Peter Sanford
IT 2700
NetLab Lab 22
11/04/2023

1.1:
Step 6:

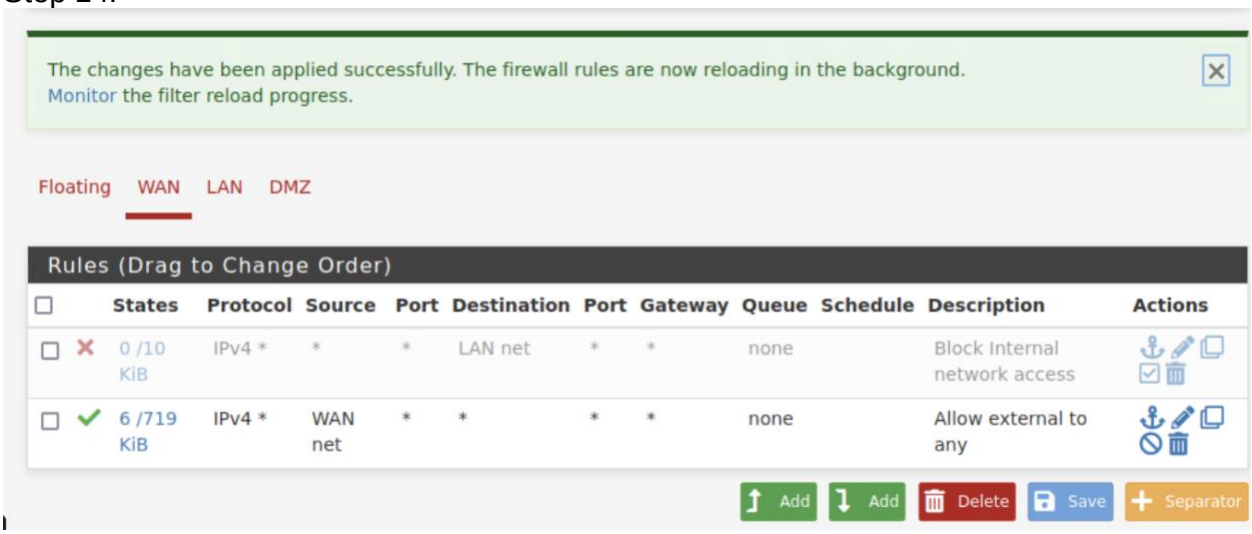
```
    so-thehive ------------------------------------------------------- [ OK ]
    so-thehive-es ---------------------------------------------------- [ OK ]
    so-wazuh --------------------------------------------------------- [ OK ]
    so-zeek ---------------------------------------------------------- [ OK ]

[sysadmin@seconion ~]$ ifconfig -a
bond0: flags=5443<UP,BROADCAST,RUNNING,PROMISC,MASTER,MULTICAST>  mtu 1500
        ether 00:50:56:00:00:ff  txqueuelen 1000  (Ethernet)
        RX packets 7215  bytes 607293 (593.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.255.0  broadcast 172.17.0.255
        ether 02:42:19:55:87:73  txqueuelen 0  (Ethernet)
        RX packets 99963  bytes 47723281 (45.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 103672  bytes 28504566 (27.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.6  netmask 255.255.255.0  broadcast 192.168.0.255
```

