

4 Host Based Firewall

A firewall is a set of rules. When a data packet moves into or out of a protected network space, its contents (in particular, information about its origin, target, and the protocol it plans to use) are tested against the firewall rules to see if it should be allowed through

To provide a Host Based Firewall, the Linux kernel includes support for:

- **Netfilter** - A set of hooks inside the Linux kernel that allows kernel modules to register callback functions with the network stack. A registered callback function is then called back for every packet that traverses the respective hook within the network stack. Includes the `ip_tables`, `ip6_tables`, `arp_tables`, and `ebtables` kernel modules. These modules are some of the significant parts of the Netfilter hook system.
- **nftables** - A subsystem of the Linux kernel providing filtering and classification of network packets/datagrams/frames. nftables is supposed to replace certain parts of Netfilter, while keeping and reusing most of it. nftables utilizes the building blocks of the Netfilter infrastructure, such as the existing hooks into the networking stack, connection tracking system, userspace queueing component, and logging subsystem. **Is available in Linux kernels 3.13 and newer.**

In order to configure firewall rules for Netfilter or nftables, a firewall utility needs to be installed. Guidance has been included for the following firewall utilities:

- UncomplicatedFirewall (**ufw**) - Provides firewall features by acting as a front-end for the Linux kernel's netfilter framework via the iptables backend. **ufw** supports both IPv4 and IPv6 networks
- **nftables** - Includes the `nft` utility for configuration of the nftables subsystem of the Linux kernel
- **iptables** - Includes the `iptables`, `ip6tables`, `arptables` and `ebtables` utilities for configuration Netfilter and the `ip_tables`, `ip6_tables`, `arp_tables`, and `ebtables` kernel modules.

Notes:

- Only **one** method should be used to configure a firewall on the system. Use of more than one method could produce unexpected results
- This section is intended only to ensure the resulting firewall rules are in place, not how they are configured

4.1 Configure a single firewall utility

Only one method should be used to configure a firewall on the system. Use of more than one method could produce unexpected results.

This section ensures that only one firewall is in use on the system and provides guidance to the subsequent subsection that should be followed for a single firewall utility configuration.

4.1.1 Ensure a single firewall configuration utility is in use (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

In Linux security, employing a single, effective firewall configuration utility ensures that only legitimate traffic gets processed, reducing the system's exposure to potential threats. The choice between **ufw**, **nftables**, and **iptables** depends on organizational needs.

Note: **iptables** is being phased out, and support for **iptables** will be reduced over time. It is recommended to transition towards either **nftables** or **ufw** as the default firewall management tool.

Rationale:

Proper configuration of a single firewall utility minimizes cyber threats and protects services and data, while avoiding vulnerabilities like open ports or exposed services. Standardizing on a single tool simplifies management, reduces errors, and fortifies security across Linux systems.

Impact:

The use of more than one firewall utility may produce unexpected results.

Audit:

Run the following script to verify that a single firewall utility is in use on the system:

```
#!/usr/bin/env bash

{
    active_firewall=() firewalls=("ufw" "nftables" "iptables")
    # Determine which firewall is in use
    for firewall in "${firewalls[@]}"; do
        case $firewall in
            nftables)
                cmd="nft" ;;
            *)
                cmd=$firewall ;;
        esac
        if command -v $cmd &> /dev/null && systemctl is-enabled --quiet $firewall && systemctl is-active --quiet $firewall; then
            active_firewall+=("$firewall")
        fi
    done
    # Display audit results
    if [ ${#active_firewall[@]} -eq 1 ]; then
        printf '%s\n' "" "Audit Results:" " ** PASS **" " - A single firewall
is in use follow the recommendation in ${active_firewall[0]} subsection ONLY"
    elif [ ${#active_firewall[@]} -eq 0 ]; then
        printf '%s\n' "" "Audit Results:" " ** FAIL **" "- No firewall in use
or unable to determine firewall status"
    else
        printf '%s\n' "" "Audit Results:" " ** FAIL **" " - Multiple firewalls
are in use: ${active_firewall[*]}"
    fi
}
```

