ACK sequence (Hint: packets 3 to 5 look interesting):

Which is the return code used by the FTP server to identify:

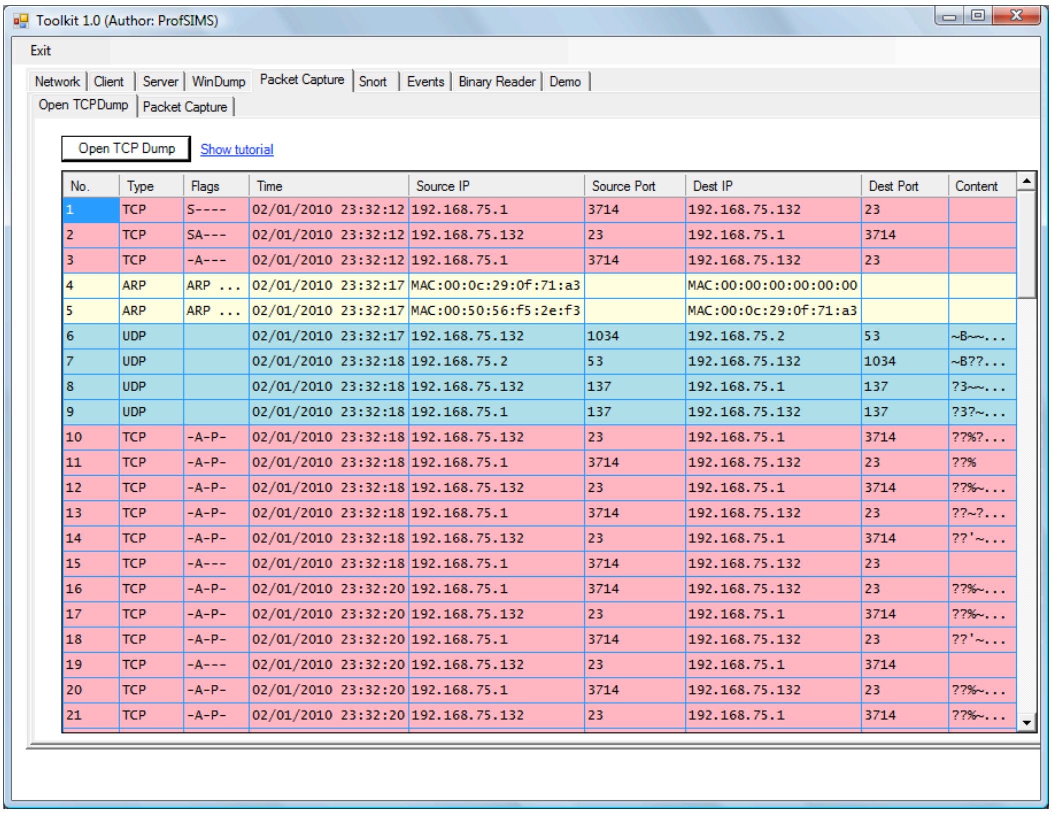
Password Required (Hint: Examine the content on Packet 9):

Server type (Hint: Examine the content on Packet 12):

Which FTP command is used to determine the current working folder (Hint: Examine the content on Packet 15):

Which FTP command is used to determine the files in a folder (Hint: Examine the content on Packet 18): **Which FTP port has been used for the FTP directory list (hint: Examine the contents of Packet 17, and the last two digits of the 227 response (first multiplied by 256 added to the second):** **Identify the data packets used to list the contents (Hint port 1046 looks interesting):** **Which FTP port has been used for the FTP file transfer (hint: it is the last two digits of the 227 response (first multiplied by 256 added to the second):** **Identify the data packets used to transfer the file:** **What is the name of the file transferred:**

9.2 Open Telnet dump (see Figure 9.2).



No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	TCP	S----	02/01/2010 23:32:12	192.168.75.1	3714	192.168.75.132	23	
2	TCP	SA---	02/01/2010 23:32:12	192.168.75.132	23	192.168.75.1	3714	
3	TCP	-A---	02/01/2010 23:32:12	192.168.75.1	3714	192.168.75.132	23	
4	ARP	ARP ...	02/01/2010 23:32:17	MAC:00:0c:29:0f:71:a3		MAC:00:00:00:00:00:00		
5	ARP	ARP ...	02/01/2010 23:32:17	MAC:00:50:56:f5:2e:f3		MAC:00:0c:29:0f:71:a3		
6	UDP		02/01/2010 23:32:17	192.168.75.132	1034	192.168.75.2	53	~B~~...
7	UDP		02/01/2010 23:32:18	192.168.75.2	53	192.168.75.132	1034	~B??...
8	UDP		02/01/2010 23:32:18	192.168.75.132	137	192.168.75.1	137	?3~~...
9	UDP		02/01/2010 23:32:18	192.168.75.1	137	192.168.75.132	137	?3?~...
10	TCP	-A-P-	02/01/2010 23:32:18	192.168.75.132	23	192.168.75.1	3714	??%?...
11	TCP	-A-P-	02/01/2010 23:32:18	192.168.75.1	3714	192.168.75.132	23	??%
12	TCP	-A-P-	02/01/2010 23:32:18	192.168.75.132	23	192.168.75.1	3714	??%~...
13	TCP	-A-P-	02/01/2010 23:32:18	192.168.75.1	3714	192.168.75.132	23	??~?...
14	TCP	-A-P-	02/01/2010 23:32:18	192.168.75.132	23	192.168.75.1	3714	??'~...
15	TCP	-A---	02/01/2010 23:32:18	192.168.75.1	3714	192.168.75.132	23	
16	TCP	-A-P-	02/01/2010 23:32:20	192.168.75.1	3714	192.168.75.132	23	??%~...
17	TCP	-A-P-	02/01/2010 23:32:20	192.168.75.132	23	192.168.75.1	3714	??%~...
18	TCP	-A-P-	02/01/2010 23:32:20	192.168.75.1	3714	192.168.75.132	23	??'~...
19	TCP	-A---	02/01/2010 23:32:20	192.168.75.132	23	192.168.75.1	3714	
20	TCP	-A-P-	02/01/2010 23:32:20	192.168.75.1	3714	192.168.75.132	23	??%~...
21	TCP	-A-P-	02/01/2010 23:32:20	192.168.75.132	23	192.168.75.1	3714	??%~...

