CSCI 4174 Assignment 2

Zhishen Yang B00601430

1.

(a) Prevent traffic from workstation 20.163 from reaching the workstation 70.5 and the tower box 70.2. Traffic from all other hosts/networks should be allowed.

To bind following ACLs on router Calgary interface E0

access-list: 1 deny 172.16.20.163 0.0.0.0

access-list: 1 permit 172.16.0.0 0.0.255.255

Interface E0

IP access-group 1 out

(b) Prevent traffic from 80.0 network from reaching 10.0 network. All other traffic must be allowed.

To bind following ACLs on router Edmonton interface E0

Access-list: 1 deny 172.16.80.0 0.0.0.255

Access-list: 1 permit 0.0.0.0 255.255.255.255

Interface E0

Ip access-group 1 out

(c) Workstations 50.75 and 50.7 should not be allowed web access on tower box 70.2. All other workstations can.

To bind following ACLs on router Calgary E0

Access-list: 101 deny tcp 172.16.50.75 0.0.0.0 172.16.70.2 0.0.0.0 eq 80

Access-list: 101 deny tcp 172.16.50.7 0.0.0.0 172.16.70.2 0.0.0.0 eq 80

Access-list: 101 permit tcp 172.16.0.0 0.0.255.255 172.16.70.2 0.0.0.0 eq 80

Interface E0

Ip access-group 101 out

(d) 80.16 can telnet to 40.89. No one else from 80.0 can telnet to 40.89. Any other host from any other subnet can telnet to 40.89.

To bind following ACLs on router Red Deer E0

Access-list: 101 permit tcp 172.16.80.16 0.0.0.0 172.16.40.89 0.0.0.0 eq 23

Access-list:101 deny tcp 172.16.80.0 0.0.0.255 172.16.40.89 0.0.0.0 eq 23

Access-list: 101 permit tcp 172.16.0.0 0.0.255.255 172.16.40.89 0.0.0.0 eq 23

Interface E0

Ip access-group 101 out

(e) 70.5 can ftp to the Edmonton router. No other host can.

To bind following ACLs on the router Edmonton router interface E1,E0,s0

Access-list: 101 permit tcp 172.16.70.5 0.0.0.0 172.16.30.1 0.0.0.0 eq 20 21

Access-list: 101 deny tcp 172.16.0.0 0.0.255.255 172.16.30.1 0.0.0.0 eq 20 21

Access-list: 101 permit tcp 172.16.70.5 0.0.0.0 172.16.20.1 0.0.0.0 eq 20 21

Access-list: 101 deny tcp 172.16.0.0 0.0.255.255 172.16.20.1 0.0.0.0 eq 20 21

Access-list: 101 permit tcp 172.16.70.5 0.0.0.0 172.16.10.1 0.0.0.0 eq 20 21

Access-list: 101 deny tcp 172.16.0.0 0.0.255.255 172.16.10.1 0.0.0.0 eq 20 21

Interface s0

Ip access-group 101 in

Interface E1

Ip access-group 101 in

Interface E0

Ip access-group 101 in

2.

The program and test files are in the folder q2, the python program is q2.py

Output:

1. Example from assignment q2:

File contains ACLs: ACL1.txt

access-list 1 deny 172.16.4.13 0.0.0.0

access-list 1 deny 172.16.5.0 0.0.0.255

access-list 1 permit 172.16.3.0 0.0.0.255

interface EO

ip access-group 1 out

File contains packets: packet1.txt

172.16.4.13 172.16.3.2

172.16.3.3 172.16.3.1

172.16.5.2 172.16.3.4

```
Please input the name of the file that contains ACL
ACL1.txt
Please input the name of hte file that contains packets
packet1.txt
['172.16.4.13', '172.16.3.2']   denied

['172.16.3.3', '172.16.3.1']   accepted

['172.16.5.2', '172.16.3.4']   denied
```

2. Example from handouts:

File contains ACLs: ACL2.txt

access-list 1 permit 172.16.0.0 0.0.255.255

interface E0

ip access-group 1 out

File contains packets: packet2.txt

172.16.0.1 172.16.3.0

182.12.12.12 172.16.4.13

172.12.0.1 172.12.3.3

145.12.12.12 172.16.4.13

```
Please input the name of the file that contains ACL
ACL2.txt
Please input the name of hte file that contains packets
packet2.txt
['172.16.0.1', '172.16.3.0']   accepted

['182.12.12.12', '172.16.4.13']  denied

['172.12.0.1', '172.12.3.3']  denied

['145.12.12.12', '172.16.4.13']  denied
```

File contains ACLs: ACL3.txt

access-list 1 deny 172.16.4.13 0.0.0.0

access-list 1 permit 172.16.0.0 0.0.255.255

interface EO

ip access-group 1 out

File contains packets: packet3.txt

172.16.4.13 172.16.3.1

172.16.0.123 172.16.3.2

172.12.12.12 172.16.3.2

```
Please input the name of the file that contains ACL
ACL3.txt
Please input the name of hte file that contains packets
packet3.txt
['172.16.4.13', '172.16.3.1']   denied

['172.16.0.123', '172.16.3.2']   accepted

['172.12.12.12', '172.16.3.2']  denied
```

File contains ACLs: ACL4.txt

access-list 1 deny 172.16.4.0 0.0.0.255

access-list 1 permit 172.16.0.0 0.0.255.255

interface EO

ip access-group 1 out

File contains packets: packet4.txt

172.16.4.13 172.16.3.1

172.16.4.245 172.16.3.1

172.16.0.123 172.16.3.2

172.12.12.12 172.16.3.2

172.16.123.13 172.16.3.1

```
Please input the name of the file that contains ACL
ACL4.txt
Please input the name of hte file that contains packets
packet4.txt
['172.16.4.13', '172.16.3.1']   denied

['172.16.4.245', '172.16.3.1']   denied

['172.16.0.123', '172.16.3.2']   accepted

['172.12.12.12', '172.16.3.2']  denied

['172.16.123.13', '172.16.3.1']   accepted
```

Extended ACLs output:

File contains extended ACLs: ACL5.txt

access-list 101 deny tcp 172.16.0.0 0.0.255.255 172.16.3.0 0.0.0.255 eq 20 21

access-list 101 permit ip 172.16.0.0 0.0.255.255 172.16.3.0 0.0.0.255

interface EO

ip access-group 101 out

File contains packets: packet5.txt

172.16.2.1 51000 172.16.3.0 20

172.16.4.13 51000 172.16.3.0 21

172.16.4.13 52000 172.16.3.0 80

172.16.0.12 52000 172.16.3.0 25

179.12.123.2 51000 172.16.3.0 51

```
Please input the name of the file that contains ACL
ACL5.txt
Please input the name of hte file that contains packets
packet5.txt
['172.16.2.1', '51000', '172.16.3.0', '20']   denied

['172.16.4.13', '51000', '172.16.3.0', '21']   denied

['172.16.4.13', '52000', '172.16.3.0', '80']   accepted

['172.16.0.12', '52000', '172.16.3.0', '25']   accepted

['179.12.123.2', '51000', '172.16.3.0', '51']  denied
```

File contains ACLs: ACL6.txt

access-list 101 deny tcp 172.16.4.13 0.0.0.0 172.16.3.0 0.0.0.255 eq 22

access-list 101 permit ip 172.16.0.0 0.0.255.255 172.16.3.0 0.0.0.255

interface EO

ip access-group 101 out

File contains packets: packet6.txt

172.16.2.1 51000 172.16.3.0 20

172.16.4.13 51000 172.16.3.0 22

172.16.4.13 52000 172.16.3.0 80

172.16.0.12 52000 172.16.3.0 25

179.12.123.2 51000 172.16.3.0 51

```
Please input the name of the file that contains ACL
ACL6.txt
Please input the name of hte file that contains packets
packet6.txt
['172.16.2.1', '51000', '172.16.3.0', '20']   accepted

['172.16.4.13', '51000', '172.16.3.0', '22']   denied

['172.16.4.13', '52000', '172.16.3.0', '80']   accepted

['172.16.0.12', '52000', '172.16.3.0', '25']   accepted

['179.12.123.2', '51000', '172.16.3.0', '51']  denied
```

3.

Product name: Cyberoam firewall

The new trends of the firewall design can be concluded as 1. A design can deal with the increasing number of users/devices, which enhances the identity-based security. 2. A design to increase the application control and visibility 3. A design for intrusion prevention 4. A design for firewall data reporting or visualization. 5. A design to protect remote accessing 6. A design for web application filtering

The Cyberoam firewall resides in firmware which is called Cyberoam OS. Meanwhile, the features of the Cyberoam firewall provides enhanced security to network users and meet some new trends of the firewall design.

1. Action-based intelligence and control (new design trend 1)

This feature deals with the increasing number of users. The Cyberoam firewall uses the layer 8 technology to realize this feature. This technology provides identity-based security over authentication, authorization, auditing. It has the 8^{th} layer in the network stack, which likes an abstract network layer that binds other layers. This 8^{th} layer provides better control and security over other network layers. This technology allows the network administrator to control users' activities and uniquely identify users.

2. Application control and visibility (new design trend 2)

This feature enhances the application control and visibility. It provides the application control and visibility that can set prioritization of the applications by using user identity, application, time and bandwidth. This feature also can classify risk level of the application based on the information from the database, such as characteristic, technology, etc.

3. Intrusion prevention (new design trend 3)

The Cyberoam firewall has intrusion prevention system to prevent application-level attacks, Dos, and DDoS attacks, intrusions, malware, Trojan, malicious code, etc.

4. On-appliance reporting (new design trend 4)

This feature provides more than 1200 in-depth reports to the user. These reports provide real-time visibility of user and network activities. These reports also include animated and dynamic reports, graphs to help the user to quickly understand activities from the firewall.

5. Virtual private network (new design trend 5)

The Cyberoam firewall provides the VPN that provides user secure remote access. This service ensures the security of remote access by using threat-free Tunneling technology.

6. Web filtering (new design trend 6)

This feature blocks access to harmful websites. The Cyberoam firewall has databases of millions of URLs. It blocks web access with potential risk by using this database.

7. Web application firewall (new design trend 6)

The Cyberoam firewall provides on-appliance web application firewall subscription. This feature prevents web application from attacks such as session hijacking.

8. Bandwidth management

The Cyberoam firewall also allows the administrator to manage the bandwidth for users and applications. To ensure business-critical users and applications have enough bandwidth on the network, to enhance the quality of service.

Reference: http://www.cyberoam.com/microsite/next-generation-firewall

4.

All programs and sample test files are in folder q4

(a)Digraphic substitution cipher

Name of python program for encryption in folder q4: q4.py

Name of the python for decryption in folder q4: q4-2.py

Example 1:

1. Encryption

Use q4.py

File that contains plaintext: text1.txt

COME QUICKLY WE NEED HELP

File that contains key: key1.txt

LOGARITHM

```
Please input the file that contains key
key1.txt
Please input the file that contains plain text
text1.txt
Please input the name of output file that stors encrypted text
output1.txt
```

File that contains encrypted text: output1.txt

DLHFSNCNCRZXCQQGFEEQON

2. Decryption

Use q4-2.py

File that contains key: key1.txt

LOGARITHM

File that contains encrypted text: output1.txt

DLHFSNCNCRZXCQQGFEEQON

```
Please input the file that contains key
key1.txt
Please input the file that contains encrpyted text
output1.txt
Please input name of the output file that contains decrypted text
doutput1.txt
>>>
```

File that contains decrypted text: doutput1.txt

COMEQUICKLYWENEEDHELP

Example 2:

1. Encryption

Use q4.py

File that contains plaintext: text2.txt

THE MEETING IS AT TREFFORESTS

File that contains key: key2.txt

PUZZLE

```
Please input the file that contains key
key2.txt
Please input the file that contains plain text
text2.txt
Please input the name of output file that stors encrypted text
output2.txt
```

File that contains encrypted text: output2.txt

VGFSLYPYGQHKNFVYXNFMYDQSFYYN

2. Decryption

Use q4-2.py

File that contains plaintext: key2.txt

PUZZLE

File that contains key: output2.txt

VGFSLYPYGQHKNFVYXNFMYDQSFYYN

```
Please input the file that contains key
key2.txt
Please input the file that contains encrpyted text
output2.txt
Please input name of the output file that contains decrypted text
doutput2.txt
```

File that contains decrypted text: doutput2.txt

THEMEETINGISATTREFFORESTS

(b) Matrix transposition cipher

Name of encryption program: q4-3.py

Name of decryption program: q4-4.py

Example 1:

1. Encryption

Use q4-3.py

File that contains plaintext: text3.txt

you guessed it!

File that contains key: key3.txt

3 4 1 2

```
Please input the file that contains key
key3.txt
Please input the file that contains plain text
text3.txt
Please input the name of output file that contains encrypted text
output3.txt
>>>
```

File that contains encrypted text: output3.txt

ued! s ygsiouet

2. Decryption

Use q4-4.py

File that contains plaintext: key3.txt

3 4 1 2

File that contains key: output3.txt

ued! s ygsiouet

```
Please input the file that contains key
key3.txt
Please input the file that contains encrypted text
output3.txt
Please input the name of output file that contains decrypted text
doutput3.txt
>>>
```

File that contains decrypted text: doutput3.txt

you guessed it!

Example 2:

1. Encryption

Use q4-3.py

File that contains plaintext: text4.txt

security!security!security!

File that contains key: key4.txt

5 4 1 2 3

```
Please input the file that contains key
key4.txt
Please input the file that contains plain text
text4.txt
Please input the name of output file that contains encrypted text
output4.txt
>>>
```

File that contains encrypted text: output4.txt

rsiet u!rsi sietcyetcyu!cyu!r

2. Decryption

Use q4-4.py

File that contains plaintext: key4.txt

5 4 1 2 3

File that contains key: output4.txt

rsiet u!rsi sietcyetcyu!cyu!r

```
Please input the file that contains key
key4.txt
Please input the file that contains encrypted text
output4.txt
Please input the name of output file that contains decrypted text
doutput4.txt
>>>
```

File that contains decrypted text: doutput4.txt

security!security!security!