Remediation:










Remediating to a single firewall configuration is a complex process and involves several steps. The following provides the basic steps to follow for a single firewall configuration:

1. Determine which firewall utility best fits organizational needs
2. Follow the recommendations in the subsequent subsection for the single firewall to be used
Note: Review the firewall subsection overview for the selected firewall to be used, it contains a script to simplify this process.
3. Return to this recommendation to ensure a single firewall configuration utility is in use

References:

1. <https://wiki.debian.org/DebianFirewall>
2. <https://wiki.ubuntu.com/UncomplicatedFirewall>
3. <https://assets.ubuntu.com/v1/544d9904-ubuntu-server-guide-2024-01-22.pdf>
4. <https://www.debian.org/doc/manuals/debian-reference/debian-reference.en.pdf>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 Implement and Manage a Firewall on Servers Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.2 Configure UncomplicatedFirewall

If nftables or iptables are being used in your environment, please follow the guidance in their respective section and pass-over the guidance in this section.

Uncomplicated Firewall (UFW) is a program for managing a netfilter firewall designed to be easy to use.

- Uses a command-line interface consisting of a small number of simple commands
- Uses iptables for configuration
- Rules are processed until first matching rule. The first matching rule will be applied.

Notes:

- Configuration of a live system's firewall directly over a remote connection will often result in being locked out
- Rules should be ordered so that **ALLOW** rules come before **DENY** rules.

4.2.1 Ensure ufw is installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The Uncomplicated Firewall (ufw) is a frontend for iptables and is particularly well-suited for host-based firewalls. ufw provides a framework for managing netfilter, as well as a command-line interface for manipulating the firewall

Rationale:

A firewall utility is required to configure the Linux kernel's netfilter framework via the iptables or nftables back-end.

The Linux kernel's netfilter framework host-based firewall can protect against threats originating from within a corporate network to include malicious mobile code and poorly configured software on a host.

Note: Only one firewall utility should be installed and configured. **UFW** is dependent on the iptables package

Audit:

Run the following command to verify that Uncomplicated Firewall (UFW) is installed:

```
# dpkg-query -s ufw &>/dev/null && echo "ufw is installed"
ufw is installed
```

Remediation:










Run the following command to install Uncomplicated Firewall (UFW):

```
# apt install ufw
```

References:

1. NIST SP 800-53 Rev. 5: SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.2.2 Ensure iptables-persistent is not installed with ufw (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **iptables-persistent** is a boot-time loader for netfilter rules, iptables plugin

Rationale:

Running both **ufw** and the services included in the iptables-persistent package may lead to conflict

Audit:

Run the following command to verify that the **iptables-persistent** package is not installed:

```
# dpkg-query -s iptables-persistent &>/dev/null && echo "iptables-persistent is installed"
```

Nothing should be returned

Remediation:










Run the following command to remove the **iptables-persistent** package:

```
# apt purge iptables-persistent
```

References:

1. NIST SP 800-53 Rev. 5: SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0005	M1033

4.2.3 Ensure ufw service is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

UncomplicatedFirewall (ufw) is a frontend for iptables. ufw provides a framework for managing netfilter, as well as a command-line and available graphical user interface for manipulating the firewall.

Note:

- When running `ufw enable` or starting ufw via its initscript, ufw will flush its chains. This is required so ufw can maintain a consistent state, but it may drop existing connections (eg ssh). ufw does support adding rules before enabling the firewall.
- Run the following command before running `ufw enable`.

```
# ufw allow proto tcp from any to any port 22
```

- The rules will still be flushed, but the ssh port will be open after enabling the firewall. Please note that once ufw is 'enabled', ufw will not flush the chains when adding or removing rules (but will when modifying a rule or changing the default policy)
- By default, ufw will prompt when enabling the firewall while running under ssh. This can be disabled by using `ufw --force enable`

Rationale:

The ufw service must be enabled and running in order for ufw to protect the system

Impact:

Changing firewall settings while connected over network can result in being locked out of the system.