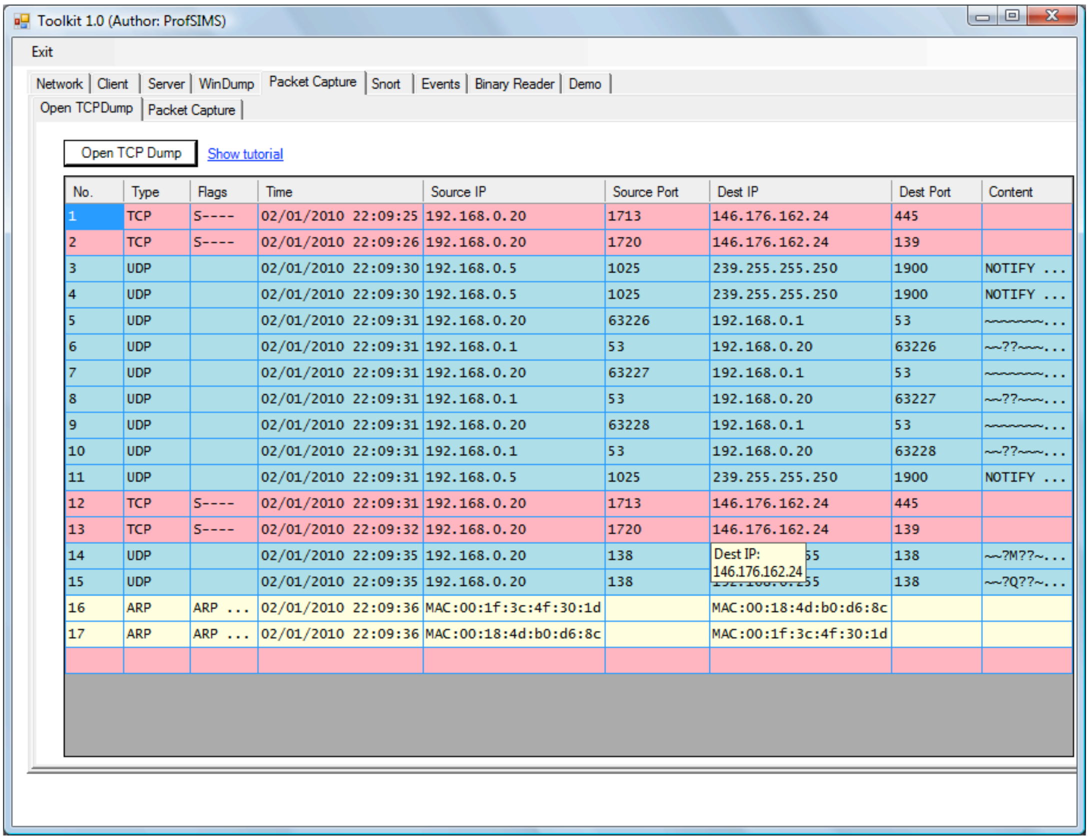
Figure 9.2: Telnet dump

Determine the following:

Host src TCP port:

Server src TCP port:
Host src IP address:
Server src IP address:
Identify the packets used for the SYN, SYN/ACK and ACK sequence:
What is the login name:
What is the password:
What commands were entered, once the Telnet connection was made:

9.3 Open dns dump (see Figure 9.3).



No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	TCP	S----	02/01/2010 22:09:25	192.168.0.20	1713	146.176.162.24	445	
2	TCP	S----	02/01/2010 22:09:26	192.168.0.20	1720	146.176.162.24	139	
3	UDP		02/01/2010 22:09:30	192.168.0.5	1025	239.255.255.250	1900	NOTIFY ...
4	UDP		02/01/2010 22:09:30	192.168.0.5	1025	239.255.255.250	1900	NOTIFY ...
5	UDP		02/01/2010 22:09:31	192.168.0.20	63226	192.168.0.1	53	~~~~~~~...
6	UDP		02/01/2010 22:09:31	192.168.0.1	53	192.168.0.20	63226	~~??~~~...
7	UDP		02/01/2010 22:09:31	192.168.0.20	63227	192.168.0.1	53	~~~~~~~...
8	UDP		02/01/2010 22:09:31	192.168.0.1	53	192.168.0.20	63227	~~??~~~...
9	UDP		02/01/2010 22:09:31	192.168.0.20	63228	192.168.0.1	53	~~~~~~~...
10	UDP		02/01/2010 22:09:31	192.168.0.1	53	192.168.0.20	63228	~~??~~~...
11	UDP		02/01/2010 22:09:31	192.168.0.5	1025	239.255.255.250	1900	NOTIFY ...
12	TCP	S----	02/01/2010 22:09:31	192.168.0.20	1713	146.176.162.24	445	
13	TCP	S----	02/01/2010 22:09:32	192.168.0.20	1720	146.176.162.24	139	
14	UDP		02/01/2010 22:09:35	192.168.0.20	138	[illegible]	138	~~?M??~...
15	UDP		02/01/2010 22:09:35	192.168.0.20	138	192.168.0.255	138	~~?Q??~...
16	ARP	ARP ...	02/01/2010 22:09:36	MAC:00:1f:3c:4f:30:1d		MAC:00:18:4d:b0:d6:8c		
17	ARP	ARP ...	02/01/2010 22:09:36	MAC:00:18:4d:b0:d6:8c		MAC:00:1f:3c:4f:30:1d		