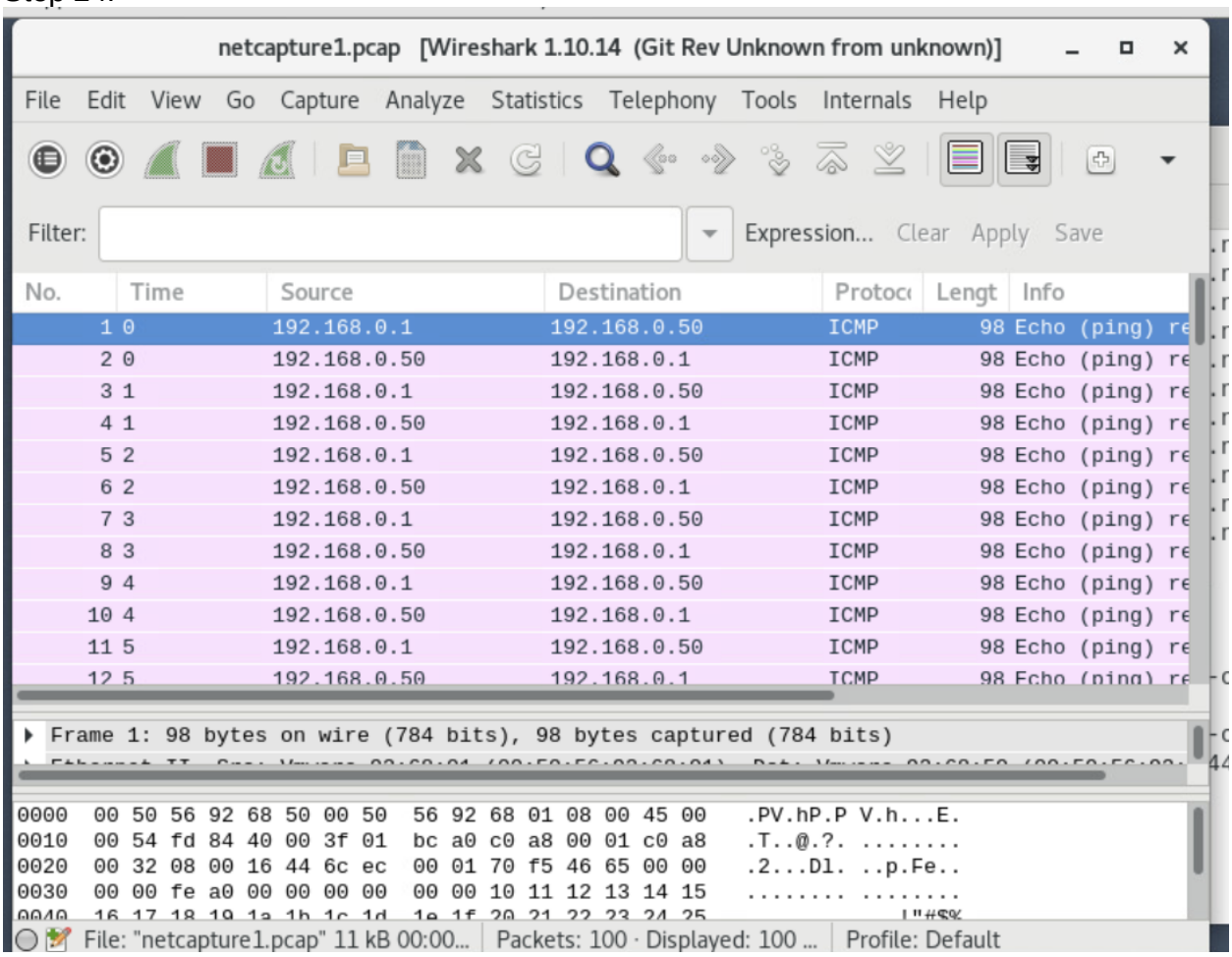
Step 14:



Step 19:

```
01:50:45.170313 IP pfsense.netlab.local > WinOS.netlab.local: ICMP echo request, id 32534, seq 6, length 64
01:50:45.170563 IP WinOS.netlab.local > pfsense.netlab.local: ICMP echo reply, id 32534, seq 6, length 64
01:50:46.194352 IP pfsense.netlab.local > WinOS.netlab.local: ICMP echo request, id 32534, seq 7, length 64
01:50:46.194693 IP WinOS.netlab.local > pfsense.netlab.local: ICMP echo reply, id 32534, seq 7, length 64
01:50:47.218343 IP pfsense.netlab.local > WinOS.netlab.local: ICMP echo request, id 32534, seq 8, length 64
01:50:47.218600 IP WinOS.netlab.local > pfsense.netlab.local: ICMP echo reply, id 32534, seq 8, length 64
01:50:48.281291 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 49031 unreachable, length 76
01:50:48.281560 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 49031 unreachable, length 76
01:50:54.645453 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 41906 unreachable, length 77
01:50:54.645500 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 54857 unreachable, length 77
01:50:55.552999 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 45758 unreachable, length 76
01:50:55.553303 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 45758 unreachable, length 76
01:51:04.612208 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 56178 unreachable, length 76
01:51:04.612475 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 56178 unreachable, length 76
01:51:12.784805 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 38057 unreachable, length 76
01:51:12.785103 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 38057 unreachable, length 76
01:51:13.687547 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 37199 unreachable, length 77
01:51:21.846520 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 42349 unreachable, length 76
01:51:21.846564 IP seconion.netlab.local > WinOS.netlab.local: ICMP seconion.netlab.local udp port 42349 unreachable, length 76
```

Step 24:



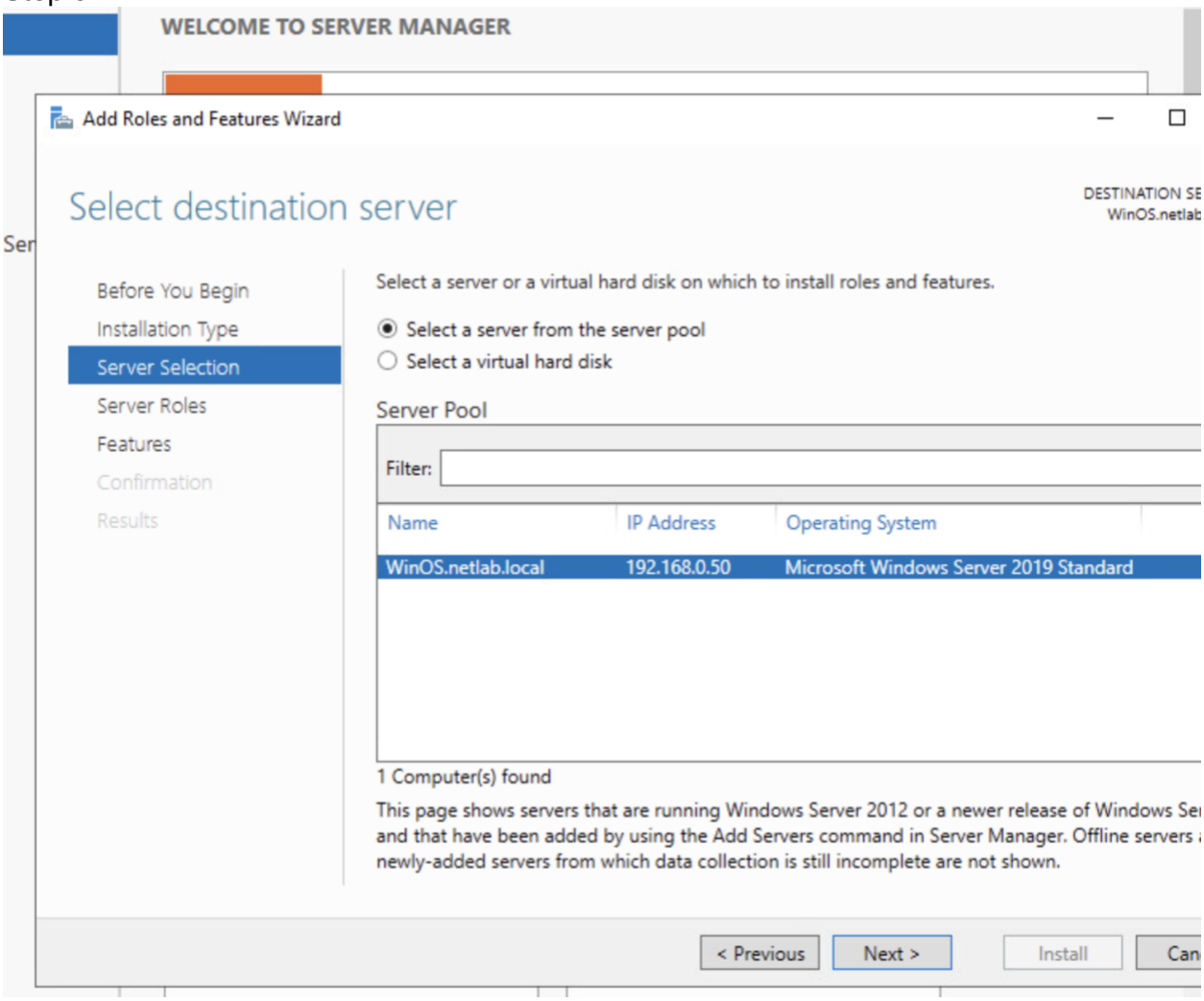
1.2:

Step 9:

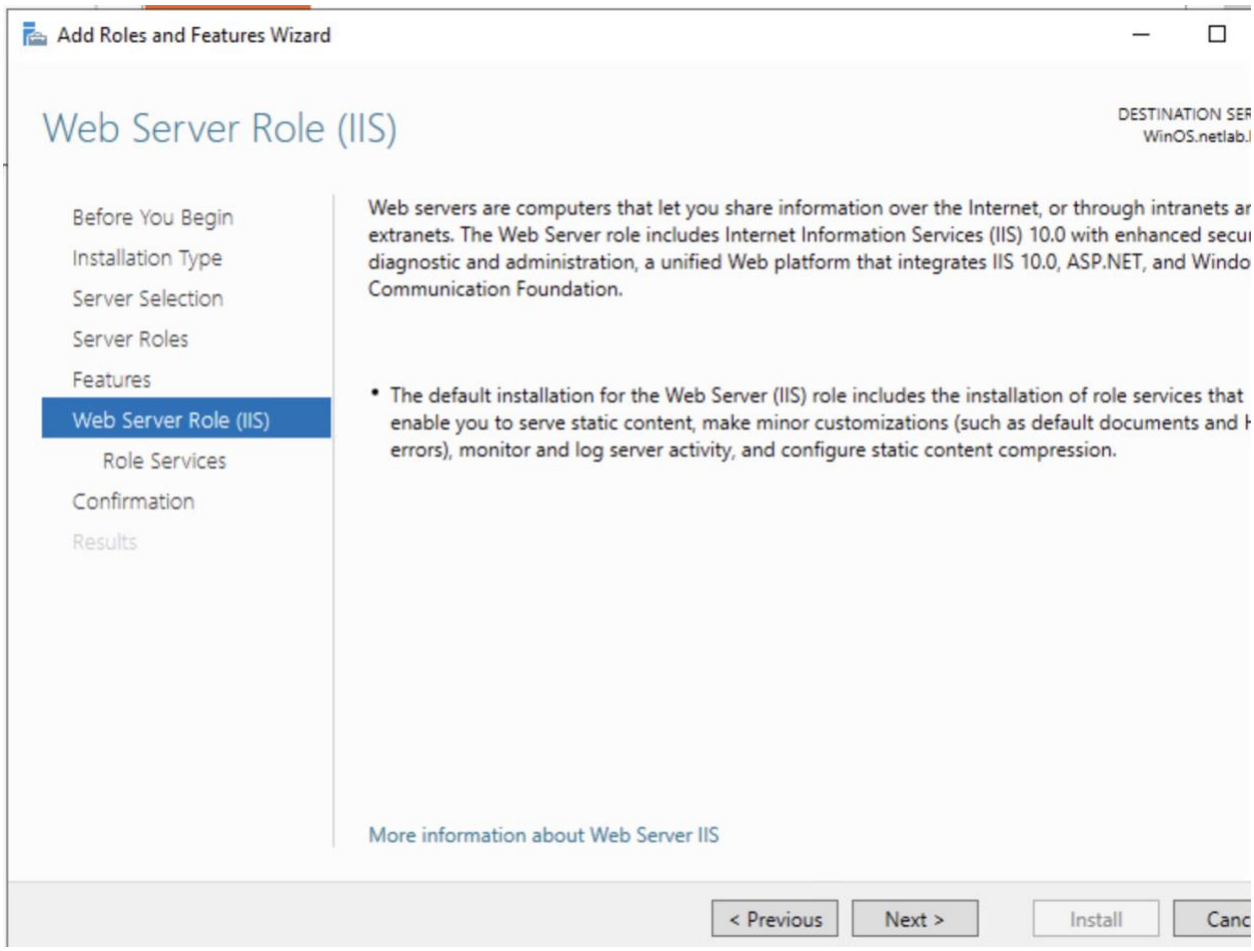
```
172.17.0.5          ether   02:42:ac:11:00:05   C                     docker0
192.168.0.50        ether   00:50:56:92:68:50   C                     ens160
172.17.0.29         ether   02:42:ac:11:00:1d   C                     docker0
172.17.0.27         ether   02:42:ac:11:00:1b   C                     docker0
172.17.0.25         ether   02:42:ac:11:00:19   C                     docker0
192.168.0.1         ether   00:50:56:92:68:01   C                     ens160
172.17.0.14         ether   02:42:ac:11:00:0e   C                     docker0
172.17.0.21         ether   02:42:ac:11:00:15   C                     docker0
172.17.0.12         ether   02:42:ac:11:00:0c   C                     docker0
```

2.1.1:

