# Case Study1

# Configure Firewalls

Name – Jagdeep Kainth

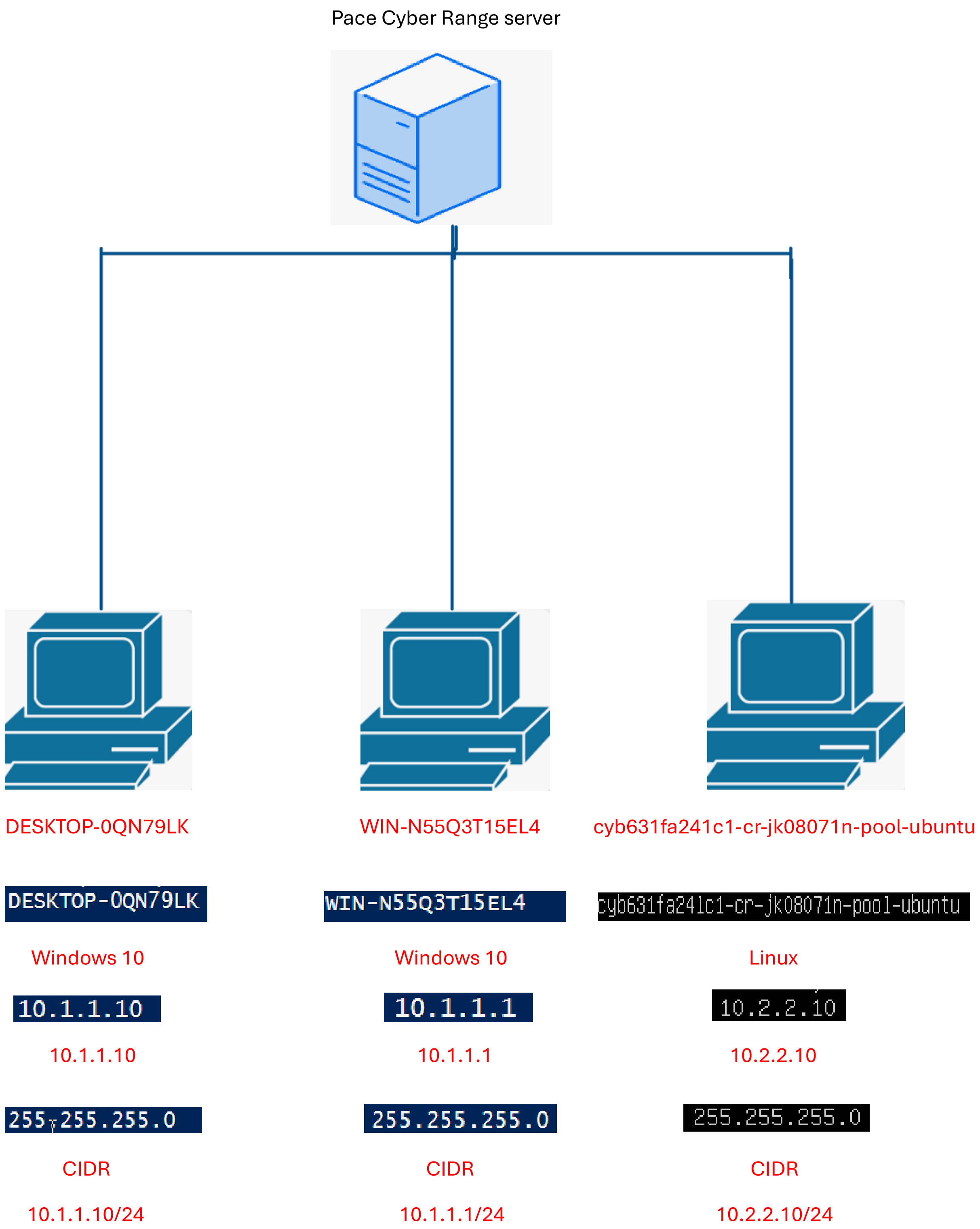
Date – October 10th, 2024

# Executive Summary

As an intern with Wonderville, my task is to strengthen the company digital security that will eventually benefit the entire town IT infrastructure. This task will be accomplished by mainly targeting the internal windows server which is at high risk if get compromised by a threat actor. We have to focus on configuring the host-based firewall rules on the internal windows server. The present network configuration on the windows server allows all host to communicate with each other and specially to the internal windows server, which will eventually make the entire IT infrastructure of the company vulnerable to a ransomware attack. To solve this problem, I have developed an access control policy that put restrictions on the unnecessary communication between hosts while still allowing essential services, which includes administrative access. To further move on achieving our security goals for the company, I prefer enabling Host-Based Firewalls on all windows servers that will prevent unauthorized access, by blocking incoming traffic from outside. We should develop our firewall rules in such a manner that only the chosen traffic is allowed, for example only the administrative access should be permitted, thereby staying protected from the potential threats. We should deploy our rules in an automated script across all windows hosts, while keeping in in mind the consistency and ease of management. This approach will make the overall IT infrastructure and networking database of the company safe from the outside threats by minimizing unwanted exposure and more focus on critical systems and services.

# Basic Task

# Network Topology Figure

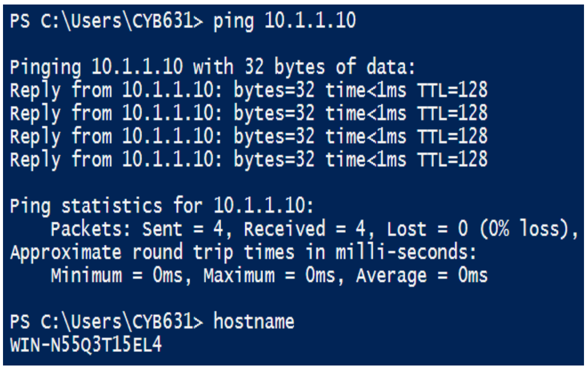


Pace Cyber Range server

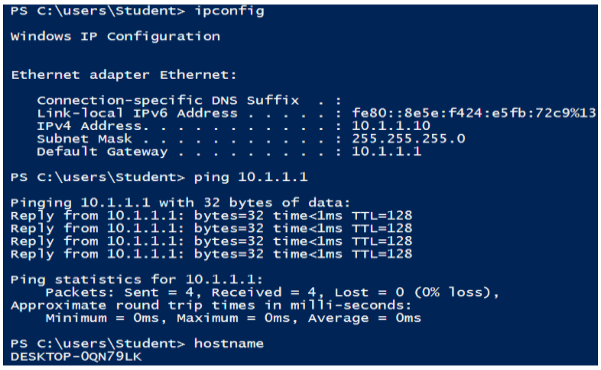
| | | |
|---|---|---|
| DESKTOP-0QN79LK | WIN-N55Q3T15EL4 | cyb631fa241c1-cr-jk08071n-pool-ubuntu |
| DESKTOP-0QN79LK | WIN-N55Q3T15EL4 | cyb631fa241c1-cr-jk08071n-pool-ubuntu |
| Windows 10 | Windows 10 | Linux |
| 10.1.1.10 | 10.1.1.1 | 10.2.2.10 |
| 10.1.1.10 | 10.1.1.1 | 10.2.2.10 |
| 255.255.255.0 | 255.255.255.0 | 255.255.255.0 |
| CIDR | CIDR | CIDR |
| 10.1.1.10/24 | 10.1.1.1/24 | 10.2.2.10/24 |

# Advanced Task

# Technical Solutions

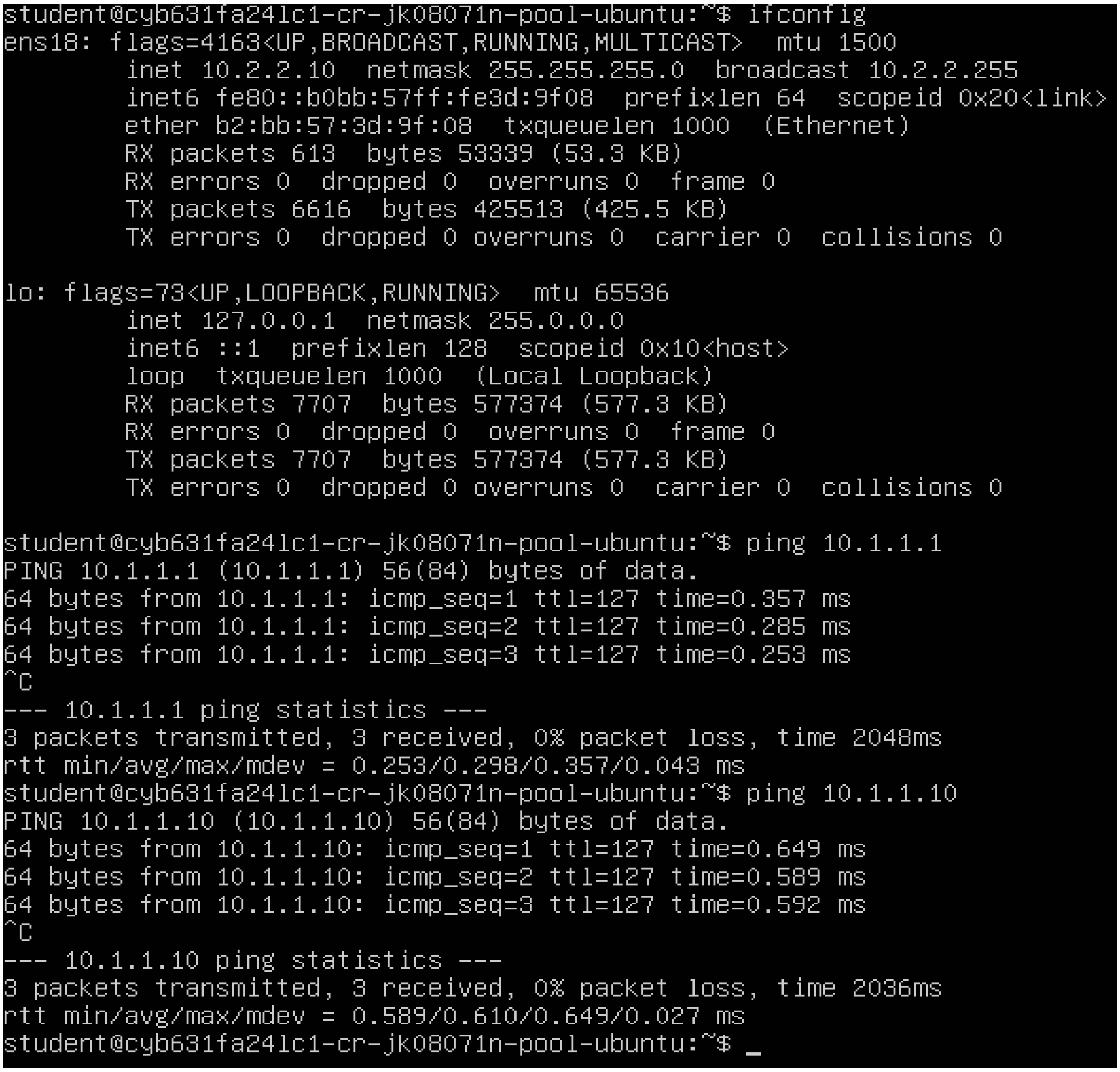


```
PS C:\Users\CYB631> ping 10.1.1.10

Pinging 10.1.1.10 with 32 bytes of data:
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128

Ping statistics for 10.1.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PS C:\Users\CYB631> hostname
WIN-N55Q3T15EL4
```



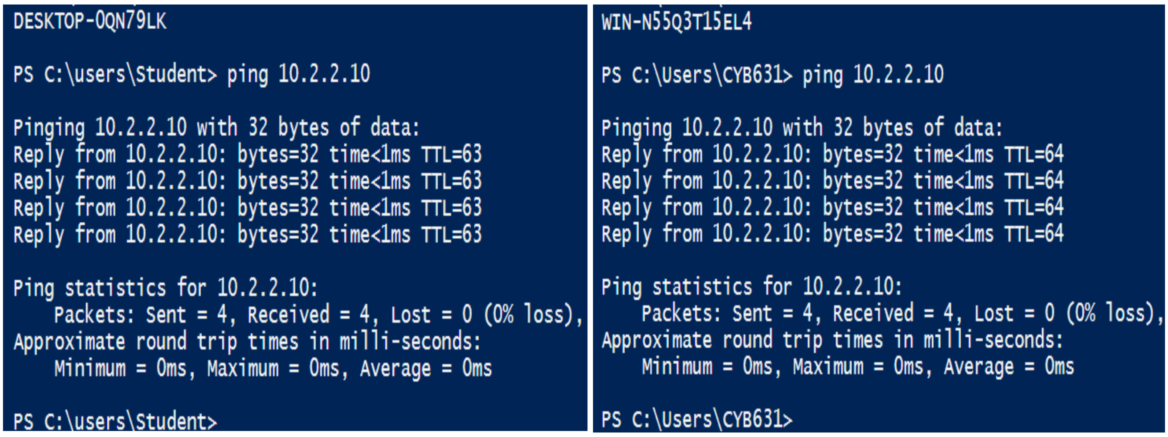
```
PS C:\users\Student> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8e5e:f424:e5fb:72c9%13
   IPv4 Address. . . . . . . . . . . : 10.1.1.10
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.1.1

PS C:\users\Student> ping 10.1.1.1

Pinging 10.1.1.1 with 32 bytes of data:
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PS C:\users\Student> hostname
DESKTOP-0QN79LK
```

**WIN-N55Q3T15EL4** can ping **DESKTOP-0QN79LK** & **DESKTOP-0QN79LK** can ping **WIN-N55Q3T15EL4**



```
student@cyb631fa241c1-cr-jk08071n-pool-ubuntu:~$ ifconfig
ens18: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.2.2.10  netmask 255.255.255.0  broadcast 10.2.2.255
        inet6 fe80::b0bb:57ff:fe3d:9f08  prefixlen 64  scopeid 0x20<link>
        ether b2:bb:57:3d:9f:08  txqueuelen 1000  (Ethernet)
        RX packets 613  bytes 53339 (53.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6616  bytes 425513 (425.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 7707  bytes 577374 (577.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7707  bytes 577374 (577.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

student@cyb631fa241c1-cr-jk08071n-pool-ubuntu:~$ ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=127 time=0.357 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=127 time=0.285 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=127 time=0.253 ms
^C
--- 10.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2048ms
rtt min/avg/max/mdev = 0.253/0.298/0.357/0.043 ms
student@cyb631fa241c1-cr-jk08071n-pool-ubuntu:~$ ping 10.1.1.10
PING 10.1.1.10 (10.1.1.10) 56(84) bytes of data.
64 bytes from 10.1.1.10: icmp_seq=1 ttl=127 time=0.649 ms
64 bytes from 10.1.1.10: icmp_seq=2 ttl=127 time=0.589 ms
64 bytes from 10.1.1.10: icmp_seq=3 ttl=127 time=0.592 ms
^C
--- 10.1.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.589/0.610/0.649/0.027 ms
student@cyb631fa241c1-cr-jk08071n-pool-ubuntu:~$ _
```

**cyb631fa241c1-cr-jk08071n-pool-ubuntu** can ping both **WIN-N55Q3T15EL4** & **DESKTOP-0QN79LK**



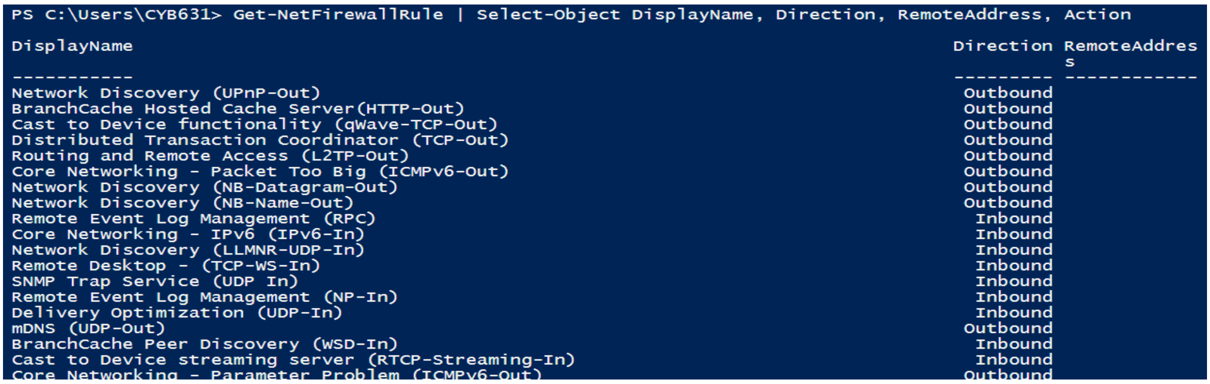
```
DESKTOP-0QN79LK

PS C:\users\Student> ping 10.2.2.10

Pinging 10.2.2.10 with 32 bytes of data:
Reply from 10.2.2.10: bytes=32 time<1ms TTL=63
Reply from 10.2.2.10: bytes=32 time<1ms TTL=63
Reply from 10.2.2.10: bytes=32 time<1ms TTL=63
Reply from 10.2.2.10: bytes=32 time<1ms TTL=63

Ping statistics for 10.2.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PS C:\users\Student>
```

```
WIN-N55Q3T15EL4

PS C:\Users\CYB631> ping 10.2.2.10

Pinging 10.2.2.10 with 32 bytes of data:
Reply from 10.2.2.10: bytes=32 time<1ms TTL=64
Reply from 10.2.2.10: bytes=32 time<1ms TTL=64
Reply from 10.2.2.10: bytes=32 time<1ms TTL=64
Reply from 10.2.2.10: bytes=32 time<1ms TTL=64

Ping statistics for 10.2.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PS C:\Users\CYB631>
```

Both **WIN-N55Q3T15EL4** & **DESKTOP-0QN79LK** can ping **cyb631fa241c1-cr-jk08071n-pool-ubuntu**

From the given screenshots above we can easily proof that the whole system is vulnerable to outside threats because all the hosts can easily ping each other and there is no firewall configured in the internal network as well as outside hosts(Linux) which can be malicious can also ping internal host.

We used the **[1]** PowerShell command to find out the current firewall rules that there was no rules configured that isolates the internal network from outside hosts.



```
PS C:\Users\CYB631> Get-NetFirewallRule | Select-Object DisplayName, Direction, RemoteAddress, Action

DisplayName                                                                Direction RemoteAddres
                                                                                     s
-----------                                                                --------- ------------
Network Discovery (UPnP-Out)                                                Outbound
BranchCache Hosted Cache Server(HTTP-Out)                                   Outbound
Cast to Device functionality (qWave-TCP-Out)                                Outbound
Distributed Transaction Coordinator (TCP-Out)                               Outbound
Routing and Remote Access (L2TP-Out)                                        Outbound
Core Networking - Packet Too Big (ICMPv6-Out)                               Outbound
Network Discovery (NB-Datagram-Out)                                         Outbound
Network Discovery (NB-Name-Out)                                             Outbound
Remote Event Log Management (RPC)                                            Inbound
Core Networking - IPv6 (IPv6-In)                                             Inbound
Network Discovery (LLMNR-UDP-In)                                             Inbound
Remote Desktop - (TCP-WS-In)                                                 Inbound
SNMP Trap Service (UDP In)                                                   Inbound
Remote Event Log Management (NP-In)                                          Inbound
Delivery Optimization (UDP-In)                                               Inbound
mDNS (UDP-Out)                                                              Outbound
BranchCache Peer Discovery (WSD-In)                                          Inbound
Cast to Device streaming server (RTCP-Streaming-In)                          Inbound
Core Networking - Parameter Problem (ICMPv6-Out)                            Outbound
```

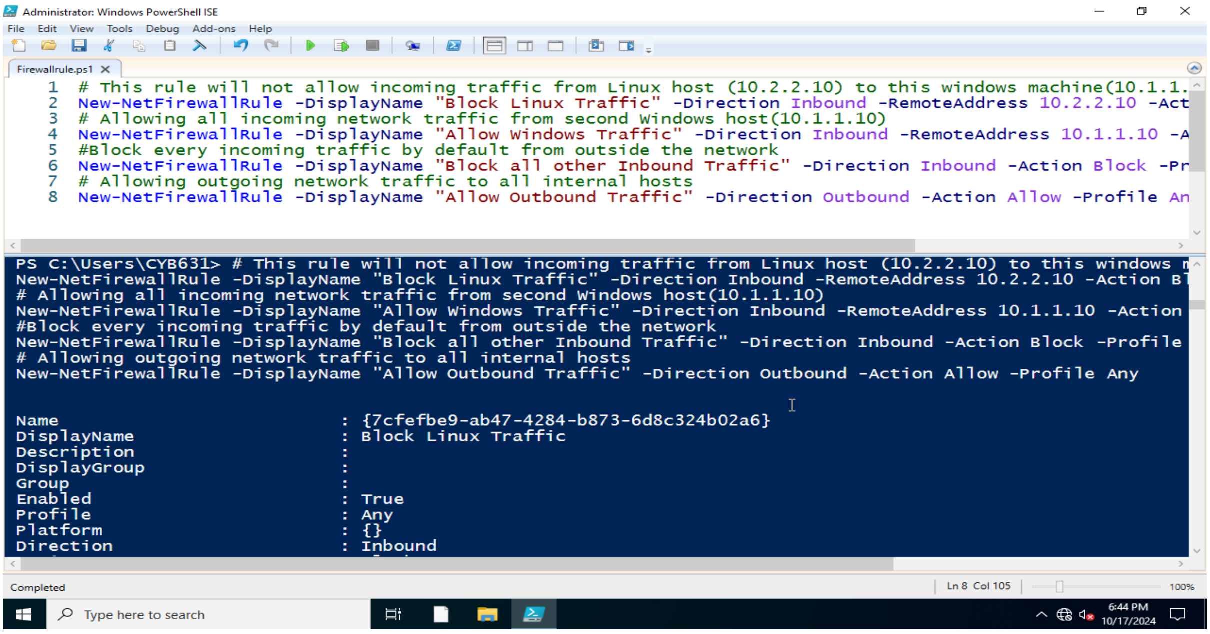
Our approach is to secure the **WIN-N55Q3T15EL4** windows 10 host which have all the database of Wonderville saved and if this host security got compromised, then the whole IT system will collapse. So in order to protect the confidentiality, integrity, availability of the data of the company ,we should block all kind of incoming traffic from outside host to the internal windows server. Also we should make sure that two internal host can reach out to each other and their connection should not be affected due to configured firewall.

# Evidence

Policy –

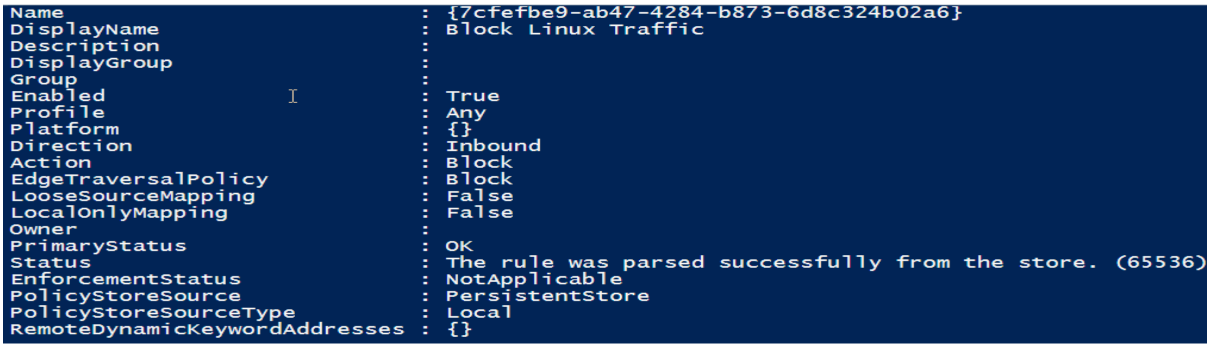
- Objective 1 – block all traffic between the Linux host **cyb631fa241c1-cr-jk08071n-pool-ubuntu** with Ip address 10.2.2.10.
- Objective 2 - The main windows host **WIN-N55Q3T15EL4** with Ip address 10.1.1.1 should not be reach by any external host.
- Objective 3 – The windows host **DESKTOP-0QN79LK** with IP address 10.1.1.10 should be able to ping the main windows host **WIN-N55Q3T15EL4** (10.1.1.1)

To implement these firewall policy rules, we will run a PowerShell script **[2]** Firewallrule.ps1 on the host **WIN-N55Q3T15EL4**

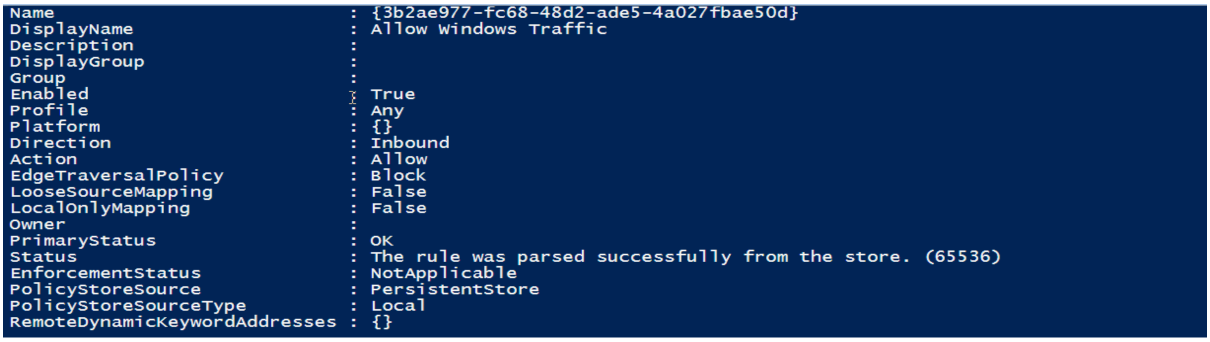


```
# This rule will not allow incoming traffic from Linux host (10.2.2.10) to this windows machine(10.1.1.
New-NetFirewallRule -DisplayName "Block Linux Traffic" -Direction Inbound -RemoteAddress 10.2.2.10 -Act
# Allowing all incoming network traffic from second Windows host(10.1.1.10)
New-NetFirewallRule -DisplayName "Allow Windows Traffic" -Direction Inbound -RemoteAddress 10.1.1.10 -A
#Block every incoming traffic by default from outside the network
New-NetFirewallRule -DisplayName "Block all other Inbound Traffic" -Direction Inbound -Action Block -Pr
# Allowing outgoing network traffic to all internal hosts
New-NetFirewallRule -DisplayName "Allow Outbound Traffic" -Direction Outbound -Action Allow -Profile An
```

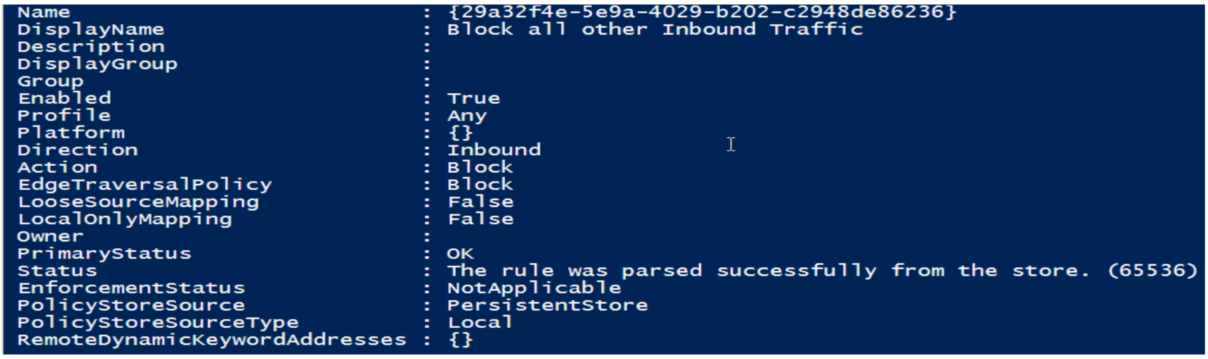
```
PS C:\Users\CYB631> # This rule will not allow incoming traffic from Linux host (10.2.2.10) to this windows m
New-NetFirewallRule -DisplayName "Block Linux Traffic" -Direction Inbound -RemoteAddress 10.2.2.10 -Action Bl
# Allowing all incoming network traffic from second Windows host(10.1.1.10)
New-NetFirewallRule -DisplayName "Allow Windows Traffic" -Direction Inbound -RemoteAddress 10.1.1.10 -Action
#Block every incoming traffic by default from outside the network
New-NetFirewallRule -DisplayName "Block all other Inbound Traffic" -Direction Inbound -Action Block -Profile
# Allowing outgoing network traffic to all internal hosts
New-NetFirewallRule -DisplayName "Allow Outbound Traffic" -Direction Outbound -Action Allow -Profile Any


Name                          : {7cfefbe9-ab47-4284-b873-6d8c324b02a6}
DisplayName                   : Block Linux Traffic
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
```



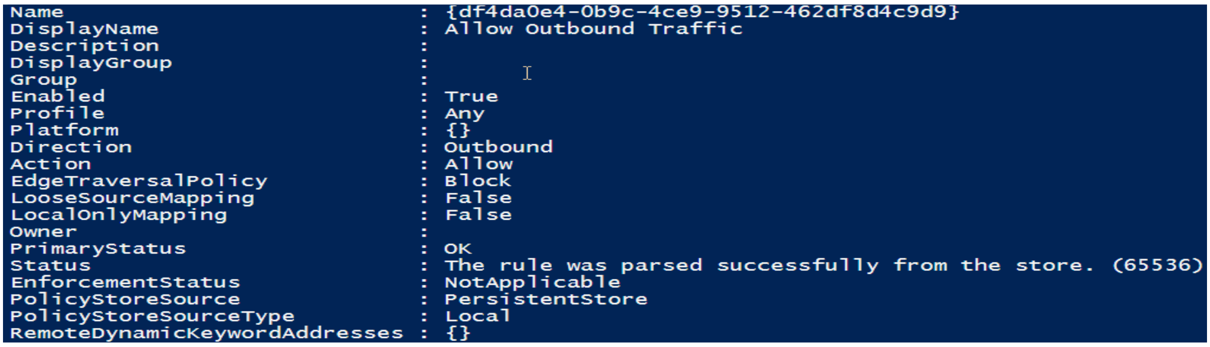
```
Name                            : {7cfefbe9-ab47-4284-b873-6d8c324b02a6}
DisplayName                     : Block Linux Traffic
Description                     :
DisplayGroup                    :
Group                           :
Enabled                         : True
Profile                         : Any
Platform                        : {}
Direction                       : Inbound
Action                          : Block
EdgeTraversalPolicy             : Block
LooseSourceMapping              : False
LocalOnlyMapping                : False
Owner                           :
PrimaryStatus                   : OK
Status                          : The rule was parsed successfully from the store. (65536)
EnforcementStatus               : NotApplicable
PolicyStoreSource               : PersistentStore
PolicyStoreSourceType           : Local
RemoteDynamicKeywordAddresses   : {}
```



```
Name                             : {3b2ae977-fc68-48d2-ade5-4a027fbae50d}
DisplayName                      : Allow Windows Traffic
Description                      :
DisplayGroup                     :
Group                            :
Enabled                          : True
Profile                          : Any
Platform                         : {}
Direction                        : Inbound
Action                           : Allow
EdgeTraversalPolicy              : Block
LooseSourceMapping               : False
LocalOnlyMapping                 : False
Owner                            :
PrimaryStatus                    : OK
Status                           : The rule was parsed successfully from the store. (65536)
EnforcementStatus                : NotApplicable
PolicyStoreSource                : PersistentStore
PolicyStoreSourceType            : Local
RemoteDynamicKeywordAddresses    : {}
```

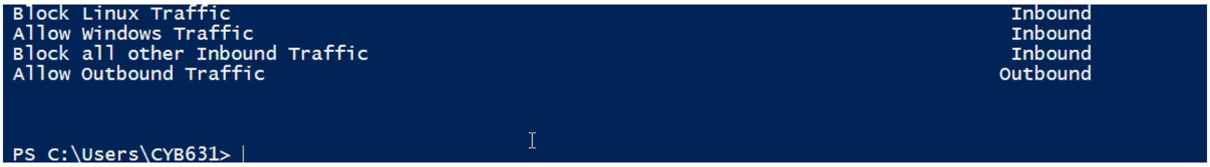


```
Name                             : {29a32f4e-5e9a-4029-b202-c2948de86236}
DisplayName                      : Block all other Inbound Traffic
Description                      :
DisplayGroup                     :
Group                            :
Enabled                          : True
Profile                          : Any
Platform                         : {}
Direction                        : Inbound
Action                           : Block
EdgeTraversalPolicy              : Block
LooseSourceMapping               : False
LocalOnlyMapping                 : False
Owner                            :
PrimaryStatus                    : OK
Status                           : The rule was parsed successfully from the store. (65536)
EnforcementStatus                : NotApplicable
PolicyStoreSource                : PersistentStore
PolicyStoreSourceType            : Local
RemoteDynamicKeywordAddresses    : {}
```



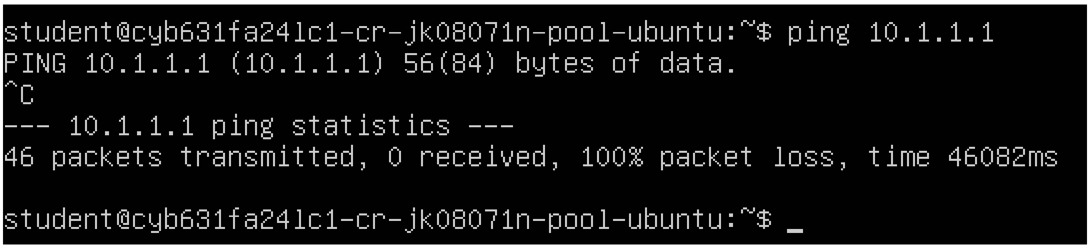
```
Name                             : {df4da0e4-0b9c-4ce9-9512-462df8d4c9d9}
DisplayName                      : Allow Outbound Traffic
Description                      :
DisplayGroup                     :
Group                            :
Enabled                          : True
Profile                          : Any
Platform                         : {}
Direction                        : Outbound
Action                           : Allow
EdgeTraversalPolicy              : Block
LooseSourceMapping               : False
LocalOnlyMapping                 : False
Owner                            :
PrimaryStatus                    : OK
Status                           : The rule was parsed successfully from the store. (65536)
EnforcementStatus                : NotApplicable
PolicyStoreSource                : PersistentStore
PolicyStoreSourceType            : Local
RemoteDynamicKeywordAddresses    : {}
```

Then we run PowerShell script **[1]** on host **WIN-N55Q3T15EL4** to double check if the firewall is properly configured or not.



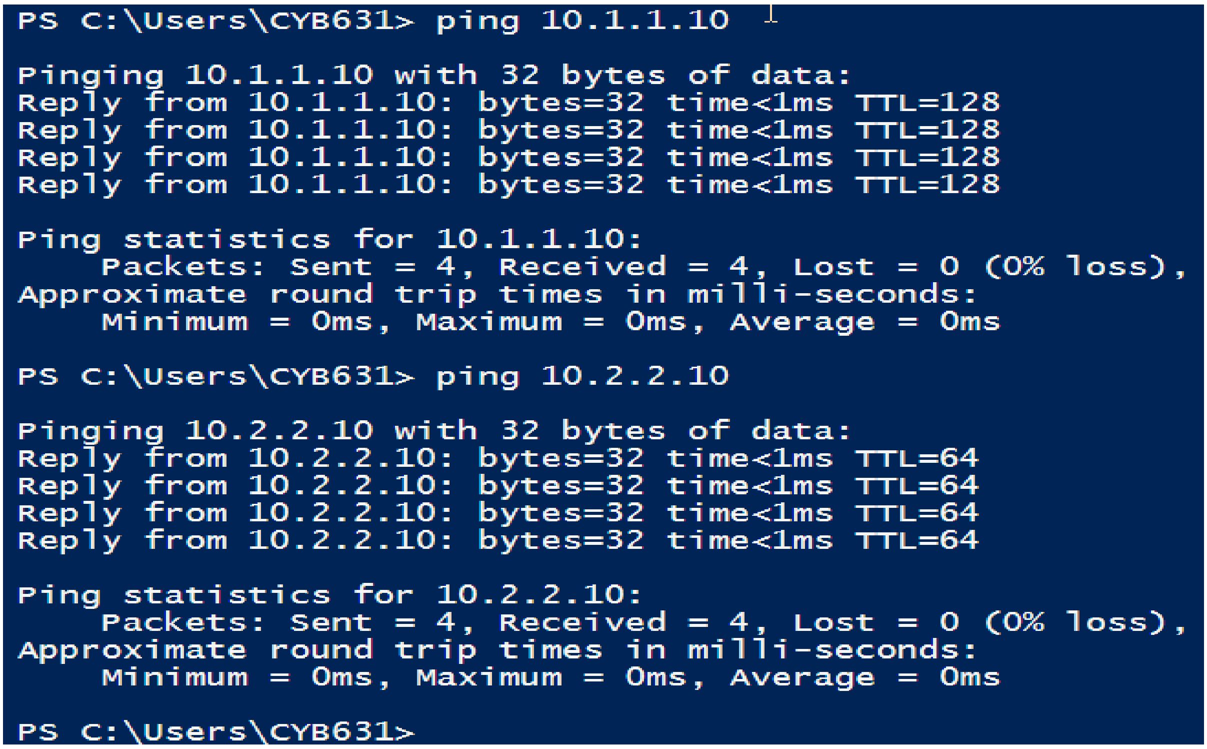
```
Block Linux Traffic                                                    Inbound
Allow Windows Traffic                                                  Inbound
Block all other Inbound Traffic                                        Inbound
Allow Outbound Traffic                                                Outbound


PS C:\Users\CYB631> |
```

Hence we can see it is properly configured, now we will test the firewall practically by Pinging the internal windows host **WIN-N55Q3T15EL4** from the Linux Host **cyb631fa241c1-cr-jk08071n-pool-ubuntu**

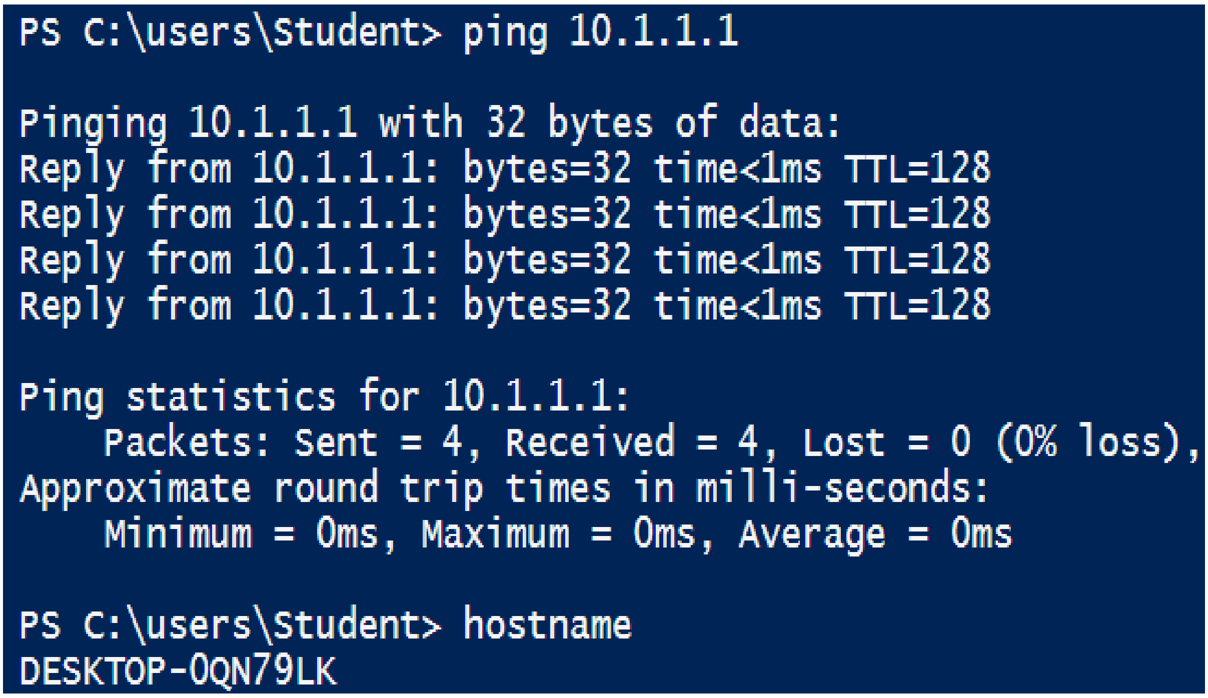


```
student@cyb631fa241c1-cr-jk08071n-pool-ubuntu:~$ ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
^C
--- 10.1.1.1 ping statistics ---
46 packets transmitted, 0 received, 100% packet loss, time 46082ms

student@cyb631fa241c1-cr-jk08071n-pool-ubuntu:~$ _
```

As we can see the Linux server has failed to ping the Windows main host, but when we try to ping the Linux host and other windows host from the main internal windows host. The ping was successful. When we try to ping the main internal host from an inside other window host between the network, ping was successful. Hence proved that our applied successfully configured firewall rules are working correctly.



```
PS C:\Users\CYB631> ping 10.1.1.10

Pinging 10.1.1.10 with 32 bytes of data:
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128
Reply from 10.1.1.10: bytes=32 time<1ms TTL=128

Ping statistics for 10.1.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PS C:\Users\CYB631> ping 10.2.2.10

Pinging 10.2.2.10 with 32 bytes of data:
Reply from 10.2.2.10: bytes=32 time<1ms TTL=64
Reply from 10.2.2.10: bytes=32 time<1ms TTL=64
Reply from 10.2.2.10: bytes=32 time<1ms TTL=64
Reply from 10.2.2.10: bytes=32 time<1ms TTL=64

Ping statistics for 10.2.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PS C:\Users\CYB631>
```



```
PS C:\users\Student> ping 10.1.1.1

Pinging 10.1.1.1 with 32 bytes of data:
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128
Reply from 10.1.1.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PS C:\users\Student> hostname
DESKTOP-0QN79LK
```

# Recommendations

The proposed approach ensures that only allowed communication occurs between the Windows hosts by implementing PowerShell-based firewall rules, thereby meeting the criteria for network security. By default, the core firewall rule blocks all incoming traffic; however, it permits traffic from trusted hosts (such as 10.1.1.10). This is a traditional method of protecting internal networks by restricting access by default and allow access under very specific guidelines. The cost-effectiveness of the solution, which makes use of already-existing technologies like PowerShell and built-in Windows firewall capabilities without the need for extra software makes it extremely practical for Wonderville's small IT department. The PowerShell scripts provide scalability as the network expands and are simple to implement across all machines. Because PowerShell is widely used and the annotated scripts are easy to maintain, very little training is needed. Granular control over network traffic, enhanced security via a default deny-all approach, and ease of maintenance are among the advantages, with log monitoring providing proof of efficacy. However, To maintain the highest level of security, it is necessary to handle issues including the possibility of human error when defining rules, controlling firewall settings for remote employees, and guaranteeing continuous network change monitoring. In summary, this firewall solution offers Wonderville's internal network a strong, affordable, and scalable security solution that strikes a compromise between protection and ease of use for the company's tiny IT staff. The community may strengthen its cybersecurity defences without significantly raising the cost or complexity of operations by expanding on the current infrastructure.

# Case reflection

While working on the case study, I assumed several things. The first thing I assumed was the Wonderville's company employees mindset of having basic knowledge of PowerShell which eventually allow me in maintaining firewall rules on a PowerShell. I also assumed that configuring the firewall policies will not affect the remote workers who use VPNs services while staying at home. And since, the moment I examined the network topology, I immediately decided to implement host to host configuration inside the internal network, as there was no external network in the topology. One thing I learned from the case study is balancing the security with usability. Proper rule prioritization is important to prevent issues related to connectivity of critical services when configuring firewalls. It would be beneficial to have more experience handling common issues like incorrectly set firewall rules or connection difficulties for future case studies.

# Attachments

**[1]**

```
Get-NetFirewallRule | Select-Object DisplayName, Direction, RemoteAddress, Action
```

**[2] # Firewallrule.ps1**

```
# This rule will not allow incoming traffic from Linux host (10.2.2.10) to this Windows machine (10.1.1.1)

New-NetFirewallRule -DisplayName "Block Linux Traffic" -Direction Inbound -RemoteAddress 10.2.2.10 -Action Block

# Allowing all incoming network traffic from the second Windows host (10.1.1.10)

New-NetFirewallRule -DisplayName "Allow Windows Traffic" -Direction Inbound -RemoteAddress 10.1.1.10 -Action Allow

# Block every incoming traffic by default from outside the network.

New-NetFirewallRule -DisplayName "Block All Other Inbound Traffic" -Direction Inbound -Action Block -Profile Any

# Allowing outgoing network traffic to all internal hosts

New-NetFirewallRule -DisplayName "Allow Outbound Traffic" -Direction Outbound -Action Allow -Profile Any
```