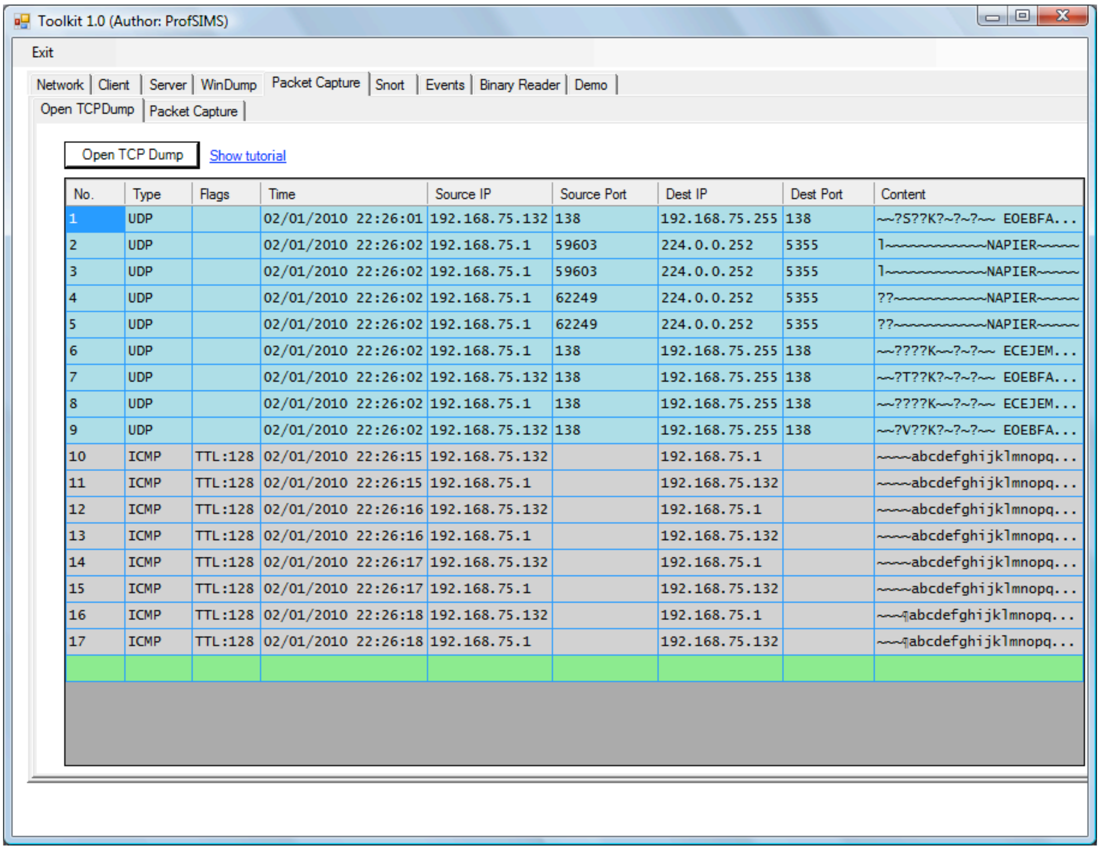
Figure 9.3: DNS dump

Determine the following:

What is the transport layer protocol used for DNS:

Host src UDP port:
Server (DNS) src UDP port:
Host src IP address:
Server (DNS) src IP address:
Identify the data packets used to for the DNS lookup:

9.4 Open ping dump (see Figure 9.4).



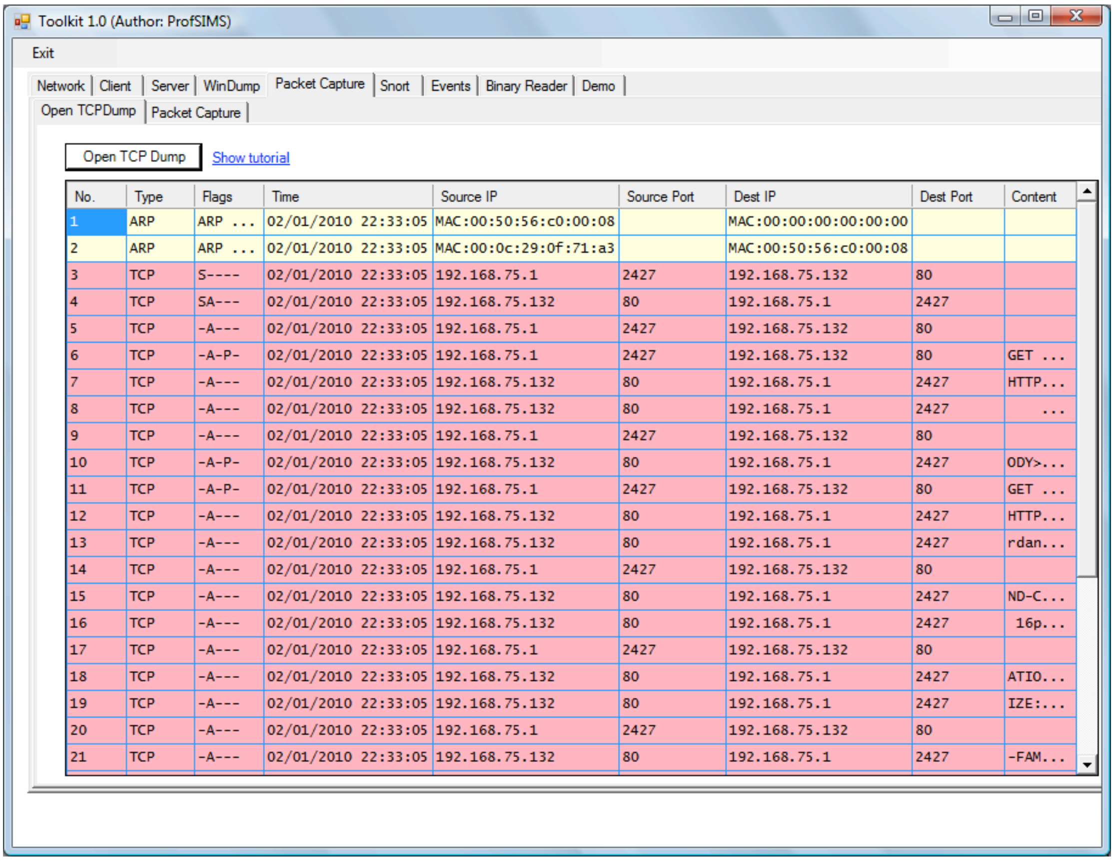
No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	UDP		02/01/2010 22:26:01	192.168.75.132	138	192.168.75.255	138	~~?S??K?~?~?~~ EOEBFA...
2	UDP		02/01/2010 22:26:02	192.168.75.1	59603	224.0.0.252	5355	l~~~~~~~~~~~NAPIER~~~~~
3	UDP		02/01/2010 22:26:02	192.168.75.1	59603	224.0.0.252	5355	l~~~~~~~~~~~NAPIER~~~~~
4	UDP		02/01/2010 22:26:02	192.168.75.1	62249	224.0.0.252	5355	??~~~~~~~~~~NAPIER~~~~~
5	UDP		02/01/2010 22:26:02	192.168.75.1	62249	224.0.0.252	5355	??~~~~~~~~~~NAPIER~~~~~
6	UDP		02/01/2010 22:26:02	192.168.75.1	138	192.168.75.255	138	~~????K~~?~?~~ ECEJEM...
7	UDP		02/01/2010 22:26:02	192.168.75.132	138	192.168.75.255	138	~~?T??K?~?~?~~ EOEBFA...
8	UDP		02/01/2010 22:26:02	192.168.75.1	138	192.168.75.255	138	~~????K~~?~?~~ ECEJEM...
9	UDP		02/01/2010 22:26:02	192.168.75.132	138	192.168.75.255	138	~~?V??K?~?~?~~ EOEBFA...
10	ICMP	TTL:128	02/01/2010 22:26:15	192.168.75.132		192.168.75.1		~~~~abcdefghijklmnopq...
11	ICMP	TTL:128	02/01/2010 22:26:15	192.168.75.1		192.168.75.132		~~~~abcdefghijklmnopq...
12	ICMP	TTL:128	02/01/2010 22:26:16	192.168.75.132		192.168.75.1		~~~~abcdefghijklmnopq...
13	ICMP	TTL:128	02/01/2010 22:26:16	192.168.75.1		192.168.75.132		~~~~abcdefghijklmnopq...
14	ICMP	TTL:128	02/01/2010 22:26:17	192.168.75.132		192.168.75.1		~~~~abcdefghijklmnopq...
15	ICMP	TTL:128	02/01/2010 22:26:17	192.168.75.1		192.168.75.132		~~~~abcdefghijklmnopq...
16	ICMP	TTL:128	02/01/2010 22:26:18	192.168.75.132		192.168.75.1		~~~¶abcdefghijklmnopq...
17	ICMP	TTL:128	02/01/2010 22:26:18	192.168.75.1		192.168.75.132		~~~¶abcdefghijklmnopq...

