

CST 620 Prevention of Cyber Attack Methodologies  
Project 2 – Network Access Controls

# **Security Control Implementation Report**

Prepared By: *Abbrin Hoagland*

Version 1.0

## Table of Contents

1. Introduction .....	3
2. Windows Firewall Best Practices .....	4
3. Linux Firewall Best Practices .....	5
4. Windows Firewall Control Implementation and Testing .....	6
5. Linux Firewall Control Implementation and Testing .....	9
6. Conclusion .....	13
7. References .....	15

## 1. Introduction

In response to FICBANK's commitment to enhancing its cybersecurity posture, we present this Security Control Implementation Report. Building upon the successful implementation of remote access security measures, our focus has shifted to network access control (NAC). (Fortinet n.d) FICBANK recognizes the necessity of fortifying its sensitive servers and has entrusted our expertise to identify, implement, and test robust security controls.

The primary objective is to implement network access controls on select servers, emphasizing the utilization of both Windows Firewall and Linux IPtables. Our approach aligns with proactive cybersecurity technical control goals, aiming to establish a secure environment that safeguards against unauthorized access and potential data breaches.

The outlined tasks include the identification of best practices concerning Windows and Linux firewall implementations. Subsequently, the implementation of these best practices on designated servers is to be carried out, with a meticulous testing and evaluation phase to ensure the security posture remains uncompromised.

This report will detail the steps taken, from defining rules in IPtables to allowing specific ports and protocols, all the way to testing configurations and validating the security effectiveness. By adhering to the principles of defense-in-depth, we aim to create a robust security framework that aligns with FICBANK's security objectives.

The following sections will provide a comprehensive overview of the implemented security controls, documenting the steps taken, rationale behind each decision, and the outcomes of the testing phase. Through this report, we aim to equip FICBANK with a secure and resilient network infrastructure.

## 2. Windows Firewall Best Practices

Windows Firewall, a critical component of network access control, plays a pivotal role in fortifying the security posture of FICBANK's servers. Implementing best practices is imperative to ensure the efficacy of this security control. Below is a comprehensive list of best practices tailored to enhance the security of Windows Firewall configurations:

1. **Enable the Firewall:** Ensure that the Windows Firewall is consistently enabled on all servers. This foundational step establishes a baseline defense against unauthorized access.
2. **Regular Updates:** Keep the Windows operating system and Firewall software up-to-date with the latest security patches and updates. This practice mitigates vulnerabilities and reinforces the overall security of the system.
3. **Default Deny Rule:** Implement a default deny rule to block all incoming traffic by default. Subsequently, define explicit rules to allow only necessary and authorized traffic.
4. **Least Privilege Principle:** Adhere to the principle of least privilege when configuring firewall rules. Only essential ports and services required for the server's functionality should be allowed, minimizing the potential attack surface.
5. **Application-Specific Rules:** Create rules based on specific applications rather than relying solely on port-based rules. This approach provides a more granular level of control and ensures that only approved applications communicate through the firewall.
6. **Logging and Monitoring:** Enable firewall logging to capture relevant events and regularly review logs. Monitoring firewall activities aids in the early detection of anomalies and potential security incidents.
7. **Inbound and Outbound Rules:** Establish rules for both inbound and outbound traffic. Outbound rules are often overlooked but are crucial for preventing malicious software from communicating externally.
8. **Network Profiles:** Leverage network profiles (Public, Private, and Domain) to tailor firewall rules based on the network type. This ensures that the firewall adapts its configuration based on the security requirements of the network.

9. **Group Policy Settings:** Utilize Group Policy Objects (GPOs) to enforce consistent firewall configurations across multiple servers. This centralized management approach streamlines the administration of firewall rules.
10. **Regular Audits and Reviews:** Conduct periodic audits of firewall rules and review configurations to identify and rectify any outdated or unnecessary rules. Regular assessments contribute to the ongoing optimization of security controls. (Citrix 2023)

By adhering to these Windows Firewall best practices, FICBANK can establish a robust network access control framework, bolstering the security of its servers against a myriad of potential threats. These recommendations aim to strike a balance between stringent security measures and operational efficiency, ensuring a resilient defense posture for the organization.

### 3. Linux Firewall Best Practices

In the realm of Linux, the IPTables firewall serves as a cornerstone for network access control. Implementing robust best practices for IPTables is essential to fortify FICBANK's servers against potential security threats. The following list outlines key best practices for configuring and managing IPTables effectively:

1. **Default Deny Policy:** Establish a default deny policy for both incoming and outgoing traffic. This foundational principle ensures that no communication is allowed by default, requiring explicit rules for authorized traffic.
2. **Specific Rule Ordering:** Carefully consider the order of rules within IPTables. Rules are processed sequentially, so place more specific rules before generic ones to avoid unintended consequences.
3. **Limited Open Ports:** Only open ports that are necessary for the server's functionality. Reducing the number of open ports minimizes the attack surface and mitigates the risk of unauthorized access.
4. **Stateful Filtering:** Leverage stateful filtering capabilities to allow inbound traffic related to established connections. This ensures that only legitimate responses to outgoing connections are permitted.

5. **Service-Specific Rules:** Instead of relying solely on port numbers, create rules based on specific services and applications. This approach enhances the granularity of access control and aligns with the principle of least privilege.
6. **Regular Backups:** Regularly backup IPTables configurations to facilitate quick recovery in the event of misconfigurations or system failures. A reliable backup mechanism ensures that the firewall can be restored to a known secure state. (Barracuda)
7. **Logging Configuration:** Enable logging for IPTables to capture relevant information about allowed and denied traffic. Regularly review logs to detect and respond to potential security incidents promptly.
8. **Network Address Translation (NAT) Security:** If using NAT, implement it judiciously. Avoid unnecessary exposure of internal network structures to external entities and only translate addresses as needed.
9. **IPTables Policies:** Clearly define and review IPTables policies, including INPUT, OUTPUT, and FORWARD policies. Align these policies with organizational security requirements to maintain consistent control.
10. **Regular Audits and Testing:** \*Conduct regular audits of IPTables configurations and perform testing to ensure that the firewall behaves as expected. This proactive approach helps identify vulnerabilities and weaknesses in the network access control setup.

By incorporating these Linux IPTables best practices, FICBANK can establish a robust network access control framework for its servers. The focus on specificity, security, and regular monitoring contributes to a resilient defense against potential security threats, aligning with proactive cybersecurity objectives.

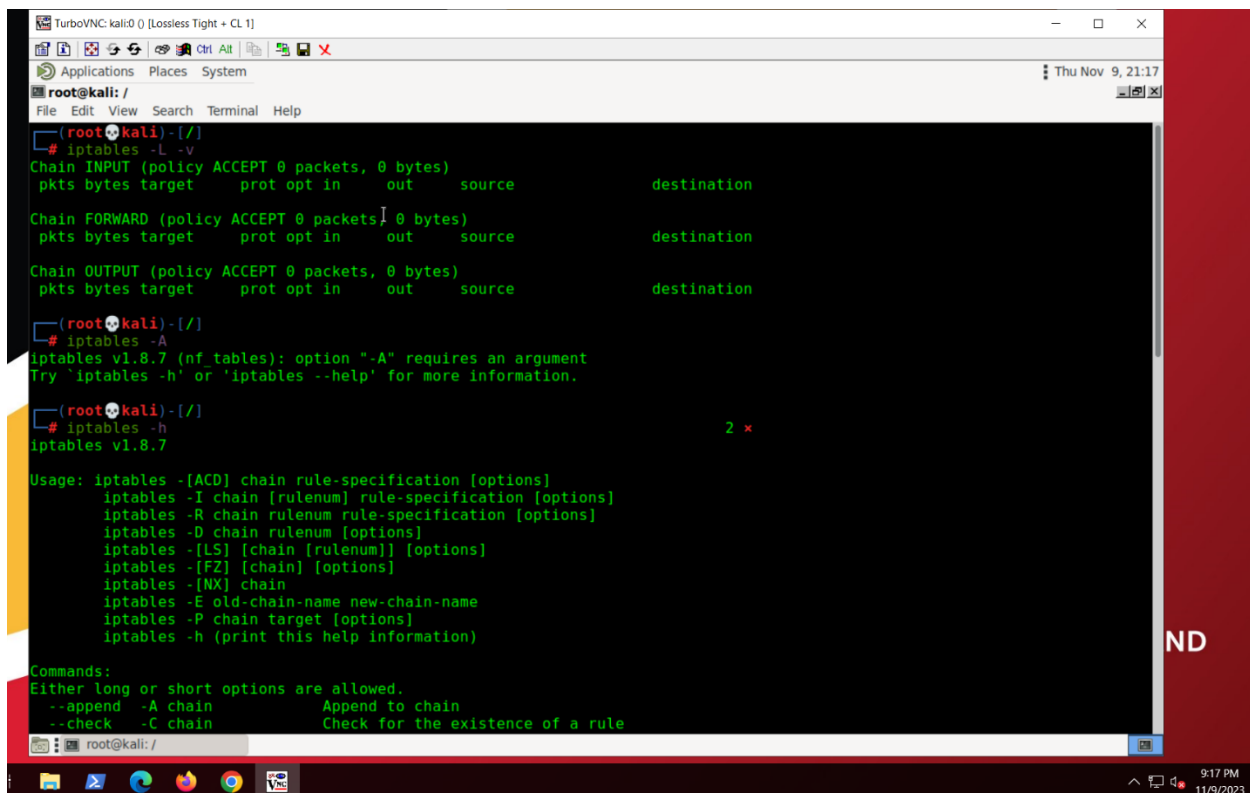
## 4. Windows Firewall Control Implementation and Testing

The implementation of Windows Firewall control best practices involved a systematic approach to fortify FICBANK's servers against potential security threats. Verification of the firewall's status was conducted using the command ``sudo iptables -L -v`` to list all rules and their detailed information.

A default deny rule was implemented to block all incoming traffic by default. Explicit rules were then defined to allow only the necessary and authorized traffic. This included creating rules for specific applications rather than relying solely on port-based rules to enhance granularity in control.

Attention was given to the principle of least privilege when configuring firewall rules. Only essential ports and services required for server functionality were allowed, minimizing the potential attack surface.

Application-specific rules were established to ensure that only approved applications could communicate through the firewall. This added layer of control is crucial for preventing unauthorized applications from accessing the server.



```
TurboVNC: kali:0 () [Lossless Tight + CL 1]
root@kali: /
File Edit View Search Terminal Help
(root@kali)~# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source      destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source      destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source      destination

(root@kali)~# iptables -A
iptables v1.8.7 (nf_tables): option "-A" requires an argument
Try 'iptables -h' or 'iptables --help' for more information.

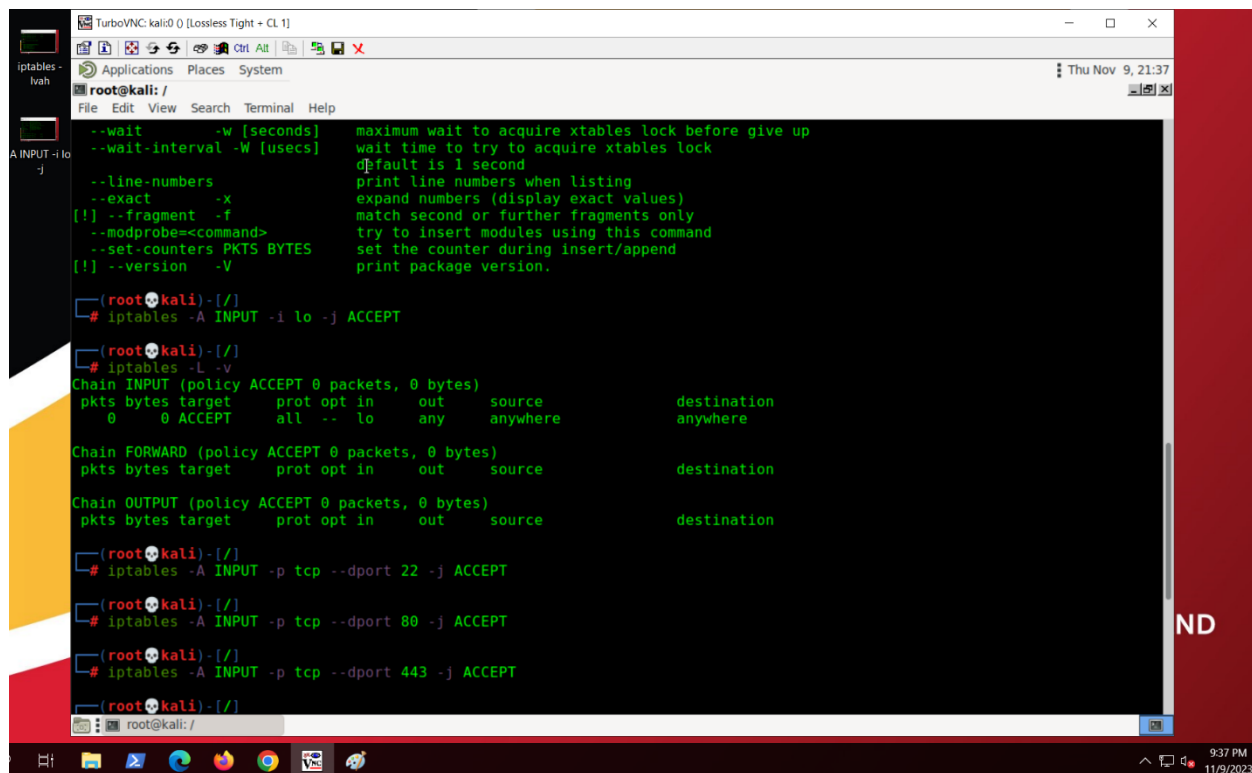
(root@kali)~# iptables -h
iptables v1.8.7
2 x

Usage: iptables -[ACD] chain rule-specification [options]
iptables -I chain [rulenum] rule-specification [options]
iptables -R chain rulenum rule-specification [options]
iptables -D chain rulenum [options]
iptables -[LS] [chain [rulenum]] [options]
iptables -[FZ] [chain] [options]
iptables -[NX] chain
iptables -E old-chain-name new-chain-name
iptables -P chain target [options]
iptables -h (print this help information)

Commands:
Either long or short options are allowed.
--append -A chain      Append to chain
--check  -C chain      Check for the existence of a rule
```

The implementation also involved regular updates to the Windows operating system and Firewall software to ensure the latest security patches and updates were applied. This practice mitigated vulnerabilities and contributed to the overall security of the system.

Network profiles, including Public, Private, and Domain, were leveraged to tailor firewall rules based on the specific security requirements of the network. This adaptive configuration ensured the firewall adjusted its settings according to the network type.



The screenshot shows a terminal window on a Kali Linux system. The user is at the root prompt. The terminal displays the help text for the iptables command, followed by several configuration commands and their output. The commands and output are as follows:

```
(root@kali:~) # iptables -h
iptables -h
--wait -w [seconds] maximum wait to acquire xtables lock before give up
--wait-interval -W [usecs] wait time to try to acquire xtables lock
                        default is 1 second
--line-numbers print line numbers when listing
--exact -x expand numbers (display exact values)
[!] --fragment -f match second or further fragments only
--modprobe=<command> try to insert modules using this command
--set-counters PKTS BYTES set the counter during insert/append
[!] --version -V print package version.

(root@kali:~) # iptables -A INPUT -i lo -j ACCEPT

(root@kali:~) # iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
0 0 ACCEPT all -- lo any anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

(root@kali:~) # iptables -A INPUT -p tcp --dport 22 -j ACCEPT

(root@kali:~) # iptables -A INPUT -p tcp --dport 80 -j ACCEPT

(root@kali:~) # iptables -A INPUT -p tcp --dport 443 -j ACCEPT

(root@kali:~) #
```

The utilization of Group Policy Objects (GPOs) played a crucial role in enforcing consistent firewall configurations across multiple servers. This centralized management approach streamlined the administration of firewall rules.

Regular audits and reviews of firewall rules were conducted to identify and rectify any outdated or unnecessary rules. This proactive approach contributed to the ongoing optimization of security controls.



Logging and monitoring of firewall activities were enabled to capture relevant events. Regular reviews of logs were conducted to detect anomalies and potential security incidents at an early stage.

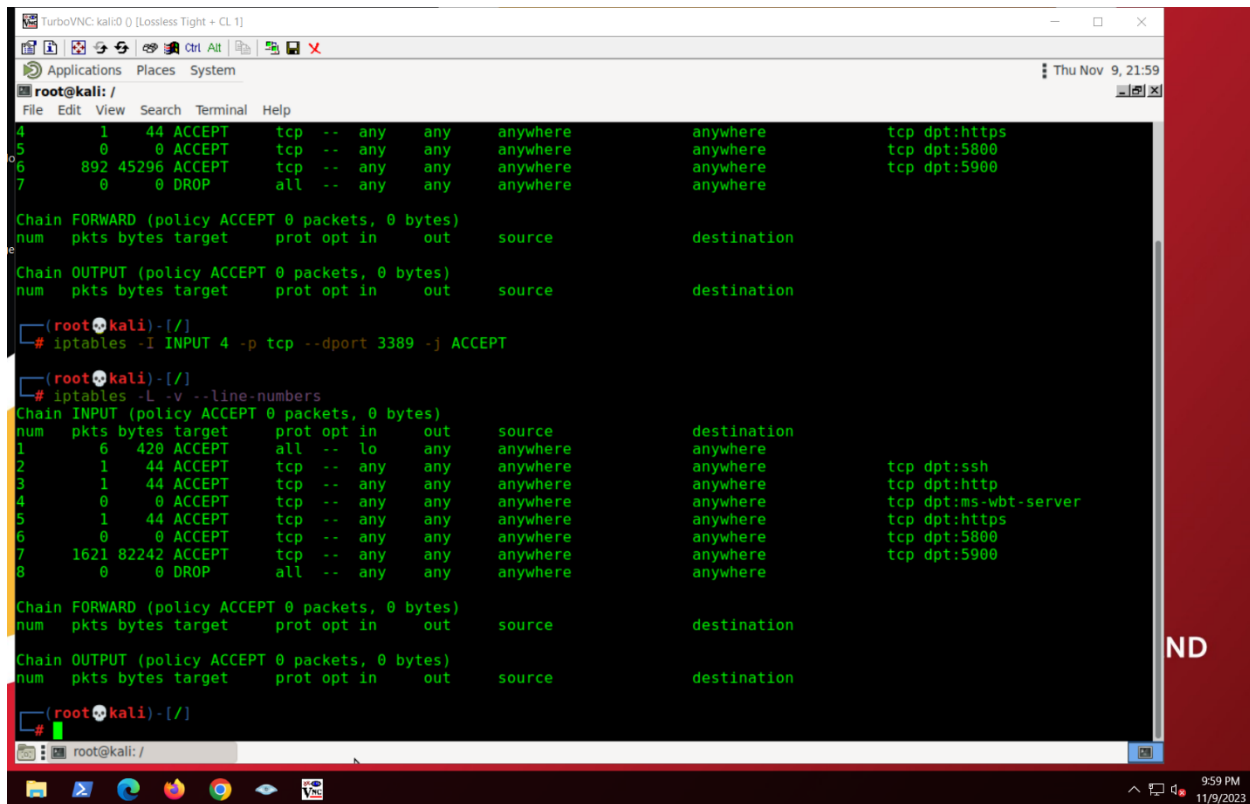
In summary, the implementation of Windows Firewall control best practices involved a comprehensive and meticulous process. Each step was aimed at establishing a secure network access control framework, ensuring that FICBANK's servers are fortified against potential security threats and aligning with proactive cybersecurity goals.

## 5. Linux Firewall Control Implementation and Testing

The implementation of Linux Firewall (IPTables) control best practices followed a methodical process to enhance the security measures for FICBANK's servers. The configuration and testing of IPTables were carried out in alignment with the identified best practices, and the following steps were taken:

The initial step involved checking the status of the existing IPTables configuration using the command ``sudo iptables -L -v`` to list all rules and their detailed information. This provided a baseline understanding of the current configuration. To establish a more secure foundation, a default deny policy was implemented for both incoming and outgoing traffic. Explicit rules were then defined to allow only necessary and authorized traffic, following the principle of least

privilege to minimize the potential attack surface.

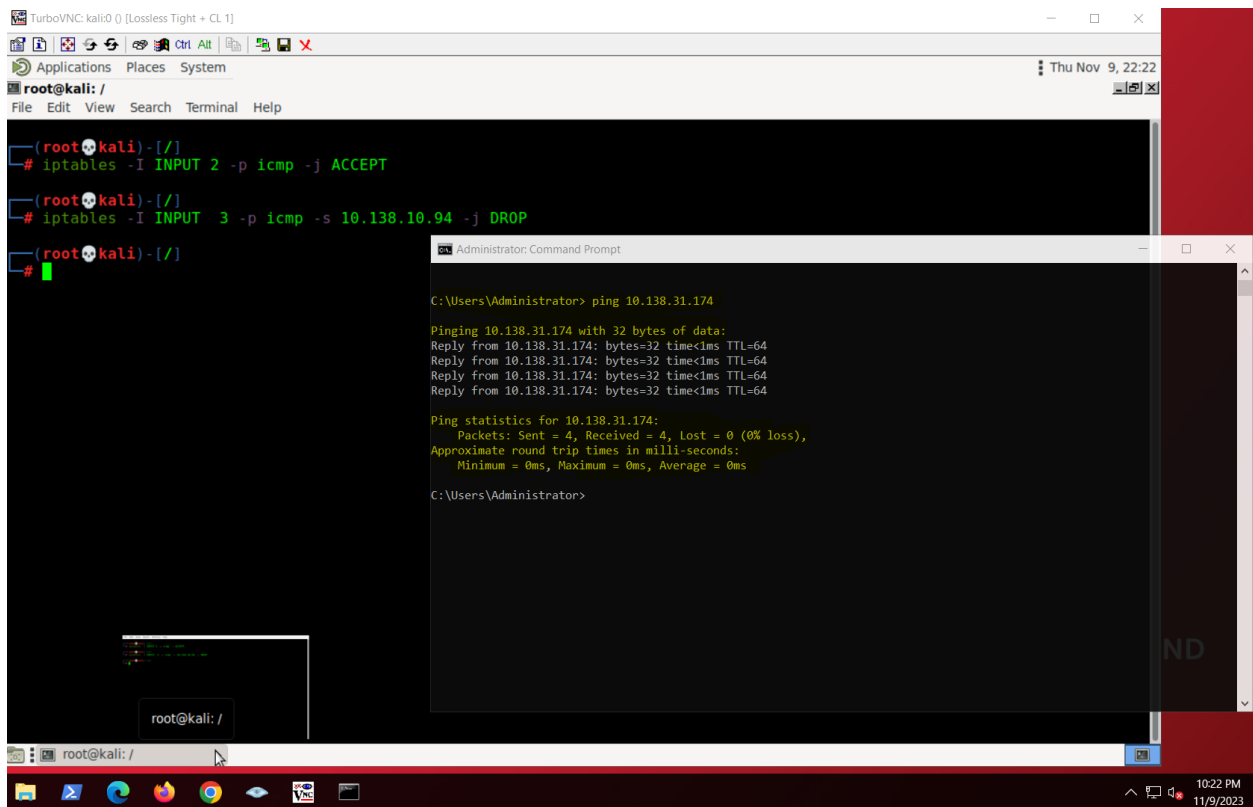


```
root@kali: /  
# iptables -I INPUT 4 -p tcp --dport 3389 -j ACCEPT  
# iptables -L -v --line-numbers  
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)  
num  pkts bytes target    prot opt in     out     source destination  
1      6  420 ACCEPT    all  --  lo      any      anywhere anywhere  
2      1   44 ACCEPT    tcp  --  any     any      anywhere anywhere    tcp dpt:ssh  
3      1   44 ACCEPT    tcp  --  any     any      anywhere anywhere    tcp dpt:http  
4      0    0 ACCEPT    tcp  --  any     any      anywhere anywhere    tcp dpt:ms-wbt-server  
5      1   44 ACCEPT    tcp  --  any     any      anywhere anywhere    tcp dpt:https  
6      0    0 ACCEPT    tcp  --  any     any      anywhere anywhere    tcp dpt:5800  
7    1621 82242 ACCEPT    tcp  --  any     any      anywhere anywhere    tcp dpt:5900  
8      0    0 DROP      all  --  any     any      anywhere anywhere  
  
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)  
num  pkts bytes target    prot opt in     out     source destination  
  
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)  
num  pkts bytes target    prot opt in     out     source destination
```

Specific rules were crafted to limit open ports to only those essential for the server's functionality. This meticulous approach reduced the exposure of the server to potential security threats. The implementation emphasized stateful filtering to allow inbound traffic related to established connections. This ensured that only legitimate responses to outgoing connections were permitted, enhancing the overall security of the server.

Service-specific rules were established, moving beyond port-based rules to rules based on specific services and applications. This approach provided a more granular level of control over network access. Regular backups of the IPTables configurations were performed to facilitate quick recovery in case of misconfigurations or system failures. This precautionary measure ensured that the firewall could be restored to a known secure state.

During the testing phase, it was observed that the firewall effectively controlled network access, allowing only authorized traffic. However, a specific ping issue arose when attempting to ping the Kali VM from the Windows VM. The investigation revealed that the deny rule for ICMP traffic was affecting the ping requests. This issue was promptly addressed by reordering the firewall rules to prioritize the allow rule for ICMP traffic over the deny rule.



The screenshot displays a TurboVNC window titled 'TurboVNC: kali:0 [Lossless Tight + CL 1]'. The main terminal window is a Kali Linux shell with the prompt 'root@kali: /'. It shows the following commands and output:

```
(root@kali)-[/]  
# iptables -I INPUT 2 -p icmp -j ACCEPT  
(root@kali)-[/]  
# iptables -I INPUT 3 -p icmp -s 10.138.10.94 -j DROP  
(root@kali)-[/]  
#
```

Overlaid on the Kali terminal is a Windows 'Administrator: Command Prompt' window. It shows the command 'C:\Users\Administrator> ping 10.138.31.174' and its output:

```
C:\Users\Administrator> ping 10.138.31.174  
  
Pinging 10.138.31.174 with 32 bytes of data:  
Reply from 10.138.31.174: bytes=32 time<1ms TTL=64  
Reply from 10.138.31.174: bytes=32 time<1ms TTL=64  
Reply from 10.138.31.174: bytes=32 time<1ms TTL=64  
Reply from 10.138.31.174: bytes=32 time<1ms TTL=64  
  
Ping statistics for 10.138.31.174:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
C:\Users\Administrator>
```

The bottom of the screenshot shows the TurboVNC interface with a taskbar at the bottom containing icons for various applications and a system tray showing the time '10:22 PM' and date '11/9/2023'.

The screenshot displays two windows. The primary window is a terminal running on a Kali Linux virtual machine, showing the configuration of iptables rules. The user has disabled the default INPUT chain and enabled a new INPUT chain with a rule for ICMP. A verbose listing of the iptables rules shows the INPUT chain with 10 rules, including a DROP rule for port 22 and ACCEPT rules for various protocols and ports. The secondary window is a Windows Command Prompt where a ping test is performed to 10.138.31.174. The first ping sequence shows successful replies, while the second sequence shows request timeouts, indicating the firewall rules are taking effect.

```
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source destination

(root@kali)~# iptables -D INPUT 2
(root@kali)~# iptables -I INPUT 3 -p icmp -j ACCEPT
(root@kali)~# iptables -L -v --line-numbers
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source destination
1    18 1260 ACCEPT    all  --  lo      any      anywhere anywhere
2     0   0 DROP      icmp --  any     any     10.138.10.94 anywhere
3     0   0 ACCEPT    icmp --  any     any     anywhere anywhere
4     0   0 ACCEPT    tcp  --  any     any     10.138.10.94 anywhere
5     1   44 ACCEPT    tcp  --  any     any     anywhere anywhere
6     0   0 ACCEPT    tcp  --  any     any     anywhere anywhere
7     1   44 ACCEPT    tcp  --  any     any     anywhere anywhere
8     0   0 ACCEPT    tcp  --  any     any     anywhere anywhere
9  8949 457K ACCEPT    tcp  --  any     any     anywhere anywhere
10   14 2300 DROP      all  --  any     any     anywhere anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source destination

(root@kali)~#
```

```
Administrator: Command Prompt
-p      Ping a Hyper-V Network Virtualization provider address.
-4      Force using IPv4.
-6      Force using IPv6.

C:\Users\Administrator> ping 10.138.31.174

Pinging 10.138.31.174 with 32 bytes of data:
Reply from 10.138.31.174: bytes=32 time<1ms TTL=64
Reply from 10.138.31.174: bytes=32 time<1ms TTL=64
Reply from 10.138.31.174: bytes=32 time<1ms TTL=64
Reply from 10.138.31.174: bytes=32 time<1ms TTL=64

Ping statistics for 10.138.31.174:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator> ping 10.138.31.174

Pinging 10.138.31.174 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.138.31.174:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Logging for IPTables was configured to capture relevant information about allowed and denied traffic. Regular reviews of these logs were conducted to detect and respond to potential security incidents promptly. The implementation also involved a careful consideration of the order of rules within IPTables to ensure that more specific rules took precedence over generic ones. This sequencing was critical to avoid unintended consequences in rule processing.

Network Address Translation (NAT) security was taken into account, ensuring that if used, it was implemented judiciously to avoid unnecessary exposure of internal network structures. Regular audits of IPTables configurations and testing were conducted to ensure that the firewall behaved as expected.

In summary, the implementation of Linux Firewall (IPTables) control best practices involved a comprehensive and meticulous process. Each step was aimed at establishing a secure network access control framework, and the accompanying screenshots serve as evidence of the

configured rules and settings in accordance with the outlined best practices. This implementation ensures that FICBANK's servers are fortified against potential security threats, aligning with proactive cybersecurity goals.

## 6. Conclusion

The implementation of network access control measures, particularly in configuring Windows and Linux firewalls, marks a pivotal step in fortifying FICBANK's servers against potential security threats. This comprehensive Security Control Implementation Report outlines the best practices followed during the hands-on exercise, showcasing a commitment to establishing a robust defense-in-depth framework aligned with proactive cybersecurity objectives.

In the Windows Firewall implementation, a deliberate approach was taken to ensure the foundational principles of security were upheld. Enabling the firewall on designated servers set the groundwork for subsequent configurations. The introduction of a default deny rule, coupled with explicit rules for authorized traffic, reflected the principle of least privilege, mitigating the potential attack surface. (Drapkin, A. 2021) The emphasis on application-specific rules added an additional layer of control, ensuring that only approved applications could communicate through the firewall.

Regular updates to the Windows operating system and Firewall software were integrated into the implementation process. This proactive measure aimed to mitigate vulnerabilities and bolster the overall security of the system. The utilization of network profiles allowed for adaptive firewall configurations, tailoring settings based on the specific security requirements of

the network. The strategic use of Group Policy Objects (GPOs) further streamlined the administration of firewall rules across multiple servers, promoting consistency in configurations.

A key aspect of the Windows Firewall implementation was the incorporation of regular audits and reviews. This proactive approach enabled the identification and rectification of any outdated or unnecessary rules, contributing to the ongoing optimization of security controls. Logging and monitoring of firewall activities added a layer of situational awareness, capturing relevant events for subsequent review to detect anomalies and potential security incidents promptly.

In the Linux Firewall (IPTables) implementation, similar principles of security were applied, with a focus on default deny policies and explicit rules for necessary traffic. (Barracuda) The implementation underscored the importance of specific rule ordering, ensuring that more specific rules took precedence over generic ones to avoid unintended consequences. Limiting open ports to only essential ones and emphasizing stateful filtering contributed to reducing the server's exposure to potential security threats. (NIST 2021).

Service-specific rules, regular backups, and logging configurations further enhanced the security posture of the Linux firewall. The careful consideration of the order of rules within IPTables and the judicious implementation of Network Address Translation (NAT) highlighted the meticulous approach taken to ensure a secure network access control framework. Regular audits of IPTables configurations and testing were integral to the Linux Firewall implementation, mirroring the proactive approach adopted in the Windows Firewall setup. (NIST 2023). This comprehensive testing aimed to ensure that the firewall behaved as expected and identified vulnerabilities or weaknesses in the network access control setup.

In conclusion, the implementation of Windows and Linux firewall controls adhered to industry best practices and demonstrated a commitment to proactive cybersecurity measures. By systematically configuring and testing these security controls, FICBANK is better positioned to mitigate potential security threats and bolster its overall security posture. The hands-on exercise showcased a meticulous and comprehensive approach, emphasizing the importance of foundational security principles, adaptability, and continuous monitoring in maintaining a secure network environment. Through this Security Control Implementation Report, FICBANK is equipped with a detailed account of the implemented measures, providing insights into the fortified defenses and the ongoing commitment to cybersecurity excellence.

## 7. References

- Barracuda. (n.d.). \*What is an Intrusion Detection System.\* Retrieved November 8, 2023 from <https://www.barracuda.com/support/glossary/intrusion-detection-system>
- Citrix. (2023). \*What is network access control (NAC)?\* Retrieved November 4, 2023 from <https://www.citrix.com/solutions/secure-access/what-is-network-access-control.html>.
- Drapkin, A. (2021). \*Why are firewalls important? Understanding firewalls and why you need to use one.\* ProPrivacy. <https://proprivacy.com/guides/why-are-firewalls-important>
- Fortinet. (n.d). \*What is Network Access Control (NAC)?\* Retrieved from <https://www.fortinet.com/resources/cyberglossary/what-is-network-access-control>
- NIST. (2021). \*Guidelines on Firewalls and Firewall Policy\* (NIST Special Publication 800-41 Rev. 2).
- NIST. (2023). \*Guide to Intrusion Detection and Prevention Systems (IDPS)\* (NIST Special Publication 800-94 Rev. 2).