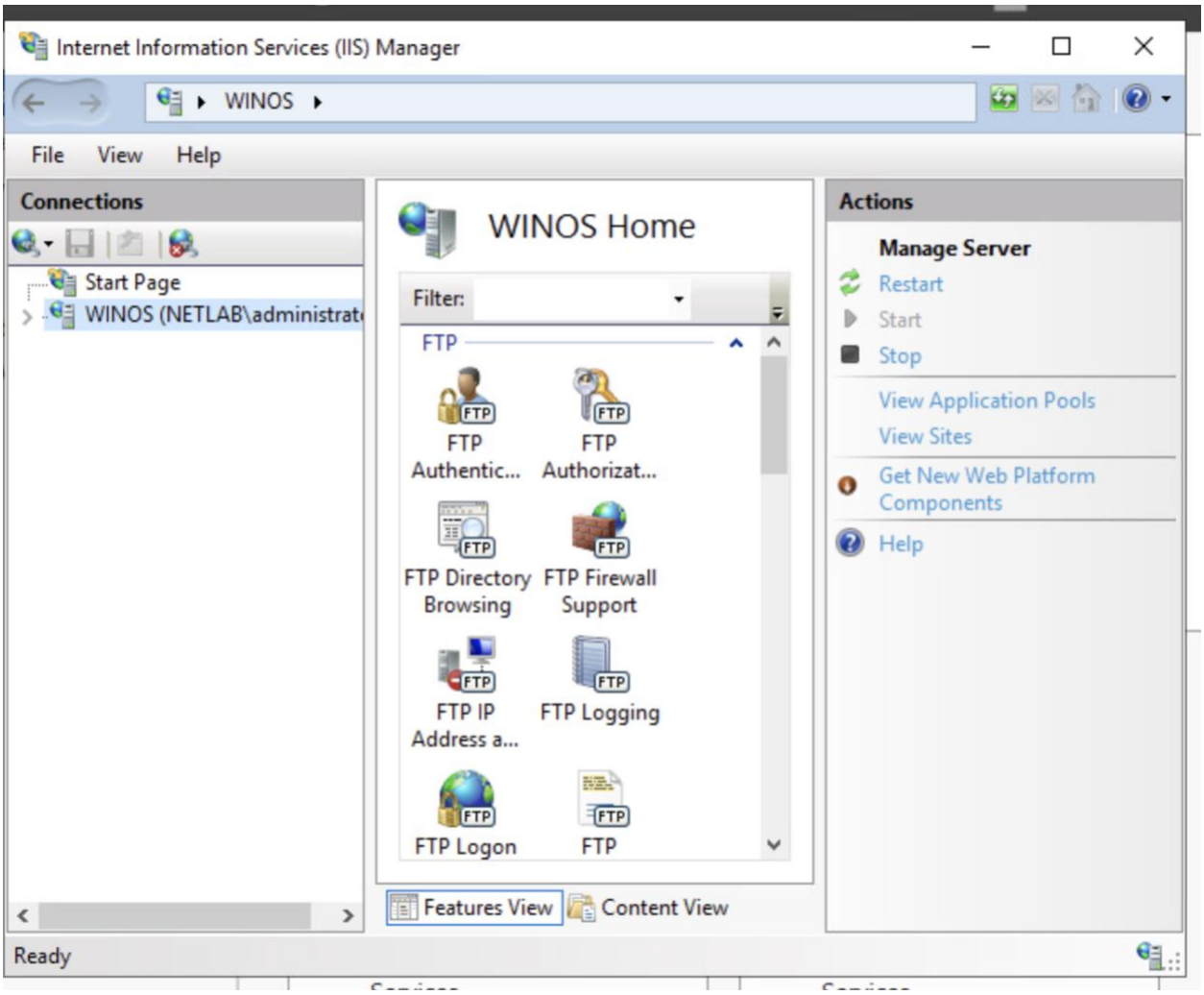
Step 6:



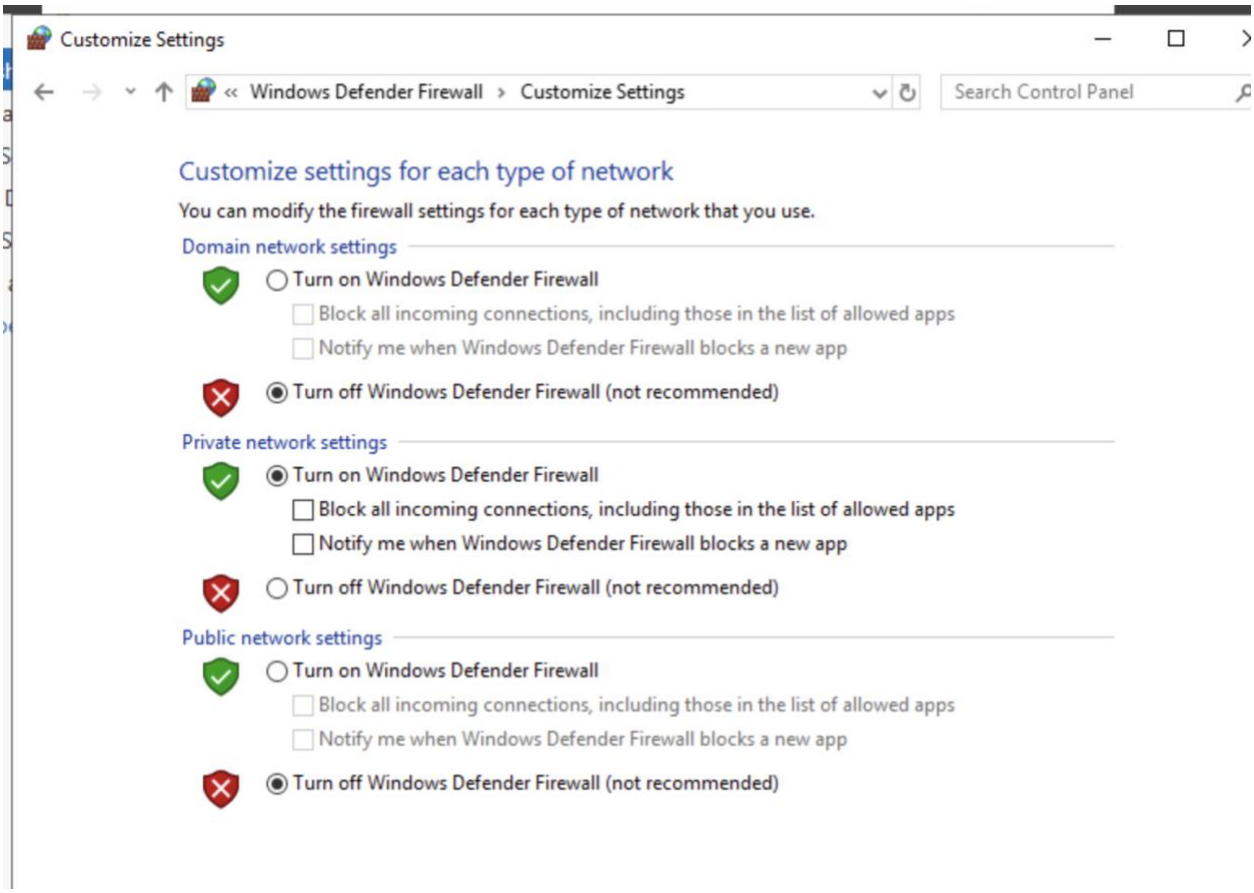
Step 11:


Add Roles and Features Wizard
Web Server Role (IIS)
DESTINATION SER
WinOS.netlab.
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Web Server Role (IIS)
Role Services
Confirmation
Results
Web servers are computers that let you share information over the Internet, or through intranets ar
extranets. The Web Server role includes Internet Information Services (IIS) 10.0 with enhanced secu
diagnostic and administration, a unified Web platform that integrates IIS 10.0, ASP.NET, and Windo
Communication Foundation.
The default installation for the Web Server (IIS) role includes the installation of role services that
enable you to serve static content, make minor customizations (such as default documents and
errors), monitor and log server activity, and configure static content compression.
More information about Web Server IIS
< Previous
Next >
Install
Canc

Step 16:


Internet Information Services (IIS) Manager
WINOS
File
View
Help
Connections
Start Page
WINOS (NETLAB\administrat
WINOS Home
Filter:
FTP
FTP Authentic...
FTP Authorizat...
FTP Directory Browsing
FTP Firewall Support
FTP IP Address a...
FTP Logging
FTP Logon
FTP
Features View
Content View
Actions
Manage Server
Restart
Start
Stop
View Application Pools
View Sites
Get New Web Platform Components
Help
Ready

Step 26:



2.1.2:

Step 2:

```
rtt min/avg/max/mdev = 0.392/0.631/1.048/0.108 ms

┌──(kali㉿kali)-[~]
└─$ ftp 192.168.0.50
Connected to 192.168.0.50.
220 Microsoft FTP Service
Name (192.168.0.50:kali): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp>
```

Step 6:

No.	Time	Source	Destination	Protoc	Lengt	Info
44	15.04666192	192.168.0.50	192.168.0.1	FTP	81	Response: 220 M
88	28.90746307	192.168.0.1	192.168.0.50	FTP	70	Request: USER a
89	28.90775021	192.168.0.50	192.168.0.1	FTP	126	Response: 331 A
103	32.90944293	192.168.0.1	192.168.0.50	FTP	70	Request: PASS a
104	32.91183842	192.168.0.50	192.168.0.1	FTP	75	Response: 230 U
106	32.91210461	192.168.0.1	192.168.0.50	FTP	60	Request: SYST
107	32.91226598	192.168.0.50	192.168.0.1	FTP	70	Response: 215 W
190	60.25142544	192.168.0.1	192.168.0.50	FTP	60	Request: QUIT
191	60.25181985	192.168.0.50	192.168.0.1	FTP	68	Response: 221 G

2.2:

Step 4:

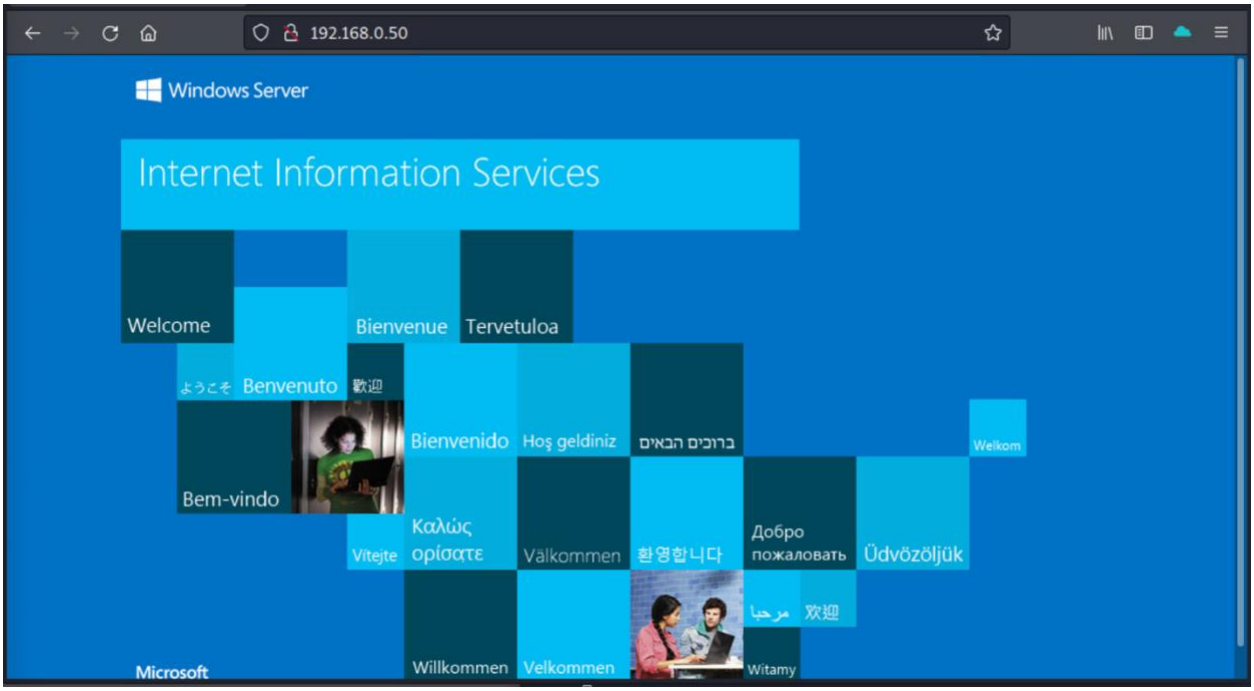
```
┌──(kali㉿kali)-[~]
└─$ sftp sysadmin@172.16.1.10
The authenticity of host '172.16.1.10 (172.16.1.10)' can't be established.
ECDSA key fingerprint is SHA256:Q/tBtXJLxJyOgvr6JheGkrFVSAUoEYYubMgwCPGDhW0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.1.10' (ECDSA) to the list of known hosts.
sysadmin@172.16.1.10's password:
Connected to 172.16.1.10.
sftp>
```

Step 8:

```
sysadmin@seconion:~
File  Edit  View  Search  Terminal  Help
172.17.0.13          ether   02:42:ac:11:00:0d   C                     docker0
172.17.0.16          ether   02:42:ac:11:00:10   C                     docker0
172.17.0.11          ether   02:42:ac:11:00:0b   C                     docker0
172.17.0.9           ether   02:42:ac:11:00:09   C                     docker0
172.17.0.7           ether   02:42:ac:11:00:07   C                     docker0
172.17.0.5           ether   02:42:ac:11:00:05   C                     docker0
192.168.0.50         ether   00:50:56:92:68:50   C                     ens160
172.17.0.29          ether   02:42:ac:11:00:1d   C                     docker0
172.17.0.27          ether   02:42:ac:11:00:1b   C                     docker0
172.17.0.25          ether   02:42:ac:11:00:19   C                     docker0
192.168.0.1          ether   00:50:56:92:68:01   C                     ens160
172.17.0.14          ether   02:42:ac:11:00:0e   C                     docker0
172.17.0.21          ether   02:42:ac:11:00:15   C                     docker0
172.17.0.12          ether   02:42:ac:11:00:0c   C                     docker0
172.17.0.19          ether   02:42:ac:11:00:13   C                     docker0
172.17.0.10          ether   02:42:ac:11:00:0a   C                     docker0
172.17.0.17          ether   02:42:ac:11:00:11   C                     docker0
172.17.0.6           ether   02:42:ac:11:00:06   C                     docker0
172.17.0.4           ether   02:42:ac:11:00:04   C                     docker0
172.17.0.30          ether   02:42:ac:11:00:1e   C                     docker0
172.17.0.28          ether   02:42:ac:11:00:1c   C                     docker0
[sysadmin@seconion ~]$ sudo wireshark
[sudo] password for sysadmin:
[sysadmin@seconion ~]$
```

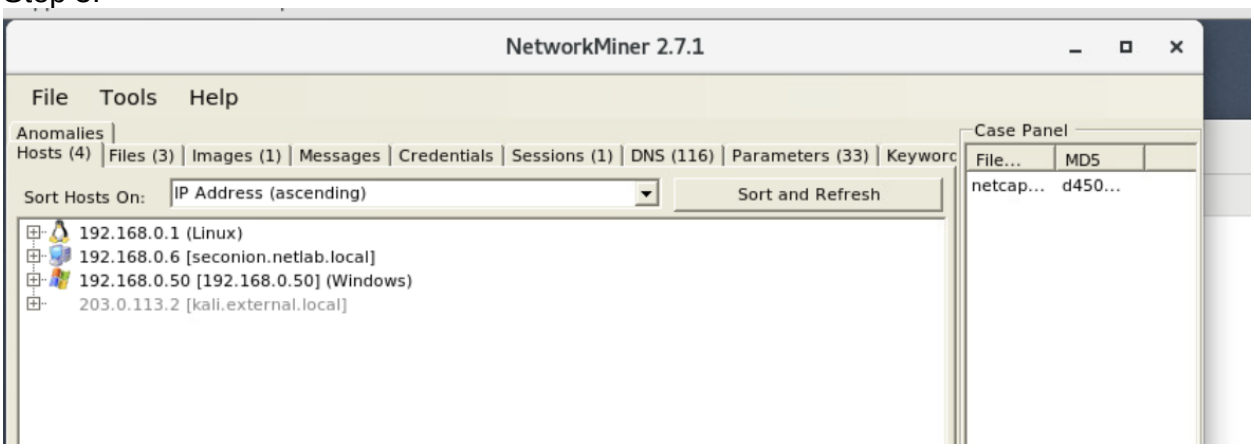
3.1:

Step 2:

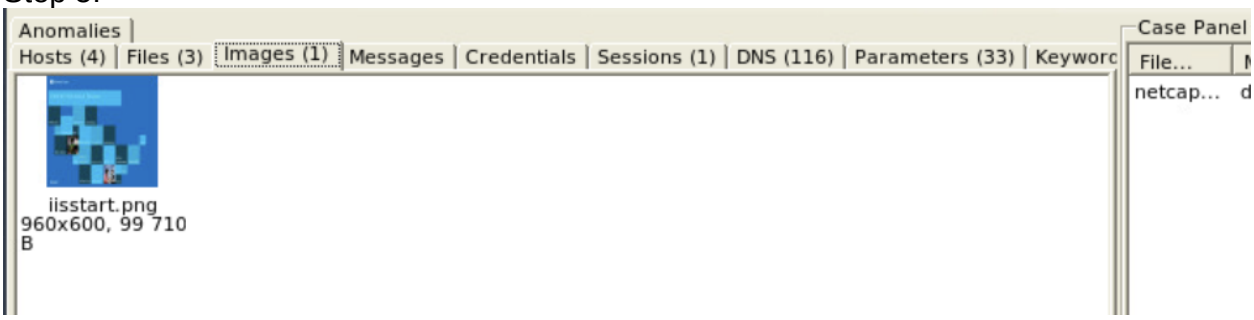


3.2:

Step 3:



Step 5:



Commentary:

In this lab many different things to capture internet traffic and parse it for good information. We setup an FTP server and examined the packets when a user connects, and with SFTP as well. I learned that learning to monitor network traffic is vital to knowing what's going on and who's doing what. Knowing this information, companies should monitor their own networks and train employees on how to use tools like Wireshark.