Figure 9.4: ICMP dump

Determine the following:

Host src IP address:
Server (DNS) src IP address:
Identify the data packets used to for the ping:
How many ECHO's where send from the host, and how many replies where there:

9.5 Open **webpage** dump (see Figure 9.5).



No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	ARP	ARP ...	02/01/2010 22:33:05	MAC:00:50:56:c0:00:08		MAC:00:00:00:00:00:00		
2	ARP	ARP ...	02/01/2010 22:33:05	MAC:00:0c:29:0f:71:a3		MAC:00:50:56:c0:00:08		
3	TCP	S----	02/01/2010 22:33:05	192.168.75.1	2427	192.168.75.132	80	
4	TCP	SA---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	
5	TCP	-A---	02/01/2010 22:33:05	192.168.75.1	2427	192.168.75.132	80	
6	TCP	-A-P-	02/01/2010 22:33:05	192.168.75.1	2427	192.168.75.132	80	GET ...
7	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	HTTP...
8	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	...
9	TCP	-A---	02/01/2010 22:33:05	192.168.75.1	2427	192.168.75.132	80	
10	TCP	-A-P-	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	ODY>...
11	TCP	-A-P-	02/01/2010 22:33:05	192.168.75.1	2427	192.168.75.132	80	GET ...
12	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	HTTP...
13	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	rdan...
14	TCP	-A---	02/01/2010 22:33:05	192.168.75.1	2427	192.168.75.132	80	
15	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	ND-C...
16	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	16p...
17	TCP	-A---	02/01/2010 22:33:05	192.168.75.1	2427	192.168.75.132	80	
18	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	ATIO...
19	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	IZE:...
20	TCP	-A---	02/01/2010 22:33:05	192.168.75.1	2427	192.168.75.132	80	
21	TCP	-A---	02/01/2010 22:33:05	192.168.75.132	80	192.168.75.1	2427	-FAM...

Figure 9.5: Web dump

Determine the following:

Host src TCP port:

Server src TCP port:

Host src IP address:

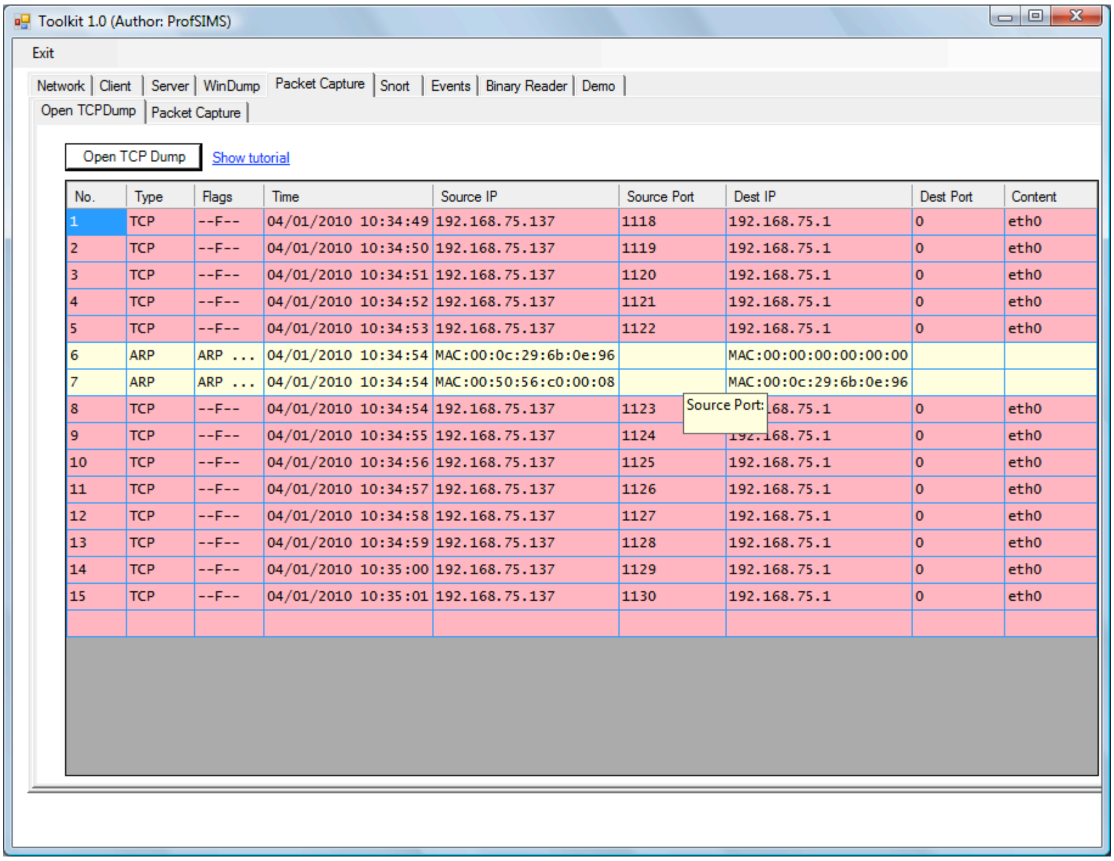
Server src IP address:

Identify the packets used for the SYN, SYN/ACK and ACK sequence:

What is the HTTP command used to get the default page (Hint: put your cursor over the content of the 4th data packet):

What is the HTTP response to a successful request (Hint: put your cursor over the content of the 5th data packet):

9.6 Open **hping_fin** dump (see Figure 9.6). We can see that a remote host is sending TCP segments with the FIN flag sent.