Audit:

Run the following command to verify that the **ufw** daemon is enabled:

```
# systemctl is-enabled ufw.service  
enabled
```

Run the following command to verify that the **ufw** daemon is active:

```
# systemctl is-active ufw  
active
```

Run the following command to verify ufw is active

```
# ufw status  
Status: active
```

Remediation:

Run the following command to unmask the **ufw** daemon:

```
# systemctl unmask ufw.service
```

Run the following command to enable and start the **ufw** daemon:

```
# systemctl --now enable ufw.service  
active
```










Run the following command to enable ufw:

```
# ufw enable
```

References:

1. <http://manpages.ubuntu.com/manpages/precise/en/man8/ufw.8.html>
2. NIST SP 800-53 Rev. 5: SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0005	M1018

4.2.4 Ensure ufw loopback traffic is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Configure the loopback interface to accept traffic. Configure all other interfaces to deny traffic to the loopback network (127.0.0.0/8 for IPv4 and ::1/128 for IPv6).

Rationale:

Loopback traffic is generated between processes on machine and is typically critical to operation of the system. The loopback interface is the only place that loopback network (127.0.0.0/8 for IPv4 and ::1/128 for IPv6) traffic should be seen, all other interfaces should ignore traffic on this network as an anti-spoofing measure.

Audit:

Run the following command and verify loopback interface to accept traffic:

```
# grep -P -- 'lo|127.0.0.0' /etc/ufw/before.rules
```

Output includes:

```
# allow all on loopback
-A ufw-before-input -i lo -j ACCEPT
-A ufw-before-output -o lo -j ACCEPT
```

Run the following command and verify all other interfaces deny traffic to the loopback network (127.0.0.0/8 for IPv4 and ::1/128 for IPv6)

```
# ufw status verbose

To Action From
--
Anywhere DENY IN 127.0.0.0/8
Anywhere (v6) DENY IN ::1
```

Note: `ufw status` only shows rules added with `ufw` and not the rules found in the `/etc/ufw` rules files where allow all on loopback is configured by default.

Remediation:

Run the following commands to configure the loopback interface to accept traffic:

```
# ufw allow in on lo
# ufw allow out on lo
```

Run the following commands to configure all other interfaces to deny traffic to the loopback network:

```
# ufw deny in from 127.0.0.0/8
# ufw deny in from ::1
```










Default Value:

```
# allow all on loopback
-A ufw-before-input -i lo -j ACCEPT
-A ufw-before-output -o lo -j ACCEPT
```

References:

1. NIST SP 800-53 Rev. 5: SC-7
2. <https://manpages.ubuntu.com/manpages/jammy/en/man8/ufw-framework.8.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 Implement and Manage a Firewall on Servers Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.2.5 Ensure ufw outbound connections are configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Configure the firewall rules for new outbound connections.

Note:

- Changing firewall settings while connected over network can result in being locked out of the system.
- Unlike iptables, when a new outbound rule is added, ufw automatically takes care of associated established connections, so no rules for the latter kind are required.

Rationale:

If rules are not in place for new outbound connections all packets will be dropped by the default policy preventing network usage.

Audit:

Run the following command and verify all rules for new outbound connections match site policy:

```
# ufw status numbered
```

Remediation:










Configure ufw in accordance with site policy. The following commands will implement a policy to allow all outbound connections on all interfaces:

```
# ufw allow out on all
```

References:

1. NIST SP 800-53 Rev. 5: SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.2.6 Ensure ufw firewall rules exist for all open ports (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Services and ports can be accepted or explicitly rejected.

Note:

- Changing firewall settings while connected over network can result in being locked out of the system
- The remediation command opens up the port to traffic from all sources. Consult ufw documentation and set any restrictions in compliance with site policy

Rationale:

To reduce the attack surface of a system, all services and ports should be blocked unless required.

- Any ports that have been opened on non-loopback addresses need firewall rules to govern traffic.
- Without a firewall rule configured for open ports, the default firewall policy will drop all packets to these ports.
- Required ports should have a firewall rule created to allow approved connections in accordance with local site policy.
- Unapproved ports should have an explicit deny rule created.

Audit:

Run the following script to verify a firewall rule exists for all open ports:

```
#!/usr/bin/env bash

{
    unset a_ufwout; unset a_openports
    while read -r l_ufwport; do
        [ -n "$l_ufwport" ] && a_ufwout+=("$l_ufwport")
    done <<(ufw status verbose | grep -Po '^h*\d+\b' | sort -u)
    while read -r l_openport; do
        [ -n "$l_openport" ] && a_openports+=("$l_openport")
    done <<(ss -tuln | awk '($5!~/%lo:/ && $5!~/127.0.0.1:/ && $5!~/\[?::1\]?:/) {split($5, a, ":"); print a[2]}' | sort -u)
    a_diff=$(printf '%s\n' "${a_openports[@]}" "${a_ufwout[@]}"
"${a_ufwout[@]}" | sort | uniq -u)
    if [[ -n "$a_diff[*]" ]]; then
        echo -e "\n- Audit Result:\n  ** FAIL **\n- The following port(s) don't
have a rule in UFW: $(printf '%s\n' \n"${a_diff[*]}")\n- End List"
    else
        echo -e "\n- Audit Passed -\n- All open ports have a rule in UFW\n"
    fi
}
```

Remediation:

For each port identified in the audit which does not have a firewall rule, evaluate the service listening on the port and add a rule for accepting or denying inbound connections in accordance with local site policy:

Examples:

```
# ufw allow in <port>/<tcp or udp protocol>
# ufw deny in <port>/<tcp or udp protocol>
```

Note: Examples create rules for from any, to any. More specific rules should be centered when allowing inbound traffic e.g only traffic from this network.







Example to allow traffic on port 443 using the tcp protocol from the 192.168.1.0 network:

```
ufw allow from 192.168.1.0/24 to any proto tcp port 443
```

References:

1. NIST SP 800-53 Rev. 5: SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 Implement and Manage a Firewall on Servers Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.2.7 Ensure ufw default deny firewall policy (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

A default deny policy on connections ensures that any unconfigured network usage will be rejected.

Note: Any port or protocol without an explicit allow before the default deny will be blocked

Rationale:

With a default accept policy the firewall will accept any packet that is not configured to be denied. It is easier to allow list acceptable usage than to deny list unacceptable usage.

Impact:

Any port and protocol not explicitly allowed will be blocked. The following rules should be considered before applying the default deny.

```
ufw allow out http
ufw allow out https
ufw allow out ntp # Network Time Protocol
ufw allow out to any port 53 # DNS
ufw allow out to any port 853 # DNS over TLS
ufw logging on
```

Audit:

Run the following command and verify that the default policy for **incoming** , **outgoing** , and **routed** directions is **deny** , **reject** , or **disabled**:

```
# ufw status verbose | grep Default:
```

Example output:

```
Default: deny (incoming), deny (outgoing), disabled (routed)
```

Remediation:










Run the following commands to implement a default *deny* policy:

```
# ufw default deny incoming
# ufw default deny outgoing
# ufw default deny routed
```

References:

1. NIST SP 800-53 Rev. 5: SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.3 Configure nftables

If Uncomplicated Firewall (UFW) or iptables are being used in your environment, please follow the guidance in their respective section and pass-over the guidance in this section.

nftables is a subsystem of the Linux kernel providing filtering and classification of network packets/datagrams/frames and is the successor to iptables. The biggest change with the successor **nftables** is its simplicity. With iptables, we have to configure every single rule and use the syntax which can be compared with normal commands. With **nftables**, the simpler syntax, much like BPF (Berkely Packet Filter) means shorter lines and less repetition. Support for **nftables** should also be compiled into the kernel, together with the related **nftables** modules. Please ensure that your kernel supports nf_tables before choosing this option.

Notes:

- This section broadly assumes starting with an empty **nftables** firewall ruleset (established by flushing the rules with nft flush ruleset).
- Remediation steps included only affect the live system, you will also need to configure your default firewall configuration to apply on boot.
- Configuration of a live systems firewall directly over a remote connection will often result in being locked out. It is advised to have a known good firewall configuration set to run on boot and to configure an entire firewall structure in a script that is then run and tested before saving to boot.

The following script will implement the firewall rules of this section and open port 22(ssh) from anywhere. Opening the ports for port 22(ssh) needs to be updated in accordance with local site policy. **Allow port 22(ssh) needs to be updated to only allow systems requiring ssh connectivity to connect, as per site policy.**

Save the script below as **/etc/nftables.rules**


```
#!/sbin/nft -f

# This nftables.rules config should be saved as /etc/nftables.rules
# flush nftables ruleset
flush ruleset
# Load nftables ruleset
# nftables config with inet table named filter
table inet filter {
    # Base chain for input hook named input (Filters inbound network
    packets)
    chain input {
        type filter hook input priority 0; policy drop;

        # Ensure loopback traffic is configured
        iif "lo" accept
        ip saddr 127.0.0.0/8 counter packets 0 bytes 0 drop
        # If IPv6 is enabled on the system ensure IPv6 loopback
        traffic is configured
        ip6 saddr ::1 counter packets 0 bytes 0 drop

        # Ensure established connections are configured
        ip protocol tcp ct state established accept
        ip protocol udp ct state established accept

        # Accept port 22 (SSH) traffic from anywhere
        tcp dport ssh accept
    }

    # Base chain for hook forward named forward (Filters forwarded
    network packets)
    chain forward {
        type filter hook forward priority 0; policy drop;
    }

    # Base chain for hook output named output (Filters outbound network
    packets)
    chain output {
        type filter hook output priority 0; policy drop;
        # Ensure outbound and established connections are configured
        ip protocol tcp ct state established,related,new accept
        ip protocol udp ct state established,related,new accept
    }
}
```

Run the following command to load the file into nftables

```
# nft -f /etc/nftables.rules
```

All changes in the nftables subsections are temporary.

To make these changes permanent:

Run the following command to create the nftables.rules file

```
nft list ruleset > /etc/nftables.rules
```

Add the following line to **/etc/nftables.conf**

```
include "/etc/nftables.rules"
```

4.3.1 Ensure nftables is installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

nftables provides a new in-kernel packet classification framework that is based on a network-specific Virtual Machine (VM) and a new nft userspace command line tool. nftables reuses the existing Netfilter subsystems such as the existing hook infrastructure, the connection tracking system, NAT, userspace queuing and logging subsystem.

Notes:

- nftables is available in Linux kernel 3.13 and newer
- Only one firewall utility should be installed and configured
- Changing firewall settings while connected over the network can result in being locked out of the system

Rationale:

nftables is a subsystem of the Linux kernel that can protect against threats originating from within a corporate network to include malicious mobile code and poorly configured software on a host.

Audit:

Run the following command to verify that **nftables** is installed:

```
# dpkg-query -s nftables &>/dev/null && echo "nftables is installed"
nftables is installed
```

Remediation:










Run the following command to install **nftables**:

```
# apt install nftables
```

References:

1. NIST SP 800-53 Rev. 5: CA-9

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.3.2 Ensure ufw is uninstalled or disabled with nftables (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Uncomplicated Firewall (UFW) is a program for managing a netfilter firewall designed to be easy to use.

Rationale:

Running both the **nftables** service and **ufw** may lead to conflict and unexpected results.

Audit:

Run the following commands to verify that **ufw** is **either** not installed or inactive. Only one of the following needs to pass.

Run the following command to verify that **ufw** is not installed:

```
# dpkg-query -s ufw &>/dev/null && echo "ufw is installed"
```

Nothing should be returned

-OR-

Run the following commands to verify **ufw** is disabled and **ufw.service** is not enabled:

```
# ufw status

Status: inactive
# systemctl is-enabled ufw.service

masked
```

Remediation:

Run **one** of the following to either remove **ufw** or disable **ufw** and mask **ufw.service**:
Run the following command to remove **ufw**:

```
# apt purge ufw
```

-OR-

Run the following commands to disable **ufw** and mask **ufw.service**:










```
# ufw disable
# systemctl stop ufw.service
# systemctl mask ufw.service
```

Note: **ufw disable** needs to be run before **systemctl mask ufw.service** in order to correctly disable **UFW**

References:

1. NIST SP 800-53 Rev. 5: SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 Implement and Manage a Firewall on Servers Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0005	M1033

4.3.3 Ensure iptables are flushed with nftables (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

nftables is a replacement for iptables, ip6tables, ebtables and arptables

Rationale:

It is possible to mix iptables and nftables. However, this increases complexity and also the chance to introduce errors. For simplicity flush out all iptables rules, and ensure it is not loaded

Audit:

Run the following commands to ensure no iptables rules exist

For iptables:

```
# iptables -L
```

No rules should be returned

For ip6tables:

```
# ip6tables -L
```

No rules should be returned

Remediation:

Run the following commands to flush iptables:

For iptables:

```
# iptables -F
```










For ip6tables:

```
# ip6tables -F
```

References:

1. NIST SP 800-53 Rev. 5: CA-9, SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0005	

4.3.4 Ensure a nftables table exists (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Tables hold chains. Each table only has one address family and only applies to packets of this family. Tables can have one of five families.

Rationale:

nftables doesn't have any default tables. Without a table being built, nftables will not filter network traffic.

Impact:

Adding rules to a running nftables can cause loss of connectivity to the system

Audit:

Run the following command to verify that a nftables table exists:

```
# nft list tables
```

Return should include a list of nftables:

Example:

```
table inet filter
```

Remediation:

Run the following command to create a table in nftables

```
# nft create table inet <table name>
```










Example:

```
# nft create table inet filter
```

References:

1. NIST SP 800-53 Rev. 5: CA-9, SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1047

4.3.5 Ensure nftables base chains exist (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Chains are containers for rules. They exist in two kinds, base chains and regular chains. A base chain is an entry point for packets from the networking stack, a regular chain may be used as jump target and is used for better rule organization.

Rationale:

If a base chain doesn't exist with a hook for input, forward, and delete, packets that would flow through those chains will not be touched by nftables.

Impact:

If configuring nftables over ssh, **creating a base chain** with a policy of **drop** will cause loss of connectivity.

Ensure that a rule allowing ssh has been added to the base chain prior to setting the base chain's policy to drop

Audit:

Run the following commands and verify that base chains exist for **INPUT**.

```
# nft list ruleset | grep 'hook input'

type filter hook input priority 0;
```

Run the following commands and verify that base chains exist for **FORWARD**.

```
# nft list ruleset | grep 'hook forward'

type filter hook forward priority 0;
```

Run the following commands and verify that base chains exist for **OUTPUT**.

```
# nft list ruleset | grep 'hook output'

type filter hook output priority 0;
```

Remediation:










Run the following command to create the base chains:

```
# nft create chain inet <table name> <base chain name> { type filter hook <(input|forward|output)> priority 0 \; }
```

Example:

```
# nft create chain inet filter input { type filter hook input priority 0 \; }  
# nft create chain inet filter forward { type filter hook forward priority 0 \; }  
# nft create chain inet filter output { type filter hook output priority 0 \; }
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0005	M1047

4.3.6 Ensure nftables loopback traffic is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Configure the loopback interface to accept traffic. Configure all other interfaces to deny traffic to the loopback network

Rationale:

Loopback traffic is generated between processes on machine and is typically critical to the operation of the system. The loopback interface is the only place that loopback network traffic should be seen, all other interfaces should ignore traffic on this network as an anti-spoofing measure.

Audit:

Run the following commands to verify that the loopback interface is configured:
Run the following command to verify the loopback interface is configured to accept network traffic:

```
# nft list ruleset | awk '/hook input/,/}/' | grep 'iif "lo" accept'
```

Example output:

```
iif "lo" accept
```

Run the following command to verify network traffic from an IPv4 loopback interface is configured to drop:

```
# nft list ruleset | awk '/hook input/,/}/' | grep 'ip saddr'
```

Example output:

```
ip saddr 127.0.0.0/8 counter packets 0 bytes 0 drop
```

- IF - IPv6 is enabled on the system:

Run the following command to verify network traffic from an IPv6 loopback interface is configured to drop:

```
# nft list ruleset | awk '/hook input/,/}/' | grep 'ip6 saddr'
```

Example output:

```
ip6 saddr ::1 counter packets 0 bytes 0 drop
```

Remediation:

Run the following commands to implement the loopback rules:

```
# nft add rule inet filter input iif lo accept
# nft add rule inet filter input ip saddr 127.0.0.0/8 counter drop
```

- **IF** - IPv6 is enabled on the system:










Run the following command to implement the IPv6 loopback rule:

```
# nft add rule inet filter input ip6 saddr ::1 counter drop
```

References:

1. NIST SP 800-53 Rev. 5: CA-9, SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 Implement and Manage a Firewall on Servers Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0005	

4.3.7 Ensure nftables outbound and established connections are configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Configure the firewall rules for new outbound, and established connections

Rationale:

If rules are not in place for new outbound, and established connections all packets will be dropped by the default policy preventing network usage.

Audit:

Run the following commands and verify all rules for established incoming connections match site policy: site policy:

```
# nft list ruleset | awk '/hook input/,/}/' | grep -E 'ip protocol (tcp|udp)
ct state'
```

Output should be similar to:

```
ip protocol tcp ct state established accept
ip protocol udp ct state established accept
```

Run the following command and verify all rules for new and established outbound connections match site policy

```
# nft list ruleset | awk '/hook output/,/}/' | grep -E 'ip protocol (tcp|udp)
ct state'
```

Output should be similar to:

```
ip protocol tcp ct state established,related,new accept
ip protocol udp ct state established,related,new accept
```

Remediation:










Configure nftables in accordance with site policy. The following commands will implement a policy to allow all outbound connections and all established connections:

```
# nft add rule inet filter input ip protocol tcp ct state established accept
# nft add rule inet filter input ip protocol udp ct state established accept
# nft add rule inet filter output ip protocol tcp ct state
new,related,established accept
# nft add rule inet filter output ip protocol udp ct state
new,related,established accept
```

References:

1. NIST SP 800-53 Rev. 5: CA-9, SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 Implement and Manage a Firewall on Servers Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562	TA0011	M1031, M1037

4.3.8 Ensure nftables default deny firewall policy (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Base chain policy is the default verdict that will be applied to packets reaching the end of the chain.

Rationale:

There are two policies: accept (Default) and drop. If the policy is set to **accept**, the firewall will accept any packet that is not configured to be denied and the packet will continue transversing the network stack.

It is easier to allow list acceptable usage than to deny list unacceptable usage.

Note:

- Allow port 22(ssh) needs to be updated to only allow systems requiring ssh connectivity to connect, as per site policy.
- Changing firewall settings while connected over network can result in being locked out of the system.

Impact:

If configuring nftables over ssh, creating a base chain with a policy of drop will cause loss of connectivity.

Ensure that a rule allowing ssh has been added to the base chain prior to setting the base chain's policy to drop

Audit:

Run the following commands and verify that base chains contain a policy of **DROP**.

```
# nft list ruleset | grep 'hook input'

type filter hook input priority 0; policy drop;
# nft list ruleset | grep 'hook forward'

type filter hook forward priority 0; policy drop;
# nft list ruleset | grep 'hook output'

type filter hook output priority 0; policy drop;
```

Remediation:

Run the following command for the base chains with the input, forward, and output hooks to implement a default DROP policy:

```
# nft chain <table family> <table name> <chain name> { policy drop \; }
```

Example:

```
# nft chain inet filter input { policy drop \; }

# nft chain inet filter forward { policy drop \; }

# nft chain inet filter output { policy drop \; }
```










Default Value:

accept

References:

1. Manual Page nft
2. NIST SP 800-53 Rev. 5: CA-9, SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.3.9 Ensure nftables service is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The nftables service allows for the loading of nftables rulesets during boot, or starting on the nftables service

Rationale:

The nftables service restores the nftables rules from the rules files referenced in the `/etc/nftables.conf` file during boot or the starting of the nftables service

Audit:

Run the following command and verify that the nftables service is enabled:

```
# systemctl is-enabled nftables  
enabled
```

Remediation:










Run the following command to enable the nftables service:

```
# systemctl enable nftables
```

References:

1. NIST SP 800-53 Rev. 5: CA-9, SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 <u>Implement and Manage a Firewall on End-User Devices</u> Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 <u>Apply Host-based Firewalls or Port Filtering</u> Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031, M1037

4.3.10 Ensure nftables rules are permanent (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

nftables is a subsystem of the Linux kernel providing filtering and classification of network packets/datagrams/frames.

The nftables service reads the `/etc/nftables.conf` file for a nftables file or files to include in the nftables ruleset.

A nftables ruleset containing the input, forward, and output base chains allow network traffic to be filtered.

Note: Saving the script and following the instruction in the Configure nftables section overview will implement the rules in the configure nftable section, open port 22(ssh) from anywhere, and applies nftables ruleset on boot.

Rationale:

Changes made to nftables ruleset only affect the live system, you will also need to configure the nftables ruleset to apply on boot

Audit:

Run the following commands to verify that input, forward, and output base chains are configured to be applied to a nftables ruleset on boot:

Run the following command to verify the input base chain:

```
# [ -n "$(grep -E '^s*include' /etc/nftables.conf)" ] && awk '/hook
input/,,/' $(awk '$1 ~ /^s*include/ { gsub("\"", "", $2); print $2 }'
/etc/nftables.conf)
```

Output should be similar to:

```
type filter hook input priority 0; policy drop;

# Ensure loopback traffic is configured
iif "lo" accept
ip saddr 127.0.0.0/8 counter packets 0 bytes 0 drop
ip6 saddr ::1 counter packets 0 bytes 0 drop

# Ensure established connections are configured
ip protocol tcp ct state established accept
ip protocol udp ct state established accept

# Accept port 22 (SSH) traffic from anywhere
tcp dport ssh accept
```

Review the input base chain to ensure that it follows local site policy

Run the following command to verify the forward base chain:

```
# [ -n "$(grep -E '^s*include' /etc/nftables.conf)" ] && awk '/hook
forward/,,/' $(awk '$1 ~ /^s*include/ { gsub("\"", "", $2); print $2 }'
/etc/nftables.conf)
```

Output should be similar to:

```
# Base chain for hook forward named forward (Filters forwarded
network packets)
chain forward {
    type filter hook forward priority 0; policy drop;
}
```

Review the forward base chain to ensure that it follows local site policy.

Run the following command to verify the forward base chain:

```
# [ -n "$(grep -E '^s*include' /etc/nftables.conf)" ] && awk '/hook
output/,,/' $(awk '$1 ~ /^s*include/ { gsub("\"", "", $2); print $2 }'
/etc/nftables.conf)
```

Output should be similar to:

```
# Base chain for hook output named output (Filters outbound network
packets)
chain output {
    type filter hook output priority 0; policy drop;
    # Ensure outbound and established connections are configured
    ip protocol tcp ct state established,related,new accept
    ip protocol udp ct state established,related,new accept
}
```

Review the output base chain to ensure that it follows local site policy.

Remediation:

Edit the `/etc/nftables.conf` file and un-comment or add a line with **include** `<Absolute path to nftables rules file>` for each nftables file you want included in the nftables ruleset on boot

Example:

```
# vi /etc/nftables.conf
```










Add the line:

```
include "/etc/nftables.rules"
```

References:

1. NIST SP 800-53 Rev. 5: CA-9, SC-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 Implement and Manage a Firewall on Servers Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.004	TA0011	M1031