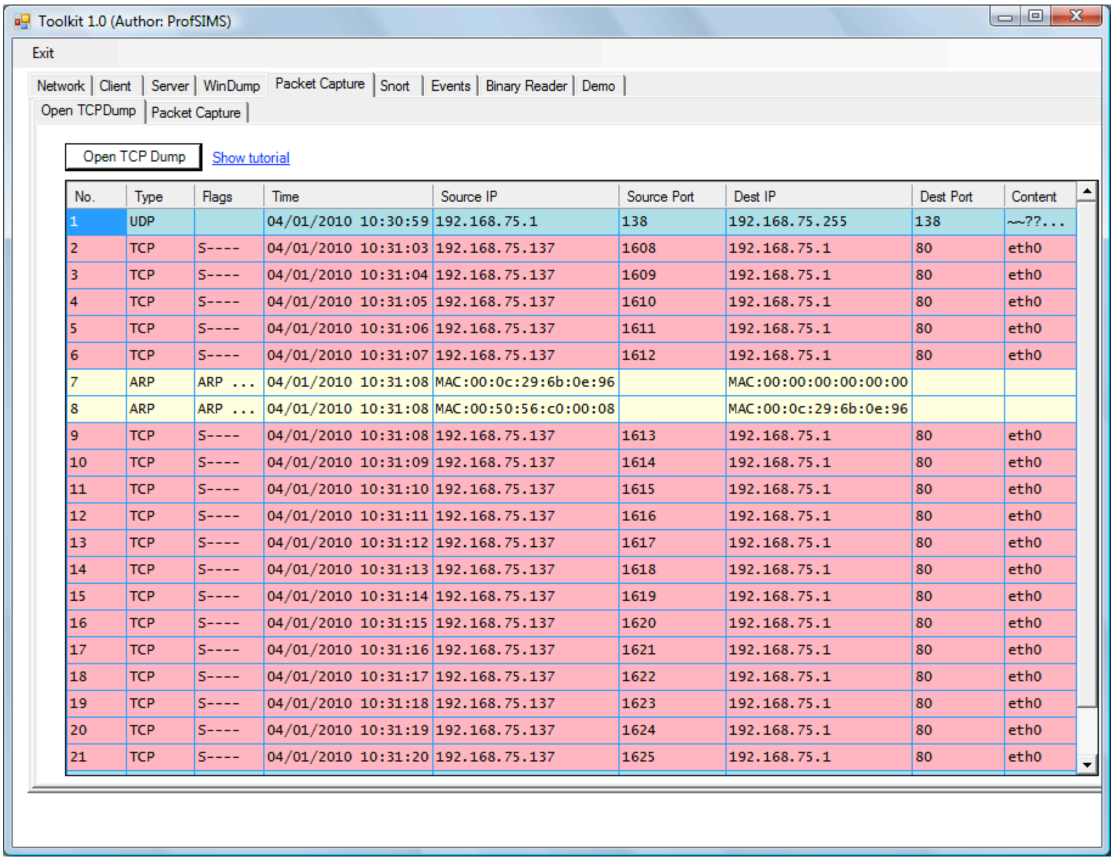
No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	TCP	--F--	04/01/2010 10:34:49	192.168.75.137	1118	192.168.75.1	0	eth0
2	TCP	--F--	04/01/2010 10:34:50	192.168.75.137	1119	192.168.75.1	0	eth0
3	TCP	--F--	04/01/2010 10:34:51	192.168.75.137	1120	192.168.75.1	0	eth0
4	TCP	--F--	04/01/2010 10:34:52	192.168.75.137	1121	192.168.75.1	0	eth0
5	TCP	--F--	04/01/2010 10:34:53	192.168.75.137	1122	192.168.75.1	0	eth0
6	ARP	ARP ...	04/01/2010 10:34:54	MAC:00:0c:29:6b:0e:96		MAC:00:00:00:00:00:00		
7	ARP	ARP ...	04/01/2010 10:34:54	MAC:00:50:56:c0:00:08		MAC:00:0c:29:6b:0e:96		
8	TCP	--F--	04/01/2010 10:34:54	192.168.75.137	1123	68.75.1	0	eth0
9	TCP	--F--	04/01/2010 10:34:55	192.168.75.137	1124	192.168.75.1	0	eth0
10	TCP	--F--	04/01/2010 10:34:56	192.168.75.137	1125	192.168.75.1	0	eth0
11	TCP	--F--	04/01/2010 10:34:57	192.168.75.137	1126	192.168.75.1	0	eth0
12	TCP	--F--	04/01/2010 10:34:58	192.168.75.137	1127	192.168.75.1	0	eth0
13	TCP	--F--	04/01/2010 10:34:59	192.168.75.137	1128	192.168.75.1	0	eth0
14	TCP	--F--	04/01/2010 10:35:00	192.168.75.137	1129	192.168.75.1	0	eth0
15	TCP	--F--	04/01/2010 10:35:01	192.168.75.137	1130	192.168.75.1	0	eth0

Figure 9.6: hping_fin dump

Determine the following:

Sending src TCP port range:
Receiver src TCP port:
Sending src IP address:
Receiver src IP address:

9.7 Open **hping_port80** dump (see Figure 9.7). We can see that a remote host is sending TCP segments with the SYN flag sent.



No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	UDP		04/01/2010 10:30:59	192.168.75.1	138	192.168.75.255	138	~~??...
2	TCP	S----	04/01/2010 10:31:03	192.168.75.137	1608	192.168.75.1	80	eth0
3	TCP	S----	04/01/2010 10:31:04	192.168.75.137	1609	192.168.75.1	80	eth0
4	TCP	S----	04/01/2010 10:31:05	192.168.75.137	1610	192.168.75.1	80	eth0
5	TCP	S----	04/01/2010 10:31:06	192.168.75.137	1611	192.168.75.1	80	eth0
6	TCP	S----	04/01/2010 10:31:07	192.168.75.137	1612	192.168.75.1	80	eth0
7	ARP	ARP ...	04/01/2010 10:31:08	MAC:00:0c:29:6b:0e:96		MAC:00:00:00:00:00:00		
8	ARP	ARP ...	04/01/2010 10:31:08	MAC:00:50:56:c0:00:08		MAC:00:0c:29:6b:0e:96		
9	TCP	S----	04/01/2010 10:31:08	192.168.75.137	1613	192.168.75.1	80	eth0
10	TCP	S----	04/01/2010 10:31:09	192.168.75.137	1614	192.168.75.1	80	eth0
11	TCP	S----	04/01/2010 10:31:10	192.168.75.137	1615	192.168.75.1	80	eth0
12	TCP	S----	04/01/2010 10:31:11	192.168.75.137	1616	192.168.75.1	80	eth0
13	TCP	S----	04/01/2010 10:31:12	192.168.75.137	1617	192.168.75.1	80	eth0
14	TCP	S----	04/01/2010 10:31:13	192.168.75.137	1618	192.168.75.1	80	eth0
15	TCP	S----	04/01/2010 10:31:14	192.168.75.137	1619	192.168.75.1	80	eth0
16	TCP	S----	04/01/2010 10:31:15	192.168.75.137	1620	192.168.75.1	80	eth0
17	TCP	S----	04/01/2010 10:31:16	192.168.75.137	1621	192.168.75.1	80	eth0
18	TCP	S----	04/01/2010 10:31:17	192.168.75.137	1622	192.168.75.1	80	eth0
19	TCP	S----	04/01/2010 10:31:18	192.168.75.137	1623	192.168.75.1	80	eth0
20	TCP	S----	04/01/2010 10:31:19	192.168.75.137	1624	192.168.75.1	80	eth0
21	TCP	S----	04/01/2010 10:31:20	192.168.75.137	1625	192.168.75.1	80	eth0

Figure 9.7: hping_fin dump

Determine the following:

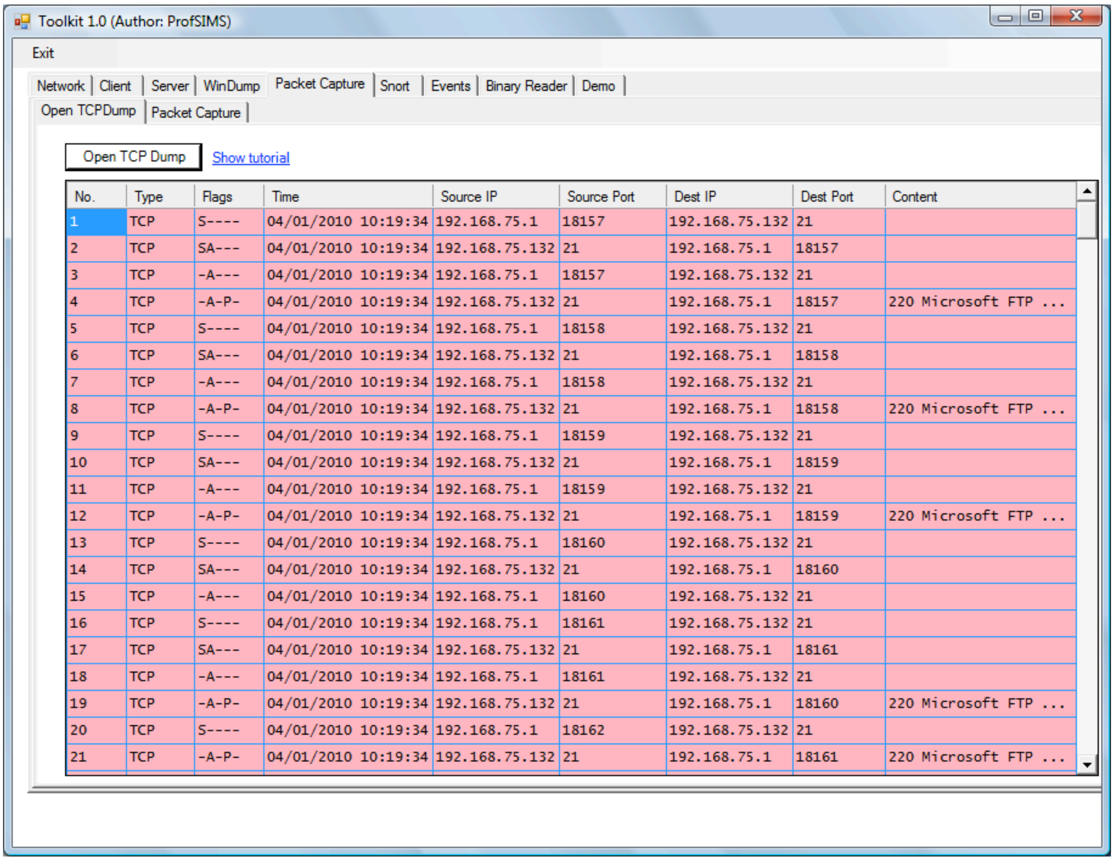
Sending src TCP port range:

Receiver src TCP port:

Sending src IP address:

Receiver src IP address:

9.8 Open **hydra_ftp** dump (see Figure 9.8). We can see that a Hydra attack has been conducted on our server.



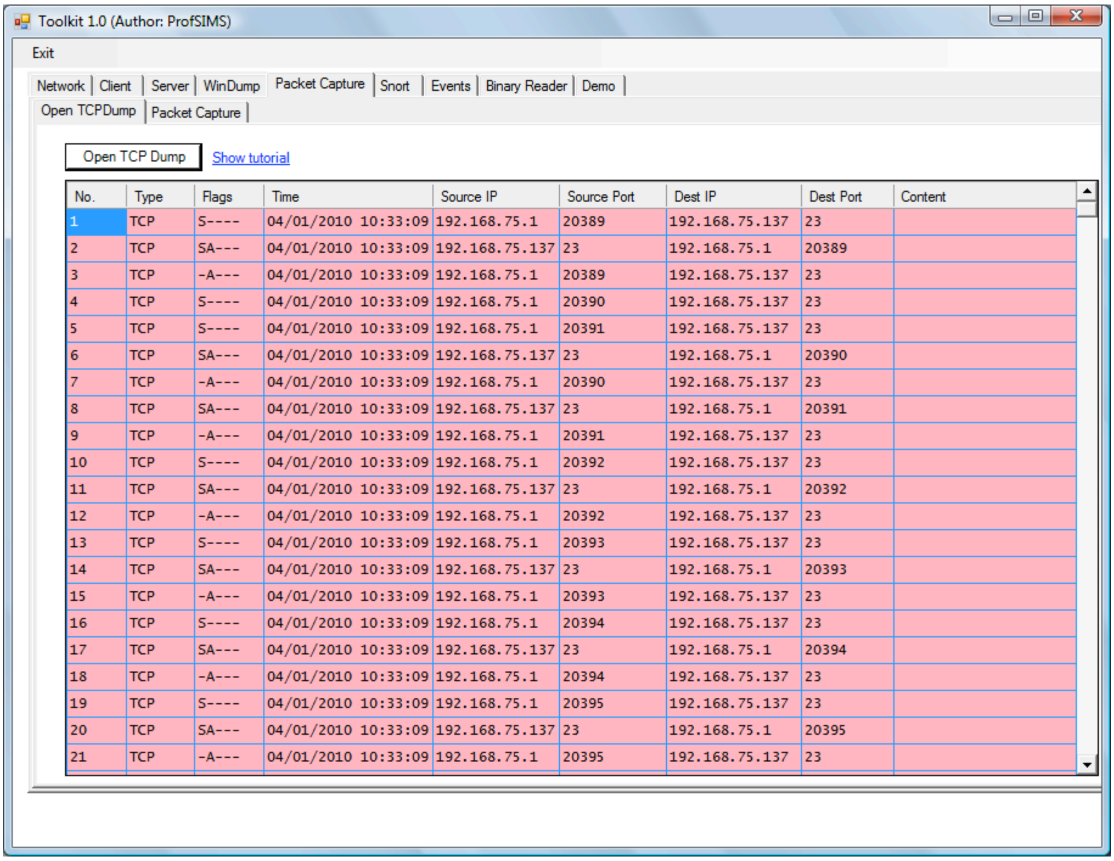
No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	TCP	S----	04/01/2010 10:19:34	192.168.75.1	18157	192.168.75.132	21	
2	TCP	SA---	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18157	
3	TCP	-A---	04/01/2010 10:19:34	192.168.75.1	18157	192.168.75.132	21	
4	TCP	-A-P-	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18157	220 Microsoft FTP ...
5	TCP	S----	04/01/2010 10:19:34	192.168.75.1	18158	192.168.75.132	21	
6	TCP	SA---	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18158	
7	TCP	-A---	04/01/2010 10:19:34	192.168.75.1	18158	192.168.75.132	21	
8	TCP	-A-P-	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18158	220 Microsoft FTP ...
9	TCP	S----	04/01/2010 10:19:34	192.168.75.1	18159	192.168.75.132	21	
10	TCP	SA---	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18159	
11	TCP	-A---	04/01/2010 10:19:34	192.168.75.1	18159	192.168.75.132	21	
12	TCP	-A-P-	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18159	220 Microsoft FTP ...
13	TCP	S----	04/01/2010 10:19:34	192.168.75.1	18160	192.168.75.132	21	
14	TCP	SA---	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18160	
15	TCP	-A---	04/01/2010 10:19:34	192.168.75.1	18160	192.168.75.132	21	
16	TCP	S----	04/01/2010 10:19:34	192.168.75.1	18161	192.168.75.132	21	
17	TCP	SA---	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18161	
18	TCP	-A---	04/01/2010 10:19:34	192.168.75.1	18161	192.168.75.132	21	
19	TCP	-A-P-	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18160	220 Microsoft FTP ...
20	TCP	S----	04/01/2010 10:19:34	192.168.75.1	18162	192.168.75.132	21	
21	TCP	-A-P-	04/01/2010 10:19:34	192.168.75.132	21	192.168.75.1	18161	220 Microsoft FTP ...

Figure 9.8: Hydra_ftp dump

Determine the following:

Sending src TCP port range:
Receiver src TCP port:
Sending src IP address:
Receiver src IP address:
What are the logins used:
What are the passwords used:
What is the successful login/password:

9.9 Open **hydra_telnet** dump (see Figure 9.9). We can see that a Hydra attack has been conducted on our server.



No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	TCP	S----	04/01/2010 10:33:09	192.168.75.1	20389	192.168.75.137	23	
2	TCP	SA---	04/01/2010 10:33:09	192.168.75.137	23	192.168.75.1	20389	
3	TCP	-A---	04/01/2010 10:33:09	192.168.75.1	20389	192.168.75.137	23	
4	TCP	S----	04/01/2010 10:33:09	192.168.75.1	20390	192.168.75.137	23	
5	TCP	S----	04/01/2010 10:33:09	192.168.75.1	20391	192.168.75.137	23	
6	TCP	SA---	04/01/2010 10:33:09	192.168.75.137	23	192.168.75.1	20390	
7	TCP	-A---	04/01/2010 10:33:09	192.168.75.1	20390	192.168.75.137	23	
8	TCP	SA---	04/01/2010 10:33:09	192.168.75.137	23	192.168.75.1	20391	
9	TCP	-A---	04/01/2010 10:33:09	192.168.75.1	20391	192.168.75.137	23	
10	TCP	S----	04/01/2010 10:33:09	192.168.75.1	20392	192.168.75.137	23	
11	TCP	SA---	04/01/2010 10:33:09	192.168.75.137	23	192.168.75.1	20392	
12	TCP	-A---	04/01/2010 10:33:09	192.168.75.1	20392	192.168.75.137	23	
13	TCP	S----	04/01/2010 10:33:09	192.168.75.1	20393	192.168.75.137	23	
14	TCP	SA---	04/01/2010 10:33:09	192.168.75.137	23	192.168.75.1	20393	
15	TCP	-A---	04/01/2010 10:33:09	192.168.75.1	20393	192.168.75.137	23	
16	TCP	S----	04/01/2010 10:33:09	192.168.75.1	20394	192.168.75.137	23	
17	TCP	SA---	04/01/2010 10:33:09	192.168.75.137	23	192.168.75.1	20394	
18	TCP	-A---	04/01/2010 10:33:09	192.168.75.1	20394	192.168.75.137	23	
19	TCP	S----	04/01/2010 10:33:09	192.168.75.1	20395	192.168.75.137	23	
20	TCP	SA---	04/01/2010 10:33:09	192.168.75.137	23	192.168.75.1	20395	
21	TCP	-A---	04/01/2010 10:33:09	192.168.75.1	20395	192.168.75.137	23	

Figure 9.9: Hydra_telnet dump

Determine the following:

Sending src TCP port range:

Receiver src TCP port:

Sending src IP address:

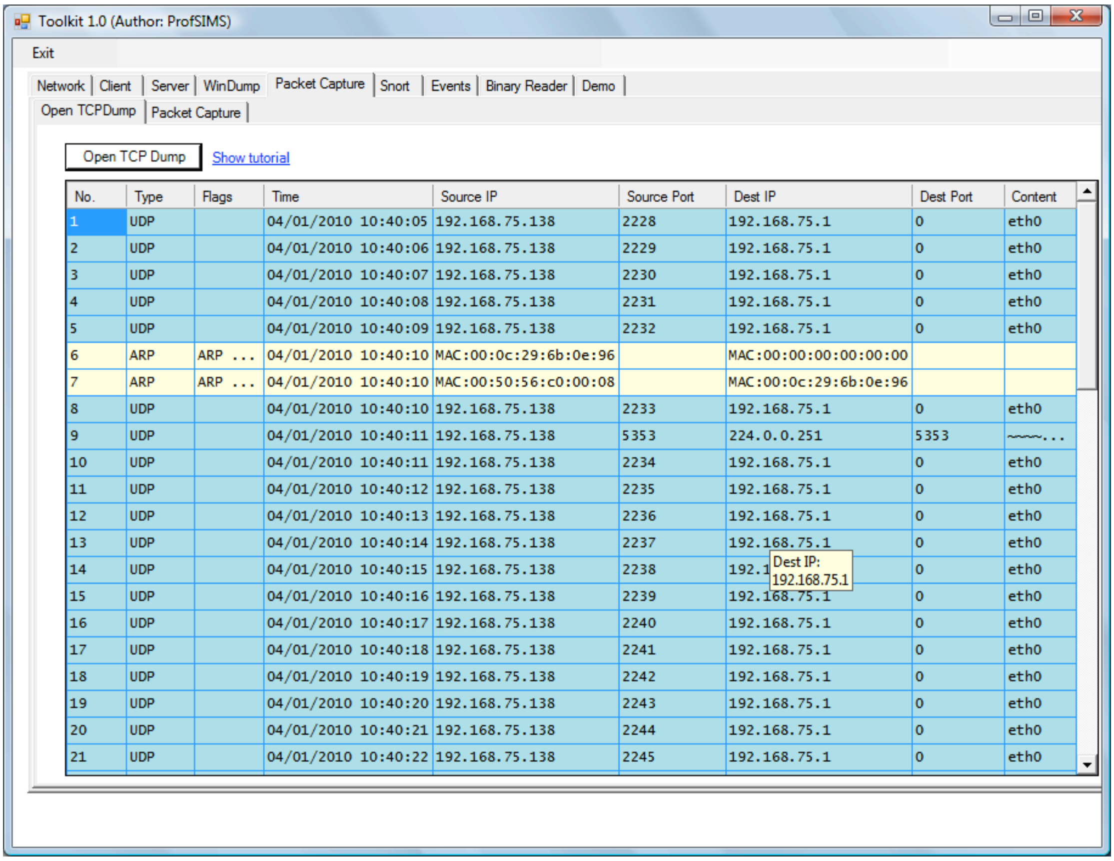
Receiver src IP address:

What are the logins used:

What are the passwords used:

What is the successful login/password:

9.10 Open **hping_udp_scan** dump (see Figure 1.10).



No.	Type	Flags	Time	Source IP	Source Port	Dest IP	Dest Port	Content
1	UDP		04/01/2010 10:40:05	192.168.75.138	2228	192.168.75.1	0	eth0
2	UDP		04/01/2010 10:40:06	192.168.75.138	2229	192.168.75.1	0	eth0
3	UDP		04/01/2010 10:40:07	192.168.75.138	2230	192.168.75.1	0	eth0
4	UDP		04/01/2010 10:40:08	192.168.75.138	2231	192.168.75.1	0	eth0
5	UDP		04/01/2010 10:40:09	192.168.75.138	2232	192.168.75.1	0	eth0
6	ARP	ARP ...	04/01/2010 10:40:10	MAC:00:0c:29:6b:0e:96		MAC:00:00:00:00:00:00		
7	ARP	ARP ...	04/01/2010 10:40:10	MAC:00:50:56:c0:00:08		MAC:00:0c:29:6b:0e:96		
8	UDP		04/01/2010 10:40:10	192.168.75.138	2233	192.168.75.1	0	eth0
9	UDP		04/01/2010 10:40:11	192.168.75.138	5353	224.0.0.251	5353	~~~...
10	UDP		04/01/2010 10:40:11	192.168.75.138	2234	192.168.75.1	0	eth0
11	UDP		04/01/2010 10:40:12	192.168.75.138	2235	192.168.75.1	0	eth0
12	UDP		04/01/2010 10:40:13	192.168.75.138	2236	192.168.75.1	0	eth0
13	UDP		04/01/2010 10:40:14	192.168.75.138	2237	192.168.75.1	0	eth0
14	UDP		04/01/2010 10:40:15	192.168.75.138	2238	192.1	0	eth0
15	UDP		04/01/2010 10:40:16	192.168.75.138	2239	192.168.75.1	0	eth0
16	UDP		04/01/2010 10:40:17	192.168.75.138	2240	192.168.75.1	0	eth0
17	UDP		04/01/2010 10:40:18	192.168.75.138	2241	192.168.75.1	0	eth0
18	UDP		04/01/2010 10:40:19	192.168.75.138	2242	192.168.75.1	0	eth0
19	UDP		04/01/2010 10:40:20	192.168.75.138	2243	192.168.75.1	0	eth0
20	UDP		04/01/2010 10:40:21	192.168.75.138	2244	192.168.75.1	0	eth0
21	UDP		04/01/2010 10:40:22	192.168.75.138	2245	192.168.75.1	0	eth0

Figure 9.10: Hping_UDP_scan

Determine the following:

Sending src UDP port range:

Receiver src UDP port:

Sending src IP address:

Receiver src IP address:

9.19 Tripwire tutorial

On-line demo:
http://buchananweb.co.uk/adv_security_and_network_forensics/tripwire/tripwire.htm

Note: The labs in this section require a virtual image defined in Appendix A.

9.11 Run the Linux virtual image (User name: Administrator, Password: napier123). Within the virtual image, run a Terminal and determine its IP address using **ifconfig**.

☞ What are the IP addresses of the server and the network address which will be used to connect to the virtual image:

9.12 Go to the /etc/tripwire folder, and view the twpol.txt file. Next run the following commands:

```
twadmin --create-polfile --cfgfile ./tw.cfg --site-keyfile  ./site.key
./twpol.txt
```

```
tripwire --init --cfgfile /etc/tripwire/tw.cfg --polfile
/etc/tripwire/tw.pol --site-keyfile /etc/tripwire/site.key --local-keyfile
/etc/tripwire/ubuntu-local.key
```

9.13 Go to the /etc/passwd file and change the owner to “napier”. Next go to the /tmp folder and change the ownership of this file too. Next run a check with tripwire:

```
tripwire –-check
```

☞ What do you observe from the results:

9.14 Create a new folder in your home directory, and add a rule to the policy file for Tripwire, and see if you can detect any changes on this folder.

☞ Rule used: